Pattern Recognition Letters 150 (2021) 258–264

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Soft-Boundary Label Relaxation with class placement constraints for semantic segmentation of the railway environment

Yuki Furitsu[a,*], Daisuke Deguchi[a], Yasutomo Kawanishi[a], Ichiro Ide[a], Hiroshi Murase[a], Hiroki Mukojima[b], Nozomi Nagamine[b]

[a] Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464–8601, Japan
[b] Railway Technical Research Institute, 2-8-38 Hikari-cho, Kokubunji-shi, Tokyo, 185–8540, Japan

ARTICLE INFO

Article history:
Received 18 January 2021
Revised 25 May 2021
Accepted 4 July 2021
Available online 27 July 2021

Edited by Prof. S. Sarkar

Keywords:
Semantic segmentation
Railway
Label relaxation

ABSTRACT

In this paper, we focus on the challenging task of the semantic segmentation of train front-view images. Managing trackside facilities can be done by using detailed and precise information about the surrounding railway environment. Semantic segmentation enables us to understand the 2D environment, but there is no adequate large-scale dataset available for training a CNN for this purpose. Some attempts have been made to generate pseudo-data from unlabeled sequential frames to compensate for the lack of volume in training data, but the moving speed of trains makes it difficult to apply them directly. We aim to solve this problem by proposing the Soft Boundary Label Relaxation (Soft-BLR) method, which considers label boundaries extending over multiple pixels to cope with more severely distorted pseudo-data and to better train the CNN in the initial training stage. Furthermore, we modify the loss function to penalize inference results based on the distance from the label boundary to solve the misalignment problems of border pixels. Through experimental evaluation, we report that the proposed method outperforms previous methods on not only the semantic segmentation of challenging railway images, but also that of general street-view images.

1. Introduction

Railways are valued means of transportation due to their speed, capacity, and reliability, and their extension reaches a total of more than a million kilometers around the Globe. To cope with such characteristics, railway operators especially emphasize on the safety and the prevention of accidents. From simple railway signals to more advanced Automatic Train Stop (ATS) systems, various technologies are used to ensure the safety of passengers. However, the collection of geological / geometrical positions and the types of trackside facilities are currently done manually with high human cost, and some railway operators are even unaware of where and what trackside facilities exist along their tracks due to managing problems within different departments. Daily maintenance of such facilities is also essential, yet it is still being done by manual and/or visual inspection. Therefore, a fully automatic technology that can collect data about trackside facilities and can be used for their maintenance is in crucial need for railway operators. To meet such needs, the use of semantic segmentation for railway environment understanding is currently being considered.

Semantic segmentation, a task of allocating a single semantic label to each and every pixel within an image, can be used to understand the surrounding environment in detail. Almost every modern method of semantic segmentation utilizes Convolutional Neural Network (CNN), and thus requires supervised data [3]. For this reason, building an adequate dataset for semantic segmentation is a substantial issue. A typical pixel-level manual annotation of an image takes more than an hour [4]. Training a CNN model generally requires a massive volume of training data, and constructing a large-scale dataset for every application is unrealistic. Although domain adaptation has been studied to transfer training results to similar domains, such as synthetic to real-world data [10], this approach cannot be applied across dissimilar domains like from street environment to the railway environment.

To cope with such lack of sufficient training data, Zhu et al. [18] originally proposed joint image-label propagation to generate pseudo-data using a small number of labeled images and neighboring sequential unlabeled images for the street-view image domain. They also introduced Boundary Label Relaxation (BLR) to cope with distorted training data generated by joint image-label propagation. In joint image-label propagation, pseudo-data are generated by

* Corresponding author.
E-mail address: furitsuy@murase.is.i.nagoya-u.ac.jp (Y. Furitsu).

https://doi.org/10.1016/j.patrec.2021.07.014
0167-8655/

Fig. 1. Boundary Label Relaxation (BLR) in 1D label space. The ground truth of the border pixels (shown in green) are modified to contain class labels that appear in both sides of the border. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

transforming both an image and its label using densely calculated optical flows between sequential frames. During this transformation, multiple labels can propagate to a single pixel location, thus making the class boundary ambiguous. BLR is introduced to take into account such ambiguity during the training of a CNN. They were able to augment the Cityscapes dataset [4] to eleven times its original size and train a CNN effectively using this method, but the following problems remain:

(i) Large-scale augmentation is only possible when the dataset is taken at a high frame-rate and with small camera movement. With data taken from cameras mounted on trains, high operation speed of trains makes it difficult to propagate the labels for generating pseudo-data. Distant pixels can be propagated to a single location, increasing the label ambiguity. However, the original method [18] only considers BLR for areas within distance 1 of a label boundary.
(ii) BLR modifies the label space to allow multiple labels as ground truth in a single pixel location (Fig. 1). This means that inferences with wrong label alignment may be considered as being correct.

