symmetry

MDPI

Article

Semantic Image Segmentation with Deep Convolutional Neural Networks and Quick Shift

Sanxing Zhang [1,2,†], Zhenhuan Ma [1,2,†], Gang Zhang [1,2], Tao Lei [1,*], Rui Zhang [1,2] and Yi Cui [1]

[1] Institute of Optics and Electronics Chinese Academy of Science, Chengdu 610209, China; sanxingZhang27@163.com (S.Z.); mazhenhuan17@mails.ucas.edu.cn (Z.M.); zhanggang@ioe.ac.cn (G.Z.); zhangrui182@mails.ucas.ac.cn (R.Z.); cuiyi@ioe.ac.cn (Y.C.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: taoleiyan@ioe.ac.cn
† These authors contributed equally to this work.

Received: 6 February 2020; Accepted: 3 March 2020; Published: 6 March 2020

Abstract: Semantic image segmentation, as one of the most popular tasks in computer vision, has been widely used in autonomous driving, robotics and other fields. Currently, deep convolutional neural networks (DCNNs) are driving major advances in semantic segmentation due to their powerful feature representation. However, DCNNs extract high-level feature representations by strided convolution, which makes it impossible to segment foreground objects precisely, especially when locating object boundaries. This paper presents a novel semantic segmentation algorithm with DeepLab v3+ and super-pixel segmentation algorithm-quick shift. DeepLab v3+ is employed to generate a class-indexed score map for the input image. Quick shift is applied to segment the input image into superpixels. Outputs of them are then fed into a class voting module to refine the semantic segmentation results. Extensive experiments on proposed semantic image segmentation are performed over PASCAL VOC 2012 dataset, and results that the proposed method can provide a more efficient solution.

Keywords: semantic segmentation; deep convolutional neural network; superpixel; quick shift; class voting

1. Introduction

Semantic image segmentation is a typical computer vision problem. Its task is to assign different categories to each pixel in an image according to the object of interest [1]. In the past several years, due to a large amount of training images and high-performance GPUs, deep learning techniques-in particular, supervised approaches such as deep convolutional neural networks (DCNNs)-have achieved relentless success in various high-level computer vision tasks, such as image classification, object detection, semantic segmentation, etc. [2–4]. The key advantage of these deep learning techniques is to learn high-level feature representations in an end-to-end fashion, which are more discriminative than traditional ones. Inspired by the success of deep learning techniques in image classification tasks, researchers explored the capabilities of such networks for pixel-level annotations and proposed many prominent deep learning networks for semantic segmentation.

Nowadays, most DCNNs for semantic segmentation are based on a common pioneer: fully convolutional network (FCN) proposed by Long et al. [5]. It transforms the well-known DCNNs used for image classification, such as AlexNet [6], VGG [7], GoogleNet [8], into fully convolutional ones by replacing the fully connected layers with convolutional ones in order to output spatial feature maps instead of classification probabilities. Those feature maps are then decoded [5] to produce

dense pixel-level annotations. FCN is considered a milestone in deep learning techniques for semantic segmentation, since it demonstrates how DCNNs can be trained end-to-end to solve this problem, efficiently learning how to produce dense pixel-level predictions for input of arbitrary sizes. It achieved 20% relative improvement in segmentation accuracy over traditional methods on the PASCAL VOC 2012 dataset [9].

