An efficient semantic segmentation method based on transfer learning from object detection

Wei Yang¹,² | Jianlin Zhang¹ | Zhongbi Chen¹ | Zhiyong Xu¹

¹ Institute of Optics and Electronics, Chinese Academy of Science, Chengdu, People’s Republic of China
² University of Chinese Academy of Sciences, Beijing, People’s Republic of China

Correspondence
Jianlin Zhang, Institute of Optics and Electronics, Chinese Academy of Science, Chengdu 610209, People’s Republic of China.
Email: jian@ioe.ac.cn

Abstract
Nowadays, numerous semantic segmentation techniques were used to complex scenes such as urban streets. However, speed issues are not considered in most of these methods, and real-time methods do not mainly include enough accuracy. In this paper, an efficient semantic segmentation method is proposed, using the feature extractor of a real-time object detection model, Darknet53, as the backbone of DeepLabv3+. By the high accuracy of DeepLabv3+ structure and great efficiency of Darknet53, a mean intersection was obtained over union of 76.3% in Cityscapes test set, and fast inference speed simultaneously (0.178 s per frame on one GTX 1080Ti GPU). A huge imbalance of objects was noticed on Cityscapes dataset. To solve this problem, a Focal Loss like loss function was proposed to concentrate more on the hard difficult pixels. Moreover, an atrous convolution block was proposed to extract more high-level features. Based on the experimental results, it is proved that these changes contribute to a better result on the Cityscapes test set (77.8% mean Intersection over Union) and faster inference speed (0.171 s per frame). Authors’ model achieves state-of-art results on Cityscapes test set (79.1% mean Intersection over Union) after fine-tuning on Cityscapes coarsely annotated dataset.

1 | INTRODUCTION

Presently, semantic segmentation is popular with its great potential for driver assistance and scene understanding. It can identify the objects and preserve their contour by categorising each pixel into various groups. By the recent development of convolutional neural networks (ConvNets), numerous ConvNets-based methods lead to remarkable findings on semantic segmentation [1–5]. These approaches are usually oriented by the transfer learning from image classification models such as VGG [6] and ResNet [7]. However, ignore inference speed is normally in these high accuracy approaches, which is a critical factor in numerous computer vision applications such as automatic drive.

To address the speed issue, some researchers suggest their real-time solutions [8–10]. The real-time object semantic segmentation methods are still far behind compared to the real-time object detection methods such as SSD [11] and YOLO [12] achieving amazing speed without losing too much accuracy. Object detection techniques are required to output a tight bounding box around each object in the image and a category prediction, which is a bit similar to the semantic segmentation task.

Hence, we propose an efficient semantic segmentation technique based on transfer learning from Darknet53 [13] to Deeplabv3+ [14]. Darknet53 is a ResNet style network, which can achieve 77.2% top1 accuracy and 93.8% top5 accuracy on ImageNet [15]. Its results are close to the result of ResNet-152 [7]; however, fewer layers are used in DarkNet53. Darknet53 is used in YOLOv3 as the backbone to achieve 51.5% mAP-50 (the mean average precision of intersection over the union between prediction and ground truth larger than 50%) with a speed of 22ms per frame for COCO [16] object detection challenge. Deeplabv3+ is an encoder-decoder architecture network with the advantage of atrous convolutions to maintain the reception field of the kernel, and atrous spatial pyramid pooling (ASPP) to extract multi-scale features. Deeplabv3+ achieves 82.1% mIoU on Cityscapes [17] test set using Xception [18] as its backbone and taking advantage of multi-scale output fusion. Darknet53 is efficient and able to extract features containing the information of the object’s position and category. To extract
multi-scale high-level features and achieve high-accuracy results, the structure of DeepLabv3+ can be used. We use the pre-trained Darknet53 in YOLOv3 as backbone to integrate their advantages. We construct ASPP and the decoder parts similar to DeepLabv3+. After training our model on the Cityscapes training set (2975 pixel-level annotated high-resolution images), we got mean Intersection-over-Union (mIoU) of 76.3% on the test set and 0.178s per frame speed on 1024$\times$2048 resolution image.

