Improving Nighttime Curb Segmentation with Domain Translation

Gyeongho Cho[1] · Changsun Ahn[1] · Juhui Gim[*2]

[1]*School of Mechanical Engineering, Pusan National University, Busan 46241, Korea*
[2]*Department of Robot, Control and Instrumentation Engineering, Changwon National University, Gyeongnam 51140, Korea*

Abstract : This research sought to address the crucial challenge of enhancing nighttime curb segmentation in autonomous driving systems. The study delved into the limitations of using cameras, especially in low-light conditions, and its effects on image semantic segmentation and curb detection. To overcome these challenges, a camera-based method was proposed, leveraging both synthetic day images and real images for domain transfer. The algorithm comprised a dedicated generator network for translation and a segmentation network trained with a conventional loss function. The results of the experiment demonstrated a 124 % improvement in curb segmentation performance by F1 score, along with a 24 % increase in precision, compared to the benchmark. The findings underscored the method's significant potential in augmenting nighttime curb segmentation. This approach would be poised to contribute substantially to the development of safer autonomous vehicles that are equipped with heightened perception capabilities.

Key words : Autonomous vehicle, Curb detection, Contrastive unpaired translation, Domain translation, Semantic segmentation

1. Introduction

With the increasing interest and investment in autonomous driving system, research on autonomous vehicle technology is advancing. Research on autonomous vehicle can be divided into generally four parts: perception, localization, decision-making, and control.[1-4] Perception is the process of recognizing the vehicle's surrounding situation to understand its current state. Localization measures the vehicle's current position directly or indirectly. Decision-making involves determining what actions to take in the current state, and control is the process of executing those actions. Among these, perception is the first and crucial step in autonomous driving, and the accuracy of perception can significantly impact the other three functions.[5] In extreme cases, inaccurate perception may lead to serious accidents. Therefore, accurate and robust perception is essential for autonomous driving system.[6]

In perception for autonomous driving, environmental sensors

measuring the vehicle's surroundings and inertial sensors measuring the vehicle's motion states are utilized. Environmental sensors commonly include cameras, LiDAR, and radar, while inertial sensors include an Inertial Measurement Unit(IMU). Cameras, being the most informative sensor, play a fundamental role, and semantic segmentation is a primary method for recognizing objects from camera images.[7,8] This method can detect curbs, lane markings, road signs, traffic signals, pedestrians, and other vehicles. Curb detection is crucial as it signifies the boundary of the roadway, ensuring that the autonomous vehicle remains in the appropriate region.

There are various methods for detecting curbs using cameras,[9-15] classified into filter-based[9-12] and deep learning-based approaches.[13-15] Filter-based methods require engineer intuition, demanding significant effort and time. In contrast, deep learning-based methods require less effort for design and offer superior performance.

There is a critical issue in using camera images for curb

*Corresponding author, E-mail: juhuigim@changwon.ac.kr

(a) day   (b) night

Fig. 1 Curb in images of (a) daytime and (b) nighttime

detection. Particularly at night, when the visibility of camera images is lower than during the day, the visibility is poor, as shown in Fig. 1 where the red boxes indicate the curbs. Due to the insufficient lighting in nighttime scenes, information within the images decreases, leading to a reduction in object distinguishability. This adversely affects the performance of image semantic segmentation, increasing the risk of accidents for autonomous vehicles operating at night. While lane markings are often designed to be easily visible in nighttime driving environments through the use of special coatings, curbs are typically located in areas not illuminated by vehicle headlights. Additionally, the visibility of curbs in camera images is further compromised as the emphasis is on spatial distinctiveness rather than visibility. Consequently, it has been observed that the detection performance of curbs significantly declines during image segmentation in nocturnal environments. Although methods utilizing LiDAR in addition to camera images have been proposed to enhance curb detection performance at night,[16-21] they have not been widely adopted in production vehicles yet due to the cost and maintenance considerations associated with LiDAR. Thus, image-based curb detection remains crucial as it provides a more affordable and lower-maintenance solution, despite the higher perception performance offered by LiDAR.

This study introduces a curb detection algorithm that combines semantic segmentation with a domain translation technique aimed at enhancing the visibility of nighttime images, all without relying on less practical and costly sensors. The domain translation technique, designed using the Contrastive Unpaired Translation(CUT)[22] approach, effectively minimizes the domain gap between night and day scenes.

The subsequent sections of the paper are organized as follows: Chapter 2 delves into the foundational techniques employed in crafting the proposed algorithm. Chapter 3

provides a detailed explanation of the algorithm. Chapter 4 presents the evaluation results of the algorithm. Finally, Chapter 5 summarizes the conclusions drawn from the study.