DeepLab series [10–13] are successful and popular in DCNNs based semantic segmentation model. DeepLab v1 [10] introduces atrous convolution [14] in DCNN to effectively enlarge the receptive field without increasing the number of network parameters. To localize objects boundaries, it combines the last layer of DCNN with fully connected CRF [15]. Due to these two advanced techniques, it reached 71.6% mIOU accuracy in the PASCAL VOL 2012 dataset. Based on DeepLab v1, DeepLab v2 [11] further proposes atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales. By employing multiple parallel atrous convolutional layers with different dilation rates, ASPP can exploit multi-scale features, thus capturing objects as well as image context at multiple scales. DeepLab v2 combines atrous convolution, ASPP and fully connected CRF, achieving 79.9% mIOU accuracy in the PASCAL VOC 2012 dataset. DeepLab v3 [12] incorporates improved ASPP, batch normalization and a better way to encode multi-scale context to further improve performance, reaching 85.7% mIOU accuracy in the PASCAL VOC 2012 dataset. Improved ASPP involves concatenation of image-level features, a 1x1 convolution and three 3x3 atrous convolution with different dilation rates. Batch normalization is used after each of the parallel convolution layers. Fully connected CRF is abandoned from Deeplab v3. In Deeplab v3+ [13], ASPP and Encoder-Decoder structure are used. The Decoder module refines the segmentation results in pixel-level. Deeplab v3+ further explores the Xception model [16] and applies the depthwise separable convolution [17] to ASPP and the Decoder module. In PASCAL VOC 2012 *test* set and Cityscapes datasets, it achieved performance of 89.0% and 82.1% separately.

Superpixel has been commonly used as a preprocessing step for image segmentation since it was first introduced by Ren et al. [18] in 2003, because it reduces the number of inputs for subsequent processing steps and adheres well to objects boundaries. It groups pixels into perceptually meaningful atomic regions, thus can enable feature computing on a meaningful image representation. In the past decades, a large number of superpixel algorithms have been proposed. Quick shift [19] is a mode-seeking based clustering algorithm, which has a relatively good boundary adherence. It first initializes the segmentation using medoid shift [20], then moves each data point in the feature space to the nearest neighbor that increases the Parzen density estimation [21]. Simple Linear Iterative Clustering (SLIC) [22] adopts a k-means clustering approach with a distance metric that depends on both spatial and intensity differences to efficiently generate superpixels. Felzenszwalb [23] is a graph-based approach used for image segmentation. It performs an agglomerative clustering of pixels as nodes on a graph such that each superpixel is the minimum spanning tree of the constituent pixels.

Although DeepLab v3+ has achieved good performance in semantic image segmentation, it still has some shortcomings. One of the main problems is that it adopts DCNN for semantic segmentation, which consists of strided pooling and convolution layers. They increase receptive field but aggregate the context information while discarding the boundary information. However, semantic segmentation needs the exact alignment of class maps and thus, needs the boundary information to be preserved. To ttackle the challenging problem, this paper presents a novel method to refine the object boundaries of the segmentation results output from DeepLab v3+, which unites the benefits of DeepLab v3+ with the superpixel segmentation algorithm-quick shift [19]. The main methods are as follows: (i) Using DeepLab v3+ to obtain the class-indexed score map of the same size of the input image; (ii) Segmenting the input image into superpixels by quick shift; (iii) Inputing the output of these two modules into the category voting module to refine the object boundary of the segmentation result. The proposed method in this paper improves the semantic segmentation results both qualitatively and quantitatively, especially on object boundaries. Experiments on the PASCAL VOC 2012 dataset verify the effectiveness of the proposed method.

The paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 presents the experimental results of the proposed method on the PASCAL VOC 2012 dataset, and comparisons with other methods. Section 4 discusses and analyses the experimental results. Finally, conclusions are drawn in Section 5.

2. Methodology

In this section, a robust framework is proposed for semantic image segmentation. It captures the class index score map by DeepLab v3+ and segments the input image into superpixels by quick shift. The object boundary of the segmentation result is refined by a class voting module.

2.1. Motivation

(1) It is hard for DCNNs based semantic segmentation methods to produce semantic segmentation results with accurate objects boundaries. There are two main reasons. First, the memory of GPU is limited, so DCNNs should adopt strided pooling and convolution to reduce parameters. Second, it is difficult to assign labels for pixels on objects boundaries because the cascaded feature maps generated by DCNNs blur them. In order to precisely segment foreground objects from background, DCNNs should have the following two properties. First, it should classify objects boundaries precisely. Second, for pixels on objects boundaries, class score computed for the target class should be close to class scores of other classes.

As shown in Figure 1, DCNN centers on the red and yellow points of the input image, respectively, to extract features in their respective receptive fields. At the aeroplane's boundaries, the image regions corresponding to the aeroplane and the background pixels have a great overlap, resulting in features extracted by DCNN are very similar and therefore difficult to classify pixels on aeroplane's boundaries.

