This ACCV 2020 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2020
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Road Obstacle Detection Method Based on an Autoencoder with Semantic Segmentation

Toshiaki OHGUSHI, Kenji HORIGUCHI, and Masao YAMANAKA

TOYOTA MOTOR CORPORATION,
Otemachi, Chiyoda-ku, Tokyo, 100-0004, JAPAN
{toshiaki_ohgushi, kenji_horiguchi, masao_yamanaka}@mail.toyota.co.jp

Abstract. Accurate detection of road obstacles is vital for ensuring safe autonomous driving, particularly on highways. However, existing methods tend to perform poorly when analyzing road scenes with complex backgrounds, because supervised approaches cannot detect unknown objects that are not included in the training dataset. Hence, in this study, we propose a road obstacle detection method using an autoencoder with semantic segmentation that was trained with only data from normal road scenes. The proposed method requires only a color image captured by a common in-vehicle camera as input. It then creates a resynthesized image using an autoencoder consisting of a semantic image generator as the encoder and a photographic image generator as the decoder. Extensive experiments demonstrate that the performance of the proposed method is comparable to that of existing methods, even without postprocessing. The proposed method with postprocessing outperformed state-of-the-art methods on the Lost and Found dataset. Further, in evaluations using our Highway Anomaly Dataset, which includes actual and synthetic road obstacles, the proposed method significantly outperformed a supervised method that explicitly learns road obstacles. Thus, the proposed machine-learning-based road obstacle detection method is a practical solution that will advance the development of autonomous driving systems.

1 Introduction

In recent years, advanced driving support systems have been rapidly developed to realize autonomous driving in the future. Human–machine interfaces linked with these systems will be able to support safe, secure, and comfortable driving by informing drivers about changes in the driving environment (e.g., due to traffic congestion, weather, and road obstacles) detected by preceding vehicles and passed on to subsequent vehicles.

According to a report by the Ministry of Land, Infrastructure, Transport, and Tourism in Japan [1], approximately 340,000 road obstacles were identified in 2018 (i.e., almost 1,000 road obstacles per day). Such obstacles regularly cause severe accidents. Therefore, the automation of road obstacle detection as a social system is urgently required because the detection and removing of these road obstacles are performed manually at present.

Several driving environment recognition methods have been proposed based on this background. However, these methods are not suitable for vehicles already on the market because they require special sensors, such as stereo cameras, LIDAR, and radar. Further, special sensors that can be used for autonomous driving are prohibitively expensive and require considerable power. In particular, a machine-learning-based approach is a potential alternative to special-sensor-based approaches. However, collecting a large amount of data required for supervised learning is impractical as the colors, shapes, sizes, and textures of road obstacles can vary substantially, as shown in Fig. 1. Thus, training a classifier to robustly detect road obstacles (i.e., unknown objects) is almost impossible.

In this paper, we propose a road obstacle detection method based on an autoencoder with semantic segmentation. The proposed method requires only a color image captured by a common in-vehicle camera as input. From this image, the method creates a resynthesized image using an autoencoder comprising a semantic image generator [2] as the encoder and a photographic image generator [3] as the decoder. The method then calculates the perceptual loss [3] between the input and resynthesized images and multiplies it by the entropy for the semantic image to generate an anomaly map. Finally, the method localizes road obstacles in the image by applying postprocessing to the anomaly map. Specifically, we sharpen the anomaly map by applying a standard technique for calculating the visual saliency in an image [4][5].

Through extensive experiments, we demonstrate that the performance of the proposed method is comparable with that of existing methods, even without postprocessing. Moreover, the proposed method with postprocessing outperforms state-of-the-art methods on one of the largest publicly available datasets [6]. Additionally, in our tests with the proposed highway dataset, which includes imagery with actual road obstacles, we show that the proposed method provides significant advantages over a supervised method that explicitly learns road obstacles using a semantic segmentation technique.

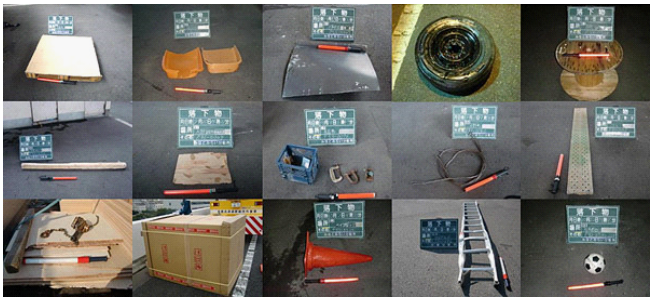

Fig. 1. Examples of road obstacles [7]. Although some obstacles are more common than others (e.g., burst tire debris, road cones, plywood, square lumber, and scrap iron), predicting exactly what might fall from a truck or a car on the road (e.g., a soccer ball) is impossible.

2 Previous Work

Early studies in the field of road obstacle detection in highway environments strongly relied on stereo vision techniques. For example, Hancock [8] used laser reflectance and stereo vision to detect small road obstacles at long distances. Similarly, William et al. [9] used a multibaseline stereo technique to detect small road obstacles (approximately 14 cm high) at a distance of over 100 m. Even relatively recent research uses stereo cameras or the structure-from-motion technique to detect road obstacles. For instance, Subaru Eyesight [10] is a representative stereo-vision-based system that robustly detects large road obstacles. In addition, Mobileye [11] is a commercially available system that can robustly detect large obstacles at close range using only a monocular camera. Further, Tokudome et al. [12] developed a novel real-time environment recognition system for autonomous driving using a LIDAR sensor.

