2018 IEEE International Conference on Systems, Man, and Cybernetics

Convolutional Networks with Bracket-style Decoder for Semantic Scene Segmentation

Cam-Hao Hua, Thien Huynh-The, Sungyoung Lee
Department of Computer Science and Engineering
Kyung Hee University, Gyeonggi-do, 446-701, Korea
Email: {hao.hua,thienht,sylee}@oslab.khu.ac.kr

***Abstract*—To build up a state-of-the-art semantic scene segmentation model, a balanced combination between coarsely and finely contextual details is required for eliminating class-wise ambiguities and reaching high accuracy of pixel-wise labeling, respectively. Accordingly, with deep learning integration, prior works have achieved impressive performance in general, but found difficulties in correctly labeling medium to small objects. For the purpose of overcoming such issue, this paper proposes a deep convolutional network with bracket-style decoder, namely B-Net, to leverage the utilization of features learned at middle layers in the backbone networks (encoder) for constructing a final prediction map of densely enhanced semantic information. In particular, every feature map of interest combines with its adjacent version of higher spatial resolution through lateral connection modules to produce finer outputs that repeat such routine round-by-round until retrieving the finest-resolution map for dense prediction. Consequently, benchmarking results on CamVid dataset showed the effectiveness of the proposed method with mean class-wise accuracy, pixel-wise accuracy, and mean union intersection of 76.2%, 87.1%, and 66.4%, respectively.**

***Index Terms*—Convolutional Neural Networks, CNN, semantic image segmentation, pixel-wise labeling, bracket-style decoder.**

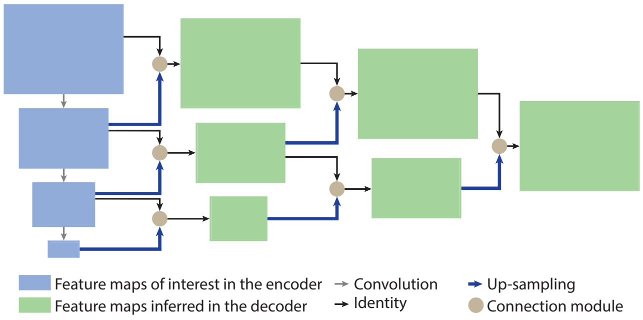
Feature maps of interest in the encoder
Feature maps inferred in the decoder
Convolution
Identity
Up-sampling
Connection module

Fig. 1. Conceptual diagram of the proposed B-Net: Top-down blue rectangles represent typically fine-to-coarse feature maps (with spatial size decreasing gradually) in the backbone CNN (encoder). For decoding, every of those feature maps, except for the highest-resolution one, is up-sampled and then combined with its adjacent higher-resolution version by connection modules (copper circles) to infer finer output feature maps (green rectangles) that repeat the same procedure round-by-round until retrieving one final prediction map of highest resolution (the rightmost green rectangle).

I. Introduction

Since gaining huge advantages from the recently fast-paced development of computing resources, e.g. graphical processing units (GPUs), along with the exponential growth of big visual data, deep Convolutional Neural Network (CNN) has emerged with breakthrough performance in various areas of computer vision. Accordingly, there has been an enormous amount of research works which attempts to utilize CNNs for tackling semantic segmentation, a problem of labeling every pixel in an input image given a predefined number of object classes for complete scene understanding. Obviously, such high-level task can be widely deployed in many modern applications like autonomous driving, augmented reality, computational photography, etc [1] and hence, becomes one of the top trending research fields in machine vision.

It can be realized that most of the state-of-the-art semantic segmentation methods utilize backbone CNN architectures (which are originally designed for dealing with large-scale image classification problem, e.g. AlexNet [2], VGGNet [3], GoogleNet [4], ResNet [5]) as fine-to-coarse feature encoders from which diverse manipulations are referred to generate pixel-wise labeled maps having same spatial size as that of the inputs. Concretely, in the aforementioned deep models, outputs of late layers get significantly deeper while their spatial dimension reduces drastically compared to those of earlier layers, e.g. it is 32x downscale from the raw input to the resulting map of the 5^{th} convolutional layer in the VGG16-Net [3]. Subsequently, for effectively transforming the coarse feature maps containing global context to the final image filled with intensities of label values without any ambiguities and distortions, it is critical to define a scheme be able to balancedly integrate various contextual information learned from different layers of the backbone CNN into a proper up-sampling strategy. With such observations on decoding manner, existing deep learning based research works can be categorized into two major groups: *asymmetric-* and *symmetric-structured* networks. Generally, the former represents semantic segmentation models that use distinguished up-sampling plan and/or various layer-wise ensemble styles, while the latter reverses the architecture of the feed-forward network with supplemental manipulations to infer desirable prediction maps.

