Importance-Aware Semantic Segmentation for Autonomous Driving System

Bi-ke Chen, Chen Gong, Jian Yang
School of Computer Science and Engineering, Nanjing University of Science and Technology
Nanjing, 210094, China
{bikechen, chen.gong, csjyang}@njust.edu.cn

Abstract

Semantic Segmentation (SS) partitions an image into several coherent semantically meaningful parts, and classifies each part into one of the predetermined classes. In this paper, we argue that 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, pedestrians in the scene are much more important than sky 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. 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 deep neural networks for realizing SS in intelligent driving system. The experiments on CamVid and Cityscapes datasets reveal that by employing the proposed loss function, the existing deep learning models including FCN, SegNet and ENet are able to consistently obtain the improved segmentation results on the pre-defined important classes for safe-driving.

1 Introduction

Semantic Segmentation (SS) partitions an image into regions that represent meaningful objects, which serves as an important tool for the subsequent image analysis such as scene understanding. In recent years, autonomous driving system has gained much popularity, in which SS has played an important role in detecting obstacles and recognizing road conditions.

Apparently, good SS results in autonomous driving system will help it precisely understand the scene, and thus leading to safe decision-making and vehicle control. However, we argue that the SS for autonomous driving system is quite different from the conventional SS problems. For conventional 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. In contrast, the objects in the scene 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, grassland and sun.

In this sense, current SS methods are not suitable for dealing with autonomous driving problem. For example, the traditional works [Shotton *et al.*, 2008; Ladicky *et al.*, 2010; Gong *et al.*, 2015] based on handcrafted features and recent Deep Convolutional Neural Network (DCNN) based methods [Long *et al.*, 2015; Vijay Badrinarayanan and Cipolla, 2017; Paszke *et al.*, 2016] equally treat all the classes. As a result, they generate very low accuracy on segmenting the important objects as mentioned above.

From above analyses, we see that existing methodologies cannot render reliable segmentation results for autonomous driving, as they all adopt the cross-entropy [de Boer *et al.*, 2005] loss function for model training, which equally evaluates the errors incurred by all image pixels without focusing on the important objects. 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.

Inspired by [Szegedy *et al.*, 2013], we propose a novel loss function with hierarchical structure as shown in Figure 3. In this structure, the objects with different 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 [Long *et al.*, 2015;

Vijay Badrinarayanan and Cipolla, 2017; Paszke *et al.*, 2016] with our proposed importance-aware loss. The experimental results on two typical autonomous driving datasets including CamVid [Brostow *et al.*, 2009] and Cityscapes [Cordts *et al.*, 2016] demonstrate that the important objects can be segmented more precisely than existing approaches.

The rest of this paper is organized as follows. Some related works are reviewed in Section 2. 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 and backward propagation rules for our proposed loss. In Section 5, we provide the experimental results on the representative traffic datasets including CamVid and Cityscapes. Finally, our paper is concluded in Section 6.

2 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 handcrafted features and using Random Forest [Shotton *et al.*, 2008; Brostow *et al.*, 2008; Silberman *et al.*, 2012] or boosting-based [Sturgess *et al.*, 2009; Kontschieder *et al.*, 2011; Zhang *et al.*, 2010] classifiers for predicting the class of image pixels. 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) [Sturgess *et al.*, 2009; Ladicky *et al.*, 2010; Ren *et al.*, 2012] are used to suppress the per-pixel prediction noise output by the classifiers.

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 [Farabet *et al.*, 2012; Grangier *et al.*, 2011; Gatta *et al.*, 2014] employ the features extracted by DCNN for class prediction. To make SS an end-to-end process, a fully convolutional network (FCN) [Long *et al.*, 2015] is applied and shows very promising results. Based on [Long *et al.*, 2015], many other methods [Chen *et al.*, 2016; Zheng *et al.*, 2015; Shen *et al.*, 2016; Eigen and Fergus, 2015] 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 [Vijay Badrinarayanan and Cipolla, 2017] and some other works like [Noh *et al.*, 2015; Hong *et al.*, 2015; Yang *et al.*, 2016] belong to this type.

Recently, several works have been done to apply SS to autonomous driving. [Pohlen *et al.*, 2016] 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 [Cordts *et al.*, 2016]. To further improve the efficiency and achieve real-time segmentation, more neural networks are designed for self-driving system such as ENet [Paszke *et al.*, 2016] and the work [Treml *et al.*, 2016].

