Scene Generation from Backgrounds to Objects and Anything in Between: A Deep Learning Robotics Survey

Bouchra Bouqata*
Amazon Robotics, Boston, MA, USA, bouqatab@amazon.com
Krishna Aswani
Amazon, San Francisco, CA, USA, aswanika@amazon.com
David Bailey
Amazon, Seattle, WA, USA, dbail@amazon.com

The recent rapid progress of deep learning algorithms in generating realistic images, especially in Generative Adversarial Networks (GAN) and Variational Auto-Encoders (VAE), has helped advance new applications. Examples of such applications range from generating and manipulating new synthetic data for self-driving cars, to building/urban architectures, to interior design, and gaming. Furthermore, several applications have benefited from deep learning generative advancement, such as robotics manipulations in structured and unstructured environments, virtual fashion clothes try-on, and item identification on the go. This survey paper provides a review of techniques for image generation from background outdoor scenes, to building facades and objects, and anything in between. In particular, we will cover scene generation such as outdoor landscapes, building facades and indoor scenes. For each category, we will compare the existing state of the art algorithms and techniques, and discuss their performance and gaps limitations on a wide variety of inputs. Additionally, we will discuss challenges and future trends to advance the state of the art in realistic image generation.

Additional Keywords and Phrases: Deep Learning, Generative Adversarial Networks, Autonomous Systems, Robotics, Synthetics Data Generation.

INTRODUCTION

Recent advances in deep learning algorithms in generating realistic images, specifically Generative Adversarial Networks (GAN) [1-14], have inspired break-throughs in multiple applications. Examples of such applications range from generating and manipulating new synthetic data for self-driving cars [15-16], to building/urban architectures [24, 25], to interior design [26, 27], and gaming [28]. Furthermore, several applications have benefited from deep learning generative advancement, such as robotics manipulations in structured and unstructured environments, virtual fashion clothes try-on, and item identification on the go.

Some of the most prevalent applications of deep learning are object recognition and detection models in robotics and autonomous driving systems (e.g., self-driving cars, self-manipulating robots). Deep Learning helps these systems to learn primitives and features that provide an input to a downstream robotic control system [16] through perception [17,18], prediction [19,20] and planning [21]. Furthermore, autonomous driving system evaluation requires the ability to realistically replay a large set of diverse and complex scenarios in simulation capturing sensor properties, seasons, time of day, and weather. Developing simulators that support the levels of realism required for autonomous system evaluation is a challenging task. There are many ways to design simulators, including simulating mid-level object representations [22, 23]. Frameworks for autonomous driving that support realistic sensor simulation are traditionally built on top of gaming engines such as Unreal or Unity [22]. The environment and its object models are created and arranged manually, to approximate real-world scenes of interest. The overall process is time-consuming and not scalable. Real-life testing of robotic systems is expensive and slow.

Furthermore, a fundamental challenge of applying deep supervised learning and deep reinforcement learning to several applications such as robotics and self-driving cars is data availability and real-world data is difficult to collect. In fact, state-of-the-art deep learning algorithms are data-hungry and need millions to tens of millions of labeled examples with variety in it for each application domain [29, 30]. However, obtaining labeled real-world data for robotics or self-driving cars, is challenging because robots and autonomous cars are expensive, 3-dimensional data can be particularly difficult to label [19], and collecting data with robotic systems or autonomous vehicles can be dangerous [20]. Moreover, setting up hardware and sensors to collect data, training a team of labelers, and pruning labeling errors adds months to R&D and weighs heavily on project budgets. Synthetic data enables companies to test their robotics solutions in thousands of simulations, improving their robots and complementing expensive real-life testing.

Therefore, the role of synthetic data in Deep Learning is increasing rapidly. This is not only true for Deep Learning algorithms training but it can also play an important role in the creation of algorithms for image recognition and similar tasks that are becoming the baseline for Artificial Intelligence (AI) [29, 30]. Furthermore, the latest innovations in machine learning and deep learning techniques are promising to help generate synthetic data with higher quality and in larger quantities. We call it Machine Learning for Machine Learning.

In the next sections, we briefly review the state-of-the-art in realistic scene generation such as outdoor landscapes and building facades and indoor scene such as bedrooms. For each category, we will compare the existing state-of-the art algorithms and techniques, discuss their performance and limitations on a wide variety of inputs. Additionally, we will discuss challenges and future trends to push the state of the art in realistic image generation.

