TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 0?/?? pp???-???
Volume 18, Number 3, June 2013

Attention to Difficulty: A Cascade Coarse to Fine Network Architecture for Semantic Segmentation

Wang Zhenyang, Deng Zhidong*, and Wang Shiyao

Abstract: Semantic segmentation, also know as scene labeling, is a fundamental topic in computer vision. The goal is to assign a category label to each pixel of the image . Recently, convolutional neural networks have attracted increasing attention on semantic segmentation due to the capabilities of extracting hierarchical features. Since it is required to make dense predictions for each pixel, a simple network is hardly to obtain considerable performances on different scenes. In this paper, we propose a cascade coarse to fine network architecture, which aims to pay more attention to the difficult segmentation pixels. There are three branches in the overall network. The first branch is responsible for handling easy regions by producing coarse predictions. Subsequently, the second branch learns to distinguish the relatively difficult pixels from the entire image. Finally, the prediction results are refined by taking a weighted summation of the coarse and fine segmentation results, with the weighting coefficient predicted from the second branch. All above branches focus on their own objectives and collaboratively learn to predict from coarse to fine inference. In order to evaluate predicting performance of the proposed network, we conduct experiments on two public datasets including Sift Flow and Stanford Background Dataset. We show that the three branches can be trained in an end-to-end manner and the experimental results show that compared to all existing models, our CasNet consistently yields the best performance, with accuracy of 91.6% and 89.7%, respectively.

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

1 Introduction

Semantic segmentation has a strong application requirement in the field of environment perception and autonomous self driving car. The goal of semantic segmentation is to identify and assign a category label to each pixel in the image, which requires a complete understanding of the context of the entire image. In recent years, deep learning has made great breakthroughs in computer vision tasks, such as image classification[1], speech recognition[2], and object detection[3]. For the task of semantic segmentation, there are also many methods based on deep learning. Early research attempts to apply CNNs designed for image classification directly to semantic segmentation. Although good segmentation results can be obtained, the prediction results are rough and the edges of objects are difficult to separate correctly. This is mainly cased by the losing of location information. Consequently, the fully convolutional neural network(FCN)[4] is proposed to overcome this disadvantage, and becomes the most popular framework for semantic segmentation.

Firstly, in order to solve the problem of delineate visual objects, conditional random field, Markov random field, Gaussian conditional random field, and other variations are proposed. However, with the rapid development of CNN architectures, an end-to-end training procedure, such as ResNet[1], can achieve the same or even better segmentation results. Besides that,

• Wang Zhenyang, Deng Zhidong, Wang Shiyao are with the Department of Computer Science, Tsinghua University, Beijing 100084, China. E-mail: crazycry2010@gmail.com, michael@tsinghua.edu.cn, sy-wang14@mails.tsinghua.edu.cn
* To whom correspondence should be addressed.
Manuscript received: 2017-09-20; revised: year-month-day; accepted: year-month-day

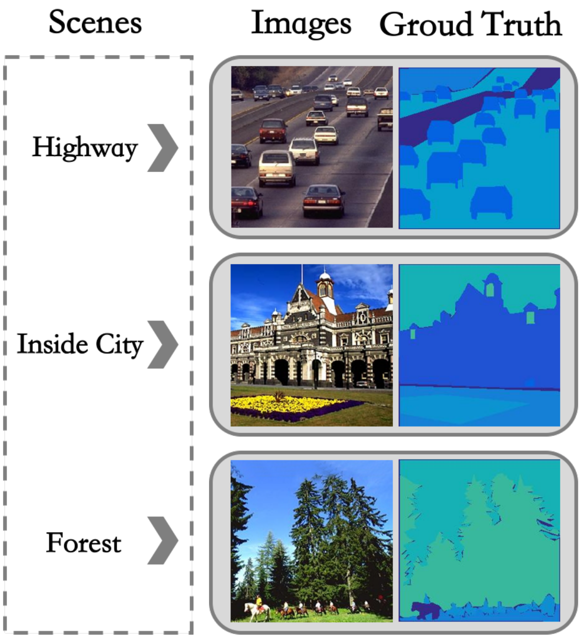


Fig. 1 Examples of semantic segmentation.

we introduce multiple segmentation branches to refine the segmentation results.

