

# GAN-Based Domain Adaptation for Creating Digital Twins of Small-Scale Driving Testbeds: Opportunities and Challenges

Sangeet Sankaramangalam Ulhas, Shenbagaraj Kannapiran, and Spring Berman

**Abstract**—In recent years, small-scale driving testbeds have been developed as controlled physical environments for the evaluation of autonomous vehicle controllers. Such controllers are heavily dependent on computer vision algorithms that enable the vehicle to perceive its surroundings. To bridge the Sim2Real content and appearance gap between simulated and real-world image data for training these algorithms, we propose a novel transfer learning approach that performs domain adaptation using StyleGAN to generate style-mixed images that closely resemble real-world images. We explain our approach within the context of our small-scale driving testbed, CHARTOPOLIS, and demonstrate it on synthetic image data of two object classes, vehicles and buildings, from the driving simulator CARLA. Our results show that this approach works on the vehicle object class while failing on the building object class. This paper thus provides a plausible approach to bridging the Sim2Real gap through the use of custom pipelines that augment image datasets using a mix of techniques for domain adaptation and domain randomization.

## I. INTRODUCTION

A variety of small-scale driving testbeds, e.g. [1]–[8] and our own testbed CHARTOPOLIS [9] (Fig. 1), have been developed in recent years to provide safe, controlled physical environments for evaluating autonomous vehicle controllers in a cost-effective manner. These controllers often rely on the vehicle’s visual perception of its surroundings, achieved through the application of computer vision algorithms to image data acquired by its onboard cameras. Hence, it is critical that these algorithms accurately identify different objects in the testbed from any possible perspective of the vehicle. While some classes of objects are standardized (e.g., traffic signs and signals, lane markings), others can vary widely in appearance, in particular the myriad other vehicles that share the roadways and the structures of the nearby built environment.

Simulations of realistic driving scenarios can be used to generate a sufficient quantity and variety of image data for training machine-learning-based computer vision algorithms that are robust and highly accurate in diverse environments, as long as this synthetic image data closely matches real-world image data in appearance and content. However, existing open-source driving simulators, such as the widely-used simulator CARLA [10], lack tools to automatically reduce this disparity, known as the Sim2Real domain gap. Customized virtual driving environments with proprietary

This work was supported by NSF EAGER Award #2146691 and the Arizona State University Center for Human, Artificial Intelligence, and Robot Teaming (CHART).

S. S. Ulhas, S. Kannapiran, and S. Berman are with Arizona State University, School for Engineering of Matter, Transport and Energy, Tempe, AZ 85287 {sulhas, shenbagaraj, spring.berman}@asu.edu



Fig. 1: CHARTOPOLIS small-scale driving testbed [9].

3D models (buildings, roads, traffic signs, traffic lights, pedestrians, and other fixtures) such as digital twins of physical small-scale driving testbeds, require labor-intensive design of each individual asset. This limits the simulators’ utility as digital twins of these testbeds, in which vehicle controllers can be developed and evaluated safely and then reliably transferred to vehicles in the physical testbeds.

Common approaches to reducing the Sim2Real gap in vision-based applications include domain randomization, domain adaptation, dynamics randomization, system identification, and residual policy learning. Here we focus on domain adaptation and domain randomization, as they can be used for dataset augmentation to close the Sim2Real gap. We give a brief overview of these techniques below.

### A. Domain Randomization

When conducting data-driven machine learning experiments on a domain or environment that is unknown or hard to train in, knowledge must be transferred from an easily accessible domain. The former is called the *target domain* and the latter is the *source domain*. This enables the zero-shot transfer of features that are already trained on the source domain. To make this transfer compatible with a wide array of target domains, we need to ensure that features are learned without overfitting. This can be achieved through dataset augmentation by randomizing both the visual and physical parameters in a feature-rich dataset. The distribution of these parameters needs to be fine-tuned to enable policy generalization over multiple domains.

### B. Domain Adaptation

Domain adaptation enables a machine learning model trained in a source domain to generalize effectively to a target domain by adapting the data from the source domain

to resemble the data from the target domain, in terms of both its visual and dynamic properties. Domain adaptation can primarily be categorized into two approaches:

1) *Feature-Level Adaptation*: This method focuses on learning features that are common to both domains and can therefore be generalized easily. However, the feature extraction process for this approach needs to be meticulous and fine-tuned to match the application at hand. It is generally achieved by learning how predefined features translate for knowledge transfer from source to target domains.

