

Superpixel-based semantic segmentation trained by statistical process control

Hyojin Park
wolfrun@snu.ac.kr

Jisoo Jeong
soo3553@snu.ac.kr

Youngjoon Yoo
i0you200@snu.ac.kr

Nojun Kwak
nojunk@snu.ac.kr

Department of Transdisciplinary
Studies,
Seoul National University,
Republic of Korea.

Abstract

Semantic segmentation, like other fields of computer vision, has seen a remarkable performance advance by the use of deep convolution neural networks. However, considering that neighboring pixels are heavily dependent on each other, both learning and testing of these methods have a lot of redundant operations. To resolve this problem, the proposed network is trained and tested with only 0.37% of total pixels by superpixel-based sampling and largely reduced the complexity of upsampling calculation. The hypercolumn feature maps are constructed by pyramid module in combination with the convolution layers of the base network. Since the proposed method uses a very small number of sampled pixels, the end-to-end learning of the entire network is difficult with a common learning rate for all the layers. In order to resolve this problem, the learning rate after sampling is controlled by *statistical process control* (SPC) of gradients in each layer. The proposed method performs better than or equal to the conventional methods that use much more samples on *Pascal Context*, *SUN-RGBD* dataset.

1 Introduction

The purpose of semantic segmentation is to segment a given image and identify the semantic information of each segment. Like many other computer vision applications, architectures based on convolutional neural networks (CNN) have been introduced and applied to improve performance of the semantic segmentation [9, 24, 26]. Especially, since the introduction of the *fully convolutional network* (FCN) based architecture proposed in [24], which showed promising performance on semantic segmentation, many studies follow this methodology [8, 21, 26, 63]. A general CNN-based semantic segmentation is largely divided into three parts as shown in Figure 1(a). First, feature extraction is performed as in [28]. Second, we upsample the reduced feature map to the original size and finally calculate probability values for each semantic class using a 1-by-1 convolution for each pixel. This is also referred to as dense classification or pixel-wise classification.

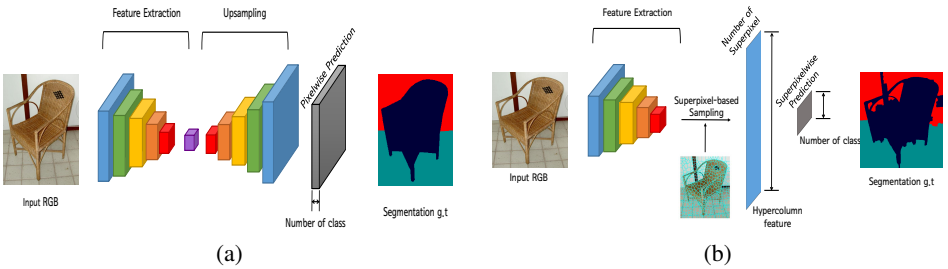


Figure 1: (a) Conventional CNN for semantic segmentation. Usually they need upsampling layer and do pixel-wise classification (b) Our proposed method. We do not need to use upsampling method and reduce the operations significantly by supersixel based sampling

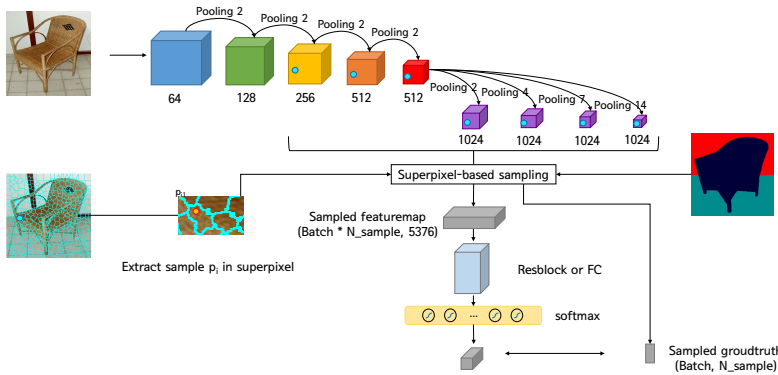


Figure 2: Overall structure of the proposed method (HP-SPS)