Some typical research works belonging to the *asymmetric-structured* group are highlighted as follows. Long *et al.* [6] made a debut for introducing an end-to-end trainable fully convolutional network (FCN) that makes feature maps of interest, which are yielded by different layers in the backbone CNN, same size to each other by implementing learnable transpose convolutions (fractionally strided convolutions) on smaller-sized ones and then fuses them for the prediction at

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00506

pixel level. However, FCN still remains two essential drawbacks: (i) unbalanced aggregation between recovered local and global information because their partial skip connection scheme does not take into account features of early layers and (ii) taking up-sampling of large stride, e.g. 8x, 16x or 32x, in a single operation clearly leads to heavy ambiguities in the final prediction map. From the literature, corresponding solutions are divided into two major ways: one tries to deploy additional network streams for collecting globally semantic information to eliminate noises and ambiguities in the main stream of pixel-wise labeling such as ParseNet [7], HistNet [8], HolisticNet [9], and another puts efforts on incorporating multi-scale features in a spatial pyramid pooling manner like DilatedNet [10], DeepLab [11], [12], and PSPNet [13]. It should be noted that in the second solution branch, all the mentioned citations introduced dilated (atrous) convolution as a more effectively computational tool compared to the vanilla convolution.

In the group of *symmetric-structured* network, the deep segmentation model is constructed under the form of encoder-decoder. Concretely, the encoder is a pre-trained CNN, which downscales the spatial dimension of feature maps for the abstract representation of visual details. Meanwhile, the decoder is layer-wise reverse version of the encoder, which up-samples feature maps extracted previously for the generation of densely labeled content. On the one hand, SegNet [14], [15] uniquely up-samples feature maps at the decoder by utilizing corresponding max pooling indices stored during the feed-forwarding stage, which benefits from end-to-end maintenance of high response features and fewer parameters but faces the risk of losing neighborhood information. On the other hand, to maximize capability of conserving contextual information and precision of localization during up-sampling at the decoding phase, other works tried to up-sample the coarsest feature map of which the up-sampled result is then directly combined with the corresponding one (in terms of spatial size) in the encoder to produce an output that continuously takes the same procedure until obtaining ultimate feature map having spatial dimension same as the input's. The combination technique can be either depth-wise concatenation as in U-Net [16], FC-DenseNet [17] or specific connection styles including refinement modules in SharpMask [18], RefineNet [19], LRN [20] and lateral connection module in FPN [21]. Furthermore, readers may refer to [1] for an exhaustive review of deep learning based semantic segmentation models.

Briefly, with respect to the up-sampling strategy in existing works, only the contemporarily coarsest feature map is up-sampled with multi-scale scheme in the pyramid mode [10]–[13] or at each staircase in the ladder-like manner [16]–[21] for further operations. As a result, features extracted from middle layers of the encoding networks are not exploited effectively [19] in these models since they just perform a single role of linking with up-sampled versions of the lower-resolution maps for ambiguities exclusion. This motivates us to propose a deep CNN with Bracket-style decoder, namely B-Net, which is able to leverage the utilization of features learned at middle layers of the encoder for reasonably boosting the accuracy of semantic segmentation. As abstractly illustrated in Fig.1, except for the highest- and lowest-resolution ones, every feature map of interest simultaneously plays both roles of (i) the one that is up-sampled and (ii) the one that merges with the up-sampled map having same resolution on the way of retrieving dense prediction map. In other words, every feature map is paired with its closest higher-resolution version to yield finer ones that continuously do the same process until forming the final finest-resolution feature map, in which semantic information is much richer than that of prior networks. Basically, the proposed approach can be classified into the *asymmetric-structured* group due to the bracket-shaped structure. For evaluation, our B-Net is benchmarked with CamVid dataset [22], [23] of which performance in terms of mean union intersection, class and pixel average accuracy is competitive with state-of-the-art techniques.