To tackle these problems, based on the fact that label boundary distortions of more than a single pixel appear when propagating labels of railway images, we propose a "Soft-BLR" that considers a larger width at the label boundary. We introduce a novel loss function of the CNN to penalize inference results based on the distance from the relaxed label boundary to solve the misalignment problems existent in the original BLR. This loss function also enables a smoother training of the CNN, as close misclassifications will have lower loss values than those of distant misclassifications. These efforts contribute to accurate semantic segmentation of railway environments, even in cases where the original annotations for training are barely available. We demonstrate the effectiveness of the proposed method through experimental analysis, and discuss future applications and possible further improvements.

2. Related work

2.1. Semantic segmentation and datasets

On general semantic segmentation, numerous studies have been made in recent years. The use of a CNN became popular with the emerge of the Fully Convolutional Network (FCN) [11], and the trend has been followed by many state-of-the-art models like SegNet [2], PSPNet [17], and DeepLabv3+ [3].

There are some datasets aiming at different domains for the purpose of training a CNN for semantic segmentation. However, existing large-scale datasets mostly consist of either object-wise images [6], or street-view images [4] [12], and cannot be directly used for other domains like the railway environment.

There are some datasets aiming exclusively at the semantic segmentation of railways. For example, RailSem19 [16] is a dataset consisting of 8500 unique images taken from train front-view cameras and their pixel-level annotations. Its annotation scheme is focused on the detection of geometric objects like rails and the trackbed around them, but rail-side facilities like pole, beam, and crossing gate are all labeled as a single class. This makes it insufficient for understanding the railway environment, as detailed labeling of each rail-side facility is required in order to figure out its position.

2.2. Data augmentation using label propagation

Data augmentation is a common technique to scale up an insufficient volume of training data. There are some approaches to propagate semantic labels of labelled frames across its sequential unlabeled frames. Patch matching methods [1] use an Expectation-Maximization (EM) based algorithm utilizing image patch based similarities or semantically consistent hierarchical regions, but multiple hyperparameters like image patch size make them difficult to optimize. Meanwhile, flow based methods [8] make use of pre-calculated optical flow between raw sequential images, and apply it to warp the semantic labels of labelled frames. However, highly accurate optical flow estimation is difficult even with the current state-of-the-art CNN based methods [14]. Recently, Zhu et al. [18] proposed joint image-label propagation, which uses motion vectors learned from video prediction models to jointly warp both the raw image and the semantic labels across their sequential frames. This method reduces the mis-alignment of semantic labels at warped frames, but require the training of the video prediction model for maximum performance.

3. Soft-Boundary label relaxation with a novel loss function

3.1. Overview of the proposed method

We build upon the idea that for a train front-view image sequence augmented with joint image-label propagation, its class boundaries will be more distorted and misaligned than that of street scenes. Using the original Boundary Label Relaxation (BLR) would not be sufficient, as severe distortions cause multiple labels to propagate to a single pixel location and make the class boundary ambiguous. Even more, it does not take into account the order of the classes assigned to pixels around the boundary, resulting in misalignments in the inference results being considered correct. To solve this problem, we propose a novel method to handle misalignments at class boundaries caused by distortions more flexibly and accurately.

Fig. 2 shows the overall process flow of the proposed method. First, we augment the training data using the joint image-label propagation [18]. Then, we train a CNN using the proposed Soft-Boundary Label Relaxation (Soft-BLR) and back-propagate the loss calculated using the novel loss function.

3.2. Soft-Boundary label relaxation (soft-BLR)

BLR was originally designed to accept multiple classes at a class boundary pixel, ensuring robustness against accumulated propagation artifacts. However, applying joint image-label propagation to train front-view video sequences generates severe distortions such that the conventional single-pixel width BLR cannot resolve. As a solution to this problem, we propose a new method called "Soft-BLR" that considers a wider class boundary and uses a novel loss function that can keep spatial class order alignment.

To be specific, we first widen the width of the "boundary pixel" to let the focused pixel's class label contain all the labels of its neighboring pixels within an arbitrary distance N. By widening the width of the class boundary, we can take into account distortions that misplace borders by a larger margin.