Semantic segmentation has a couple of problems owing to the pixel-wise classification. First, pooling is used multiple times for a wider receptive field to get richer information as in [1]. Thus, the feature map becomes much smaller than the original size. In order to solve this problem, researchers devised schemes which used less pooling layers but with a wider receptive field, and made use of CNN-based upsampling methods. Second, neighboring pixels are likely to have close relationship and share similar information. Therefore, they are likely to belong to the same semantic class. However, the class of a pixel is calculated independently without this relation. If we group the pixels that are likely to belong to the same class, we can reduce the inefficient computation and time complexity. Also, because the neighboring pixels share information, the basic assumption of the stochastic gradient descent (SGD), that the data have independent identical distribution (IID), is violated and the learning becomes inefficient [2, 3, 4].

However, none of the above studies has solved the problems mentioned above completely. The use of upsampling requires additional computation, even though less pooling is performed. In addition, there is still a problem of pixel-wise classification. This paper proposes a novel semantic segmentation algorithm based on pyramid module in combination with supersixel-based sampling to solve the problems mentioned above. More specifically, 1) we use the pyramid module to broaden the receptive field and enable feature extraction with enhanced scale-invariant properties. 2) We do not use common upsampling method [5] and changed the method more efficiently by applying supersixel-based sampling method that reduces the amount of computation in the training and testing, by using only 0.37% of the total pixels through sampling. 3) To solve the learning speed drop problem caused by the

sampling, the learning rate is further tuned to acquire stable gradients, by using statistical process control [25].

The organization of the paper is given as follows. In Section 2, the related works are briefly overviewed. Section 3 presents the proposed method of semantic segmentation using superpixel-based sampling with pyramid module and hypercolumn feature extraction. We also propose new gradient control method to compensate for the learning problem in sampled networks. Section 4 shows the experimental results using Pascal context with subtraction experiment and additional experiment using SUN-RGBD finally Section 5 concludes the paper.

2 Related Works

The performance of semantic segmentation has improved a lot since the introduction of CNN-based methodologies [9, 24, 26]. Since the introduction of the FCN [24], the scale invariant features could be obtained in a fast and easy way without the use of the image pyramid and semantic segmentation has been conducted using the popular networks such as VGGNet [28]. After then, in many methods[9, 8, 20], pooling was used to obtain features with wider receptive fields [8]. However, successive application of pooling makes the resolution of a feature smaller and it becomes quite difficult to recover the original resolution of an image. To resolve this problem and to obtain features with wider receptive fields using the reduced number of pooling, some new types of filters such as dilated convolution [33] and atrous convolution [8] have been introduced.

Nonetheless, recovering the resolution of feature map and preserving fine-detail information a major obstacle for the FCN based segmentation. Therefore, researchers have proposed the methods concatenating the features in intermediate layers as well as the final layer to obtain high quality features. The studies in [9, 8, 20, 34] belong to this line of research. The hypercolumn [11] of a pixel is defined as a stacked vector of all features in feature maps in every different layers corresponding to the pixel. The methods of *shift and stitch* [23, 24], *deconvolution* [20], and *unpooling* [9, 8, 20] gradually recover the missing information by adding extra shallow layer features of the same target resolution [34].

There are also studies different from this framework of *extract-and-expand* as in [24]. Lin *et al.* [11] combine the structure of conditional random field (CRF) and CNN, where the CNN computes the potential function value through joint learning. Then, a mean-field approximation is applied to perform semantic segmentation. In [9], they randomly sampled the feature map to cut off dependency among the pixels and delivered the gradient only to the sampled ones for statistical efficiency. However, at the time of inference, they must create hypercolumn feature maps that are in the same size as the original image. Also, at the time of training, they use as much as 10 times the number of samples than ours. Therefore very redundant operation is still performed in [9]. In [19], multi-layer features were created by average pooling in each superpixel for semantic segmentation. This study aims at rich feature representation for various resolutions without using a complex additional model. At the same time, it also enhances efficiency in testing by using superpixel. Although this work has some commonality with ours in that it uses superpixels in training and testing, the difference is that unlike theirs, 1) we used sampling to reduce the computational complexity and 2) hypercolumns of pyramid modules for better representation power.