Within semantic segmentation, imbalance and hard difficult example problems among objects are serious. Especially in urban scenes, too much area of images is always occupied by roads and cars; however, buses and trains are rare as we can see in Figure 1. Then, most of the loss is related to a large number of pixels of road and car, even if their pixels are easy to be classified. To solve this problem, weighted cross-entropy is used in most semantic segmentation methods as their loss function to relieve the imbalance. The weights are always set based on statistics or experience. In [19], instead of just randomly perturbing the dataset and taking training crops from random positions, the mini batches per epoch were compiled to show all classes approximately uniformly. We also found that our model can difficulty distinguish the road and side-walk, as well as motor and bicycle occasionally. The above issues are similar to the problem solved by Focal Loss [20]. In [20], Focal Loss is proposed to handle the imbalance between background and foreground by down-weighting the loss of easy examples and thus focusing on training the difficult negative examples. In this study, we address the imbalance issue by improving the difficult negatives weight while preserving the weight of easy examples regarded, as an improvement of Focal Loss. Based on the results our proposed Focal Loss can contribute to getting a better result on the Cityscapes test set (76.9% mIoU).

The feature is critical in semantic segmentation. To use of the features of different scales as much as possible, numerous semantic segmentation methods like U-Net [21] and SegNet [2] use all the low-level features to improve the segmentation result. Therefore, to obtain more types of high-level features, we propose an atrous convolution block including three different convolutions. Replacing 3$\times$3 atrous convolution in ASPP with our atrous convolution block, we obtain another improvement as 77.8% mIoU on the Cityscapes test set.

In total, first, we propose an efficient semantic segmentation method based on transfer learning from object detection, which can get 5.62 frames per second (fps) speed on 1024$\times$2048 resolution image and still maintain high accuracy. Then, based on experimental results it is proved that our improved Focal Loss can be used to alleviate the imbalance of the object in the dataset. Ultimately, the proposed atrous convolution block can be used to extract more high-level features to get better semantic segmentation results.

2 | APPROACH

In this section, we briefly introduce Darknet53 and DeepLabv3+. Then, we detail how we do transfer learning from Darknet53 to DeepLabv3+. Moreover, we present how our Focal Loss addresses the imbalance issue among the objects. Ultimately, we introduce our atrous convolution block.

2.1 Architecture

As previously mentioned, transfer learning is used in our method from Darknet53 to DeepLabv3+. In the following parts, we name our method as DarkLab for convenience. Darknet53 is used as a feature extractor in YOLOv3. It is trained on ImageNet at first, and then it is used as the backbone of the YOLOv3 network which will be trained on COCO dataset [16]. DarkNet53 has very simple architecture, which is presented in Table 1. Every convolution in Darknet53 is followed with batch normalisation (BatchNorm) [22] and a leaky rectified linear unit (LeakyReLU). We only use the feature extraction parts, that is, Conv1 to Res5 of Darknet53 to build our backbone similar to YOLOv3.

DeepLabv3+ is an encoder–decoder structure for semantic segmentation. Its encoder is a feature extractor followed by ASPP. At first, the feature extractor gets the features downsampled by a factor of 16. Then, ASPP utilises multi-rate atrous convolutions to obtain multi-scale features and concatenates them to get the final features. Table 2 presents the architecture of ASPP, in which the atrous rate is indicated by the rate. Decoder up-samples the feature of encoder by a factor of 4 and concatenates it with a low-level feature of feature extractor. After concatenation, to refine the features, some 3$\times$3 convolutions are used. Then, a softmax classifier is utilised to get the low-resolution result. Finally, the low-resolution result will be up-sampled by a factor of 4 to get the prediction of the same size as an input. The decoder architecture can be seen in Table 3.
In their Focal Loss, it is possible to ignore easy examples and focus on hard examples during training.

In semantic segmentation, hard example problems and class imbalance are much more serious. Owing to the particularity of semantic segmentation task that is classifying every pixel, the number ratio among objects’ pixel is huge. Moreover, in the later stage of training, most pixels are easy examples especially when mIoU between ground truth and prediction is larger than 70%. At this stage, most pixels’ predictions to ground truth are larger than the predictions to other classes. These pixels (easy examples) also contribute to the main part of loss. Mostly, the easy examples are frequently occurred objects in the Cityscapes dataset like road and building. The predictions to the ground truth of difficult negatives are not the biggest among the predictions to all the classes. These pixels usually belong to the rarely observed objects or the confusing objects (rider vs. person). The loss function used in semantic segmentation is usually cross entropy. Our Focal Loss function is:

$$FL(p_t) = -\gamma * \log(p_t)$$

Here, $\gamma$ and $\lambda$ are hyper parameters and $p_t$ is the probability of ground-truth. When $\gamma = 0$, our Focal Loss function can be observed as standard cross-entropy. Hence, Equation (1) can be seen as a weighted cross-entropy. The weight $(\lambda - p_t)^\gamma$ is negative related to $p_t$, indicating that it can be changed according to the prediction to ground truth. Hence, the difficult negatives
will possess a larger weight than easy examples. It makes difficult negatives contribute more to the loss than using cross-entropy as the loss function.

In [20], the best set is $\lambda = 1$ and $\gamma = 2$. When $\lambda = 1$, the weight $(\lambda - p_i)^\gamma$ is always less than 1 indicating that it will reduce all the losses as it is observed in Figure 3. When the example’s prediction to ground truth is larger than 0.4, its loss will be very small. This works for object detection because these examples either have high confidence (probability to ground truth larger than 0.6) or belong to useless bounding box (probability to ground truth is between 0.4 and 0.6). However, no useless pixel exits in semantic segmentation where every pixel needs to be classified. In semantic segmentation, many pixels’ predictions to ground truth are between 0.4 and 0.6 (difficult example). In our task, the number of segmentation classes is up to 19. Some of these classes are similar to each other such as roads and sidewalks, hence, normally there are two or more predictions close to each other. It always causes these pixels’ predictions to ground truth to fall between 0.4 and 0.6. The final step of our model is bilinear interpolation upsampling by a factor of 4. Among all, two predictions of the edge pixels between two objects will have larger value than others. It also cause the edge pixels’ predictions to ground truth to reduce between 0.4 and 0.6. When $\lambda = 1$, all the above-mentioned loss of pixels will be very small. In our experiments, to preserve the weight of easy examples and enlarge the weight of hard examples, we set $\lambda = 2$.

Our experimental results show that we can get our best result on Cityscapes test set (76.9% mIoU) when $\lambda = 2$ and $\gamma = 0.5$.

### 2.3 Atrous convolution block

Recent studies on semantic segmentation like PSPNet and DeepLabv3+ try to extract more high-level features. The ASPP proposed by DeepLab [3] was successfully utilised to extract features of different scales by enlarging kernel’s reception field while keeping the parameters. In DeepLabv3+, using one $1 \times 1$ convolution, three $3 \times 3$ atrous convolutions with different atrous rates (6, 12, 18) and a global pooling, high-level features are obtained in different scales from backbone’s output. Their experimental results prove that more high-level features can contribute to a better result.

To extract more high-level features, we propose an atrous convolution block consisting of three various kinds of convolution. Our atrous convolution block is very similar to the inception module used in GoogLeNet [23] as we can see in Figure 4.
where IC represents the number of input channels and OC shows the number of output channels. Our atrous convolution block uses three types of convolution to extract more kinds of features compared to using $3 \times 3$ atrous convolution similar to DeepLabv3+. Every convolution in our block is followed by a BatchNorm and LeakyReLU (slope is 0.1). To extract high-level features in more scales, we use four atrous convolution blocks of different atrous rates ($6, 12, 18, 24$) to replace the original three $3 \times 3$ atrous convolutions. We also reduce the number of output channels of atrous convolution blocks to balance the speed and accuracy. These changes give us another improvement including 77.8% mIoU on the Cityscapes test set for which the details of the implementation are described in Section 3.3.

3 | EXPERIMENTS

3.1 | Implementation details

We implement our method on PyTorch [24] and run the proposed networks on four GTX 1080Ti GPUs (each with 11GB RAM). All the convolutions in our DarkLab are followed with a BatchNorm and LeakyReLU (slope is 0.1). We selected the batch size of 12 and a crop size of $768 \times 768$ owing to the limitation of GPU memory. Data augmentation includes random crop, random mirror, random resizing and random rotation. The resizing scale is within 0.75–2 with the rotation degree limited of GPU memory. Data augmentation includes random crop, random mirror, random resizing and random rotation. The resizing scale is within 0.75–2 with the rotation degree limited.

