Standardized Max Logits: A Simple yet Effective Approach for Identifying Unexpected Road Obstacles in Urban-Scene Segmentation

Sanghun Jung[*1] Jungsoo Lee[*1] Daehoon Gwak[1] Sungha Choi[2] Jaegul Choo[1]

[1]KAIST AI [2]LG AI Research

[1]{shjung13, bebeto, daehoon.gwak, jchoo}@kaist.ac.kr [2]shachoi@korea.ac.kr

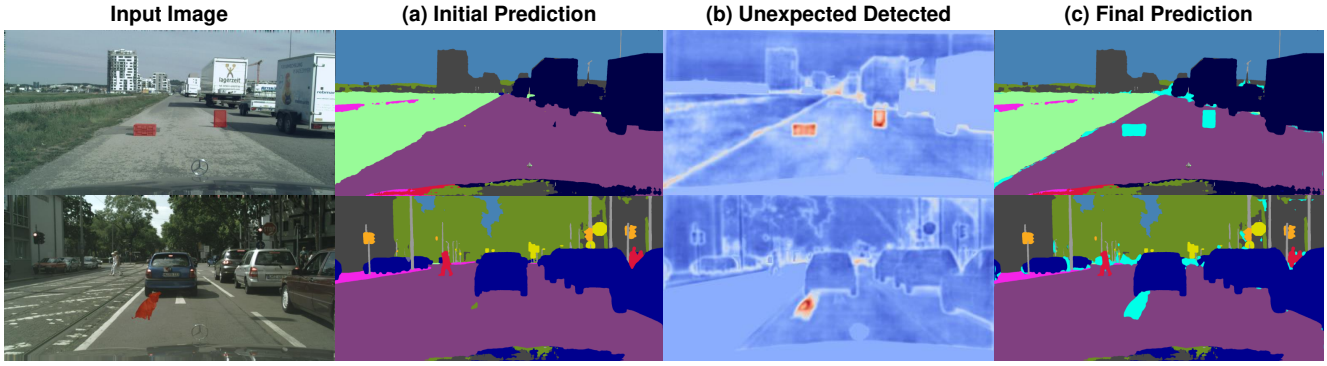

Figure 1: Results of our approach on identifying unexpected obstacles on roads. (a) Previous segmentation networks classify the unexpected obstacles (*e.g.,* dogs) as one of the pre-defined classes (*e.g.,* road) which may be detrimental from the safety-critical perspective. (b) Through our method, we detect the unexpected obstacles. (c) Finally, we can obtain the final prediction of segmentation labels with unexpected obstacles (cyan-colored objects) identified.

Abstract

Identifying unexpected objects on roads in semantic segmentation (e.g., identifying dogs on roads) is crucial in safety-critical applications. Existing approaches use images of unexpected objects from external datasets or require additional training (e.g., retraining segmentation networks or training an extra network), which necessitate a non-trivial amount of labor intensity or lengthy inference time. One possible alternative is to use prediction scores of a pre-trained network such as the max logits (i.e., maximum values among classes before the final softmax layer) for detecting such objects. However, the distribution of max logits of each predicted class is significantly different from each other, which degrades the performance of identifying unexpected objects in urban-scene segmentation. To address this issue, we propose a simple yet effective approach that ***standardizes*** *the max logits in order to align the different distributions and reflect the relative meanings of max logits within each predicted class. Moreover, we consider the local regions from two different perspectives based on the intuition that neighboring pixels share similar semantic information. In contrast to previous approaches, our method does not utilize any external datasets or require additional training, which makes our method widely applicable to existing pre-trained segmentation models. Such a straightforward approach achieves a new state-of-the-art performance on the publicly available Fishyscapes Lost & Found leaderboard with a large margin. Our code is publicly available at this link*[1].

1. Introduction

Recent studies [7, 8, 18, 34, 36, 37, 11] in semantic segmentation focus on improving the segmentation performance on urban-scene images. Despite such recent advances, these approaches cannot identify *unexpected objects* (*i.e.,* objects not included in the pre-defined classes during training), mainly because they predict all the pixels as one of the pre-defined classes. Addressing such an issue is critical especially for safety-critical applications such as autonomous driving. As shown in Fig. 1, wrongly predicting a dog (*i.e.,* an unexpected object) on the road as the road does not stop the autonomous vehicle, which may lead to roadkill. In this safety-critical point of view, the dog should be detected as an unexpected object which works as the starting point of the autonomous vehicle to handle these objects differently (*e.g.,* whether to stop the car or circumvent the dog).

Several studies [3, 22, 21, 4, 29, 2, 13] tackle the problem of detecting such unexpected objects on roads. Some ap-