However, these special-sensor-based approaches require a relatively clean road environment to compute image warping and disparity with high accuracy. In practice, vehicle vibrations render calibrating cameras with long focal lengths highly difficult because two cameras can move independently. Further, it is difficult to obtain high accuracy when using off-the-shelf active sensors over long distances. For example, a LIDAR system (e.g., Velodyne HDL-64E [13]) has a vertical angular resolution of approximately 0.4°. This implies that the maximum distance at which the system can detect only three consecutive points on a small 20-cm-high vertical object is less than 15 m. Although special-sensor-based approaches present several problems as described above, rich features can be extracted in the context of road obstacle detection, particularly when detecting small road obstacles from long distances.

Unlike special-sensor-based approaches, machine-learning-based approaches extract raw images using a common in-vehicle camera and convert the images into rich features by applying advanced machine learning techniques such as autoencoders [14][15], uncertainty-based approaches [16][17], and generative adversarial networks (GANs) [18][19]. In autoencoder-based approaches [14][15], small input patches are compared with the output from a shallow autoencoder trained on road textures only. In principle, road patches from other patches can be distinguished using these approaches. However, other patches include not only road obstacles (i.e., anomaly objects) but also normal objects, such as vehicles, traffic signs, and buildings. Therefore, these approaches yield a significant number of false positives. Uncertainty-based approaches [16][17] rely on the Bayesian SegNet framework and incorporate an uncertainty threshold to detect potentially mislabeled regions, including unknown objects. However, these approaches also yield numerous false positives in irrelevant regions (i.e., boundary regions at semantic labels, such as roads, vehicles, buildings, sky, and nature). In GAN-based approaches [18], an image is passed through an adversarial autoencoder, and then the feature loss between the output and input images is measured. These methods can be used to classify entire images, but not to identify anomalies within the images. Moreover, in GAN-based approaches [19], given a GAN trained to represent an original distribution, an algorithm searches for the latent

vector that yields the image that most closely matches the input. However, this is computationally expensive and does not identify anomalies.

A considerably different approach [20] has proven as a promising alternative to the existing techniques previously mentioned. This approach relies on the intuition that a network will yield false labels in regions that contain unexpected objects. Currently, this approach, hereinafter referred to as Resynth, obtains state-of-the-art results when tested on one of the largest publicly available road obstacle detection datasets [6]. Specifically, Resynth uses an existing semantic segmentation algorithm, such as [16] or [21], to generate a semantic map. It then passes this map to a generative network [22] that attempts to resynthesize an input image. If the image contains objects belonging to a class that the segmentation algorithm has not been trained to identify, then the corresponding pixels are mislabeled in the semantic map and, therefore, poorly resynthesized. Finally, Resynth identifies these unexpected objects by detecting significant differences between the original and synthetic images. Specifically, this method introduces a sophisticated neural network (i.e., a discrepancy network). It explicitly trains the discrepancy network to identify meaningful differences in the context of detecting unknown objects by replacing a few object instances with randomly selected labels in the ground-truth semantic map.

However, when trained in a supervised manner, this approach tends to perform poorly on images with complex backgrounds because the network only learns arbitrarily mislabeled semantic maps for normal objects instead of learning mislabeled semantic maps for unknown objects. Moreover, training the discrepancy network is not straightforward because the training process is quite difficult to perform in end-to-end manner.

3 Approach

Our basic idea is the same as that of [20]. However, our implementation is completely different from the existing method and rather reasonable. Specifically, our implementation has three types of components, namely, an autoencoder, an anomaly calculator, and a postprocessor, as shown in Fig. 2. First, the input to the autoencoder consists of only a color image (a) captured by a common in-vehicle camera. Subsequently, the autoencoder generates a semantic map (b) by applying a semantic segmentation technique [2] and creates a resynthesized image (c) using a photographic image synthesis technique [3]. Subsequently, the anomaly calculator generates an anomaly map (f) by multiplying the perceptual loss (d) and entropy (e) for the semantic map. Finally, the postprocessor outputs an obstacle score map (h) by sharpening the anomaly map for each local region (g). The details of these steps are described below.

