Informatics Bulletin, Faculty of Computers and Artificial Intelligence, Helwan University
Published Online *Vol 3 Issue 1, January 2021*
(https://fcihib.journals.ekb.eg)

Semantic Image Synthesis Manipulation for Stability Problem using Generative Adversarial Networks: A Survey

Karim Magdy, Ghada Khoriba, Hala Abbas
Computer Science Department - Faculty of Computer and Artificial Intelligence - Helwan University - Cairo, Egypt
karimagdy22@gmail.com, ghada_khoriba@fci.helwan.edu.eg, hala.abbas@fci.helwan.edu.eg

***Abstract*—Semantic image synthesis aims to transfer semantic label maps to photo-realistic images. Despite the significant successes achieved to date by state-of-the-art methods, there is a major gap between the quality of photo-realistic images and the quality of synthesized images. This gap is caused by training stability problems such as diversity of image generation, and the lack of semantic information. Also, this kind of task still poses a significant problem concerning computational time. Furthermore, opening a way to use a consistent and unified loss function for different tasks, datasets, and various generated images will be considerable assistance to tackle the challenges of training stability. In this survey, we discussed the Generative Adversarial Networks (GANs) model because of the ability to synthesize good samples directly. A literature discussion between different methods used to improve the result of GAN have been discussed which aims to produce better results and generate more samples. Moreover, a combination of different techniques from different fields was discussed.**

***Index Terms*—Generative Adversarial Networks (GANs), Semantic Image Synthesis, Local Binary Pattern (LBP)**

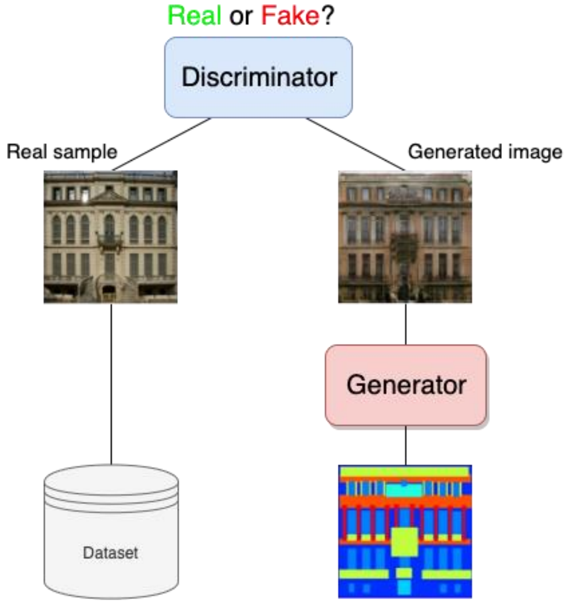


Fig. 1. GAN Architecture.

I. Introduction

Nowadays, there are a lot of tasks related to computer vision, computer graphics, and image processing fields, which have been developed to solve real-life problems. Semantic image synthesis is one of these tasks. Semantic image synthesis is an image-to-image translation task and a specific form of conditional image synthesis, which is converting a semantic label map to a photorealistic image. This task has a wide range of applications such as image editing, content generation, and medical area [1–10], and it can be solved by one of the generative model techniques. Most state-of-the-art techniques are based on Generative Adversarial Networks (GANs) [11]. GAN has gained enormous attention in the machine learning field because of their algorithms depend on directly sample from the underlying density function without relying on any assumptions such as in variational, Markov chains, or other methods related to generative models. Furthermore, GAN can learn complex data and high-dimensionality and is capable of generating realistic samples from latent space. These features have made it possible for GAN to enjoy great success. The training phase of the generator and the discriminator is illustrated in Fig 1.

Despite the significant successes achieved to date, applying GAN to the semantic image synthesis task still poses significant challenges such as the diversity of image generation, computational time, lack of semantic information, and training stability. Also, a few numbers of research tries to combine GAN with other techniques whether its computer vision, computer graphics, or image processing technique to improve GAN results.

