Importance-Aware Semantic Segmentation for Autonomous Vehicles

Bike Chen, Chen Gong, *Member, IEEE*, Jian Yang, and *Member, IEEE*

***Abstract*— Semantic segmentation (SS) partitions an image into several coherent semantically meaningful parts and classifies each part into one of the pre-determined classes. In this paper, we argue that the existing SS methods cannot be reliably applied to autonomous driving system as they ignore the different importance levels of distinct classes for safe driving. For example, pedestrian, car, and bicyclist in the scene are much more important than sky and building when driving a car, so their segmentations should be as accurate as possible. To incorporate the importance information possessed by various object classes, this paper designs an "importance-aware loss" (IAL) that specifically emphasizes the critical objects for autonomous driving. The IAL operates under a hierarchical structure and the classes with different importance are located in different levels so that they are assigned distinct weights. Furthermore, we derive the forward and backward propagation rules for IAL and apply them to four typical deep neural networks for realizing SS in an intelligent driving system. The experiments on CamVid and Cityscapes data sets reveal that, by employing the proposed loss function, the existing deep learning models, including FCN, SegNet, ENet, and ERFNet, are able to consistently obtain the improved segmentation results on the pre-defined important classes for safe driving.**

***Index Terms*— Semantic segmentation, importance-aware loss, deep leaning, autonomous driving.**

I. Introduction

SEMANTIC Segmentation (SS) separates an image into different meaningful parts that indicate distinct objects, which serves as a powerful and practical tool for the further image analysis such as scene categorization, human-machine interaction, and visual question answering. In recent years, autonomous driving system has attracted intensive attention, in which SS has played an important role in detecting obstacles and understanding traffic conditions [1], [2]. Apparently, high segmentation accuracy in autonomous driving system will make the system comprehensively understand the driving environment, and thus greatly improve the driving safety.

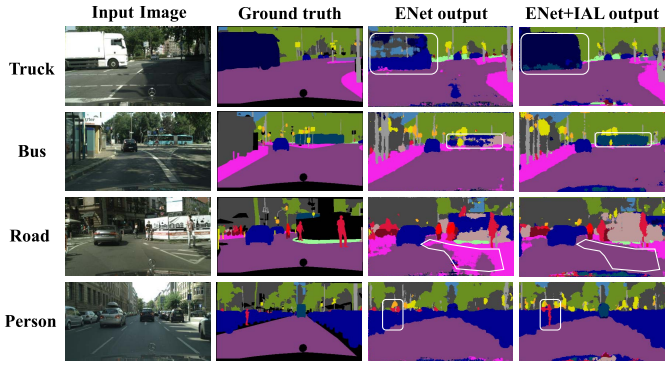


Fig. 1. Representative segmentation results of the primitive ENet and our ENet+IAL on important object classes of Cityscapes dataset. For the important classes (e.g., truck, bus, and road), we see that the regions segmented by ENet+IAL are more coherent and complete than ENet. For the class of person shown in the last row, ENet also yields much worse results than ENet+IAL. **Best viewed in color.**

However, we argue that the SS associated with autonomous driving system [3] is quite different from the conventional SS problems. For traditional SS, all the objects appeared in an image are of equal importance and one should segment all of them from the image as accurately as possible. That is to say, all image pixels share the same weight when we establish the corresponding SS models. In contrast, the objects in the real traffic scenes are not equally important for autonomous vehicles. For instance, the self-driving system should pay more attention to the objects that are closely related to safe-driving than those that are not often used for vehicle control. In other words, the SS algorithm in autonomous vehicles should segment the major obstacles and potential driving risks (e.g., pedestrians, cyclists, other vehicles, and traffic signs) with a high precision, while reducing the attention on processing less important objects such as sky, vegetation, and the buildings off the road.

In this sense, existing SS methods are improper for dealing with autonomous driving problem as they have not taken the object importance into consideration. For example, the traditional works based on handcrafted features [4], [5] and the recent Deep Convolutional Neural Network (DCNN) based methods [6]–[8] equally treat all the appeared classes. As a result, they generate very low accuracy on segmenting the important objects as mentioned above. As shown in Fig. 1, we observe that for the important objects such as truck, bus,

Manuscript received July 11, 2017; revised December 7, 2017; accepted January 25, 2018. This work was supported in part by the NSF of China under Grant U1713208, Grant 61602246 and Grant 61472187, in part by the 973 Program under Grant 2014CB349303, in part by the Program for Changjiang Scholars, in part by the NSF of Jiangsu Province under Grant BK20171430 and Grant BK20170857, and in part by the Six Talent Peak Project of Jiangsu Province of China under Grant DZXX-027. The Associate Editor for this paper was D. Fernandez-Llorca. *(Corresponding authors: Chen Gong; Jian Yang.)*

The authors are with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: bikechen@njust.edu.cn; chen.gong@njust.edu.cn; csjyang@njust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2801309

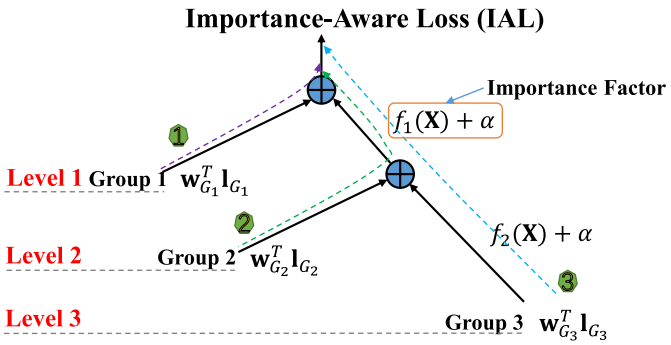


Fig. 2. The illustration of our importance-aware loss with hierarchical structure. Level 1 to Level 3 indicate the importance levels of the classes in different groups, and the more important a Group is, the higher level it stands. $\mathbf{l}_{G_1}$, $\mathbf{l}_{G_2}$, and $\mathbf{l}_{G_3}$ are respectively the loss values of three groups calculated by cross-entropy loss. Besides, $\mathbf{w}_{G_1}$, $\mathbf{w}_{G_2}$, and $\mathbf{w}_{G_3}$ are the weights for eliminating class imbalance correspondingly. The term $f_t(\mathbf{X})+\alpha$ $(t = 1, 2)$ is called *importance factor*.

and road, the segmentation results produced by the primitive ENet model [8] are incomplete. More seriously, we see that the person ahead of the car (see the last row) has been totally missed by ENet, which may pose great risk to the pedestrian's life under practical situations.

From above explanations, we see that existing methodologies cannot render reliable segmentation results for autonomous driving. This is because they all adopt the cross-entropy [9] loss function for model training, which equally evaluates the errors incurred by all image pixels without focusing on the important objects. As a result, these conventional SS approaches cannot assign different weights to different object classes. Therefore, a novel importance-aware loss function should be specifically designed for the application of automatic driving. To this end, we introduce the notion of *class importance* where pedestrians, vehicles, and other objects on the road are more important for driving than other classes such as sky and remote buildings that are off the road. Based on this notion, we design a novel loss function termed "Importance-Aware Loss" (IAL) that is able to put more emphasis on accurately segmenting the important objects than less important ones. From the last column of Fig. 1, we notice that the segmentation errors produced by the original ENet can be corrected if our proposed IAL is incorporated (i.e., "ENet+IAL"). It can be easily found that ENet+IAL can not only produce very compact segmentation results on large targets such as truck, bus, and road, but also successfully pick up small object like person.

Inspired by [10], we propose a novel loss function with hierarchical structure as shown in Fig. 2. In this structure, the objects with different degrees of importance are located in different levels, and the more important an object is, the higher level it stands. Consequently, the important objects are in higher levels than the unimportant ones, and thus they are multiplied by larger importance factors for computing the final loss. To validate our proposed loss function, we replace the cross-entropy loss utilized by representative deep learning methods [6]–[8], [11] with our proposed importance-aware loss. The experimental results on two typical autonomous driving datasets including CamVid [12] and Cityscapes [13] firmly demonstrate that the important objects can be segmented more precisely than existing approaches.

This paper is the extended version of our previous conference work [14]. Specifically, we conduct more empirical studies on the proposed algorithm including investigating the model behavior with cross-entropy loss of uniform class weights, comparing a recent efficient and effective ERFNet model, exploring the sensitivity analysis for the important tuning parameters, providing the comparisons of training time between cross-entropy loss based models and corresponding importance-aware loss based models.

Notations: We first define some notations for the ease of following descriptions. The final output of an SS algorithm is represented by a tensor $\boldsymbol{X} \in \mathbb{R}^{C\times H_{img}\times W_{img}}$ where its height and width correspond to a $H_{img} \times W_{img}$ input image, and its depth targets the one-hot encoding of the ground truth and indicates the class of each of the $H_{img} \times W_{img}$ pixels. Here the one-hot encoding is employed for class indication which has the formation $[0, \cdots, 0, 1, 0, \cdots, 0]^T$ with the element corresponding to the correct label being 1. Besides, the segmentation ground truth of an image is denoted by a matrix $\mathbf{Y} \in \mathbb{N}^{H_{img}\times W_{img}}$ with the (i, j)-th element $\mathbf{Y}_{i,j} \in \{1, 2, \cdots, C\}$ representing the corresponding label of the (i, j)-th pixel. Here C is the total number of pre-defined classes in driving environment.

Organization: The rest of this paper is organized as follows. In Section 2, some related works are reviewed. After that, we describe the proposed loss function and also the relationship with existing cross-entropy loss in Section 3. In Section 4, we derive the forward-backward propagation rules for our proposed loss function. In Section 5, we provide experimental results on the representative traffic datasets including CamVid and Cityscapes. Sensitivity analyses of parameters are also presented in this section. Finally, the entire paper is concluded in Section 6.