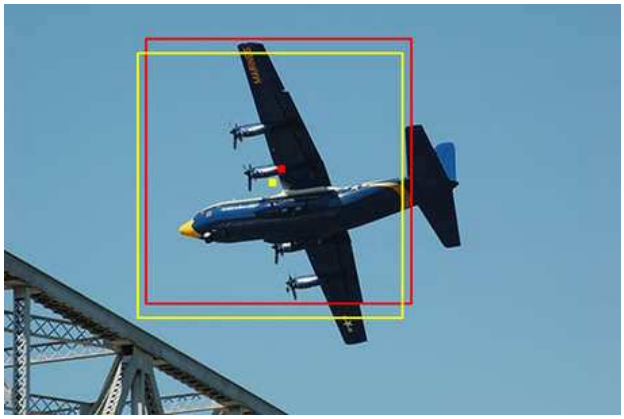


Figure 1. Two image areas processed by DCNNs.

The softmax loss function used for semantic segmentation is often simply formulated as:

$$loss = -log\frac{e^{x_k}}{\sum_{i=0}^{N} e^{x_i}} \tag{1}$$

where N is the total number of classes, and k is the target class. In the training iteration, DCNNs minimize loss, i.e., maximize x_k. In order to obtain objects boundaries accurately by interpolation, DCNNs should consider not only class score of the target class but also class scores of other classes. The loss function aforementioned only tries to maximize class score of the target class, while ignoring class scores of other classes, so it is difficult for DCNNs to output a proper score for each class.

(2) In an image, a foreground object is often composed of a series of regions. Inside these regions, its color, lightness and texture have little changes. DCNNs based semantic segmentation methods directly classify every pixel in the image and have no idea that these regions belong to the same

object. Figure 2 is a semantic segmentation result output by DeepLab v3+. From it, we can see that the areas labeled with green color are segmented separately from the regions they should belong to, which is incorrectly segmented. In order to tackle this problem, we employ a region-based method to postprocess semantic segmentation results of DeepLab v3+.

Figure 2. An example of DeepLab v3+ incorrectly segmenting areas from regions they should belong to. (In order to better demonstrate this phenomenon, we overlap the segmentation result of DeepLab v3+ on the raw image).

2.2. Main Process

Framework of the proposed method is shown in Figure 3. It consists of the following three modules: (a) DeepLab v3+, (b) superpixel segmentation-quick shift, and (c) Class Voting. In (a), we use DeepLab v3+ to obtain a class-indexed score map for the input image. In this score map, each pixel in the input image is marked with an index corresponding to its predicted class. In (b), we use quick shift algorithm to segment the image into superpixels. It outputs the superpixel index of each pixel in the image. Then, the outputs from (a) and (b) are fed into (c) to obtain the refined semantic segmentation results for the input image.

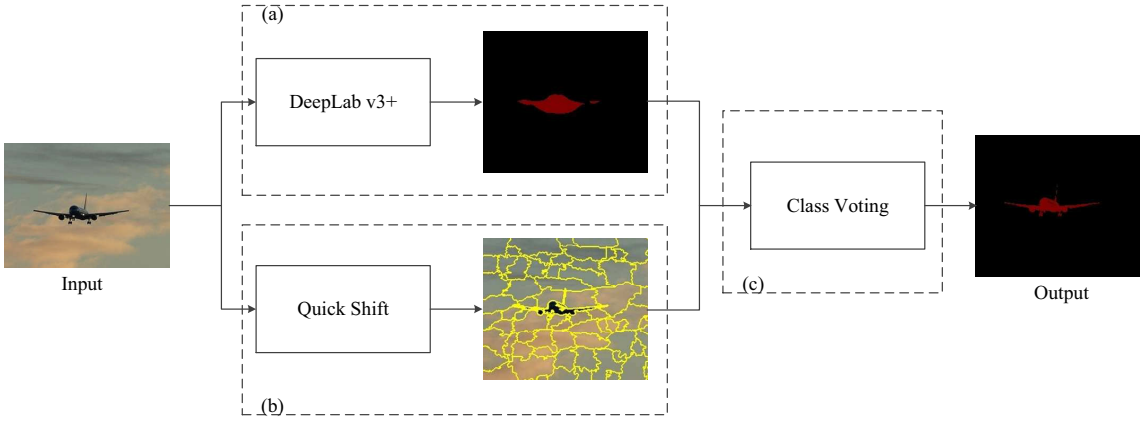


Figure 3. Framework of the proposed method.