The main goal of this paper is to illustrate the Semantic Image Synthesis task and some challenges related to this task, and some techniques used to tackle some of these challenges. One of the important techniques is the generative models and their variant probability distributions, also we discuss the importance of normalization layers for the Semantic Image Synthesis task. Besides, we mention Local Binary Pattern (LBP) technique which is one of the most common texture feature extraction techniques. Finally, we propose a combination of all of the above-mentioned techniques to tackle these challenges.

II. Methodology

The need for Forward-Looking Techniques to tackle real-world problems increases over time. These techniques will play a significant contribution, reduce the gap between current results and expected results, and drive the entire literature in the right direction to find an optimal solution for these problems. Besides, we want these techniques to be able to explain how their results and assumptions have been made. To bring us one step closer achieve that, we need a combination of different techniques from one or more fields. A discussion about semantic image synthesis and the most recent techniques used to solve this task is present in the following subsections.

A. Semantic Image Synthesis

Semantic Image Synthesis is the task of generating photo-realistic images using only a semantic segmentation map which makes this task have a wide range of applications. As mentioned in the previous section the semantic image synthesis task is facing many challenges such as the diversity of image generation, computational time, lack of semantic information, and training stability. Fig 2 illustrates Semantic Image Synthesis. Subsequent sections will discuss the recent techniques used in the Semantic Image Synthesis task such as Generative Models, Normalization Layers, and some common texture feature extraction techniques like Local Binary Pattern (LBP).

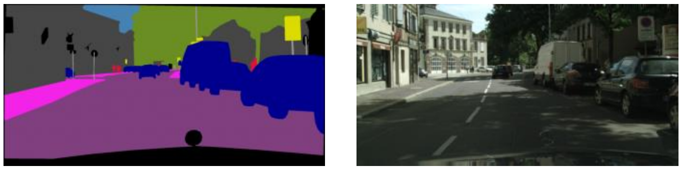

(a) Input (b) Generated image

Fig. 2. Illustration of Semantic Image Synthesis

B. Generative models

Generative models are one of the most important machine learning techniques which aim to generate an algorithm to analyze and learn the probability distribution of data. There are a lot of generative models introduced in the past years [11, 35–40] which differ from each other in the way how the probability distribution is computed, or in other words differ in the density function whether it is an explicit density or an implicit density. Also, the generative models use additional information or assumptions such as variational [35], Markov chains [38, 40], or other assumptions. Unlike previous methods, Generative adversarial Networks (GANs) [11] doesn't rely on any assumptions about any kind of information.

C. Variation of GANs

In the context of GAN, there are two neural network models. These two models represent two players in a game competing against each other. One of these models is a generator network that is trained to generate realistic samples, and the other model is a discriminator network that examines generated samples and estimates whether it is real or fake. These two models are in constant battle throughout the training process. GAN is mostly intended to solve the task of generative modeling. The goal eventually is to try to learn a program that can generate new examples of data that resemble the existing distribution of samples.

GAN has shown the capability of generating realistic images [34, 41–47], and have a lot of advances and types in many perspectives such as latent space [12–16], network architecture [23–26], and objective function [17–22], and driven design applications such as style transfer [27, 29, 30], text-to-image translation [31, 32, 48–51], face aging [52, 53], image super resolution [28, 54, 55], and video generation [56]. A toxonomy of above mentioned methods and variations illustrated in Fig 3.

GANs have shown the capability of generating realistic images. Moreover, to generate user-specific images, Conditional GANs (CGANs) [12] have been proposed for generating images of only one particular class. Latent variable is passed to the generator and the discriminator. The generator learns side-information conditional distributions, as it is able to disentangle this from the overall latent space. Similar to CGAN, latent variable in ACGAN [13] is passed to the generator .Discriminator is tasked with jointly learning real-vs-fake and the ability to reconstruct the latent variable being passed in.

In InfoGAN [14] instead of the latent variables being known a priori from a dataset, make parts of latent space randomly drawn from different distributions. Make the discriminator reconstruct these arbitrary elements of latent space that are passed into the generator. Semi-supervised GAN (SGAN) [16] is proposed in the context of semi-supervised learning. Semi-supervised learning is a promising research field between supervised learning and unsupervised learning. Unlike supervised learning, in which we need a label for every sample, and unsupervised learning, in which no labels are provided, semi-supervised learning has labels for a small subset of example. Compared to FCGAN, SGAN's discriminator is multi-headed i.e., it has softmax and sigmoid for classifying the real data and distinguishing real and fake samples respectively. Traditional GANs have no means of learning the inverse mapping. Bidirectional GAN (BiGAN) [15] is designed for this purpose.