1 SCENE GENERATION FROM OUTDOOR LANDSCAPES TO BUILDING FACADES TO INDOOR SCENES

The most prevailed applications of deep learning are object recognition and detection models in robotics and Autonomous driving systems (self-driving cars, self-manipulating robots) such as robotic control system [16] through perception [17,18], prediction [19,20] and planning [21]. Furthermore, autonomous driving system evaluation requires the ability to realistically replay a large set of diverse and complex scenarios in simulation capturing sensor properties, seasons, time of day, and weather. Developing simulators that support the levels of realism required for autonomous system evaluation is a challenging task. The traditional way of creating simulated environments and it objects are created and arranged manually, to approximate real-world scenes of interest and are often built on top of game engines [22, 23]. The overall process is time-consuming and not scalable. Synthetic data enables companies to test their robotics solutions in thousands of simulations, improving their robots and complementing expensive real-life testing. On the other hand, Synthetic data enables healthcare data professionals to allow the public use of record data while still maintaining patient confidentiality.

A common approach to deal with a limited number of labels is data augmentation [1]. Translation, noise, and other deformations can often be applied without changing the labels, thereby effectively increasing the number of training samples and reducing overfitting. In [39] the authors, propose to automatically learn augmentations with GANs. Just as GANs learn a generative model of data, conditional GANs (cGANs) learn a conditional generative model [1]. This makes cGANs suitable for image-to-image translation tasks, where it is conditioned on an input image to generate a corresponding output image. Figures 1(a), 1(b) and 1(c) show examples of results from the algorithm pix2pix [39] on Cityscapes labels to image translation, building facades labels to image translation, and day to nigh image translation, respectively.

Realistic image manipulation is challenging because it requires modifying the image appearance in a user-controlled way, while preserving the realism of the result. Understanding and modeling the natural image manifold has been a long- standing open research problem. But in the past recent years, there has been rapid advancement, fueled largely by the development of the generative adversarial networks [1]. In particular, several earlier papers [40 - 43] have shown visually impressive results sampling random images drawn from the natural image manifold. However, two reasons prevent these advances from being useful in practical applications at this time. First, the generated images, while good, are still not quite photo-realistic (plus there are practical issues in making them high resolution).

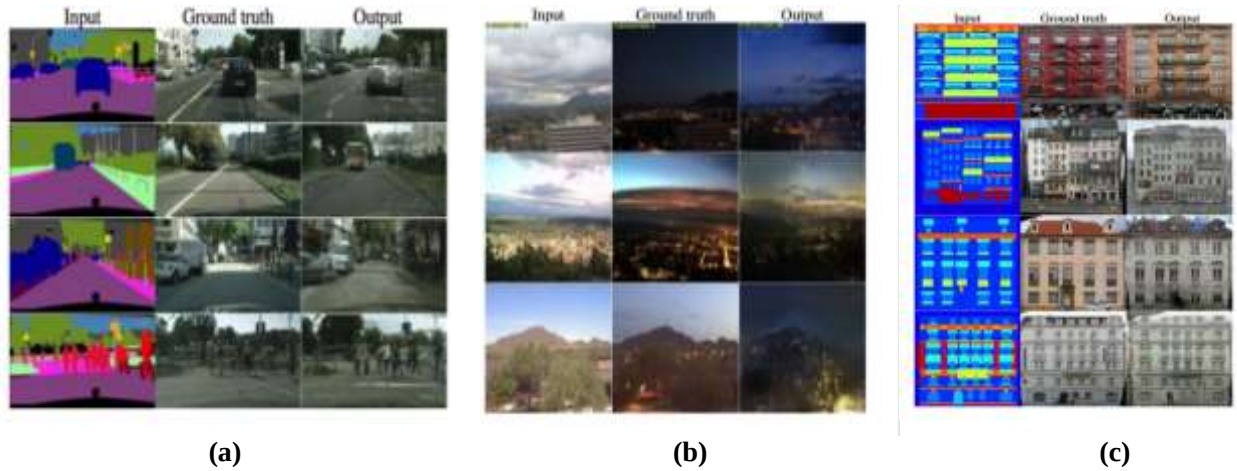

Figure 1: Examples of results from pix2pix paper [39] on (a) Cityscapes labels -> photo compared to ground truth and (b) Day-> night photo compared to ground truth (c) Facade labels -> photo compared to ground truth