We examine all our models on the Cityscapes dataset including high-resolution images up to $1024 \times 2048$. Processing high-resolution images is a major challenge for efficient semantic segmentation. The dataset includes 5000 finely annotated images split into training, validation and testing sets with 2975, 500 and 1525 images, respectively. Another 20K coarsely annotated images are also provided by the Cityscapes dataset. The test labels are not available, however, the test results can be evaluated on an online test server. We only train our models on the training set, with no validation set for training. Our accuracy results are reported utilising the commonly adopted Intersection-over-Union (IoU) metric:

$$IoU = \frac{TP}{TP + TN + FN}$$ \hspace{1cm} (2)

where TP, TN and FN are, respectively, the number of true positives, true negatives and false negatives at pixel level.

### Results and comparison

The comparison of our methods to several state-of-art methods is provided in Table 5 where the last four represent our proposed methods. DarkLab is trained with weighted cross-entropy. DarkLab-FL is DarkLab trained with Focal Loss. DarkLab-FL-ACB is DarkLab-FL trained with atrous convolution blocks in ASPP. DarkLab-FL-ACB-Coarse is

| $\lambda$ (\%) | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 |
|---------------|-----|-----|-----|-----|-----|
| 1             | 76.3| 76.6| 76.3| 75.6| 75.5|
| 2             |     | 76.9| 76.7| 76.7| 76.5|

### 3.2 | Focal Loss

We trained our DarkLab with eight groups of different $\gamma$ and $\lambda$ with Focal Loss to find out the best set of our Focal Loss, for which the results are presented in Table 4. We can get the best result with $\lambda = 2$ and $\gamma = 0.5$, and the result is getting worse by increasing $\gamma$ no matter $\lambda$ equals 1 or 2. When $\lambda = 2$, the results are almost all better than those when $\lambda = 1$ proving our suggestion that preserving the weight of easy examples and enlarging the weight of difficult negatives are more appropriate to semantic segmentation.

When $\gamma = 0.0$, our Focal Loss is equal to standard cross-entropy. Hence, we utilise the weighted cross-entropy as in ERFNet to decrease the imbalance during training. Moreover, our experimental results prove that Focal Loss can alleviate imbalance better than weighted cross-entropy.

### 3.3 | Atrous convolution block

As we mentioned above, we use four atrous convolution blocks to replace the original three $3 \times 3$ atrous convolutions, that is, ASPP2, ASPP3 and ASPP4 in Table 2. And we keep the ASPP1 and Pooling in Table 2. In the ASPP of DeepLabv3+, the number of output channels of every convolution is 256 as we can see in Table 2. In order to balance the inference speed and the number of feature's type. We change the number of output channels to 128 in our ASP. The number of output channels of $3 \times 3$, $3 \times 1$ and $1 \times 3$ convolution are 64, 32, 32, respectively. So, number of our ASP output channel is 768 which is less than the original one (1280), and it will reduce the computational complexity.

Our ASPP with atrous convolution blocks is efficient. It causes our models to occupy less GPU memory than before. Hence, during the training, we change the crop size to $800 \times 800$, and the inference time is 7 ms less on $1024 \times 2048$ resolution image with one GTX 1080Ti GPU during the inference.

### 3.4 | Results and comparison

The comparison of our methods to several state-of-art methods is provided in Table 5 where the last four represent our proposed methods. DarkLab is trained with weighted cross-entropy. DarkLab-FL is DarkLab trained with Focal Loss. DarkLab-FL-ACB is DarkLab-FL trained with atrous convolution blocks in ASPP. DarkLab-FL-ACB-Coarse is
### TABLE 5  
Comparison to state-of-art methods

| Method          | Pretrained       | Sub | mIoU (%) | Inference Time (s) |
|-----------------|------------------|-----|----------|--------------------|
| ENet [9]        | –                | 2   | 58.3     | 0.013              |
| SQ [25]         | ImageNet         | no  | 59.8     | 0.06               |
| ERFNet [8]      | ImageNet         | 2   | 69.7     | 0.024              |
| ICNet [10]      | ImageNet         | no  | 69.5     | 0.033              |
| FCN-8s [1]      | ImageNet         | no  | 65.3     | 0.5                |
| RefineNet [5]   | ImageNet         | no  | 73.6     | n/a                |
| PSPNet [4]      | ImageNet         | no  | 81.2     | n/a                |
| DeepLabv3+ [14]| ImageNet, COCO  | no  | 76.3     | 0.178              |
| DarkLab        | ImageNet, COCO  | no  | 76.8     | 0.178              |
| DarkLab-FL-ACB | ImageNet, COCO  | no  | 77.8     | 0.171              |
| DarkLab-FL-ACB-Coarse | ImageNet, COCO | no  | 79.1     | 0.171              |