2) *Pixel-Level Adaptation*: Pixel-level domain adaptation is achieved through *style-mixing* of images from the source domain to make them visually resemble images from the target domain. The most widely adopted method for this type of adaptation is the use of image-conditioned generative adversarial networks (GANs) [11]. StyleGAN [12] is a type of GAN that produces new images by mixing the styles of a set of images. StyleGAN2-ADA [12] by itself uses dataset augmentation to help train networks with limited data. We further augment our dataset using synthetic generations from StyleGAN.

In this paper, we demonstrate a novel Sim2Real transfer learning approach for object detection and segmentation that uses StyleGAN-based image data augmentation to perform domain adaptation, and thus close the appearance gap between synthetic and real-world image datasets. Our methodology primarily focuses on improving the diversity and realism in a synthetic image dataset generated from CARLA, demonstrated through two object classes: vehicles and buildings. We find that bridging the gap between small-scale testbeds and simulators will require further work to improve the capabilities of existing simulators. Specifically, data augmentation tools are needed in simulators to implement existing gap-bridging approaches involving domain adaptation and domain randomization.

## II. RELATED WORK

Sim2Real transfer has proved to be pivotal in programming machine learning algorithms for robotics, leveraging simulation as a reservoir of inexhaustible, accurately annotated data. Stocco et al. [13] provides a comprehensive study on existing simulation infrastructure available to study the Sim2Real gap and transfer techniques. Saxena et al. [14] utilized rendered objects to train a vision-based grasping model, while Gualtieri et al. and Viereck et al. [15], [16] explored Sim2Real transfer employing depth images, which simplify the complexity of real-world appearances. Despite the advantages of depth cameras, the ubiquity and cost-effectiveness of RGB cameras underscore the importance of developing systems that rely on monocular RGB images. However, the inclusion of depth data could potentially further improve the performance of generative adversarial networks (GANs) for domain adaptation. Moreover, GANs have been proven to work well as a data augmentation tool to improve the generalizability of machine learning models. GANs have been successfully demonstrated in CT image segmentation

tasks models [17] and handwriting and facial image classification tasks [18].

Domain randomization has been successfully applied to bridge the gap between synthetic and real-world data for a variety of tasks, including quadrotor navigation [19] and object grasping [20]–[23]. In particular, [20], [21] have investigated the use of randomized simulated environments for transferring robotic manipulation skills from simulation to real-world tasks. These efforts involve the generation of random textures, lighting, and camera positions within simulations, with the goal of testing the effect of randomization approaches in both visually simple virtual environments and in diverse, real-world robots. Tremblay et al. [24] applied domain randomization for car detection, whereas Sundermeyer et al. [25] proposed its use for object orientation estimation with training performed exclusively on synthetic renderings. Similar to our work, Khirodkar et al. [26] demonstrated the efficacy of combining domain adaptation with domain randomization in bridging the gap between synthetic and real data for object detection and pose estimation tasks.

Significant strides have been made in domain adaptation, especially for perception applications [27], [28], with research primarily bifurcating into feature-level and pixel-level adaptation. Feature-level adaptation seeks domain-invariant features, either by transforming pre-computed features between domains [29]–[32] or by developing a domain-invariant feature extractor, typically a convolutional neural network (CNN) [33]–[35]. Empirical evidence suggests the superiority of the latter approach in various classification tasks, with domain invariance often achieved by optimizing domain-level similarity metrics or responses from an adversarially trained domain discriminator [33], [34]. Pixel-level adaptation, on the other hand, focuses on modifying source domain images to resemble those from the target domain [36], [37], predominantly utilizing image-conditioned GANs [11].

Pioneering efforts in domain adaptation by Ganin and Lempitsky [33] introduced methods for ensuring that features learned by a model remain consistent despite shifts between domains. Following this, Zhang et al. [38] proposed a multichannel autoencoder designed to reduce the domain gap between real and synthetic data. Shrivastava et al. [39] further advanced domain adaptation through SimGAN, employing a GAN-based refiner to train systems for eye gaze estimation on synthetic images, thereby making these images' noise distribution more similar to that of real eye images. Distinct from this prior work, we introduce a novel transfer learning method that integrates both feature-level and pixel-level adaptation for enhancing Sim2Real transfer.

