CasNet: A cascade coarse-to-fine network for semantic segmentation

Article *in* Tsinghua Science & Technology · April 2019

DOI: 10.26599/TST.2018.9010044

3 authors, including:

Zhidong Deng
Tsinghua University
167 PUBLICATIONS **1,080** CITATIONS

SEE PROFILE

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/11 pp207–215
DOI: 10.26599/TST.2018.9010044
Volume 24, Number 2, April 2019

CasNet: A Cascade Coarse-to-Fine Network for Semantic Segmentation

Zhenyang Wang, Zhidong Deng*, and Shiyao Wang

Abstract: Semantic segmentation is a fundamental topic in computer vision. Since it is required to make dense predictions for an entire image, a network can hardly achieve good performance on various kinds of scenes. In this paper, we propose a cascade coarse-to-fine network called CasNet, which focuses on regions that are difficult to make pixel-level labels. The CasNet comprises three branches. The first branch is designed to produce coarse predictions for easy-to-label pixel regions. The second one learns to distinguish the relatively difficult-to-label pixels from the entire image. Finally, the last branch generates final predictions by combining both the coarse and the fine prediction results through a weighting coefficient that is estimated by the second branch. Three branches focus on their own objectives and collaboratively learn to predict from coarse-to-fine predictions. To evaluate the performance of the proposed network, we conduct experiments on two public datasets: SIFT Flow and Stanford Background. We show that these three branches can be trained in an end-to-end manner, and the experimental results demonstrate that the proposed CasNet outperforms existing state-of-the-art models, and it achieves prediction accuracy of 91.6% and 89.7% on SIFT Flow and Standford Background, respectively.

Key words: semantic segmentation; convolutional neural network; hard negative mining

1 Introduction

Semantic segmentation has a wide range of applications, such as environmental perception in robotics and self-driving car. The goal of semantic segmentation is to identify and assign a category label to each pixel in an image as shown in Fig. 1, which requires a complete understanding of the context of the whole image scene. In recent years, deep learning has led to great breakthroughs in computer vision tasks, such as image classification[1], speech recognition[2], and object detection[3]. For semantic segmentation tasks, several deep learning-based methods can also be applied. In the past, researchers have attempted to apply Convolutional Neural Networks (CNNs) designed for image classification directly to semantic segmentation. Although good segmentation results can be obtained, the prediction results are rough,

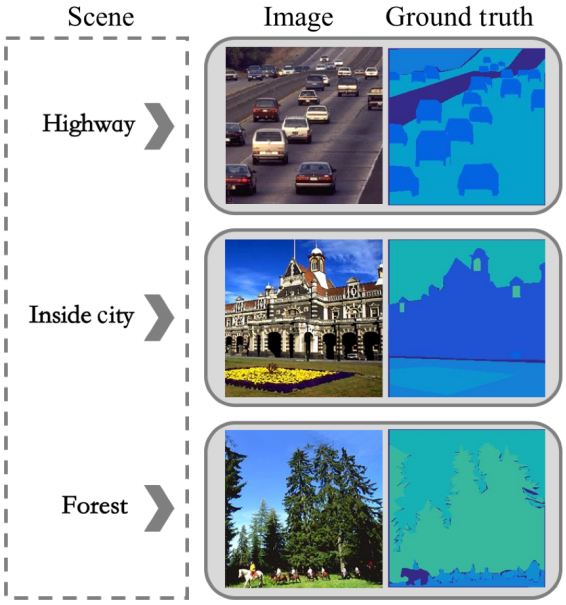


Fig. 1 Examples of semantic segmentation.

• Zhenyang Wang, Zhidong Deng, and Shiyao Wang are with the Department of Computer Science, Tsinghua University, Beijing 100084, China. E-mail: crazycry2010@gmail.com; michael@tsinghua.edu.cn; sywang14@mails.tsinghua.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2017-10-25; accepted: 2017-12-18

and it is difficult to correctly recognize the edges of objects. This is mainly caused by the lack of location information. Consequently, Fully Convolutional Neural Network (FCNN)[4] was proposed to overcome this disadvantage, and it has become the most popular framework for semantic segmentation. However, the FCNN still poses several challenges.

To improve the localization of object boundaries, Conditional Random Field (CRF), Markov Random Field (MRF), Gaussian CRF, and other variations have been proposed. In addition, with the rapid development of CNN architectures, a network via end-to-end training procedure, such as ResNet[1], can achieve the same or even better segmentation results. However, the unbalance of the easy and difficult-to-label pixels can wreck the convergence of the network. Automatic selection of these hard examples can make training effective and efficient. In fact, hard negative mining is not a new concept. As early as 1994, Sung and Poggio[5] proposed a bootstrapping method in their face detection algorithm mainly to enhance the detection capacity by changing the distribution of difficult samples.