II. Related Work

SS has been intensively studied for a long time as it is an important tool for understanding a scene. For example, some traditional methods focus on designing powerful hand-crafted features and using Random Forest method [4], [15], [16], Mean Shift technique [17], JSEG [18], Graph-based approach [19], and Statistical Region Merging method [20] Boosting-based technique [21]–[23] for predicting the class of image pixels. Specifically, they comprehensively combine different kinds of features such as motion point clouds, appearance-based descriptors, and depth information [24] to achieve a coherent spatial segmentation. Moreover, Wang and Wang [25] provided detailed analyses and assessments for some representative image segmentation methods. To improve the segmentation accuracy, some post-processing strategies have been developed to improve the initial segmentation results. For instance, the techniques based on Conditional Random Fields (CRF) [5], [21], [26] are used to suppress the per-pixel prediction noise output by the classifiers. The energy function of the CRF model usually combines the results

from pairwise relationships between mid-level cues such as superpixels, and low-level pixel-based unary and pairwise relations.

With the rapid development of deep learning, various deep neural networks have been applied to SS and achieved state-of-the-art performance. The works such as [27]–[29] employ the features extracted by DCNN for class prediction. However, the feature extraction and pixel classification in these works are isolated. To make SS an end-to-end process, Long *et al.* [6] transform a classification-purposed DCNN to output a spatial pixel-wise prediction by replacing fully connected layers with convolutional layers. Moreover, to improve the spatial details, Long *et al.* [6] fuse the coarse and high-level information to the fine and low-level information, which contributes to promising results. Based on [6], many other methods [30]–[33] are proposed which further incorporate multi-scale manipulation or post-processing based on CRF. Another important architecture for segmentation is based on the structure of encoder-decoder. SegNet [7] and some other works like [34]–[36] belong to this type. For SegNet, Vijay Badrinarayanan and Cipolla [7] use the max-pooling indices to perform non-linear upsampling, which eliminates the need for learning to upsample. Here the max-pooling indices are computed and stored in the max-pooling step of the encoder part. Then the upsampled maps are convolved with trainable filters to produce dense pixel-wise prediction.

Recently, there are some attempts to distinguish different image pixels for SS task. For example, Bulò *et al.* [37] adaptively reweight the contributions of each pixel to address the problem of long-tail distribution, which means that few object categories comprise the majority of data and consequently leading to the biased classification results. Li *et al.* [38] consider that different pixels have different levels of difficulty, and propose a difficulty-aware neural network for SS. In their network, the earlier sub-models are trained to handle easy and confident regions, while the later sub-models concentrate on harder and ambiguous regions. However, these two works are very different from our method which cares about the importance of pixels in intelligent vehicles.

Recently, several works have been done to apply SS to autonomous driving. Pohlen *et al.* [39] develops a deep neural network for segmenting the major object classes in street scenes and reaches state-of-the-art results on the Cityscapes benchmark [13]. To further improve the efficiency and achieve real-time segmentation, Paszke *et al.* [8] specifically design a new deep neural network architecture termed ENet which can be viewed as a special case of ResNet [40]. Similarly, Treml *et al.* [41] also design a new network for the embedded devices in self-driving cars. Their architecture consists of ELU activation functions, a SqueezeNet-like encoder, parallel dilated convolutions, and a decoder with SharpMask-like refinement modules. Recently, Romera *et al.* [11], [42] proposed a new efficient and effective network which is similar to the ENet. This method adopted the specifically designed non-bottleneck-1D layer and deconvolution technique to remarkably improve its performance.

Although above SS algorithms targeting self-driving have achieved encouraging performance to some extent, none of them take the importance of different classes into account, so their results are not reliable for autonomous driving as mentioned in the introduction. Therefore, this paper presents the concept of *class importance* and proposes a novel loss function with hierarchical structure. By embedding the proposed loss to four representative deep networks such as ENet, SegNet, FCN, and ERFNet, we will show that our loss is able to attract the network's attention to important objects during self-driving.

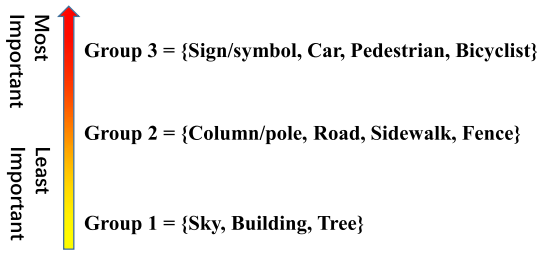


Fig. 3. The rankings of importance of 11 studied object classes. Group 3 is the most important and Group 1 is the least important.

III. The Proposed Model

In this section, we firstly detail our proposed loss function and then deduce its forward and backward propagation rules. Finally, we describe the way for applying our loss to four typical neural networks to handle the SS task.

A. The Proposed Loss Function

As mentioned in the introduction, different object classes have different levels of importance for autonomous driving, so this section introduces our proposed loss function that takes the importance information into consideration. To make the following explanations clear, we adopt CamVid [12] dataset as an instance. CamVid [12] is a widely used dataset for evaluating the self-driving performance, in which the image data is captured from the perspective of a driving automobile. This dataset suggests 11 meaningful object classes that are often appeared in a driving scenario, and in this section we use these 11 suggested classes for detailed descriptions.

First of all, safety is the most critical issue for driving where the collisions with car, pedestrian, and bicyclist are strongly opposed. Besides, the traffic lights and signs are also essential to serve as important signals, so these objects show the top level importance in our algorithm. In contrast, road, sidewalk, column/pole, and fence are less important as they only guarantee the normal driving. Sky, buildings, and tree that are off the road are not essential here as they are seldom used as a cue for car control, so they are the least important among all above 11 classes. The detailed importance levels of all the investigated classes are depicted in Fig. 3. It is worth mentioning that the users can re-define the objects' importance levels according to different criteria or their own prior knowledge.

According to the importance levels as shown in Fig. 3, we propose a novel importance-aware loss with hierarchical structure as illustrated in Fig. 2, in which different levels represent the objects with different importance. In Fig. 2,

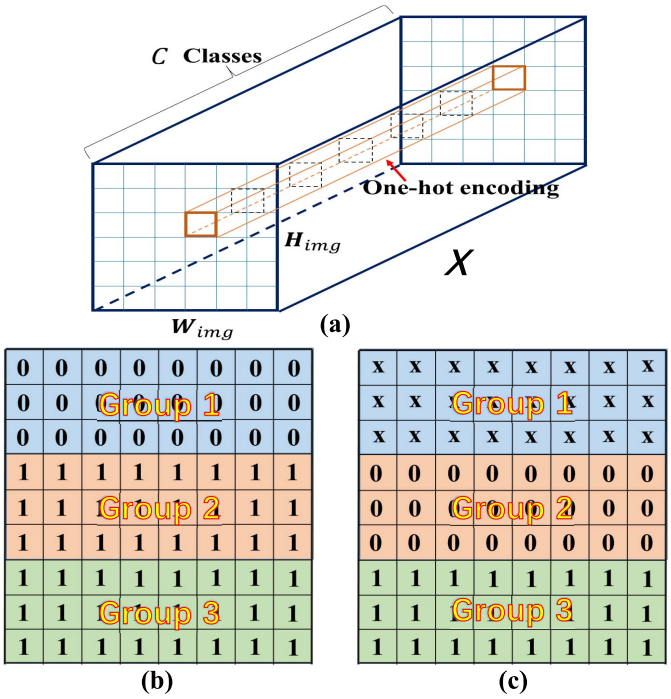


Fig. 4. Illustration of critical mathematical definitions in our method. (a) The output of the algorithm is represented by a tensor X, of which the height and width represent the $H_{img} \times W_{img}$ image, and the depth corresponds to the totally C classes. For a specific pixel, the depth corresponds to a *one-hot encoding*. (b) and (c) respectively present the $H_{img} \times W_{img}$ matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ for comparing the importance levels of three groups, in which we assume that the pixels of every Group are arranged together (see the blocks with different colors).

the vectors $\mathbf{l}_{G_1}$, $\mathbf{l}_{G_2}$, and $\mathbf{l}_{G_3}$ encode the values of cross-entropy loss [9] of the objects in Group 1, Group 2, and Group 3, respectively, and the j-th element $(\mathbf{l}_{G_i})_j$ $(i = 1, 2, 3)$ is defined by

$$(\mathbf{l}_{G_i})_j = -\sum_c \mathbf{q}_c \log(\mathbf{p}_c), \tag{1}$$

where $\mathbf{p}_c = exp(X_{c,i,j})/\sum_{k=1}^{C} exp(X_{k,i,j})$ is the probability of the (i, j)-th pixel belonging to the c-th class (c takes a value from $1, 2, \cdots, C$) based on the output X, and $\mathbf{q}$ is a one-hot encoding with the c-th element $\mathbf{q}_c$ being 1. Similar to the formation of $\mathbf{l}_{G_i}$, we use the vectors $\mathbf{w}_{G_1}$, $\mathbf{w}_{G_2}$, and $\mathbf{w}_{G_3}$ to record the corresponding weights of the objects in the three groups for avoiding class imbalance, and the object with fewer pixels is assigned larger weight [8]. The j-th element in $\mathbf{w}_{G_i}$ $(i = 1, 2, 3)$ are

$$(\mathbf{w}_{G_i})_j = \frac{1}{\ln(a + \mathbf{freq}_{i,j})}, \tag{2}$$

where $\mathbf{freq}_{i,j}$ is the total number of pixels of the j-th class in Group i divided by the total number of pixels of all images belonging to training dataset, and a is a tuning parameter that is set to 1.02 [8]. Therefore, the weighted cross-entropy losses for Group 1 to Group 3 are $\mathbf{w}_{G_1}^T\mathbf{l}_{G_1}$, $\mathbf{w}_{G_2}^T\mathbf{l}_{G_2}$, and $\mathbf{w}_{G_3}^T\mathbf{l}_{G_3}$ correspondingly.