Secondly, the unbalance of the easy and difficult pixels can also wreck the convergence of the network. Aiming to deal with the problem of unbalanced distribution, we propose a novel semantic segmentation network framework, which is inspired by hard negative example mining. In fact, hard negative mining is not a new concept. As early as 1994, Sung and Poggio [5] proposed bootstrapping method in their face detection algorithm. The main purpose is to enhance the detection capacity by changing the distribution of difficult samples. Inspired by a similar idea, a cascade coarse to fine network architecture, CasNet for short, is proposed. CasNet is composed by three branches which share a same feature extraction network. The first branch is a coarse segmentation network which is able to handle the easy and confident regions. The second branch is an attention network used to predict the probability of being a hard example for each pixel. For these difficult pixels, the third segmentation branch is proposed to refine the segmentation results.

In conclusion, we propose a cascade coarse to fine network architecture for semantic segmentation. Compared with traditional semantic segmentation networks, CasNet has the following characteristics:

- A cascade segmentation network architecture to refine the final segmentation results.
- Incorporate the idea of hard mining into an attention module.
- Validate CasNet on both Sift Flow[6] and Stanford Background[7] datasets.

2 Related Works

Semantic segmentation aims to relate an unique semantic class (road, water, sea etc.) to each pixel of the input image. Both the global and local features have great impacts on the final performance of this task. Consequently, in terms of feature representations, the mainstream approaches can be divided into traditional hand-craft features and deep features based on deep neural networks.

In recent years, traditional methods have obtained several solutions on image segmentation. Considering the context information, several methods rely on MRF, CRF or other types of graphical models to ensure consistency of labeling[?]. Besides, most methods employ pre-segmentation in order to produce super-pixels or segmented candidates, and extract features from these individual segments along with the combinations of adjacent segments.

Meanwhile, the neural networks, particularly the CNNs that yield hierarchies of features have achieved great progress in this pixel-level prediction task. [11] is the first work that use CNNs for this semantic segmentation. They propose a multi-scale convolution neural network, which extracts the feature representations from different scales of local regions. The experimental results show that the network has capacities of learning texture, shape and domain information implicitly, and achieves better performance than traditional hand-craft features. In addition, the network is also able to generalized to the RGB-D images[12]. To ensure a good visual coherence and a high class accuracy, [13] propose a method to capture long range (pixel) label dependencies in images. They use a recurrent architecture of CNNs to capture a long range label dependency while keeping a tight control over the capacity. The procedure is based on supervised deep learning strategies. [**?**] train the parametric CNNs by sampling image patches, which speeds up the training time dramatically. However, they find that patch-based CNNs suffer from local ambiguity problems. [14] estimate the global potential in a non-parametric framework and propose a large margin based CNN metric for better global potential estimation.