For the purpose of semantic segmentation, a one-hot vector (i.e. a vector where only a single element contains the value 1, while all others contain the value 0) of the ground truth $\mathbf{g}(\mathbf{x})$ is often used

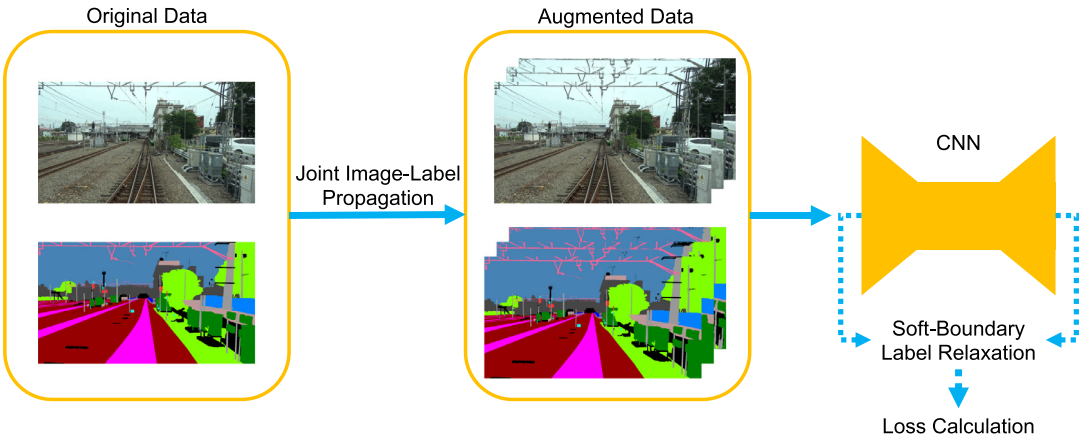

Fig. 2. Overall process flow of the proposed method.

as the target variable for each pixel. Here, $\mathbf{x}$ is a pixel location vector pointing to the pixel location in an image. When we define the target variable as $\mathbf{t}(\mathbf{x}) = \mathbf{g}(\mathbf{x})$, the cross-entropy loss of the pixel location vector can be calculated with the following equation:

$$\mathbf{L} = -\mathbf{t}(\mathbf{x})^{\mathrm{T}} \log(\mathbf{f}(\mathbf{x})), \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is the predicted likelihood of the pixel location vector. Note that $\log(\cdot)$ returns the logarithm for each element within a vector.

Let $\mathcal{N}(\mathbf{x})$ be a set of pixel location vectors that appear within distance D around the pixel location vector $\mathbf{x}$:

$$\mathcal{N}(\mathbf{x}) = \{\mathbf{y} \,|\, d(\mathbf{x}, \mathbf{y}) < D\}, \tag{2}$$

where $d(\mathbf{x}, \mathbf{y})$ is the distance between the two pixel location vectors.

Using this, $\mathbf{t}(\mathbf{x})$ can be modified as a multi-hot vector (i.e. a vector where some elements contain the value 1, while all others contain the value 0):

$$\mathbf{t}(\mathbf{x}) = \min\left(\mathbf{1}, \sum_{\tilde{\mathbf{x}} \in \mathcal{N}(\mathbf{x})} \mathbf{g}(\tilde{\mathbf{x}})\right), \tag{3}$$

where the operation $\min(\cdot, \cdot)$ is applied coordinate-wise.

When the distance limit $D = 1$, it will be equivalent to the conventional BLR [18]. In the proposed method, to consider larger distortions of a maximum of D pixels, we set $D > 1$.

However, the misalignment problem still remains. In fact, even when the border width is equal to 1, i.e. in the case of [18]'s method, the inference results to be classified as being the same as that of the ground truth even when there are misalignments. Suppose we are classifying pixels which lie along the boundary of classes A and B. We define $\mathbf{a}$ and $\mathbf{b}$ as one-hot vectors of the corresponding classes:

$$\mathbf{a} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4}$$

To consider the order of the ground-truth labels and their distance from the borders, the likelihood of the pixel at location vector $\mathbf{x}_2$ being class A must be larger than that being class B, and the difference between the two likelihoods of the classes must be larger when the pixel location vector is further away from the border at $\mathbf{x}_1$. These conditions can be formulated as:

$$P(A|\mathbf{x}_2) > P(B|\mathbf{x}_2), \tag{5}$$

$$P(A|\mathbf{x}_1) - P(B|\mathbf{x}_1) > P(A|\mathbf{x}_2) - P(B|\mathbf{x}_2). \tag{6}$$

Let $P(A|\mathbf{x}_2) = \mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{a}$. Here, the right-side of the equation denotes the likelihood of class A at pixel location $\mathbf{x}_2$. Eqs. (5) and (6) can be restated using $\mathbf{t}(\mathbf{x})$ as:

$$\mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{a} > \mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{b}, \tag{7}$$

$$\mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{a} - \mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{b} > \mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{a} - \mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{b}. \tag{8}$$

Eq. (8) can be rewritten as:

$$\mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{b} - \mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{b} > \mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{a} - \mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{a}. \tag{9}$$