II. Methodology

This section gives a thorough description of the proposed B-Net architecture, which is demonstrated in Fig.2.

A. Bracket-style Decoder

In the proposed architecture, we utilize VGG16-Net as the backbone CNN for extracting deep features from the inputs. Specifically, the bracket-style decoder takes into account 6 feature maps from the VGG16-Net, i.e. outputs of the 1^{st} convolution layer, the 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} pooling layers of which spatial dimensions are same as, $1/2$, $1/4$, $1/8$, $1/16$ and $1/32$ that of the input images, respectively. Obviously, the feature maps are chosen such that the spatial dimension is halved gradually. For convenience, we correspondingly name them *conv1_1*, *pool1*, *pool2*, *pool3*, *pool4* and *pool5*. Then, every feature map of interest (except for the one having spatial size identical to that of the input images) combines with the adjacent higher-resolution one via a predefined lateral connection module for generating finer-resolution feature maps of the latter's dimensions. In other words, each pair of feature maps results in one output having volume size same as that of the higher-resolution input. Note that a round is made when n available feature maps are densely matched to produce $n-1$ outputs. Consequently, the number of semantic feature maps decreases by one while their average spatial resolution increases round-by-round until there is only one dense feature map left with favorably semantic information for the goal of pixel-wise labeling. For instance, as shown in Fig.2, at round 1, 6 above-mentioned feature maps are paired with each other (*pool5* vs. *pool4*, *pool4* vs. *pool3*, ..., *pool1* vs. *conv1_1*) under the proposed style to generate 5 new feature maps, which continuously get through the same routine to produce other 4 finer feature maps at round 2 and so on. In our B-Net, the final feature map is retrieved after 5 rounds. Afterwards, we use a 1×1 convolutional layer with added biases to reduce the depth of that ultimate feature volume to the number of predictive object classes. As a result, feature map of unnormalized prediction probabilities corresponding

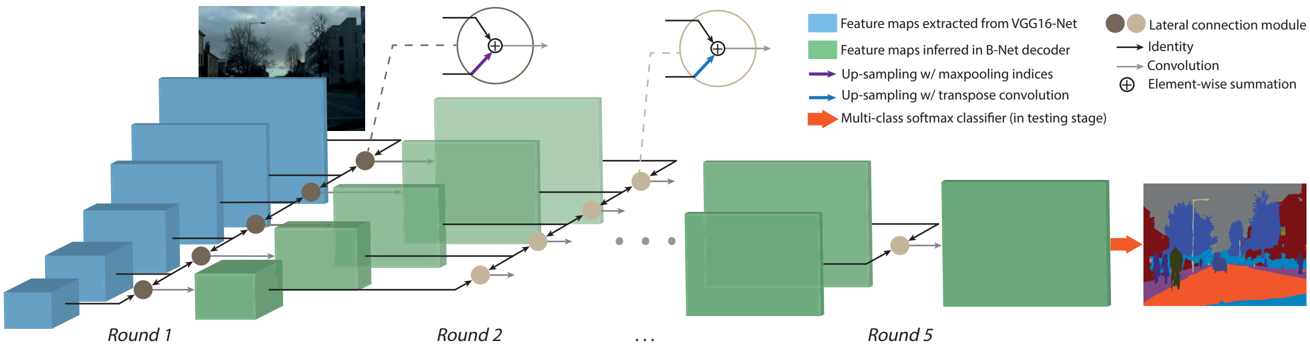
Feature maps extracted from VGG16-Net
Feature maps inferred in B-Net decoder
Up-sampling w/ maxpooling indices
Up-sampling w/ transpose convolution
Multi-class softmax classifier (in testing stage)
Lateral connection module
Identity
Convolution
Element-wise summation
Round 1
Round 2
...
Round 5

Fig. 2. Architecture of the proposed B-Net. Given an input image, fine-to-coarse feature maps of interest (in blue cuboids) are extracted from the backbone VGG16-Net [3] and then densely combined (by modules having the up-sampling scheme of using max pooling indices [14]) to produce preliminarily semantic features (in green cuboids). Next, those ones are iteratively passed through the same procedure (but combined by modules having another up-sampling plan called transpose convolution [6]) until achieving final prediction map, of which every pixel is assigned an object class by the softmax classifier. Since every inferred feature map fuses with its nearest higher-resolution map at each round and the total number of feature maps decreases by one round-by-round, such process is named *bracket-style* decoder.