In this paper, a Cascade coarse-to-fine Network architecture (CasNet) is proposed. Different from previous investigations in which a classical network was explored, we explore a model with multiple segmentation branches that function collaboratively to refine the prediction results. Specifically, our CasNet is composed of three branches that share the same feature extraction network but concentrate on their own targets. The first branch is a coarse segmentation network that can handle those easy and confident regions. To deal with the problem of unbalanced distribution, the second branch is an attention network, which is used for predicting the probability of each pixel being a hard example. For the difficult pixels, we exploit the last segmentation branch to refine the segmentation results.

In summary, we address the semantic segmentation task with a cascade coarse-to-fine network architecture. The proposed method introduces an idea of hard mining by an attention branch that is experimentally shown to have substantial practical merits. We validate the effectiveness of our CasNet model by testing on both SIFT Flow[6] and Stanford Background[7].

2 Related Work

The aim of semantic segmentation is to assign a unique semantic class to each pixel of an input image. Early studies are mainly based on hand-craft features, until the CNN was successfully applied to this task in recent years.

Traditional methods have obtained several solutions on semantic segmentation. Considering the context information, several methods based on MRF, CRF, and other types of graphical models have been proposed to ensure labeling consistency[8–10]. For semantic segmentation tasks, both global and local contexts significantly influence the final segmentation results. The traditional methods only consider the low-level features of the image; moreover, they are not adequate for extracting hierarchical representations, and are problematic to use for such tasks.

There are three methods of utilizing deep neural networks to improve architectural design. One method is mainly based on multi-scale feature ensemble, since high-level features contain global context information. To our knowledge, Farabet et al.[11] pioneered the application of CNNs to semantic segmentation. They proposed a multi-scale CNN that can extract features from different scales of local regions. The experimental results showed that their network can implicitly learn texture, shape, and domain information. A similar idea has been successfully generalized to RGB-D images by Couprie et al.[12] Zhao et al.[13] proposed a pyramid pooling module to aggregate different levels of context information, which can effectively produce good quality results on semantic segmentation tasks. Considering pyramid features, Lin et al.[14] utilized the inherent multi-scale pyramidal hierarchy of deep convolutional networks to generate feature pyramids with marginal external cost.

The second method focuses on enlarging the receptive field of neural networks. Yu and Koltun[15] proposed dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. In Ref. [16], recurrent neural networks were used to retrieve contextual information by sweeping the image horizontally and vertically in different directions: top to bottom, bottom to top, right to left, and left to right. In addition, Liang et al.[17] adopted recurrent CNN to incorporate both the local discriminative features and the global context information. Chen et al.[18] proposed atrous convolution to explicitly control the resolution of feature response.

The third method involves endowing FCNN architectures with the ability so as to provide structured

outputs. Chen et al.[18] are the pioneers to adopt CRF as a post-processing technique to refine the final segmentation results. Zheng et al.[19] built on this work. They combined the strengths of CNNs and CRFs based on probabilistic graphical modeling, making it possible to train the whole deep network end-to-end. Chen et al.[18] combined the responses at the final DCNN layer with a fully connected CRF to improve the localization of object boundaries.

In addition, researchers attempted to use pre-trained CNNs for semantic segmentation. Mostajabi et al.[20] obtained local features by using CNNs, while global feature representations were produced from Alexnet, and then the features were aggregated to predict the categories. Different from this method, a fully convolutional network that is able to take inputs of arbitrary sizes and produce correspondingly sized outputs with efficient inference and learning has been presented[4]. The researchers used CNNs trained on ImageNet as a feature extractor and transferred their learned representations by fine-tuning on the task-specific datasets.

3 Method

Inspired by online hard example mining algorithm, we propose a cascade coarse-to-fine network architecture called CasNet. Its framework is shown in Fig. 2. Given an image, a ResNet is employed to extract feature representation. Then, the proposed CasNet is utilized to learn task-specific targets.