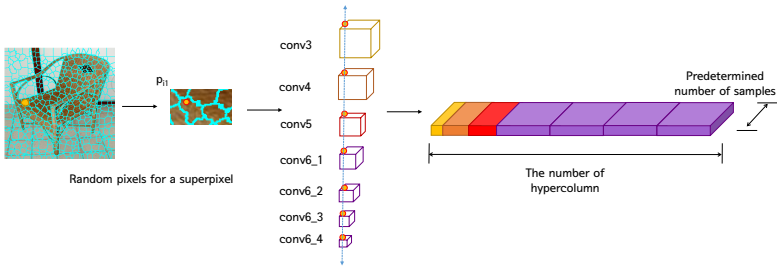


Figure 3: Hypercolumns of random pixels

3 Proposed Method

Figure 2 shows the overall framework of the proposed semantic segmentation method with superpixel-based sampling. First, the CNN features are extracted for a given image. Second, the image is divided into small regions by using a superpixel technique. Third, each region in a superpixel is represented by one or two random pixels in the region. Fourth, for each sampled pixel, hypercolumn features are generated by concatenating all the feature maps for each layer passing the pixel. Then, using the extracted hypercolumn features, segmentation is conducted for each sampled pixel in a superpixel by using the segmentation network composed of Resblock [14] or FCN. The proposed method is named as HP-SPS, an abbreviation for *Hypercolumns of Pyramid module with SuperPixel-based Sampling*. The detailed network design can be found in Table 1. More detailed explanation of each step is given below.

3.1 Feature Extraction through Hypercolumns of Pyramid Module

For semantic segmentation, we first map the input images into multi-layer CNN features having large receptive fields in order to make the features robust to scale and translation variations as in [4]. In the proposed feature network, we use the VGGnet [28] until conv5 stage. Then, the receptive field is expanded by four parallel pooling layers with pool sizes of 2, 4, 7, and 14, respectively. After pooling, 3×3 convolution with 1024 dimension is performed on each output of the four pooling. We found that the segmentation performance is better when the kernel size is equivalent to the stride size, so that the convolution intervals do not overlap. Also kernel size of 3 was better than that of 1 because it considers the information of surrounding features. We set the output feature dimensions the same (1,024) because if the feature dimensions of the layers are much different from each other, it cause to use only specific scale information [22]. After mapping the image into the feature extraction network, we concatenate the feature maps for each layers using hypercolumn method [14]. To apply the method, we track the pooling locations of the target pixel through conv3, conv4, conv5 and all the four conv6 in the proposed feature extraction network, then concatenate all the corresponding feature maps of different layers as shown in Figure 3. We note that the normalization step is required to balance the scale between the layers. l_2 normalization is adopted in our method as in [28].

3.2 Super-pixel Based Sampling for Learning and Inference

In our method, superpixel is adopted as a basic block for semantic segmentation to resolve the problem of redundancy in learning and prediction. We weakly segment the image using

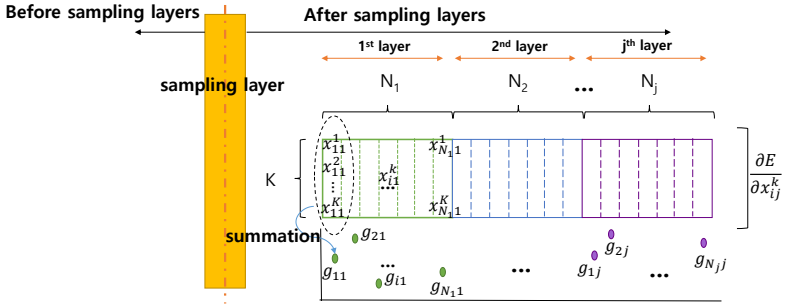


Figure 4: Method of making gradient plot from the gradients g_i 's

simple Linear Iterative Clustering (SLIC) and randomly sample the representative pixel p_i for each superpixel $s_i, i = 1, \dots, N$. Since the number of SLIC superpixels differs from image to image, we randomly select one or two pixels from the randomly chosen superpixel s_i such that the number of the total selected pixels be same for all images. Then, we extract the hypercolumn feature f_i for the selected pixel p_i representing the superpixel s_i and also record the segmentation label l_i at same position as in Section 3.1. By selecting both the superpixel and the pixels in it randomly, we try to meet the IID assumption of SGD.