Besides, for the three groups defined in Fig. 2, we introduce two $H_{img} \times W_{img}$ matrices $\mathbf{M}_t$ $(t = 1, 2)$ to model the importance relationship of three groups. For example, the $\mathbf{M}_1$ for comparing Group 1 and Groups 2, 3 is presented in Fig. 4(b), in which the elements corresponding to the classes in Group 1 are set to 0, and the elements corresponding to Groups 2∼3 are 1 indicating that they are more important than Group 1. To further compare the importance of Group 2 and Group 3, the elements of $\mathbf{M}_2$ (see Fig. 4(c)) regarding Group 2 are set to 0, and the elements of Group 3 are defined as 1 because they are more important than Group 2. In $\mathbf{M}_2$, the elements indicating Group 1 are denoted as "x" which means that the comparison of Group 1 and other groups has been done before.

Based on $\mathbf{M}_t$ $(t = 1, 2)$, we define $f_t(\mathbf{X}) + \alpha$ as an *importance factor* where α is a tuning parameter with default value 1, so the $f_t(\mathbf{X})$ $(t = 1, 2)$ in Fig. 2 are computed by

$$f_t(\mathbf{X}) = \frac{1}{2} \parallel (\mathbf{M}_t + \lambda\mathbf{E})^{\frac{1}{2}} \odot (\mathbf{X} - \mathbf{M}_t) \odot \mathbb{I}\{\mathbf{M}_t \neq \text{"x"}\} \parallel_F^2, \tag{3}$$

where $\mathbf{E}$ is an all-one matrix, and $\mathbb{I}\{\mathbf{M}_t \neq \text{"x"}\}$ returns a matrix where its element is 1 if the corresponding element $(\mathbf{M}_t)_{i,j}$ is not "x", and 0 otherwise. The notation "$\odot$" denotes the element-wise product of two matrices. $\mathbf{X}$ is a matrix with the same dimension of $\mathbf{Y}$ and its (i, j)-th element is defined by $\mathbf{X}_{i,j} = X_{c,i,j}$[1] with $c = \mathbf{Y}_{i,j}$. In (3), $\lambda \in \mathbb{R}^+$ is a tuning parameter which we set to 0.5 in this paper. Note that if λ is small, the value of $f_t(\mathbf{X})$ will be large due to the error between $\mathbf{X}_{i,j}$ and $(\mathbf{M}_t)_{i,j}$ when $(\mathbf{M}_t)_{i,j} = 1$ (i.e., the corresponding class is important). By this way, Eq. (3) encourages the model to focus on the classifications of important classes. The discussion of the effect of λ to model output is referred to Section IV-C.

Therefore, the loss of the objects in the three groups can be computed by following the arrows in Fig. 2. For instance, Group 1 has the lowest importance level, of which the importance-aware loss is $\mathbf{w}_{G_1}^T\mathbf{l}_{G_1}$; The weighted cross-entropy loss of Group 2 should be multiplied by an importance factor $f_1(\mathbf{X}) + \alpha$, so its importance-aware loss should be $(f_1(\mathbf{X}) + \alpha)(\mathbf{w}_{G_2}^T\mathbf{l}_{G_2})$. Similarly, the classes in Group 3 are the most important and thus its weighted cross-entropy loss $\mathbf{w}_{G_3}^T\mathbf{l}_{G_3}$ should be augmented by two importance factors. Consequently, the loss of Group 3 is $(f_1(\mathbf{X}) + \alpha)(f_2(\mathbf{X}) + \alpha)(\mathbf{w}_{G_3}^T\mathbf{l}_{G_3})$. Finally, the total value of our importance-aware loss is the sum of the loss values contributed by the three groups, which is

$$\begin{aligned} Loss = {} & \mathbf{w}_{G_1}^T\mathbf{l}_{G_1} \\ & + (f_1(\mathbf{X}) + \alpha)(\mathbf{w}_{G_2}^T\mathbf{l}_{G_2}) \\ & + (f_1(\mathbf{X}) + \alpha)(f_2(\mathbf{X}) + \alpha)(\mathbf{w}_{G_3}^T\mathbf{l}_{G_3}). \end{aligned} \tag{4}$$

One may argue that the weight for a certain class can be manually specified based on the frequency of the pixels belonging to this class. However, here we want to clarify that the importance of an object in our work is not governed by its size. In fact, whether an object is important or not is dependent on its impact on driving safety. For example, although the pedestrian region is much small than the sky

[1]Here all elements belonging to the (i, j)-th pixel (i.e., "$X_{:,i,j}$" in Matlab expression) have been normalized to [0, 1].

region in an image, it should be paid more attention as it is very critical to avoiding accident. Therefore, we cannot manually tune the class weights in the cross-entropy criterion simply based on the frequency of pixels of each class. Furthermore, the class weights manually specified are fixed throughout the entire training process, however the weights in our method are dynamically adjusted to obtain an optimized network. Therefore, the weights considered by our IAL are superior to the manually specified weights for all the classes.

In fact, our proposed loss is the generalization of the cross-entropy loss. Specifically, from Eq. (4) we see that if we set all importance factors $f_t(\mathbf{X})+\alpha$ $(t = 1, 2, \cdots)$ to 1, our proposed IAL function will immediately degrade into the existing cross-entropy loss with all classes sharing the equal importance. As a consequence, the Eq. (4) will become $Loss = \mathbf{w}_{G_1}^T \mathbf{l}_{G_1} + \mathbf{w}_{G_2}^T \mathbf{l}_{G_2} + \mathbf{w}_{G_3}^T \mathbf{l}_{G_3}$ which is identical to the expression yielded by the cross-entropy loss.

B. Forward and Backward Propagation Rules

In this section, we present the formation of the proposed loss function, and then deduce its related forward and backward propagation rules.

Suppose we have totally C classes that are grouped into g groups $G = \{G_1, G_2, \cdots, G_g\}$ which satisfy $G_i \neq \varnothing$ and $G_i \cap G_j = \varnothing$. For these g groups, their cross-entropy losses and corresponding weights avoiding class imbalance are $\{\mathbf{l}_{G_1}, \mathbf{l}_{G_2}, \cdots, \mathbf{l}_{G_g}\}$ and $\{\mathbf{w}_{G_1}, \mathbf{w}_{G_2}, \cdots, \mathbf{w}_{G_g}\}$, respectively. According to the above description, the *forward propagation* rule of the proposed loss function is

$$Q_1 = (f_1(\mathbf{X}) + \alpha)(\mathbf{w}_{G_2}^T \mathbf{l}_{G_2} + Q_2), \tag{5}$$

$$Q_2 = (f_2(\mathbf{X}) + \alpha)(\mathbf{w}_{G_3}^T \mathbf{l}_{G_3} + Q_3), \tag{6}$$

$$\cdots\cdots$$

$$Q_t = (f_t(\mathbf{X}) + \alpha)(\mathbf{w}_{G_{t+1}}^T \mathbf{l}_{G_{t+1}} + Q_{t+1}), \tag{7}$$

where $Q_{t+1} = (f_{t+1}(\mathbf{X}) + \alpha)(\mathbf{w}_{G_{t+2}}^T \mathbf{l}_{G_{t+2}})$ corresponds to the most important group. Therefore, the compact formation of the forward propagation rule regarding our IAL is

$$IAL = \mathbf{w}_{G_1}^T \mathbf{l}_{G_1} + Q_1. \tag{8}$$

Note that the $Q_g (g = 1, 2, \ldots, t+1)$ in the above equations are intermediate variables. In Eq. (8), $\mathbf{w}_{G_1}^T \mathbf{l}_{G_1}$ represents the cross-entropy loss value incurred by Group G_1. Q_1 is computed by Eq. (5), which is sequentially calculated by all the importance-aware losses corresponding to the rest groups $\{\mathbf{l}_{G_2}, \cdots, \mathbf{l}_{G_g}\}$ (see Eqs. (5) $\sim$ (7)). Specially, in Eq. (5), the $\mathbf{w}_{G_2}^T \mathbf{l}_{G_2}$ indicates the cross-entropy loss of Group G_2. The $\mathbf{w}_{G_2}^T \mathbf{l}_{G_2} + Q_2$ multiplied by $f_1(\mathbf{X}) + \alpha$ shows that Group G_2 is more important than Group G_1. The rationales of Eqs. (6) and (7) are similar to Eq. (5).