Introducing a new network architecture becomes a very important way in order to make a significant change in the context of GAN especially, or in deep learning generally. Deep Convolutional GAN (DCGAN) [23] is the first work that applied a deconvolutional neural networks architecture for the

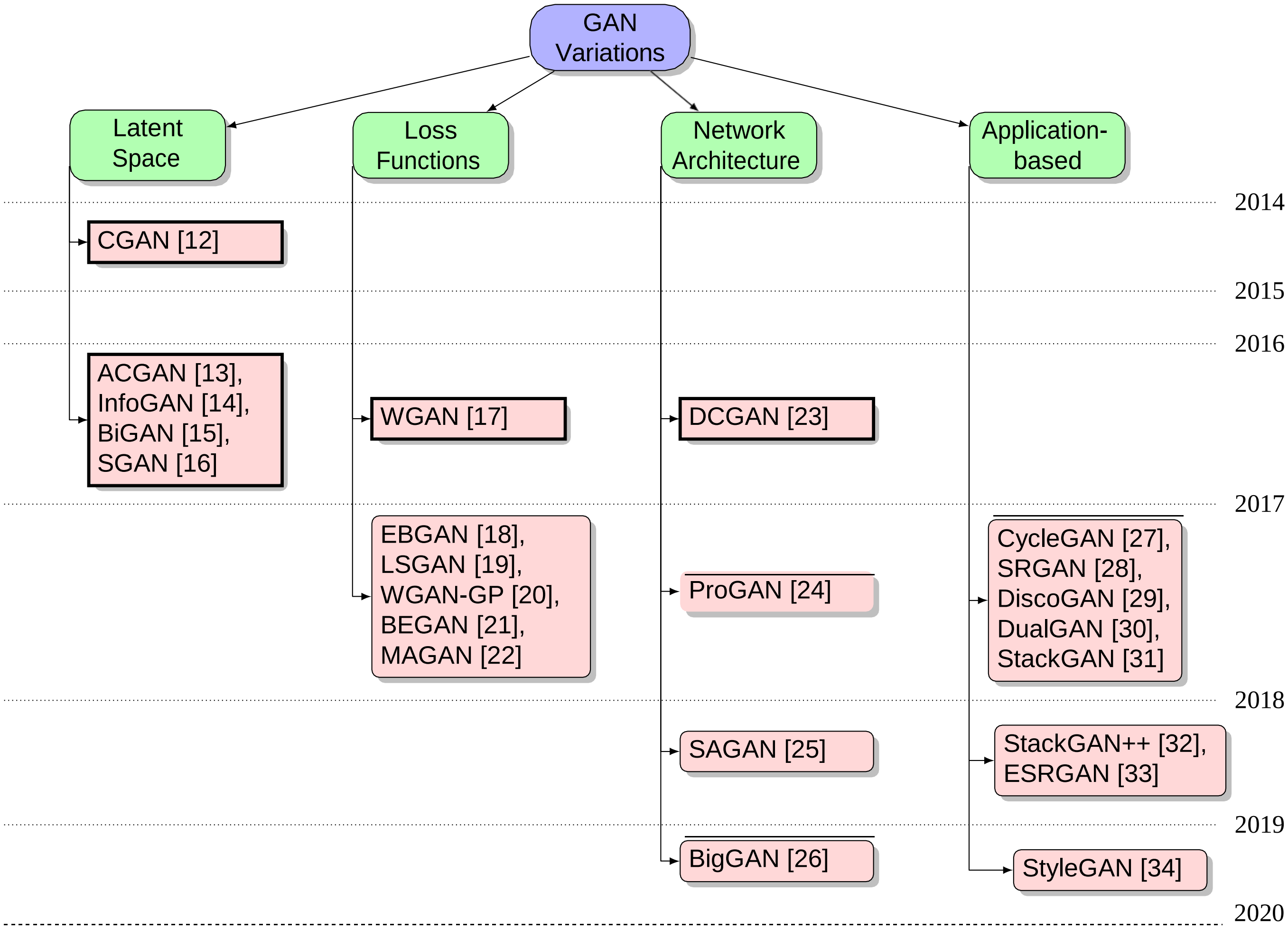

GAN Variations
Latent Space
Loss Functions
Network Architecture
Application-based
2014
2015
2016
2017
2018
2019
2020
CGAN [12]
ACGAN [13], InfoGAN [14], BiGAN [15], SGAN [16]
WGAN [17]
EBGAN [18], LSGAN [19], WGAN-GP [20], BEGAN [21], MAGAN [22]
DCGAN [23]
ProGAN [24]
SAGAN [25]
BigGAN [26]
CycleGAN [27], SRGAN [28], DiscoGAN [29], DualGAN [30], StackGAN [31]
StackGAN++ [32], ESRGAN [33]
StyleGAN [34]

Fig. 3. An overview of the footprint for variations of the most popular GANs.

generator and provides a set of ad-hoc guidelines for building architectures for images, and enabled a lot of current progress in GAN research. Progressive GAN (ProGAN) [24] aims to enhance the image by increasing its resolution and include progressive steps to expand the network architecture. This method uses a progressive growing methodology which begins with 4×4 resolution and ends with 1024×1024 resolution. At every step in the training process, new layers are added to the network architecture by learning from the previously learned features. Lately, a lot of state-of-the-art methods use the progressive growing methodology to increase the resolution of the synthesized image.

Self-attention GAN (SAGAN) [25] uses self-attention mechanism to obtain higher details for objects in images. Benefiting from the self-attention mechanism, SAGAN can capture global spatial information and long-range dependencies to synthesize images. BigGAN [26] have been designed based on SAGAN and use spectral normalization [57]. The authors focused on scalability, stability, and robustness. Also, they scale the batch size and use residual blocks with bottlenecks [58]. After BigGAN has been proposed a lot of researches in this literature uses spectral normalization and residual blocks techniques to enhance the generated samples.

The goal of a generative model is to come up with a way of matching their generated distribution to original data distribution. To Minimize the distance between the two distributions is critical for creating a system that generates content that looks good, new, and like it is from the original data distribution. In order to make that happen, a lot of progress has been made to minimize that distance. Wasserstein GAN (WGAN) [17] uses Wasserstein Metric (Earth Mover's distance) as a cost function. WGAN has been proposed to solve vanishing gradients and mode collapse problems which are common GAN problems in the training phase. But WGAN still suffers from some difficulties such as the complexity in the low-dimensional space and the interactions between the weight constraint and the cost function. For that Improved WGAN (WGAN-GP) [20] have been proposed to solve these problems by reducing the