In the example, the ground truth of $\mathbf{x}_2$ and $\mathbf{x}_1$ is of class A, so both terms $\mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{a}$ and $\mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{a}$ must have the same likelihood. Therefore, we assume their difference in the likelihood to be 0. Furthermore, we assume that the difference in the left-side of the equation to be a constant α to represent the likelihood of class B decreasing linearly with the distance from the class border. With such assumptions, the equation can be written as:

$$\mathbf{t}(\mathbf{x}_2)^{\mathrm{T}}\mathbf{b} - \mathbf{t}(\mathbf{x}_1)^{\mathrm{T}}\mathbf{b} = \alpha > 0. \tag{10}$$

To generalize these conditions to 2D label space, we first define the following function to calculate the weight of a one-hot vector of an arbitrary class $\mathbf{c}$ at pixel location $\mathbf{x}$:

$$h(\mathbf{x}, \mathbf{c}) = \max_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \{\max(0, 1 - \alpha d(\mathbf{x}, \mathbf{y}))\mathbf{g}(\mathbf{y})^{\mathrm{T}}\mathbf{c}\}. \tag{11}$$

Note that the second $\max(\cdot, \cdot)$ in Eq. 11 ensures that all class likelihoods remain a positive value.

We use chessboard distance for the aforementioned distance $d(\cdot, \cdot)$ to discretize the distance into integers.

Finally, using $\mathbf{c}_{\mathrm{all}}$, a set of all one-hot vectors corresponding to classes that appear in the dataset, we decide the multi-hot-like value $\mathbf{t}(\mathbf{x})$:

$$\mathbf{t}(\mathbf{x}) = \sum_{\mathbf{c} \in \mathbf{c}_{\mathrm{all}}} h(\mathbf{x}, \mathbf{c})\, \mathbf{c}. \tag{12}$$

An example of the representation of $\mathbf{t}(\mathbf{x})$ for each method in 1D is shown in Fig. 3. In conventional methods, CNNs are trained using one-hot vectors of the ground-truth labels, and the values within the vector suddenly changes near label boundaries (between pixel location $\mathbf{x}_2$ and $\mathbf{x}_3$). In contrast, the target variable proposed by [18] contains multi-hot values near label boundaries, and reduces the effect of misclassification near such locations. The target variable proposed in this research further extends this approach as

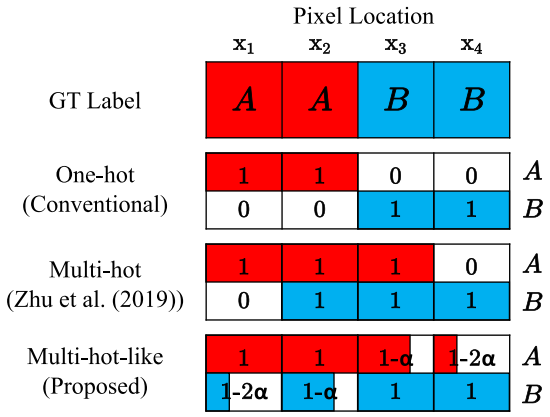

Fig. 3. 1D example of the comparison between label spaces of the target variable $\mathbf{t}(\mathbf{x})$ with its pixel location number. Conventional target variable contains one-hot class labels of the ground-truth, while the target variable defined by [18] contains multi-hot class labels. The target variable defined in the proposed method contains multi-hot-like values calculated using Eq. (12).

weighted multi-hot-like values, and acts as a soft constraint accepting ambiguity in class boundaries depending on its distance from the border.

The target variable $\mathbf{t}(\mathbf{x})$ is used as a mask in Eq. (1) to calculate the cross-entropy loss, which can then be back-propagated to train an arbitrary CNN.

4. Experimental evaluation

In this section, we evaluate the proposed Soft BLR method and compare the results against existing methods. First, we evaluate on a widely-used public dataset for a general understanding of the effects of the soft-boundary label relaxation. Then, we use private train front-view datasets to test the specific effects of the proposed method on the railway environment.

4.1. Comparative methods and evaluation metrics

We tested the following four methods:

DL DeepLabv3+ trained without any BLR [3].
BLR-1 DeepLabv3+ trained with single-pixel width BLR with the border width being 1 pixel [18].
Proposed (BLR-*N*)
DeepLabv3+ trained with multiple-pixel width BLR without any modification to the loss function with the border width being *N* pixels (simple extension of [18]).
Proposed (SBLR-*N*) DeepLabv3+ trained with soft multiple-pixel width BLR with a modification to the loss function with the border width being *N* pixels.

For evaluation metrics, we calculate the pixel Intersection-over-Union for every class (class IoU), and average them to obtain mean Intersection-over-Union (mIoU). Each metric evaluates pixel level correspondences between the ground truth and predicted labels, with higher values being better.