As a consequence, the *backward propagation* rules of IAL corresponding to Eqs. (8) and (7) are

$$\frac{\partial IAL}{\partial X} = \mathbf{w}_{G_1}^T * \frac{\partial \mathbf{l}_{G_1}}{\partial X} + \frac{\partial Q_1}{\partial X}, \tag{9}$$

$$\frac{\partial Q_t}{\partial X} = \frac{\partial f_t(\mathbf{X})}{\partial X}(\mathbf{w}_{G_{t+1}}^T \mathbf{l}_{G_{t+1}} + Q_{t+1}) + (f_t(\mathbf{X}) + \alpha)(\mathbf{w}_{G_{t+1}}^T * \frac{\partial \mathbf{l}_{G_{t+1}}}{\partial X} + \frac{\partial Q_{t+1}}{\partial X}). \tag{10}$$

where

$$\frac{\partial f_t(\mathbf{X})}{\partial X} = [(\mathbf{M}_t + \lambda_t \mathbf{E}) \odot (\mathbf{X} - \mathbf{M}_t) \odot \mathbb{I}\{\mathbf{M}_t \neq \text{“x”}\}] * \frac{\partial \mathbf{X}}{\partial X}. \tag{11}$$

By denoting

$$(\frac{\partial \mathbf{X}}{\partial X})_{:,i,j} = [0, 0, \cdots, \frac{\partial \mathbf{X}_{i,j}}{\partial X_{c,i,j}}, \cdots, 0, 0]^T \tag{12}$$

and

$$\mathbf{A} = (\mathbf{M}_t + \lambda_t \mathbf{E}) \odot (\mathbf{X} - \mathbf{M}_t) \odot \mathbb{I}\{\mathbf{M}_t \neq \text{“x”}\}, \tag{13}$$

we have

$$(\frac{\partial f_t(\mathbf{X})}{\partial X})_{:,i,j} = (\mathbf{A} * \frac{\partial \mathbf{X}}{\partial X})_{:,i,j} = \mathbf{A}_{i,j} (\frac{\partial \mathbf{X}}{\partial X})_{:,i,j}, \tag{14}$$

where $c = \mathbf{Y}_{i,j}$.

For $\frac{\partial \mathbf{l}_{G_t}}{\partial X}$, if the (i, j)-th pixel belongs to the class $(G_t)_r$ (i.e., the r-th class in Group G_t), and its corresponding weight is $(\mathbf{w}_{G_t})_r$, we obtain

$$\mathbf{h}_c = (\frac{\partial \mathbf{l}_{G_t}}{\partial X})_{c,i,j} = \begin{cases} \dfrac{exp(X_{c,i,j})}{\sum_{k=1}^{C} exp(X_{k,i,j})}, & \text{if } c \neq \mathbf{Y}_{i,j}; \\ \dfrac{exp(X_{c,i,j})}{\sum_{k=1}^{C} exp(X_{k,i,j})} - 1, & \text{if } c = \mathbf{Y}_{i,j}, \end{cases} \tag{15}$$

and then $(\mathbf{w}_{G_t}^T * \frac{\partial \mathbf{l}_{G_t}}{\partial X})_{:,i,j}$ is represented by

$$(\mathbf{w}_{G_t}^T * \frac{\partial \mathbf{l}_{G_t}}{\partial X})_{:,i,j} = (\mathbf{w}_{G_t})_r [\mathbf{h}_1, \cdots, \mathbf{h}_{c-1}, \mathbf{h}_c, \mathbf{h}_{c+1}, \cdots, \mathbf{h}_C]^T. \tag{16}$$

By using the forward-backward propagation rules in Eqs. (5)$\sim$(8) and Eqs. (9)$\sim$(10), the proposed IAL can be embedded to various deep neural networks, which will be detailed in the next section.

C. Deep Neural Networks

To verify the effectiveness of our proposed importance-aware loss (IAL), we apply IAL to four existing deep neural networks, i.e., FCN [6], SegNet [7], ENet [8], and ERFNet [11] to deal with SS problem. The configurations of these adopted neural networks are illustrated in Fig. 5.

FCN [6] has a similar architecture with VGG16 [43] network. As depicted in Fig. 5(a), the numbers of the feature maps in the neural network are 64, 128, 256, 512, 512, 4096, 4096, and 20, respectively. To obtain pixel-wise dense prediction, the upsampling layers are utilized, which enlarges the prediction with small spatial resolution to the same size as the input image. Here the last layer contains 20 feature maps as there are totally 20 classes in the investigated situation. Additionally, Long *et al.* proposed to combine the predictions from the Pool4 layer and Pool3 layer to improve the spatial details. As illustrated in Fig. 5(a), the outputs of the Pool3, Pool4, and Conv7 layers are summed up and then serve as the input of the upsampling layer. Finally, the upsampling layer is followed by a softmax classifier for pixel-wise prediction.

The setting of SegNet [7] is illustrated in Fig. 5(b), SegNet follows a typical encoder-decoder architecture. The encoder

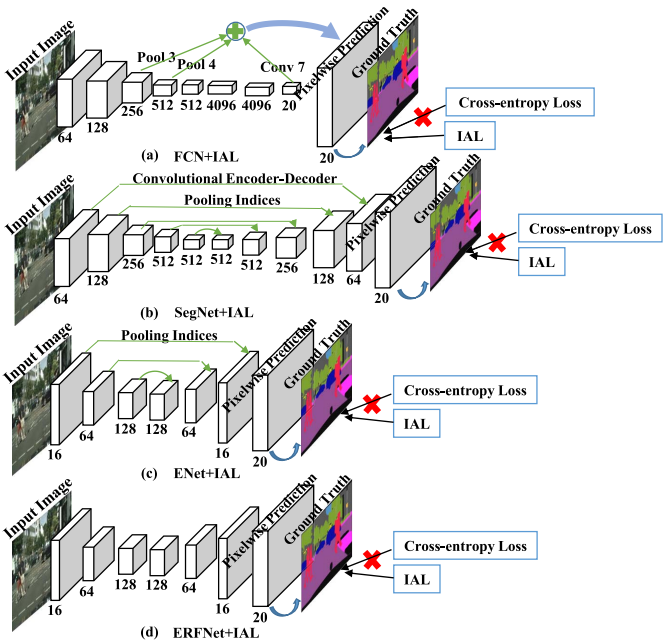


Fig. 5. The structures of four representative deep neural networks for SS problem. (a) FCN follows the main architecture of VGG16 where the numbers of the feature maps in the neural network are 64, 128, 256, 512, 512, 4096, 4096, and 20, respectively. (b) SegNet consists of an encoder and a decoder followed by a softmax classifier for pixel-wise classification. (c) ENet also follows an encoder-decoder style and the numbers of the features maps in the neural network are 16, 64, 128, 128, 64, and 16, respectively. (d) The architecture of ERFNet is the same as that of ENet, but ERFNet adopted deconvolutions instead of max-unpooling operations for upsampling, so there are no pooling indices in the ERFNet. In our experiments, we simply replace the original cross-entropy loss with the proposed IAL to introduce the class importance.

network consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network [43], and each encoder layer has a corresponding decoder layer. The difference between the encoder layers and the first 13 convolutional layers of VGG16 is that an additional module of batch normalization [44] is inserted after the convolution layers. To perform non-linear upsampling, SegNet stores the max-pooling indices to provide the guidance for upsampling layers in decoder part. Here the max-pooling indices are the locations of the maximum feature value in each pooling window of encoder part.

Different from above two networks that aim to solve general SS problem in natural images, ENet [8] is specifically designed for handling autonomous driving. Its structure is shown in Fig. 5(c) which indicates that ENet is also designed under the encoder-decoder style. Compared to the architectures of FCN and SegNet, the number of feature maps of ENet are drastically decreased, namely 16, 64, 128, 128, 64, and 16 for achieving real-time semantic segmentation. Specifically, in its input layer and first block with 16 feature maps, the input images are passed to two branches where the first branch is simply one max-pooling layer made up of non-overlapping 2×2 windows, and the second branch conducts normal convolution operations with 13 filters. Then the output feature maps of the two branches are concatenated. Sequentially, each of the rest blocks has many bottlenecks [8] which can be viewed as a special case of ResNet [40].

ERFNet [11] is a recent network which also aims at tackling autonomous driving. Its architecture is very similar to ENet (see Fig. 5(d)), so the numbers of its feature maps are also 16, 64, 128, 128, 64, and 16. Compared to ENet, however, the ERFNet introduced the non-bottleneck-1D layers to retain the learning capacity while maximizing efficiency of residual layers, and used the initial block of ENet in all downsampling layers. Besides, ERFNet incorporated Dropout in all non-bottleneck-1D layers for regularization, and adopted deconvolutions instead of max-unpooling operations for upsampling.

All above networks classify the image pixels with the cross-entropy loss after the last layer, which is incapable of differentiating the importance of objects as explained in the introduction. Therefore, we replace the original cross-entropy loss with our proposed IAL and keep other network configurations unchanged, so that the networks can pay more attention to important classes than the trivial ones.

IV. Experiments

In this section, we firstly introduce our experimental settings such as the adopted deep networks, employed datasets, parametric configurations, and evaluation metric. Then we report the experimental results of compared settings on two typical autonomous driving datasets, i.e., CamVid [12] and Cityscapes [13]. Finally, we analyze parametric sensitivity and also compare the training time between the IAL-based SS models and the original networks with cross-entropy loss.

A. Experimental Settings

As mentioned in Section III-C, FCN and SegNet are popular deep methods for conventional SS, and ENet and ERFNet are recently proposed deep networks specifically for autonomous driving application. As depicted in Fig. 5, the cross-entropy loss adopted by these models will be replaced by our IAL during the training stage, and we term them as "ENet+IAL", "SegNet+IAL", "FCN+IAL", and "ERFNet+IAL", respectively. Meanwhile, we also introduce "ENet+Uni", "SegNet+Uni", "FCN+Uni", and "ERFNet+Uni" for comparison, in which "Uni" describes uniform weights for all classes in cross-entropy loss. To achieve fair comparisons, the ENet, SegNet, FCN, ERFNet, ENet+Uni, SegNet+Uni, FCN+Uni, ERFNet+Uni as well as our ENet+IAL, SegNet+IAL, FCN+IAL and ERFNet+IAL are implemented by using the identical Torch 7 deep learning platform, so their results are directly comparable.