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

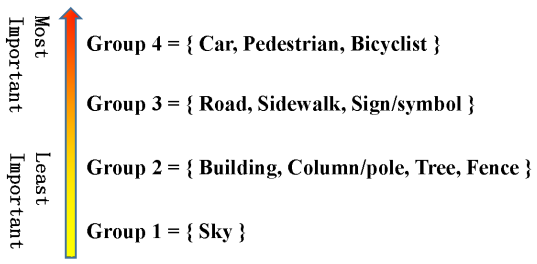

Figure 1: The rankings of importance of 11 studied object classes. Group 4 is the most important and Group 1 is the least important.

3 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. CamVid [Brostow *et al.*, 2009] 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 explanation. First of all, safety is the most critical issue for driving and the collisions with car, pedestrian and bicyclist are strongly opposed, so these objects show the top level importance in our algorithm. In contrast, road, sidewalk, and sign/symbol are less important as they only guarantee the normal driving. Sky is not essential here as it is seldom used as a cue for car control, so it is the least important among all above 11 classes. The detailed importance levels of all the investigated classes are depicted in Figure 1. Besides, it is worth mentioning that the users can re-define the objects' importance levels according to different criteria or their own prior knowledge.

To characterize the importance of objects in our method, we first define some notations for the ease of following descriptions. We define C as the number of classes in driving environment. The final output of an SS algorithm can be 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 (see Figure 2(a)). Here the one-hot encoding has the formation $[0, \cdots, 0, 1, 0, \cdots, 0]^T$ with the position of 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 real class label of the (i, j)-th pixel.

According to above mathematical definitions and the importance levels as shown in Figure 1, we propose a novel importance-aware loss with hierarchical structure as shown in Figure 3, in which different levels represent the object-

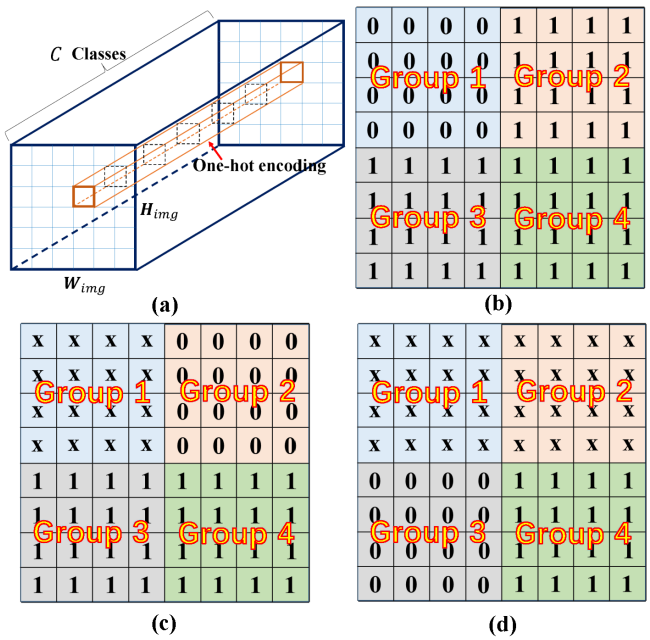

Figure 2: Illustration of critical mathematical definitions in our method. (a) The output of the algorithm is represented by a tensor $\boldsymbol{X}$, of which the height and width represent the $H_{img} \times W_{img}$ image, and the depth corresponds to the totally C classes. For a specific pixel, the depth constitutes a *one-hot encoding*. (b), (c), and (d) respectively present the $H_{img} \times W_{img}$ matrices $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$ for comparing the importance levels of four groups, in which we assume that the pixels of every group are arranged together (see the blocks with different colors).