4.2. Experiment 1: Cityscapes dataset

4.2.1. Datasets

For a general evaluation of the proposed method, we used the Cityscapes dataset [4]. This dataset includes pixel-level annotations of 5000 street-view images, and is commonly used for the evaluation of semantic segmentation frameworks. In this experiment, we use 2975 images included in the training set for training each method, and 500 images included in the validation set for quantitative analysis of the training results. We used the ImageNet [5] and the Mapillary Vistas [12] datasets for pre-training the network.

Table 1
MIoU [%] of each method for the Cityscapes dataset.

Method	mIoU
DL	75.15
BLR-1	75.80
Proposed (BLR-2)	76.02
Proposed (BLR-3)	76.58
Proposed (SBLR-1)	**76.70**
Proposed (SBLR-2)	76.44
Proposed (SBLR-3)	76.60

4.2.2. Implementation details

Our CNN architecture for the experiment is based on DeepLabv3+ [3], with the backbone being ResNeXt50 [15] considering its moderate computational complexity. The number of training epochs is set to 50, crop size to 896×896 pixels, hyperparameter α to 0.2 and the other parameters the same as those of [18]. We train and evaluate the CNN of each method once.

4.2.3. Experimental results

The results in Table 1 show that the proposed Soft-BLR proved effective in a setting of semantic segmentation where the volume of training data is sufficient, as the mIoU improved by more than 1.5% from DeepLabv3+.

4.3. Experiment 2: Train front-view dataset

4.3.1. Datasets

We build upon our previous research [7] and enlarge the train front-view dataset, which now consists of a total of 116 images with its annotations. It also contains more complex and diverse cases like an interaction with other trains and sections with multiple train tracks.

We used image sequences taken by the Railway Technology Research Institute (RTRI), Japan. The train operated at a maximum of 85 km/h, and the images were taken at 60 frames per second. From the image sequences, we extracted 116 images so that they contain various objects and backgrounds, and annotated them pixel-wise over 22 pre-defined classes. Details on the class settings are given in [7].

For the training, we split the dataset into 66 images for training and 50 images for testing. We use joint image-label propagation to augment the training data with its sequential frames, and generate $66 \times 3 = 198$ pseudo training data. Furthermore, in addition to this train front-view dataset, we used ImageNet [5], Mapillary Vistas [12], and Cityscapes [4] datasets for pre-training the network.

4.3.2. Implementation details

We follow the method proposed by [18] for synthesizing training data from sparsely annotated video frame sequences. Joint image-label propagation is used to propagate both the video frames and their annotations, resulting in better alignment of class borders. The propagation is performed by referring to the predictions made by the underlying SDCNet [13] and FlowNet2 [9] for optical flow calculation. For more details, the original paper [18] should be referred.

After scaling up the dataset using joint image-label propagation, we train an arbitrary semantic segmentation CNN model using the Soft-BLR.

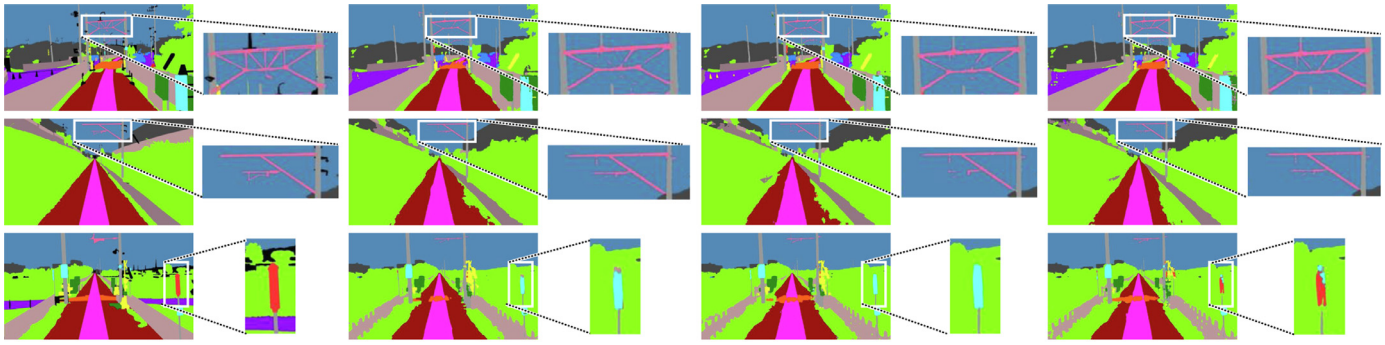

Fig. 4. Several output examples of the three methods along with the ground truth for the train front-view dataset.

Table 2
MIoU [%] of each method for the train front-view dataset.