3.1 Feature extraction network

We choose ResNet-50, which is pre-trained on ImageNet as our feature extractor. ResNet was originally designed for image classification. It won the ILSRVC 2015 competition and outperformed the human-level performance on ImageNet. It has capabilities of extracting hierarchical representations. Considering the computation resources and memory consumption, we choose ResNet-50 rather than ResNet-101, since ResNet-50 can already achieve comparable accuracy. In Fig. 2, the hexahedron presents the feature extractor. Although we use this simplified figure to present the ResNet-50, it is composed of five stages with different configurations of layers and a classification stage. The building block of ResNet can be defined below:

$$y = F(x, \{W_i\}) + x \tag{1}$$

where x and y denote the input and output of a layer, respectively. The function $F(x, \{W_i\})$ indicates the residual mapping, while W_i represents a group of learnable weights. The operation $F + x$ is performed by a shortcut connection and element-wise addition, which combines multi-scale features. The operation greatly benefits the segmentation task. For semantic segmentation, it is important for the context features to predict the correct label of each pixel instance. However, since different objects may have different contours, it is difficult to determine the boundaries of each object. The problem becomes more complicated

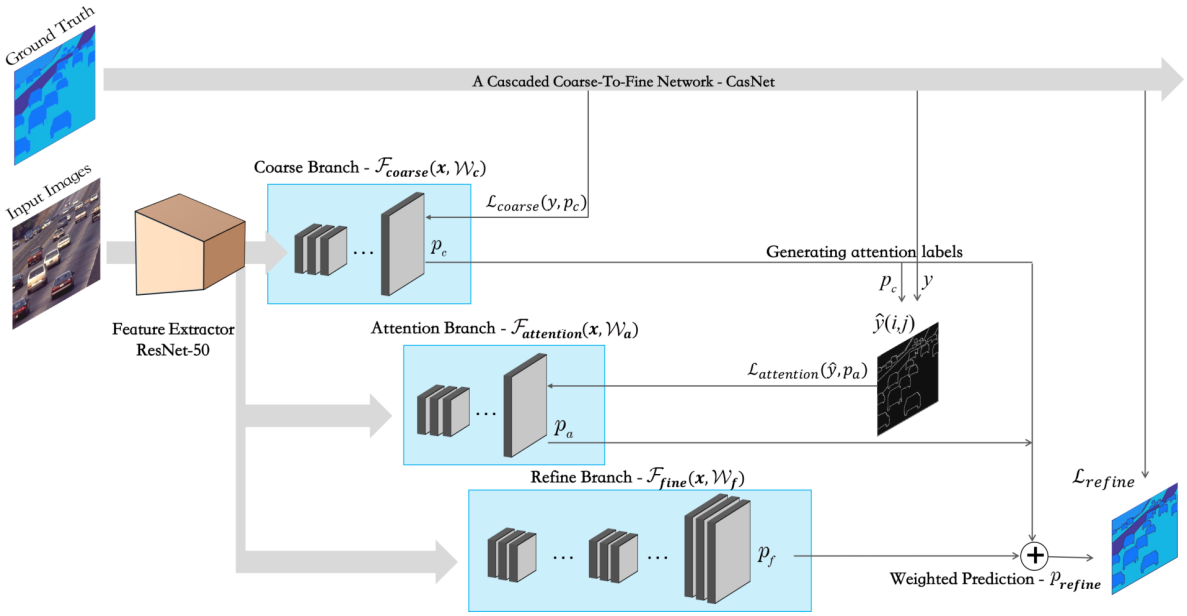


Fig. 2 A cascade coarse-to-fine network architecture for semantic segmentation.

when considering the various perspectives of each image. A simple yet effective method to solve this problem is to integrate multi-scale features for label prediction. Consequently, the residual error model itself has the property of extracting and integrating multi-scale features, which can be seen from Eq. (1). From the unraveled view by Veit et al.[21], a two-unit ResNet is equivalent to an ensemble of four sub-networks with different receptive fields. Therefore, the whole ResNet-50 can be expanded as a linearly growing ensemble of sub-networks, which can extract and integrate multi-scale features.

In addition, we implement two improvements to make ResNet-50 more suitable for semantic segmentation. First, we only keep the first three pooling layers to preserve the resolution. Thus the final resolution of the prediction is 1/8 of the original input image resolution. Second, we replace the convolutional layer in the last two stages with dilated convolutions; this can enlarge the reception field of predicted feature maps.

3.2 Cascade coarse-to-fine architecture

The CasNet architecture is shown in Fig. 2. Three horizontal lines running from the input to target are the proposed cascade branches: a coarse segmentation branch as a baseline result, an attention branch to predict the difficulty-to-label pixels, and a refine segmentation branch to achieve the final segmentation results. These three branches share a common feature extraction network while focus on their own targets.