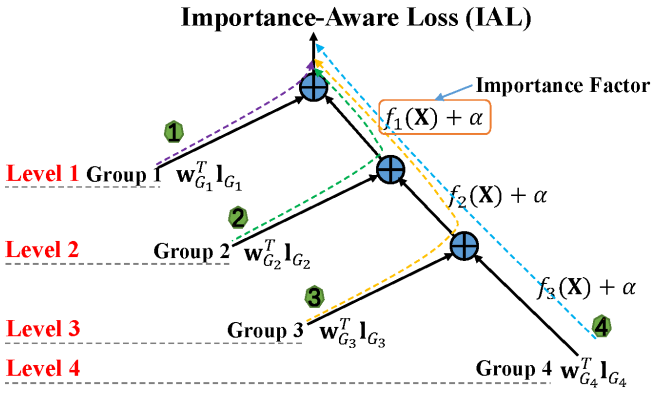

Figure 3: The illustration of our importance-aware loss with hierarchical structure. Level 1 to Level 4 indicate the importance levels of the classes in different groups, and the more important a group is, the higher level it stands. $\mathbf{l}_{G_1}$, $\mathbf{l}_{G_2}$, $\mathbf{l}_{G_3}$, and $\mathbf{l}_{G_4}$ are respectively the loss values of four groups calculated by cross-entropy loss. Besides, $\mathbf{w}_{G_1}$, $\mathbf{w}_{G_2}$, $\mathbf{w}_{G_3}$, and $\mathbf{w}_{G_4}$ are the weights for eliminating class imbalance correspondingly. The term $f_i(\mathbf{X}) + \alpha$ is called *importance factor*.

s with different importance. In Figure 3, the vectors $\mathbf{l}_{G_1}$, $\mathbf{l}_{G_2}$, $\mathbf{l}_{G_3}$ and $\mathbf{l}_{G_4}$ encode the values of cross-entropy loss [de Boer *et al.*, 2005] of the objects in Group 1, Group 2, Group 3, and Group 4, respectively, and the j-th element $(\mathbf{l}_{G_i})_j$ $(i = 1, 2, 3, 4)$ is calculated by

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

where $\mathbf{p}_c = exp(\boldsymbol{X}_{c,i,j}) / \sum_{k=1}^{C} exp(\boldsymbol{X}_{k,i,j})$ 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 $\boldsymbol{X}$, 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 $\mathbf{w}_{G_1}$, $\mathbf{w}_{G_2}$, $\mathbf{w}_{G_3}$ and $\mathbf{w}_{G_4}$ to record the corresponding weights of the objects in the four groups for avoiding class imbalance, and the object with fewer pixels is assigned larger weight [Eigen and Fergus, 2015]. The j-th element in $\mathbf{w}_{G_i}$ $(i = 1, 2, 3, 4)$ are

$$(\mathbf{w}_{G_i})_j = median\_freq / freq(i, j), \tag{2}$$

where $freq(i, j)$ is the number of pixels of the j-th class in Group i divided by the total number of pixels in images where the class $(G_i)_j$ is present, and $median\_freq$ is the median of these frequencies. Therefore, the weighted cross-entropy losses for Group 1 to Group 4 are $\mathbf{w}_{G_1}^T \mathbf{l}_{G_1}$, $\mathbf{w}_{G_2}^T \mathbf{l}_{G_2}$, $\mathbf{w}_{G_3}^T \mathbf{l}_{G_3}$ and $\mathbf{w}_{G_4}^T \mathbf{l}_{G_4}$ correspondingly.

Besides, for the four groups defined in Figure 3, we introduce three $H_{img} \times W_{img}$ matrices $\mathbf{M}_t$ $(t = 1, 2, 3)$ to model the importance relationship of four groups. For example, the $\mathbf{M}_1$ for comparing Group 1 and Groups 2, 3, and 4 is presented in Figure 2(b), in which the elements corresponding to the classes in Group 1 are set to 0, and the elements corresponding to Groups 2∼4 are 1 indicating that they are more important than Group 1. To further compare the importance of Group 2 and Groups 3 and 4, the elements of $\mathbf{M}_2$ (see Figure 2(c)) regarding Group 2 are set to 0, and the elements of Groups 3 and 4 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. Similarly, the $\mathbf{M}_3$ for comparing Group 3 and Group 4 is shown in Figure 2(d), where the elements of Group 3 and Group 4 are 0 and 1, respectively. The elements representing Group 1 and Group 2 are "x" since their importance comparisons with other groups have been studied.

Based on $\mathbf{M}_t$ $(t = 1, 2, 3)$, we define $f_t(\mathbf{X}) + \alpha$ as *importance factor* where α is a tuning parameter with default value 1, so the $f_t(\mathbf{X})$ $(t = 1, 2, 3)$ in Figure 3 are computed by

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

where $\mathbf{E}$ is an all-one matrix, and $\mathbb{I}\{\mathbf{M}_t \neq$ "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 $\mathbf{X}_{i,j} = \boldsymbol{X}_{c,i,j}$[1] with $c = \mathbf{Y}_{i,j}$. In (3), $\lambda_t \in \mathbb{R}^+$ $(t = 1, 2, 3)$ are tuning parameters and in this paper we set $\lambda_1 = \lambda_2 = \lambda_3 = 0.5$. Note that if λ_t 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

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

classifications of important classes.