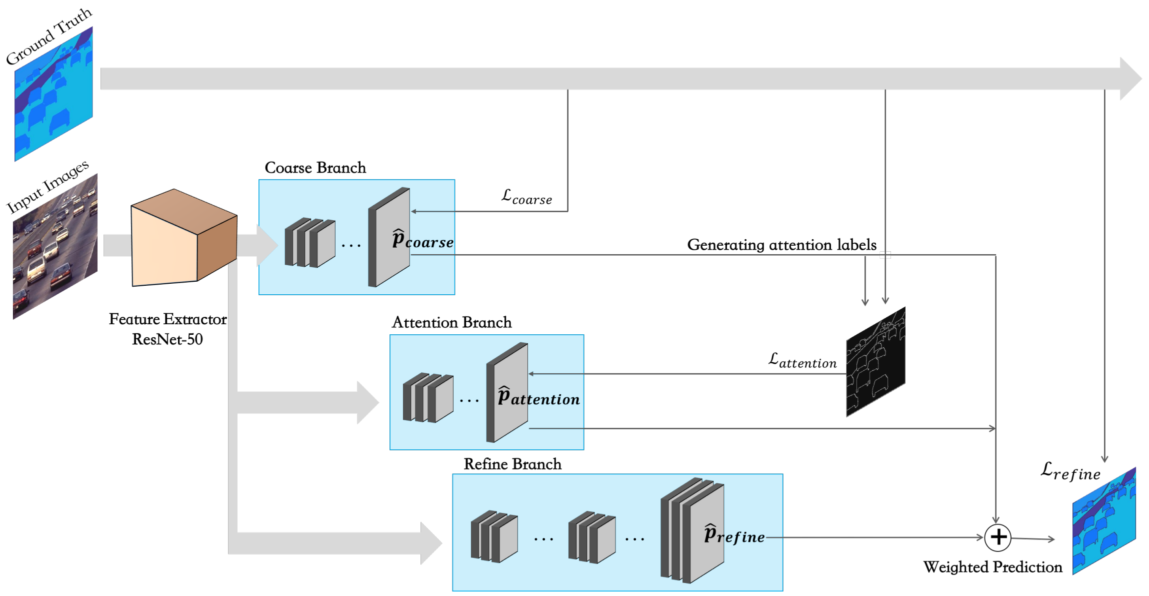


Fig. 2 A Cascade Coarse to Fine Network Architecture for Semantic Segmentation.

[15, 16] introduce quaddirectional 2D Recurrent Neural Networks to model the long range dependencies among pixels which is able to embed the global image context into the compact local representation and significantly enhance their discriminative power.

At the same time, researchers attempt to use the pre-trained CNNs for semantic segmentation. [17] obtain the local and proximal features by using the ConvNet while distant and global features are produced from Alex-net[18]. These above features are further aggregated to predict the categories. Differ from these methods, [19] present a fully convolutional network which is able to take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. They use the CNNs trained on ImageNet as a feature extractor and transfer their learned representations by fine-tuning on the task-specific datasets. [20] propose deep networks by combining the responses at the final DCNN layer with a fully connected CRF. The fully connected pairwise CRF has an ability to capture fine edge details which boosts the final performance. In [21], it presents that the main problem of the current FCN-based models is lack of the incorprating of context information. They exploit the capability of global context information by different-region-based context aggregation through a pyramid pooling module together with the proposed pyramid scene parsing network (PSPNet).

3 Method

Inspired by online hard example mining(OHEM) algorithm, we propose a cascade coarse to fine network architecture CasNet. The framework of the proposed CasNet is shown in Figure 2. Given an image, firstly a ResNet is employed to extract feature representation. Then, the proposed CasNet will be exploited to learn the task-specific objective. In the following subsection, Section 3.1 illustrates how to apply a ImageNet pre-trained model to extract feature representation of the given image. At last, Section 3.2 provides the detailed information of the cascade coarse to fine semantic segmentation network, particularly the three branches which are expected to collaboratively learn to predict from coarse to fine.

3.1 Feature Extraction Network

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

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

where x and y denote the input and output of a layer. The function $F(x, \{W_i\})$ indicates the residual mapping while the W_i represents a group of learnable weights. The operation $F+x$ is performed by a shortcut connection and element-wise which in fact combine the features of multi-scale. It benefits the segmentation task a lot. For semantic segmentation, the context is important to predict the correct label of each pixel instance. But it is difficult to determine the context boundary of each pixel, since different objects may have different contours. The problem becomes more complicated when considering the various perspective of each images. A simple yet effective method to solve this problem is to integrate multi-scale features for label predicting. Consequently, the residual error model itself has the property of extracting and integrating multi-scale features, which can be seen from Equation 1. From the unravelled view by Veit et al. [22], a two-unit ResNet is equivalent to an ensemble of four sub-networks with different receptive fields. So the whole ResNet-50 can be expanded as a linearly growing ensemble of sub-networks, which can extract and integrate multi-scale features.