Using the extracted hypercolumn feature f_i , we train the segmentation network in Figure 2 with label l_i for the superpixel s_i . For the segmentation network, Resblock [16] and FCN [24] are adopted. At test time, we assign the same class to all the pixels inside of superpixel for dense prediction. Unlike other studies, thanks to superpixel-based sampling, we do not need to recover feature map to the original resolution. Also because most of other works perform pixel-wise independent estimation with 1-by-1 convolution, they incur much computational complexity. Because neighbor pixels have high probability of sharing the same semantic information, our superpixel based sampling method reduces this complexity a lot.

However, in our method, the training is not straightforward because the number of the training sample is drastically decreased (only 0.374% of the sample compared to the pixel-wise case). This is severe considering the large dimension of the input f_i . To train the network with a relatively small number of samples, learning rate of each layer γ of the stochastic gradient should be increased so that the network parameter can be changed enough to reflect the effect of the sample p_i . When increasing the learning rate, correspondingly, we suffer from noisy gradient problem and it is critical especially for the case when just a few input samples are provided. Therefore, we propose statistical process control (SPC) method to control the noisy gradient problem and successfully train the network parameters using a restricted number of input samples. We analyze the gradients from the experiments with low and high learning rates for all the layers (*all low γ* and *all high γ* , respectively) and suggest a hybrid learning rates method where in some layers γ is set to have low values, while in other layers it is set to have high learning rate (*hybrid γ*). Note that the control of γ by SPC is applied only to the layers after the sampling.

3.3 Statistical Process Control for Tuning Learning Rates

For training the proposed segmentation network, considering the large solution space of the network parameters, relatively a small number of samples is provided. Consequently, it is difficult to extract a proper gradient for each layer, and adjusting the layer-wise learning rate is widely used solution for mitigating the effect of the noisy gradient [16]. In this section, we

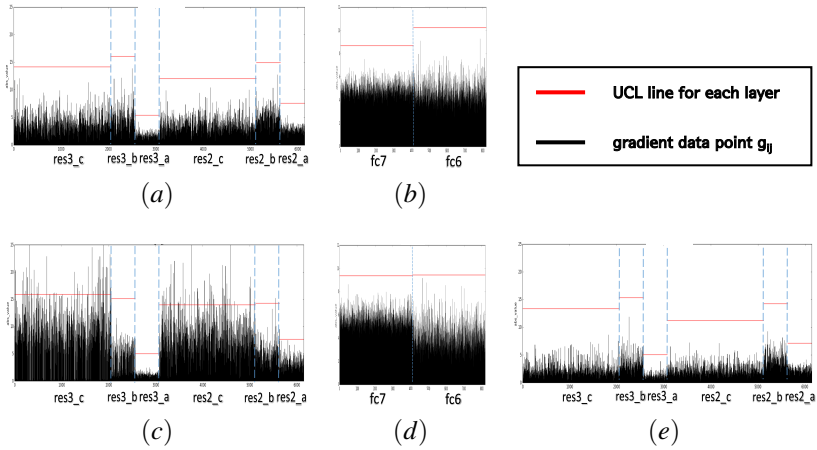


Figure 5: Gradient plot with UCL. The closer to the origin on the x-axis, the farther away from the input. (a) all low γ gradient plot in Resblock network, (b) all low γ gradient plot in FC network, (c) all high γ gradient plot in Resblock network, (d) all high γ gradient plot in FC network, (e) hybrid γ gradient plot in Resblock network

introduce SPC method for efficient selection of the layer-wise learning rate.