3.1 Autoencoder

The autoencoder comprises modules for semantic segmentation and resynthesized image generation, as shown in Fig. 2. The autoencoder generates a semantic map and a resynthesized image and outputs them to the anomaly calculator

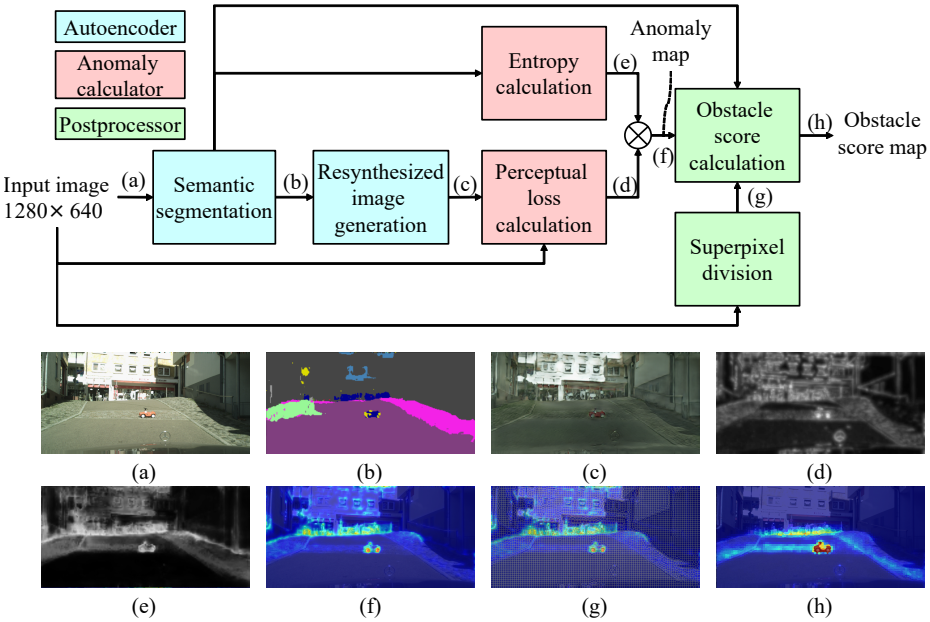

Fig. 2. Schematic overview of our road obstacle detection system. (a) Input image, (b) semantic map, (c) resynthesized image, (d) perceptual loss, (e) entropy map, (f) anomaly map, (g) superpixels, and (h) obstacle score map.

and postprocessor. In particular, we apply a representative semantic segmentation technique (ICNet [2]) to process the input image such that the module segments the input image into 20 types of semantic labels (e.g., road, car, traffic light, and traffic sign). Input images are obtained from Cityscapes, a publicly available dataset of road scenes for assessing and training vision algorithms [23]. Here, we downscale the Cityscapes training dataset to a resolution of 1,280 × 640 pixels owing to GPU memory constraints. Then, under fixed semantic segmentation model parameters, we concatenate the semantic segmentation module and resynthesized image generation module. Further, we apply an advanced resynthesized image generation technique (cascaded refinement network[3]) to process the semantic map; the module generates an image (i.e., the resynthesized image) that is exactly the same as the input image from the Cityscapes dataset [23]. Among the three types of components, only the autoencoder must learn the model parameters.

In particular, our algorithm can improve the quality of resynthesized images by employing a simple solution: connecting the decoder not to the output of the last layer (i.e., the softmax layer), but to the output of the intermediate layer (i.e., the convolution layer immediately before the softmax layer). Further, this solution performs well without additional functions such as the instance segmentation and instance level feature embedding required in Pix2PixHD [22]. Although Resynth trains the encoder and decoder completely separately owing

to heavy memory usage, our algorithm can realize end-to-end learning and rapid inference by concatenating light DNNs.

3.2 Anomaly calculator

The anomaly calculator comprises modules for entropy calculation and perceptual loss calculation, as shown in Fig. 2. The anomaly calculator generates an anomaly map that comprises an anomaly score at each pixel in the input image and then outputs this anomaly map to the postprocessor. Here, the following assumptions can be made when estimating the semantic labels of an unknown object: The semantic map contains ambiguity around the unknown object, and the resynthesized image yields significant differences in appearance with respect to the input image because of this ambiguity. Therefore, we calculate the entropy for the semantic map to measure the ambiguity and calculate the perceptual loss [3] to measure the differences in appearance. Finally, we define the product of these measures as the anomaly score. Specifically, we define the entropy for the semantic map as follows:

$$\mathcal{S} = U_{bl}\left(-\sum_{k} p^{(k)} \log(p^{(k)})\right). \tag{1}$$

Here, $p^{(k)}$ is the probability of the k-th semantic label estimated using the semantic segmentation technique [2] and U_{bl} is a bilinear-interpolation-based upconverter that upconverts the resynthesized image to the same resolution as the input image. Further, we define the perceptual loss between the input and resynthesized images as follows:

$$\mathcal{L} = \sum_{l=1}^{5} U_{bl}\left(\mathcal{L}^{(l)}\right), \tag{2}$$

$$\mathcal{L}^{(l)} = ||\Phi^{(l)}(I) - \Phi^{(l)}(R)||_1. \tag{3}$$

Here, I and R are the input and resynthesized images, respectively. In addition, $\Phi^{(l)}$ is the output from the l-th hidden layer of VGG19 [24]. Specifically, $\Phi^{(l)}(l = 1, \ldots, 5)$ are given by the outputs from conv1_2, conv2_2, conv3_2, conv4_2, and conv5_2, as shown in Fig. 3. Thus, we obtain the output from each hidden layer using VGG19 on the input and resynthesized images. Additionally, we define the L1 norm between the output from the l-th hidden layer for the input image and the output from the l-th hidden layer for the resynthesized image as perceptual loss $\mathcal{L}^{(l)}$, as shown in Eq. (3). Further, we calculate the total perceptual loss $\mathcal{L}$ by adding perceptual loss $\mathcal{L}^{(l)}(l = 1, \ldots, 5)$ after adjusting its resolution with upconverter U_{bl}, as shown in Eq. (2). Finally, we generate anomaly map $\mathcal{A}$ by taking the element-wise product of perceptual loss $\mathcal{L}$ and entropy $\mathcal{S}$ as follows:

$$\mathcal{A} = \mathcal{L} \odot \mathcal{S}. \tag{4}$$

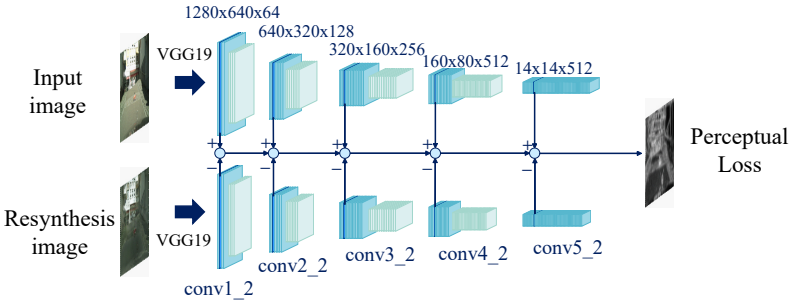

Fig. 3. Schematic overview of our perceptual loss calculation module. We apply VGG19 [24] to the input and resynthesized images. The blue circle comprises two functions. The first function calculates the channel-wise L1 norm for the difference between the output from the i-th hidden layer for the input image and the output from the i-th hidden layer for the resynthesized image. The second function applies upconverter U_{bl} to the map composed of the L1 norm values.

3.3 Postprocessor

The postprocessor consists of modules for obstacle score calculation and superpixel division, as shown in Fig. 2. Its aim is to generate an obstacle score map to localize unknown objects in the input image. First, we use simple linear iterative clustering to segment the input image into local regions referred to as superpixels [25]. We perform this because superpixels are less likely to cross object boundaries, which leads to greater accuracy in segmentation of visually salient regions. Then, we define the obstacle score in the i-th superpixel as follows:

$$L_i = \alpha_i \sum_j n_j p_j \exp\left(-\frac{r_{i,j}^2}{2w^2}\right). \tag{5}$$

where α_i is the average value of the anomaly score in the i-th superpixel, n_j is the number of pixels in the j-th superpixel, p_j is the average value of the probability for the road label in the j-th superpixel, $r_{i,j}$ is the Euclidean distance between the center position of the i-th superpixel and the center position of the j-th superpixel, and w is the median of the Euclidean distances between the center positions of every pair of superpixels. Finally, the regions in which L_i exceeds a predetermined threshold are identified as those containing an unknown object (i.e., a road obstacle).

4 Experiments

Using two separate datasets, we evaluated the ability of our method to detect road obstacles. We did not use any prior knowledge about road obstacles during training because our focus is on finding unknown anomaly objects.

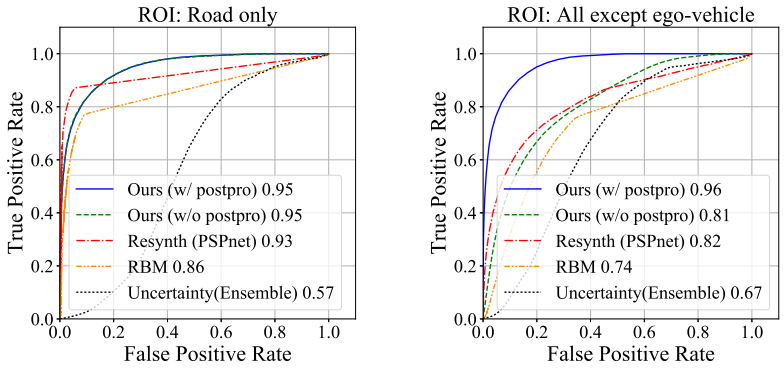

Fig. 4. ROC curves and AUROC scores for the Lost and Found dataset [6]. Resynth (PSPnet) depicts the results of the method reported in [20] using PSPnet [21] as the semantic segmentation technique. Uncertainty (Ensemble) depicts the results of the ensemble-based method reported in [17].

4.1 Lost and Found Dataset

First, we quantitatively evaluated our road obstacle detection method on a publicly available dataset, "Lost and Found" [6]. Instead of using a bounding box to mark the road obstacle region, accurate human-marked labels are provided as the ground truth in this public dataset. We followed a general methodology [20] to evaluate the accuracy of the detected road obstacle region. Specifically, all evaluated methods output a pixel-wise anomaly score. We compared the resulting maps with the ground-truth anomaly annotations using ROC curves and the AUROC (area under the ROC curve) metric. We considered representative existing methods (Resynth [20], a restricted Boltzmann machine [15], and an uncertainty-based method (Uncertainty) [17]) as baselines. Further, we evaluated our approach using obstacle score maps (i.e., with postprocessing) and using only anomaly maps (i.e., without postprocessing).

