*Sensors and Materials*, Vol. 36, No. 9 (2024) 3947–3956
MYU Tokyo



S & M 3777

# Improving Performance of Instance Segmentation Model for Building Object Detection Using Contrastive Unpaired Translation

Seung Bae Jeon, Gyusang Kim, Minjae Choi, and Myeong-Hun Jeong*

Department of Civil Engineering, Chosun University,
309 Pilmun-daero, Dong-gu, Gwangju 61452, Republic of Korea

(Received August 8, 2024; accepted September 9, 2024)

***Keywords:*** contrastive unpaired translation, building segmentation, image translation

With the advancements in data collection processes and sensors, a vast amount of data is now available, driving the increasing application and utilization of deep-learning-based artificial intelligence technologies. For instance, object detection through image data is utilized in various fields such as traffic safety, crime prevention and public safety, environmental monitoring, and disaster response in urban areas. Deep-learning-based object detection models exhibit high performance, but there are limitations in performance when the training data is restricted. There are issues with degraded object detection performance owing to differences in environmental factors (occlusion and illumination) in the data collected under different solar altitudes or shooting conditions from the training data. The aim of this study is to enhance the performance of object segmentation by mitigating different environmental factors between the training and collected data using the contrastive-unpaired-translation (CUT) algorithm, one of the image-to-image translation algorithms. In this study, we aim to improve object segmentation performance by generating images under environmental conditions (e.g., shadows and shading) similar to those of the training data. The object segmentation model used in this study was You Only Look Once version 8, and instance segmentation was performed by inputting data with and without applying CUT. The results showed an improvement of approximately 11.11% in mAP@50. Furthermore, statistical verification confirmed that this is a significant difference. The results of this study confirmed the potential of improving instance segmentation performance through image translation techniques, which will contribute to autonomous driving and unmanned services.

## 1. Introduction

Urban areas, with their heavy traffic and high population density, necessitate appropriate management and operation. The rapid advancements in sensor technology and data collection processes generate and distribute vast amounts of data, which can be utilized to manage and operate infrastructure in urban areas, contributing to the advancement of related technologies.[1,2]

*Corresponding author: e-mail: mhjeong@chosun.ac.kr
https://doi.org/10.18494/SAM5267

ISSN 0914-4935 
https://myukk.org/

Recently, deep-learning-based technologies in computer vision have been increasingly applied to the diverse data generated in urban areas.[3,4] Specifically, segmenting buildings in urban areas enables the monitoring of individual structures to analyze land use patterns and ensure compliance with building codes and regulations. Moreover, in disaster scenarios such as earthquakes and fires, it helps to rapidly assess damage to individual buildings and establish recovery plans, as well as identify and take preventive measures against aging or hazardous structures.

Deep-learning-based models have shown superior performance over traditional models previously used in these fields.[5,6] However, a limitation of deep learning models is that their performance is dependent on the training data. For instance, the performance of object detection can degrade owing to environmental factors (occlusion and illumination) that vary depending on different shooting conditions even within the same space. While it is possible to build a model by including diverse cases in the training data, this requires a vast amount of training data and a significant computational cost, and different training data for the same object can degrade model performance. Therefore, it is infeasible to consistently collect image data under the same conditions, necessitating research to address these issues.

Among studies aimed at mitigating issues arising from varying environmental factors, low-light image enhancement research addresses performance degradation in instance segmentation models through image preprocessing. Simple methods include setting the ISO sensitivity of a digital camera higher to amplify the light entering the lens or using histogram equalization (HE) to evenly distribute pixel values. Although these methods can simply process images to potentially improve object detection performance, they have limitations that can degrade performance, such as increased noise. In addition, simple structured models struggle with the nonlinear transformations required for enhancing low-light images. To overcome these limitations and achieve more refined adjustments, models capable of effectively handling nonlinearity are required.

Deep-learning-based models, with their multilayer structures, handle nonlinearity better than conventional methods and effectively learn complex patterns within images, making them advantageous for correcting noise in images. Consequently, various deep-learning-based models have been developed for image enhancement, incorporating technologies such as convolutional layers, rectified linear units, and residual connections, and have outperformed traditional methods.[7]

These deep-learning-based models require paired images for training. To mitigate the performance degradation in object detection owing to differing environmental factors between the training data and the input data for inference, it is necessary to use models trained on all possible cases. However, constructing training data that includes every different environmental factor occurring in urban areas is nearly impossible and demands significant computational resources.