If we look at CNN as a manufacturing system, we produce a gradient through a process called backpropagation, where the quality of production depends on the parameters controlling the process. If the scattering of the gradients is too large, the learning will not work properly. On the other hand, if the scattering is small, the learning will be done properly and the good performance will be obtained. SPC [25] is the most popular method for quality control which is based on statistics. SPC monitors data in the current state to understand whether current state is good for producing high quality products. Here, we use the control chart method where the *upper control limit* (UCL) and the *lower control limit* (LCL) are used in the quality control. Both UCL and LCL are set by the mean and the standard deviation of the process and they determine if there is a problem with the process.

In our learning, control chart method is applied to the gradient after 12 epochs. We only apply UCL because we use the absolute value of a gradient. We make a control chart for the same input at the same iteration using each network structure. After sampling, the depth of feature map in the j -th layer has N_j dimension. We make a gradient data point g_{ij} for the i -th feature map or slice ($i \in \{1, \dots, N_j\}$) by summing the absolute gradient values of each feature in the slice. This is shown in (2) and Figure 4. Here, K is the number of features in a slice and x_{ij}^k is the k -th feature of the i -th feature map in the j -th layer. First, the mean μ_j and the standard deviation σ_j of the gradient are calculated as in (1). Second, σ_j^{low} is the j -th layer's standard deviation of the gradients from the experiment of *all low* γ . UCL is defined using μ_j and σ_j^{low} , as in (2). The constant C is a parameter for controlling the regularity of the process, and in our method, it is set to 6.

$$g_{ij} = \sum_{k=1}^K \left\| \frac{\partial E}{\partial x_{ij}^k} \right\|, \quad \mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} g_{ij}, \quad \sigma_j = \sqrt{\frac{1}{N_j} \sum_{i=1}^{N_j} (g_{ij} - \mu_j)^2}, \quad (1)$$

$$UCL = \mu_j + C\sigma_j^{\text{low}} \quad (2)$$

Figure 5 shows the plots of the gradient sums g_{ij} 's in different layers after the sampling, and the red line in each subfigure represents the UCL line from *all low* γ standard deviation

1x1,1024	Eltwise sum	1x1,512	Eltwise sum	1x1,1024	Eltwise sum	1x1, 4096	1x1, 4096	
1x1,1024		1x1,512		1x1,1024				
1x1,2048		1x1,2048		1x1,2048				
1x1,2048	Block 1		Block 2		Block 3		Fully Convolution	
ResBlock			ResBlock			FC6		FC7

Table 1: Network design after sampling layer for semantic segmentation

σ_j^{low} . In Figure 5(c)(d), we show the gradient plot for the *all high* γ experiment where the γ is set to 10 times the case for the *all low* γ experiment in Figure 5(a)(b). As in figure 5(c), since the magnitude of g_{ij} exceeds the red line and also starts to fluctuates after passing res3_c layer, we reduce the learning rate γ of the convolution layers placed after res3_c layer to eliminate noisy gradient from the high learning rate. We note that the learning rate regarding res2_c is not controlled despite that the gradient of res2_c is unstable either. It is because the backpropagation for res3_c is performed before the calculation of gradient for res2_c. It means that the control of the gradient for res3_c also brings change of the gradient in res2_c. Finally reducing the gradients for the higher layer (res3_c) automatically brings stable gradients of all the layers. Figure 5(e) depicts the experiment with hybrid γ , which clearly shows g_{ij} becomes stable and results in improvement in performance. For the FC case, Figure 5(d) shows less fluctuations in spite of the high value of γ . In such cases, we make no changes in the learning rate γ .

4 Experiment

Pascal Context dataset [20], which is based on VOC2010 dataset, has additional detailed categories. The most frequently used 59 categories are selected from more than 400 categories and the rest are merged into one category. The number of training and validation images are 4,998 and 5,105, respectively.

SUN-RGBD dataset [29] is more difficult to segment than the Pascal Context dataset due to variations in the shape, size and pose of the objects in the same category. The dataset contains RGB and depth images from NYU depth v2 [27], Berkeley B3DO [14], and SUN3D [8] datasets and has 37 indoor environment categories. The training set contains 5,285 images, and the test set has 5,050 images.