Therefore, the loss of the objects in the four groups can be computed by following the arrows in Figure 3. 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 loss value of the classes in Group 3 is $(f_1(\mathbf{X})+\alpha)(f_2(\mathbf{X})+\alpha)(\mathbf{w}_{G_3}^T\mathbf{l}_{G_3})$. The classes in Group 4 are the most important and thus its weighted cross-entropy loss $\mathbf{w}_{G_4}^T\mathbf{l}_{G_4}$ should be augmented by three importance factors. Consequently, the loss of Group 4 is $(f_1(\mathbf{X})+\alpha)(f_2(\mathbf{X})+\alpha)(f_3(\mathbf{X})+\alpha)(\mathbf{w}_{G_4}^T\mathbf{l}_{G_4})$. Finally, the total value of our importance-aware loss is the sum of the loss values contributed by the four groups, which is

$$\begin{aligned} Loss =& \mathbf{w}_{G_1}^T\mathbf{l}_{G_1}+ \\ & (f_1(\mathbf{X})+\alpha)(\mathbf{w}_{G_2}^T\mathbf{l}_{G_2})+ \\ & (f_1(\mathbf{X})+\alpha)(f_2(\mathbf{X})+\alpha)(\mathbf{w}_{G_3}^T\mathbf{l}_{G_3})+ \\ & (f_1(\mathbf{X})+\alpha)(f_2(\mathbf{X})+\alpha)(f_3(\mathbf{X})+\alpha)(\mathbf{w}_{G_4}^T\mathbf{l}_{G_4}). \end{aligned} \tag{4}$$

From above analyses, we see that the cross-entropy losses of important classes will be augmented by more importance factors than the less importance ones. As a result, the more important an object class is, the greater the importance-aware loss value it will obtain. Moreover, from Eq. (4) we see that if we set all importance factors $f_t(\mathbf{X})+\alpha$ $(t=1,2,3)$ to 1, our proposed IAL function will degrade into the existing cross-entropy loss with all classes sharing the equal importance.

4 Forward and Backward Propagation Rules

Here, we give a general description 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

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

As a consequence, the *backward propagation* rules of IAL corresponding to Eqs. (8), (5), (6), and (7) are

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

$$\begin{aligned} \frac{\partial Q_1}{\partial \boldsymbol{X}}=&\frac{\partial f_1(\mathbf{X})}{\partial \boldsymbol{X}}(\mathbf{w}_{G_2}^T\mathbf{l}_{G_2}+Q_2) \\ &+(f_1(\mathbf{X})+\alpha)(\mathbf{w}_{G_2}^T*\frac{\partial \mathbf{l}_{G_2}}{\partial \boldsymbol{X}}+\frac{\partial Q_2}{\partial \boldsymbol{X}}), \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial Q_2}{\partial \boldsymbol{X}}=&\frac{\partial f_2(\mathbf{X})}{\partial \boldsymbol{X}}(\mathbf{w}_{G_3}^T\mathbf{l}_{G_3}+Q_3) \\ &+(f_2(\mathbf{X})+\alpha)(\mathbf{w}_{G_3}^T*\frac{\partial \mathbf{l}_{G_3}}{\partial \boldsymbol{X}}+\frac{\partial Q_3}{\partial \boldsymbol{X}}), \end{aligned} \tag{11}$$

$$\cdots\cdots$$

$$\begin{aligned} \frac{\partial Q_t}{\partial \boldsymbol{X}}=&\frac{\partial f_t(\mathbf{X})}{\partial \boldsymbol{X}}(\mathbf{w}_{G_{t+1}}^T\mathbf{l}_{G_{t+1}}+Q_{t+1}) \\ &+(f_t(\mathbf{X})+\alpha)(\mathbf{w}_{G_{t+1}}^T*\frac{\partial \mathbf{l}_{G_{t+1}}}{\partial \boldsymbol{X}}+\frac{\partial Q_{t+1}}{\partial \boldsymbol{X}}). \end{aligned} \tag{12}$$