To address the above issues, in this study, we applied unpaired image-to-image translation. Unpaired image-to-image translation learns to map from the source domain to the target domain, which can then be applied to mitigate the differences in

environmental factors caused by variations in solar elevation angle, illumination, and shooting angle. For instance, it can adjust training and test data taken under different environmental conditions as if they were obtained simultaneously.

The aim of this study is to mitigate the degradation of object detection performance by generating fake images converted to similar environmental conditions as the training data, using the contrastive-unpaired-translation (CUT) algorithm, which performs image translation with a single image from each domain. This involves applying CUT to datasets with environmental factors different from those of training the instance segmentation model for buildings in urban areas, thus creating images with conditions similar to the training data. Subsequently, to verify if the proposed method improves the performance of the instance segmentation model, instance segmentation was performed on the You Only Look Once version 8 (YOLOv8) model using data with and without CUT applied. Statistical verification techniques were then applied to analyze and present whether the performance differences were significant. The overall flow is illustrated in Fig. 1.

## 2. Related Works

When applying a pretrained instance segmentation model to new data, the performance is often limited by the training data. This particularly applies to large urban areas, in which primarily aerial or satellite imagery is used, making it difficult to expect data with the same environmental conditions. These variations can restrict the performance of segmentation models. For instance, the objects not shaded in the training data may appear shaded in the test data, or differences in illumination may lead to reduced segmentation model performance.

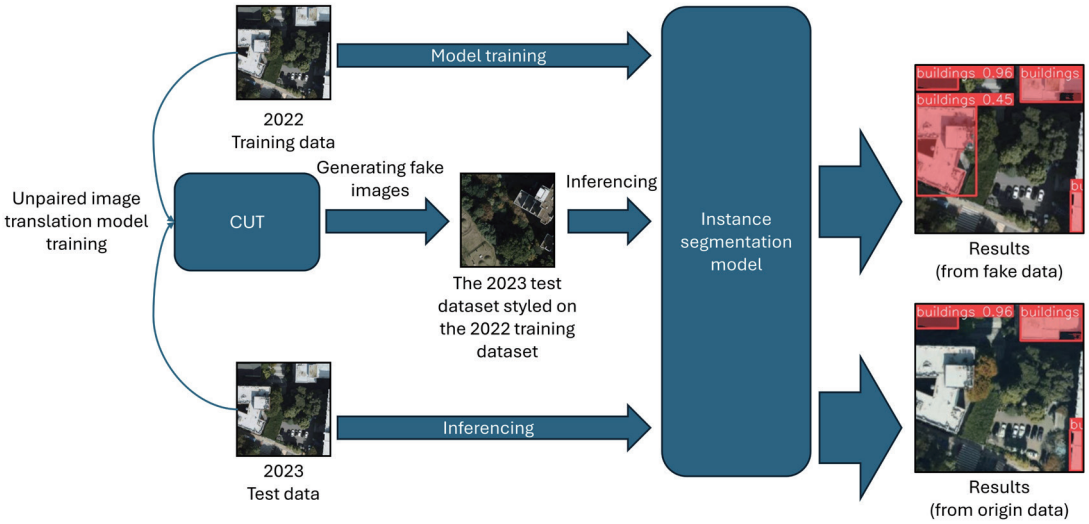


Fig. 1. (Color online) Architecture of the proposed framework.

To mitigate problems arising from these environmental factors, image quality improvement techniques can be applied. One of the most well-known contrast enhancement methods is HE, which enhances contrast and visibility by converting low-contrast images with a narrow pixel value range into high-contrast images with a wide pixel value range. However, since HE redistributes pixel values assuming a single histogram if the pixel value distribution is inappropriate, the image may become distorted. Adaptive HE (AHE) improves upon HE by dividing the image into equal-sized segments and performing HE on each segment individually.[8] These methods, including HE and its variations, have extremely simple structures and can efficiently enhance haze images, thereby improving the performance of deep-learning-based models for detecting buildings and vehicles under poor visibility conditions.[9]

The superior image enhancement performance of adaptive HE compared with HE implies the necessity of nonlinear transformations for image enhancement. Although methods such as AHE perform arbitrary local transformations by dividing the image, more advanced methods are required to efficiently execute nonlinear transformations. Deep-learning models, starting with the multilayer perceptron, have been developed to solve nonlinear problems and can efficiently perform nonlinear transformations.