## III. ILLUSTRATIVE EXAMPLE: CHARTOPOLIS

Here, we describe components of the Sim2Real gaps between full-scale driving environments, our small-scale driving testbed CHARTOPOLIS (Fig. 1), and the CARLA driving simulator. Figure 2 [9] illustrates the control architecture of CHARTOPOLIS and its simulator counterpart. The CHARTOPOLIS testbed is designed to 1:18 scale,

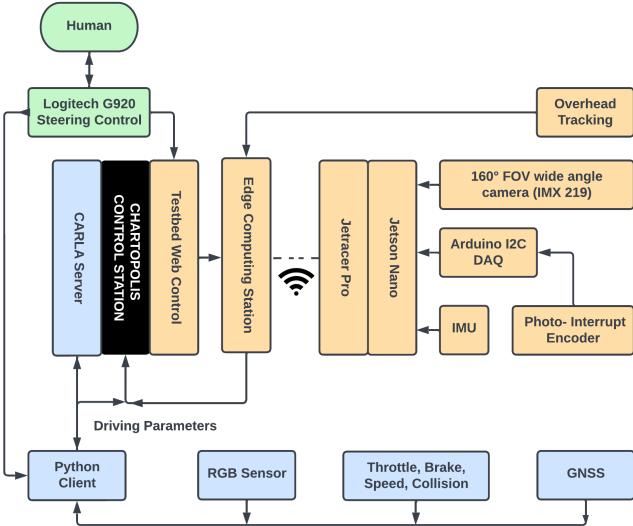


Fig. 2: CHARTOPOLIS control architecture, including components of the human–robot interface (green), physical testbed (orange), and CARLA driving simulator (blue) [9].



Fig. 3: Modified JetRacer AI Pro robotic car [40].

while other testbeds with similar capabilities [1], [41] are 1:10 scale. Robotic cars designed for these testbeds, such as the JetRacer (Fig. 3) and other scale-model vehicles, have very different dynamics than full-scale vehicles. We have found that the physics engines of driving simulators generally reproduce real-world vehicle dynamics, but cannot accurately reproduce the dynamics of small robot cars like the JetRacer. Hence, when including internal forces and actuation mechanics, the simulation environments for the small-scale and full-scale physical domains are too different to be generalizable for training algorithms that depend on the vehicles’ dynamics.

However, small-scale driving testbeds can still be useful for control strategy evaluation and human-robot interaction (HRI) experiments that do not require realistic vehicle dynamics. Given that the JetRacer and other robot cars look very different from real-world cars, object detection and other classification techniques cannot reliably recognize them under the same object class, unless trained specifically with

images of both small-scale robot cars and real-world cars. This motivates, in part, our transfer learning approach to close the Sim2Real gap.

Digital twinning between CARLA and CHARTOPOLIS is facilitated by the ROS-Bridge library support for CARLA. The sensors of the JetRacer robotic car are synchronized and controlled using ROS2, which is also compatible with the CARLA Python API. The possibility of using other simulation platforms to enhance the testbed’s functionality, such as the NVIDIA Omniverse [42], is currently being explored. CARLA is slated to add support for Unreal Engine 5 [43] and Omniverse, which would further close the appearance gap between the simulated environment and the testbed.

#### IV. APPROACH

A dataset consisting of 10772 synthetic images ( $800 \times 600$  pixels) was generated using the driving simulator CARLA, each image containing multiple instances of the two object classes: vehicles and buildings. Every image was generated with its corresponding set of semantic segmentation, instance segmentation, depth image, and ground truth bounding boxes for the object classes. We generated a new set of images that maintain the geometric integrity of the original synthetic data while infusing it with styles learned from a dataset of real-world images. This was achieved by training a StyleGAN-2-ADA model on a diverse real-world image dataset [44], [45] for each object class and then applying style mixing to the CARLA-generated synthetic images of such objects. The style-mixed images exhibit visual characteristics that are similar to those of real-world images of the same object class. For vehicles, these characteristics include vehicle make and color, while for buildings, they include architectural styles and façades. By replacing masked pixels corresponding to vehicles or buildings in the original synthetic images with these style-mixed versions, we effectively introduce a richer set of features that models must learn, thereby improving their generalizability to real-world data. The style-mixing process is carefully controlled to ensure that the fundamental feature attributes of each object are preserved, while still achieving a significant transformation in their appearance.

The following steps describe our data augmentation pipeline, illustrated in Fig. 4.

- 1) Generate 10772 *synthetic images* containing instances of each object class (vehicles and buildings) using the CARLA simulator.
- 2) Save 9963 object class ground truths using object detector bounding boxes from CARLA’s instance and segmentation masks.
- 3) In each image, generate black-and-white masks for object class ground truths using the ground truth from instance segmentation.
- 4) Extract masked pixels of each detected object and save a cropped version.
- 5) Extract latent representations of both synthetic and real images using a pre-trained StyleGAN encoder.
- 6) Use the resulting mixed latent vectors with the StyleGAN discriminator to synthesize the corresponding