### TABLE 6  
Class result of different DarkLab only trained on Cityscapes finely annotated data

| Method          | Roa | Sid | Bui | Wal | Fen | Pol | Tli | Tsi | Veg | Ter | Sky | Per | Rid | Car | Tru | Bus | Tra | Mot | Bic |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DarkLab        | 98.2| 83.8| 92.1| 51.1| 56.5| 63.2| 71.7| 75.4| 93.0| 70.0| 95.1| 84.9| 69.7| 95.1| 67.5| 74.8| 66.2| 67.0| 74.5|
| DarkLab-FL     | 98.3| 84.4| 92.4| 51.0| 55.8| 62.9| 72.2| 76.1| 93.0| 68.7| 95.2| 85.5| 68.8| 95.4| 69.0| 78.5| 70.6| 67.9| 74.7|
| DarkLab-FL-ACB | 98.4| 84.6| 92.6| 52.8| 55.9| 63.9| 72.9| 76.5| 93.1| 70.1| 95.3| 85.7| 69.2| 95.6| 72.8| 82.5| 73.9| 67.0| 74.7|
| DarkLab(I)-FL-ACB | 98.4| 84.6| 92.4| 48.6| 56.6| 63.2| 71.4| 76.7| 93.2| 69.7| 95.5| 84.6| 66.8| 95.5| 66.2| 80.7| 71.3| 66.9| 73.9|

DarkLab-FL-ACB fine-tuned with Cityscapes coarse dataset after training on Cityscapes finely annotated dataset providing another improvement as 79.1% mIoU on Cityscapes test set. This comparison proves that our method can get a good balance between accuracy and speed, while acquiring better accuracy compared to all approaches focusing on speed. The high-accuracy methods do not provide the inference time since they use some augmentation method during the test such as flipping and multi-scale input. Then, they fuse these outputs to get better results. Hence, these methods have problem in calculating the inference time. Here, we just use the original image as our input without any other trick. However, our methods still result in a state-of-art outcome close to those high accuracy approaches. Moreover, our model can process the 1024 × 2048 resolution image in 0.171 s. Hence, our method can be easily applied to the scenes requiring fast processing speed and high segmentation accuracy.

Based on Figure 5, it is understood that our methods can make very good predictions on complex urban street scenes even containing small objects in distance. Furthermore, it is indicated that our proposed Focal Loss and atrous convolution block can deal with those confusing objects like road and side walk better.

Since YOLOv3 does not provide parameters of Darknet53 trained on ImageNet, we retrain Darknet53 on ImageNet for 90 epoches with the same training strategy similar to YOLOv3 obtaining 76.5% top1 and 93.0% top5 accuracy on ImageNet validation set. Then, we build the backbone of DeepLabv3+ with the Darknet53 trained on ImageNet. We obtain 76.7% mIoU on Cityscapes test set after building the ASPP with atrous convolution blocks and training the model with Focal Loss. Compared to the result of DarkLab-FL-ACB (77.8% mIoU), our method proves transfer learning from object detection is more suitable to semantic segmentation.

The results of our methods on Cityscapes test set are shown in Table 6. These methods are all only trained with Cityscapes finely annotated data. DarkLab(I) indicates the backbone where Darknet53 only pre-trained on ImageNet training set (transfer learning from image classification). The list of classes (from left to right) includes road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorbike and bicycle. Comparing segmentation results between DarkLab and DarkLab-FL, we can find out that there is no significant differences in results on the commonly observed objects such as road, vegetation, sky and car. However, the segmentation results of DarkLab-FL are significantly improved on those rarely seen objects such as truck, bus and train compared to the results of DarkLab. It proves that our Focal Loss can be used to alleviate the class imbalance during the training of semantic segmentation. Comparing segmentation results between DarkLab-FL and DarkLab-FL-ACB, we can find that DarkLab-FL-ACB can get a better result on all the objects except for motorbike. It proves that our proposed atrous convolution block can be used in ASPP to enhance the semantic segmentation result. The only difference between DarkLab-FL-ACB and DarkLab(I)-FL-ACB is that DarkLab-FL-ACB uses
transformation learning from object detection while DarkLab(I)-FL-ACB uses transfer learning from image classification. Moreover, the segmentation results of DarkLab-FL-ACB on these 19 objects are either close or much better compared to the results of DarkLab(I)-FL-ACB. It demonstrates that transfer learning from object detection can help us obtain better semantic segmentation results compared to the transfer learning from image classification. In total, our proposed methods can be used to improve the semantic segmentation results, and they are also easy to be used in other semantic segmentation model to get better results.