where

$$\begin{aligned} \frac{\partial Q_{t+1}}{\partial \boldsymbol{X}}=&\frac{\partial f_{t+1}(\mathbf{X})}{\partial \boldsymbol{X}}(\mathbf{w}_{G_{t+2}}^T\mathbf{l}_{G_{t+2}}) \\ &+(f_{t+1}(\mathbf{X})+\alpha)(\mathbf{w}_{G_{t+2}}^T*\frac{\partial \mathbf{l}_{G_{t+2}}}{\partial \boldsymbol{X}}), \end{aligned} \tag{13}$$

and

$$\frac{\partial f_t(\mathbf{X})}{\partial \boldsymbol{X}}=[(\mathbf{M}_t+\lambda_t\mathbf{E})\odot(\mathbf{X}-\mathbf{M}_t)\odot\mathbb{I}\{\mathbf{M}_t\neq\text{“x”}\}]*\frac{\partial \mathbf{X}}{\partial \boldsymbol{X}}. \tag{14}$$

By denoting

$$(\frac{\partial \mathbf{X}}{\partial \boldsymbol{X}})_{:,i,j}=[0,0,\cdots,\frac{\partial \mathbf{X}_{i,j}}{\partial \boldsymbol{X}_{c,i,j}},\cdots,0,0]^T \tag{15}$$

and

$$\mathbf{A}=(\mathbf{M}_t+\lambda_t\mathbf{E})\odot(\mathbf{X}-\mathbf{M}_t)\odot\mathbb{I}\{\mathbf{M}_t\neq\text{“x”}\}, \tag{16}$$

we have

$$(\frac{\partial f_t(\mathbf{X})}{\partial \boldsymbol{X}})_{:,i,j}=(\mathbf{A}*\frac{\partial \mathbf{X}}{\partial \boldsymbol{X}})_{:,i,j}=\mathbf{A}_{i,j}(\frac{\partial \mathbf{X}}{\partial \boldsymbol{X}})_{:,i,j}\ , \tag{17}$$

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

For $\frac{\partial \mathbf{l}_{G_t}}{\partial \boldsymbol{X}}$, if the (i,j)-th pixel belongs to 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

$$\mathbf{h}_c=(\frac{\partial \mathbf{l}_{G_t}}{\partial \boldsymbol{X}})_{c,i,j}=\begin{cases}\dfrac{exp(\boldsymbol{X}_{c,i,j})}{\sum_{k=1}^{C}exp(\boldsymbol{X}_{k,i,j})}, & \text{if } c\neq\mathbf{Y}_{i,j};\\ \dfrac{exp(\boldsymbol{X}_{c,i,j})}{\sum_{k=1}^{C}exp(\boldsymbol{X}_{k,i,j})}-1, & \text{if } c=\mathbf{Y}_{i,j},\end{cases} \tag{18}$$

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

$$(\mathbf{w}_{G_t}^T*\frac{\partial \mathbf{l}_{G_t}}{\partial \boldsymbol{X}})_{:,i,j}=(\mathbf{w}_{G_t})_r[\mathbf{h}_1,\cdots,\mathbf{h}_{c-1},\mathbf{h}_c,\mathbf{h}_{c+1},\cdots,\mathbf{h}_C]^T. \tag{19}$$

Table 1: The comparison results of various methods on the Groups 1 and 2 of CamVid dataset. The records that IAL are better than the original network are marked in bold.

	Group 1	Group 2			
	Sky	Building	Column/Pole	Tree	Fence
ENet	**95.1**	**74.7**	35.4	77.8	**51.7**
ENet+IAL	88.2	68.5	**56.8**	**80.8**	42.2
SegNet	**92.4**	**88.8**	27.5	87.3	**49.3**
SegNet+IAL	85.7	81.4	**44.1**	**90.7**	40.2
FCN	**93.5**	**93.7**	33.1	91.2	**53.3**
FCN+IAL	86.7	85.9	**53.1**	**94.8**	43.5

Table 2: The comparison results of various methods on the Groups 3 and 4 of CamVid dataset. The records that IAL are better than the original network are marked in bold.