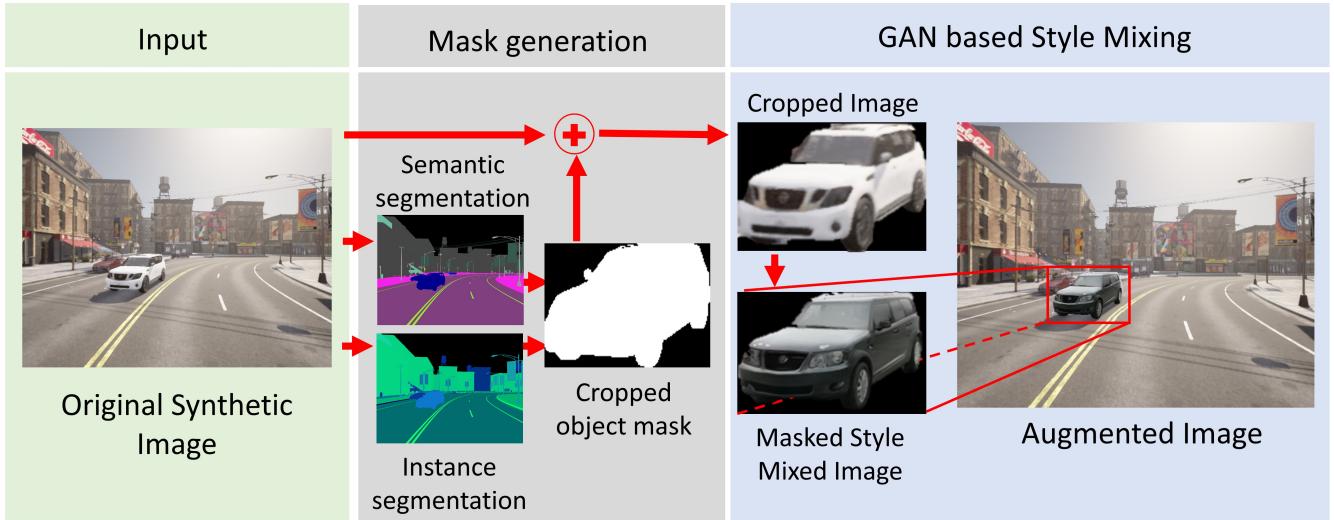


Fig. 4: Dataset augmentation pipeline using StyleGAN2-ADA for *style-mixing*.

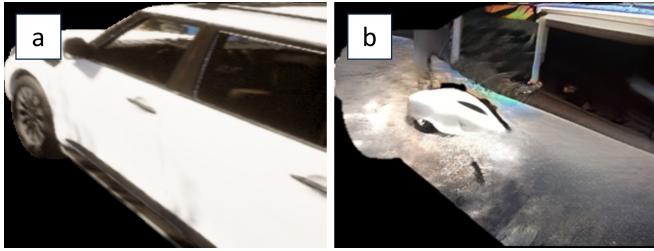


Fig. 5: Objects that are not fully visible in the image are not accurately reconstructed. (a) Original synthetic image from CARLA; (b) style-mixed image.

*projected images.*

- 7) Perform style mixing on the masked pixels in each projected image by combining the style features from the real images with the content features from the synthetic images.
- 8) Extract the masked pixels from the style-mixed images and discard the surrounding generated environment.
- 9) Replace the masked pixels in each original synthetic image with the pixels from the corresponding style-mixed version to form an *augmented image*.
- 10) Save the augmented images as a separate dataset to facilitate ablation studies with varied levels of data augmentation.

## V. RESULTS

### A. Vehicles

Understanding the depth and geometry of the images is an integral part of context-based image reconstruction. As shown in Fig. 5, monocular RGB images that do not fully depict the geometry of the object cannot be used for reconstruction with this approach using StyleGAN.

The model can also be pose-agnostic at times, as depicted in Fig. 6. By filtering out such edge cases, the dataset can be augmented to a degree with randomized vehicles

as seen in Fig. 7. These style-mixed images can then be easily integrated into our dataset, as shown in Fig. 4. This accomplishes the randomization of vehicles without using the built-in CARLA vehicle asset library or any other external libraries. Also, it is to be noted that the process of importing custom vehicles into CARLA [46] is a laborious process that might not yield results that are usable in vision-based experiments [47], [48] since a high-fidelity rendered vehicle model import will require several man-hours.

