Small Object Sensitive Segmentation of Urban Street Scene With Spatial Adjacency Between Object Classes

Dazhou Guo, Ligeng Zhu, Yuhang Lu, Hongkai Yu, and Song Wang, *Senior Member, IEEE*

***Abstract*— Recent advancements in deep learning have shown an exciting promise in the urban street scene segmentation. However, many objects, such as poles and sign symbols, are relatively small, and they usually cannot be accurately segmented, since the larger objects usually contribute more to the segmentation loss. In this paper, we propose a new boundary-based metric that measures the level of spatial adjacency between each pair of object classes and find that this metric is robust against object size-induced biases. We develop a new method to enforce this metric into the segmentation loss. We propose a network, which starts with a segmentation network, followed by a new encoder to compute the proposed boundary-based metric, and then trains this network in an end-to-end fashion. In deployment, we only use the trained segmentation network, without the encoder, to segment new unseen images. Experimentally, we evaluate the proposed method using CamVid and CityScapes data sets and achieve a favorable overall performance improvement and a substantial improvement in segmenting small objects.**

***Index Terms*— Small objects segmentation, spatial adjacency, semantic segmentation, urban street scene.**

I. Introduction

SEMANTIC segmentation aims to assign a categorical label to each pixel in an image [1]–[3], and it plays an important role in image understanding [4]–[6]. The recent success of deep convolutional neural networks (CNNs) [7]–[13] has made remarkable progress in pixel-level semantic segmentation tasks [14]–[18]. But the segmentation of small objects is usually inaccurate [19], as small objects usually contribute less to the segmentation loss. For example, as shown in Fig. 1(c), sign symbols and poles only take a small fraction of the overall urban street scene, and they could be overlooked in the segmentation. However, accurately segmenting small objects is of great importance in many applications, such as autonomous driving, where driving safety and precise navigation are dependent on the segmentation and recognition of small-sized poles and traffic signs [20]–[25]. In this paper, we take urban street scene segmentation as a study case and develop new CNN-based semantic segmentation methods that can better handle small-sized classes.

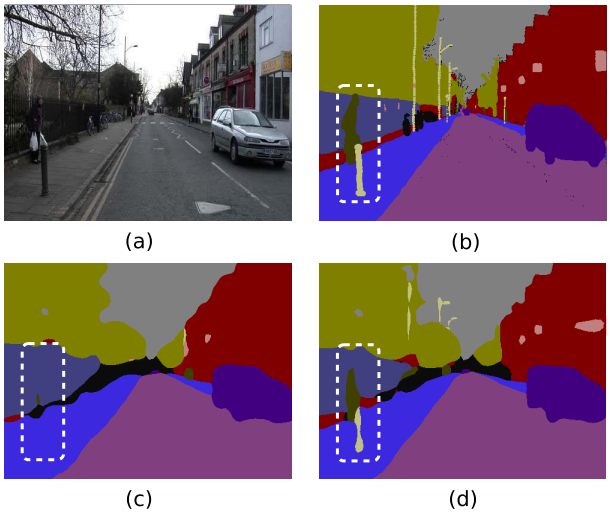


Fig. 1. An illustration of the effectiveness of the proposed method, which improves the segmentation results of FCN-8s [14] by using proposed ISBEncoder, especially on small objects. (a) Input. (b) Ground Truth. (c) Without ISBEncoder. (d) With ISBEncoder.

One common strategy towards improving the segmentation accuracy of small objects is to increase the scale of input images, to enhance the resolution of small objects, or to produce high-resolution feature maps [20]–[22], [26], [27]. This strategy is usually implemented in CNNs by reducing object size induced biases, and training the network to generate multi-scale representation which enhances high-level small-scale features with multiple low-level feature layers. However, those approaches require data augmentation or increase of the feature dimension. Simply increasing the scale of input images often results in heavy time consumption for both training and testing [20]. The multi-scale representation constructed by the low-level features works as a black-box and cannot guarantee the constructed features are interpretable [22].

Manuscript received June 5, 2018; revised October 28, 2018; accepted December 10, 2018. Date of publication December 19, 2018; date of current version March 21, 2019. This work was supported in part by NSF under Grant 1658987 and in part by NSFC under Grant 61672376 and Grant U1803264. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dacheng Tao. *(Dazhou Guo and Ligeng Zhu contributed equally to this work.) (Corresponding author: Song Wang.)*

D. Guo and Y. Lu are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29201 USA (e-mail: guo22@email.sc.edu; yuhang@email.sc.edu).

L. Zhu is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02319 USA (e-mail: ligeng@mit.edu).

H. Yu is with the Department of Computer Science, University of Texas–Rio Grande Valley, Edinburg, TX 78539 USA (e-mail: hongkai.yu@utrgv.edu).

S. Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29201 USA, and also with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: songwang@cec.sc.edu).

Digital Object Identifier 10.1109/TIP.2018.2888701

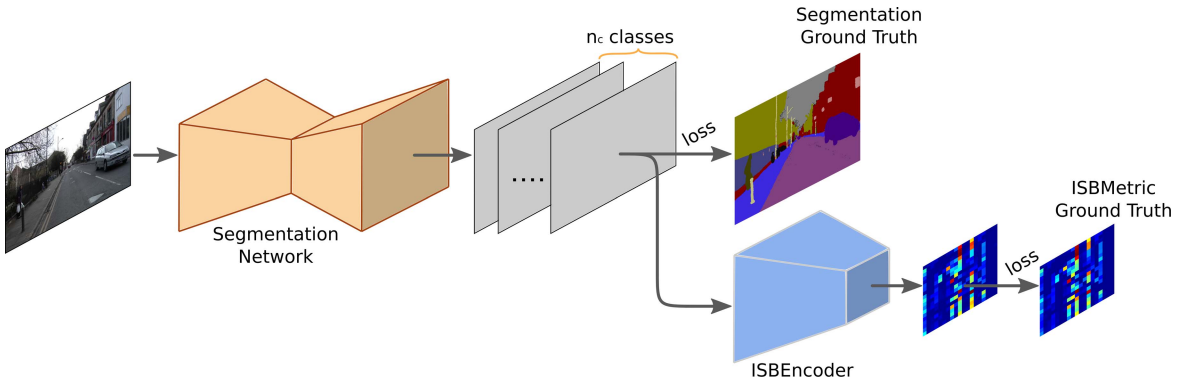


Fig. 2. An overview of the proposed pipeline – segmentation network with the ISBEncoder. The ISBEncoder is capable of encoding the level of spatial adjacency between object-class pairs into the segmentation network.

Post-processing is another strategy towards improving the accuracy of small object segmentation [17], [28]. As post-processing is not integrated into the segmentation network, the network cannot update its weights according to the post processed results in the training phase [29].

In this paper, we propose an Inter-class Shared Boundary based Metric (ISBMetric) to quantify the level of adjacency between each pair of object classes. Specifically, ISBMetric calculates the proportional length of the shared boundaries. For example, ISBMetric between object classes **A** and **B** is the proportion between the length of their shared boundaries and the perimeter of **A** or **B**. In addition, by quantifying this ratio based level of spatial adjacency, the proposed ISBMetric is robust against object size induced biases, such that small objects can contribute more to the segmentation loss. We demonstrate that the enforcement of the ISBMetric can help improve the segmentation accuracy of small objects. We propose an ISBMetric based encoder (ISBEncoder) for the purpose. In particular, the proposed ISBEncoder takes the prediction from the segmentation networks as the input and its output is guided by the ISBMetric matrix calculated using segmentation ground truth. In deployment, we only use the trained segmentation network, without the ISBEncoder, to segment new unseen images, such that no extra time or cost is added to the segmentation network. The proposed pipeline – segmentation network with ISBEncoder – is illustrated in Fig. 2 and it can be trained in an end-to-end fashion.