The ROC curves and AUROC scores obtained using these methods are shown in Fig. 4. The curves on the left were obtained by restricting the evaluation to the road, as defined by the ground-truth annotations. Similarly, the curves on the right were computed over the entire images, excluding the ego-vehicle regions only. The performance of our approach without postprocessing is comparable to that of Resynth [20] and superior to that of the other methods. Moreover, our approach with postprocessing achieves the highest AUROC scores among all methods. In particular, the road obstacle detection performance is substantially improved by applying postprocessing, as shown in Fig. 4.

Figure 5 shows an example of maps generated for an image with a road obstacle, which is captured primarily in the middle of the road (a). The semantic segmentation module outputs false labels in the road obstacle region (b). Then, the resynthesized image generation module obtains an image with significant

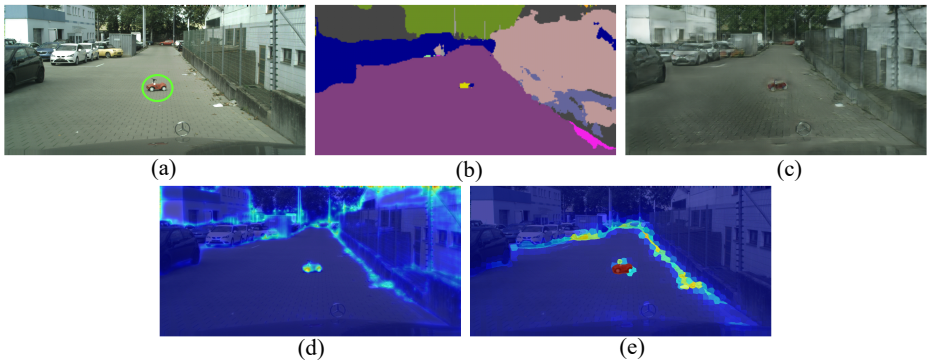

Fig. 5. Example of maps generated for a synthetic image with a road obstacle. (a) Input image, (b) semantic map, (c) resynthesized image, (d) anomaly map, and (e) obstacle score map.

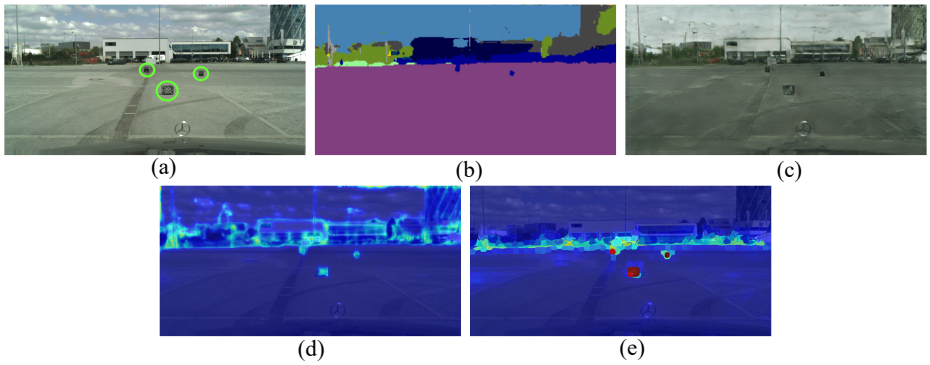

Fig. 6. Example of maps generated for a synthetic image with multiple road obstacles. (a) Input image, (b) semantic map, (c) resynthesized image, (d) anomaly map, and (e) obstacle score map.

differences in appearance with respect to the road obstacle region in the input image (c). Thus, the anomaly calculator generates relatively high anomaly scores around the road obstacle (d). Finally, the road obstacle is highlighted by the postprocessor, which enhances the anomaly score only around the road and suppresses the anomaly score in other regions (e).

The yellow car parked sideways causes a misclassification during semantic segmentation. Such misclassifications decrease the quality of resynthesized images as well, and these may lead to false positives in road obstacle detection.

Figure 6 shows an example of maps generated for an image that contains multiple obstacles (a). Although relatively small obstacles are captured, the anomaly score accounts for them accurately (e). For this case as well as the above, we can observe that misclassifications caused around obstacle regions (b); and it leads to poor quality of resynthesis (c).

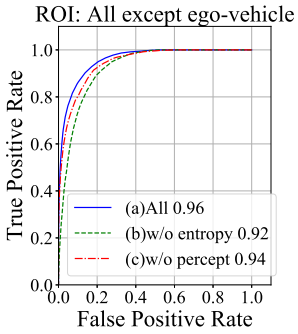

Fig. 7. ROC curves and AUROC scores for ablation experiments.

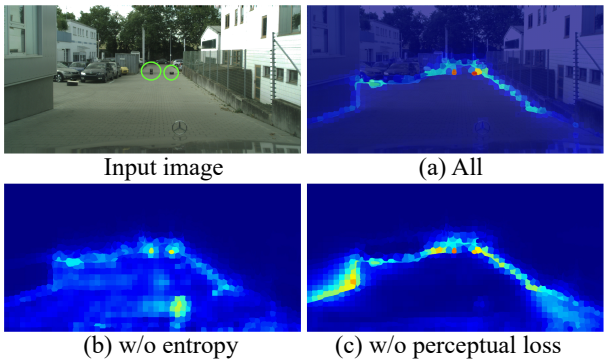

Fig. 8. Output image examples for ablation experiments.