Second, these generative models are set up to produce images by sampling a latent vector-space, typically at random. So, these methods are not able to create and manipulate visual content in a user-controlled fashion. In [44], the authors proposed DCGAN, where they used a GAN to learn the manifold of natural images, but without employing it for image generation. Instead, they used it as a constraint on the output of various image manipulation operations, to make sure the results lie on the learned manifold at all times. This idea reenabled to reformulated several editing operations, specifically color and shape manipulations, in a natural and data-driven way. The model automatically adjusts the output keeping all edits as realistic as possible. Figure 2 shows some examples of DCGAN generating a new image from scratch based on user's initial drawings (or renderings)/landmarks, called in the form of a segmentation map with labels. The user interactively can edit the generated images and the model transfers the resulting changes in shape and color back to the original image. The low resolution and missing texture and details as well as low quality on general imagery limits how far the editing approach of DCGAN can get, as shown in Figure 2.

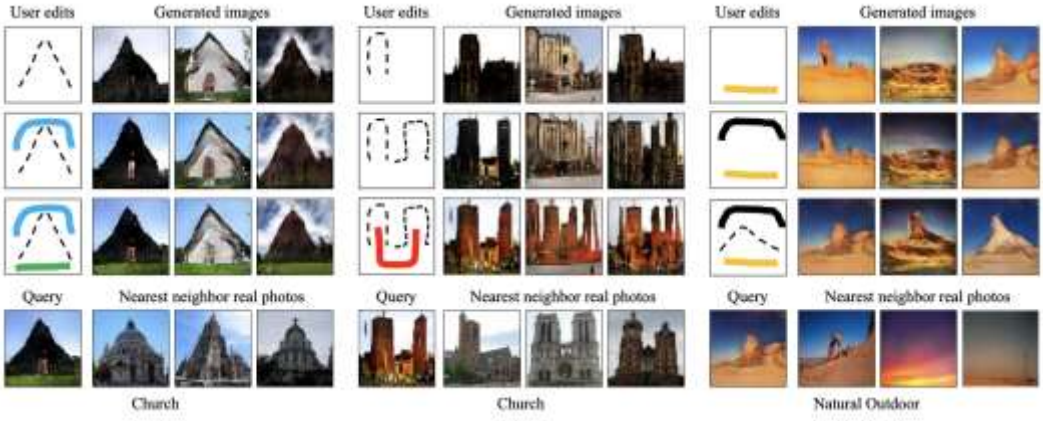

Figure 2: Examples of results from DCGAN paper [44] on Interactive image generation where the user uses a brush tool to generate an image from strokes and adding more to refine the results. The last raw shows the most similar real images to the generated image.

Specifically, there is a need to control the layout of the generated image using a segmentation mask that has labels for each semantic region and "add" realistic styles to each region according to their labels. For example, a face generation application would use region labels like eyes, hair, nose, mouth, etc. and a landscape painting application would use labels like water, forest, sky, clouds, etc. While there are multiple proposed architectures [39, 47, 48, 49] that address synthesizing photorealistic images given an input semantic layout, the most popular method for is spatially-adaptive normalization (SPADE, also known as GauGAN) [45] as shown in Figure 3. SPADE is a spatially-adaptive normalization, which is a simple but effective layer for synthesizing photorealistic images given an input semantic layout. Previous method directly feed the semantic layout as input to the deep network, which is then processed through stacks of convolution, normalization, and

nonlinearity layers. This is suboptimal as the normalization layers tend to “wash away” semantic information. To address the issue, SPADE uses the input layout for modulating the activations in normalization layers through a spatially-adaptive, learned transformation.