to the labels is obtained. Finally, multi-class softmax classifier (in testing stage) assigns each pixel of the retrieved prediction map a specific label. Fundamentally, there are two apparent advantages of using the bracket structure: (i) missing or ambiguous details are suppressed significantly since every up-sampled feature map is always refined by unifying with the equivalent one in terms of volume size in the encoder and (ii) semantically contextual information is densely enhanced in the final feature map because such up-sampling plus dense mixture strategy is applied for all fine-to-coarse feature maps in all rounds.

B. Lateral Connection Module

Regarding the combination manner between two neighboring feature maps of finer and coarser resolution as displayed in Fig.2, we adopt the lateral connection style in [21] because of its simplicity yet reasonableness for the B-Net architecture. However, there are several differences from the original concept. In particular, instead of initially passing through a 1×1 convolutional layer for reducing the channel dimension, the finer-resolution input is directly added to up-sampled version of the coarser-resolution one in element-wise manner because our up-sampling scheme is already designed to make volume size of the latter suit that of the former. Subsequently, the total feature map goes through a 3×3 convolution layer for the purpose of minimizing unexpected artifacts during up-sampling. As mentioned before, the finalized output, which contains essential responses combined from the two inputs, is continuously merged with another neighbor at next round under similar process until reaching the final feature map having same spatial dimension as that of the input image.

C. Up-sampling Strategy

With respect to the step of up-sampling in the aforementioned lateral connection module, we utilize two popular strategies, i.e. up-sampling using max pooling indices [14] and transpose convolution [6], to maximize the performance of B-Net. To our best knowledge, this is the first work employing two types of up-sampling technique within a semantic segmentation network. Apparently, except for *conv1_1*, the remaining 5 feature maps of interest are outputs of pooling layers in VGG16-Net, therefore, using the unpooling technique proposed by [14] for up-sampling is clearly an optimal option at the first round of the proposed decoder. In details, values in the *pool1*, *pool2*, *pool3*, *pool4* and *pool5* with previously stored locations are directly transferred to the corresponding positions in the up-sampled map, which is initially filled with zeros. Therefore, it is undeniable that such scheme totally preserves the contextual information of highest responses and accordingly improves the capability of collecting more meaningful knowledge for the densely labeling objective. Additionally, a 1×1 convolutional layer is applied to reduce the channel dimension of the up-sampled feature maps to be fit with that of the corresponding higher-resolution one. As of the second round, 3×3 trainable transpose convolution layers, which firstly construct a map having spatial resolution as desired and then carry out convolution with learnable weights, are used for up-sampling the coarser map because the former max pooling indices cannot be deployed anymore. Also, this fractionally strided convolution is designed to make all dimensions of the lower- and higher-resolution feature map identical. In summary, compared to using the transpose convolution in all rounds, having the first round of using the unpooling technique gives us two merits: (i) the total number of trainable parameters in the proposed architecture decreases by 3.5 millions and (ii) better performance in terms of average global accuracy, average class accuracy and mean intersection of union (mIoU) with higher rates of $0.5-1.7\%$ on the CamVid dataset as reported in Table I. Note that the evaluation metrics and model settings used for this comparison are enumerated in section III.B and C, respectively.

III. Experiments

In this section, we intensively experiment the proposed B-Net on the CamVid dataset for the purpose of benchmarking its

Up-sampling Strategy	only transpose conv.	unpooling + transpose conv.
Number of parameters	29.42M	**25.92M**
Average class accuracy (%)	72.8	**73.8**
Average global accuracy (%)	87.3	**87.8**
mIoU (%)	63	**64.7**

TABLE I
PERFORMANCE OF TWO DIFFERENT UP-SAMPLING STRATEGIES IN B-NET ON CAMVID DATASET [22], [23]. BOLDFACE NUMBERS SHOW BETTER PERFORMANCE.

effectiveness. Specifically, we firstly have a short introduction of the benchmark dataset and evaluation metrics, then enumerate configuration details for training, and finally analyze on-hand experimental results.