* indicates equal contribution

[1]https://github.com/shjung13/Standardized-max-logits

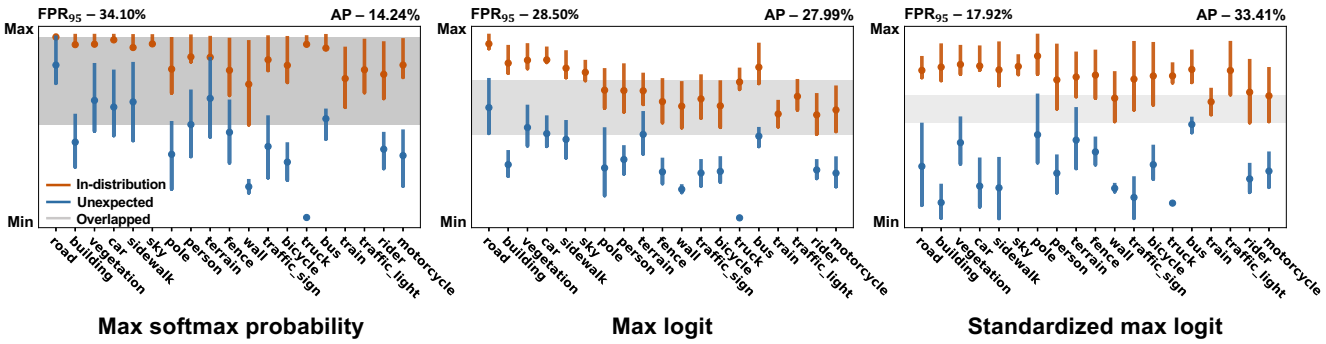

Figure 2: Box plots of MSP, max logit, and standardized max logit in Fishyscapes Static. X-axis denotes the classes which are sorted by the occurrences of pixels in the training phase. Y-axis denotes the values of each method. Red and blue represent the distributions of values in in-distribution pixels and unexpected pixels, respectively. The lower and upper limits of each bar indicate the Q1 and Q3 while the dot represents the mean value of its predicted class. The gray indicates the overlapped regions of the two groups. The opacity of the gray region is proportional to the FPR at TPR 95%. Standardizing the max logits in a class-wise manner clearly reduces the FPR.

proaches [2, 4] utilize external datasets [30, 20] as samples of unexpected objects while others [22, 33, 21, 27] leverage image resynthesis models for erasing the regions of such objects. However, such approaches require a considerable amount of labor intensity or necessitate a lengthy inference time. On the other hand, simple approaches which leverage only a pre-trained model [16, 19, 17] are proposed for out-of-distribution (OoD) detection in image classification, the task of detecting images from a different distribution compared to that of the train set. Based on the intuition that a correctly classified image generally has a higher maximum softmax probability (MSP) than an OoD image [16], MSP is used as the anomaly score (*i.e.,* the value used for detecting OoD samples). Alternatively, utilizing the max logit [15] (*i.e.,* maximum values among classes before the final softmax layer) as the anomaly score is proposed, which outperforms using MSP for detecting anomalous objects in semantic segmentation. Note that *high* prediction scores (*e.g.,* MSP and max logit) indicate *low* anomaly scores and vice versa.

However, directly using the MSP [16] or the max logit [15] as the anomaly score has the following limitations. Regarding the MSP [16], the softmax function has the fast-growing exponential property which produces highly confident predictions. Pre-trained networks may be highly confident with OoD samples which limits the performance of using MSPs for detecting the anomalous samples [19]. In the case of the max logit [15], as shown in Fig. 2, the values of the max logit have their own ranges in each predicted class. Due to this fact, the max logits of the unexpected objects predicted as particular classes (*e.g.,* road) exceed those of other classes (*e.g.,* train) in the in-distribution objects. This can degrade the performance of detecting unexpected objects on evaluation metrics (*e.g.,* AUROC and AUPRC) that use the same threshold for all classes.

In this work, inspired by this finding, we propose standardizing the max logits in a class-wise manner, termed *standardized max logits* (SML). Standardizing the max logits aligns the distributions of max logits in each predicted class, so it enables to reflect the relative meanings of values within a class. This reduces the false positives (*i.e.,* in-distribution objects detected as the unexpected objects, highlighted as gray regions in Fig. 2) when using a single threshold.

Moreover, we further improve the performance of identifying unexpected obstacles using the local semantics from two different perspectives. First, we remove the false positives in boundary regions where predicted class changes from one to another. Due to the class changes, the boundary pixels tend to have low prediction scores (*i.e.,* high anomaly scores) compared to the non-boundary pixels [32, 1]. In this regard, we propose a novel *iterative boundary suppression* to remove such false positives by replacing the high anomaly scores of boundary regions with low anomaly scores of neighboring non-boundary pixels. Second, in order to remove the remaining false positives in both boundary and non-boundary regions, we smooth them using the neighboring pixels based on the intuition that local consistency exists among the pixels in a local region. We term this process as *dilated smoothing*.

The main contributions of our work are as follows:

- We propose a *simple yet effective* approach for identifying unexpected objects on roads in urban-scene semantic segmentation.
- Our proposed approach can easily be applied to various existing models since our method does not require additional training or external datasets.
- We achieve a new state-of-the-art performance on the publicly available Fishyscapes Lost & Found Leaderboard[2] among the previous approaches with a large

[2]https://fishyscapes.com/

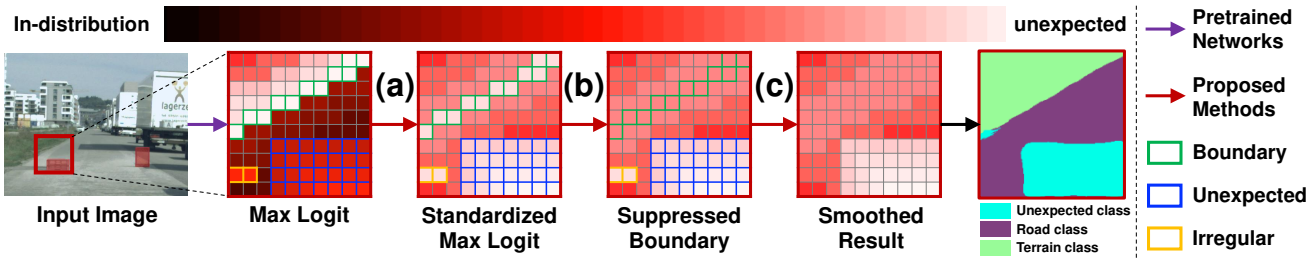

Figure 3: Overview of our method. We obtain the max logits from a segmentation network and (a) standardize it using the statistics obtained from the training samples. (b) Then, we iteratively replace the standardized max logits of boundary regions with those of surrounding non-boundary pixels. (c) Finally, we apply dilated smoothing to consider local semantics in broad receptive fields.

margin and negligible computation overhead while not requiring additional training and OoD data.