### B. Buildings

Creating the exact replica of CHARTOPOLIS in CARLA is challenging due to the import mechanism of buildings into the simulator. CARLA does not support whole models of buildings to be imported with ease. Instead, they have to be imported in parts, segregated by floors and other features like windows, shape, and textures. Further, buildings when imported as a single 3D model using the Unreal Editor, can have visual artifacts associated with lighting conditions. As seen in Fig. 8, models that do not have custom-defined UV lighting meshes can end up having shadow artifacts based on their shape and edges. When trying to implement domain randomization using StyleGAN, the model fails to understand the geometry of the buildings from its images generated through masks from instance segmentation. All projections (Fig. 10) result in interpolated texture randomization without retaining the geometry of the building. Segmentation-based masks also brought in missing pixels in the cropped images because of objects that appear between the building and the simulated camera sensor (Fig. 10a). StyleGAN2-ADA was custom trained on facades dataset which consists of 506 Building Facades and segmentations obtained from Pix2Pix datasets [51] and a CARLA-based custom building dataset to remedy these problems. As of writing this paper, this approach has been unsuccessful on buildings in CARLA.

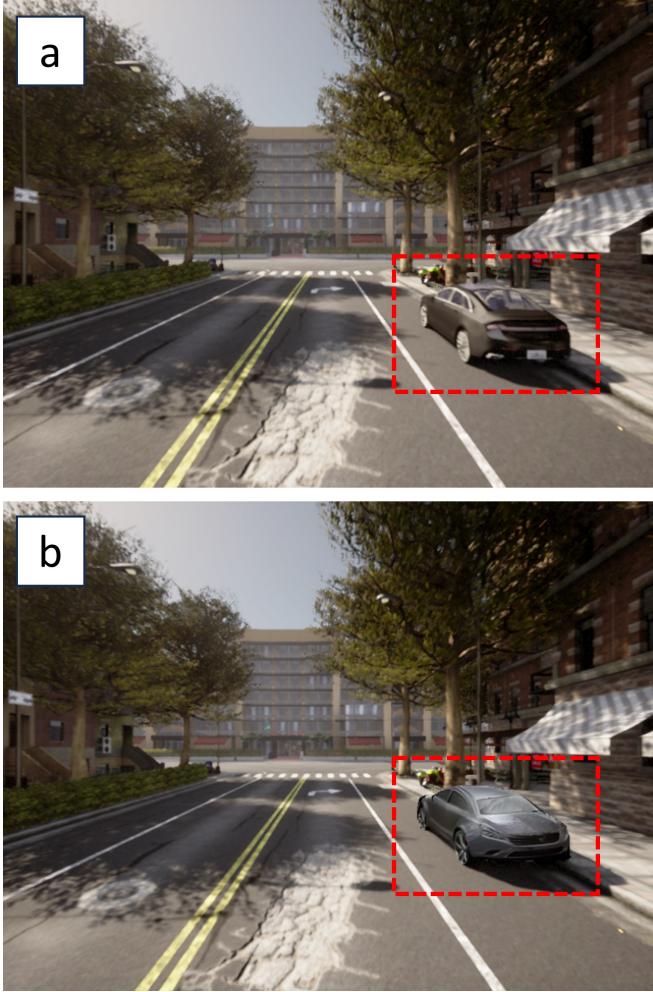


Fig. 6: Faulty (pose-agnostic) style-mixed randomization of a car image. (a) Original synthetic image from CARLA; (b) augmented image.

## VI. CONCLUSIONS AND FUTURE WORK

We have established that the implementation of domain adaptation and domain randomization techniques in digital twins of small-scale traffic testbeds requires case-by-case pipeline implementations for data augmentation. This approach opens up new avenues for research in these disciplines. This could help bypass photo-realistic asset generation that would require countless man-hours and highlights the potential of generative models in improving the utility of synthetic data for machine learning tasks.

There are various possible extensions of this work to address challenges in closing the Sim2Real gap. This paper only considers two object classes and does not address the appearance gap between small-scale robot cars and full-size vehicles. Therefore, further work needs to be done to expand support for other object classes that must be integrated into the testbed ecosystem and to develop reliable methods to close this appearance gap. Moreover, it is necessary to understand the source of unexpected discrepancies that our preliminary studies have revealed in the perfor-

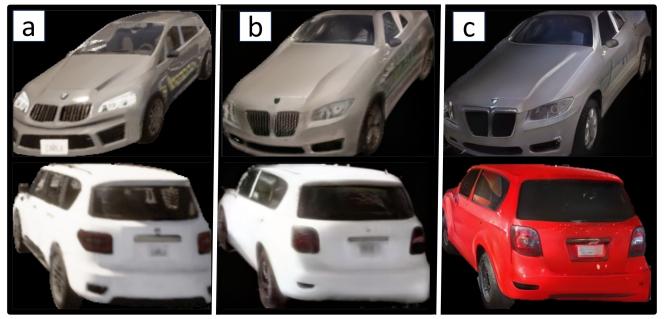


Fig. 7: Style mixing of synthetic images of cropped cars. (a) Original synthetic image from CARLA; (b) projected image; (c) augmented image.