3.2.1 Coarse segmentation branch

The coarse segmentation branch is a baseline model for semantic segmentation, as shown in the first row in Fig. 2. We adopt an FCNN that consists of two convolutional layers to predict the semantic results for relatively easy and confident regions. Since the resolution is 1/8 of the original input image resolution, the feature maps are up-sampled by bilinear interpolation. Finally, a pixel-wise softmax loss function is adopted to predict the probabilities of each pixel. We first formulate the coarse segmentation branch that produces the probability map as Eq. (2), and the loss function is defined as Eq. (3):

$$p_c(i,j) = \mathcal{F}_{coarse}(x, \mathcal{W}_c) \tag{2}$$

$$\mathcal{L}_{coarse}(y, p_c) = -\frac{1}{N}[\sum_{(i,j)\in I} \log(p_c^{y(i,j)}(i,j))] \tag{3}$$

where (i,j) is the pixel location of the given image I, and x is the input feature extracted by the feature extraction network in Section 3.1. $\mathcal{F}_{coarse}$ represents the coarse segmentation branch with trainable weights $\mathcal{W}_c$, and $p_c(i,j)$ denotes the computed probability of each pixel. Particularly, $p_c(i,j)$ in Eq. (2) is a K-dimensional vector (whose elements sum to 1) that represents the estimated probabilities for K classes, while $p_c^{y(i,j)}(i,j)$ in Eq. (3) accounts for the estimated probability of ground truth category $y(i,j)$. Therefore, Eq. (3) shows the standard $softmax$ loss which accumulates the loss of each pixel.

Equations (2) and (3) are used to train the coarse segmentation branch and produce the coarse prediction results that are useful to the following two branches.

3.2.2 Attention branch

After the first segmentation stage, there are still several regions that cannot be determined correctly by the coarse segmentation network. To our knowledge, each input image contains an overwhelming number of easy-to-label pixel instances and a small number of difficult-to-label pixel instances. Focusing on these difficult pixel instances can make the training process converge faster and more efficiently. However, we have no labels to indicate the difficult regions.

From previous studies, hard example mining is one of the commonly used training techniques for machine learning. The traditional implementation is a continuous iterative process that can be divided into two steps. First, the training model is fixed to figure out the difficult examples, and the training set is updated by adding a certain amount of difficult examples. Second, with the updated training set, the model is re-trained.

In this paper, the two-step process of hard example mining is improved to an end-to-end learning framework. For semantic segmentation, each pixel should be assigned a category label. Therefore, a single image contains enough training samples for hard example mining. The attention branch is used to predict the segmentation difficulty of each pixel in terms of coarse segmentation branch results. It shares the same feature extraction network with the coarse segmentation branch. Moreover, they even have the similar network structure. As shown in Eq. (4), $\mathcal{F}_{attention}$ is the attention branch with the learnable weights $\mathcal{W}_a$ and the input is also x, which is the shared feature in Eq. (2). The major difference is that the attention branch is a two-category semantic segmentation network, while the

coarse branch is responsible for learning much more categories.

$$p_a(i,j) = \mathcal{F}_{attention}(x, \mathcal{W}_a) \tag{4}$$

During the training process, the attention branch is supervised by a 0/1 label map that indicates whether it is easy or difficult to predict the pixel label in the corresponding position. The attention branch is cascade behind the coarse segmentation branch, and thus, the label map $\hat{y}$ can be generated by a comparison between the coarse segmentation branch prediction $p_c^k(i,j)$ and the segmentation ground truth $y(i,j)$ in Eq. (5). The value 0 indicates that the coarse segmentation branch misclassifies the pixel, while 1 represents a correct prediction. The 0/1 label map is used as the ground truth by the attention branch in Eq. (6), supervising the attention branch to learn segmentation difficulties of each pixel.

$$\hat{y}(i,j) = \begin{cases} 1, & \arg\max\limits_{k \in \mathcal{K}} p_c^k(i,j) = y(i,j); \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$\mathcal{L}_{attention}(\hat{y}, p_a) = -\frac{1}{N}[\sum_{(i,j)\in I} \log(p_a^{\hat{y}(i,j)}(i,j))] \tag{6}$$

where $\hat{y}$ is the pixel-wise binary label, and $\arg\max\limits_{k \in \mathcal{K}} p_c^k(i,j)$ denotes the category which holds the maximum estimated probability among all the categories $\mathcal{K}$. If this category is equal to the ground truth, positive value 1 will be assigned to $\hat{y}$, indicating that the coarse segmentation branch correctly predicted the pixel-wise labels. Otherwise, 0 will be the new label of this pixel, indicating that it is difficult for the coarse branch to correctly predict this label. The attention branch prediction is an important basis used by the following refine segmentation branch to generate the final prediction.

In addition, during the testing process, the attention branch heuristically filters out the online hard examples.