Method	mIoU
DL	58.12
BLR-1	58.73
Proposed (BLR-2)	59.85
Proposed (BLR-3)	59.59
Proposed (SBLR-1)	59.13
Proposed (SBLR-2)	59.85
Proposed (SBLR-3)	**60.35**

Training settings are the same as those for the Cityscapes dataset in Experiment 1, except that the number of training epochs is set to 100. We train the CNN ten times per method, and report the average mIoU.

4.3.3. Experimental results

Fig. 4 shows an example of the outputs of the three methods with enlarged views of detailed structures, and Table 2 shows the resulting mIoUs. Using multiple-pixel width for BLR proved effective with mIoU improvement of more than 1.1% from BLR-1 to BLR-2, and modifying the loss function further improved it by more than 0.5% from BLR-2 to SBLR-3 in a setting where the volume of training data is insufficient.

4.4. Experiment 3: Extended train front-view dataset

4.4.1. Datasets

We further test the class-wise segmentation ability of the proposed method to find the best value for the hyperparameter α using an extended version of the train front-view dataset. This dataset is built upon the dataset used in Experiment 2, and consists of 315 fully annotated images.

For the training, we split the dataset into 265 images for training and 50 images for testing. In addition to this train front-view dataset, we used ImageNet [5], Mapillary Vistas [12], and Cityscapes [4] datasets for pre-training the network.

4.4.2. Implementation details and evaluation metrics

In this experiment, we do not augment the training data using joint image-label propagation. We also test different values for the hyperparameter α. Note that for $\alpha = 0.4$, SBLR-3 yields the same likelihoods in Eq. (11) as SBLR-2, therefore is not included in the experiment. Other implementation details are the same as those for Experiment 2. We train the CNN five times per method, and report the average mIoU for each α. We also report the class IoU for the best value of α.

Table 3
MIoU [%] of each method for the extended train front-view dataset.

Method	α	mIoU
DL	0.0	63.93
BLR-1	0.0	65.68
Proposed (BLR-2)	0.0	66.43
Proposed (BLR-3)	0.0	65.79
Proposed (SBLR-1)	0.1	65.98
Proposed (SBLR-2)	0.1	66.26
Proposed (SBLR-3)	0.1	66.42
Proposed (SBLR-1)	0.2	**67.23**
Proposed (SBLR-2)	0.2	66.32
Proposed (SBLR-3)	0.2	66.22
Proposed (SBLR-1)	0.3	66.01
Proposed (SBLR-2)	0.3	66.63
Proposed (SBLR-3)	0.3	65.73
Proposed (SBLR-1)	0.4	66.26
Proposed (SBLR-2)	0.4	66.75

4.4.3. Experimental results

Table 3 shows the resulting mIoUs, and Table 4 shows the class IoU for every class. Note that class IoUs for the class “traffic light” are missing as it did not appear in the testing data. Using multiple-pixel width for BLR proved effective with mIoU improvement of more than 0.7% from BLR-1 to BLR-2, and modifying the loss function further improved it by 0.8% from BLR-2 to SBLR-1 ($\alpha = 0.2$) in a setting where the volume of training data is sufficient. The best value of the hyperparameter α was 0.2 for SBLR-1, 0.1 for SBLR-3, and 0.4 for SBLR-2. Regarding the class IoU for $\alpha = 0.2$, SBLR-1 gave the best IoU in eight out of the twenty-two classes, while still showing comparative results in others.

5. Discussion and applications

5.1. Effectiveness of the proposed method

From the results of Experiment 1 (Section 4.2), the effectiveness of not only the modified loss function, but also widening the width of BLR on a general dataset was observed. We did not augment the training data, so in theory there should not be any distortion in label boundaries. However, the ground-truth pixel-level annotations by human annotators are not always perfect. The true label boundary can be off by several pixels, and in such cases the widened BLR can help the CNN to not focus too much on such misplacements.

Table 4
Class IoU [%] of each method for the extended train front-view dataset. Note that the hyperparameter α was set to 0.2.