Besides, there are two improvements adopted by ResNet-50 to make it more suitable for semantic segmentation. First, we only keep the first three pooling layers in order to preserve the resolution. So the final resolution of the prediction is 1/8 of the original input image. Secondly, we replace the convolutional layer in the last two stages with the dilated convolutions. It can help to enlarge the reception field of predicted feature maps.

3.2 A Cascade Coarse to Fine Architecture

The architecture of CasNet is shown as Figure 2. There are three horizontal lines running from input to the target. They are the proposed cascade branches: a coarse segmentation branch as a baseline result, an attention branch to predict the difficulty of labelling each pixel instance, and a refine segmentation branch to refine the final segmentation results. These three branches share a common feature extraction network while focus on their own targets.

3.2.1 The Coarse Segmentation Branch

The coarse segmentation branch is a baseline model for semantic segmentation which is on the first row in Figure 2. We adopt a FCN that consisted of two convolutional layers to predict the semantic classes for relatively easy and confident regions. Since the resolution is 1/8 of the original input image, the feature maps are up-sampled by bilinear interpolation. Finally, a pixel-wise softmax loss is adopted to predict the probabilities of each pixel. We first formulate the coarse segmentation branch which produces the probability map as Equation 2, and the loss function is also defined as Equation 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's location of the given image and x is the input features extracted by feature extraction network in Section 3.1. $\mathcal{F}_{coarse}$ represents the coarse segmentation branch with the trainable weights $\mathcal{W}_c$. And the $p_c(i,j)$ denotes the computed probability of each pixel. Particularly, the $p_c(i,j)$ in Equation 2 is a K dimensional vector (whose elements sum to 1) that represents the estimated probability of the class label taking on each of the K different possible values while the $p_c^{y(i,j)}(i,j)$ in Euation 3 is account for the estimated probability of ground truth category $y(i,j)$. So the Equation 3 shows the standard $softmax$ loss which accumulates the loss of each pixels and then is averaged by the number of pixels N.

Equation 2 and 3 allow to train the coarse segmentation branch and produce the coarse prediction results that are useful for the following two branches.

3.2.2 The Attention Branch

After the first stage of segmentation, there still exists several regions which cannot be determined by the coarse segmentation network. To our knowledge, each segmentation image contain a large number of easy pixel instances and a small number of difficulty pixel instances. Paying more attention on these difficulty pixel instances can make the training process converge faster and efficiently. However, we have no labels that indicate which regions are difficult.

From previous research work, hard example mining is one of the commonly used training techniques for machine learning. The traditional implementation is a continuous iterative process which could be divided into two steps. Firstly, the training model is fixed to figure out the difficult examples, and the training set is updated by adding a certain rate of difficult examples. Secondly, 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. So a single image contain enough training samples for hard example mining. The attention branch is used to predict the segmentation difficulty of each pixel in terms of the results of coarse segmentation branch. It shares the same feature extraction network as the coarse segmentation branch. Moreover, they even have the similar network structure. As shown in Equation 4, the $\mathcal{F}_{attention}$ is the attention branch with the learnable weights $\mathcal{W}_a$ and the input is also x, which is the shared feature as Equation 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 label map with 0/1 values, indicating easy or difficult for the pixel in corresponding position. The attention branch is cascaded behind the coarse segmentation branch, so the label map $\hat{y}$ can be generated by a comparison between the prediction of the coarse segmentation branch $p_c^k(i,j)$ and the segmentation ground truth $y(i,j)$. 0 indicates the pixel is misclassified by the coarse segmentation branch, while 1 represents a correct prediction. The 0/1 label map is used as the ground truth of the attention branch in Equation 6, supervising the attention branch to learn the segmentation difficulty of each pixel.

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

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

where $\hat{y}$ is the pixel-wise binary label. $\arg\max\limits_{k \in \mathcal{K}} p_c^k(i,j)$ 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 this pixel is correctly predicted by coarse segmentation branch. Otherwise, 0 will be the new label of this pixel that represents it is difficult for coarse branch. The prediction of attention will be the important ratio used for generating the final prediction of the following refine segmentation branch.