3.2.3 Refine segmentation branch

The refine segmentation branch is cascade behind the coarse and the attention segmentation branches as shown in the third row in Fig. 2. This branch is more complicated compared with the first two branches. It contains a fine segmentation network $\mathcal{F}_{fine}$, as shown in Eq. (7), and a weighted summation $p_{refine}(i,j)$, as shown in Eq. (8), to refine the final segmentation results.

$$p_f(i,j) = \mathcal{F}_{fine}(x, \mathcal{W}_f) \tag{7}$$

$$p_{refine}(i,j) = p_a(i,j) \cdot p_c(i,j) + (1 - p_a(i,j)) \cdot p_f(i,j) \tag{8}$$

where $p_f(i,j)$ is the prediction result produced by the fine segmentation network $\mathcal{F}_{fine}$ and parameters $\mathcal{W}_f$. After obtaining the coarse prediction, fine prediction, and the attention branch results, the $p_{refine}(i,j)$ is formulated using Eq. (8), which is the weighted summation of the coarse prediction $p_c(i,j)$ and fine prediction $p_f(i,j)$. Furthermore, $p_a(i,j)$ is obtained by the attention branch which means that if the pixel has a high probability of being an easy instance, a higher weight will be assigned to the coarse prediction; otherwise, more attention is given to the fine branch results. Finally, this refine prediction $p_{refine}(i,j)$ and the task labels provide deep supervision to the whole network.

Since it is difficult for the coarse segmentation branch to correctly segment all the pixels, the pixels which can be segmented correctly by the coarse segmentation branch are denoted as easy pixel instances, while the others are denoted as difficult ones. A fine segmentation network is introduced to reclassify the difficult pixel instances. Inspired by the PSPNet[20], pyramid pooling is adopted by the fine segmentation network to extract multi-scale features. The final segmentation result is a weighted summation of the coarse segmentation branch and the fine segmentation network predictions, with the weighting coefficient predicted by the attention branch. The final segmentation result is influenced by the coarse segmentation branch if the pixel is predicted as an easy instance. Otherwise, it is influenced by the fine segmentation network.

$$\mathcal{L}_{refine}(y, p_{refine}) = -\frac{1}{N}[\sum_{(i,j)\in I} \log(p_{refine}^{y(i,j)}(i,j))] \tag{9}$$

Such three branches of our CasNet are successively cascade and constitute an end-to-end learning network with multiple loss functions.

4 Experimental Results

4.1 Datasets

We proved the effectiveness of our CasNet on two semantic segmentation datasets: SIFT Flow[6] and Stanford Background[7]. The SIFT Flow dataset contained 2688 images and 33 labels. Each image had a resolution of 256×256 pixels with BGR three channels. Among them, 2488 images were used as

training set, while the remaining 200 images were used for testing. The dataset defined 33 semantic categories, but the distribution of category was non-uniform.

The Stanford Background dataset contained 715 images of outdoor scenes with different image sizes. All images had approximately 320×240 pixels on average, where each image contained at least one foreground object. To be consistent with previous studies, 5-fold cross validation was used for evaluation. Therefore, 572 images were used for training, while the other 143 were used for testing. The Stanford Background dataset contained eight semantic categories, and the distribution of each category was more balanced than that of the SIFT Flow dataset.

4.2 Network configuration

The implementation of our CasNet is based on an open-source platform Caffe[22]. The stochastic gradient descent algorithm was used by the training procedure for end-to-end training. Our CasNet adopted pre-trained models like in most related works[13] on semantic segmentation. The learning rate was initialized to 1×10^{-4} and decreased by a factor of 10 whenever the accuracy of the validation set stopped improving. The learning rate was repeatedly decreased twice. The momentum and the weight decay were set to 0.9 and 0.0001, respectively.

Data argumentation is widely applied to semantic segmentation in order to avoid overfitting. Different kinds of data argumentation procedures are used to ameliorate the diversity of data samples to improve the generalization ability of the network. In this study, we also employed this kind of procedures with a combination of scaling and translation. Larger input size and batch size can improve the segmentation performance. However, because of both computation and memory limitations, we randomly cropped 233×233 squares from the multi-scale input images, and the training was done in a mini-batch size of 4.

4.3 Ablation study

In this section, we describe the ablation study so as to illustrate the proposed network effectiveness. The comparative results on both the SIFT Flow and the Stanford Background are listed in Table 1.

Method (a) is the coarse branch model which can be regarded as a simple baseline. Here, an FCNN that consisted of two convolutional layers was utilized to predict the semantic results for relatively easy and confident regions. This coarse branch could achieve 89.2% prediction accuracy on SIFT Flow and 88.5% on Stanford Background, respectively.

Method (b) was conducted by a combination of both the coarse and refine segmentation branches. Without the learnable attention branch, we fused the predictions from the coarse branch and the refine one with a static weight (e.g., 0.5) to generate the final prediction results. Compared to the simple coarse branch, the combination improved the overall performance by 0.8% on SIFT Flow and 0.7% on Stanford Background, respectively.