Ablation experiment To investigate how each component contributes to anomaly detection performance, ablation experiments were conducted. Fig. 7 shows the ROC curve and AUC scores of model (a), which contains all modules shown in Fig 2; model (b), from which the entropy calculation module has been removed; and model (c), from which the perceptual loss calculation module has been removed.

The results show that model (c) outperforms model (b). This could result from the fact that perceptual loss tends to produce relatively more false negatives(Fig. 8(b)), as perceptual loss responds to a corrupt portion of the resynthesized image sensitively and produces blurred score maps owing to the lower resolution of the latter layers of VGG (i.e., conv4_2, conv5_2). Meanwhile, the entropy score calculated from semantic segmentation labels can catch the edges of class boundaries. However, the entropy score tends to be high even in well-resynthesized areas, and the perceptual loss score is low (e.g., the corner of the building on the left in Fig. 8(c)).

By multiplying the perceptual loss map and entropy map, an improved anomaly score that reflects the benefits of both maps can be acquired (Fig. 8(a)). Actually, we can verify that the best AUC score is obtained when all components are used.

4.2 Our Highway Anomaly Dataset

We quantitatively evaluated our road obstacle detection method in tests using our highway anomaly dataset, which is shown in Fig. 9. The dataset is composed of (a) a training dataset captured under normal highway driving conditions without road obstacles, (b) a validation dataset, and (c) a test dataset; the latter two datasets include actual and synthetic road obstacles, such as traffic cones and objects falling from other vehicles. The respective datasets include approximately 5000, 300, and 200 photo and segmentation ground truth image pairs.

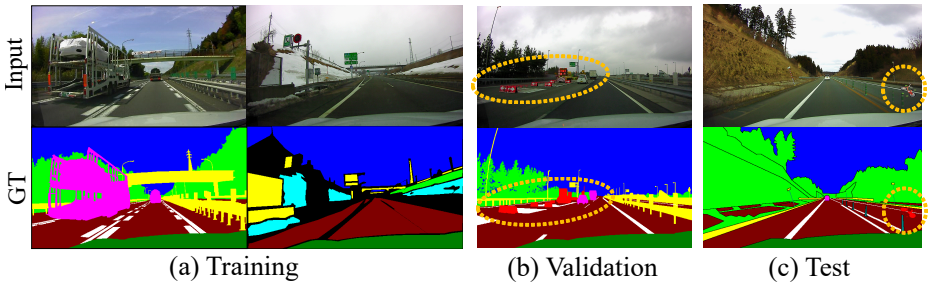

Fig. 9. Examples from our Highway Anomaly Dataset.

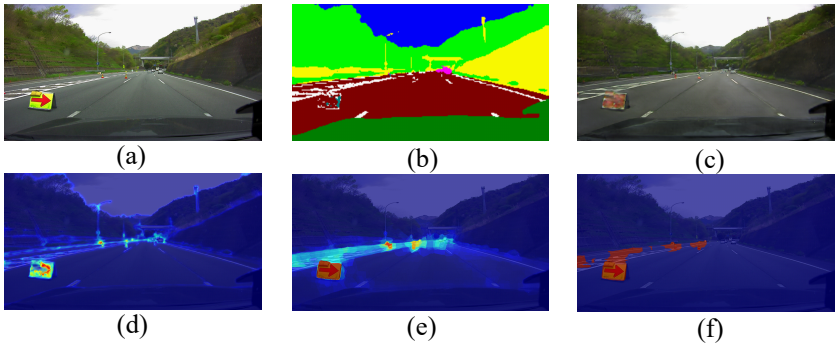

Fig. 10. Example of maps generated for a synthetic image. (a) Input image, (b) semantic map, (c) resynthesized image, (d) anomaly map, (e) obstacle score map, and (f) detection result.

First, we trained our autoencoder with the Cityscapes dataset and fine-tuned it using our training dataset. The validation dataset was then used for determining the threshold for detecting road obstacle areas. Finally, we evaluated the performance using the test dataset and the threshold described above.

Figure 10 shows an example of maps generated for a synthetic image (a), in which a warning sign can be observed. The semantic segmentation module outputs false labels in the road obstacle region (b). Then, the resynthesized image generation module obtains an image with significant differences in appearance with respect to the road obstacle region in the input image (c). Hence, the anomaly calculator generates relatively high anomaly scores around the road obstacle (d). Further, the postprocessor sharpens the anomaly score around the road and suppresses the anomaly score in other regions, such as the boundaries between the trees and sky in the distance (e). Finally, we identify the region wherein the obstacle score $\mathcal{L}$ exceeds the predetermined threshold $\mathcal{T}$ as the road obstacle (f). We determined the threshold $\mathcal{T}$ using 300 validation images (different from the abovementioned 200 test images) including actual and synthetic road obstacles, such that the F-measure of the detected road obstacle regions was maximized.