Here, pixel accuracy (pixel Acc.) measures the ratio of correctly estimated pixels. Mean accuracy (mean Acc.) is the average of category-specific pixel accuracy and mean intersection of union (mean IU) is the average of category-specific ratio of the intersection versus union between the ground truth and the estimation result.

4.1 Implementation Detail

All the experiments were performed using the caffe library [15] and we use Pixelnet [9] pre-trained caffe model. For the new layer, we used "xavier" [11] initialization and dropout [80] from the pyramid module. We set the momentum as 0.9 and the weight decay 0.0005. The learning rate started at 10^{-6} and decreased to 10^{-7} after 16 epoch. The superpixel was created using SLIC[10].

4.2 Result from SPC

To analyze the effect of the number of the superpixels between the performance, a cross experiment was conducted as in Table 2. For ‘Train-250s’, ‘Train-750s’ and ‘Train-1600s’,

image size:448 × 448	Train-750s		Train-1600s		image size:224 × 224	Train-225s		Train-750s	
	mean Acc.	mean IU	mean Acc.	mean IU		mean Acc.	mean IU	mean Acc.	mean IU
Test-750s	46.975	34.976	48.275	35.603	Test-225s	45.976	32.866	46.618	33.84
Test-1600s	47.081	35.061	48.397	35.713	Test-750s	46.343	33.161	47.256	34.431

Table 2: Cross experiment between number of sampling and performance

	all low γ	all high γ	hybrid γ_1	hybrid γ_2
mean Acc.	50.808	50.691	51.503	51.928
mean IU	37.936	37.828	38.911	39.659

Table 3: Performance of different γ policy using Resblock

250, 750 and 1600 superpixels were used. In the same way, ‘Test-250s’, ‘Test-750s’ and ‘Test-1600s’ were conducted by 250, 750 and 1600 number of superpixels. We used a same network to both 224×224 and 448×448 image size, but only detailed setting of pyramid module was slightly changed. As shown in Table 2, the performance was proportional to the number of samples in the train. Surprisingly, the number of superpixels in the test was not significantly affected. That is, as mentioned in Section 3.3, a small number of sample was provided compared with a large solution space.

Table 3 shows performance improvement of tuning learning rate of each layer γ with SPC analysis. In the experiment, *high* γ was set to 10 times larger learning rate than *low* γ case. We note that setting high value of γ did not always occur good results. By using the proposed statistical process control (SPC) technique, we determined which layer’s γ should be small. The performance was improved when reducing the γ value of the last convolution layer before softmax layer in Resblock network. In *hybrid* γ_1 , the γ in last convolution layer was set to 5 times larger than learning rate than *low* γ case. The hybrid γ_2 used the equivalent learning rate to *low* γ case in last convolution layer. Consequently, this experiment shows that the higher performance can be achieved by setting learning rate of the last layer γ to be smaller than usual case.

4.3 Pascal Context

For training and testing the Pascal Context data, we used provided train/validation set to train/test the proposed method. All the image were resized to 448×448 , and 750 superpixels are used for each image. To analyze the effect of the superpixel method (SLIC), we set the baseline method which uses the same number of the grid dividing the image. The ‘sample(superpixel)’ and ‘sample(grid)’ in Table 4 refers to the case using superpixel and grid for dividing the region of the image. Both sampling methods generate hypercolumn using conv 3, 4, 5 and 6 in Figure 2, but conv 6 was not used for pyramid module. Instead, pooling with size 14×14 was used for having same receptive field and convolution filter output feature depth is 4096 for same depth of feature map with HP-SPS.

For segmentation, we used fully convolution layer (FC). As shown in Table 4, mean Acc./mean IU were increased when SLIC is used, and HP-SPS which using pyramid module enhanced the performance more about 2.4% and 1.5%, respectively. This means that large receptive field is not the sufficient condition of the high performance. In order to show the effectiveness of SPC, we used two methods each of which uses 3 Resblocks or 2 FCs, respectively, mentioned in Table 1. For Resblock, we do same procedure as described in Section 4.2. For the *hybrid* γ FC 10 case, FC layers will have 10 times higher learning rate after sampling and the performance was stable irrespective of the setting of the γ . But *hybrid* γ FC 15 case, FC layer have 15 times higher than before sampling and we should adjust γ from SPC analysis. We reduced the γ of ‘fc7’ and it showed slightly better performance.