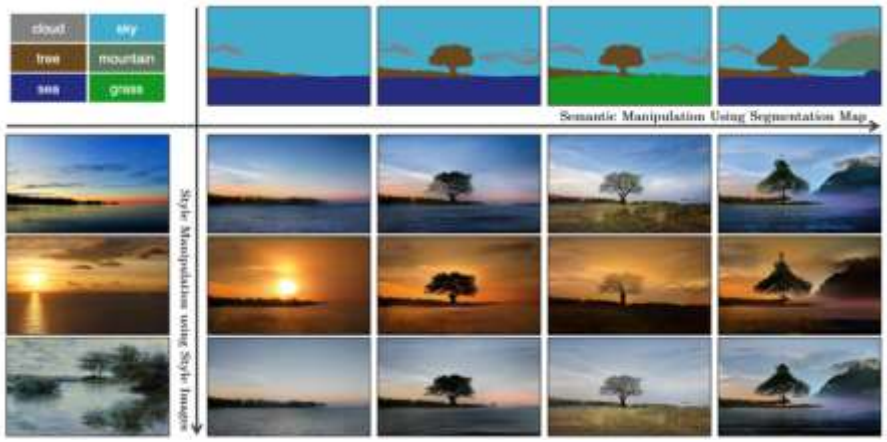

Figure 3: Examples of results from SPADE [45] where a segmentation map (with labels specifying the region of interest such as sky, grass and a tree) and a target style image are provided as an input and a landscape image is generated.

However, SPADE image generation is limited to only one style for each image. This can be a problem if for example different output styles are desired for the image’s different compositional elements. Recent studies [50, 51, 52] have demonstrated that higher quality results can be obtained if style information is injected as normalization parameters in multiple layers in the network, for example using AdaIN [53]. Therefore, in [46] the authors proposed SEAN that aims at addressing SPADE’s architectural shortcomings of only inputting its style information at the beginning of network processing. Figure 4, shows a visual comparison of results from pix2pix [39], SPADE [45] and SEAN [46] results on segmentation labels for ADE20K (indoor scenes: bedrooms), Cityscapes (outdoor urban scenes), Facades (building facades for outdoor scenes).

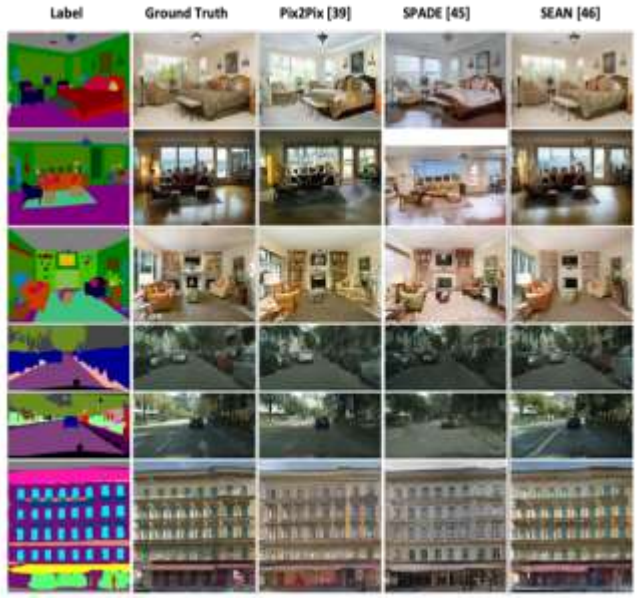

Figure 4: Visual comparison of pix2pix [39], SPADE [45] and SEAN [46] algorithms of semantic image synthesis results on the ADE20K, CityScapes and Facades datasets: images source [46]

Based on experiments with SPADE, some types of segmentation labels “bleed” into other labels and the generated images have a lot of artifacts as shown in Figure 5. In order to better control the quality of the generated images by SPADE, we recommend careful choice of the segmentation labels to use first. In fact, we

recommended to use a “strong” segmentation label first and then add on the “weaker” segmentation labels to create a separation. Examples of “strong” segmentation labels include: Grass, Rock, Mountain, Sand, Bush, Wall-stone, Clouds, Sky, Sea, and Stone.