2.3. DeepLab v3+ Recap

2.3.1. Architecture

DeepLab v3+ [13] was proposed by Cheng et al. in 2018, which has achieved state-of-the-art performance in PASCAL VOC 2012 dataset. As shown in Figure 4, DeepLab v3+ is a novel Encoder-Decoder architecture which employs DeepLab v3 [12] as Encoder module and a simple

yet effective Decoder module. It applies ResNet-101 as backbone, adopts atrous convolution in deep layers to enlarge receptive field. After ResNet-101, an ASPP module is on the top of it to aggregate the multi-scale contextual information. The Decoder concatenates the low-level features from ResNet-101 with upsampled deep-level multi-scale features extracted from ASPP. Finally, it upsamples the concatenated feature maps to produce the final semantic segmentation results.

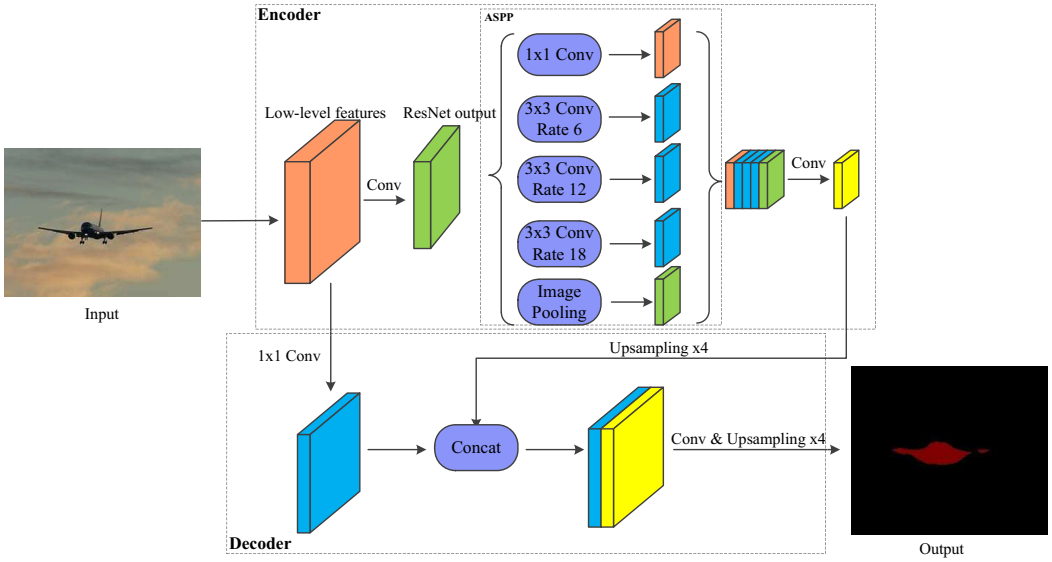


Figure 4. Encoder-Decoder architecture of DeepLab v3+.

2.3.2. Training Details

We implement DeepLab v3+ with PyTorch and experiment it on PASCAL VOC 2012 dataset. The implemented model using $outputstride = 16$ during training and evaluation, without using multi-scale and left-right flipped inputs [13]. The dataset contains 20 foreground object classes and one background class. It officially consists of 1,464 images in *train* set and 1449 images in *val* set. We also augment the dataset with additional annotations provided by [24], resulting in a total of 10,582 training images. These 10,582 images are used as the *train_aug* set to train the model following the training strategy in DeepLab v3+ [13] and the *val* set for evaluation.