2. Related Work

2.1. Semantic segmentation on urban driving scenes

Recent studies [7, 8, 18, 34, 36, 37, 11, 5, 28, 26] have strived to enhance the semantic segmentation performance on urban scenes. The studies [18, 34] consider diverse scale changes in urban scenes or leverage the innate geometry and positional patterns found in urban-scene images [8]. Moreover, several studies [5, 28, 26] have proposed more efficient architectures to improve the inference time, which is critical for autonomous driving. Despite the advances, unexpected objects cannot be identified by these models, which is another important task for safety-critical applications. Regarding the importance of such a task from the safety-critical perspective, we focus on detecting unexpected obstacles in urban-scene segmentation.

2.2. Detecting unexpected objects in semantic segmentation

Several studies [2, 4, 3] utilize samples of unexpected objects from external datasets during the training phase. For example, by assuming that the objects cropped from the ImageNet dataset [30] are anomalous objects, they are overlaid on original training images [2] (*e.g.*, Cityscapes) to provide samples of unexpected objects. Similarly, another previous work [4] utilizes the objects from the COCO dataset [20] as samples of unexpected objects. However, such methods require retraining the network by using the additional datasets, which hampers to utilize a given pre-trained segmentation network directly.

Other work [22, 33, 21, 27] exploits the image resynthesis (*i.e.,* reconstructing images from segmentation predictions) for detecting unexpected objects. Based on the intuition that image resynthesis models fail to reconstruct the regions with unexpected objects, these studies use the discrepancy between an original image and the resynthesized image with such objects excluded. However, utilizing an extra image resynthesis model to detect unexpected objects necessitates a lengthy inference time that is critical in semantic segmentation. In the real-world application of semantic segmentation (*e.g.,* autonomous driving), detecting unexpected objects should be finalized in real-time. Considering such issues, we propose a simple yet effective method that can be applied to a given segmentation model without requiring additional training or external datasets.

3. Proposed Method

This section presents our approach for detecting unexpected road obstacles. We first present how we standardize the max logits in Section 3.2 and explain how we consider the local semantics in Section 3.3.

3.1. Method Overview

As our method overview is illustrated in Fig. 3, we first obtain the max logits and standardize them, based on the finding that the max logits have their own ranges according to the predicted classes. These different ranges cause unexpected objects (pixels in blue boxes) predicted as a certain class to have higher max logit values (*i.e.,* lower anomaly scores) than in-distribution pixels in other classes. This issue is addressed by standardizing the max logits in a class-wise manner since it enables to reflect the relative meanings within each predicted class.

Then, we remove the false positives (pixels in green boxes) in boundary regions. Generally, false positives in boundary pixels have lower prediction scores than neighboring in-distribution pixels. We reduce such false positives by iteratively updating boundary pixels using anomaly scores of neighboring non-boundary pixels. Additionally, there exist a non-trivial number of pixels that have significantly different anomaly scores compared to their neighboring pixels, which we term as *irregulars* (pixels in yellow boxes). Based on the intuition that local consistency (*i.e.,* neighboring pixels sharing similar semantics) exists among pixels in a local region, we apply the smoothing filter with broad receptive fields. Note that we use *the negative value of the final SML* as the anomaly score.

The following describes the process of how we obtain the max logit and the prediction at each pixel with a given image and the number of pre-defined classes. Let $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$ and C denote the input image and the number of

pre-defined classes, where H and W are the image height, and width, respectively. The logit output $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ can be obtained from the segmentation network before the softmax layer. Then, the max logit $\boldsymbol{L} \in \mathbb{R}^{H\times W}$ and prediction $\hat{\boldsymbol{Y}} \in \mathbb{R}^{H\times W}$ at each location h, w are defined as

$$\boldsymbol{L}_{h,w} = \max_c \mathbf{F}_{c,h,w} \tag{1}$$

$$\hat{\boldsymbol{Y}}_{h,w} = \arg\max_c \mathbf{F}_{c,h,w}, \tag{2}$$

where $c \in \{1, ..., C\}$.

3.2. Standardized Max Logits (SML)

As described in Fig. 2, standardizing the max logits aligns the distributions of max logits in a class-wise manner. For the standardization, we obtain the mean μ_c and variance σ_c^2 of class c from the training samples. With the max logit $\boldsymbol{L}_{h,w}$ and the predicted class $\hat{\boldsymbol{Y}}_{h,w}$ from the Eqs. (1) and (2), we compute the mean μ_c and variance σ_c^2 by

$$\mu_c = \frac{\sum_i \sum_{h,w} \mathbb{1}(\hat{\boldsymbol{Y}}^{(i)}_{h,w} = c) \cdot \boldsymbol{L}^{(i)}_{h,w}}{\sum_i \sum_{h,w} \mathbb{1}(\hat{\boldsymbol{Y}}^{(i)}_{h,w} = c)} \tag{3}$$

$$\sigma_c^2 = \frac{\sum_i \sum_{h,w} \mathbb{1}(\hat{\boldsymbol{Y}}^{(i)}_{h,w} = c) \cdot (\boldsymbol{L}^{(i)}_{h,w} - \mu_c)^2}{\sum_i \sum_{h,w} \mathbb{1}(\hat{\boldsymbol{Y}}^{(i)}_{h,w} = c)}, \tag{4}$$

where i indicates the i-th training sample and $\mathbb{1}(\cdot)$ represents the indicator function.