We evaluate the proposed method using two urban street scene datasets: CamVid [30] and CityScapes [31], and achieve improved results, especially for the small object classes. The effectiveness of the proposed ISBMetric and ISBEncoder is tested and evaluated by combining to many state-of-the-art segmentation networks. To sum up, the main contributions of this paper are: **1)** We propose a new ISBMetric to measure the level of spatial adjacency between each pair of object classes. The ISBMetric is robust against the object size induced biases, such that small object classes can contribute more to the overall loss. **2)** We propose a new ISBEncoder to enforce the ISBMetric in the segmentation of urban street scene. The proposed ISBEncoder can be easily combined to many state-of-the-art segmentation networks. **3)** We achieve substantially improved segmentation accuracy of small object classes and improved segmentation accuracy of large object classes using the proposed method without adding extra time or cost during the deployment.

II. Related Works

Segmenting small objects, e.g., poles, traffic signs, and pedestrians, in urban street scene is of great importance in intelligent vehicles [32]–[34]. Various methods [17], [22] have been proposed to address this challenging task. To improve the small object segmentation accuracy in urban street scene, one common strategy is to use multi-scale input images, such that the resolution of small objects in the image is enhanced. This strategy is usually implemented in CNNs, such that the network can learn both high-level large-scale and high-level small-scale features. Thus, it can reduce the object size induced biases. However, the training of the network requires data augmentation, not to mention the heavy time consumption in both training and testing.

Another strategy to improve the segmentation accuracy of small objects in urban street scene is to use context based post-processing, e.g., Markov Random Field (MRF) and fully connected Conditional Random Field (CRF). Most fully convolutional network (FCN) based methods exploit context information by constructing MRFs or CRFs as the post-processing stage to the overall network [17]. However, the post-processing stage is disconnected from the training of the network [17] and the network cannot adapt its weights based on the post-processing outputs [29].

Also related to this paper are several scene parsing approaches that partially address the small object segmentation problem. In [11], a Siamese network is proposed to learn the global context similarity between images. It tries to improve the segmentation performance of the small object classes by re-weighting the classes based on how often the classes appear in the dataset. In [13], a FoveaNet with CRFs is proposed to correct the distortion caused by the camera perspective projection. It tries to improve segmentation performance of the objects crowding around the vanishing point.

Different from small object segmentation, small object detection only provides a bounding box to the target. In [32], a multi-stage feature is proposed for classification by bridging the connections between large objects and small objects with skip layers to increase the discrimination of small objects.

In [33], multi-stage features are used to integrate global shape information with local distinctive information to learn the detectors. In [34], hinge loss is introduced to the CNNs resulting in a faster and more stable convergence with better performance. In [20], network is trained to generate multi-scale representation which enhances high-level small-scale features with multiple low-level feature layers. In [22], a perceptual generative adversarial network is proposed for small object detection, by minimizing representation difference between small objects and normal size objects.

Several obstacle detection approaches [23]–[25] are proposed to detect potentially hazardous objects on the road. In [23], a stereo vision based method is proposed to detect obstacles on the road. In [25], a multi-stage MergeNet is proposed to detect obstacles, in which each stage tackles a different task. The output feature maps from each stage are later merged and used for the obstacle detection. Several RGB-D semantic segmentation approaches are also proposed. In [35], a context-aware receptive field (CaRF) is proposed to improve the segmentation performance. The CaRF provides better control over the relevant contextual information of the learned features, leading to a more focused domain which is easier to learn. In [36], a feature transformation network is proposed to improve the segmentation performance and bridge the convolutional networks and deconvolutional networks by discovering common features between RGB images and depth maps. However, our work on small object segmentation is different from their works in task and usage, not to mention that the training of the models usually requires additional depth map/stereo images.

III. Methodology

A. Overview

The proposed pipeline, as shown in Fig. 2, consists of two components: the default segmentation network and the proposed ISBEncoder. Specifically, the default segmentation network can be any network used for semantic segmentation. The proposed ISBEncoder takes the prediction from the segmentation network as input, and its output is guided by the ISBMetric calculated using segmentation ground truth. The overall pipeline can be trained in an end-to-end fashion.

B. Inter-Class Shared Boundary Metric (ISBMetric)

To evaluate the level of spatial adjacency between each pair of object classes, we define the ISBMetric m_{isb} as a $n_c \times n_c$ matrix with n_c being the number of object classes, i.e., segmentation labels. This metric is computed from the segmentation map s, where $s(x, y) \in \{1, 2, \cdots, n_c\}$ is the segmentation-class label at pixel (x, y).

The value of $m_{isb}(i, j)$ is the ratio of the length of the boundary shared by the i^{th} and the j^{th} object classes to the i^{th} object class' perimeter. Let l_i denote the i^{th} object class' perimeter, and let l_{ij} denote the length of the shared boundary between the i^{th} and j^{th} object classes. The (i, j)-th element in the ISBMetric is $m_{isb}(i, j) = \frac{l_{ij}}{l_i}$, while the value of (j, i)-th element in the ISBMetric is $m_{isb}(j, i) = \frac{l_{ij}}{l_j}$. The value of $m_{isb}(i, i)$ is set to 0 for $i = 1, 2, \cdots, n_c$. As the perimeters of different object classes are usually different, i.e., $l_i \neq l_j$ if $i \neq j$, the ISBMetric m_{isb} is usually asymmetric.

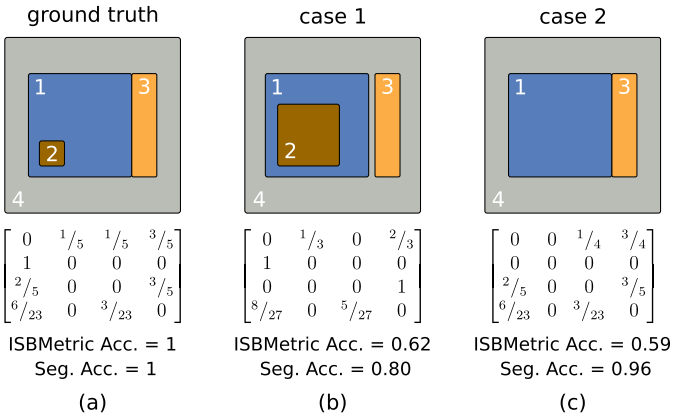


Fig. 3. An illustration of proposed ISBMetric of (a) ground-truth segmentation of an image, and (b-c) two sample segmentation results of the image. Below each segmentation are the corresponding ISBMetric matrix, ISBMetric accuracy and segmentation accuracy.

As shown by an example in Fig. 3(a), a segmentation map consists of four object classes: 1, 2, 3, 4 with rectangular outer boundary of dimensions 100×100, 25×25, 25×100, and 175×175 pixels, respectively. The ISBMetric m_{isb} is calculated as:

- The perimeter of segmented object 1 is $100 \times 4 + 25 \times 4 = 500$ (combined outer and inner boundaries). The lengths of the boundaries shared by objects 1 and 2, shared by objects 1 and 3, and shared by objects 1 and 4 are 100, 100 and 300, respectively. The first row of the m_{isb} is $[0, \frac{1}{5}, \frac{1}{5}, \frac{3}{5}]$.
- The perimeter of segmented object 2 is 100, and object 2 is fully enclosed by object 1 and is not adjacent to objects 3 and 4. The length of the boundaries shared by objects 1 and 2 is 100. The second row of the m_{isb} is $[1, 0, 0, 0]$.
- The perimeter of segmented object 3 is 250. The object 3 is not adjacent to object 2. The lengths of boundaries shared by objects 1 and 3, and shared by object 3 and 4 are 100 and 150. The third row of the m_{isb} is $[\frac{2}{5}, 0, 0, \frac{3}{5}]$.
- The perimeter of segmented object 4 is 1, 150. The object 4 is not adjacent to object 2. The lengths of boundaries shared by objects 1 and 4, and shared by objects 3 and 4 are 300 and 150. The fourth row of the m_{isb} is $[\frac{6}{23}, 0, \frac{3}{23}, 0]$.