2. Background Knowledge

Our primary approach for proposing a segmentation algorithm involves combining a segmentation technique with a domain translation technique to enhance curb detection performance at night. To elaborate on the details of our approach, we introduce the background research for the proposed method.

2.1 PSPNet for Segmentation

Semantic image segmentation entails classifying pixels in an image based on semantic categories, essentially assigning each pixel a label corresponding to its semantic content. Typically learned through supervised learning with images and corresponding labels, research in semantic image segmentation often centers on determining the model architecture and effective training methods. Convolutional Neural Networks(CNNs), well-known for feature extraction in images, are commonly employed in this context.[23-25] Recently, there has been significant attention on methods utilizing Vision Transformers (VIT),[26-28] known for their superior performance compared to CNNs. However, VIT poses challenges in training as it requires a substantial amount of data.

Meanwhile, Pyramid Scene Parsing Network(PSPNet)[25] is renowned as one of the popular image semantic segmentation networks. PSPNet consists of an encoder and decoder, utilizing different sizes of average pooling between them. This design allows PSPNet to consider both large-scale and small-scale features. Therefore, we employed PSPNet for segmentation in the proposed algorithm.

2.2 CUT for Domain Translation

Domain translation is a task involving the transfer of images from one specific domain to another. Numerous studies have aimed to enhance performance by modifying various aspects such as the training structure, loss functions, models, and input data. Image-to-image domain translation is typically categorized into paired translation and unpaired translation, commonly trained using the

Generative Adversarial Network(GAN) framework.[29]

When a sufficient amount of paired data is available, establishing a clear relationship between the source and target domains allows for the existence of correct answers for domain translation. An influential learning method for paired translation is Pix2Pix,[30] which applies the conditional GAN idea to image-to-image translation.[31] Pix2Pix employs a generator that takes input data from the source domain and a discriminator that considers both target and source domain data, maximizing paired characteristics to enhance learning performance.

In the case of unpaired data, where no clear correlation between the source and target domains exists, research has focused on defining relationships before and after domain translation to improve performance. A representative example is CycleGAN,[32] which utilizes two generators for transforming from A to B and B to A, along with discriminators for each domain. The loss function is designed based on the premise that when A is transformed into B and then reconstructed back into A, the results should be similar.

However, the loss function of CycleGAN is often too strict, imposing limitations on performance improvement. To address this issue, CUT was developed. CUT applies contrastive learning to unpaired image-to-image translation, defining the loss function at the feature level rather than the pixel level, thereby enhancing learning performance.

In this study, where the source domain is daytime and the target domain is nighttime, each with unpaired data, the structure of CUT was modified to determine the training structure.

3. Proposed Method

We propose a new segmentation algorithm for enhancing nighttime curb segmentation performance. The algorithm design process consists of an algorithm architecture design and a training architecture design. The algorithm architecture is designed to be implemented as a detection function of autonomous vehicles. The training architecture is designed for effective training the algorithm network.

3.1 Algorithm Architecture

The algorithm architecture comprises a domain translation network and a segmentation network, as
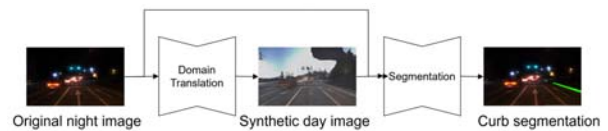

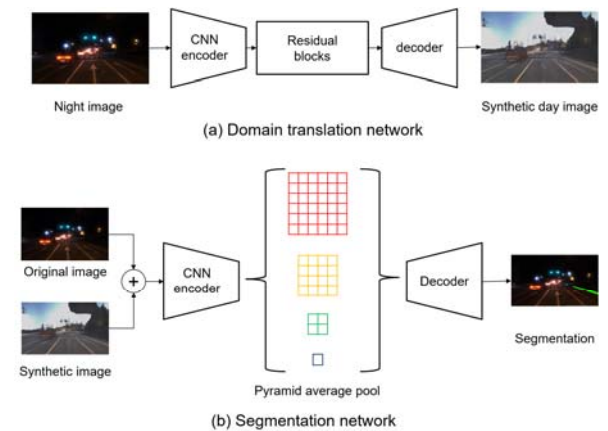Fig. 2 Algorithm architecture of proposed method

Fig. 3 Network structure for (a) domain translation and (b) segmentation

illustrated in Fig. 2. The domain translation network aims to enhance the visibility of nighttime images by transforming the input images from the night domain to the day domain. The domain translation network is designed with typical encoder-decoder structure with residual blocks, as depicted in Fig. 3(a). The synthetic image generated by the domain translation network exhibits higher visibility than the original input image.