Next, we standardize the max logits by the obtained statistics. The SML $\boldsymbol{S} \in \mathbb{R}^{H\times W}$ in a test image at each location h, w is defined as

$$\boldsymbol{S}_{h,w} = \frac{\boldsymbol{L}_{h,w} - \mu_{\hat{\boldsymbol{Y}}_{h,w}}}{\sigma_{\hat{\boldsymbol{Y}}_{h,w}}}. \tag{5}$$

3.3. Enhancing with Local Semantics

We explain how we apply iterative boundary suppression and dilated smoothing by utilizing the local semantics.

3.3.1 Iterative boundary suppression

To address the problem of wrongly predicting the boundary regions as false positives and false negatives, we iteratively suppress the boundary regions. Fig. 4 illustrates the process of iterative boundary suppression. We gradually propagate the SMLs of the neighboring non-boundary pixels to the boundary regions, starting from the outer areas of the boundary (green-colored pixels) to inner areas (gray-colored pixels). To be specific, we assume the boundary width as a particular value and update the boundaries by iteratively reducing the boundary width at each iteration. This process is defined as follows. With a given boundary width r_i at the i-th iteration and the semantic segmentation output

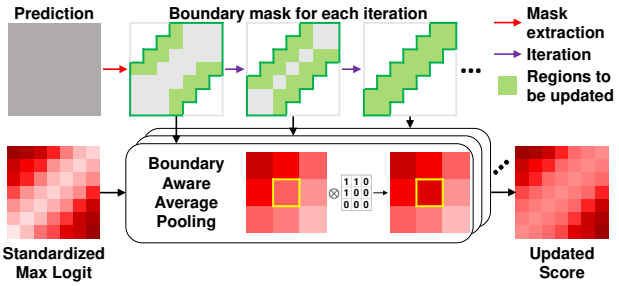

Figure 4: How iterative boundary suppression works. After standardizing the max logits, we apply average pooling by only using the SMLs of non-boundary pixels (*i.e.,* boundary-aware average pooling) for several iterations. The boundary mask is obtained from a prediction output of a segmentation network.

$\hat{\boldsymbol{Y}}$, we obtain the non-boundary mask $\boldsymbol{M}^{(i)} \in \mathbb{R}^{H\times W}$ at each pixel h, w as

$$\boldsymbol{M}^{(i)}_{h,w} = \begin{cases} 0, & \text{if } {}^{\exists}h', w' \ \textit{s.t.,}\ \hat{\boldsymbol{Y}}_{h,w} \neq \hat{\boldsymbol{Y}}_{h',w'} \\ 1, & \text{otherwise} \end{cases}, \tag{6}$$

for ${}^{\forall}h', w'$ that satisfies $|h - h'| + |w - w'| \leq r_i$.

Next, we apply the boundary-aware average pooling on the boundary pixels as shown in Fig. 4. This applies average pooling on a boundary pixel only with the SMLs of neighboring non-boundary pixels. With the boundary pixel b and its receptive field $\mathcal{R}$, the boundary-aware average pooling (BAP) is defined as

$$BAP(\boldsymbol{S}^{(i)}_{\mathcal{R}}, \boldsymbol{M}^{(i)}_{\mathcal{R}}) = \frac{\sum_{h,w} \boldsymbol{S}^{(i)}_{h,w} \times \boldsymbol{M}^{(i)}_{h,w}}{\sum_{h,w} \boldsymbol{M}^{(i)}_{h,w}}, \tag{7}$$

where $\boldsymbol{S}^{(i)}_{\mathcal{R}}$ and $\boldsymbol{M}^{(i)}_{\mathcal{R}}$ denote the patch of receptive field $\mathcal{R}$ on $\boldsymbol{S}^{(i)}$ and $\boldsymbol{M}^{(i)}$, and $(h, w) \in \mathcal{R}$ enumerates the pixels in $\mathcal{R}$. Then, we replace the original value at the boundary pixel b using the newly obtained one. We iteratively apply this process for n times by reducing the boundary width by $\Delta r = 2$ at each iteration. We also set the size of receptive field $\mathcal{R}$ as 3×3. In addition, we empirically set the number of iterations n and initial boundary width r_0 as 4 and 8.

3.3.2 Dilated smoothing

Since iterative boundary suppression only updates boundary pixels, the irregulars in the non-boundary regions are not addressed. Hence, we address these pixels by smoothing them using the neighboring pixels based on the intuition that the local consistency exists among the pixels in a local region. In addition, if the adjacent pixels used for iterative boundary suppression do not have sufficiently low or high anomaly scores, there may still exist boundary pixels that remain as false positives or false negatives even after the process. In this regard, we broaden the receptive fields of the smoothing filter using dilation [35] to reflect the anomaly scores beyond boundary regions.