When both the spatial adjacency between the object classes and object size are changed, as illustrated in Fig. 3(b) – right-shift the object 3 by 1 pixel and enlarge the object 2 to a size of 50×50, ISBMetric m_{isb} will be changed as follow:

- The perimeter of segmented object 1 is changed to 600. The objects 1 and 3 are no longer adjacent, and the length of the boundaries shared by them is changed to 0. The first row of the m_{isb} is changed to $[0, \frac{1}{3}, 0, \frac{2}{3}]$.

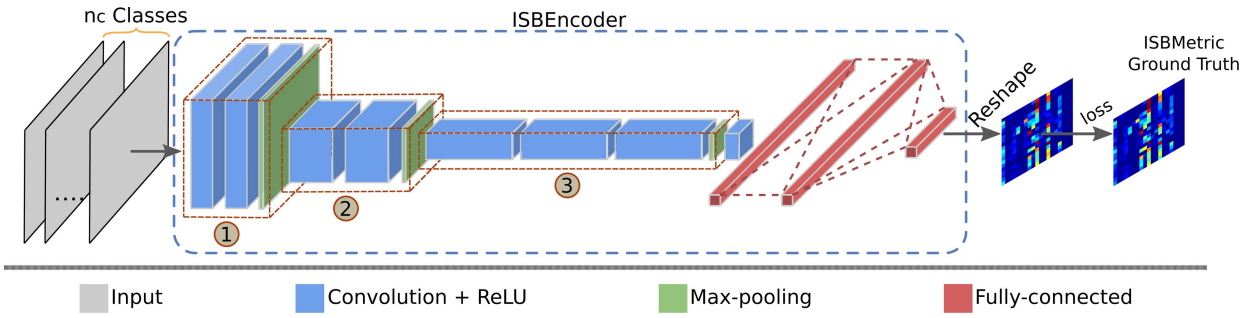


Fig. 4. An illustration of the proposed ISBEncoder architecture.

- The object 2 is still enclosed by object 1. Although the object 2 is enlarged to a size of 50×50, the second row of the m_{isb} is still $[1, 0, 0, 0]$.
- The object 3 is not adjacent to objects 1 and 2, and is fully enclosed by object 4. The third row of the m_{isb} is changed to $[0, 0, 0, 1]$.
- The perimeter of segmented object 4 is changed to $1,350$. The fourth row of the m_{isb} is changed to $[\frac{8}{27}, 0, \frac{5}{27}, 0]$.

We can see that the proposed ISBMetric is highly sensitive to the spatial adjacency changes of objects but is very robust against the object size induced biases, e.g., the value of this metric does not rely much on the object size. This way, the small object classes can contribute more to the segmentation loss, which helps improve the segmentation accuracy of the small object classes. We adopt the 4-connectivity neighboring system to measure the boundaries of object classes, where the 4-connectivity neighboring system [37] is defined regarding pixel neighborhoods. For a pixel at (x, y), a 4-connectivity neighboring $\{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$ contains only the pixels above, below, to the left and to the right of the center pixel (x, y). The boundary length is calculated considering the boundary pixel's 4 neighbors.

C. Rationale of the ISBMetric Based Encoder (ISBEncoder)

The ISBMetric is calculated based on the prediction from the segmentation network. For a $3 \times h \times w$ input RGB image, the prediction of the segmentation network is a $n_c \times h \times w$ matrix. Each pixel in the prediction map contains a set of probabilities of this pixel being in class $c \in \{1, \cdots, n_c\}$.

One obstacle towards implementing the proposed ISBMetric is that, in the training phase, it first needs to convert the predicted set of probabilities to a discrete class label, such that each pixel only has one class label. This way, the boundaries of the classes can be determined.

Let $s^{pred}(c, x, y)$ denote the probability of the c^{th} class at pixel (x, y). The discrete-class label of pixel (x, y) is determined by the index of the maximum value of the predicted class probabilities:

$$c^* = \underset{c \in \{1, \cdots, n_c\}}{\arg\max}\ s^{pred}(c, x, y), \tag{1}$$

where c^* denotes the index of the maximum value.

Using gradient descent based optimization approach for network parameters updating, the partial derivative of the forward propagation function w.r.t the network parameters must exist [38]. However, the derivative of the Eq. (1) w.r.t the index $c \in \{1, \cdots, n_c\}$ does not exist [39]. Therefore, the partial derivatives of the ISBMetric loss w.r.t. the associated network parameters cannot be retrieved. As a result, we **cannot** directly employ the ISBMetric to the network.

To circumvent this dilemma, we train a separate network to simulate the calculation of the proposed ISBMetric by taking s^{pred} as the input. The ISBEncoder works as an add-on component to the segmentation network, which aims to calculate the ISBMetric m_{isb} using the predictions (before the loss layer) from the segmentation network, i.e., the ISBEncoder extends the original segmentation network by adding a sub-network to perform a new task – ISBMetric estimation. The loss between m_{isb}^{pred} and m_{isb}^{gt} affects the parameter tuning in the segmentation network. As the loss between the m_{isb}^{pred} and m_{isb}^{gt} highlights the small object classes, the segmentation network would be forced to put more emphasis on correctly segmenting the small object classes.

D. ISBEncoder Architecture

The ISBEncoder network architecture is modified from the VGG-16 [8] network. The ISBEncoder takes the n_c-channel prediction from the segmentation network as input. This is followed by a series of three convolution blocks (①, ② and ③), as denoted in dashed brown boxes in Fig. 4, one bottleneck layer [11], and three fully connected layers. The detailed layer-wise settings are reported in Table I.

All convolutional layers in the ISBEncoder are followed by the Rectified Linear Unit (ReLU) non-linear activation layer to introduce element-wise non-linearity [40]. To reduce the feature dimensions, alleviate the memory demand and accelerate the training process [11], we introduce a bottleneck layer using 1×1 convolution kernel with stride 1 and padding 0 after the third convolutional block (③). Three fully-connected layers are applied after the bottleneck layer: the first two have $4,096$ channels each, the third performs $n_c \times n_c$-way ISBMetric prediction and thus outputs $n_c \times n_c$ channels (one for each element in the ISBMetric m_{isb}).

In training, the weights of the kernels in the three convolutional blocks (①, ② and ③) are initialized from the VGG-16 net. The weights of bottleneck layer and fully-connected layers are initialized using Xavier initialization [41]. The output of the third fully-connected layer is then reshaped to a $n_c \times n_c$ matrix to match the dimension of the

TABLE I
DETAILED CONFIGURATION OF THE PROPOSED ISBENCODER ARCHITECTURE. WE USE THE IMAGE SIZE OF 360 × 480 FOR DEMONSTRATION

Layer	Kernel Size	Stride	Pad	Output Dimension
Input	-	-	-	$n_c \times 360 \times 480$
conv1_1	3×3	1	1	$64 \times 360 \times 480$
conv1_2	3×3	1	1	$64 \times 360 \times 480$
pool1	2×2	2	0	$64 \times 180 \times 240$
conv2_1	3×3	1	1	$128 \times 180 \times 240$
conv2_2	3×3	1	1	$128 \times 180 \times 240$
pool2	2×2	2	0	$128 \times 90 \times 120$
conv3_1	3×3	1	1	$256 \times 90 \times 120$
conv3_2	3×3	1	1	$256 \times 90 \times 120$
conv3_3	3×3	1	1	$256 \times 90 \times 120$
pool3	2×2	2	0	$256 \times 45 \times 60$
conv4	1×1	1	0	$32 \times 45 \times 60$
fc5	-	-	-	4096
fc6	-	-	-	4096
fc7	-	-	-	$n_c \times n_c$

ground truth ISBMetric. The last layer is the mean-square error loss layer, which is used to calculate the distances between the predicted ISBMetric and the ground truth ISBMetric. The experimental evaluation of the ISBEncoder will be reported in Sections IV-C.2 and IV-C.3.