The low-light net (LLNet) extracted 422500 patches from 169 standard test images and trained a neural network model consisting of an encoder and a decoder for image enhancement, resulting in significantly higher performance than those of statistical methods such as HE, contrast-limiting adaptive HE (CLAHE), and gamma adjustment.[10] Liu *et al.* built a model that simultaneously performs image enhancement and object detection using a pretrained enhancement model and a feature-guided module to detect objects underwater.[11]

These deep-learning models can perform image enhancement at a higher level than traditional simple statistical methods and improve the performance of object detection models. Although these models require training data for image enhancement, obtaining image data of the same scene with different specific factors is challenging. Even if obtained, it is difficult to gather enough to sufficiently train the deep-learning model's weights. Therefore, such training data is mostly composed of synthetically darkened and noise-added examples.

Xu *et al.* addressed haze image enhancement by embedding physical lighting models that provide information for image enhancement, such as dark channels, into a deep neural network, performing both image enhancement and object detection.[12] This model composition does not require separate training data for image enhancement but is limited to mitigating specific factors such as dehazing.

To overcome these limitations, we proposed the use of image translation. Image translation maps images from different domains and can be used in areas such as style transfer or data augmentation. This technique, such as image enhancement, can be used to improve the performance of models for object detection or segmentation.

Arruda *et al.* applied an unpaired image-to-image translator to annotated day and night datasets to generate fake-night images, transferring the annotation information from the day dataset to the annotated fake-night images for training a car detection model.[13] The trained model was then used to detect cars in original night images. This approach improved the performance of object detection models, similarly to studies on enhancing darkened images.

Li *et al.* used cycle consistency loss to translate optical images to SAR images, mitigating limitations of optical images (e.g., weather conditions) to perform change detection.[14] In their study, similar to image enhancement, they aimed to improve the performance of change detection models by mitigating environmental factors such as snow or rain. Both image enhancement and image translation can be utilized to improve the performance of computer vision models for specific tasks and are applied in various fields.[15]

Motivated by the above studies, in this study, we applied the CUT algorithm, one of the unpaired image-to-image translation algorithms, to enhance the performance of building segmentation models in urban areas. The CUT model can perform image translation with only one image from each domain without requiring vast data for training the generator model. Moreover, rather than using cycle consistency loss, which assumes a bijection relationship between two domains and thus imposes constraints on the transformation, the CUT algorithm maximizes mutual information using contrastive learning.[16] This approach was used to mitigate all environmental differences between the training and test data and to perform building segmentation in urban areas.

## 3. Data and Methods

The data used in this study consists of a true orthophoto of Mapo-gu, Seoul, collected in 2022 and 2023, with a resolution of 12 cm. The area covered by the true orthophoto is approximately 23.87 km². For model training, the images captured in 2022 were divided into a total of 538 images, each 256 × 256 in size. Figure 2 shows the spatial location and the true orthophoto.

Out of these, 498 images were classified as the training dataset and 20 as the validation dataset, with building objects annotated for instance segmentation. Subsequently, a test dataset was constructed by the 2023 orthophotography, and the CUT model was applied to generate fake images that match the environmental factors of the 2022 training data. The original and fake images from the test dataset were used as input for the instance segmentation model in the experiments.

The proposed methods are divided into two main parts: the model for instance segmentation and the model for generating fake images. For instance segmentation, YOLOv8 was used, and for generating fake images, the image translation model CUT was employed.

YOLO, first introduced in 2016, is a deep-learning-based network for real-time object detection. It divides the input image into fixed-size grid cells, calculating values such as confidence scores and class probabilities for each cell to perform segmentation. The initial model used Darknet, which consisted of 24 convolutional layers and 2 fully connected layers. Subsequent versions have improved accuracy and speed through enhancements to the backbone and introducing methods such as feature pyramids.[17–19]

YOLOv8, developed by Ultralytics in January 2023, includes models ranging from nano to xlarge in size.[20] Among them, YOLOv8m achieved a mAP of 50.2 on COCO, a large-scale object detection dataset consisting of more than 200000 labeled images across 80 different categories, and has exhibited high performance across various domains. YOLOv8 uses a