The segmentation network functions by processing the concatenation of the original image and the synthetic image as its input. Including the original image in the input prevents the segmentation network from losing crucial information, thereby ensuring a performance level at least equivalent to conventional methods. The segmentation network is structured with an encoder-decoder architecture featuring a pyramid average pool network, as depicted in Fig. 3(b). The output of the segmentation network involves pixel-level classification with two classes, determining whether a pixel corresponds to a curb or not.

3.2 Network Training Method

To train the networks within the algorithm architecture, we propose a new training method illustrated in Fig. 4. The architecture consists of four networks: Generator (*G*),
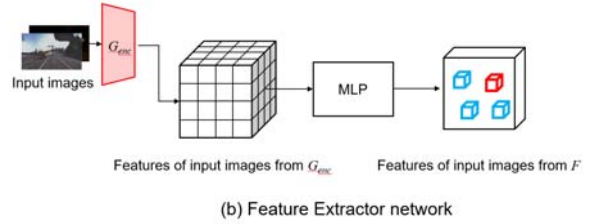
(a) Stage 1

(b) Stage 2

Fig. 4 Training architecture (a) Stage1 for training the segmentation network and (b) Stage 2 for training domain translation network

Feature extractor (F), Discriminator (D), and Segmentation (S) networks. G_{enc}, the encoder part of G, is used to extract some features from synthetic images. In this proposed structure, the generator performs domain transition, and its network is trained using the loss function adopted from the CUT method.

The training process consists of two stages: *Stage* 1 and *Stage* 2, each executed in turn and iteratively. In *Stage* 1, the discriminator and the segmentation network are trained, while the generator remains frozen, as depicted in Fig. 4(a). The discriminator evaluates both authentic daytime domain images and synthetic daytime images

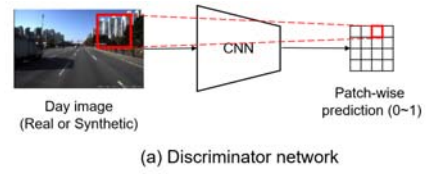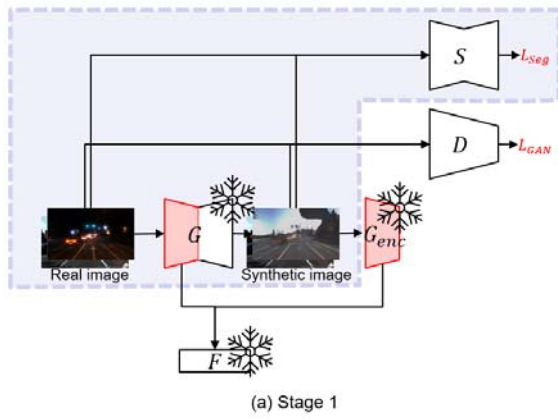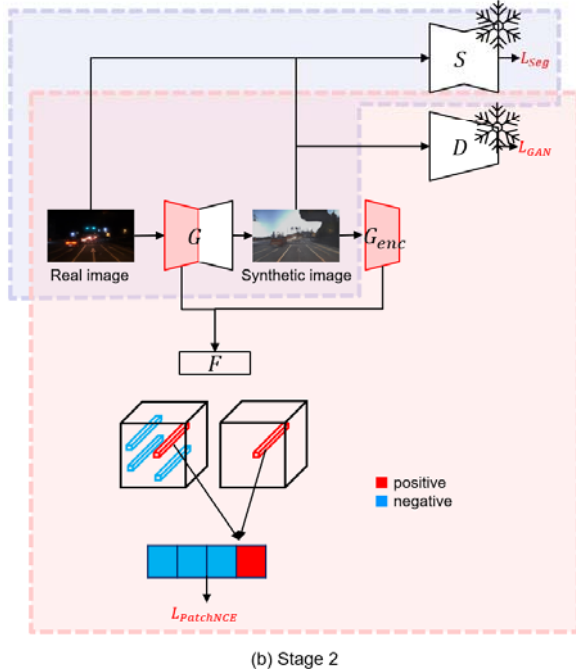(a) Discriminator network

(b) Feature Extractor network

Fig. 5 Functions of (a) discriminator (b) feature extractor

generated by the generator, determining whether the input is genuine or not, as illustrated in Fig. 5(a). Meanwhile, the segmentation network handles the original nighttime image and its corresponding synthetic daytime image, generated from the nighttime images. When provided with a daytime image, the segmentation network takes an array that concatenates two original daytime images as its input. The total loss of *Stage* 1 is given by:

$$L_{stage1} = w_{adv}L_{adv}(G,D,X,Y) + w_{seg}L_{seg}(G,S,X,Y,C), \qquad (1)$$

where w_{adv} and w_{seg} are weighting factors for the adversarial GAN loss L_{adv} and the segmentation loss L_{seg}, respectively. X refers the dataset of nighttime, Y stands for the dataset of daytime, C means the segmentation labels for given datasets.