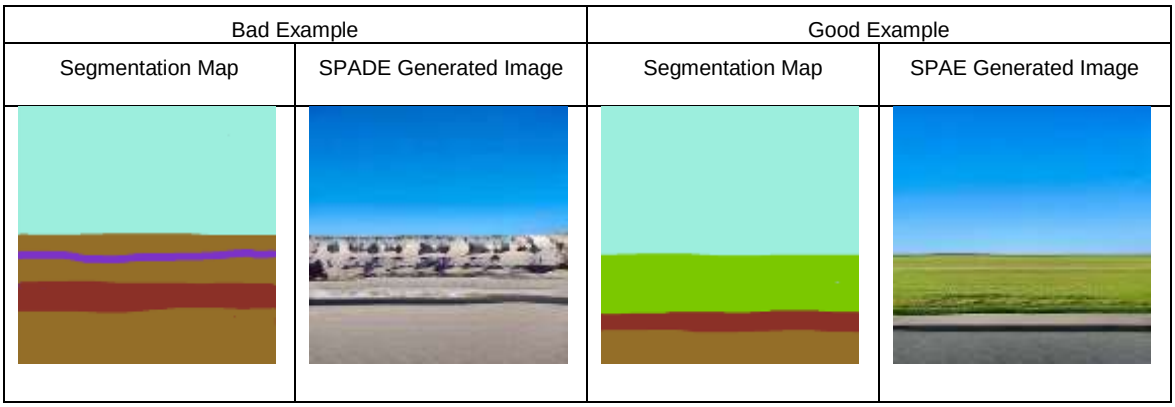

Figure 5: Generated images using SPADE (GauGAN) [45] that shows a good and bad example of strong (Grass) vs. weak (Gravel) labels to control the sky vs. pavement and gravel

CONCLUSION

In this survey paper, we provided a review of deep learning techniques for image generation from diverse areas including background outdoor scenes to building facades that can provide synthetic data to advance several applications including autonomous driving and robotics. However, there are several challenges when applications need very specific data. Most of the deep learning based synthetic data generation algorithms, while providing great diverse and new data, fall short on controlling the generation process to provide the best distribution and content variety for the application at hand. For future work, it would be interesting to explore procedural synthetic data generation covering cases when simply placing the objects at random is not enough. The nature of the model used for procedural generation may differ depending on the specific field of application. This will allow for both learning from real or manually prepared synthetic data and then generating new samples.

REFERENCES

[1] 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, 2014, pp. 2672–2680.

[2] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” in Advances in neural information processing systems, 2016, pp. 469–477.

[3] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.

[4] M. O. Turkoglu, L. Spreeuwers, W. Thong, and B. Kicanaoglu, “A layer-based sequential framework for scene generation with GANs,” in Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, United States, 2019.

[5] H. Wu, S. Zheng, J. Zhang, and K. Huang, “GP-GAN: Towards realistic high-resolution image blending,” arXiv preprint arXiv:1703.07195, 2017.

[6] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O’Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, “SalGAN: Visual saliency prediction with generative adversarial networks,” arXiv preprint arXiv:1701.01081, 2017.

[7] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, “Training generative neural networks via maximum mean discrepancy optimization,” arXiv preprint arXiv:1505.03906, 2015.

[8] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, “Pose guided person image generation,” in Advances in Neural Information Processing Systems, 2017, pp. 406–416.

[9] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in Advances In Neural Information Processing Systems, 2016, pp. 613–621.

[10] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, “High-resolution image inpainting using multi-scale neural patch synthesis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6721–6729.

[11] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in Proceedings of the 34th International Conference on Machine Learning, vol. 70. JMLR, 2017, pp. 2642–2651.

[12] Y. Li, K. Swersky, and R. Zemel, “Generative moment matching networks,” in International Conference on Machine Learning, 2015, pp. 1718– 1727.

[13] J.-Y. Zhu, P. Krahenb ¨ uhl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in ¨ European Conference on Computer Vision. Springer, 2016, pp. 597–613.

[14] C. Lassner, G. Pons-Moll, and P. V. Gehler, “A generative model of people in clothing,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 853–862.

[15] Chelsea Finn et al. “Deep spatial autoencoders for. visuomotor learning”. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2016, pp. 512–519.

[16] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, “SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving”, in Proceedings of CVPR 2020, pp. 11118 – 11127