In DeepLab v3+, it preprocesses input images by resizing and cropping them to fixed size of 513×513. When computing the value of mIOU, the annotated objects boundaries in ground-truths are not taken into consideration. Thus, mIOU can not be used to evaluate whether the pixels on objects boundaries are classified right or not. In order to evaluate the accuracy of the proposed method in localizing objects boundaries, we compute the value of mIOU of the implemented DeepLab v3+ in two steps. First, we follow the same training process as in [13] with fixed input image size of 513×513. Second, we recompute the value of mIOU with the model trained in the first step, but the inputs of the model are images of arbitrary sizes and without preprocessing. At the same time, we labeled objects boundaries in ground-truths as background when computing the value of mIOU. In the first step, the model we implemented reaches the performance of 78.89% mIOU in the PASCAL VOC 2012 *val* set, which is slightly higher than the result (78.85%) shown in [13]. This verifies that our implementation is right. In the second step, the model achieves the performance of 68.34% mIOU in the PASCAL VOC 2012 *val* set, which is used for comparison with the proposed method in Section 3.

2.4. Quick Shift

Quick shift [19] is one of the most popular superpixel segmentation algorithms. Its principle is based on an iterative mode-seeking that identifies modes in a set of data points. A mode is defined as

the densest location in a certain feature space which is composed of all the data points. Given N data points $x_1, x_2, \ldots, x_N \in X \subseteq R^d$, quick shift first computes Parzen density estimation [21]:

$$P(x_i) = \frac{1}{N} \sum_{j=1}^{N} k\Big(D\big(x_i, x_j\big)\Big), \quad x_i, x_j \in R^d \tag{2}$$

where $k(x)$ is the kernel function, which is usually an isotropic Gaussian window. $D(x_i, x_j)$ is the distance between data point x_i and x_j. Then, it moves the center of the kernel window to the nearest neighbor of x_i, in order to extend the search path to the next data point. At which there is an increasing density P:

$$y_i = \underset{j:P(x_j)>P(x_i)}{\arg\min} \; D(x_i, x_j) \tag{3}$$

When all the data points are connected with one another, a threshold is used to separate modes. Different clusters of the data points can then be separated.

Quick shift may be used for any feature space, but for the purpose of this paper we restrict it to 5D feature space to use it for image segmentation. The 5D feature space is consisting of 3D RGB color information and 2D location information. After computing the Parzen density estimation for each image pixel, quick shift constructs a tree connecting each pixel to its nearest neighbor that has higher density value. Then each pixel connects to its closest higher density pixel parent that achieves the minimum distance. It generates a forest of pixels whose branches are labeled with a distance value. This specifies a hierarchical segmentation of the image. Superpixels can be identified by applying a threshold to cut the branches that is larger than it.

We apply quick shift to partition the input image into superpixels, as shown in Figure 5.

(a) (b)

Figure 5. Superpixel segmentations by quick shift. (**a**) shows raw image, (**b**) shows quick shift superpixels.

2.5. Class Voting

DeepLab v3+ outputs a class-indexed score map for the input image, each pixel of which is labeled with an index corresponding to its predicted class. At the same time, superpixels of the image are obtained by the quick shift algorithm. Each pixel in the image is labeled with a superpixel index. Then, the total number of pixels belonging to each class in each superpixel are counted. Finally, the Class Voting module votes on each superpixel to the class which contains the maximum number of pixels in it. A pseudo-code implementation is shown in Algorithm 1.

Algorithm 1 classVoting()

Input: $classNum$: number of classes; $clusterNum$: number of superpixels segmented by quick shift; $quickshift(W,H)$: output of quick shift; $deeplabv3plus(W,H)$: output of DeepLab v3+;
Output: $segment(W,H)$;
initial $clusterStat(i,j) = 0, \quad i = 0,\ldots,clusterNum-1, j = 0,\ldots,classNum-1$
initial $clusterVote(i) = 0, \quad i = 0,\ldots,clusterNum-1$
for $i = 0$ to $W-1$ **do**
  for $j = 0$ to $H-1$ **do**
    $clusterStat(quickshift(i,j), deeplabv3plus(i,j)) += 1$
  end for
end for
$clusterVote(i) = \max(clusterStat(i,j)), \quad j = 0,\ldots,classNum-1$
for $i = 0$ to $W-1$ **do**
  for $j = 0$ to $H-1$ **do**
    $segment(i,j) = clusterVote(quickshift(i,j))$
  end for
end for
return $segment(W,H)$;

3. Experiments and Results

3.1. Experimental Design

In order to evaluate the effectiveness of the proposed method, we have conducted the following experiments: (I) The mIOU value of the proposed method against DeepLab v3+; (II) The superpixel segmentation algorithm in module (b) of Figure 3 is replaced to demonstrate the superiority of quick shift against them by SLIC and Felzenszwalb. The experiments are implemented on an NVIDIA GTX TITAN Xp GPU with 12 GB of memory.