E. The Overall Pipeline

In the phase of pipeline training, the pipeline weights are optimized based on 1) segmentation loss and 2) ISBEncoder loss. For the segmentation network, we follow [14] and adopt the sigmoid cross-entropy loss for training:

$$\mathcal{L}_{seg} = \frac{-1}{n_c \cdot h \cdot w} \sum_{c=1}^{n_c} \sum_{y=1}^{h} \sum_{x=1}^{w} \big(s^{gt}(c,x,y) \log s^{pred}(c,x,y) + (1 - s^{gt}(c,x,y)) \log(1 - s^{pred}(c,x,y)) \big), \quad (2)$$

where $s^{gt}(c,x,y)$ denote the ground truth (0 or 1) of class c at pixel (x,y). For the ISBEncoder, we use mean square error (MSE) loss for training:

$$\mathcal{L}_{isb} = \frac{1}{n_c \cdot n_c} \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Big(m_{isb}^{pred}(i,j) - m_{isb}^{gt}(i,j) \Big)^2, \quad (3)$$

where m_{isb}^{pred} and m_{isb}^{gt} are the predicted and ground truth ISBMetric matrices, respectively. Then, the overall loss combines segmentation network loss and the ISBEncoder loss:

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda \mathcal{L}_{isb}, \quad (4)$$

where λ is a balance coefficient. The empirical selection of λ is discussed in Section IV-C. The proposed pipeline is trained in an end-to-end fashion.

IV. EXPERIMENTS

A. Datasets & Evaluation Metric

The CamVid dataset [30] is a road scene segmentation dataset which is of practical interest for various autonomous driving related problems. The dataset consists of 367 training images, 100 validation images, and 233 testing images. In total, there are 11 semantic classes that are pixel-level annotated. The resolution of the images is 360 × 480. Based on the object size [22], we denote **sign symbol**, **pedestrian**, **pole**, **bicyclist** as small-object classes. All the other 7 object classes are denoted as large-object classes.

The CityScapes dataset [31] is a recently released dataset for semantic urban street scene understanding. The dataset consists of 5,000 finely pixel-level annotated images: 2,975 training images, 500 validate images, and 1,525 testing images. In total, there are 19 semantic classes. The resolution of the image is $1,024 \times 2,048$. In addition, 20,000 coarsely annotated images are provided. In this paper, we only use finely annotated images for training. Based on the object size [22], we denote **pole**, **traffic light**, **traffic sign**, **person**, **rider**, **motorcycle** and **bicycle** as small-object classes. All the other 12 object classes are denoted as large-object classes.

For CamVid dataset, the ground-truth segmentation maps for the training, validation, and testing are given. For the CityScapes dataset, the ground-truth segmentation maps for the training and validation datasets are given while the ground-truth segmentations for the testing dataset are hidden for the user. The testing results of the CityScapes dataset are obtained via on-line submissions.

The metrics used for segmentation performance evaluation in this paper are: Class intersection over union (IoU), mean intersection over union (mIoU), mean small-object class intersection over union (mIoU_S), and mean large-object class intersection over union (mIoU_L).

B. Implementation Details

The proposed method is implemented using Caffe[1] based SegNet,[2] PyTorch[3] based FCN,[4] GCN,[5] DLA,[6] and Tensorflow[7] based DeepLab[8] and PSPNet[9] with Intel Core i7 6700K, and with NVIDIA 1080 Ti GPU with mini-batch Stochastic Gradient Descent (SGD).

The overall pipeline consists of two components: 1) Segmentation network, 2) ISBEncoder. The segmentation network weights are initialized from their pre-trained models. The segmentation network and ISBEncoder are jointly trained using SGD with momentum of 0.9, weight decay of 0.0001 and adaptive learning rates. The mini-batch size is 3 when using the CamVid dataset for the network training, and the mini-batch size is 1 when using CityScapes dataset for the network training. We initially set the learning rate as suggested in [12], [14]–[16], and [18]. We use the "poly" learning rate policy [17], where the initial learning rate is multiplied by $\left(1 - \frac{iter}{max\_iter}\right)^{power}$ with $power = 0.9$ [18]. The number

[1] http://caffe.berkeleyvision.org/
[2] https://github.com/alexgkendall/caffe-segnet
[3] https://github.com/pytorch
[4] https://github.com/wkentaro/pytorch-fcn
[5] https://github.com/zijundeng/pytorch-semantic-segmentation
[6] https://github.com/ucbdrive/dla
[7] https://www.tensorflow.org/
[8] https://github.com/tensorflow/models/tree/master/research/deeplab
[9] https://github.com/holyseven/PSPNet-TF-Reproduce

TABLE II

THE COMPARISON RESULTS OF SMALL OBJECT CLASSES (LEFT) AND LARGE OBJECT CLASSES (RIGHT) ON CAMVID TESTING DATASET (IN PERCENTAGE). FOR EACH OBJECT CLASS, THE NUMBERS WITH BETTER AND THE BEST PERFORMANCE ARE HIGHLIGHTED IN BLUE AND RED, RESPECTIVELY

	Sign symbol	Pedestrain	Pole	Bicyclist	$mIoU_S$	Building	Tree	Sky	Car	Road	Sidewalk	Fence	$mIoU_L$	mIoU
ALE	24.4	29.1	13.6	28.6	23.9	73.4	70.2	91.1	64.2	91.1	72.4	31.0	70.5	53.6
Liu & He	21.4	28.0	8.3	8.5	16.6	66.8	66.6	90.1	62.9	85.8	63.5	17.8	64.8	47.2
SuperParsing	25.4	11.6	5.2	8.9	12.8	70.4	54.8	83.5	43.3	83.4	57.4	18.3	58.7	42.0
SegNet	13.4	25.3	16.0	24.8	19.9	68.7	52.0	**87.0**	**58.5**	**86.2**	60.5	17.9	61.5	46.4
SegNet + ISBMetric-w	19.2	25.7	18.3	26.4	22.4	71.7	51.8	86.2	58.1	81.9	62.3	20.1	61.7	47.4
SegNet + ISBEncoder	**19.7**	**26.8**	**19.2**	**28.1**	**23.5**	**72.4**	**52.4**	86.8	57.3	83.3	**63.7**	**22.7**	**62.7**	**48.4**
FCN-8s	32.4	32.3	11.9	37.1	28.4	76.3	**71.2**	**87.9**	78.0	**91.9**	73.0	32.2	72.9	56.7
FCN-8s + ISBMetric-w	37.2	38.1	15.7	**43.1**	33.5	75.9	71.1	87.8	78.2	91.1	75.3	36.1	73.6	59.1
FCN-8s + ISBEncoder	**37.6**	**38.5**	**16.8**	42.5	**33.9**	**81.8**	70.1	87.7	**78.9**	91.4	**77.2**	**37.0**	**74.9**	**60.0**
DLA	48.8	58.6	27.8	55.4	47.7	**83.2**	**77.2**	**91.2**	83.6	**94.3**	81.1	32.0	77.5	66.7
DLA-34 + ISBMetric-w	50.8	60.1	**29.2**	55.9	49.0	81.7	75.6	90.9	**84.3**	93.2	81.8	33.6	77.3	67.0
DLA-34 + ISBEncoder	**51.2**	**60.3**	29.1	**56.8**	**49.4**	82.1	75.9	91.1	84.1	93.9	**82.3**	**34.7**	**77.7**	**67.4**

of overall training iterations is 20k for both CamVid and CityScapes datasets.

C. Experimental Results and Discussion

1) Comparisons to Baselines and Existing Methods: To demonstrate the effectiveness of the proposed method, we evaluate the proposed method using the baseline segmentation networks with spatial pyramid pooling-based architecture (e.g., FCN-8s [14], SegNet [15], GCN [16], PSPNet [18], and DeepLabV3 [12]), and the baseline segmentation network with feature pyramid network-based architecture (e.g., DLA [42]). For CamVid dataset, three baseline segmentation networks – FCN-8s, SegNet, and DLA – are trained and evaluated. For CityScapes dataset, five baseline segmentation networks – FCN-8s, GCN, PSPNet, DeepLabV3, and DLA are trained and evaluated. The proposed pipeline is also compared with several existing segmentation methods: ALE [43], SuperParsing [44], Liu & He [45], Deeplab-LFOV [17], and FoveaNet [13].