We use the CamVid dataset [12] mentioned in Section III-A and a recent Cityscapes [13] dataset for our experiments. CamVid contains 367 training images, 101 validation images, and 233 test images. The resolution of images in this dataset is 960×720 which will be downsampled to 480×360 for accelerating the training stage of SS models. Cityscapes is also a high-quality dataset for semantic scene understanding captured from the view of cockpit, which contains 2975 color training images, 500 validation images, and 1525 test images. The resolution of all images is 2048×1024. Similar to the operation on CamVid dataset, for the models ENet, ENet+Uni, ENet+IAL,

ERFNet, ERFNet+Uni, and ERFNet+IAL, we will downsample these images by 2 times before training, and for the rest models, the resolution of these images will be scaled into 512×256. In Cityscapes dataset, we pick up 19 the most frequently occurred classes from the original 35 classes based on the official evaluation metrics [13], and their importance groupings from trivial to important are
Group 1 = {Sky, Building, Vegetation, Terrain, Wall};
Group 2 = {Pole, Road, Sidewalk, Fence};
Group 3 = {Traffic sign, Traffic light, Car, Truck, Bus, Train, Motorcycle, Person, Rider, Bicycle};

For a certain deep neural work, the Adam optimization algorithm [45] is employed for model training, as this algorithm allows the training process to converge very quickly. We start with the learning rate 10^{-3} and gradually decrease it by a factor of 0.1 after every 100 epochs. Besides, we fix the mini-batch size to 8 images, set the momentum to 0.9, and fix the weight decay for ℓ_2 regularization to 5×10^{-4}. The iteration number is 300 for all models trained on two datasets. In addition, ENet, ENet+Uni, ENet+IAL, SegNet, SegNet+Uni, SegNet+IAL, ERFNet, ERFNet+Uni, and ERFNet+IAL are performed in two stages: first we only train the part of encoder to map an input image to a downsampled label; then we append corresponding decoder to the trained encoder to perform upsampling and train the overall network followed by a pixel-wise classifier. For the models FCN, FCN+Uni, and FCN+IAL, we train them all at once via an end-to-end fashion on CamVid and Cityscapes datasets, respectively.

Note that we adopt the weights defined by Eq. (2) in all above SS models for dealing with class imbalance. Besides, we follow [8] and use the intersection-over-union (IoU) and class accuracy (ClassAcc) to evaluate the performance of compared methods on different datasets. The IoU is defined as

$$\text{IoU} = \frac{\text{TP}}{\text{TP + FP + FN}}, \tag{17}$$

where TP, FP, and FN denote the numbers of true positive, false positive, and false negative pixels, respectively. Furthermore, in order to show the overall performance, we use the metric of Mean IoU which is the average IoU over all classes. The class accuracy is defined as

$$\text{ClassAcc} = \frac{\text{TP}}{\text{TP + FN}}, \tag{18}$$

where TP and FN have been explained above. We use the metric of class average accuracy (ClassAvg) to express the overall effectiveness, which is the average ClassAcc over all classes. The main difference between IoU and ClassAcc is that there is an additional FP in the denominator of Eq. (17). Additionally, pixels labeled as 'Void' or 'Unlabelled' will be ignored when one computes the loss in training an SS model and computes the ClassAcc, ClassAvg, and Mean IoU scores in testing stage.

B. Quantitative and Qualitative Analysis

In this section, we compare the performances of ENet, ENet+Uni, ENet+IAL, and SegNet, SegNet+Uni, SegNet+IAL, and FCN, FCN+Uni, FCN+IAL, and ERFNet, ERFNet+Uni, ERFNet+IAL under the above experimental settings. Specifically, after training the above deep neural networks, we observe their ClassAcc, ClassAvg, and Mean IoU scores on test sets of CamVid and on validation sets of Cityscapes. The experimental results of compared methods on the investigated classes of the two datasets are shown in Tables I∼II and Tables III∼IV, respectively.

TABLE I

THE COMPARISON RESULTS (%) OF VARIOUS METHODS ON THE GROUPS 1 AND 2 OF CAMVID DATASET. THE BEST RECORDS AMONG THE ORIGINAL CNN (I.E., ENET/SEGNET/FCN/ERFNET), CNN+CROSS-ENTROPY LOSS WITH UNIFORM WEIGHTS, AND CNN+IAL ARE MARKED IN BOLD

	Group 1			Group 2			
	Sky	Building	Tree	Column/Pole	Road	Sidewalk	Fence
ENet	95.1	74.7	77.8	35.4	95.1	86.7	51.7
ENet+Uni	**95.7**	**84.2**	**80.2**	22.6	**97.4**	85.7	30.3
ENet+IAL	93.3	75.7	73.8	**59.4**	95.4	**91.9**	**62.0**
SegNet	93.3	74.7	**87.4**	**48.8**	93.9	**89.3**	**46.5**
SegNet+Uni	92.4	**88.3**	82.6	9.6	94.2	85.4	20.7
SegNet+IAL	**93.4**	76.2	73.7	44.4	**95.4**	85.9	37.0
FCN	91.5	83.7	80.9	17.2	91.9	87.4	43.9
FCN+Uni	86.9	**88.1**	74.9	19.6	**95.7**	77.1	26.4
FCN+IAL	**92.3**	70.0	**81.2**	**24.4**	91.0	**89.7**	**44.0**
ERFNet	94.6	80.7	**87.6**	44.2	96.4	**92.7**	**46.3**
ERFNet+Uni	**96.3**	**88.7**	80.1	29.3	**97.4**	87.8	33.7
ERFNet+IAL	95.6	70.0	78.9	**53.6**	95.5	92.1	44.6

From the results shown in Tables I and II, we find that by embedding our IAL to the adopted deep models, the ClassAcc values of the investigated important classes like sign/symbol, pedestrian, and bicyclist can be significantly improved when compared with the results of the cross-entropy (i.e., with uniform weights and class balance) loss based deep models. Not surprisingly, some ClassAcc values on unimportant classes such as sky, building, and tree weakly drop because they are trained with small weights by our IAL. However, if we compute the ClassAvg averaged over all classes for all compared methods (see the "ClassAvg" column in Table II), we see that these networks equipped with IAL are still able to achieve better performance than the original networks with the cross-entropy loss, and the improvements are 8.9 for ENet, 3.0 for SegNet, 7.1 for FCN, and 3.5 for ERFNet, respectively. Meanwhile, all ClassAvg values of CNN (i.e., ENet, SegNet, FCN, and ERFNet)+IAL are largely better than that of CNN+Uni.

From the results in Table III and Table IV, we observe that the important classes in Group 3 are segmented with very high ClassAcc by ENet+IAL, SegNet+IAL, FCN+IAL, and ERFNet+IAL such as traffic sign, traffic light, person, and bicycle. Specifically, the ClassAcc values of the traffic sign, traffic light, person, and bicycle generated by ENet+IAL are as high as 86.0, 79.0, 89.3, and 85.9, which are significantly better than the results of ENet+Uni that are 59.9, 27.9, 83.0 and 66.8, as well as the results of ENet that are 79.4, 71.9, 88.5, and 84.7, respectively. Similar to the advantages of ENet+IAL to ENet and ENet+Uni, the improvements of SegNet+IAL over SegNet+Uni are 50.3, 44.7, 17.6, and 27.1, and the performance gain of SegNet+IAL over SegNet are 13.3, 30.9, 11.4, and 5.3. For other important classes such as car, train,

TABLE II

THE COMPARISON RESULTS (%) OF VARIOUS METHODS ON THE GROUP 3 OF CAMVID DATASET. THE BEST RECORDS AMONG THE ORIGINAL CNN (I.E., ENET/SEGNET/FCN/ERFNET), CNN+CROSS-ENTROPY LOSS WITH UNIFORM WEIGHTS, AND CNN+IAL ARE MARKED IN BOLD

	Group 3					
	Sign/symbol	Car	Pedestrian	Bicyclist	ClassAvg	Mean IoU
ENet	51.0	82.4	67.2	34.1	68.3	51.3
ENet+Uni	41.7	79.1	61.6	15.2	63.1	52.5
ENet+IAL	**63.9**	**88.7**	**80.0**	**65.2**	**77.2**	**57.6**
SegNet	26.6	**82.1**	38.5	42.3	65.8	**51.2**
SegNet+Uni	0.7	78.2	25.5	0.5	52.6	44.8
SegNet+IAL	**43.7**	80.8	**75.7**	**50.6**	**68.8**	51.0
FCN	32.8	84.3	36.3	23.6	61.2	**49.6**
FCN+Uni	30.1	83.8	48.0	27.3	59.8	49.1
FCN+IAL	**59.8**	**84.7**	**72.1**	**42.1**	**68.3**	48.8
ERFNet	44.8	79.6	65.2	52.4	71.3	**57.5**
ERFNet+Uni	20.0	79.0	37.7	37.9	62.5	54.0
ERFNet+IAL	**66.6**	**89.7**	**78.8**	**57.6**	**74.8**	54.5

TABLE III

THE COMPARISON RESULTS (%) OF VARIOUS METHODS ON THE GROUPS 1 AND 2 OF CITYSCAPES DATASET. THE BEST RECORDS AMONG THE ORIGINAL CNN (I.E., ENET/SEGNET/FCN/ERFNET), CNN+CROSS-ENTROPY LOSS WITH UNIFORM WEIGHTS, AND CNN+IAL ARE MARKED IN BOLD