During the testing process, the attention branch heuristically filter out the hard examples online as well.

3.2.3 The Refine Segmentation Branch

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

Since the coarse segmentation branch is hard to segment all the pixels correctly, the pixel which can be segment correctly by the coarse segmentation branch is denote as easy pixel instances, while the others are difficult ones. A fine segmentation network is introduced to focus on reclassification of the difficult pixel instances. Inspired by the PSPNet [21], pyramid pooling is adopted by the fine segmentation network to extract multi-scale features. And the final segmentation results is a weighted summation of the coarse segmentation branch and the fine segmentation network, with the weighting coefficient predicted by the attention branch. The final segmentation results relay more on the coarse segmentation branch if the pixel is predicted as an easy one. Otherwise, the fine segmentation network takes up a larger proportion.

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

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

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

where $p_f(i,j)$ is the prediction result produced by fine segmentation network $\mathcal{F}_{fine}$ and parameters $\mathcal{W}_f$. After obtaining coarse prediction, fine prediction and the attention results, the $p_{refine}(i,j)$ is formulated in Equation 8 which is the weighted summation of the coarse prediction $p_c(i,j)$ and fine prediction $p_f(i,j)$. $p_a(i,j)$ is obtained by attention branch which means that if the pixel has high probability of being an easy instance, we will assign higher weight to the coarse prediction, otherwise pay more attention

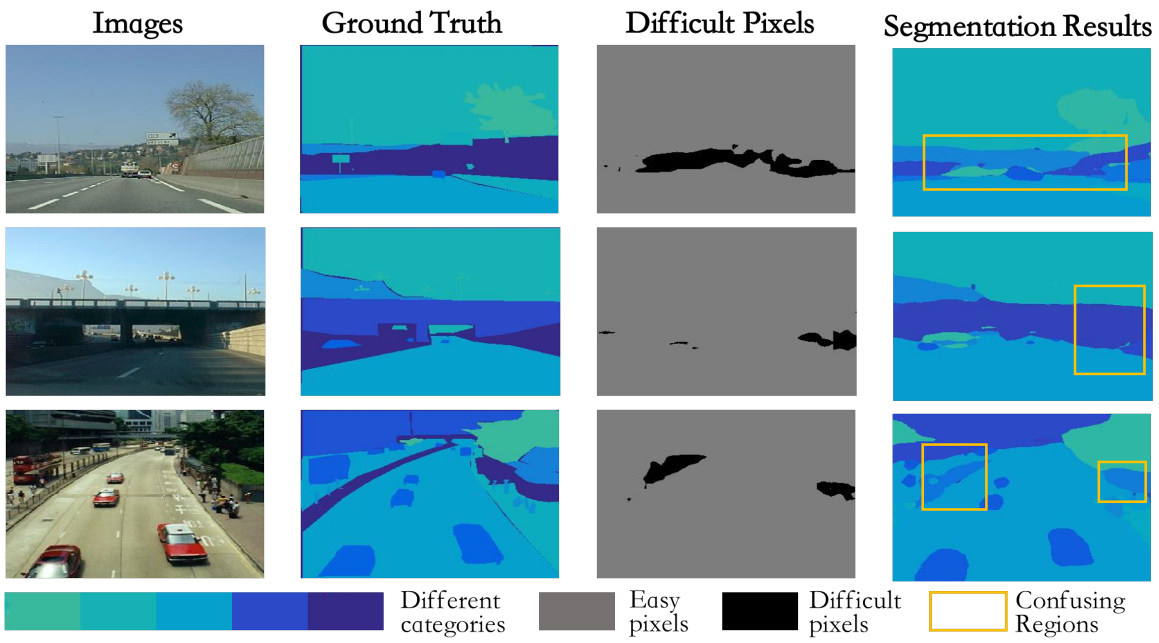


Fig. 3 The visualization predicting result of CasNet.

to the results of fine branch. Finally, this refine prediction $p_{refine}(i,j)$ and the task labels provide deep supervision to the whole network.