	Group 3			Group 4			
	Road	Sidewalk	Sign/Symbol	Car	Pedestrian	Bicyclist	Mean IoU
ENet	95.1	86.7	**51.0**	**82.4**	67.2	34.1	68.3
ENet+IAL	**95.3**	**90.9**	43.7	79.1	**72.3**	**50.9**	**69.9**
SegNet	97.2	84.4	**20.5**	**82.1**	57.1	30.7	65.2
SegNet+IAL	**97.4**	**88.5**	17.6	80.8	**61.4**	**45.8**	**66.7**
FCN	**98.1**	89.5	**25.1**	**84.5**	64.6	38.6	69.6
FCN+IAL	96.3	**91.8**	21.5	82.2	**69.5**	**57.6**	**71.2**

5 Experimental Results

To verify the effectiveness of our proposed importance-aware loss (IAL) function, we apply IAL to three existing deep neural networks, i.e. FCN [Cordts *et al.*, 2016], SegNet [Vijay Badrinarayanan and Cipolla, 2017] and ENet [Paszke *et al.*, 2016], to deal with SS problem. Among them, FCN and SegNet are popular deep methods for conventional SS, and ENet is a recently proposed deep network specifically for autonomous driving application. The cross-entropy loss adopted by these models will be replaced with our IAL during the training stage, and we term them as "FCN+IAL", "SegNet+IAL", and "ENet+IAL", respectively. Besides, we follow [Cordts *et al.*, 2016; Everingham *et al.*, 2015] and use the intersection-over-union (IoU) score to evaluate the performances of compared methods on different object classes.

We use the CamVid dataset [Brostow *et al.*, 2009] mentioned in Section 3 and a recent Cityscapes [Cordts *et al.*, 2016] dataset for our experiments. CamVid contains 367 training images, 26 validation images, and 233 test images. The resolution of images in this dataset is 960×720. 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. In Cityscapes dataset, we pick up 19 the most frequently occurred classes from the original 35 classes, and their importance groupings from trivial to important are

Group 1 = { Sky };

Group 2 = { Building, Wall, Fence, Vegetation, Terrain };

Group 3 = { Road, Sidewalk, Train };

Group 4 = { Person, Rider, Car, Truck, Bus, Motorcycle, Bicycle, Traffic light, Traffic sign, Pole }.

On both datasets, we train the neural networks on training sets, and observe their IoU scores on test sets. The experimental results of compared methods on the investigated classes of the two datasets are shown in Tables 1∼2 and Tables 3∼4, respectively. For the CamVid dataset, the results of ENet and SegNet are directly originated from [Paszke *et al.*, 2016], and we implement FCN by ourselves as no prior results on this dataset have been reported. For the Cityscapes dataset, the results of ENet, SegNet and FCN in Tables 3 and 4 are provided by the original papers [Paszke *et al.*, 2016; Pohlen *et al.*, 2016; Cordts *et al.*, 2016].

From the results shown in Tables 1 and 2, we observe that by employing our IAL, the IoU scores of important classes like pedestrain, bicyclist, car and sign/symbol can be significantly improved when compared with the settings without IAL. Not surprisingly, the IoU scores on some unimportant classes such as building and sky drop because they are trained with small weights by our IAL. However, if we compute the IoU scores averaged over all classes for all compared methods (see the last column in Table 2), we find that the networks with IAL are still able to achieve better performance than the original networks with cross-entropy loss, and the improvements are 1.6 for ENet, 1.5 for SegNet, and 1.6 for FCN.

From the results in Table 3 and Table 4, we see that the important classes in Group 4 are segmented with very high IoU scores by FCN+IAL, ENet+IAL and SegNet+IAL, such as person, rider, car, truck, bus, and bicycle. Specifically, the IoU scores of person and truck generated by ENet+IAL are as high as 87.7 and 73.5, which are significantly better than the results of ENet, FCN, and SegNet. For some unimportant classes in Group 2, the performances of IAL-based models are inferior to the original models. However, they will not have large impact on safe-driving as explained above. Furthermore, the last column of Table 4 reveals that the segmentation results of ENet, SegNet and FCN on the entire image are improved by utilizing our IAL. Besides, another interesting finding is that sky, which is inessential, is also segmented more precisely by IAL-based models than the corresponding

Table 3: The comparison results of various methods on the Groups 1, 2, and 3 of Cityscapes dataset. The records that IAL are better than the original network are marked in bold.