In **Method (c)**, the attention branch is incorporated. The prediction of this attention branch was used as weighting coefficients for promoting the prediction results rather than a static weight. It further enhanced the performance by 1.4% and 1.2%, respectively.

In summary, explicitly modeling the segmentation difficulties of each pixel is quite necessary, and in this study, the combination of coarse and refine branches could collaboratively make dense predictions for all the pixels within a given image. The benefits from the above modules are that the overall mapping accuracy is improved from 89.2% to 91.6% on SIFT Flow and from 88.5% to 89.7% on the Stanford Background, respectively.

4.4 Comparison with state-of-the-art models

We first carry out several comparative experiments on SIFT Flow. The pixel-level semantic segmentation is usually measured by two accuracy metrics: pixel accuracy and class accuracy. The average pixel accuracy is the percentage of the total number of pixels that are correctly classified on the test set, and it is usually evaluated by the intersection-over-union. The average

Table 1 Ablation study on SIFT Flow and Stanford Background datasets.

Dataset	Method	Coarse branch	Refine branch	Attention branch	Pixel acc. (%)
SIFT Flow	(a)	√			89.2
	(b)	√	√		91.0 $\uparrow_{0.8}$
	(c)	√	√	√	91.6 $\uparrow_{1.4}$
Stanford Background	(a)	√			88.5
	(b)	√	√		89.2$\uparrow_{0.7}$
	(c)	√	√	√	89.7$\uparrow_{1.2}$

category accuracy is the average of the correct rate for each category of pixel classification. The experimental results (Table 2) demonstrate that our CasNet achieved an accuracy of 91.6% and outperformed existing state-of-the-art models.

To prove the generalization of the semantic segmentation learning scheme, we tested CasNet on another dataset, Stanford Background, using the same architecture and configurations used in SIFT Flow. Table 3 shows that on the Stanford Background, our CasNet achieves 89.7% pixel average accuracy and 75.4% classification accuracy. Some of the prediction results are shown in Fig. 3.

In Tables 2 and 3, we compare our CasNet with a baseline model that is composed of one ResNet and two FCNN layers, which is indicated by He et al.[1] The baseline model is a typical FCNN segmentation network based on ResNet-50, and it has the same network architecture as the CasNet refine branch. It outperforms the other methods by using a strong feature extractor. Hence, a better feature extractor is quite important to any performance improvements task. However, our CasNet still increases by about 1% performance improvements compared to the baseline segmentation network since our CasNet effectively refines the segmentation results.

Table 2 Segmentation results on SIFT Flow.

Method	Pixel acc. (%)	Class acc. (%)
Liu et al.[6]	76.7	–
Tighe and Lazebnik[23] SVM	75.6	41.4
Tighe and Lazebnik[24] SVM+MRF	78.6	39.2
Farabet et al.[11] natural	72.3	50.8
Farabet et al.[11] balanced	78.5	29.6
Pinheiro and Collobert[25]	77.7	29.8
Liang et al.[17]	84.3	41.0
Long et al.[4]	85.9	53.9
Jin et al.[26]	86.9	56.5
He et al.[1]	90.52	–
Ours	**91.6**	**52.5**

Table 3 Segmentation results on the Stanford Background benchmark.

Method	Pixel acc. (%)	Class acc. (%)
Gould et al.[7]	76.4	–
Tighe and Lazebnik[23]	77.5	–
Eigen and Fergus[27]	75.3	66.5
Singh and Kosecka[28]	74.1	62.2
Lempitsky et al.[9]	81.9	72.4
Liang et al.[17]	83.1	74.8
Jin et al.[26]	86.6	79.0
Ours	**89.7**	**75.4**

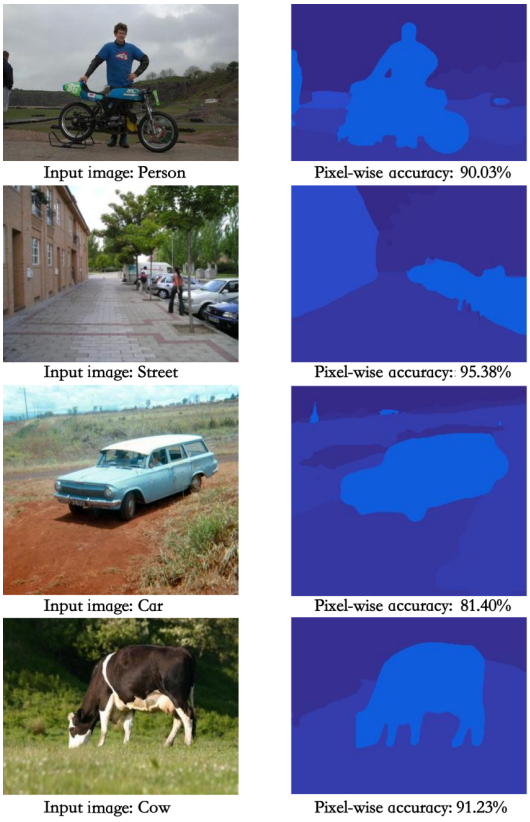


Fig. 3 Prediction results on Stanford Background dataset.