The three branches are cascaded one by one, and constitute an end-to-end learning network with multiple loss functions.

4 Experimental results

In this subsection, we present the detailed information about the implementation of CasNet. Moreover, we compare our model to the current state-of-the-art methods and this proposed model achieves superior performances among the existing methods.

4.1 Datasets

We prove the effectiveness of CasNet on two semantic segmentation datasets, which are SIFT Flow and Stanford Background, respectively. The SIFT Flow dataset contains 2688 samples each has 256x256 pixels with RGB channels. 2488 images are used as a training set while the rest 200 images are used for testing. The dataset defines a total of 33 semantic categories, but the distribution of category is nonuniform.

The Stanford Background dataset contains 715 images with different image sizes, but no more than 320x240 pixels. According to the previous research and testing methods, this paper divides the dataset by 5x cross validation method, and 572 images are used as training samples while 143 samples as the test samples. The Stanford Background dataset contains eight semantic categories, and the category distribution is more balanced than the SIFT Flow dataset.

4.2 Network Configuration

The implementation of our CasNet is based on public platform Caffe [23]. The training procedure uses stochastic gradient descent (SGD) algorithm via end-to-end learning. Our model adopt pre-trained models like most related work [21] on semantic segmentation. The learning rate is initially set to 1e-4 and then decreased by a factor of 10 when the validation set accuracy stopped improving. In total, the learning rate is repeatedly decreased 2 times. The momentum and weight decay are set to 0.9 and 0.0001.

Data argumentation is widely applied to semantic segmentation in order to avoid overfit. Various kinds of transformations are used to expand the training sizes so as to improve the generalisation of the proposed networks. In our experiments, we also employ this kind of methods by expanding the input images 1-2 times and randomly cropping 233x233 area for training. Larger input size and mini-batch can help improve the segmentation performance. Due to the limitation of both computation and memory, we only use 233*233 images and the mini-batch is 4.

4.3 Comparison Results

During the comparison, we first conduct several experiments on SIFT Flow dataset. Firstly, we compare CasNet with a baseline model, which is composed by a ResNet and two FCN layers. The baseline model is a typical FCN segmentation network based on ResNet-50, and has a same network architecture as the coarse branch of CasNet. The comparison result is show as Table 1, CasNet leads to about 1% gain in comparison to the baseline ResNet segmentation network. Since cascade coarse to fine network architecture is effective for refining the segmentation results.

Table 1 The Comparison Result with Baseline Model on SIFT Flow.

Methods	Pixel acc.
Baseline(ResNet)	90.52
CasNet	91.6

A visualisation of the intermediate network feature is show in Figure 3, which consists of four columns. The first column presents the input images and second column shows the ground truth labels. In particular, the third column indicates hard pixels predicted by the attention branch of CasNet while the last column is the segmentation results of the network. As can be seen from the third and fourth columns of Figure 3, the CasNet can indeed predict the hard pixel samples which are indicated in the yellow square box of last column in Figure 3.

In addition, on the SIFT Flow dataset, we compare the CasNet to other segmentation methods. Pixel-level semantic segmentation is usually measured by two accuracy measures: Pixel Accuracy and Class Accuracy. The average pixel accuracy is the percentage of the total number of pixels that correctly classified on the test set, and it is usually measured by the intersection-over-union (IoU). The average category accuracy is the average of the correct rate for each category of pixel classification. The experimental results which is shown in Table 2 which prove that CasNet achieves an accuracy of 91.6%, and outperform the current state-of-the-art results.

Table 2 The Segmentation Results on SIFT Flow.