Based on PASCAL VOC 2012 dataset [9], we conducted a large number of experiments to evaluate the performance of the method in terms of quality and quantity. The qualitative metric is an important evidence of supporting our claims in an intuitive way. The quantitative metric is more convincing, which is measured in terms of pixel intersection-over-union averaged(mIOU) across the 21 classes. The definition of mIOU is as follows:

$$mIOU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{4}$$

where $k+1$ is the number of classes (including background classes), p_{ij} indicates the number pixels in class i that is classified as class j.

3.2. Qualitative Evaluation

We provide some exemplary results on PASCAL VOC 2012 *val* set in Figure 6. Note that the details of the target edges in Figure 6d are significantly improved than Figure 6c, and the segmentation results in Figure 6c often include parts of the background. In other words, the proposed method can estimate more accurate objects boundaries than DeepLab v3+, even when the foreground objects are very small. It can also match the ground-truth more consistently and segments foreground boundaries more precisely with images that contain foreground objects in complex environments. The results in Figure 6 implies the effectiveness of the proposed method in localizing objects boundaries when conducing semantic segmentation tasks.

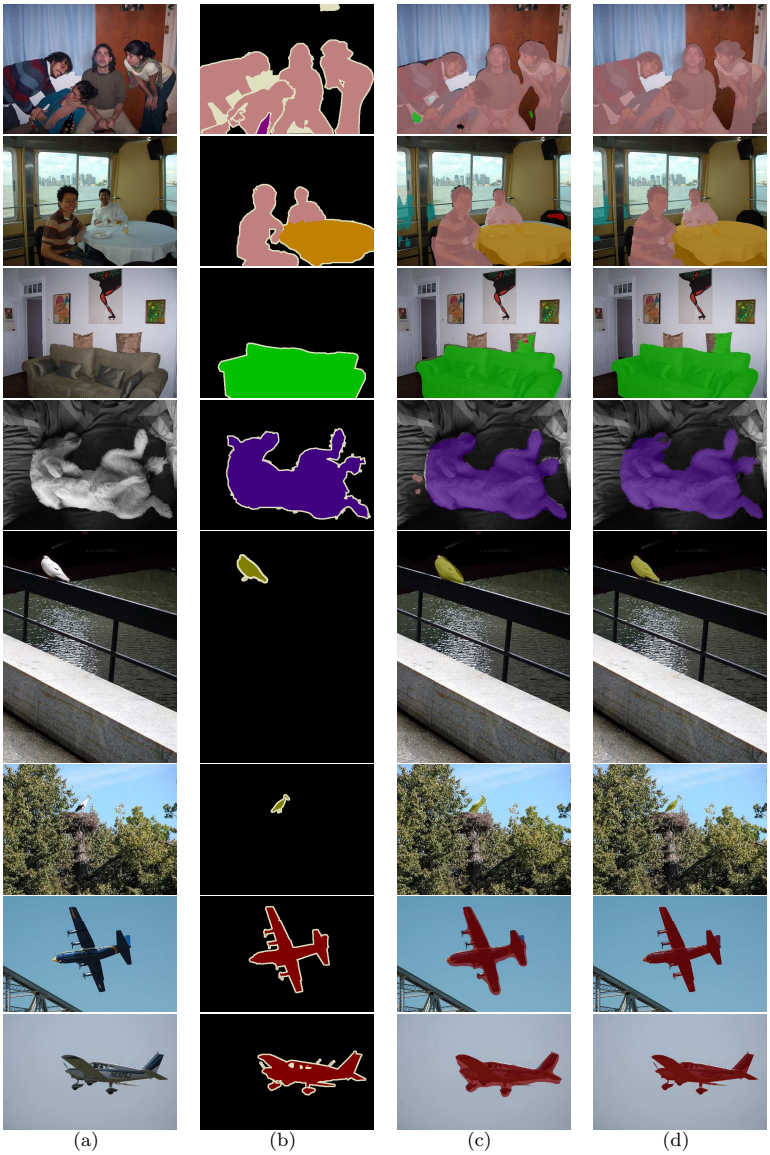


Figure 6. Visualized comparison between DeepLab v3+ and the proposed method on the PASCAL VOC 2012 *val* set. (**a**) shows raw images, (**b**) shows the ground-truths, (**c**) shows segmentation results delivered by DeepLab v3+, (**d**) shows the refined segmentation results obtained from the proposed method. (For better visualization, we overlap segmentation results on raw images in (**c**,**d**).)

3.3. Quantitative Evaluation

For the quantitative evaluation of the proposed method, four groups of experiments conducted: (1) DeepLab v3+, (2) DeepLab v3+ + SLIC , (3) DeepLab v3+ + Felzenszwalb, and (4) DeepLab v3+ + quick shift. To simplify ablation study, all methods are trained on VOC 2012 train and tested on VOC 2012 val. Table 1 lists the quantitative results of all the