Using CamVid dataset for evaluation, the quantitative results are shown in Table II. We observe that by employing the proposed ISBEncoder to the baseline segmentation network, the IoU scores of the small objects classes can be significantly improved when comparing to the settings without the ISBEncoder. Combining the ISBEncoder to the SegNet, FCN-8s, and DLA baseline segmentation networks, it shows 3.6%, 5.5% and 1.7% improvements for small-object classes ($mIoU_S$), respectively. It also shows 1.2%, 2.0% and 0.2% improvements for large-object classes ($mIoU_L$) using the SegNet, FCN-8s, and DLA respectively. The overall mIoU improvements are 2.0% for SegNet, 3.3% for FCN-8s, and 0.7% for DLA.

Using CityScapes dataset for evaluation, the quantitative results are shown in Table III. We also observe significant improvements on segmenting small object classes after employing the ISBEncoder. It shows 3.2%, 3.0%, 2.8%, 1.7%, and 2.2% improvements for small-object classes ($mIoU_S$) using FCN-8s, GCN, PSPNet, DeepLab V3, and DLA, respectively. It shows 1.6%, 0.3%, 0.3%, 0.03%, and 0.04% improvements for large-object classes ($mIoU_L$) using FCN-8s, GCN, PSPNet, DeepLab V3, and DLA, respectively. The overall mIoU improvements by including ISBEncoder are 2.2% for FCN-8s, 1.3% for GCN, 1.3% for PSPNet, 0.7% for DeepLab V3, and 0.8% for DLA.

To demonstrate the effectiveness of the proposed ISBMetric, we conduct an additional experiment using the weighted loss function whose weights are based on the ISBMetric. The conventional weighted loss function used in semantic segmentation, e.g., weighted sigmoid cross-entropy loss, requires a single value as the weight for each object class. However, the weight for each object class in ISBMetric is a row vector, which makes it difficult to be directly applied to the weighted loss function. Alternatively, as each row in the ISBMetric is associated with an object class, we calculate the row-wise MSE between the ISBMetric of the segmentation prediction and the ISBMetric ground-truth, and use the calculated row-wise Mean Square Error (row-wise MSE) to weigh each object class in the segmentation loss function (weighted sigmoid cross-entropy loss). In the experiment, we first calculate the ISBMetric using the segmentation predictions. Then, we calculate the row-wise MSE based on m_{isb}^{pred} and m_{isb}^{gt}, and use the calculated row-wise MSE as the weight of the object class to train the segmentation network. Experimental results are shown in Tables II and III, in which "ISBMetric-w" denotes the method uses the ISBMetric based weighted loss function. Using CamVid testing dataset for evaluation, the experimental results demonstrate that weighing the object classes using the row-wise MSE of the ISBMetric shows 2.5%, 5.1%, and 1.3% $mIoU_S$ improvements, and 1.0%, 2.3%, and 0.4% mIoU improvements for the SegNet, FCN-8s, and DLA, respectively. Using CityScapes testing dataset for evaluation, it shows shows 2.5%, 2.2%, 2.2%, 1.3%, and 1.6% $mIoU_S$ improvements, and 1.6%, 0.8%, 0.9%, 0.3%, and 0.4% mIoU improvements for the FCN-8s, GCN, PSPNet, DeepLab v3, and DLA, respectively. In comparison to the proposed method, the segmentation performance of small object classes when using the ISBMetric based weighted loss function is better than the baselines but is slightly worse than the proposed method.

To visually demonstrate the effectiveness of the proposed ISBEncoder, we provide representative segmentation results of FCN-8s and PSPNet with or without ISBEncoder on CamVid

TABLE III

THE COMPARISON RESULTS OF SMALL OBJECT CLASSES (TOP) AND LARGE OBJECT CLASSES (BOTTOM) ON CITYSCAPES TESTING DATASET (IN PERCENTAGE). FOR EACH OBJECT CLASS, THE NUMBERS WITH BETTER AND THE BEST PERFORMANCE ARE HIGHLIGHTED IN BLUE AND RED, RESPECTIVELY

	pole	traffic light	traffic sign	person	rider	motorcycle	bicycle	mIoU$_S$
FoveaNet	62.5	69.0	77.3	80.6	60.3	65.8	76.2	70.2
Deeplab-LFOV	29.7	44.5	55.4	71.2	49.4	47.9	58.6	51.0
FCN-8s	46.6	60.0	64.7	77.2	48.2	50.4	59.8	58.1
FCN-8s + ISBMetric-w	48.1	62.8	**66.3**	78.8	**52.0**	50.9	65.2	60.6
FCN-8s + ISBEncoder	**48.8**	**63.1**	66.2	**79.2**	51.5	**52.6**	**67.9**	**61.3**
GCN	34.1	55.5	61.7	72.5	52.3	55.0	63.1	56.3
GCN + ISBMetric-w	39.7	56.2	65.9	72.8	52.9	56.1	65.8	58.5
GCN + ISBEncoder	**40.2**	**56.7**	**66.1**	**74.4**	**54.9**	**56.3**	**66.7**	**59.3**
PSPNet	62.9	69.7	77.7	80.8	61.8	66.0	77.8	71.0
PSPNet + ISBMetric-w	65.1	71.7	82.1	**83.3**	**64.6**	67.2	78.4	73.2
PSPNet + ISBEncoder	**65.8**	**72.8**	**82.4**	83.2	64.5	**67.7**	**79.9**	**73.8**
DeepLab v3	70.0	77.1	81.3	87.6	73.4	72.1	78.2	77.1
DeepLab v3 + ISBMetric-w	71.7	78.8	**82.8**	89.1	73.6	73.7	79.3	78.4
DeepLab v3 + ISBEncoder	**72.2**	**79.2**	82.7	**89.2**	**74.3**	**74.1**	**79.9**	**78.8**
DLA	63.2	70.8	75.2	84.2	64.7	64.4	73.2	70.8
DLA + ISBMetric-w	65.2	73.1	77.5	84.8	65.9	64.7	75.4	72.4
DLA + ISBEncoder	**65.5**	**73.3**	**79.4**	**85.3**	**66.2**	**65.2**	**75.9**	**73.0**