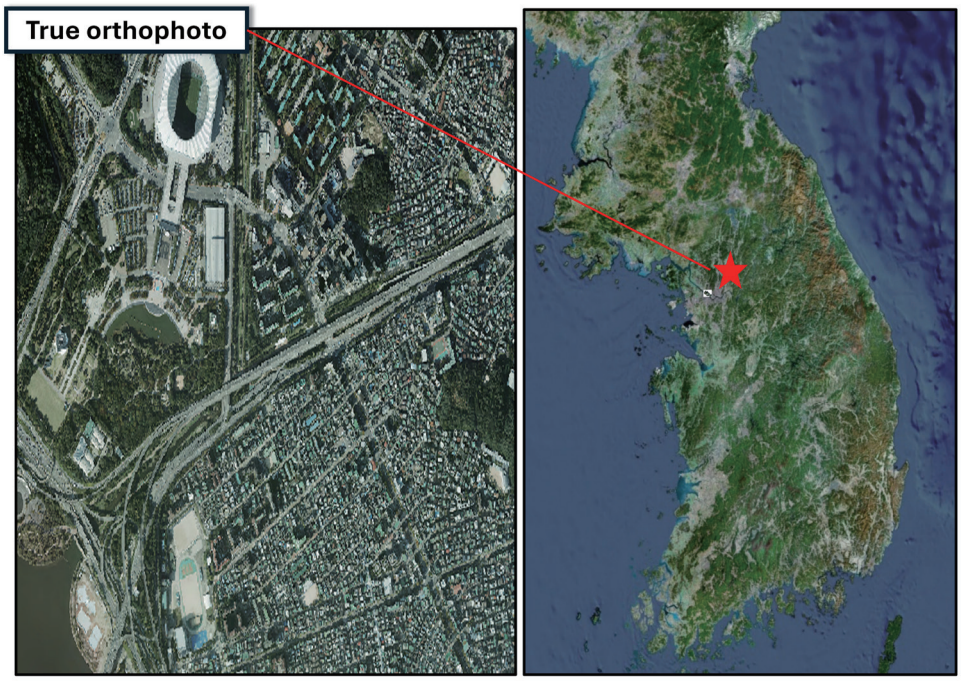


Fig. 2. (Color online) Study area and true orthophoto.

modified CSPDarknet32 backbone, where the CSP layers from YOLOv5 have been replaced with the C2f module. The spatial pyramid pooling fast layer accelerates computation by pooling image features into fixed-size maps. It uses convolution, batch normalization, and SiLU activation (CBS), and the head is divided into separate tasks for process objectness, classification, and regression.

The CUT algorithm is used for image-to-image translation when corresponding image pairs are unavailable. Since obtaining images with only specific factors differing at the same location is nearly impossible, the unpaired image translation algorithm was deemed suitable for this study.

The previously utilized cycle generative adversarial network (GAN) ensured the mapping between fake and original images to be consistent by using cycle consistency but assumed a bijective relationship between the two domains, thus imposing restrictions on transformations. The CUT algorithm eliminates this cycle consistency and uses patch wise contrastive loss to perform image translation between different domains.

CUT is trained using three loss functions: the adversarial loss common to most GAN-based models, patch wise contrastive loss for maximizing mutual information, and identity loss for preserving bidirectionality.

Adversarial loss is a core component of the GAN structure and enhances the quality of generated images through the interaction between the generator and the discriminator. The adversarial loss is defined as follows:[21]

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{y\sim Y}\left[logD(x)\right] + \mathbb{E}_{x\sim X}\log\left(1 - D(G(x))\right), \tag{1}$$

where $D$ is a discriminator, $G$ is a generator model, and $x$ is an input image.

CUT applies the principles of infoNCE loss for mutual information maximization to the image translation task, using patchNCE loss. Mutual information refers to the amount of shared information between two vectors and is indicative of their similarity. InfoNCE loss is a loss function designed to maximize mutual information between two vectors.[22] Consequently, patchNCE loss ensures that the generated image retains detailed features of the original image by comparing patch features.

PatchNCE loss maximizes the similarity between queries and positive keys while minimizing the similarity between queries and negative examples, ensuring that the generated image retains the detailed features of the original image. PatchNCE loss is defined as follows:[16]

$$\mathcal{L}_{PatchNCE} = -\mathbb{E}\left[\log\frac{exp(sim\left(f_q, f_k^+\right)}{\exp\left(sim\left(f_q, f_k^+\right)\right) + \sum_{f_k^-}\exp\left(sim\left(f_q, f_k^-\right)\right)})\right], \tag{2}$$

where $f_q$ is the query (patch feature from the generated image), $f_k^+$ is the positive key (patch feature from the same location in the original image), $f_k^-$ are the negative keys (patch features from different locations in the original image), and $sim()$ is a function indicating the similarity between vectors, typically using the dot product.