Methods	Pixel acc.	Class acc.
Liu et al.[6]	76.7	-
Tighe et al. SVM[25]	75.6	41.4
Tighe et al. SVM+MRF[26]	78.6	39.2
Farabet et al. natural[11]	72.3	50.8
Farabet et al. balanced[11]	78.5	29.6
Pinheiro et al.[13]	77.7	29.8
Liang et al. RCNN[24]	84.3	41.0
Shelhamer et al. FCN-8s[4]	85.9	53.9
CasNet	91.6	52.5

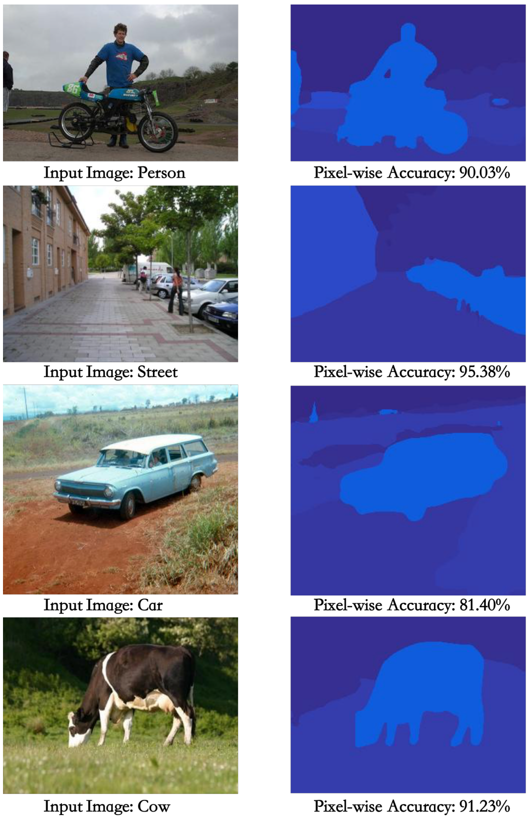
In order to prove the generalisation of the learning scheme of the semantic segmentation, we test the CasNet on another dataset called Stanford Background by using the same architecture and configurations. Table 3 shows that on the Stanford Background dataset, CasNet achieved 89.7% pixel average accuracy and 75.4% classification accuracy. Some of the prediction results are shown in Figure 4, which can be clearly seen the predicted results.

Table 3 The Segmentation Results on Stanford background.

Methods	Pixel acc.	Class acc.
Gould et al.[7]	76.4	-
Tighe and Lazebnik[25]	77.5	-
Eigen and Fergus[28]	75.3	66.5
Singh and Kosecka[27]	74.1	62.2
Lempitsky et al.[10]	81.9	72.4
Liang et al. RCNN[24]	83.1	74.8
CasNet	89.7	75.4

5 Conclusions

Inspired by the idea of hard mining, we propose a novel cascade coarse to fine segmentation network architecture. This network include three branches, in which the first is a coarse segmentation network. While the second branch is an attention network used to predict the difficulty of segmenting each pixel. And the third branch refines the final segmentation results by taking a weighted average of multiple branches. In order to evaluate the performance of CasNet, we conduct experiments on two public datasets including Sift Flow Dataset and Stanford Background Dataset. We show how to train these three branches in an end-to-end manner. And finally, the experimental results show that compared to all existing models, our CasNet consistently yields the best performance, with the accuracy of 91.6% and 89.7%, respectively.



Input Image: Person
Pixel-wise Accuracy: 90.03%
Input Image: Street
Pixel-wise Accuracy: 95.38%
Input Image: Car
Pixel-wise Accuracy: 81.40%
Input Image: Cow
Pixel-wise Accuracy: 91.23%

Fig. 4 The predict results on Stanford Background dataset.

Acknowledgements

This work was supported in part by the National Science Foundation of China (NSFC) under Grant Nos. 91420106, 90820305, and 60775040, and by the National High-Tech R&D Program of China under Grant No. 2012AA041402.

References

References

[1] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[2] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013: 6645-6649.

[3] Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.

[4] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.

[5] Sung, K-K., and Tomaso Poggio. ”Example-based learning for view-based human face detection.” IEEE Transactions on pattern analysis and machine intelligence 20.1 (1998): 39-51.

[6] Liu C, Yuen J, Torralba A. Sift flow: Dense correspondence across scenes and its applications[J]. IEEE transactions on pattern analysis and machine intelligence, 2011, 33(5): 978-994.