Method	Flat	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Nature	Sky	Human	Vehicle
DL	52.53	87.92	73.92	70.63	81.17	–	0.70	89.58	97.83	73.32	63.31
BLR-1	52.71	87.57	72.99	69.67	81.47	–	0.00	89.01	97.81	71.47	65.39
Proposed (BLR-2)	51.46	87.94	74.58	**71.06**	80.95	–	31.04	89.45	**97.86**	72.69	65.51
Proposed (BLR-3)	51.31	87.95	72.96	69.06	80.86	–	**38.39**	89.52	97.85	74.63	61.55
Proposed (SBLR-1)	**58.41**	**88.16**	73.44	70.55	**81.95**	–	28.29	**89.82**	97.82	**75.09**	62.83
Proposed (SBLR-2)	54.69	87.84	**75.93**	70.10	81.49	–	20.72	89.63	97.85	72.66	**66.12**
Proposed (SBLR-3)	50.03	87.42	72.59	70.87	81.64	–	28.95	89.53	97.84	73.32	65.87
Method	Train	2-wheel	Rail	Track	Crossing	Facility	C. gate	O. facility	Rw. light	Rw. sign	Plat.
DL	81.53	49.90	93.68	85.80	77.38	81.76	69.09	54.78	44.01	67.35	56.43
BLR-1	80.98	35.01	93.36	85.87	**81.77**	79.65	68.65	56.61	**55.44**	64.30	67.31
Proposed (BLR-2)	87.99	43.57	93.80	86.23	79.68	81.93	70.08	**58.10**	45.28	66.28	58.56
Proposed (BLR-3)	85.45	34.09	92.73	85.60	77.48	81.41	**71.11**	55.93	52.61	66.37	56.72
Proposed (SBLR-1)	83.32	34.74	93.71	**87.02**	78.60	**83.23**	68.72	57.49	50.15	66.37	**74.47**
Proposed (SBLR-2)	87.77	**50.56**	**93.91**	86.21	74.57	80.69	68.83	56.21	47.81	67.29	59.83
Proposed (SBLR-3)	**89.54**	49.34	93.80	86.03	77.65	82.26	70.13	57.71	49.91	**68.58**	57.56

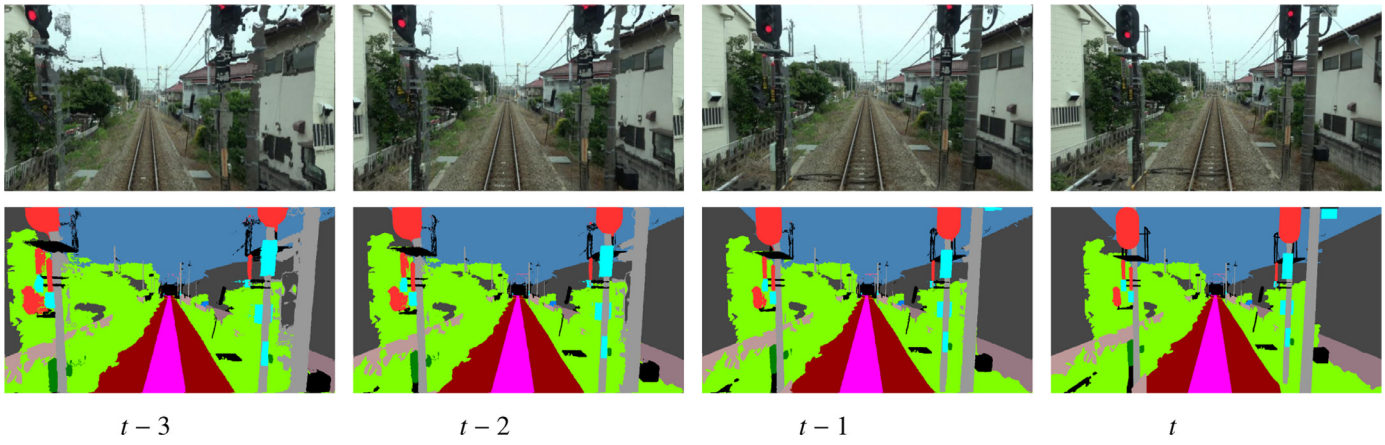

Fig. 5. Example of distorted images and labels generated by propagation for the train front-view dataset. The image and the annotation in the t-th frame were propagated for three sequential frames.

From the results of Experiment 2 (Section 4.3), we can clearly see that the proposed method outperforms existing methods with a larger margin than the improvement of BLR-1 from the original DeepLabv3+. Furthermore, the new loss function and the modified label space seems also effective, and the best result showed improvement of more than 2% from the original DeepLabv3+.

From the results of Experiment 3 (Section 4.4), we can also see a similar trend as those in previous experiments.

Widening the pixel width of BLR seems effective with and without the modifications to the loss function for training on augmented pseudo data. This may be due to the severe distortion generated in them, as seen in Fig. 5. Such distortions displace border pixels by chessboard distances of more than 1, which the conventional BLR with single-pixel width could not handle. Meanwhile, SBLR-1 shows the best mIoU in Experiment 1 and Experiment 3, as they do not involve pseudo data generation resulting in less distortions in the training data compared to Experiment 2. Overall, modifying the loss function with Soft-BLR proves to be effective for all settings of the experiments.

Also, we limited the number of propagation to ± 1 frames during the training stage in Experiment 2. Propagating over more frames would generate images and labels that are clearly unrealistic (Fig. 5, first and second columns), which implies the limitations of the underlying SDCNet [13] and FlowNet2 [9] in calculating the optical flow between distant frames and predicting the missing parts. Pre-training them with train front-view images should enable further propagations, but even then, the fast moving speed of trains would limit the propagation length to a shorter period compared to that of street scenes.