Identity loss helps the generator maintain the style or color of the original image, preserving details during the image translation process. It often uses L1 Loss for this purpose.

## 4. Experimental Results

We conducted experiments to evaluate the effect of the CUT algorithm on enhancing the object detection performance of YOLOv8. The YOLOv8 model was trained using the 2022 training data, and experiments were conducted using the 2023 test data, which had different environmental factors. The 2023 test data was divided into two versions: one with the CUT applied and one without.

The CUT algorithm was applied as a single image unpaired translation using the 2022 training data and the 2023 test data. Despite being from the same locations, the images had different environmental factors between 2022 and 2023. Therefore, fake 2023 test data was generated through image translation. The generated fake images were made from the 2023 data but translated to have similar environmental factors to 2022. Figure 3 shows the transformed images.

Object detection was then performed using the YOLOv8 model, and the performancs characteristics observed before and after applying the CUT algorithm were compared. The metrics used for the comparison were mAP@50 and mAP@50-95. Table 1 shows the results of the model performance evaluation.


Fig. 3. (Color online) Results of image translation.

Table 1
Results obtained with and without CUT. All scores are expressed as percentages (%).

| Model | mAP@50 | mAP@50–95 |
|---|---|---|
| YOLOv8 (without CUT) | 74.49 | 56.42 |
| YOLOv8 (with CUT) | 85.60 | 63.88 |

The YOLOv8 model trained with the data processed using the CUT algorithm improved by approximately 11.11% in mAP@50 and 7.46% in mAP@50-95. To verify the statistical significance of this performance improvement, a statistical validation method was applied. Before the statistical validation, the Shapiro–Wilk test was conducted to test the normality of the data.[23] The Shapiro–Wilk test is a statistical method to validate whether data follows a normal distribution. It is particularly useful when the sample size is small. The test results showed $p$-value < 0.05, indicating that the data did not follow a normal distribution.

Considering the non-normal distribution of the data and the small sample size, a bootstrap $t$-test was performed. The basic t-test is used to test the difference in means between two groups but assumes the normality of the data. However, since real data often do not follow normality, the bootstrap method was used for more flexible testing. Bootstrap involves resampling from the sample multiple times to estimate the distribution of the statistics without assuming the sample's distribution. This method is particularly useful for small sample sizes. After performing 1000

resamplings and conducting the bootstrap t-test, the results showed $p$-value < 0.05, confirming a significant difference between the means of the two groups.

Through statistical validation, we verified that the performance improvement was significant. This confirms the effectiveness of the CUT algorithm in mitigating environmental factor differences between training and test data and enhancing object detection performance.

## 5. Discussion and Conclusions

The aim of this study was to enhance object detection performance in urban areas using the CUT algorithm. Existing object detection models experienced performance degradation issues owing to environmental factor differences between training and test data. To address this, the CUT algorithm was applied to generate fake images that had been transformed to have environmental conditions similar to those of the training data, thus improving object detection performance.

The results showed that the YOLOv8 model with the CUT algorithm improved by approximately 11.11% in mAP@50 and 7.46% in mAP@50-95. This indicates that mitigating environmental factor differences can enhance the performance of object detection models. The proposed CUT algorithm effectively performed image translation without the need for paired image pairs, making it suitable for handling data with various environmental factors. In addition, statistical validation confirmed that the performance improvement was not merely coincidental.

In conclusion, we demonstrated that the CUT algorithm, without relying on cycle consistency, could effectively perform image translation, thus enhancing object detection performance on data with various environmental factors. The results of experiments confirmed that using the CUT algorithm could improve the object detection performance by effectively mitigating environmental factor differences between training and test data. However, the implementation of additional deep-learning models to enhance images and improve object detection performance inevitably increases computational costs and may potentially degrade performance in real-time applications. Although this method introduces extra computational overhead, it remains more efficient and feasible than training a model on perfectly uniform data or significantly expanding the training dataset to cover nearly all possible scenarios.