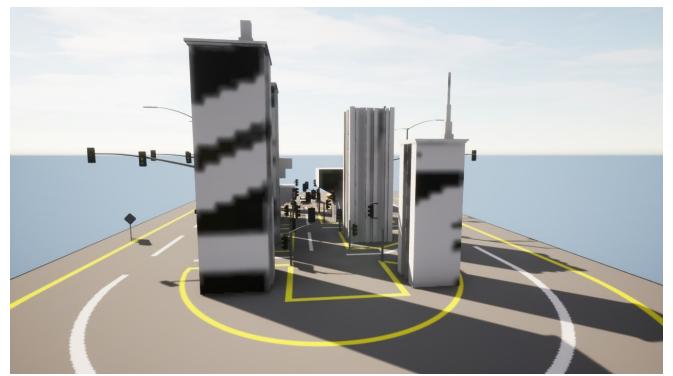


Fig. 8: Simulation of CHARTOPOLIS in CARLA. Lighting artifacts appear on the buildings due to the lack of custom-defined UV texture maps.

mance of object detection models that are trained on the unperturbed (baseline) dataset versus the dataset augmented using StyleGAN. An example is illustrated in Fig. 9, which shows that the model trained on the augmented dataset can exhibit higher confidence scores for synthetic images (from CARLA) and more true positive detections in some real-world images (from the KITTI dataset), but it can also exhibit lower confidence scores for other real-world images (from the Stanford car dataset) compared to the model trained on the baseline dataset. In addition, it is possible to perturb the shape of an object class to generate additional augmentations using GANs [52]. However, since GAN training is unstable without an intrinsic metric for evaluating the quality of the generated samples, the dataset would need to be carefully curated and the augmentations would need to be supervised. This approach would also introduce a host of challenges associated with shadow matching for the perturbed shapes. The computational trade-off between this approach and rendering entire new images using the graphics engine requires further investigation. Furthermore, the performance of approaches that use models other than GANs for generating augmented images can be compared to our method.

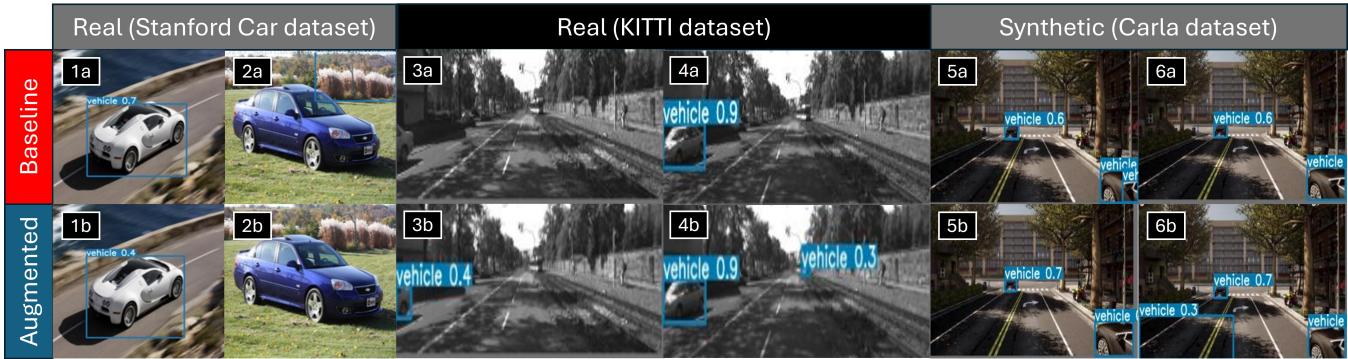


Fig. 9: Object detection performance when trained on the dataset generated solely from CARLA (Baseline dataset) vs. the style-mixed dataset (Augmented dataset). 1a, 1b, 2a, 2b: Images from the Stanford car dataset [49]; 3a, 3b, 4a, 4b: Images from the KITTI dataset [50]; 5a, 5b, 6a, 6b: Images generated using CARLA.