The adversarial GAN loss is expressed in the form of Binary Cross-Entropy (BCE) as:

$$L_{adv}(G,D,X,Y) = \underset{y \sim Y}{E}\left(\log D(y)\right) + \underset{x \sim X}{E}\left(1 - \log D(G(x))\right). \qquad (2)$$

The segmentation loss is defined using Cross-Entropy (CE), as follows:

$$L_{seg}(G,S,X,Y,C) =$$
$$\underset{\substack{x \sim X \\ c_x \sim C_x}}{E} CE(S(x,G(x)),c_x) + \underset{\substack{y \sim Y \\ c_y \sim C_y}}{E} CE(S(y,y),c_y). \qquad (3)$$

It is noteworthy that the blue-boxed part of the training architecture of *Stage* 1 shown in Fig. 4(a) is exactly the same as the algorithm architecture. Therefore, the

segmentation network is trained similarly to the algorithm architecture.

In *Stage* 2, the generator and feature extractor are trained, while the discriminator and segmentation network remain frozen. Fig. 4(b) illustrates the training architecture of *Stage* 2. The total loss of *Stage* 2 is defined as follows:

$$
\begin{aligned}
L_{stage2} &= w_{adv}L_{adv}(G,D,X,Y) + w_{con}L_{con}(G,F,X) \\
&\quad + w_{seg}L_{seg}(G,S,X,Y,C),
\end{aligned}
\tag{4}
$$

where L_{con} is the patch noise contrastive estimation (PatchNCE) loss[22] and w_{con} is the corresponding weighting factor.

The PatchNCE loss is crucial in the context of CUT. PatchNCE focuses on feature-level similarity rather than pixel-level similarity between real and synthetic images. Consequently, it offers a less stringent definition of loss for image comparison, making it suitable for reconstruction loss in unpaired image-to-image translation. As shown in Fig. 5(b), a feature extractor is required to compare two images at the feature level.

It is also worthwhile to mention that the red boxed part of the training architecture of *Stage* 2, as shown in Fig. 4(b), is exactly the same as the training architecture of CUT. The generator is trained similarly to CUT for domain transition, and we incorporated segmentation loss into the CUT architecture to account for segmentation performance in image generation.

The objective of overall training process including *Stages* 1 and 2 is as follows:

$$
\min_{\theta_G,\theta_S,\theta_F} \max_{\theta_D} \{L_{stage1} + L_{stage2}\},
\tag{5}
$$

where θ_G, θ_S, θ_F, and θ_D network parameters of G, S, F and D, respectively. The overall training flow of the proposed method is shown in Table 1.

3.3 Dataset and Training Result

The "Autonomous Driving 3-D Image under Unusual Environmental Conditions" dataset was acquired from AI-Hub.[33] Comprising 11,763,600 data points, it includes front-view images, LiDAR point clouds, 2D segmentation annotations, bounding boxes of 3D objects, and additional environmental information such as weather. A subset of the

Table 1 Nighttime curb segmentation training algorithm

Algorithm: Training method of nighttime curb segmentation
Input: night and day domain dataset X, Y; generator G; discriminator D; Feature Extractor F; Segmentation network S;

1:	Initialize G, D, F, S with random weights
2:	**for** epoch **do**
3:	**for** batch **do**
4:	Get batch $x\sim X$, $y\sim Y$
5:	<Stage 1>
6:	Freeze G, F and activate S, D
7:	Generate Synthetic day image $\tilde{x}$ from night image x
8:	Calculate Adversarial GAN loss
9:	Predict curb at night domain
10:	Predict curb at day domain
11:	Calculate Segmentation loss
12:	Calculate Stage1 loss
13:	Update network S, D
14:	<State 2>
15:	Freeze S, D and activate G, F
16:	Generate Synthetic day image $\tilde{x}$ from night image x
17:	Calculate Adversarial GAN loss
18:	Calculate PatchNCE loss
19:	Calculate Segmentation loss
20:	Calculate Stage 2 loss
21:	Update networks G, F
22:	**end for**
23:	**end for**