	Group 1	Group 2						Group 3				
	Sky	Pole	Building	Wall	Fence	Vegetation	Terrain	Road	Sidewalk	Train	Traffic Light	Traffic Sign
ENet	90.6	43.5	**85.0**	32.2	33.2	88.6	61.4	**96.3**	74.2	**48.1**	34.1	44.0
ENet+IAL	**96.9**	**63.7**	80.0	**39.4**	**41.6**	**88.8**	**67.1**	95.7	**84.4**	27.1	**60.3**	**72.9**
SegNet	91.8	35.7	**84.0**	28.5	29.0	87.0	63.8	**96.4**	73.2	**44.2**	39.8	45.2
SegNet+IAL	**98.2**	**52.3**	79.1	**34.9**	**36.3**	**87.2**	**69.7**	95.8	**83.3**	24.9	**65.4**	**71.9**
FCN	92.9	43.0	**88.7**	34.7	44.0	90.9	68.6	**97.3**	77.6	**45.9**	57.7	62.0
FCN+IAL	**97.0**	**63.0**	83.5	**42.5**	**54.7**	**91.1**	**75.0**	96.2	**88.3**	32.1	**70.6**	**75.2**

Table 4: The comparison results of various methods on the Groups 4 of Cityscapes dataset. The records that IAL are better than the original network are marked in bold.

	Group 4							
	Person	Rider	Car	Truck	Bus	Motorcycle	Bicycle	Mean IoU
ENet	65.5	38.4	90.6	36.9	50.5	**38.8**	55.4	58.3
ENet+IAL	**87.7**	**41.3**	**92.4**	**73.5**	**76.2**	24.1	**69.7**	**67.5**
SegNet	62.8	42.8	89.3	38.1	43.1	**35.8**	51.9	57.0
SegNet+IAL	**84.1**	**46.0**	**91.1**	**75.9**	**65.0**	22.2	**65.3**	**65.7**
FCN	75.4	50.5	91.9	35.3	49.1	**50.7**	65.2	64.3
FCN+IAL	**90.4**	**56.6**	**93.7**	**68.5**	**74.6**	31.5	**81.5**	**71.9**

baselines. Perhaps this is because the accurate segmentations of other objects also render valuable cues for partitioning the unimportant regions.

To intuitively present the effectiveness of our proposed loss function, we provide some representative segmentation results of ENet and ENet+IAL in Figure 4. For the important objects with large size (e.g. truck, bus, and road), we see that the regions segmented by ENet+IAL are very compact and most pixels of the corresponding regions are correctly classified. Comparatively, the original ENet yields much worse outputs than 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 ENet. For example, the traffic light indicated by a circle is rather small, and it is missed by 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 ENet. In contrast, ENet+IAL clearly identifies the pole from the background as indicated by the white circle. Here we only present the results related to ENet as ENet is the state-of-the-art deep neural network specifically designed for the application of self-driving. However, the comparisons between FCN vs. FCN+IAL and SegNet vs. SegNet+IAL also reveal the similar results.

According to above qualitative and quantitative 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 IoU score. Therefore, IAL is quite suitable for the application of autonomous driving.

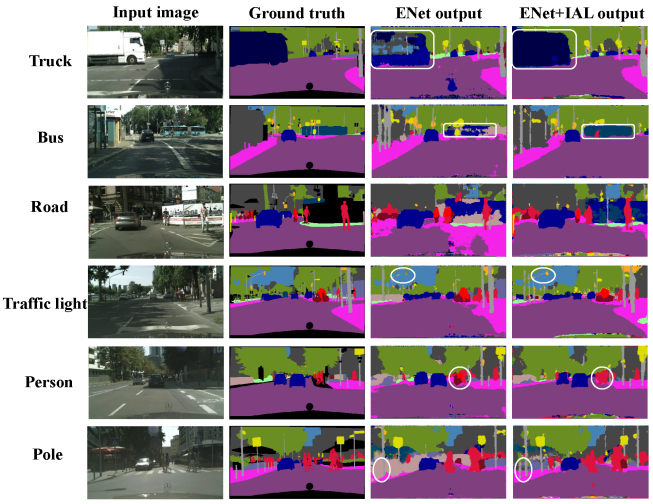

Figure 4: Representative segmentation results of ENet and ENet+IAL on important classes of Cityscapes dataset.

6 Conclusion

Semantic segmentation in driving environment is quite different from its traditional implementations, as various classes might have different levels of importance for safety driving. 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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61602246, 91420201, and 61472187), the 973 Program (No. 2014CB349303), and the Program for Changjiang Scholars.