PixelNet which uses random sampling is related to our train method. As shown in Table

proposed method	mean Acc.	mean IU	Others	mean Acc.	mean IU
sampling (grid)	44.735	33.179	FCN-8s [12]	50.7	37.8
sampling (superpixel)	45.536	34.457	DeepLab (without CRF) [8]	-	37.6
HP-SPS	47.95	35.929	CRF+RNN [13]	-	39.3
HP-SPS (hybrid γ , FC 10)	51.711	39.071	ParseNet [14]	-	40.4
HP-SPS (hybrid γ , FC 15)	52.01	39.248	PixelNet [9]	51.5	41.4
HP-SPS (Resblock)	50.808	37.936	Region based [6]	49.9	32.5
HP-SPS (hybrid γ , Resblock)	51.928	39.659	IFCN [17]	57.7	45

Table 4: Comparison of our results with the baseline methods and others on Pascal context dataset

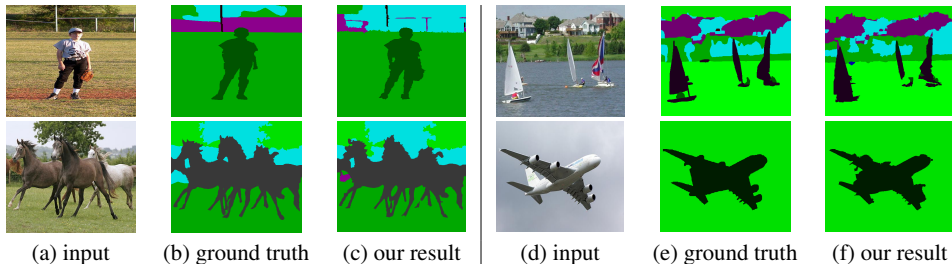


Figure 6: Some prediction example of our method on Pascal context dataset

4, Proposed method achieved better mean Acc. but slightly worse mean IU. Compared to the other similar region based method [6] applying selective search [15] to create the region, our method achieved much better performance, as shown in Table 4. Figure 6 shows several results of our method HP-SPS.

4.4 SUN-RGBD dataset

Table 5 shows the performances of various methods. In the experiment, we resized an image to 448×448 and extract 750 samples based on *Simple Linear Iterative Clustering* (SLIC) [16]. We used pre-trained caffe model of pixelNet [9] using only RGB images like other works. Our method used *fully convolution* (FC) layers after sampling for which, a 20-times increased learning rate was used. The method of Lin *et al.* [17] calculates potential function value by *convolution neural network* (CNN) and applies mean-field approximation for semantic segmentation. In addition, it applies bilinear upsampling to score map and uses *conditional random field* (CRF) to sharpen boundaries. It shows state-of-the-art performance but it uses a very complex model than others. IFCN [17] using VGGNet [28] deals with every

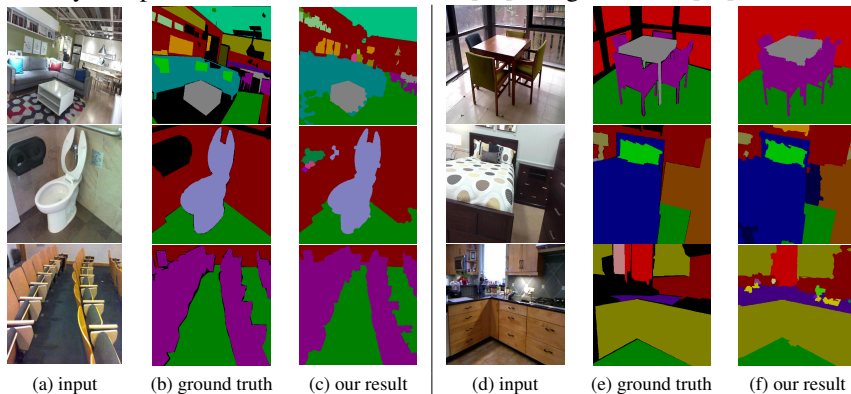


Figure 7: Example of our method on SUN-RGBD dataset

method	pixel Acc.	mean Acc.	mean IU
HP-SPS (ours)	75.67	50.06	37.96
Lin <i>et al.</i> [14]	78.40	53.40	42.30
IFCN [23]	76.90	53.46	40.74
SegNet [8]	72.63	44.76	31.84
DeepLab [9]	71.90	42.21	32.08
FCN [24]	68.18	38.41	27.39
DeconvNet [21]	66.13	33.28	22.57