Table 2 Hyperparameters for training

Parameter	Value
Batch size	2
Max epoch	300
Learning rate for G	2×10^{-4}
Learning rate for D	2×10^{-4}
Learning rate for F	2×10^{-4}
Learning rate for S	1×10^{-3}
w_{adv} for *Stage* 1	1.0
w_{seg} for *Stage* 1	1.0
w_{adv} for *Stage* 2	1.0
w_{seg} for *Stage* 2	1.0
w_{con} for *Stage* 2	1.0
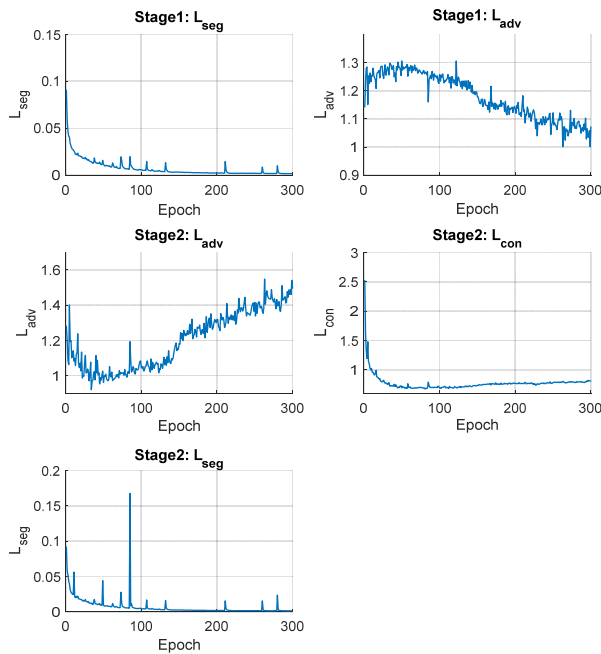
Fig. 6 Training loss of the proposed method

Fig. 7 (a) The architecture and (b) the network structure of the benchmark algorithm

4. Algorithm Evaluation

4.1 Test Dataset

The test data utilized in this study is also extracted from the AI-Hub dataset. In validating the proposed method, a distinct test dataset, separate from the training phase, is employed. Specifically, 100 night images and 177 day images are utilized for testing.

4.2 Benchmark Algorithm

The comparison between the proposed method and a benchmark method is conducted using the test dataset. In Fig. 7, the network and training architecture of the benchmark algorithm are illustrated. The benchmark algorithm infers curb segments based on the original image, utilizing a cross-entropy loss function identical to L_{seg} in the proposed method. The parameters of the benchmark network align with the segmentation network of the proposed network, and the learning rate for the benchmark algorithm is set at 1×10^{-3}.

4.3 Metrics

For validation, metrics such as Precision and F1 score are employed, which are commonly used in segmentation tasks. Precision represents the proportion of correct positive predictions among all positive predictions, indicating a high level of confidence in the predictions when Precision is high. F1 score is the harmonic mean of Precision and Recall, where Recall is the ratio of correct

data, comprising 1,000 night domain images and segmentation annotations sampled from the full night dataset, along with 2,759 day data from the complete day dataset, was used for training. The image size was resized from 1,920 × 1,200 to 640 × 400.

The major hyperparameters for training are listed in Table 2. The validation was conducted in a Python 3.10.6. For implementing deep learning network, we use pytorch 1.13.0. All algorithms were processed by the workstation which consists of Intel Xeon CPU E5-2630 v4 @ 2.20 GHz and Nvidia GeForce RTX 4090 GPU.

In Fig. 6, the training loss across epochs is depicted. Both L_{seg} for *Stages* 1 and 2 exhibit a gradual decrease. Additionally, L_{con} for *Stage* 2 also shows a gradual decrease. The adversarial losses, L_{adv}, for both *Stage* 1 and *Stage* 2 are trained in an adversarial manner. Consequently, at the initial stages of training, L_{adv} for *Stage* 2 decreases while L_{adv} for *Stage* 1 increases, indicating the generator's initial advantage over the discriminator. However, as training progresses, L_{adv} for *Stage* 2 increases, and L_{adv} for *Stage* 1 decreases, signifying a shift where the discriminator begins to outperform the generator.
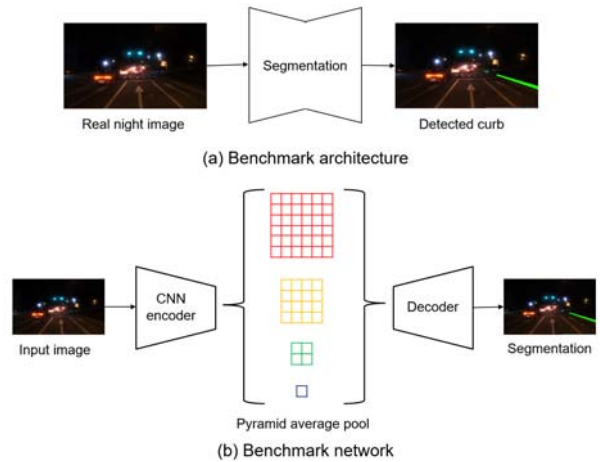
positive predictions to all actual positive ground truth, revealing how many of the positive answers were correctly predicted.