	road	sidewalk	building	wall	fence	vegetation	terrain	sky	car	truck	bus	train	mIoU$_L$	mIoU
FoveaNet	97.8	82.9	91.4	48.6	54.3	91.9	60.7	94.4	93.6	56.8	80.2	60.4	76.1	73.9
Deeplab-LFOV	97.3	77.7	87.7	43.6	40.4	89.4	67.0	92.7	91.4	48.7	56.7	49.1	70.1	63.1
FCN-8s	**97.4**	**78.4**	86.1	32.9	42.7	**89.4**	61.3	**90.9**	90.1	32.3	48.5	42.5	66.0	63.1
FCN-8s + ISBMetric-w	97.2	78.2	**86.3**	34.8	**45.1**	88.6	67.9	90.2	91.4	34.1	47.9	44.4	67.2	64.7
FCN-8s + ISBEncoder	97.3	78.2	86.0	**35.3**	44.9	89.4	**69.7**	90.0	**92.2**	**34.3**	**48.6**	**45.2**	**67.6**	**65.3**
GCN	97.3	78.5	88.4	**44.5**	48.3	90.1	69.5	**92.2**	91.0	54.6	61.6	51.6	72.3	66.4
GCN + ISBMetric-w	97.5	77.9	89.1	44.1	47.9	90.7	70.6	91.8	91.1	**54.8**	60.7	50.7	72.2	67.2
GCN + ISBEncoder	**97.6**	**79.2**	**89.1**	43.7	**48.4**	**91.2**	**71.4**	92.1	**91.7**	53.2	**61.6**	**51.6**	**72.6**	**67.7**
PSPNet	98.2	86.4	**92.9**	**58.4**	62.4	**91.6**	64.3	94.3	**95.4**	81.5	88.1	80.1	82.8	78.4
PSPNet + ISBMetric-w	97.9	86.3	92.2	56.8	65.1	90.9	66.1	94.1	95.3	**82.7**	**88.7**	79.2	82.9	79.4
PSPNet + ISBEncoder	**98.1**	**86.5**	91.7	57.4	**65.4**	91.1	**66.3**	**94.3**	95.2	82.6	88.4	**80.2**	**83.1**	**79.7**
DeepLab v3	**98.6**	**86.2**	**93.5**	**55.2**	63.2	93.8	72.3	**95.9**	96.3	75.1	90.4	**85.1**	83.8	81.3
DeepLab v3 + ISBMetric-w	96.9	85.8	91.7	54.9	62.9	**94.1**	72.5	95.8	96.1	75.3	90.6	84.9	83.5	81.6
DeepLab v3 + ISBEncoder	97.8	85.7	93.2	55.1	**64.3**	94.0	**73.3**	95.3	**96.4**	**75.7**	**90.6**	84.6	**83.8**	**82.0**
DLA	**98.3**	**83.5**	**92.1**	48.3	53.5	93.1	70.5	**94.9**	95.1	52.5	67.1	**56.9**	75.5	73.8
DLA + ISBMetric-w	96.7	83.1	90.6	48.1	53.1	93.3	70.7	94.8	94.9	52.7	67.4	56.7	75.2	74.1
DLA + ISBEncoder	97.5	82.9	91.9	**48.3**	**53.7**	**93.5**	**71.9**	94.5	**95.3**	**52.8**	**67.5**	56.3	**75.5**	**74.6**

and CityScapes datasets in Fig. 5. For small-object classes, we find that the regions segmented using ISBEncoder are more accurate, e.g., the poles, sign symbols and person, indicated by dashed rectangles, which are insufficiently segmented or totally missing when using the baseline method. By employing proposed ISBEncoder to the baseline segmentation network, it can better capture the missing components and render more accurate segmentation results.

2) Evaluation of the ISBEncoder Accuracy: We conduct three experiments to evaluate how accurate the ISBEncoder can simulate the ISBMetric matrix calculation: 1) We use the segmentation ground truth s^{gt} as the input. The output m_{isb}^{pred} is then compared to the m_{isb}^{gt}. 2) We first convert the prediction map s^{pred}, which is generated from the segmentation network, to its discrete map s^{disc} using Eq. (1). And then, we follow the same procedures described in Section III-B to calculate the m_{isb}^{disc} using the discrete map. We also use the converted discrete map as the input of the ISBEncoder, and compare the output m_{isb}^{pred} to the ISBMetric m_{isb}^{disc}. 3) We directly use the prediction map s^{pred} as the input of the ISBEncoder. The output m_{isb}^{pred} is then compared to the ISBMetric m_{isb}^{disc}.

To evaluate the accuracy of the ISBEncoder, we use Eq. (3) to calculate the mean square error between the ISBMetric matrix predicted using the ISBEncoder and the ISBMetric matrix calculated using the procedures described in Section III-B. A sample m_{isb}^{pred} and m_{isb}^{gt} are illustrated in Fig. 6. As shown in Table IV, using 1) the segmentation ground truth maps, 2) the discrete maps, and 3) the segmentation prediction maps as the inputs of the ISBEncoder, the mean square errors of the small object classes (MSE$_S$), the large object classes (MSE$_L$), and all classes (MSE) are all very small.

To evaluate the performance of the ISBEncoder in handling the object translation in the image, we conduct an experiment on a synthetic 2D shapes dataset [46]. Particularly, the synthetic dataset contains three objects of different shapes: Circle, square, and triangles.[10] There are 20, 000 paired images for training and 500 for testing. In each paired images, the same shapes are of the same size and color but of different locations and occlusions. In the images of different pairs, the sizes of the

[10]http://visualdynamics.csail.mit.edu/

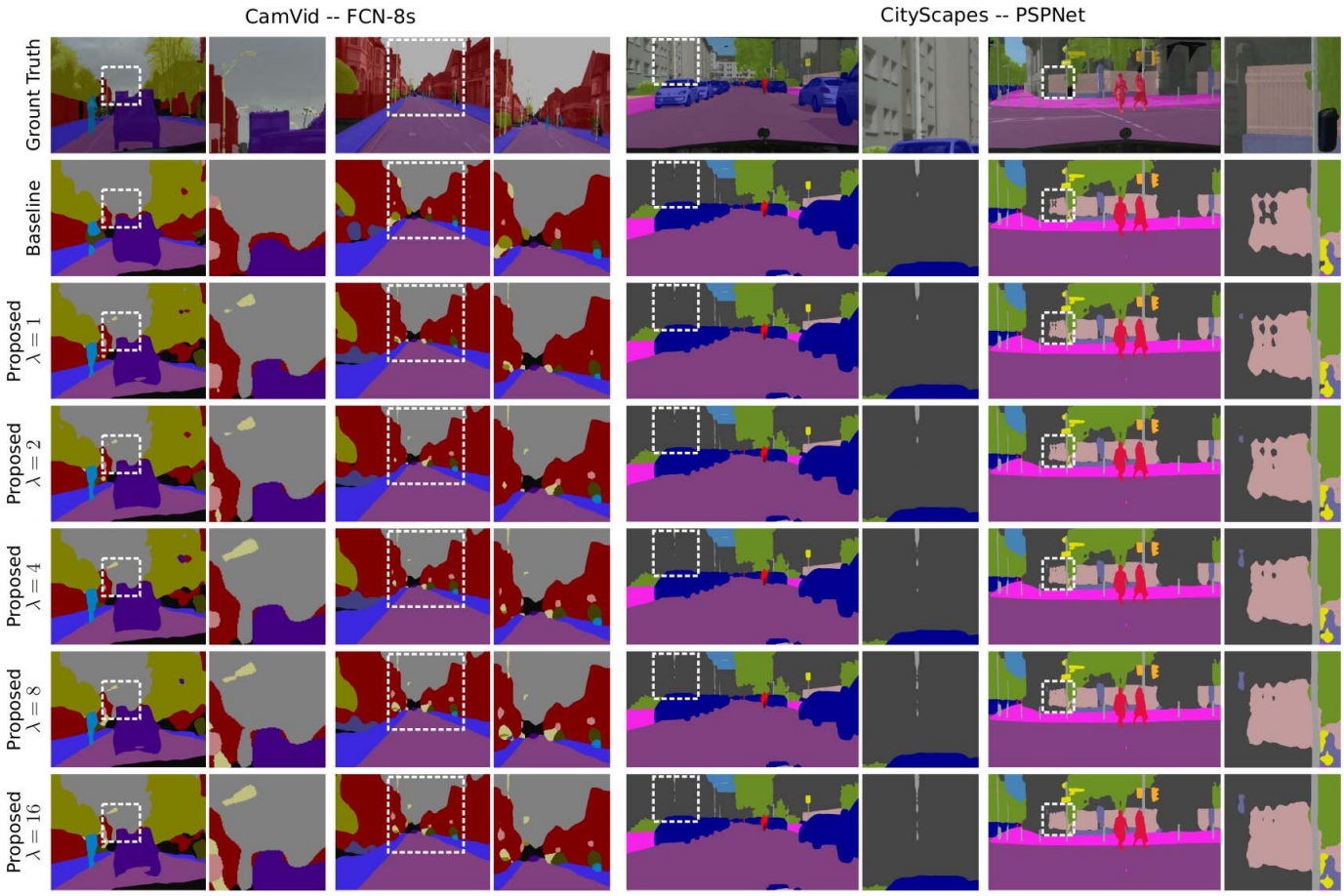


Fig. 5. Examples of semantic segmentation results on CamVID (left) and CityScapes (right) validation datasets. For visualization purpose, the ground-truth segmentation is superimposed to the input image, and the dashed rectangles are enlarged for highlighting improvements.