A. Benchmark dataset: CamVid

CamVid [22], [23] is the abbreviation of Cambridge-driving Labeled Video Database which originally records different road scenes in 10 minutes by a dashboard camera to simulate viewpoints from a driving automobile. Then, totally 701 resulted 720×960 video frames are per-pixel annotated with 32 semantic labels, which makes this dataset a tough challenge for the semantic segmentation problem. Same as previous works, we use the split of 367 training, 101 validation and 233 testing images with 12 finalized labels consisting of building, tree, sky, car, sign-symbol, road, pedestrian, fence, column-pole, side-walk, bicyclist and the unlabeled (background) to evaluate the proposed method. Note that all the images are down-sampled to the size of 360×480 beforehand.

B. Evaluation Metrics

Same as prior works, we use three major criteria, i.e. mean global accuracy (mGA) for pixel-wise based evaluation, mean class accuracy (mCA) and mean intersection of union (mIoU) for class-wise oriented evaluation. As stated in [1], the class-wise based measures, i.e. mCA and mIoU, represent the effectiveness of a per-pixel labeling algorithm more accurately than the pixel-wise based metric like mGA. Note that providing an in-depth analysis of these benchmarks is beyond the scope of this work.

C. Training Configurations

In the training stage, each batch of 3 images is sequentially fed into the proposed B-Net. Remarkably, no pre-processing techniques are embedded in our work. Besides, in the pre-trained backbone VGG16-Net, we add trainable batch normalization [24] after each convolutional layer with added biases and before the non-linear activation ReLU [2]. The similar order of operation, i.e. convolution, biases addition, batch normalization and ReLU, is also applied to the convolution layer after the element-wise summation step in the lateral connection modules. Regarding loss calculation strategy for dealing with the issue of unbalanced amount of pixels in each label, this work follows the procedure of [14] in which weighted cross-entropy loss is utilized with corresponding weights inferred using median frequency balancing [25]. Accordingly, Adam optimizer [26] is applied to our model with learning rate of 0.001, decay rate for moving average of gradient's first and second moment of 0.9 and 0.999, respectively. Finally, the proposed B-Net is implemented using Tensorflow [27] framework with one Titan X GPU and 32GB RAM.

Notably, with no proper regularization scheme, the overfitting issue, which causes poor performance on unseen data, is still on the horizon. Therefore, by empirical experiments on the CamVid dataset, we have found that the proposed B-Net becomes much robust against the overfitting problem by following regularization setting: imposing dropout [28] with dropping rate of 0.5 to (i) the *pool3*, *pool4*, and *pool5* of which corresponding outputs replace them if they play the role of lower-resolution input of lateral connection modules at round 1 and (ii) all the coarser-resolution inputs of lateral connection modules at round 2. Indeed, utilizing dropout at early layers of the encoder and/or later rounds of the bracket-style decoder is not a good choice since it can destroy generic features like edges, blobs which are important details in the semantic segmentation problem and/or causes high possibility of over-regularization, respectively. Obviously shown in Table II, compared to the default setting, taking into account the regularization scheme can improve mCA by 2.4% and mIoU by 1.7% despite achieving mGA with 0.7% lower rate. However, as noted previously, the class-wise measures are higher-fidelity metrics for the evaluation of such per-pixel labeling problem.

D. Experimental Results and Discussion

For an explicit evaluation and fair comparison (since different works use dissimilar metrics), along with summarized mCA, mGA and mIoU, we report two separated groups of experimental results: one contains class-wise accuracies in Table II and another includes class-wise union intersection values of each object class in Table III.

In terms of mCA and mGA, despite not being the most state-of-the-art, the proposed B-Net is still competitive in the leading portion with mCA of 76.2% and mGA of 87.1%. About mIoU, our B-Net achieves 66.4% which outperforms 1.1% – 16.2% over the compared works.