If a model is solely focused on increasing Precision, the model aims to minimize positive predictions for a higher value of Precision. Similarly, if a model is solely focused on increasing Recall, the model aims to maximize positive predictions for a higher value of Recall. Therefore, F1 provides a balanced metric to evaluate segmentation performance. Additionally, excessively low Precision may lead to 'spurious detections' and create issues in autonomous vehicle control. Hence, both Precision and F1 are utilized as crucial validation metrics.

4.4 Result

Fig. 8 presents the validation results comparing the benchmark method to the proposed method. The proposed algorithm demonstrates a 24 % higher Precision and a 124
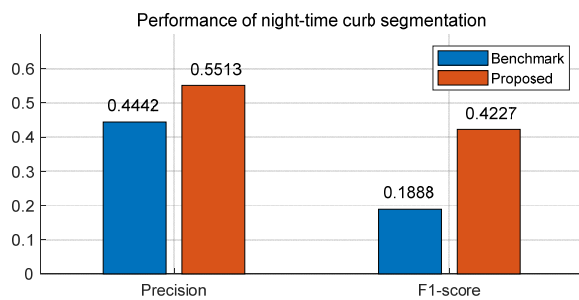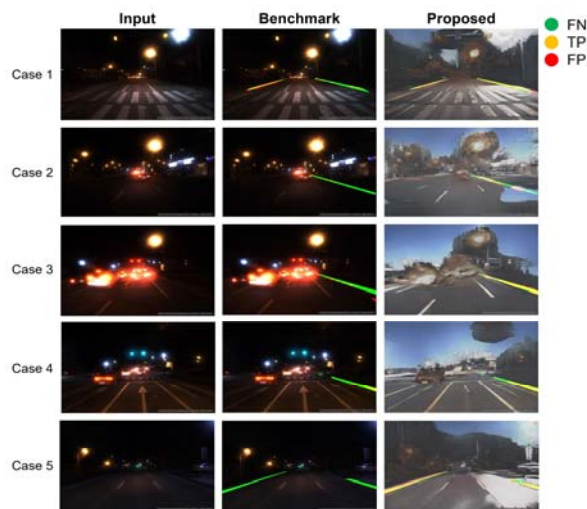
% higher F1-score for nighttime segmentation compared to the benchmark algorithm.

In Fig. 9, segmentation results for five representative cases are illustrated. The first column displays the original night image, while the second and third columns represent predictions from the benchmark algorithm and the proposed algorithm. The background image in the third column consists of synthetic day images created by the generator. The green and yellow areas indicate the ground truth curb, while the red and yellow areas represent the curb predictions of each algorithm. Although the synthetic images resemble daytime scenes, they lack realism. This is attributed to the L_{seg} of *Stage* 2, which prompts the generator to prioritize segmentation performance. Consequently, the generator produces images that are realistic yet optimized for segmentation performance. In Case 1, the benchmark algorithm fails to detect the curb on the right side, whereas the proposed method successfully

Fig. 8 Performance of nighttime curb segmentation

Table 3 Performance metrics of sample cases

Case number	Benchmark		Proposed	
	Precision	F1-score	Precision	F1-score
Case 1	0.5462	0.4203	0.6229	0.6718
Case 2	0.8375	0.0663	0.5426	0.4606
Case 3	0.4322	0.1399	0.8521	0.8573
Case 4	1.0000	0.4214	0.9895	0.7394
Case 5	0.5052	0.4901	0.5593	0.6233