[17] K. He, G. Gkioxari, P. Dollar, and R. Girshick. “Mask r-cnn.” in ICCV, 2017

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. “Ssd: Single shot multibox detector.” in ECCV, pp. 21–37. Springer, 2016.
[19] M. Bansal, A. Krizhevsky, and A.Ogale. ”Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst.” in RSS, 2019.
[20] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction.” In arXiv preprint arXiv:1910.05449, 2019
[21] M. Everett, Y.F. Chen, and J. P. How. “Motion planning among dynamic, decision-making agents with deep reinforcement learning.” in IROS, 2018.
[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “Carla: An open urban driving simulator.” In arXiv preprint arXiv:1711.03938, 2017
[23] M. Bansal, A. Krizhevsky, and A. Ogale. “Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst.” in RSS, 2019.
[24] S. Chaillou, “ ArchiGAN Intelligence x Architecture,” In: Yuan P., Xie M., Leach N., Yao J., Wang X. (eds) Architectural Intelligence. Springer, Singapore, 2020
[25] Y. Yoshimura, B. Cai, Z. Wang and C. Ratti, “Deep Learning Architect: Classification for Architectural Design through the Eye of Artificial Intelligence,” https://arxiv.org/pdf/1812.01714.pdf , 2018
[26] S. Zhang, Z. Han, Y.-K. Lai, M. Zwicker and H. Zhang, “Stylistic scene enhancement GAN: mixed stylistic enhancement generation for 3D indoor scenes.” In Visual Computer 35, pp. 1157-1169, 2019
[27] X. Li, H. Li, C. Wang, X. Hu and W. Zhang, “Visual-attention GAN for interior sketch colorization,” In IET Image Processing, pp. 997-1007, 2021
[28] R. Rodriquez Torrado, A. Khalifa, M. Cerny Green, N. Justesen, S. Risi and J. Togelius, “Bootstrapping Conditional GANs for Video Game Level Generation,” in IEEE Conference on Games (CoG) 2020.
[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE. 2009, pp. 248–255.
[30] Y. Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: CoRR abs/1609.08144 (2016). url: http://arxiv.org/abs/1609.08144.
[31] G. Sreenu, M. A. Saleem Durai, “Intelligent video surveillance: a review through deep learning techniques for crowd analysis,” in Journal of Big Data, 2019.
[32] L. Shana, C. S. Christopher, “Video Surveillance using Deep Learning - A Review,” In ICRAECC, 2019.
[33] D. Hazra and Y.-C. Byn, “SynSigGAN: Generative Adversarial Networks for Synthetic Biomedical Signal Generation,” in MDPI Biology, 2020.
[34] H. Zhang, Z. Huang, “Medical Image Synthetic Data Augmentation Using GAN,” in Proceedings of the 41th International Conference on Computer Science and Application Engineering, 2020.
[35] L. Bargsten, A. Schlaefer, “SpeckleGAN: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing,” in International Journal of Computer Assisted Radiology and Surgery, pp: 1427-1436, 2020
[36] J. Islam, Y. Zhang, “GAN-based synthetic brain PET image generation,” In Brain Informatics, 2020.
[37] D. Garcia Torres, “Generation of Synthetic Data with Generative Adversarial Networks,” Thesis KTH Royal Institute of Technology Information and Communication Technology, Second Cycle, Sweden 2018
[38] A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, “Learning to compose domain-specific transformations for data augmentation,” 2017, in arXiv preprint arXiv:1709.01643.
[39] P. Isola, J.-Y. Zhu, T. Zhou. A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” 2018, arXiv preprint arXiv:1611.07004v3.
[40] A. Radford, L. Metz, S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in ICLR, 2016.
[41] D. P. Kingma, M.Welling, “Auto-encoding variational bayes,” In ICLR, 2014.
[42] E. Denton, S. Chintala, A. Szlam, R. Fergus, “Deep generative image models using a laplacian pyramid of adversarial networks,” In NIPS, 2015.
[43] A. Dosovitskiy, T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” In NIPS, 2016.
[44] J.-Y. Zhu, P. Krahenbuhl, E. Shechtman and A. A. Efros, “Generative Visual Manipulation on the Natural Image Manifold,” in arXiv preprint arXiv:1609.03552v3
[45] T. Park, M.-Y. Liu, T.-C. Wang, J.-Y. Zhu, “Semantic Image Synthesis with Spatially-Adaptive Normalization,” in CVPR 2019.
[46] P. Zhu, R. Abdal, Y. Qin, P. Wonka, “SEAN: Image Synthesis with Semantic Region-Adaptive Normalization,” in arXiv preprint arXiv:1911.12861v2, 2020.
[47] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in arXiv preprint arXiv:1707.09405v1, 2017.
[48] X. Qi, Q. Chen, J. Jia, and V. Koltun, “Semi-parametric image synthesis”, in arXiv preprint, arXiv:1804.10992v1, 2018.
[49] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[50] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in arXiv preprint, arXiv:1812.04948v3, 2018
[51] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, “Few-shot unsupervised image-to-image translation,” in ICCV, 2019
[52] Y. Alharbi, N. Smith, and P. Wonka, “Latent filter scaling for multimodal unsupervised image-to-image translation,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1458–1466, 2019
[53] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in arXiv preprint, arXiv:1703.06868v2, 2017