Models	Additional training		Utilizing OoD Data	mIoU	FS Lost & Found		FS Static	
	Seg. Network	Extra Network			**AP** ↑	FPR_{95} ↓	**AP** ↑	FPR_{95} ↓
MSP [16]	✗	✗	✗	80.30	1.77	44.85	12.88	39.83
Entropy [16]	✗	✗	✗	80.30	2.93	44.83	15.41	39.75
Density - Single-layer NLL [3]	✗	✓	✗	80.30	3.01	32.90	40.86	21.29
kNN Embedding - density [3]	✗	✗	✗	80.30	3.55	30.02	44.03	20.25
Density - Minimum NLL [3]	✗	✓	✗	80.30	4.25	47.15	62.14	17.43
Density - Logistic Regression [3]	✗	✓	✓	80.30	4.65	24.36	57.16	13.39
Image Resynthesis [22]	✗	✓	✗	81.40	5.70	48.05	29.60	27.13
Bayesian Deeplab [25]	✓	✗	✗	73.80	9.81	38.46	48.70	15.50
OoD Training - Void Class	✓	✗	✓	70.40	10.29	22.11	45.00	19.40
Ours	✗	✗	✗	80.33	**31.05**	**21.52**	**53.11**	**19.64**
Discriminative Outlier Detection Head [2]	✓	✓	✓	79.57	31.31	19.02	96.76	0.29
Dirichlet Deeplab [24]	✓	✗	✓	70.50	34.28	47.43	31.3	84.60

Table 1: Comparison with previous approaches reported in Fishyscapes Leaderboard. Models are sorted by the AP scores in Fishyscapes Lost & Found test set. We achieve a new state-of-the-art performance among the approaches that do not require additional training on the segmentation network or OoD data on Fishyscapes Lost & Found dataset. Bold fonts indicate the highest performance in its evaluation metric among approaches that do not 1) retrain segmentation networks, 2) train extra networks, and 3) utilize OoD data.

For the smoothing filter, we leverage the Gaussian kernel since it is widely known that the Gaussian kernel removes noises [12]. With a given standard deviation σ and convolution filter size k, the kernel weight $\boldsymbol{K} \in \mathbb{R}^{k \times k}$ at location i, j is defined as

$$\boldsymbol{K}_{i,j} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\Delta i^2 + \Delta j^2}{2\sigma^2}\right), \tag{8}$$

where $\Delta i = i - \frac{(k-1)}{2}$ and $\Delta j = j - \frac{(k-1)}{2}$ are the displacements of location i, j from the center. In our setting, we set the kernel size k and σ to 7 and 1, respectively. Moreover, we empirically set the dilation rate as 6.

4. Experiments

This section describes the datasets, experimental setup, and quantitative and qualitative results.

4.1. Datasets

Fishyscapes Lost & Found [3] is a high-quality image dataset containing real obstacles on the road. This dataset is based on the original Lost & Found [29] dataset. The original Lost & Found is collected with the same setup as Cityscapes [9], which is a widely used dataset in urban-scene segmentation. It contains real urban images with 37 types of unexpected road obstacles and 13 different street scenarios (*e.g.,* different road surface appearances, strong illumination changes, and etc). Fishyscapes Lost & Found further provides the pixel-wise annotations for 1) unexpected objects, 2) objects with pre-defined classes of Cityscapes [9], and 3) void (*i.e.,* objects neither in pre-defined classes nor unexpected objects) regions. This dataset includes a public validation set of 100 images and a hidden test set of 275 images for the benchmarking.

Fishyscapes Static [3] is constructed based on the validation set of Cityscapes [9]. Regarding the objects in the PASCAL VOC [10] as unexpected objects, they are overlaid on the Cityscapes validation images by using various blending techniques to match the characteristics of Cityscapes. This dataset contains 30 publicly available validation samples and 1,000 test images that are hidden for benchmarking.

Road Anomaly [22] contains images of unusual dangers which vehicles confront on roads. It consists of 60 web-collected images with anomalous objects (*e.g.,* animals, rocks, and etc.) on roads with a resolution of 1280×720. This dataset is challenging since it contains various driving circumstances such as diverse scales of anomalous objects and adverse road conditions.

4.2. Experimental Setup

Implementation Details We adopt DeepLabv3+ [6] with ResNet101 [14] backbone for our segmentation architecture with the output stride set to 8. We train our segmentation networks on Cityscapes [9] which is one of the widely used datasets for urban-scene segmentation. We use the same pre-trained network for all experiments.

Evaluation Metrics For the quantitative results, we compare the performance by the area under receiver operating characteristics (AUROC) and average precision (AP). In addition, we measure the false positive rate at a true positive rate of 95% (FPR_{95}) since the rate of false positives in high-recall areas is crucial for safety-critical applications. For the qualitative analysis, we visualize the prediction results using the threshold at a true positive rate of 95% (TPR_{95}).

Baselines We compare ours with the various approaches reported in the Fishyscapes leaderboard. We also report results on the Fishyscapes validation sets and Road Anomaly with previous approaches that do not utilize external datasets or require additional training for fair comparisons. Additionally, we compare our method with approaches that are not reported in the Fishyscapes leaderboard. Thus, we

Models	mIoU	FS Lost & Found			FS Static			Road Anomaly		
		AUROC ↑	AP ↑	FPR_{95} ↓	AUROC ↑	AP ↑	FPR_{95} ↓	AUROC ↑	AP ↑	FPR_{95} ↓
MSP [16]	80.33	86.99	6.02	45.63	88.94	14.24	34.10	73.76	20.59	68.44
Max Logit [15]	80.33	92.00	18.77	38.13	92.80	27.99	28.50	77.97	24.44	64.85
Entropy	80.33	88.32	13.91	44.85	89.99	21.78	33.74	75.12	22.38	68.15
kNN Embedding - Density [3]	80.30	-	4.1	22.30	-	-	-	-	-	-
†SynthCP* [33]	80.33	88.34	6.54	45.95	89.90	23.22	34.02	76.08	24.86	64.69
Ours	80.33	**96.88**	**36.55**	**14.53**	**96.69**	**48.67**	**16.75**	**81.96**	**25.82**	**49.74**

Table 2: Comparison with other baselines in the Fishyscapes validation sets and the Road Anomaly dataset. † denotes that the results are obtained from the official code with our pre-trained backbone and * denotes that the model requires additional learnable parameters. Note that the performance of kNN Embedding - Density is provided from the Fishyscapes [3] team.

include the previous method using max logit [15] and SynthCP [33] that leverages an image resynthesis model for such comparison. Note that SynthCP requires training of additional networks.