A visualization of the network intermediate feathers is shown in Fig. 4. It consists of four columns. The first column presents the input images, and the second lists the ground truths. The third indicates the hard pixels predicted by the CasNet attention branch, while the last one is the segmentation results of the overall network. It is readily observed from the third and fourth columns that our CasNet is able to make prediction of those hard pixel samples that are indicted in a yellow-colored bounding box in the last column in Fig. 4.

5 Conclusion

Inspired by the concept of hard mining, we propose a novel cascade coarse-to-fine segmentation network architecture. This network comprises three successive branches. The first branch is a coarse segmentation network. The second one is an attention network used to predict the difficulty-to-label pixels, and the third one is a refine segmentation network for generating the final segmentation results. The last branch further combines both the coarse and fine prediction results

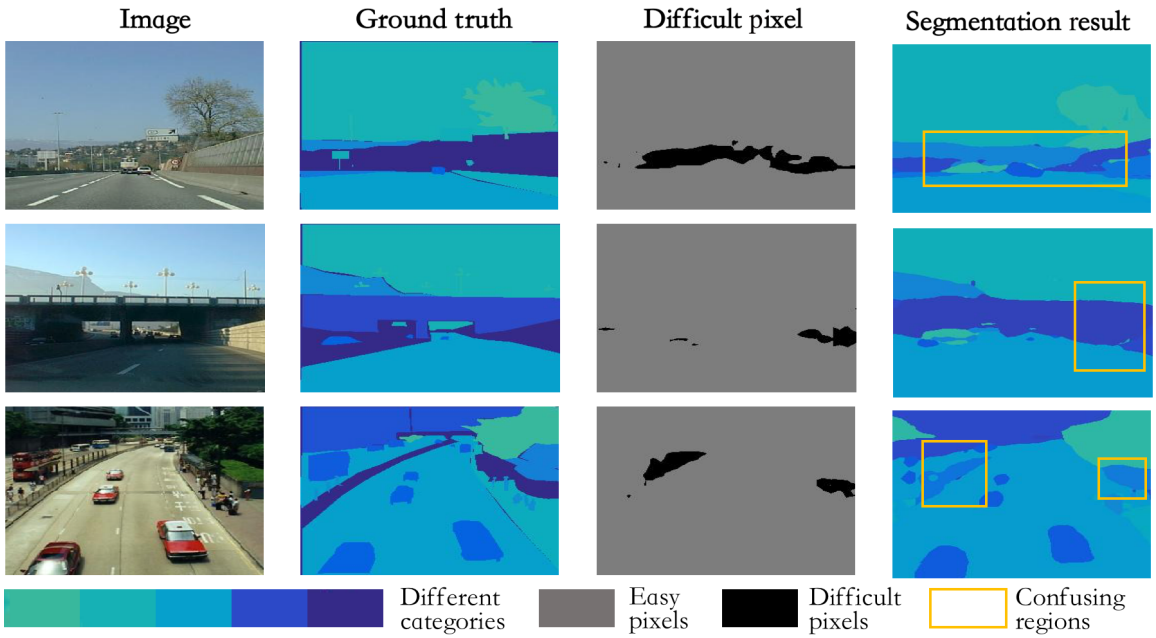


Fig. 4 CasNet visualization predictions.

through a weighting coefficient which is estimated by the attention branch. Finally, the experimental results show that our CasNet outperforms existing models, and it achieves an accuracy of 91.6% and 89.7% for the SIFT flow and the Stanford Background, respectively.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2017YFB1302200) and by Joint Fund of NORINCO Group of China for Advanced Research (No. 6141B010318).

References

[1] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[2] A. Graves, A. R. Mohamed, and G. Hinton, Speech recognition with deep recurrent neural networks, in *Proc. 2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013, pp. 6645–6649.

[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, arXiv preprint arXiv: 1312.6229, 2013.

[4] J. Long, E. Shelhamer, and T. Darrell, Fully convolutional networks for semantic segmentation, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 3431–3440.

[5] K. K. Sung and T. Poggio, Example-based learning for view-based human face detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 39–51, 1998.