	Group 1					Group 2			
	Sky	Building	Vegetation	Terrain	Wall	Pole	Road	Sidewalk	Fence
ENet	**97.7**	90.4	92.0	**79.6**	**60.4**	**72.6**	96.9	**88.5**	**64.1**
ENet+Uni	94.8	**93.9**	**94.0**	66.3	35.1	50.7	**98.0**	84.2	59.6
ENet+IAL	93.9	89.4	86.9	68.4	34.9	62.0	95.6	88.2	63.7
SegNet	**97.2**	83.9	91.3	54.9	**43.1**	**58.9**	94.7	**85.5**	38.5
SegNet+Uni	93.0	**92.4**	**91.5**	**59.3**	12.0	38.7	**97.6**	78.4	45.9
SegNet+IAL	94.7	71.3	80.8	41.4	3.7	56.0	94.5	83.9	**54.7**
FCN	94.0	83.4	**94.4**	**61.9**	**33.7**	31.0	95.3	**84.0**	35.5
FCN+Uni	94.3	**91.7**	93.6	43.3	17.0	28.8	**98.1**	77.6	25.0
FCN+IAL	**94.8**	84.1	87.0	49.3	5.1	**41.2**	96.4	81.6	**42.9**
ERFNet	**98.0**	94.0	95.4	**75.8**	**68.5**	73.6	98.1	**91.7**	**67.1**
ERFNet+Uni	97.6	**95.2**	**95.9**	73.0	60.2	66.0	**98.8**	90.2	60.8
ERFNet+IAL	97.3	90.7	92.7	67.1	52.6	**73.7**	97.9	91.2	61.6

TABLE IV

THE COMPARISON RESULTS (%) OF VARIOUS METHODS ON THE GROUP 3 OF CITYSCAPES DATASET. THE BEST RECORDS AMONG THE ORIGINAL CNN (I.E., ENET/SEGNET/FCN/ERFNET), CNN+CROSS-ENTROPY LOSS WITH UNIFORM WEIGHTS, AND CNN+IAL ARE MARKED IN BOLD

	Group 3											
	Traffic Sign	Traffic Light	Car	Truck	Bus	Train	Motorcycle	Person	Rider	Bicycle	ClassAvg	Mean IoU
ENet	79.4	71.9	94.7	70.4	75.0	**59.7**	40.9	88.5	59.7	84.7	**77.2**	**60.6**
ENet+Uni	59.9	27.9	93.5	0.0	60.8	0.0	0.0	83.0	0.0	66.8	56.2	47.1
ENet+IAL	**86.0**	**79.0**	**95.6**	**87.8**	**85.6**	43.1	**42.0**	**89.3**	**60.4**	**85.9**	75.7	56.3
SegNet	52.6	21.6	**92.8**	45.9	41.9	17.6	3.3	70.2	38.3	70.3	58.0	**43.9**
SegNet+Uni	15.6	7.8	92.0	13.8	1.0	52.8	2.6	64.0	0.0	48.5	47.7	38.3
SegNet+IAL	**65.9**	**52.5**	92.2	**60.4**	**51.1**	**67.8**	**19.7**	**81.6**	**49.0**	**75.6**	**63.0**	40.2
FCN	47.6	13.4	93.3	1.3	**29.1**	1.0	0.5	69.3	3.6	63.8	49.3	39.2
FCN+Uni	44.0	11.1	91.2	2.3	3.4	12.3	0.3	63.2	0.1	57.8	45.0	38.6
FCN+IAL	**59.5**	**47.3**	**94.5**	**28.7**	12.3	**44.9**	**20.3**	**80.5**	**26.0**	**69.6**	**56.1**	**41.3**
ERFNet	81.0	79.0	96.6	**74.9**	85.1	82.0	57.0	89.6	**69.6**	86.8	**82.3**	**70.0**
ERFNet+Uni	73.8	64.8	96.1	73.7	80.8	70.1	53.4	84.3	64.2	82.4	78.0	68.1
ERFNet+IAL	**84.9**	**85.2**	**97.7**	73.5	**88.4**	**82.7**	**57.2**	**92.9**	69.5	**88.1**	81.3	65.2

and motorcycle, the ClassAcc values of FCN+IAL are also higher than that of FCN and FCN+Uni. For some unimportant classes in Group 1, the performances of the IAL-based models are inferior to the original models. However, they will not have large impact on safe-driving as explained above.

To intuitively present the effectiveness of our proposed loss function, we provide some representative segmentation results of ENet, ENet+Uni, ENet+IAL, SegNet, SegNet+Uni, SegNet+IAL, FCN, FCN+Uni, FCN+IAL, ERFNet, ERFNet+Uni, and ERFNet+IAL, correspondingly. The segmentation results here are originated from the test set of CamVid and the validation set of Cityscapes, which are illustrated in Figs. 6, 7 and Figs. 8, 9.

For the performance on CamVid dataset, Fig. 6 shows some representative segmentation results of the ENet, ENet+Uni, and ENet+IAL. Specifically, for the important classes with large size (see the rows of car, bicyclist, and sidewalk), we find that the interested regions segmented by the ENet+IAL are highly compact, and the shapes of the segmented objects are also more close to that of the ground truth. Other critical objects such as sign/symbol, pedestrian, and pole are quite small in the image, so they are very likely to bring about

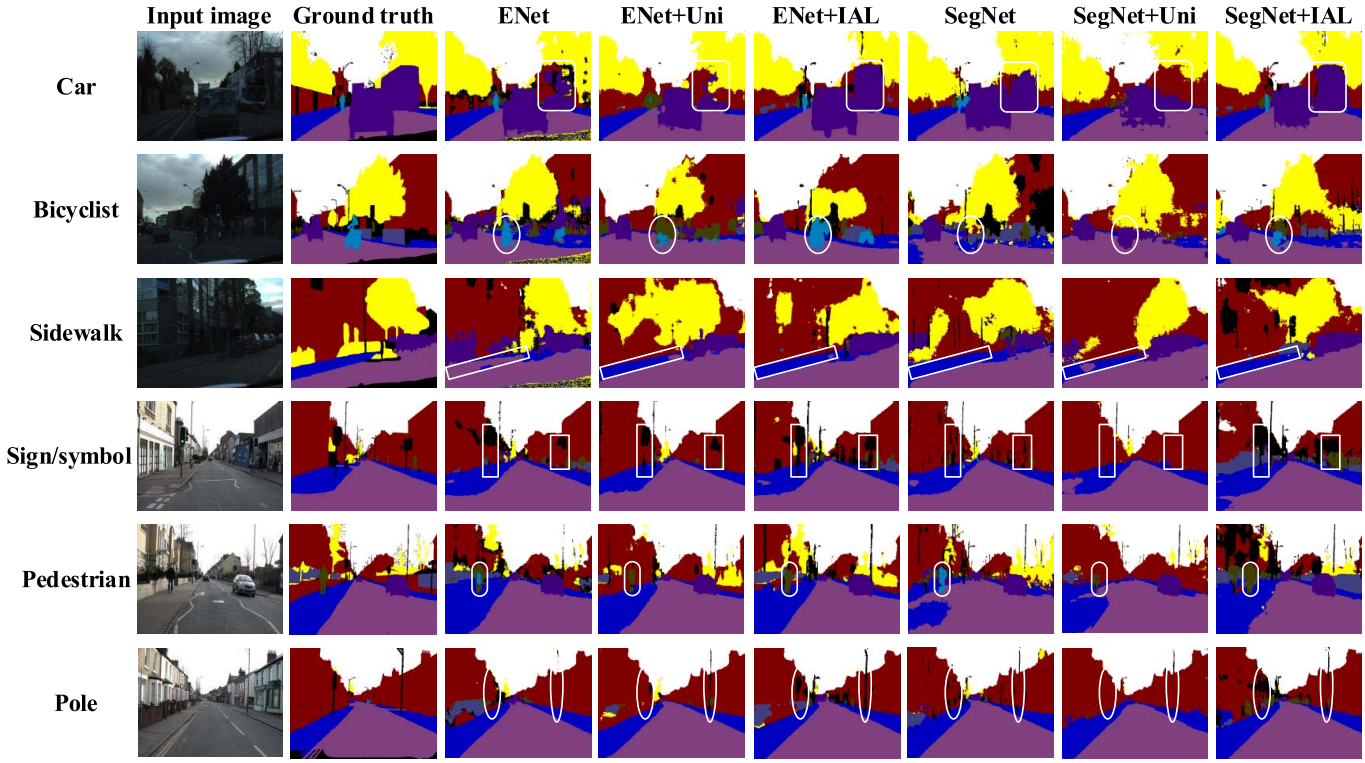


Fig. 6. Representative segmentation results of ENet, ENet+Uni, ENet+IAL, and SegNet, SegNet+Uni, SegNet+IAL on important classes of CamVid dataset. **Best viewed in color.**