With respect to class-wise performance, our optimal model, i.e. B-Net with dropout, achieves best per-class accuracies in 5 out of 11 labels as reported in Table II. Especially, high margin is reached at Tree (93.7% vs. $\leq$ 88.1%) and Road (98.6% vs. $\leq$ 98%). Moreover, about the measure of per-class intersection of union in Table III, the proposed approach gives a superior performance over the rest in Sky, Pedestrian, Fence, and Column-pole with higher rates of more than 2.9%, 4.5%, 6.8% and 5.9%, respectively. Clearly, the fact that repeated connections between all contemporary feature maps in bracket manner has not only helped the deep model label middle to small details like Pedestrian, Fence, and Column-pose more accurately but also enhanced the ability of refining allocation of large objects like Tree, Sky, Road, and Car. On the other hand, it should be noted that the proposed technique faces heavy failure in labeling pixels belonging to the Sidewalk

Approach	Building	Tree	Sky	Car	Sign-symbol	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mCA	mGA	mIoU
SegNet [14]	88	87.3	92.3	80	29.5	97.6	57.2	49.4	27.8	84.8	30.7	65.9	88.6	50.2
Bayesian SegNet [15]	80.4	85.5	90.1	86.4	**67.9**	93.8	73.8	64.5	**50.8**	91.7	54.6	76.3	86.9	63.1
ReSeg [29]	86.8	84.7	93	87.3	48.6	98	63.3	20.9	35.6	87.3	43.5	68.1	**88.7**	58.8
LRN [20]	89.8	88.1	78.5	86.3	61.2	96.8	**82.1**	59	45.4	**92.6**	**69.7**	**77.2**	-	61.7
B-Net (no dropout)	**90.3**	88.1	**96**	81.4	51.3	98.2	80	52.1	42.1	72.6	59.2	73.8	87.8	64.7
B-Net (with dropout)	86.6	**93.7**	95.6	**87.8**	47.7	**98.6**	**82.1**	**64.9**	46.3	72.4	62.6	76.2	87.1	**66.4**

TABLE II

Experimental results (%) of **per-class accuracy**, mean per-class accuracy (mCA), mean global accuarcy (mGA) and mean intersection of union (mIoU) on CamVid [22], [23] dataset. Boldface numbers represent the best performance at each criterion.

Approach	Building	Tree	Sky	Car	Sign-symbol	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mIoU
DeepLab-LFOV [30]	81.5	74.6	89	82.2	42.3	**92.2**	48.4	27.2	14.3	75.4	50.1	61.6
Dilation8 [10]	**82.6**	**76.2**	89.9	**84**	**46.9**	**92.2**	56.3	35.8	23.4	75.3	**55.5**	65.3
LRN [20]	78.6	73.6	76.4	75.2	40.1	91.7	43.5	41	30.4	**80.1**	46.5	61.6
B-Net (no dropout)	81.5	75	92.3	78.8	41.6	89	59.2	41.1	33.2	67	53	64.7
B-Net (with dropout)	81.4	75.3	**92.8**	82.5	42.8	89.2	**60.8**	**47.8**	**36.3**	66.4	54.8	**66.4**

TABLE III

Experimental results (%) of **per-class intersection of union** and mean intersection of union (mIoU) on CamVid [22], [23] dataset. Boldface numbers represent the best performance at each criterion.

class, with only 72.4% compared to the state-of-the-art of 92.6% in terms of class-level accuracy and 66.4% vs. 80.1% in terms of class-wise intersection of union. Therefore, intensive investigation into such issue is the main focus of our future work.

Furthermore, some typical qualitative results compared with those of SegNet [14] are illustrated in Fig. 3. It can be observed that our approach outperforms the competitor in labeling medium to small details like Pedestrian, Bicyclist, Fence, Sign-symbol, and Column-pole for better scene understanding. Also, with respect to large object labeling, the proposed B-Net can reduce misclassification between the truck, which belongs to Car class, and the Building significantly as shown in the second row of Fig.3.

IV. Conclusion

This paper has presented an end-to-end trainable deep CNN model with bracket-style decoder, namely B-Net, for semantic image segmentation. By a coarse-to-fine strategy that every feature map of interest extracted from a backbone network combines with its nearest higher-resolution version via predefined lateral connection modules, of which finer outputs continuously do the same routine at each round until only one finest-resolution feature map is left, semantic information is densely refined and enhanced in bracket manner for better pixel-wise labeling. Besides, it is worth noting that our work utilized two different up-sampling scheme, i.e. unpooling technique and transpose convolution, to improve the performance. Accordingly, experimental results have indicated that the proposed B-Net is competitive with state-of-the-art techniques, especially on mean intersection of union metric. Furthermore, to resolve existing drawbacks and leverage the proposed approach to new heights, our future work focuses on investigating the failures and exhaustively exploiting the potentiality of the bracket-style decoder with more variants and alternatively modern backbone networks.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP (Institute for Information & communications Technology Promotion). This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00655).