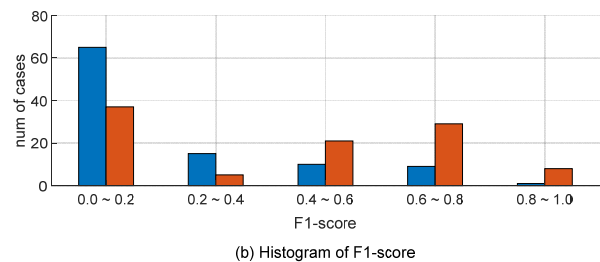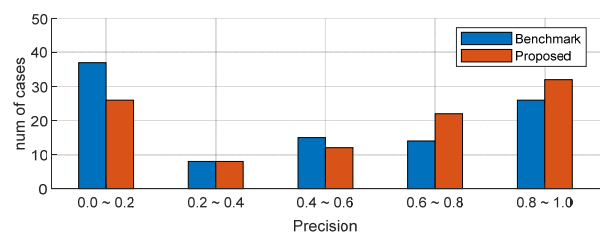Fig. 9 Qualitative results of five sample cases

Fig. 10 Histogram of performance metrics

identifies all curbs in the image.

Table 3 provides performance metrics for the cases depicted in Fig. 9. The F1-score of the proposed method surpasses that of the benchmark algorithm in every case. In Cases 2 and 4, where the curb is nearly undetected, Precision values are remarkably high, indicating minimal false positives. However, the overall performance is better evaluated through the F1-score.

Fig. 10 illustrates the histogram of the performance metrics. Compared to the benchmark algorithm, the proposed algorithm demonstrates a reduction in cases with low performance and an increase in cases with high performance.

In summary, the results and analysis lead to the conclusion that the proposed algorithm structure, combined with the training methodologies, effectively enhances segmentation performance during nighttime conditions.

5. Conclusion

In this research, we proposed a method to enhance nighttime curb segmentation performance. The central concept of our algorithm architecture involved using synthetic day images and real images as inputs for the segmentation network. To generate synthetic images, we designed a generator network for domain translation. The segmentation network underwent training with a typical loss function for pixel-wise segmentation, while the translation network was trained using the loss function adopted from the CUT method. The proposed algorithm demonstrated a 24 % increase in precision and a 124 % improvement in curb segmentation performance by F1 score compared to the benchmark. This study reveals that the method significantly enhances nighttime curb segmentation performance. Consequently, the enhanced nighttime curb segmentation is anticipated that autonomous vehicles will possess more robust perception capabilities, fostering the development of safer systems.

Acknowledgement

References

1) S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus and M. H. Ang, "Perception, Planning, Control, and Coordination for Autonomous Vehicles," Machines, Vol.5, No.1, p.6, 2017.
2) J. Tang, S. S. Liu, S. W. Pei, S. Zuckerman, C. Liu, W. S. Shi and J. L. Gaudiot, "Teaching Autonomous Driving Using a Modular and Integrated Approach," IEEE Annual Computer Software and Applications Conference, pp.361-366, 2018.
3) F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stübler, M. Bach, P. Hatzelmann, C. Wild and K. Dietmayer, "Autonomous Driving at Ulm University: A Modular, Robust, and Sensor-Independent Fusion Approach," IEEE Intelligent Vehicles Symposium, pp.666-673, 2015.
4) C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky and A. M. Bayen, "Flow: A Modular Learning Framework for Mixed Autonomy Traffic," IEEE Transactions on Robotics, Vol.38, No.2, pp.1270-1286, 2022.
5) Đ. Petrović, R. Mijailović and D. Pešić, "Traffic Accidents with Autonomous Vehicles: Type of Collisions, Manoeuvres and Errors of Conventional Vehicles' Drivers," Transportation Research Procedia, Vol.45, pp.161-168, 2020.
6) M. Kim, Y. S. Kim, H. S. Jeon, D. S. Kum and K. B. Lee, "Autonomous Driving Technology Trend and Future Outlook: Powered by Artificial Intelligence," Transactions of KSAE, Vol.30, No.10, pp.819-830, 2022.
7) L. Wang, D. Li, H. Liu, J. Z. Peng, L. Tian and Y. Shan, "Cross-Dataset Collaborative Learning for Semantic Segmentation in Autonomous Driving," AAAI Conference on Artificial Intelligence, pp.2487-2494, 2022.
8) K. Muhammad, T. Hussain, H. Ullah, J. Del Ser, M. Rezaei, N. Kumar, M. Hijji, P. Bellavista and V. H. C. de Albuquerque, "Vision-Based Semantic Segmentation in Scene Understanding for Autonomous Driving: Recent Achievements, Challenges, and Outlooks," IEEE Transactions on Intelligent Transportation Systems, Vol.23, No.12, pp.22694-22715, 2022.
9) S. Panev, F. Vicente, F. De la Torre and V. Prinet, "Road Curb Detection and Localization With Monocular Forward-View Vehicle Camera," IEEE Transactions on Intelligent Transportation Systems, Vol.20, No.9, pp.3568-3584, 2019.