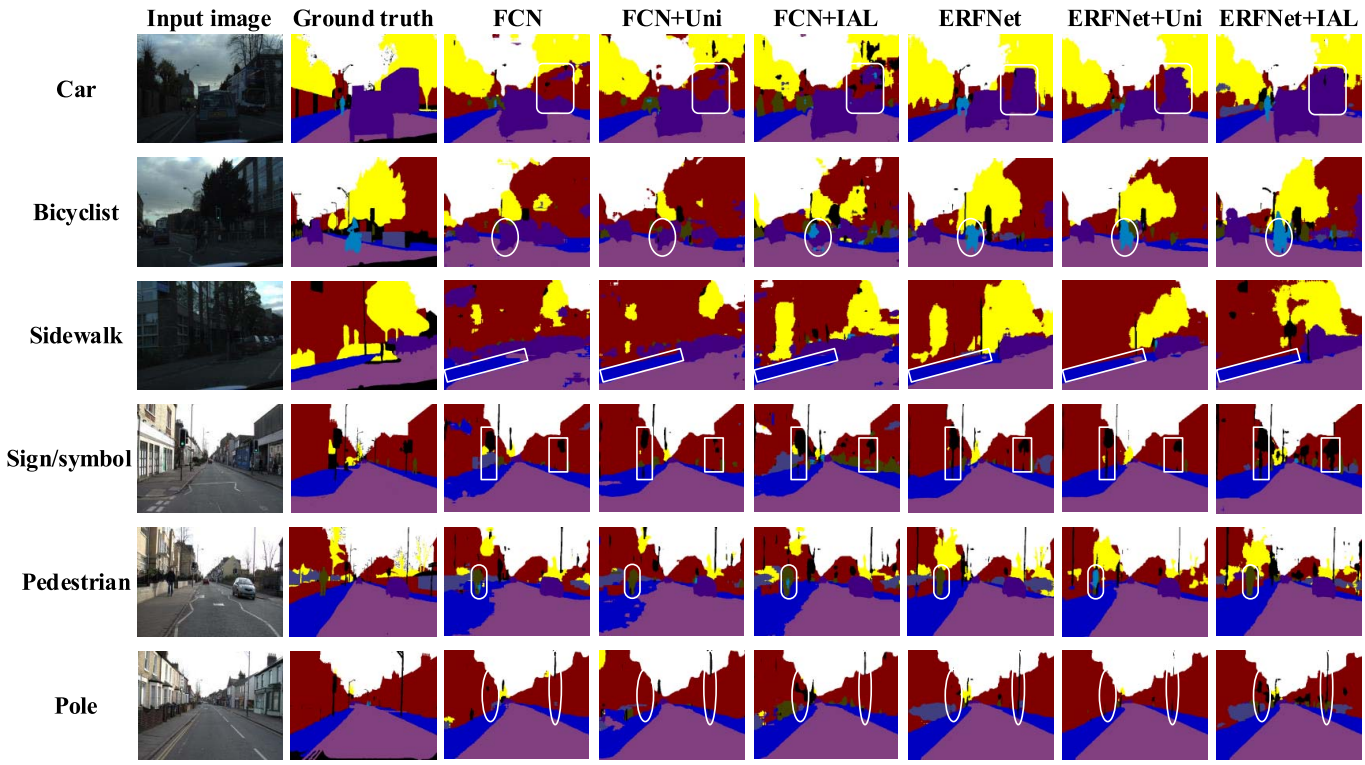


Fig. 7. Representative segmentation results of FCN, FCN+Uni, FCN+IAL, and ERFNet, ERFNet+Uni, ERFNet+IAL on important classes of CamVid dataset. **Best viewed in color.**

imperfect results by using the ENet+Uni and original ENet as no extra attention have been paid to above objects by the two nets. However, our ENet+IAL successfully picks them up and achieves accurate segmentation results. Therefore, IAL is effective in emphasizing the small but critical targets, and thus is useful for SS tasks.

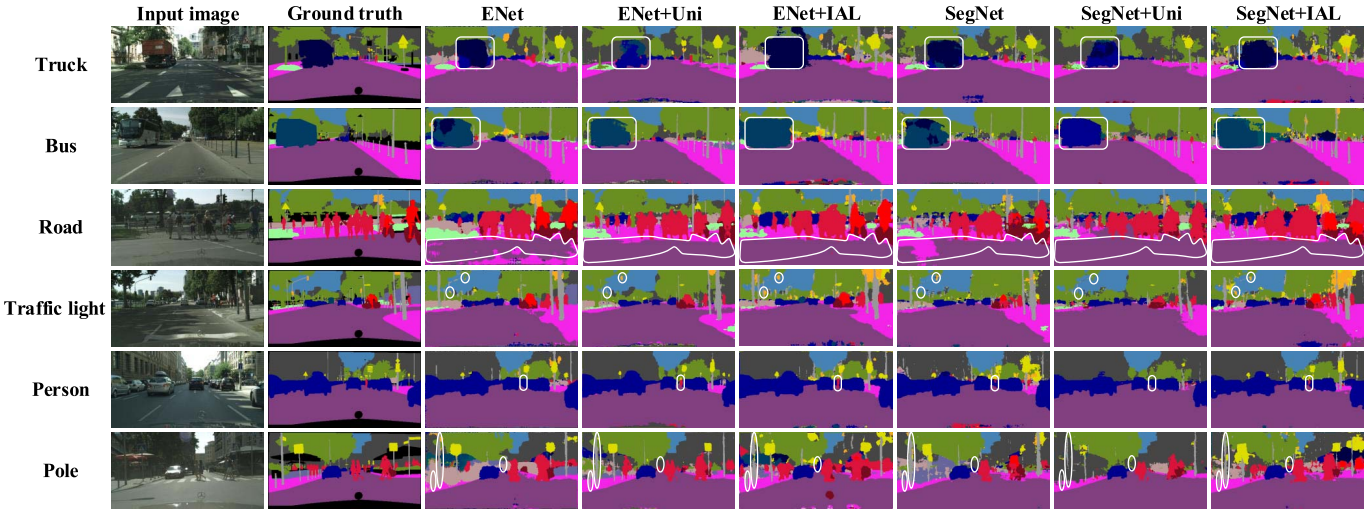


Fig. 8. Representative segmentation results of ENet, ENet+Uni, ENet+IAL, and SegNet, SegNet+Uni, SegNet+IAL on important classes of Cityscapes dataset. **Best viewed in color.**

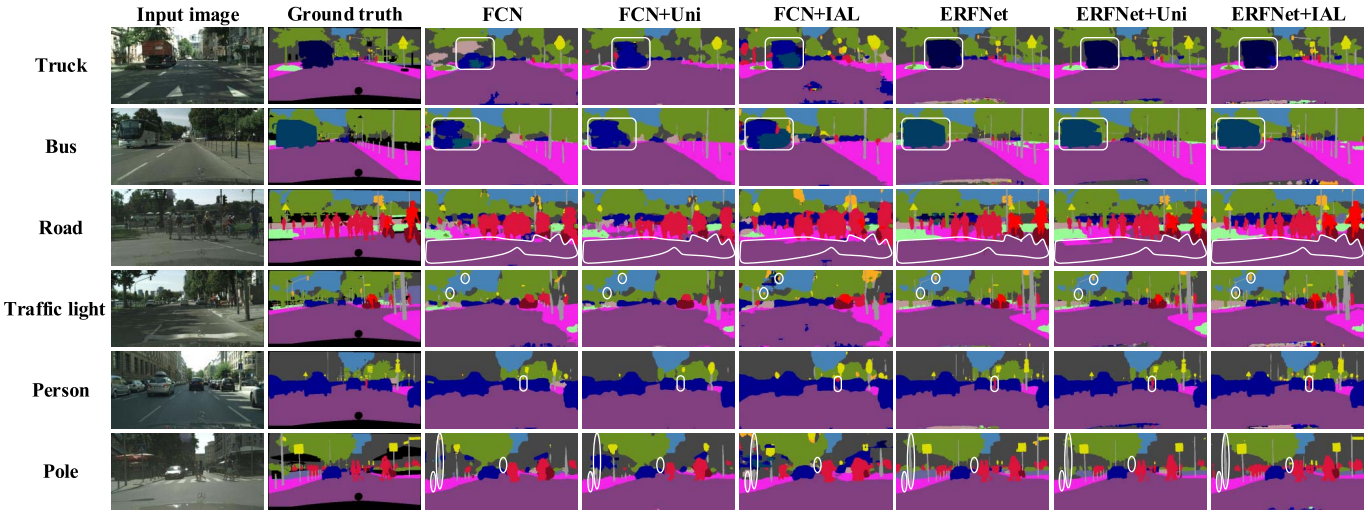


Fig. 9. Representative segmentation results of FCN, FCN+Uni, FCN+IAL, and ERFNet, ERFNet+Uni, ERFNet+IAL on important classes of Cityscapes dataset. **Best viewed in color.**

To demonstrate the effectiveness of IAL on Cityscapes dataset, Fig. 8 also depicts some typical segmentation results of the ENet, ENet+Uni, and ENet+IAL. We see that for the important objects dominated in an image (e.g., truck, bus, and road), the regions segmented by the ENet+IAL are very coherent and most pixels of the corresponding regions are correctly classified. Comparatively, the original ENet yields much worse outputs than the ENet+IAL such as the incomplete truck, bus, and road. For the important objects with small size (e.g., traffic light, person, and pole), the ENet+IAL also generates more similar segmentation results to ground truth than the ENet. For example, the traffic light indicated by a circle is rather small, and it is nearly missed by the ENet+Uni and ENet. However, our ENet+IAL successfully picks it up and renders accurate segmentation. The pole in the last row is so tiny that it is completely misclassified by the ENet. In contrast, the ENet+IAL clearly identifies the pole from the background as indicated by the white circle. Similarly, the comparisons among CNN (i.e., SegNet, FCN, and ERFNet), CNN+Uni, and CNN+IAL also reveal the similar results, which are illustrated in Fig. 7 and Fig. 9, respectively.

According to above quantitative and qualitative results, we conclude that the proposed hierarchical importance-aware loss can improve the segmentation quality of the important objects with a large margin in terms of ClassAcc. Therefore, IAL is quite suitable for the application of autonomous driving.