TABLE IV
THE ISBENCODER MEAN SQUARE ERRORS OF THE SMALL OBJECT CLASSES (MSE_S), THE LARGE OBJECT CLASSES (MSE_L), AND ALL CLASSES (MSE) ON CAMVID AND CITYSCAPES TESTING DATASETS

Input	Dataset	MSE_S	MSE_L	MSE
s^{gt}	CamVid	6.08×10^{-5}	6.09×10^{-5}	6.09×10^{-5}
	CityScapes	5.88×10^{-5}	5.89×10^{-5}	5.89×10^{-5}
s^{disc}	CamVid	6.12×10^{-5}	6.13×10^{-5}	6.13×10^{-5}
	CityScapes	5.91×10^{-5}	5.92×10^{-5}	5.92×10^{-5}
s^{pred}	CamVid	6.32×10^{-5}	6.34×10^{-5}	6.33×10^{-5}
	CityScapes	5.97×10^{-5}	5.99×10^{-5}	5.98×10^{-5}

same shapes are chosen randomly. We train the whole pipeline using the synthetic dataset. Quantitatively, we calculate the MSE between m_{isb}^{pred} and m_{isb}^{gt} for each images, it yields a mean $MSE_{2D} = 1.64 \times 10^{-5}$. The experimental results show that the mean MSE_{2D} is very small. As the predicted m_{isb}^{pred} is consistent with the ground-truth m_{isb}^{gt}, we conclude that the proposed ISBEncoder can accurately capture location variations.

3) Impact of the ISBEncoder: To evaluate the impact of the ISBEncoder, we first conduct experiments using different hyper-parameter values $\lambda \in \{0, 1, 2, 4, 8, 16\}$. Secondly, we qualitatively evaluate the feature map with or without the ISBEncoder. Thirdly, we demonstrate the impact of the ISBEncoder during network training.

Firstly, from the results shown in Tables V and VI, for small object classes, we find that, by increasing λ, the segmentation accuracy first increases to a maximum value and then slightly decrease. This can also be observed in Fig. 5, where the ability of predicting small objects is first boosted along with increased λ by weighting more on the ISBMetric loss. However, further increasing λ will cause a possible increase in false positives. For example, in the second column of Fig. 5, the poles in the white dashed-rectangle are insufficiently segmented by the baseline segmentation network. By increasing λ (from 1 to 4), the network shows an improved performance on capturing more pole pixels. However, keep increasing λ (from 8 to 16), the network mistakenly classify the pixels around the poles in the building as poles, such that the poles become visually thicker as λ increases.

Secondly, we demonstrate the impact of the ISBEncoder by visualizing a sample feature map (after "conv6"/the last convolutional layer in the PSPNet) using a sample training image from the CityScapes dataset. As shown in Fig. 7(c), most of the small objects are not highlighted in the feature map when using the baseline segmentation network. Whereas,

TABLE V

THE COMPARISON RESULTS OF SMALL OBJECT CLASSES (LEFT) AND LARGE OBJECT CLASSES (RIGHT) USING THE PROPOSED PIPELINE WITH DIFFERENT HYPERPARAMETER λ (IN EQ. (4)) ON CAMVID TESTING DATASET (IN PERCENTAGE). THE BASELINE SEGMENTATION NETWORK IS FCN-8S. FOR EACH OBJECT CLASS, THE NUMBERS WITH THE BEST PERFORMANCE ARE HIGHLIGHTED IN RED

	Sign symbol	Pedestrain	Pole	Bicyclist	mIoU_S	Building	Tree	Sky	Car	Road	Sidewalk	Fence	mIoU_L	mIoU
$\lambda = 0$	32.4	32.3	11.9	37.1	28.4	76.3	**71.2**	**87.9**	78.0	**91.9**	73.0	32.2	72.9	56.7
$\lambda = 1$	36.9	37.6	12.6	42.0	32.3	81.3	70.0	87.7	79.2	91.6	77.4	36.7	74.8	59.4
$\lambda = 2$	37.6	**38.5**	16.8	**42.5**	33.9	**81.8**	70.1	87.9	78.9	91.4	77.2	**37.0**	**74.9**	**60.0**
$\lambda = 4$	38.4	37.8	**17.2**	42.3	**33.9**	81.8	68.9	87.5	**79.1**	91.4	**77.3**	36.6	74.7	59.8
$\lambda = 8$	**40.0**	37.2	13.3	40.2	32.7	80.2	69.8	87.1	77.9	91.4	77.1	35.1	74.1	59.0
$\lambda = 16$	37.7	35.4	14.3	40.7	32.0	76.0	69.4	87.1	78.5	91.0	75.6	36.1	73.4	58.3

TABLE VI

THE COMPARISON RESULTS OF SMALL OBJECT CLASSES (TOP) AND LARGE OBJECT CLASSES (BOTTOM) USING PROPOSED PIPELINE WITH DIFFERENT HYPER-PARAMETER λ (IN EQ. (4)) ON CITYSCAPES TESTING DATASET (IN PERCENTAGE). THE BASELINE SEGMENTATION NETWORK IS PSPNET. FOR EACH OBJECT CLASS, THE NUMBERS WITH THE BEST PERFORMANCE ARE HIGHLIGHTED IN RED

	pole	traffic light	traffic sign	person	rider	motorcycle	bicycle	mIoU_S
$\lambda = 0$	62.9	69.7	77.7	80.8	61.8	66.0	77.8	71.0
$\lambda = 1$	64.8	70.7	79.2	82.1	63.6	66.7	79.4	72.4
$\lambda = 0$	65.0	71.2	81.1	82.4	64.4	66.9	79.6	72.9
$\lambda = 4$	**65.9**	72.6	82.2	82.9	**64.7**	67.5	79.8	73.7
$\lambda = 8$	65.8	**72.8**	**82.4**	**83.2**	64.5	**67.7**	**79.9**	**73.8**
$\lambda = 16$	65.6	69.9	80.9	81.8	63.9	66.5	79.8	72.6

	road	sidewalk	building	wall	fence	vegetation	terrain	sky	car	truck	bus	train	mIoU_L	mIoU
$\lambda = 0$	**98.2**	86.4	**92.9**	**58.4**	62.4	**91.6**	64.3	94.3	95.4	81.5	88.1	80.1	82.8	78.4
$\lambda = 1$	98.1	85.9	91.5	56.2	63.7	90.7	64.9	94.4	94.8	82.8	87.4	79.4	82.5	78.8
$\lambda = 2$	98.1	85.9	91.5	56.2	64.4	90.7	65.7	94.4	94.8	82.8	87.4	79.4	82.6	79.0
$\lambda = 4$	98.1	86.4	91.8	56.9	64.9	91.4	66.2	**94.6**	**95.2**	82.5	**88.7**	**80.5**	**83.1**	79.6
$\lambda = 8$	98.1	**86.5**	91.7	57.4	**65.4**	91.1	**66.3**	94.3	95.2	**82.6**	88.4	80.2	83.1	**79.7**
$\lambda = 16$	98.0	86.3	91.6	56.7	64.7	90.8	65.6	93.5	94.9	82.4	87.7	80.3	82.7	79.0

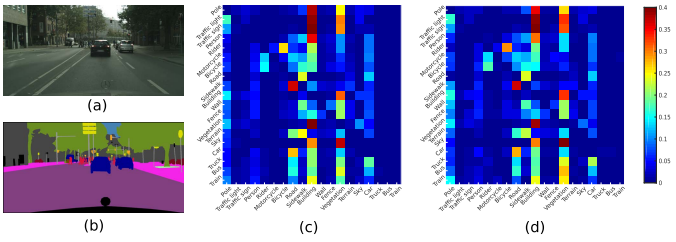


Fig. 6. An illustration of the predicted ISBMetric m_{isb}^{pred} and the ground truth ISBMetric m_{isb}^{gt} of a sample image from CityScapes training dataset. The intensity of the colorbar denotes the value. (a) Input Image. (b) Ground Truth. (c) Predicted ISBMetric (m_{isb}^{pred}). (d) Ground Truth ISBMetric (m_{isb}^{gt}).