compared methods. The best one is highlighted in bold. As shown in Table 1, the proposed method achieves higher performance than the others. To be specific, our proposed method outperforms DeepLab v3+ by 1.26% in the PASCAL VOC 2012 *val* set. The superpixel segmentation algorithm-quick shift- is superior to SLIC and Felzenszwalb by 1.85% and 0.8%, respectively.

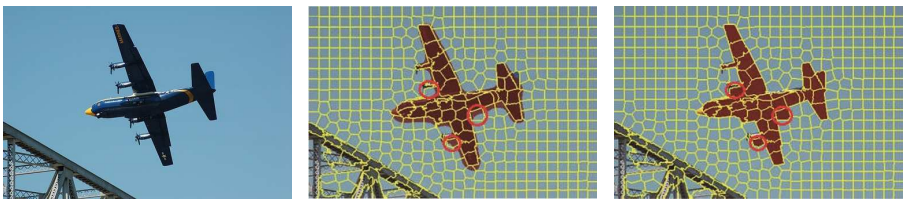
Table 1. Performance on the PASCAL VOC 2012 *val* set.

Serial Number	Method	mIOU
1	DeepLab v3+	68.34%
2	DeepLab v3+ + SLIC	67.94%
3	DeepLab v3+ + Felzenszwalb	68.65%
4	DeepLab v3+ + quick shift(ours)	69.20%

4. Discussion

4.1. Why Quick Shift Superior to SLIC and Felzenszwalb

When partitioning an image into superpixels, SLIC evenly distribute seed points in the image. Thus, regions that belong to same object are often force to be divided into different superpixels, which causes misclassification. As shown in Figure 7, it incorrectly segments the background of sky as one part of the airplane on its boundaries. We label the wrongly segmented superpixels in Figure 7 with red circles.



(a) (b) (c)

Figure 7. Image segmented using SLIC. (**a**) shows the raw image, (**b**) shows applying SLIC on segmentation results from DeepLab v3+, (**c**) shows outputs from Class Voting module in Figure 3 by using SLIC as the superpixel segmentation algorithm. (In order to better explain the question, we overlap the segmentation result of the target object on the raw image in (**b**,**c**).)

Felzenszwalb often oversegments images. As shown in Figure 8, Felzenszwalb partitions the image into more than thousands of superpixels. In extreme cases, the number of superpixels it produces is equal to the number of pixels in the image, which makes no improvement over segmentation results from DeepLab v3+.

(a) (b) (c)

Figure 8. Image segmented using Felzenszwalb. (**a**) shows applying Felzenszwalb on the raw image, (**b**) shows outputs from Class Voting module in Figure 3 by using Felzenszwalb as the superpixel segmentation algorithm, (**c**) shows segmentation results from DeepLab v3+. (In order to better explain the question, we overlap the segmentation result of the target object on the raw image in (**b**,**c**)).