4.3. Evaluation Results

This section provides the quantitative and qualitative results. We first show the results on Fishyscapes datasets and Road Anomaly, and then present the comparison results with various backbone networks. Additionally, we report the computational cost and the qualitative results by comparing with previous approaches.

4.3.1 Comparison on Fishyscapes Leaderboard

Table 1 shows the leaderboard result on the test sets of Fishyscapes Lost & Found and Fishyscapes Static. The Fishyscapes Leaderboard categorizes approaches by checking whether they require retraining of segmentation networks or utilize OoD data. In this work, we add the *Extra Network* column under the *Additional Training* category. Extra networks refer to the extra learnable parameters that need to be trained using a particular objective function other than the one for the main segmentation task. Utilizing extra networks may require a lengthy inference time, which could be critical for real-time applications such as autonomous driving. Considering such importance, we add this category for the evaluation.

As shown in Table 1, we achieve a new state-of-the-art performance on the Fishyscapes Lost & Found dataset with a large margin, compared to the previous models that do not require additional training of the segmentation network and external datasets. Additionally, we even outperform 6 previous approaches in Fishyscapes Lost & Found and 5 models in Fishyscapes Static which fall into at least one of the two categories. Moreover, as discussed in the previous work [3], retraining the segmentation network with additional loss terms impair the original segmentation performance(*i.e.,* mIoU) as can be shown in the cases of Bayesian Deeplab [25], Dirichlet Deeplab [24], and OoD Training with void class in Table 1. This result is publicly available on the Fishyscapes benchmark website.

4.3.2 Comparison on Fishyscapes validation sets and Road Anomaly

For a fair comparison, we compare our method on Fishyscapes validation sets and Road Anomaly with previous approaches which do not require additional training and OoD data. As shown in Table 2, our method outperforms other previous methods in the three datasets with a large margin. Additionally, our method achieves a significantly lower FPR_{95} compared to previous approaches.

4.3.3 Qualitative Analysis

Fig. 5 visualizes the pixels detected as unexpected objects (*i.e.,* white regions) with the TPR at 95%. While previous approaches using MSP [16] and max logit [15] require numerous in-distribution pixels to be detected as unexpected, our method does not. To be more specific, regions that are less confident (*e.g.,* boundary pixels) are detected as unexpected in MSP [16] and max logit [15]. However, our method clearly reduces such false positives which can be confirmed by the significantly reduced number of white regions.

5. Discussion

In this section, we conduct an in-depth analysis on the effects of our proposed method along with the ablation studies.

Models	AUROC ↑	AP ↑	FPR_{95} ↓
Max Logit	92.00	18.77	38.13
SML	96.54	27.61	15.46
SML + B Supp.	96.82	31.63	14.58
SML + D. Smoothing	96.70	36.00	15.65
SML + B Supp. + D. Smoothing	**96.89**	**36.55**	**14.53**

Table 3: Ablation study on our proposed methods. B Supp. and D. Smoothing refer to iterative boundary suppression and dilated smoothing, respectively.

5.1. Ablation Study

Table 3 describes the effect of each proposed method in our work with the Fishyscapes Lost & Found validation set. SML achieves a significant performance gain over using the max logit [15]. Performing iterative boundary suppression on SMLs improves the overall performance (*i.e.,* 4% increase in AP and 1% decrease in FPR_{95}). On the other hand, despite the increase in AP, performing dilated