10) A. Y. Hata, F. S. Osorio and D. F. Wolf, "Robust Curb Detection and Vehicle Localization in Urban Environments," IEEE Intelligent Vehicles Symposium, pp.1264-1269, 2014.

11) F. Oniga, S. Nedevschi and M. M. Meinecke, "Curb Detection Based on a Multi-Frame Persistence Map for Urban Driving Scenarios," International IEEE Conference on Intelligent Transportation Systems, pp.67-72, 2008.

12) A. Seibert, M. Hähnel, A. Tewes and R. Rojas, "Camera Based Detection and Classification of Soft Shoulders, Curbs and Guardrails," IEEE Intelligent Vehicles Symposium, pp.853-858, 2013.

13) M. Cheng, Y. Zhang, Y. Su, J. M. Alvarez and H. Kong, "Curb Detection for Road and Sidewalk Detection," IEEE Transactions on Vehicular Technology, Vol.67, No.11, pp.10330-10342, 2018.

14) G. Weld, E. Jang, A. Li, A. Zeng, K. Heimerl and J. E. Froehlich, "Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery," International ACM SIGACCESS Conference on Computers and Accessibility, pp.196-209, 2019.

15) H. Zhou, H. Wang, H. D. Zhang and K. Hasith, "LaCNet: Real-time End-to-End Arbitrary-shaped Lane and Curb Detection with Instance Segmentation Network," IEEE International Conference on Control, Automation, Robotics and Vision, pp.184-189, 2020.

16) G. Zhao and J. Yuan, "Curb Detection and Tracking Using 3D-LIDAR Scanner," IEEE International Conference on Image Processing, pp.437-440, 2012.

17) Y. H. Zhang, J. Wang, X. N. Wang and J. M. Dolan, "Road-Segmentation-Based Curb Detection Method for Self-Driving via a 3D-LiDAR Sensor," IEEE Transactions on Intelligent Transportation Systems, Vol.19, No.12, pp.3981-3991, 2018.

18) J. A. Guerrero, R. Chapuis, R. Aufrère, L. Malaterre and F. Marmoiton, "Road Curb Detection using Traversable Ground Segmentation: Application to Autonomous Shuttle Vehicle Navigation," IEEE International Conference on Control, Automation, Robotics and Vision, pp.266-272, 2020.

19) G. J. Wang, J. Wu, R. He and S. Yang, "A Point Cloud-Based Robust Road Curb Detection and Tracking Method," IEEE Access, Vol.7, pp.24611-24625, 2019.

20) Y. Jung, S. W. Seo and S. W. Kim, "Curb Detection and Tracking in Low-Resolution 3D Point Clouds Based on Optimization Framework," IEEE Transactions on Intelligent Transportation Systems, Vol.21, No.9, pp.3893-3908, 2020.

21) K. S. Lee and T. -H. Park, "Curbs Detection using 3D Lidar Sensor," KSAE Spring Conference Proceedings, pp.1402-1403, 2018.

22) T. Park, A. A. Efros, R. Zhang and J. -Y. Zhu, "Contrastive Learning for Unpaired Image-to-Image Translation," European Conference on Computer Vision, pp.319-345, 2020.

23) J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Conference on Computer Vision and Pattern Recognition, pp.3431-3440, 2015.

24) O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer-Assisted Intervention, pp.234-241, 2015.

25) H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," IEEE Conference on Computer Vision and Pattern Recognition, pp.2881-2890, 2017.

26) A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold and S. Gelly, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint arXiv: 2010.11929, 2020.

27) Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," IEEE/CVF International Conference on Computer Vision, pp.10012-10022, 2021.

28) E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez and P. Luo, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," Advances in Neural Information Processing Systems, pp.12077-12090, 2021.

29) I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Networks," Communications of the ACM, pp.139-144, 2020.

30) P. Isola, J. -Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," IEEE Conference on Computer Vision and Pattern Recognition, pp.1125-1134, 2017.

31) M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," arXiv preprint arXiv:1411.1784, 2014.

32) J. Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," IEEE/CVF International Conference on Computer Vision, pp.2242-2251, 2017.

33) AI-Hub, "Autonomous Driving 3-D Image under Unusual Environmental Conditions," https://www.aihub.or.kr/, 2020.