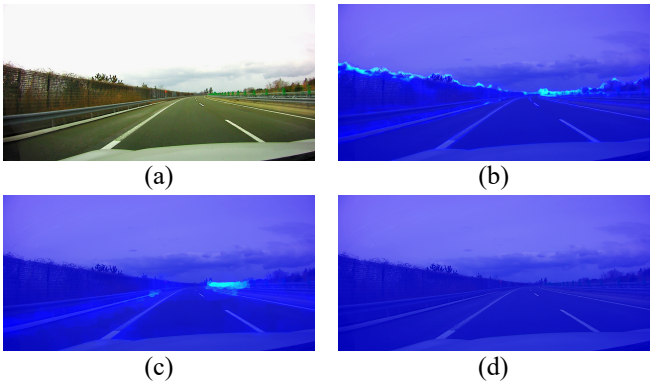

Fig. 11. Example of maps generated for an image without road obstacles. (a) Input image, (b) anomaly map, (c) obstacle score map, and (d) detection result.

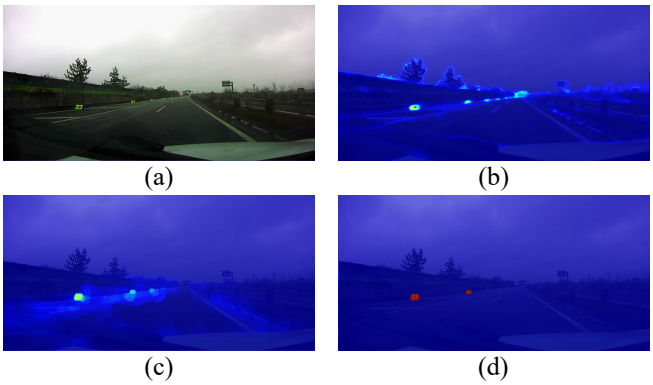

Fig. 12. Example of maps generated for an image containing small road obstacles. (a) Input image, (b) anomaly map, (c) obstacle score map, and (d) detection result.

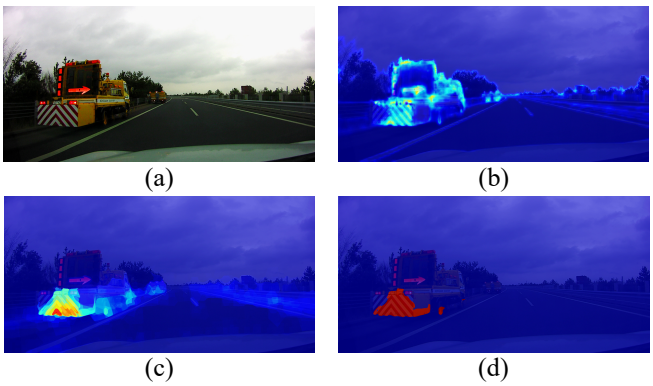

Fig. 13. Example of maps generated for an image containing large road obstacles. (a) Input image, (b) anomaly map, (c) obstacle score map, and (d) detection result.

Table 1. Performance comparison.

Evaluation Metric	w/ explicit learning	w/o explicit learning		
	ICNet	GAN Resynth	Ours	
			w/o post-processing	w/ post-processing
mean F-measure	0.142	0.103	0.231	0.300
global F-measure	0.197	0.040	0.333	0.452
mean IoU	0.092	0.063	0.154	0.219
global IoU	0.109	0.020	0.200	0.292

Figure 11 shows an example of maps generated for an image without road obstacles, which can be observed as a normal driving environment on the highway (a). The anomaly map has relatively high scores around the boundaries between the trees and road in the input image (b). However, the anomaly scores are suppressed by the postprocessing of the anomaly map (c). Consequently, our approach succeeds in obtaining the true negative as an entire image by thresholding the obstacle score map (d). Our approach yields almost no false positives in normal driving environments, such as that shown in (a).

Figure 12 shows an example of maps generated for an image containing road obstacles, which are warning signs that are temporarily arranged at equal intervals on the road (a). The anomaly map can detect extremely small road obstacles in the distance (b). Further, even if the anomaly scores are suppressed by applying postprocessing for the anomaly map (c), our approach succeeds in detecting small road obstacles at a distance (d).

Figure 13 shows an example of maps generated for an image containing large road obstacles; in this case, the obstacles are sections of a large emergency vehicle temporarily parked on the roadside (a). The emergency vehicle in the image consists of a truck, a lift attached to a base, and a warning sign attached to the truck. The truck should be recognized as a normal object, whereas the lift and warning sign should be detected as anomalies (i.e., road obstacles). The anomaly map succeeds in obtaining relatively high scores around the lift and warning sign (b). However, the anomaly scores for the warning sign are suppressed by the postprocessing of the anomaly map (c). Therefore, our approach fails to discriminate the warning sign from the road, although it succeeds in detecting the lift close to the road (d). In principle, it is quite difficult for our approach to detect road obstacles that are not on roads.

Finally, we compared the performance of four different approaches: ICNet [2], Resynth [20], our approach with postprocessing, and our approach without postprocessing. Specifically, we explicitly trained the ICNet to learn road obstacles using the above validation images. Table 1 compares the performance of these four approaches. Our approach outperforms ICNet [2] and Resynth [20], even without postprocessing, as shown in Table 1. Here, the mean F-measure is the average of the F-measures calculated for each test image, the global F-measure

indicates the F-measure calculated using all test images, the mean Intersection-over-Union (IoU) indicates the average value of the IoUs calculated for each test image, and the global IoU indicates the IoU calculated using all test images.