Furthermore, this study was conducted using a dataset limited to a specific urban area in South Korea. Future work will focus on expanding the dataset to include more diverse regions and environmental conditions, highlighting the need for further research with larger datasets and various domains to generalize the results of this study.

## Acknowledgments

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (RS-2024-00354270) and “Regional Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-002).

## References


1 M. S. Khan, S. B. Jeon, and M.-H. Jeong: Remote Sens. **13** (2021) 4976. https://doi.org/10.3390/rs13244976
2 S. B. Jeon, M.-H. Jeong, T.-y. Lee, and D. Cho: Sens. Mater. **35** (2023) 3393. https://doi.org/10.18494/SAM4437
3 T. Azfar, J. Li, H. Yu, R. L. Cheu, Y. Lv, and R. Ke: Data Sci. Transp. **6** (2024) 1. https://doi.org/10.1007/s42421-023-00086-7
4 R. Marasinghe, T. Yigitcanlar, S. Mayere, T. Washington, and M. Limb: Sustainable Cities Soc. **100** (2023) 105047. https://doi.org/10.1016/j.scs.2023.105047
5 J. Tian, Q. Jin, Y. Wang, J. Yang, S. Zhang, and D. Sun: J. Eng. Appl. Sci. **71** (2024) 76. https://doi.org/10.1186/s44147-024-00411-z
6 S. B. Jeon, S. Kang, M.-H. Jeong, and H. Lee: Case Stud. Construct. Mater. **20** (2024) e03331. https://doi.org/10.1016/j.cscm.2024.e03331
7 S. Zhang, C. Liu, Y. Zhang, S. Liu, and X. Wang: Sensors **23** (2023) 7713. https://doi.org/10.3390/s23187713
8 S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld: Comput. Vision, Graphics, Image Process. **39** (1987) 355. https://doi.org/10.1016/S0734-189X(87)80186-X
9 J. Dawar, P. Raheja, U. Vashisth, I. Cheng, and A. Basu: 2020 IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW) (2020) 1. https://doi.org/10.1109/ICMEW46912.2020.9106053
10 K. G. Lore, A. Akintayo, and S. Sarkar: Pattern Recognit. **61** (2017) 650. https://doi.org/10.1016/j.patcog.2016.06.008
11 H. Liu, F. Jin, H. Zeng, H. Pu, and B. Fan: IEEE Trans. Neural Networks Learn. Syst. **34** (2023) 2123. https://doi.org/10.1109/TNNLS.2023.3274926
12 X. Xu, S. Wang, Z. Wang, X. Zhang, and R. Hu: ACM Trans. Multimedia Comput., Commun., Appl. (TOMM) **17** (2021) 1. https://doi.org/10.1145/3414839
13 V. F. Arruda, T. M. Paixao, R. F. Berriel, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos: 2019 Int. Joint Conf. Neural Networks (IJCNN) (2019) 1. https://doi.org/10.1109/IJCNN.2019.8852008
14 X. Li, Z. Du, Y. Huang, and Z. Tan: ISPRS J. Photogramm. Remote Sens. **179** (2021) 14. https://doi.org/10.1016/j.isprsjprs.2021.07.007
15 C. Liu and B. Xu: Comput.-Aided Civ. Infrastruct. Eng. **37** (2022) 1737. https://doi.org/10.1111/mice.12849
16 T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu: Computer Vision–ECCV 2020: 16th Eur. Conf., Glasgow, UK, August 23–28, 2020, Proceedings, Part IX **16** (2020) 319. https://doi.org/10.1007/978-3-030-58545-7_19
17 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2016) 779. https://doi.org/10.1109/CVPR.2016.91
18 J. Redmon and A. Farhadi: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2017) 7263. https://doi.org/10.1109/CVPR.2017.690
19 C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, and W. Nie: arXiv preprint arXiv:2209.**02976** (2022). https://doi.org/10.48550/arXiv.2209.02976
20 Ultralytics: Yolov8, https://github.com/ultralytics/ultralytics (June 19, 2024).
21 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio: Adv. Neural Inf. Process. Syst. **27** (2014). https://doi.org/10.48550/arXiv.1406.2661
22 A. v. d. Oord, Y. Li, and O. Vinyals: arXiv preprint arXiv:1807.03748 (2018). https://doi.org/10.48550/arXiv.1807.03748
23 S. S. Shapiro and M. B. Wilk: Biometrika **52** (1965) 591. https://doi.org/10.1093/biomet/52.3-4.591