[7] Gould S, Fulton R, Koller D. Decomposing a scene into geometric and semantically consistent regions[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009: 1-8.

[8] Russell C, Kohli P, Torr P H S. Associative hierarchical crfs for object class image segmentation[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009: 739-746.

[9] Kumar M P, Koller D. Efficiently selecting regions for scene understanding[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 3217-3224.

[10] Lempitsky V, Vedaldi A, Zisserman A. Pylon model for semantic segmentation[C]//Advances in neural information processing systems. 2011: 1485-1493.

[11] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8): 1915-1929.

[12] Couprie C, Farabet C, Najman L, et al. Indoor semantic segmentation using depth information[J]. arXiv preprint arXiv:1301.3572, 2013.

[13] Pinheiro P H O, Collobert R. Recurrent Convolutional Neural Networks for Scene Labeling[C]//ICML. 2014: 82-90.

[14] Shuai B, Wang G, Zuo Z, et al. Integrating parametric and non-parametric models for scene labeling[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 4249-4258.

[15] Shuai B, Zuo Z, Wang G. Quaddirectional 2d-recurrent neural networks for image labeling[J]. IEEE Signal Processing Letters, 2015, 22(11): 1990-1994.

[16] Shuai B, Zuo Z, Wang B, et al. Dag-recurrent neural networks for scene labeling[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 3620-3629.

[17] Mostajabi M, Yadollahpour P, Shakhnarovich G. Feedforward semantic segmentation with zoom-out features[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3376-3385.

[18] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[19] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.

[20] Chen L C, Papandreou G, Kokkinos I, et al. Semantic image segmentation with deep convolutional nets and fully connected crfs[J]. arXiv preprint arXiv:1412.7062, 2014.

[21] Zhao H, Shi J, Qi X, et al. Pyramid Scene Parsing Network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[22] Veit, Andreas, Michael J. Wilber, and Serge Belongie. ”Residual networks behave like ensembles of relatively shallow networks.” Advances in Neural Information Processing Systems. 2016.

[23] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.

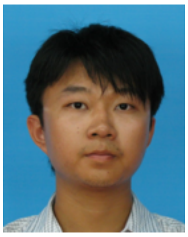
[24] Liang, Ming, Xiaolin Hu, and Bo Zhang. ”Convolutional neural networks with intra-layer recurrent connections for scene labeling.” Advances in Neural Information Processing Systems. 2015.

[25] Tighe J, Lazebnik S. Superparsing: scalable nonparametric image parsing with superpixels[C]//European conference on computer vision. Springer Berlin Heidelberg, 2010: 352-365.

[26] Tighe J, Lazebnik S. Finding things: Image parsing with regions and per-exemplar detectors[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 3001-3008.

[27] Singh G, Kosecka J. Nonparametric scene parsing with adaptive feature relevance and semantic context[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013: 3151-3157.

[28] Eigen D, Fergus R. Nonparametric image parsing using adaptive neighbor sets[C]//Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012: 2799-2806.



Wang Zhenyang received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2011 and has been pursuing the Ph.D. degree from Tsinghua University, Beijing, China, since 2011, both in computer science. Her research interests include computer vision, deep learning, and machine learning.

Wang Shiyao received the B.S. degree from Tianjin University, Tianjin, China, in 2014 and has been pursuing the Ph.D. degree from Tsinghua University, Beijing, China, since 2014, both in computer science. Her research interests include computer vision, deep learning, and machine learning.

Deng Zhidong received the B.S. degree from Sichuan University, Chengdu, China, in 1986 and the Ph.D. degree from Harbin Institute of Technology, Harbin, 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, Beijing, China, where in 1994, he became an Associate Professor. From 1996 to 1997, he served as a Research Associate at Hong Kong Polytechnic University, Kowloon, Hong Kong, China. From 2001 to 2003, he was a Visiting Professor at the Washington University in St. Louis, St. Louis, MO, 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.