Table 5: SUN-RGBD result. DeepLab, FCN and DeconvNet results are copied from [8]

	mean ACC	mean IU	time(sec)
sp(250)	51.33	38.51	2.4
sp(750)	52.01	39.25	3.1
sp(1600)	52.16	39.42	4.4

Table 6: Computational CPU time and performance on 448×448 input

feature map from pool3 for upsampling. However, the pixel Acc. of our method and IFCN are not significantly different. The proposed method is better than SegNet [8], FCN [24] and DeconvNet [21] which applying complex methods for upsampling. Also, our method achieved the better performance than DeepLab [9] which employing the atrous algorithm for wide receptive field. Figure 7 shows several exemplary results of our method on SUN-RGBD dataset.

4.5 Speed according to the number of sampling

We reduce the number of sample , and get better speed than pixelNet. With using Intel (R) Core (TM) i7-4790D CPU 4.00 GHz without GPU, PixelNet used about 30.33 seconds on 224×224 image. FCN-8s, One of the popular works, took 6.85 second and SegNet [8] marked 5.4 second on 448×448 input. We evaluated the performances using various number of superpixels for Pascal Context dataset on 448 image. Trained model with 750s superpixels was used for all experiment in Table 6.

5 Conclusion

Because most methods in semantic segmentation perform pixel-wise classification, there are many redundancy operations in both train and test. More specifically, because neighboring pixels have a high probability to be the same class, unnecessary operations are needed to estimate semantic category on all the pixels. In addition, there are also unnecessary operations in the feature extraction that requires a smaller feature maps enlarged to the size of the original image. Also, in the training phase, they do not meet the IID assumption of SGD because neighboring pixels are highly correlated. This paper comprehensively solves these problems by using superpixel-based sampling and uses hypercolumn with pyramid module for robust feature representation. Besides, since only 0.374% of the pixels are sampled, a learning problem arises, which is solved through statistical process control. We evaluated the proposed method on the Pascal Context and SUN-RGBD dataset and compared the performance with similar methodologies. The proposed method shows equal or better performance and is more efficient than the compared methods.

6 Acknowledgement

The research was supported by the Green Car development project through the Korean MTIE (10063267) and ICT R&D program of MSIP/IITP (2017-0-00306).

References

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010.
- [2] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.
- [3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [4] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta Ramanan, et al. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *arXiv preprint arXiv:1702.06506*, 2017.
- [5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
- [6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Region-based semantic segmentation with end-to-end training. In *European Conference on Computer Vision*, pages 381–397. Springer, 2016.
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.
- [8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.
- [9] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.
- [10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.
- [11] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [13] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, volume 39. Springer Science & Business Media, 2009.

- [14] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.
- [15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.
- [16] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.
- [17] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016.
- [18] Wei Liu, Andrew Rabinovich, and Alexander Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.
- [19] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feed-forward semantic segmentation with zoom-out features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2015.
- [20] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.
- [21] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
- [22] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.
- [23] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.
- [24] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.
- [25] Walter Andrew Shewhart. *Economic control of quality of manufactured product*. ASQ Quality Press, 1931.
- [26] Bing Shuai, Ting Liu, and Gang Wang. Improving fully convolution network for semantic segmentation. *arXiv preprint arXiv:1611.08986*, 2016.

- [27] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. *Computer Vision—ECCV 2012*, pages 746–760, 2012.
- [28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [29] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.
- [30] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [31] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.
- [32] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1625–1632, 2013.
- [33] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.
- [34] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *BMVC*, 2016.
- [35] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.