5.2. Differences in the class IoU

Here, we investigate the differences in the class IoUs of each method when training on the railway dataset. Table 2 shows the class IoU for the 22 classes defined in the train front-view dataset. We can see that in most cases, one of the proposed methods show the highest class IoU. The margin of improvement varies for each class, but it is generally larger for classes that contain thin and/or small objects like “overhead facility” and “railway light”.

For conventional semantic segmentation methods, classifying such objects is especially challenging, since misclassifying them by a few pixels during training would have no difference to doing so by any margin. Even more, calculating optical flow and predicting their propagations are also difficult, leading to more distortions within the augmented dataset. Multiple-pixel width BLR along with a new label space to disallow misalignments would enable the training stage of the CNN to gradually predict the class borders better, as closer misclassifications would have lower loss values. Even in cases where the training data contain severely distorted class borders, there will be less effects on the training of the CNN thanks to the Soft-BLR.

Moreover, the class IoU of “traffic sign” vastly improves with the use of the proposed method. Objects belonging to this class rarely appeared in the dataset, and conventional semantic segmentation methods seem to almost ignore it completely. Such behavior is thought to be the result of strict class boundaries and loss calculations, penalizing even subtle misclassifications and discouraging the CNN to predict uncommon class labels. Using the proposed method relaxed such strict boundary borders and/or loss calcula-

tions, encouraging the CNN to learn a wider variety of classes and their boundaries.

6. Conclusion

In this paper, we focused on the challenging task of the semantic segmentation of train front-view images and proposed the Soft-Boundary Label Relaxation (Soft-BLR) method as a solution. It extends the width of the class boundary to multiple pixels to cope with more severely distorted pseudo-data. Furthermore, we proposed a novel loss function to penalize inference results based on the distance from the label boundary to solve the misalignment problem.

Through experimental evaluation, we confirmed that the proposed method clearly outperforms previous researches on the semantic segmentation of both a large-scale street-view dataset and small-scale train front-view datasets.

Future work includes improving the method of data augmentation itself to enable the training of a better representation of the railway environment, and applying semantic segmentation to real-world railway maintenance tasks such as the inspection of building limits for safe train passing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Acknowledgment

Parts of this research were supported by MEXT, Grant-in-Aid for Scientific Research JP17H00745.

References

[1] V. Badrinarayanan, F. Galasso, R. Cipolla, Label propagation in video sequences, in: Proc. 2010 IEEE Conf. Comput. Vis. Pattern Recognit., 2010, pp. 3265–3272.
[2] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 2481–2495.
[3] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proc. 15th Eur. Conf. Comput. Vis. (Part VII), 2018, pp. 833–851.
[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes dataset for semantic urban scene understanding, in: Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3213–3223.
[5] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: Proc. 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.
[6] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.
[7] Y. Furitsu, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, H. Mukojima, N. Nagamine, Semantic segmentation of railway images considering temporal continuity, in: Proc. 5th Asian Conf. Pattern Recognit. (Part I), 2019, pp. 639–652.
[8] R. Gadde, V. Jampani, P.V. Gehler, Semantic video CNNS through representation warping, in: Proc. 16th IEEE Int. Conf. Comput. Vis., 2017, pp. 4463–4472.
[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, Flownet 2.0: Evolution of optical flow estimation with deep networks, in: Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2462–2470.
[10] Y. Li, L. Yuan, N. Vasconcelos, Bidirectional learning for domain adaptation of semantic segmentation, in: Proc. 2019 IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6929–6938.
[11] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. 2015 IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.
[12] G. Neuhold, T. Ollmann, S. Rota Bulo, P. Kontschieder, The mapillary vistas dataset for semantic understanding of street scenes, in: Proc. 16th IEEE Int. Conf. Comput. Vis., 2017, pp. 4990–4999.
[13] F.A. Reda, G. Liu, K.J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, B. Catanzaro, SDC-Net: video prediction using spatially-displaced convolution, in: Proc. 15th Eur. Conf. Comput. Vis. (Part VII), 2018, pp. 718–733.
[14] D. Sun, X. Yang, M.Y. Liu, J. Kautz, PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume, in: Proc. 2018 IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8934–8943.
[15] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1492–1500.
[16] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, C. Beleznai, RailSem19: a dataset for semantic rail scene understanding, in: Proc. 2019 IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2019, pp. 1221–1229.
[17] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2881–2890.
[18] Y. Zhu, K. Sapra, F.A. Reda, K.J. Shih, S. Newsam, A. Tao, B. Catanzaro, Improving semantic segmentation via video propagation and label relaxation, in: Proc. 2019 IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8856–8865.