References

[1] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, "A review on deep learning techniques applied to semantic segmentation," *CoRR*, vol. abs/1704.06857, 2017.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[7] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *CoRR*, vol. abs/1506.04579, 2015.

[8] W. Zhe, L. Hongsheng, O. Wanli, and W. Xiaogang, "Learnable histogram: Statistical context features for deep neural networks," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 246–262.

[9] H. Hu, Z. Deng, G. Zhou, F. Sha, and G. Mori, "Recalling holistic information for semantic segmentation," *CoRR*, vol. abs/1611.08061, 2016.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.

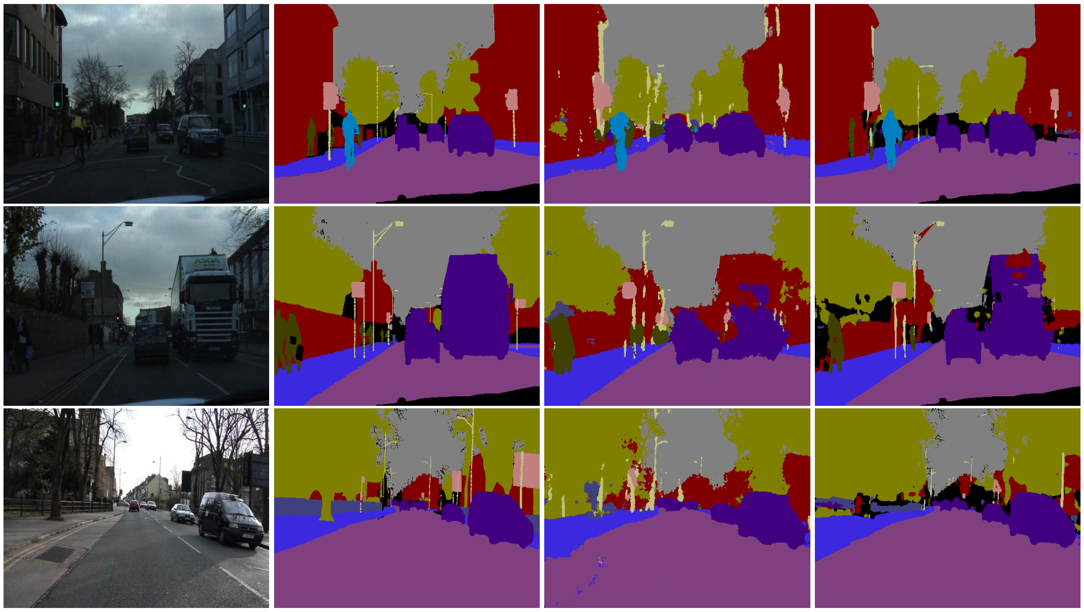

Fig. 3. Several visual results on the CamVid [22], [23] dataset. Displayed contents include raw images, corresponding ground-truths, results of SegNet [14] and ours. Compared to SegNet [14], our approach outperforms in labeling medium to small objects like car, pedestrian, bicyclist, fence, sign-symbol, column-pole. Besides, as for large objects, the misclassification between the truck (labeled in purple) and the building (labeled in red) as shown in the 2^{nd} row reduces significantly.

[11] 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," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2018.

[12] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6230–6239.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.

[15] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *CoRR*, vol. abs/1511.02680, 2015.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[17] S. Jgou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1175–1183.

[18] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 75–91.

[19] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5168–5177.

[20] M. A. Islam, S. Naha, M. Rochan, N. D. B. Bruce, and Y. Wang, "Label refinement network for coarse-to-fine semantic segmentation," *CoRR*, vol. abs/1703.00551, 2017.

[21] T. Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936–944.

[22] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV (1)*, 2008, pp. 44–57.

[23] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[25] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. IEEE Computer Society, 2015, pp. 2650–2658.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[29] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2016, pp. 426–433.

[30] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.