4 Conclusion

In this study, we propose a new notion to do transfer learning from efficient object detection model YOLOv3 for semantic segmentation. With the structure of DeepLabv3+, we obtain a state-of-art result on the Cityscapes test set with fast inference speed. Taking the advantages of the Focal Loss used in RetinaNet [20], we handle the class imbalance in the training set. Making some changes to Focal Loss helps us to get a better result. Moreover, we propose a atrous convolution block which consists of three different convolutions to extract high-level features of more types. It helps us get another better segmentation accuracy with faster inference speed. It is also proved that transfer learning from object detection is better than transfer learning from image classification for semantic segmentation. Based on the experimental results in this work, it is indicated that our method can get a good balance between accuracy and efficiency on high-resolution images. However, we are still far away from achieving real-time inference speed constituting our future research direction.

References

1. Shelhamer E., et al.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(4), 640–651 (2017)
2. Badrinarayanan V., et al.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(12), 2481–2495 (2017)
3. Chen L.C., et al.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40(4), 834–848 (2018)
4. Zhao H., et al.: Pyramid scene parsing network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6230–6239. IEEE, Piscataway (2017)
5. Lin G., et al.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5168–5177. IEEE, Piscataway (2017)
6. Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. Paper presented at international conference on learning representations (ICLR), San Diego, 7–9 May 2015
7. He K., et al.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Piscataway (2016)
8. Romera E., et al.: ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst. 19(1), 263–272 (2018)
9. Paszke A., et al.: Enet: A deep neural network architecture for real-time semantic segmentation, arXiv:1606.02147, 2016
10. Zhao H., et al.: ICNet for real-time semantic segmentation on high-resolution images, arXiv:1704.08545v1, 2017
11. Liu W., et al.: SSD: Single shot multibox detector, In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2, 2016
12. Redmon J., et al.: You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE, Piscataway (2016).
13. Redmon J., Farhadi A.: YOLOv3: An incremental improvement, arXiv:1804.02767, 2018
14. Chen L.C., et al.: Encoder-decoder with atrous separable convolution for semantic image segmentation, In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11211. Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_49, 2018
15. Deng J., et al.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. IEEE, Piscataway (2009)
16. Lin T.Y., et al.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (ECCV), pp. 740–755. Springer, Cham (2014).
17. Cordts M., et al.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223. IEEE, Piscataway (2016)
18. Chollet F.: Xception: Deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807. IEEE, Piscataway (2017)
19. Bulo S.R., et al.: In-place activated BatchNorm for memory-optimized training of DNNs, arXiv:1712.02616, 2017
20. Lin TY., et al.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007. IEEE, Piscataway (2017)
21. Ronneberger O., et al.: U-Net: Convolutional networks for biomedical image segmentation, In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28, 2015
22. Ioffe S., Szegedy C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: JMLR International Conference on Machine Learning (ICML), pp. 448–456. Curran Associates, Inc., New York (2016)
23. Szegedy C., et al.: Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1–9, https://doi.org/10.1109/CVPR.2015.7298594, 2015.24. Paszke A., et al.: Automatic differentiation in PyTorch. In: Proceedings of Advances in Neural Information Processing System (NIPS). Curran Associates, Inc., New York (2017)
25. Treml M., et al.: Speeding up semantic segmentation for autonomous driving. In: Proceedings of Advances in Neural Information Processing System (NIPS). Curran Associates, Inc., New York (2016)

How to cite this article: Zhang WJ, Chen Z, Xu Z. An efficient semantic segmentation method based on transfer learning from object detection. IET Image Process. 2021;15:57–64. https://doi.org/10.1049/ipr.12005