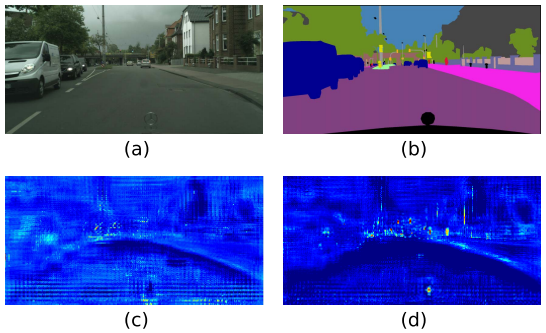


Fig. 7. An illustration of the impact of the proposed ISBEncoder on the feature map after the "conv6"/the last convolutional layer in the PSPNet. The input image is from the CityScapes training dataset. (a) Input. (b) Ground Truth. (c) Feature Map Without ISBEncoder. (d) Feature Map With ISBEncoder.

as shown in Fig. 7(d), the small objects are better highlighted in the feature map when employing the ISBEncoder to the segmentation network.

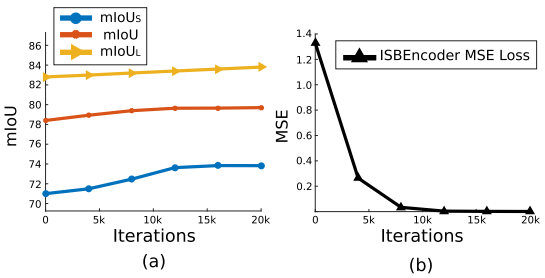


Fig. 8. (a) The mIoU of large object classes, overall classes, and small object classes over 20k training iterations on CityScapes training dataset (in percentage). (b) The mean square error (MSE) loss of ISBEncoder over 20k training iterations. The baseline segmentation network is PSPNet.

Thirdly, we demonstrate impact of the ISBEncoder during training. The ISBEncoder convergence is shown in Fig. 8, from which we can see that mIoU_S and mIoU_L are improved with the increase of iterations. The ISBEncoder MSE loss is converged after 8k iterations. We observe that the improvement of mIoU_S is more significant, while the improvement of mIoU_L is smaller. In summary, both qualitative and quantitative results verify that the proposed ISBEncoder can effectively improve the segmentation accuracy of small objects.

V. CONCLUSION

This paper proposed an ISBMetric to measure the level of spatial adjacency between each pair of object classes, and proposed an ISBEncoder to enforce the ISBMetric in the segmentation of urban street scene. Based on the experiment results, the proposed method can substantially improve the

segmentation accuracy of small objects, as well as improve the overall segmentation performance. The proposed ISBMetric is evaluated based on FCN-8s, SegNet, GCN, PSPNet, DeepLab V3, and DLA networks on CamVid and CityScapes datasets. Moreover, the proposed ISBEncoder can be easily combined to many state-of-the-art segmentation networks without adding extra time or cost in deployment.

REFERENCES

[1] X. Gao, B. Wang, D. Tao, and X. Li, "A relay level set method for automatic image segmentation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 518–525, Apr. 2011.

[2] X. Yang, X. Gao, D. Tao, X. Li, and J. Li, "An efficient MRF embedded level set method for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 9–21, Jan. 2015.

[3] K. Zhang, Q. Liu, H. Song, and X. Li, "A variational approach to simultaneous image segmentation and bias correction," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1426–1437, Aug. 2015.

[4] J. Peng, J. Shen, and X. Li, "High-order energies for stereo segmentation," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1616–1627, Jul. 2016.

[5] H. Lu, R. Zhang, S. Li, and X. Li, "Spectral segmentation via midlevel cues integrating geodesic and intensity," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2170–2178, Dec. 2013.

[6] H. Lu, G. Fang, X. Shao, and X. Li, "Segmenting human from photo images based on a coarse-to-fine scheme," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 889–899, Jun. 2012.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[10] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3640–3649.

[11] W.-C. Hung *et al.*, "Scene parsing with global context embedding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2650–2658.

[12] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. (2017). "Rethinking atrous convolution for semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1706.05587v2

[13] X. Li *et al.*, "FoveaNet: Perspective-aware urban scene parsing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 784–792.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[16] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1743–1751.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. (2016). "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs." [Online]. Available: https://arxiv.org/abs/1606.00915

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2881–2890.

[19] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2016, pp. 1–9.

[20] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2129–2137.

[21] Z. Meng, X. Fan, X. Chen, M. Chen, and Y. Tong, "Detecting small signs from large images," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2017, pp. 217–224.

[22] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1951–1959.

[23] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, "Lost and found: Detecting small road hazards for self-driving vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 1099–1106.

[24] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2017, pp. 1025–1032.

[25] K. Gupta, S. A. Javed, V. Gandhi, and K. M. Krishna. (2018). "MergeNet: A deep net architecture for small obstacle discovery." [Online]. Available: https://arxiv.org/abs/1803.06508

[26] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5325–5334.

[27] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.

[28] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

[29] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1529–1537.

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

[31] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3213–3223.

[32] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul./Aug. 2011, pp. 2809–2813.

[33] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3626–3633.

[34] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.

[35] D. Lin, G. Chen, D. Cohen-Or, P.-A. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1320–1328.

[36] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang, "Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands: Springer, 2016, pp. 664–679.

[37] G. Feng, S. Wang, and T. Liu, "New benchmark for image segmentation evaluation," *Proc. SPIE*, vol. 16, no. 3, p. 033011, Jul. 2007.

[38] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE Int. Conf. Neural Netw.*, Mar./Apr. 1993, pp. 586–591.

[39] V. Stoyanov, A. Ropson, and J. Eisner, "Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 725–733.

[40] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

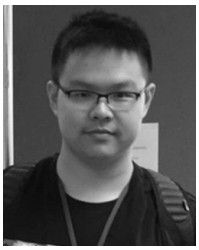
[42] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. (2017). "Deep layer aggregation." [Online]. Available: https://arxiv.org/abs/1707.06484

[43] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE 12nd Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 739–746.

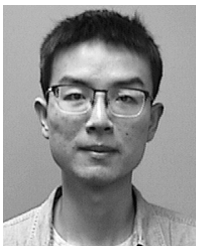
[44] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 352–365.

[45] B. Liu and X. He, "Multiclass semantic video segmentation with object-level active inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4286–4294.
[46] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 91–99.

Dazhou Guo received the B.S. degree in electronic engineering from the Dalian University of Technology, Dalian, China, in 2008, and the M.S. degree in information and informatics engineering from Tianjin University, Tianjin, China, in 2010. He is currently pursuing the Ph.D. degree in computer science with the University of South Carolina, Columbia, SC, USA. His research interests include computer vision, medical image processing, and machine learning.



Ligeng Zhu received the B.Sc. degree in computer science from Simon Fraser University in 2019 and the B.Eng. degree in computer science from Zhejiang University in 2019. He is currently a Visiting Student at the Massachusetts Institute of Technology. His research interests mainly focus on efficient machine learning and computer vision.



Yuhang Lu received the B.E. degree from the Chengdu University of Technology in 2013 and the M.E. degree from Wuhan University in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of South Carolina. His research interests include computer vision, machine learning, and image processing.

Hongkai Yu received the Ph.D. degree in computer science and engineering from the University of South Carolina, Columbia, SC, USA, in 2018. He then joined the Department of Computer Science, University of Texas–Rio Grande Valley, Edinburg, TX, USA, as an Assistant Professor. His research interests include computer vision, machine learning, deep learning, and intelligent transportation system.

Song Wang received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 2002. He was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning. He is a Senior Member of the IEEE and a member of the IEEE Computer Society. He is currently serving as the Publicity/Web Portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society and an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition Letters*, and *Electronics Letters*.