References

[Brostow *et al.*, 2008] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and Recognition Using Structure from Motion Point Clouds. In *ECCV*, pages 44–57, 2008.

[Brostow *et al.*, 2009] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, pages 88–97, 2009.

[Chen *et al.*, 2016] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *Computer Science*, pages 357–361, 2016.

[Cordts *et al.*, 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, pages 3213–3223, 2016.

[de Boer *et al.*, 2005] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A Tutorial on the Cross-Entropy Method. *Annals OR*, pages 19–67, 2005.

[Eigen and Fergus, 2015] David Eigen and Rob Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture. In *CVPR*, pages 2650–2658, 2015.

[Everingham *et al.*, 2015] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. In *IJCV*, pages 98–136, 2015.

[Farabet *et al.*, 2012] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers. In *ICML*, pages 575–582, 2012.

[Gatta *et al.*, 2014] Carlo Gatta, Adriana Romero, and Joost van de Weijer. Unrolling Loopy Top-Down Semantic Feedback in Convolutional Deep Networks. In *CVPR Workshops*, pages 504–511, 2014.

[Gong *et al.*, 2015] Chen Gong, Dacheng Tao, Wei Liu, Stephen J. Maybank, Meng Fang, Keren Fu, and Jie Yang. Saliency propagation from simple to difficult. In *CVPR*, pages 2531–2539, 2015.

[Grangier *et al.*, 2011] David Grangier, Léon Bottou, and Ronan Collobert. Deep Convolutional Networks for Scene Parsing. In *ICML Workshops*, 2011.

[Hong *et al.*, 2015] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation. In *NIPS*, pages 1495–1503, 2015.

[Kontschieder *et al.*, 2011] Peter Kontschieder, Samuel Rota Bulò, Horst Bischof, and Marcello Pelillo. Structured class-labels in random forests for semantic image labelling. In *ICCV*, pages 2190–2197, 2011.

[Ladicky *et al.*, 2010] Lubor Ladicky, Paul Sturgess, Karteek Alahari, Christopher Russell, and Philip H. S. Torr. What, Where and How Many? Combining Object Detectors and CRFs. In *ECCV*, pages 424–437, 2010.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[Noh *et al.*, 2015] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. In *ICCV*, pages 1520–1528, 2015.

[Paszke *et al.*, 2016] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. *CoRR*, abs/1606.02147, 2016.

[Pohlen *et al.*, 2016] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *CoRR*, abs/1611.08323, 2016.

[Ren *et al.*, 2012] Xiaofeng Ren, Liefeng Bo, and Dieter Fox. RGB-D scene labeling: Features and algorithms. In *CVPR*, pages 2759–2766, 2012.

[Shen *et al.*, 2016] Xiaoyong Shen, Aaron Hertzmann, Jiaya Jia, Sylvain Paris, Brian L. Price, Eli Shechtman, and Ian Sachs. Automatic portrait segmentation for image stylization. *Computer Graphics Forum*, pages 93–102, 2016.

[Shotton *et al.*, 2008] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, pages 1–8, 2008.

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, pages 746–760, 2012.

[Sturgess *et al.*, 2009] Paul Sturgess, Karteek Alahari, Lubor Ladicky, and Philip H. S. Torr. Combining Appearance and Structure from Motion Features for Road Scene Understanding. In *BMVC*, pages 1–11, 2009.

[Szegedy *et al.*, 2013] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep Neural Networks for object detection. In *NIPS*, pages 2553–2561, 2013.

[Treml *et al.*, 2016] Michael Treml, José Arjona-Medina, Michael Unterthiner, et al. Speeding up Semantic Segmentation for Autonomous Driving. In *NIPS Workshop*, 2016.

[Vijay Badrinarayanan and Cipolla, 2017] Alex Kendall Vijay Badrinarayanan and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2017.

[Yang *et al.*, 2016] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee, and Ming-Hsuan Yang. Object Contour Detection with a Fully Convolutional Encoder-Decoder Network. In *CVPR*, pages 193–202, 2016.

[Zhang *et al.*, 2010] Chenxi Zhang, Liang Wang, and Ruigang Yang. Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In *ECCV*, pages 708–721, 2010.

[Zheng *et al.*, 2015] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *ICCV*, pages 1529–1537, 2015.