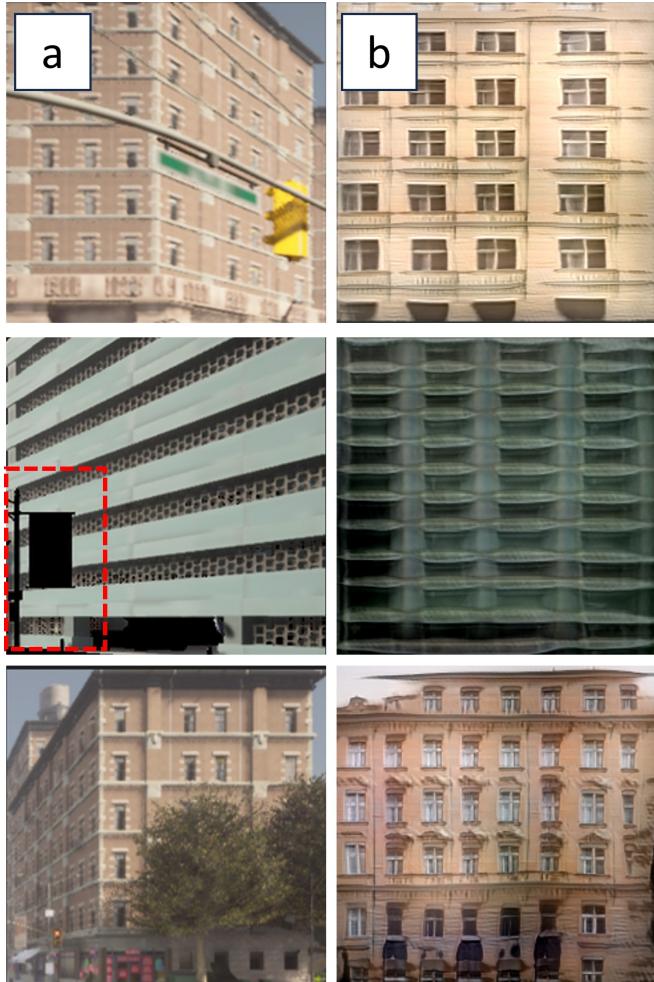


Fig. 10: StyleGAN-based reconstruction of buildings from a facades dataset based on the pix2pix dataset [51]. (a) Original synthetic image from CARLA; (b) augmented image.

## REFERENCES

- [1] M. Kloock, P. Scheffe, J. Maczijewski, A. Kampmann, A. Mokhtarian, S. Kowalewski, and B. Alrifae, “Cyber-physical mobility lab: An open-source platform for networked and autonomous vehicles,” in *2021 European Control Conference (ECC)*, 2021, pp. 1937–1944.
- [2] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang *et al.*, “Duckietown: an open, inexpensive and flexible platform for autonomy education and research,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1497–1504.
- [3] A. Stager, L. Bhan, A. Malikopoulos, and L. Zhao, “A scaled smart city for experimental validation of connected and automated vehicles,” *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 130–135, 2018.
- [4] N. Hyldmar, Y. He, and A. Prorok, “A fleet of miniature cars for experiments in cooperative driving,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3238–3244.
- [5] C. Berger, “From a competition for self-driving miniature cars to a standardized experimental platform: concept, models, architecture, and evaluation,” *arXiv preprint arXiv:1406.7768*, 2014.
- [6] B. Vincke, S. Rodriguez Florez, and P. Aubert, “An open-source scale model platform for teaching autonomous vehicle technologies,” *Sensors*, vol. 21, no. 11, p. 3850, 2021.
- [7] Quanser, “Quanser self-driving car research studio,” April 2021. [Online]. Available: <https://www.quanser.com/products/self-driving-car-research-studio/>
- [8] J. Dong, Q. Xu, J. Wang, C. Yang, M. Cai, C. Chen, Y. Liu, J. Wang, and K. Li, “Mixed Cloud Control Testbed: Validating Vehicle-Road-Cloud Integration via Mixed Digital Twin,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2723–2736, Apr. 2023.
- [9] S. S. Ulhas, A. Ravichander, K. A. Johnson, T. P. Pavlic, L. Gharavi, and S. Berman, “CHARTOPOLIS: A small-scale labor-art-ory for research and reflection on autonomous vehicles, human–robot interaction, and sociotechnical imaginaries,” in *Workshop on Miniature Robot Platforms for Full Scale Autonomous Vehicle Research, IROS 2022: IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, arXiv preprint arXiv:2210.00377.
- [10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on Robot Learning*. PMLR, 2017, pp. 1–16.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [12] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 104–12 114, 2020.
- [13] A. Stocco, B. Pulfer, and P. Tonella, “Mind the gap! a study on the transferability of virtual vs physical-world testing of autonomous driving systems,” *IEEE Transactions on Software Engineering*, 2022.
- [14] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng, “A vision-based system for grasping novel objects in cluttered environments,” in *Robotics Research: The 13th International Symposium (ISRR)*. Springer, 2011, pp. 337–348.
- [15] A. ten Pas, M. Gualtieri, K. Saenko, and R. P. Jr., “Grasp pose detection in point clouds,” *CoRR*, vol. abs/1706.09911, 2017. [Online]. Available: <http://arxiv.org/abs/1706.09911>
- [16] U. Viereck, A. ten Pas, K. Saenko, and R. W. Platt, “Learning a visuomotor controller for real world robotic grasping using simulated