Compared with SLIC and Felzenszwalb, quick shift automatically finds the proper number of clusters and merges similar superpixels, which makes it achieving the best performance.

4.2. The Influence of Parameter σ on Segmentation Results

In Equation (2), we use Gaussian function (as shown in Equation (5)) as the kernel function.

$$P(x) = \frac{1}{2\pi\sigma^2 N} \sum_{i=0}^{N} e^{\frac{-\|x-x_i\|^2}{2\sigma^2}} \quad (5)$$

σ is the width of Gaussian kernel. Smaller σ causes the density estimate of pixel x_i only calculating local information, which leads to oversegmentation. Larger σ smooths the density of each location, which leads to fewer clusters. As shown in Table 2, we have experimented different value of σ on the PASCAL VOC 2012 *train* set, and found that 5 is the experimentally optimal one.

Table 2. mIOU with different σ on the PASCAL VOC 2012 *train* set.

σ	**mIOU**
3	78.07%
5	**78.21%**
7	77.85%
9	76.98%

5. Conclusions

DCNNs with deep feature representation of the images are driving significant advances in semantic segmentation. Nonetheless, its drawback is susceptible to interference from segmenting objects boundaries. Therefore, we propose a novel semantic segmentation algorithm with DCNNs and quick shift. Compared with DeepLab v3+, the proposed method can localize accurate objects boundaries, which is meaningful in practical applications, such as object recognition, automatic drive, etc. Even though the proposed method works very well in most cases, it is still fails since the boundary of the object is difficult to distinguish from the surrounding background in a dark environment. In the future work, we will investigate the interplay of DCNNs and superpxiel algorithms to tackle this problem.

Author Contributions: Conceptualization, S.Z., Z.M., Y.C. and T.L.; methodology, S.Z., Z.M. and T.L.; conceived and designed the experiments, S.Z., R.Z. and T.L.; performed the experiments, Z.M.; writing—original draft preparation, S.Z. and T.L.; writing—review and editing, G.Z., Y.C. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Youth Innovation Promotion Association, Chinese Academy of Sciences (Grant No. 2016336).

Acknowledgments: The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
2. Gu, Y.; Wang, Y.; Li, Y. A Survey on Deep Learning-Driven Remote Sensing Image Scene Understanding: Scene Classification, Scene Retrieval and Scene-Guided Object Detection. *Appl. Sci.* **2019**, *9*, 2110. [CrossRef]
3. Ma, H.; Liu, Y.; Ren, Y.; Yu, J. Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3. *Remote Sens.* **2020**, *12*, 44. [CrossRef]
4. Lu, Z.; Chen, D. Weakly Supervised and Semi-Supervised Semantic Segmentation for Optic Disc of Fundus Image. *Symmetry* **2020**, *12*, 145. [CrossRef]
5. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25, Lake Tahoe, CA, USA, 3–6 December 2012; pp. 1097–1105.

7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
9. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
10. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
11. 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 Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
12. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
13. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 833–851.
14. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
15. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In Proceedings of the Advances in Neural Information Processing Systems 24, Granada, Spain, 12–14 December 2011; pp. 109–117.
16. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
17. Sifre, L.; Mallat, S. Rigid-Motion Scattering for Image Classification. Ph.D. Thesis, Ecole Normale Superieure, Paris, France, 2014.
18. Ren, X.; Malik, J. Learning a classification model for segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 10–17.
19. Vedaldi, A.; Soatto, S. Quick shift and kernel methods for mode seeking. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 705–718.
20. Sheikh, Y. A.; Khan, E.A.; Kanade, T. Mode-seeking by medoidshifts. In Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
21. Erdil, E.; Ghani, M.U.; Rada, L.; Argunsah, A.O.; Unay, D.; Tasdizen, T.; Cetin, M. Nonparametric joint shape and feature priors for image segmentation. *IEEE Trans. Image Process.* **2017**, *26*, 5312–5323. [CrossRef] [PubMed]
22. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
23. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vision.* **2004**, *59*, 167–181. [CrossRef]
24. Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; Malik, J. Semantic contours from inverse detectors. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 991–998.