[6] C. Liu, J. Yuen, and A. Torralba, Sift flow: Dense correspondence across scenes and its applications, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[7] S. Gould, R. Fulton, and D. Koller, Decomposing a scene into geometric and semantically consistent regions, in *Proc. IEEE 12th Int. Conf. Computer Vision*, Kyoto, Japan, 2009, pp. 1–8.

[8] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr, Associative hierarchical CRFs for object class image segmentation, in *Proc. IEEE 12th Int. Conf. Computer Vision*, Kyoto, Japan, 2009, pp. 739–746.

[9] V. Lempitsky, A. Vedaldi, and A. Zisserman, A pylon model for semantic segmentation, in *Proc. 24th Int. Conf. Neural Information Processing Systems*, Granada, Spain, 2011, pp. 1485–1493.

[10] X. M. He, R. S. Zemel, and M. A. Carreira-Perpinan, Multiscale conditional random fields for image labeling, in *Proc. 2004 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, pp. 695–702.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, Learning hierarchical features for scene labeling, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013.

[12] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, Indoor semantic segmentation using depth information, arXiv preprint arXiv: 1301.3572, 2013.

[13] H. S. Zhao, J. P. Shi, X. J. Qi, X. G. Wang, and J. Y. Jia, Pyramid scene parsing network, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017.

[14] T. Y. Lin, P. Dollár, R. Girshick, K. M. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object

detection, arXiv preprint arXiv: 1612.03144, 2016.

[15] F. Yu and V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv preprint arXiv: 1511.07122, 2015.

[16] F. Visin, A. Romero, K. Cho, M. Matteucci, M. Ciccone, K. Kastner, Y. Bengio, and A. Courville, Reseg: A recurrent neural network-based model for semantic segmentation, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, USA, 2016, pp. 41–48.

[17] M. Liang, X. L. Hu, and B. Zhang, Convolutional neural networks with intra-layer recurrent connections for scene labeling, in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 937–945.

[18] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, arXiv preprint arXiv: 1606.00915, 2016.

[19] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Z. Su, D. L. Du, C. Huang, and P. H. S. Torr, Conditional random fields as recurrent neural networks, in *Proc. 2015 IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 1529–1537.

[20] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, Feedforward semantic segmentation with zoom-out features, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 3376–3385.

[21] A. Veit, J. M. Wilber, and S. Belongie, Residual networks behave like ensembles of relatively shallow networks, in *Advances in Neural Information Processing Systems 29*, Barcelona, Spain, 2016, pp. 550–558.

[22] Y. Q. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 675–678.

[23] J. Tighe and S. Lazebnik, Superparsing: Scalable nonparametric image parsing with superpixels, in *Proc. 11th European Conf. Computer Vision*, Heraklion, Greece, 2010, pp. 352–365.

[24] J. Tighe and S. Lazebnik, Finding things: Image parsing with regions and per-exemplar detectors, in *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 3001–3008.

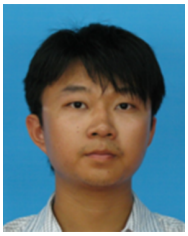
[25] P. O. Pinheiro and R. Collobert, Recurrent convolutional neural networks for scene labeling, in *Proc. 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 82–90.

[26] X. J. Jin, Y. P. Chen, Z. Q. Jie, J. S. Feng, and S. C. Yan, Multi-path feedback recurrent neural networks for scene parsing, in *Proc. 31st AAAI Conf. on Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 4096–4102.

[27] D. Eigen and R. Fergus, Nonparametric image parsing using adaptive neighbor sets, in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 2799–2806.

[28] G. Singh and J. Kosecka, Nonparametric scene parsing with adaptive feature relevance and semantic context, in *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 3151–3157.

Zhidong Deng received the BS degree from Sichuan University, China, in 1986 and the PhD degree from Harbin Institute of Technology, China, in 1991, respectively, both in computer science and automation. From 1992 to 1994, he was a postdoctoral researcher at the Computer Science Department, Tsinghua University, China, where in 1994, he became an associate professor. From 1996 to 1997, he served as a research associate at Hong Kong Polytechnic University, China. From 2001 to 2003, he was a visiting professor at the Washington University in St. Louis, USA. He has been a full professor at Tsinghua University since 2000. His current research interests include artificial intelligence, deep learning, computational neuroscience, computational biology, driverless car, robotics, wireless sensor network, and virtual reality.



Zhenyang Wang received the BS degree from Harbin Institute of Technology, Harbin, China, in 2011 and is pursuing the PhD degree in Tsinghua University, Beijing, China. His research interests include computer vision, deep learning, and machine learning.

Shiyao Wang received the BS degree from Tianjin University, China, in 2014 and is pursuing the PhD degree in Tsinghua University, Beijing, China. Her research interests include computer vision, deep learning, and machine learning.