- depth images,” *ArXiv*, vol. abs/1706.04652, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:23439582>
- [17] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, “Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks,” *Scientific Reports*, vol. 9, no. 1, p. 16884, 2019.
- [18] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” *arXiv preprint arXiv:1711.04340*, 2017.
- [19] F. Sadeghi and S. Levine, “Cad2rl: Real single-image flight without a single real image,” *arXiv preprint arXiv:1611.04201*, 2016.
- [20] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.
- [21] S. James, A. J. Davison, and E. Johns, “Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task,” in *Conference on Robot Learning*. PMLR, 2017, pp. 334–343.
- [22] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” *arXiv preprint arXiv:1710.06542*, 2017.
- [23] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3803–3810.
- [24] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, “Training deep networks with synthetic data: Bridging the reality gap by domain randomization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 969–977.
- [25] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, “Implicit 3d orientation learning for 6d object detection from rgb images,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 699–715.
- [26] R. Khirodkar, D. Yoo, and K. Kitani, “Domain randomization for scene-specific car detection and pose estimation,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1932–1940.
- [27] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, “Visual domain adaptation: A survey of recent advances,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.
- [28] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” *arXiv:1702.05374*, 2017.
- [29] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *AAAI Conference on Artificial Intelligence*, 2016.
- [30] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *International Conference on Computer Vision (ICCV)*, 2011.
- [31] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *Journal of Machine Learning Research (JMLR)*, 2016.
- [32] M. Long and J. Wang, “Learning transferable features with deep adaptation networks,” in *International Conference on Machine Learning (ICML)*, 2015.
- [33] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 1180–1189.
- [34] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, “Domain separation networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2016.
- [35] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [36] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *International Conference on Computer Vision (ICCV)*, 2017.
- [38] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam, “Learning classifiers from synthetic data using a multichannel autoencoder,” *arXiv preprint arXiv:1503.03163*, 2015.
- [39] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2107–2116.
- [40] Waveshare, “JetRacer Pro AI kit, high speed AI racing robot powered by Jetson Nano, Pro version,” 2023. [Online]. Available: <https://www.waveshare.com/jetracer-pro-ai-kit.htm>
- [41] M. O’Kelly, H. Zheng, D. Karthik, and R. Mangharam, “F1tent: An open-source evaluation environment for continuous control and reinforcement learning,” in *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, ser. Proceedings of Machine Learning Research, H. J. Escalante and R. Hadsell, Eds., vol. 123. PMLR, 08–14 Dec 2020, pp. 77–89. [Online]. Available: <https://proceedings.mlr.press/v123/o-kelly20a.html>
- [42] “NVIDIA Omniverse Cloud APIs,” Mar 2024. [Online]. Available: <https://carla.org/2024/03/18/nvidia-omniverse-cloud-apis/>
- [43] “CARLA Unreal Engine 5 teaser,” Mar 2024. [Online]. Available: <https://www.youtube.com/watch?v=J4pHRM1Q5nQ>
- [44] J. Pinkney, “Awesome Pretrained StyleGAN2,” <https://github.com/justinpinkney/awesome-pretrained-stylegan2>, 2023, accessed on: 2024-02-04.
- [45] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop,” *arXiv preprint arXiv:1506.03365*, 2015.
- [46] CARLA Team, “Adding a New Vehicle - CARLA Documentation,” [https://carla.readthedocs.io/en/latest/tuto\\_A\\_add\\_vehicle/](https://carla.readthedocs.io/en/latest/tuto_A_add_vehicle/), 2023, accessed on: 2024-02-04.
- [47] TUM-AVS, “Edgar Digital Twin Tools for CARLA,” [https://github.com/TUM-AVS/edgar\\_digital\\_twin/tree/main/tools/CARLA](https://github.com/TUM-AVS/edgar_digital_twin/tree/main/tools/CARLA), 2023, accessed on: 2024-02-04.
- [48] P. Karle, T. Betz, M. Bosk, F. Fent, N. Gehrke, M. Geisslinger, L. Gressenbuch, P. Hafemann, S. Huber, M. Hübner *et al.*, “Edgar: An autonomous driving research platform—from feature development to real-world application,” *arXiv preprint arXiv:2309.15492*, 2023.
- [49] T. Kramberger and B. Potočnik, “LSUN-Stanford car dataset: Enhancing large-scale car image datasets using deep learning for usage in GAN training,” *Applied Sciences*, vol. 10, no. 14, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/14/4913>
- [50] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research (IJRR)*, 2013.
- [51] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.
- [52] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, “EditGAN: High-precision semantic image editing,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.