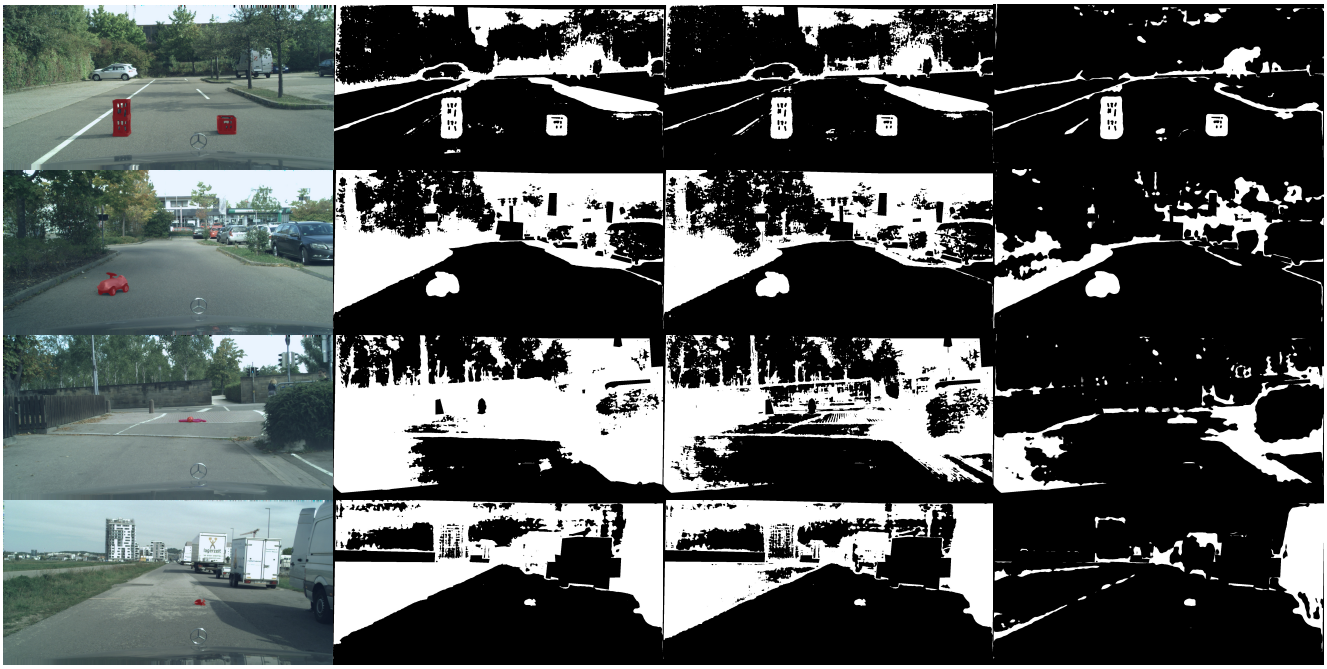

Figure 5: Unexpected objects detected with TPR_{95}. We compare our method with MSP [16] and max logit [15]. White pixels indicate objects which are identified as unexpected objects. Our method significantly reduces the number of false positive pixels compared to the two approaches.

smoothing on SMLs without iterative boundary suppression results in an unwanted slight increase in FPR_{95}. The following is the possible reason for the result. When dilated smoothing is applied without iterative boundary suppression, the anomaly scores of non-boundary pixels may be updated with those of boundary pixels. Since the non-boundary pixels of in-distribution objects have low anomaly scores compared to the boundaries, it may increase false positives. Such an issue is addressed by performing iterative boundary suppression before applying dilated smoothing. After the boundary regions are updated with neighboring non-boundary regions, dilated smoothing increases the overall performance without such error propagation.

5.2. Analysis

This section provides an in-depth analysis on the effects on segmentation performance, comparison with various backbones, and comparison on computational costs.

Model	Original	MSP	Max Logit	Ours
mIoU (%)	80.33	19.22	26.19	**68.65**

Table 4: mIoU on the Cityscapes validation set with the unexpected obstacle detection threshold at TPR_{95} on Fishyscapes Lost & Found validation set.

5.2.1 Effects on the segmentation performance

Table 4 shows the mIoU on the Cityscapes validation set with the detection threshold at TPR_{95}. By applying the detection threshold, the segmentation model predicts a non-trivial amount of in-distribution pixels as the unexpected ones. Due to such false positives, the mIoU of all methods decreased from the original mIoU of 80.33%. To be more specific, using MSP [16] and max logit [15] result in significant performance degradation. On the other hand, our approach maintains a reasonable performance of mIoU even with outstanding unexpected obstacle detection performance. This table again demonstrates the practicality of our work since it both shows reasonable performance in the segmentation task and the unexpected obstacle detection task.

Backbone	Models	mIoU	AUROC ↑	AP ↑	FPR_{95} ↓
MobileNet V2 [31]	MSP	75.70	86.00	2.60	48.05
	Max Logit		91.89	7.15	36.24
	Ours		**96.18**	**16.95**	**16.63**
ShuffleNet V2 [23]	MSP	72.71	86.33	4.06	45.68
	Max Logit		90.06	8.67	45.36
	Ours		**95.26**	**14.42**	**23.17**
ResNet50 [14]	MSP	77.76	86.25	3.50	45.03
	Max Logit		89.47	8.95	48.99
	Ours		**95.24**	**18.54**	**19.57**

Table 5: Comparison with MSP and max logit on Fishyscapes Lost & Found dataset. The backbone networks are trained with the output stride of 16.

5.2.2 Comparison with various backbones

Since our method does not require additional training or extra OoD datasets, our method can be adopted and used easily on any existing pre-trained segmentation networks. To verify the wide applicability of our approach, we report the performance of identifying anomalous objects with

various backbone networks including MobileNetV2 [31], ShuffleNetV2 [23], and ResNet50 [23]. As shown in Table 5, our method significantly outperforms the other approaches [16, 15] using the same backbone network with a large improvement in AP. This result clearly demonstrates that our method is applicable widely regardless of the backbone network.

Models	GFLOPs	Infer. Time (ms)
ResNet-101 [14]	2139.86	60.54
Ours (SML)	2139.86	61.41
Ours (SML + B Prop.)	2140.01	74.66
Ours (SML + B Prop. + D. Smoothing)	2140.12	75.02
SynthCP [33]	4551.11	146.90

Table 6: Comparison of computational cost. Metrics are measured with the image size of 2048×1024 on NVIDIA GeForce RTX 3090 GPU. The inference time is averaged over 100 trials.

5.2.3 Comparison on computational cost

To demonstrate that our method requires a negligible amount of computation cost, we report GFLOPs (*i.e.,* the number of floating-point operations used for computation) and the inference time. As shown in Table 6, our method requires only a minimal amount of computation cost regarding both GFLOPs and the inference time compared to the original segmentation network, ResNet-101 [14]. Also, among several studies which utilize additional networks, we compare with a recently proposed approach [33] that leverages an image resynthesis model. Our approach requires substantially less amount of computation cost compared to SynthCP [33].