C. Parametric Sensitivity

Our IAL contains two critical parameters α and λ, and in this section we will show how the variations of these two parameters influence the final results. As described in the Section III-A, α is a tuning parameter that should be larger than 1 to guarantee that the important classes obtain higher weights. Besides, the parameter λ encourages the IAL-based SS models to pay more attention to the classification of important objects. In order to accelerate the training stage of SS models, the size of images in CamVid dataset is downsampled by 2 times. Meanwhile, we rescaled the image resolution of Cityscapes dataset into 512×256. After obtaining trained models, we investigate the ClassAvg values on the test set of CamVid and the validation set of Cityscapes. The ClassAvg values under different selections of α and λ achieved by

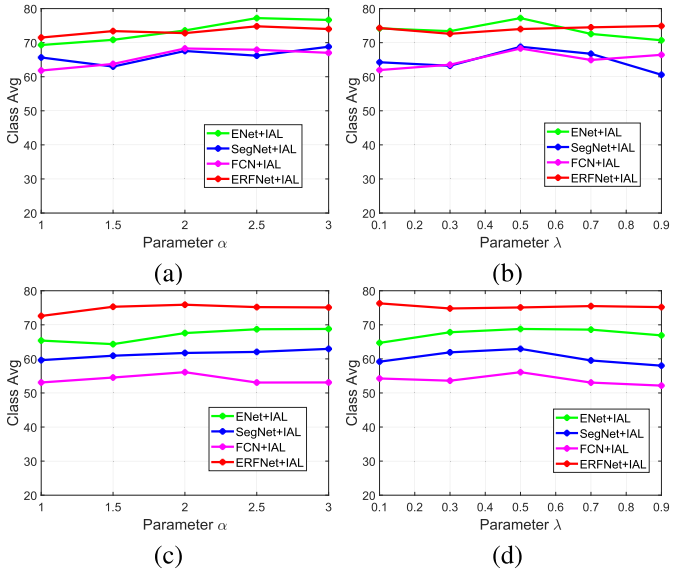


Fig. 10. Sensitivity analysis for the parameters α and λ on the test set of CamVid and the validation set of Cityscapes. (a) and (c) respectively represent the ClassAvg values by varying the parameters of $\alpha = \{1, 1.5, 2, 2.5, 3\}$ with the parameter λ fixed to 0.5 on CamVid and Cityscapes datasets. (b) and (d) plot the ClassAvg values of ENet+IAL, SegNet+IAL, FCN+IAL, and ERFNet+IAL when $\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ with α fixed to 2.5 on CamVid and Cityscapes datasets, respectively.

ENet+IAL, SegNet+nIAL, FCN+IAL, and ERFNet+IAL are reported in Fig. 10. We find that the outputs of these models are generally stable under $\alpha \in [1, 3]$ and $\lambda \in [0.1, 0.9]$, which means that the segmentation results will not be significantly influenced by the choices of these two parameters. Therefore, the α and λ in our method can be easily tuned for practical use.

D. Comparison of Training Time

This section compares the training time of ENet vs. ENet+IAL, SegNet vs. SegNet+IAL, FCN vs. FCN+IAL, and ERFNet vs. ERFNet+IAL on CamVid and Cityscapes datasets. Here we do not include the training time of ENet+Uni, SegNet+Uni, FCN+Uni, and ERFNet+Uni because their structures are identical to those of ENet, SegNet, FCN, and ERFNet correspondingly, so deploying the uniform class weights will not influence the training time of ENet, SegNet, FCN and ERFNet. We aim to study whether the incorporation of IAL will increase the training time. All SS models are trained on two K80 GPU, and the mini-batch size and iteration number are set to 8 and 300, respectively. From the results provided in Table V, we see that the IAL incurs very little extra time cost when compared with the network equipped with the cross-entropy loss. Meanwhile, Section IV-B reveals that the IAL based networks are able to significantly improve the segmentation performance of the cross-entropy loss based counterparts. Therefore, the proposed IAL is both effective and efficient.

V. Conclusion

Semantic segmentation in driving environment is quite different from its traditional implementations for general natural images, as various classes might have different levels of importance for driving safety. Based on this argument, this paper proposes a novel hierarchical importance-aware loss (IAL) so that the object classes with different importance are adaptively allocated different weights during the model training stage. As a result, the objects that are critical for safe-driving can be segmented more accurately than the traditional SS methods as revealed by the experiments. Moreover, our loss function IAL is general in nature and can be easily combined with many other existing SS algorithms for various applications with the consideration of class importance.

TABLE V

Training Time of Various Deep Models on CamVid and Cityscapes Datasets (Unit: Hour)

	CamVid	Cityscapes
ENet	3.43	21.81
ENet+IAL	3.64	24.32
SegNet	16.00	100.49
SegNet+IAL	16.31	103.11
FCN	16.25	91.04
FCN+IAL	16.26	96.17
ERFNet	4.17	20.00
ERFNet+IAL	4.79	22.86

References

[1] S. Di, H. Zhang, C.-G. Li, X. Mei, D. Prokhorov, and H. Ling, "Cross-domain traffic scene understanding: A dense correspondence-based transfer learning approach," *IEEE Trans. Intell. Transp. Syst.* Mar. 2018, vol. 19, no. 3, pp. 745–757.

[2] M. Altun and M. Celenk, "Road scene content analysis for driver assistance and autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3398–3407, Dec. 2017.

[3] L. Xu, Y. Wang, H. Sun, J. Xin, and N. Zheng, "Design and implementation of driving control system for autonomous vehicle," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 22–28.

[4] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[5] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr, "What, where and how many? Combining object detectors and CRFs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 424–437.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 3431–3440.

[7] 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.

[8] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. (2016). "ENet: A deep neural network architecture for real-time semantic segmentation." [Online]. Available: https://arxiv.org/abs/1606.02147

[9] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, 2005.

[10] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2553–2561.

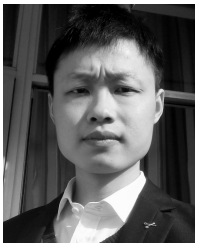
[11] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[12] 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.

[13] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3213–3223.

[14] B.-K. Chen, C. Gong, and J. Yang, "Importance-aware semantic segmentation for autonomous driving system," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 1504–1510.

[15] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 44–57.
[16] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 746–760.
[17] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[18] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
[19] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
[20] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, Nov. 2004.
[21] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2009, pp. 1–11.
[22] P. Kontschieder, S. R. Bulò, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2190–2197.
[23] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 708–721.
[24] Q. Wang and S. Y. Li, "Database of human segmented images and its application in boundary detection," *IET Image Process.*, vol. 6, no. 3, pp. 222–229, Apr. 2012.
[25] Q. Wang and Z. Wang, "A subjective method for image segmentation evaluation," in *Proc. 9th Asian Conf. Comput. Vis. Comput. Vis. (ACCV)*, Xi'an, China, 2009, pp. 53–64.
[26] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2759–2766.
[27] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Scene parsing with multiscale feature learning, purity trees, and optimal covers," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 575–582.
[28] D. Grangier, L. Bottou, and R. Collobert, "Deep convolutional networks for scene parsing," in *Proc. Int. Conf. Mach. Learn. (ICML) Workshops*, 2011, pp. 1–2.
[29] C. Gatta, A. Romero, and J. van de Veijer, "Unrolling loopy top-down semantic feedback in convolutional deep networks," in *Proc. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2014, pp. 504–511.
[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 357–361.
[31] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1529–1537.
[32] X. Shen *et al.*, "Automatic portrait segmentation for image stylization," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 93–102, 2016.
[33] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2015, pp. 2650–2658.
[34] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[35] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1495–1503.
[36] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 193–202.
[37] S. R. Bulò, G. Neuhold, and P. Kontschieder. (2017). "Loss max-pooling for semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1704.02966
[38] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. (2017). "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade." [Online]. Available: https://arxiv.org/abs/1704.01344
[39] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. (2016). "Full-resolution residual networks for semantic segmentation in street scenes." [Online]. Available: https://arxiv.org/abs/1611.08323
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[41] M. Treml *et al.*, "Speeding up semantic segmentation for autonomous driving," in *Proc. Neural Inf. Process. Syst. (NIPS) Workshops*, 2016, pp. 1–7.
[42] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1789–1794.
[43] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[44] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167
[45] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980



Bike Chen received the B.E. degree in information and computing science from Xuzhou University of Technology, China, in 2015. He is currently pursuing the M.S. degree on pattern recognition and intelligent systems with Nanjing University of Science and Technology, under the supervision of Prof. J. Yang and Prof. C. Gong. His current research interests include machine learning, robotics, and evolutionary computation.

Chen Gong received the B.E. degree from East China University of Science and Technology in 2010 and the dual Ph.D. degree from Shanghai Jiao Tong University and University of Technology Sydney in 2016 and 2017, under the supervision of Prof. J. Yang and Prof. D. Tao, respectively. He is currently a Full Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has authored over 30 technical papers at prominent journals and conferences, such as IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, IEEE T-CSVT, IEEE T-MM, CVPR, AAAI, and IJCAI. He received the National Scholarship from the Ministry of Education in 2013 and 2014, respectively, the Excellent Self-financed Overseas Student Scholarship from the China Scholarship Council in 2015, the IBM Excellent Student Scholarship in 2015, and the Excellent Doctoral Dissertation of Shanghai Jiao Tong University in 2017.

Jian Yang received the B.E. degree in mathematics from Xuzhou Normal University in 1995, the M.S. degree in applied mathematics from Changsha Railway University in 1998, and the Ph.D. degree in pattern recognition and intelligent systems from Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2002. From 2003 to 2007, he was a Post-Doctoral Fellow at University of Zaragoza, Hong Kong Polytechnic University, and New Jersey Institute of Technology, respectively. He is currently a Professor with the School of Computer Science and Engineering, NJUST. He has authored over 100 scientific papers on pattern recognition and computer vision. His research interests include pattern recognition, computer vision, and machine learning. He is a fellow of IAPR. He is currently an Associate Editor of *Pattern Recognition Letters* and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. His journal papers have been cited over 4000 times in the ISI Web of Science and 8000 times in the Web of Scholar Google.