The processing time required for creating an obstacle score map composed of 1,280 × 640 pixel images was approximately 1 s when using a Tesla V100 equipped with 16.0 GB RAM. Regarding the processing time for each component in our system, the autoencoder required 72.5 [ms] (13.8 fps), the anomaly calculator required 1,053 ms (0.95 fps), and the postprocessor required 667 ms (1.5 fps). This observation indicates that the computation time should be improved, particularly for the anomaly calculator. This remains an issue for further research.

5 Conclusion

In this study, we proposed a road obstacle detection method based on an autoencoder with semantic segmentation. The proposed method is purely unsupervised; therefore, it does not require any prior knowledge of road obstacles. In particular, the method requires only a color image captured by a common in-vehicle camera as input. The method creates a resynthesized image using an autoencoder composed of a semantic image generator as the encoder and a photographic image generator as the decoder.

Subsequently, the method calculates the perceptual loss between the input and resynthesized images and multiplies the perceptual loss by the entropy for the semantic image to generate an anomaly map. Finally, the method localizes a road obstacle in the image by applying visual-saliency-based postprocessing to the anomaly map.

In particular, the method can improve the quality of resynthesized images by employing a simple solution: connecting the decoder not to the output of the last layer (i.e., the softmax layer) but to the output of the intermediate layer (i.e., the convolution layer immediately before the softmax layer). Moreover, this solution performs well without additional functions, such as instance segmentation or instance level feature embedding. Although the existing method must train the encoder and decoder completely separately owing to heavy memory usage, it can realize end-to-end learning and rapid inference by concatenating light DNNs.

Through extensive experiments, we demonstrated that the performance of the proposed method is comparable to that of existing methods, even without postprocessing. Additionally, the proposed method with postprocessing outperforms state-of-the-art methods on one of the largest publicly available datasets. Further, in evaluations using our Highway Anomaly Dataset containing actual and synthetic road obstacles, the proposed method significantly outperformed a supervised method that explicitly learns road obstacles using a semantic segmentation technique. This unsupervised machine-learning-based road obstacle detection method is a practical solution that will advance the development of autonomous driving systems.

References

1. Ministry of Land, Infrastructure, Transport and Tourism: Number of fallen objects handled by expressway companies in 2018. (https://www.mlit.go.jp/road/sisaku/ijikanri/pdf/h30rakkabutu_nexco.pdf)
2. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: ICNet for real-time semantic segmentation on high-resolution images. In: The European Conference on Computer Vision (ECCV). (2018)
3. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. 2017 IEEE International Conference on Computer Vision (ICCV) (2017) 1520–1529
4. Goferman, S., Zelnik-Manor, L., Tal, A.: Context-aware saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2010) 2376–2383
5. Masao, Y.: Salient region detection by enhancing diversity of multiple priors. IPSJ Transactions on Mathematical Modeling and Its Applications **9** (2016) 13–22
6. Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., Mester, R.: Lost and found: detecting small road hazards for self-driving vehicles. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2016) 1099–1106
7. Shutoko; Metropolitan Expressway Company Limited: Current state of road obstacles. (http://www.shutoko.jp/use/safety/emergency)
8. Hancock, J.: High-speed obstacle detection for automated highway applications. Technical report, CMU Technical Report (1997)
9. Williamson, T., Thorpe, C.: Detection of small obstacles at long range using multibaseline stereo. In: IEEE International Conference on Intelligent Vehicles. (1998)
10. SUBARU: EyeSight. (http://www.subaru.com/engineering/eyesight.html)
11. Yoffie, D.B.: Mobileye: The Future of Driverless Cars. HBS CASE COLLECTION. Harvard Business School Case (2015)
12. Tokudome, N., Ayukawa, S., Ninomiya, S., Enokida, S., Nishida, T.: Development of real-time environment recognition system using lidar for autonomous driving. (2017) 1–4
13. Velodyne: HDL-64E. (http://velodynelidar.com/lidar/)
14. Munawar, A., Vinayavekhin, P., Magistris, G.D.: Limiting the reconstruction capability of generative neural network using negative learning. 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP) (2017) 1–6
15. Creusot, C., Munawar, A.: Real-time small obstacle detection on highways using compressive rbm road reconstruction. 2015 IEEE Intelligent Vehicles Symposium (IV) (2015) 162–167
16. Alex Kendall, V.B., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In Tae-Kyun Kim, Stefanos Zafeiriou, G.B., Mikolajczyk, K., eds.: Proceedings of the British Machine Vision Conference (BMVC), BMVA Press (2017) 57.1–57.12
17. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems 30. Curran Associates, Inc. (2017) 6402–6413

18. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training (2018)
19. Schlegl, T., Seeböck, P., Waldstein, S., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. (2017) 146–157
20. Lis, K., Nakka, K., Fua, P., Salzmann, M.: Detecting the unexpected via image resynthesis. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
21. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)
22. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018)
23. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. (2015)
25. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels (2010)