Models	ΔAUROC $\uparrow$	ΔAP $\uparrow$	ΔFPR$_{95}$ $\downarrow$
MSP + B. Supp. + D. S.	-0.60	1.08	3.24
Max Logit + B. Supp. + D. S.	-0.51	-1.45	2.60
SML + B. Supp. + D. S.	**0.35**	**8.95**	**-0.93**

Table 7: Comparison of metric gains after iterative boundary suppression and dilated smoothing on MSP, max logit, and SML. B Supp. and D. S refer to iterative boundary suppression and dilated smoothing, respectively.

5.3. Effects of Standardized Max Logit

Table 7 describes how SML enables applying iterative boundary suppression and dilated smoothing. Applying iterative boundary suppression and dilated smoothing on other approaches does not improve the performance or even aggravates in the cases of MSP [16] and max logit [15]. On the other hand, it significantly enhances the performance when applied to SML. The following are the possible reasons for such observation. As aforementioned, the overconfidence of the softmax layer elevates the MSPs of anomalous objects. Since the MSPs of anomalous objects and in-distribution objects are not distinguishable enough, applying iterative boundary suppression and dilated smoothing may not improve the performance.

Additionally, iterative boundary suppression and dilated smoothing require the values to be scaled since it performs certain computations with the values. In the case of using max logits, the values of each predicted class differ according to the predicted class. Performing the iterative boundary suppression and dilated smoothing in such a case aggravates the performance because the same max logit values in different classes represent different meanings according to their predicted class. SML aligns the differently formed distributions of max logits which enables to utilize the values of neighboring pixels with certain computations.

6. Conclusions

In this work, we proposed a *simple yet effective* method for identifying unexpected obstacles on roads that do not require external datasets or additional training. Since max logits have their own ranges in each predicted class, we aligned them via standardization, which improves the performance of detecting anomalous objects. Additionally, based on the intuition that pixels in a local region share local semantics, we iteratively suppressed the boundary regions and removed irregular pixels that have distinct values compared to neighboring pixels via dilated smoothing. With such a straightforward approach, we achieved a new state-of-the-art performance on Fishyscapes Lost & Found benchmark. Additionally, extensive experiments with diverse datasets demonstrate the superiority of our method to other previous approaches. Through the visualizations and in-depth analysis, we verified our intuition and rationale that standardizing max logit and considering the local semantics of neighboring pixels indeed enhance the performance of identifying unexpected obstacles on roads. However, there still remains room for improvements; 1) dilated smoothing might remove unexpected obstacles that are as small as noises, and 2) the performance depends on the distribution of max logits obtained from the main segmentation networks.

We hope our work inspires the following researchers to investigate such practical methods for identifying anomalous objects in urban-scene segmentation which is crucial in safety-critical applications.

7. Acknowledgement

We deeply appreciate Hermann Blum and FishyScapes team for their sincere help in providing the baseline performances and helping our team to update our model on the FishyScapes Leaderboard.

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government(MSIT) (No. 2019-0-00075, Artificial Intelligence Graduate School Program(KAIST) and No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2019R1A2C4070420).

References

[1] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Semantic segmentation with boundary neural fields. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3602–3610, 2016.

[2] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Dense outlier detection and open-set recognition based on training with noisy negative images. *arXiv preprint arXiv:2101.09193*, 2021.

[3] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *arXiv preprint arXiv:1904.03215*, 2019.

[4] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. *arXiv preprint arXiv:2012.06575*, 2020.

[5] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. Hardnet: A low memory traffic network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3552–3561, 2019.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[7] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T. Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11580–11590, June 2021.

[8] Sungha Choi, Joanne T. Kim, and Jaegul Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 9373–9383, 2020.

[9] 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 *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3213–3223, 2016.

[10] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015.

[11] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3146–3154, 2019.

[12] Estevão S Gedraite and Murielle Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proc. of the International Symposium on Electronics in Marine (ELMAR)*, pages 393–396, 2011.

[13] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Dense open-set recognition with synthetic outliers generated by real nvp. *arXiv preprint arXiv:2011.11094*, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[15] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2020.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

[17] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7167–7177, 2018.

[18] Xin Li, Zequn Jie, Wei Wang, Changsong Liu, Jimei Yang, Xiaohui Shen, Zhe Lin, Qiang Chen, Shuicheng Yan, and Jiashi Feng. Foveanet: Perspective-aware urban scene parsing. In *Proc. of IEEE international conference on computer vision (ICCV)*, pages 784–792, 2017.

[19] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

[21] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting road obstacles by erasing them. *arXiv preprint arXiv:2012.13633*, 2020.

[22] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proc. of IEEE international conference on computer vision (ICCV)*, pages 2151–2161, 2019.

[23] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[24] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7047–7058, 2018.

[25] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

[26] Dong Nie, Jia Xue, and Xiaofeng Ren. Bidirectional pyramid networks for semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[27] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road obstacle detection method based on an autoencoder

with semantic segmentation. In *Proc. of the Asian Conference on Computer Vision (ACCV)*, pages 223–238, 2020.

[28] Marin Oršić and Siniša Šegvić. Efficient semantic segmentation with pyramidal fusion. *Pattern Recognition*, 110:107611, 2021.

[29] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106, 2016.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4510–4520, 2018.

[32] Chi Wang, Yunke Zhang, Miaomiao Cui, Jinlin Liu, Peiran Ren, Yin Yang, Xuansong Xie, XianSheng Hua, Hujun Bao, and Weiwei Xu. Active boundary loss for semantic segmentation. *arXiv preprint arXiv:2102.02696*, 2021.

[33] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L. Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 145–161, 2020.

[34] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3684–3692, 2018.

[35] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

[36] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8856–8865, 2019.

[37] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proc. of IEEE international conference on computer vision (ICCV)*, pages 593–602, 2019.