TABLE I
SUMMARY OF SEMANTIC IMAGE SYNTHESIS RELATED WORKS

Method	**Contributions**	**Pros**	**Cons**
Pix2pix [27] - 2016	(1) Employs a conditional GAN. (2) Propose Image-to-Image translation using paired databases.	(1) Provide a lot of implementation techniques. (2) Implement method with a lot of databases.	(1) Prone to failure for high-resolution image generation tasks.
CRN [28] - 2017	(1) Applies a cascaded refinement network with regression loss. (2) Adopt a modified perceptual loss to synthesize images.	(1) Can generate high-resolution images.	(1) Images are lack fine details and realistic textures.
Pix2pixHD [29] - 2017	(1) Implement a multi-scale generator and multi-scale discriminators. (2) Apply LSGAN [19]. (3) Use instance-level object semantic information.	(1) Generate high-resolution images. (2) Stabilize training compared to previous works. (3) Improve image quality.	(1) Layout information cannot be well preserved in the generator.
SIMS [30] - 2018	(1) Semi-parametric method. (2) Memory bank (3) Using Refinement network.	(1) Produce high-resolution images.	(1) Requires some assumptions about the data distributions. (2) Lack of semantic information.
SPADE [31] - 2019	(1) Uses the label maps. (2) Takes a noise map as input of the generator. (3) Use hinge loss term.	(1) Can predict spatially-adaptive affine transformations. (2) Achieves a multi-model synthesis.	(1) Feature modulation by simple affine transformations is limited in representational power and flexibility.
CC-FPSE [32] - 2019	(1) Employs a weight prediction network for generator with conditional convolutions. (2) Uses feature-pyramid semantic-embedding discriminator.	(1) Solve the problem of representational power and flexibility	(1) The quality of the generated images could be improved.
SEAN [33] - 2019	(1) Introduce Semantic region-adaptive normalization layer.	(1) Improves the quality of the synthesized images. (2)Improves the per-region style encoding. (3) Allows the user to select a different style input image for each semantic region.	(1) Lack of training stability.
EdgeGAN [9] - 2020	(1) Using attention mechanism. (2) contains a multi-modality discriminator. (3)Use parameter-sharing convolutional encoder. (4)Implement edge generator and image generator. (5) Introduce attention guided edge transfer module and the semantic preserving module.	(1) Improve the quality of the generated images. (2) Produce detailed structural information and spatial resolution loss.	(1) Consumes high computational time.
TSIT [4] - 2020	(1) Consists of multi-scale feature normalization (FADE and FAdaIN). (2) Contain two-stream network design (content and style).	(1) Scale to various tasks in both unsupervised and supervised settings.	(1) The loss function is not consistent with all the diversity of the generated image.
DAGAN [8] - 2020	(1) Using attention mechanism. (2) Design two modules Spatial Attention Module (SAM) and Channel Attention Module (CAM)	(1) Improving feature representations. (2) Improve the quality of the generated images. (3) Solve the lack of effective semantic constraints. (4) Concentrate on the structural correlations in both spatial and channel.	(1) Lack of Normalization layer usage. (2) Lack of traning stability.

modeling capacity of the discriminator. Also, it introduces the gradient penalty which represents the weight constraint.

LSGAN [19] has a setup similar to WGAN. However, instead of learning a critic function, LSGAN learns a loss function. The loss for real samples should be lower than the loss for fake samples. This allows the LSGAN to put a high focus on fake samples that have a really high margin. Instead of using a discriminator like how the original GAN does. Energy Based GAN (EBGAN) [18] uses an autoencoder to estimate reconstruction loss by applying these steps: (1) train an autoencoder on the original data, (2) now run generated images through this autoencoder, and (3) poorly generated images will have awful reconstruction loss, and thus this now becomes a good measure. Boundary Equilibrium GAN (BEGAN) [21] is an iteration on EBGAN. It instead uses the autoencoder reconstruction loss similar to WGAN's loss function. Margin Adaptation GAN (MAGAN) [22] is another variation of EBGAN. EBGAN has a margin as a part of its loss function to produce a hinge loss. What MAGAN does is reduce that margin monotonically over time, instead of keeping it constant. The result of this is that the discriminator will autoencode real samples better.

A lot of application-driven design researches have been made based on GAN to solve various types of problems or tasks. CycleGAN [27] aims to solve a problem called image-to-image translation using unpaired images. The image-to-image translation is the task of translating one image domain to another. There are many tasks underlying image-to-image translation like arbitrary style transfer, semantic image synthesis, and multi-model image synthesis. CycleGAN consists of two mapping functions and two discriminators, and two loss functions (adversarial loss and cycle-consistency loss).

DiscoGAN [29] and DualGAN [30] are also proposed to tackle some image-to-image translation problems. StyleGAN [34] uses ProGAN as a baseline, and it uses a lot of techniques such as (1) Adaptive Instance Normalization (AdaIn), (2) A constant learned input (noise input), and (3) Bilinear up/downsampling operations with tuned hyperparameters.

StackGAN [31, 32] proposed for test-to-image translation task, also StackGAN consists of two stages with two discriminators and two generators. The first stage represents a low-resolution and the second stage for a high-resolution image. SRGAN [28] for super-resolution images and it is the first research which has been effective in achieving single image super-resolution. Enhanced SRGAN (ESRGAN) [33] is proposed to solve the problems of SRGAN and improve the details of the synthesized image.

D. Semantic Image Synthesis with GAN

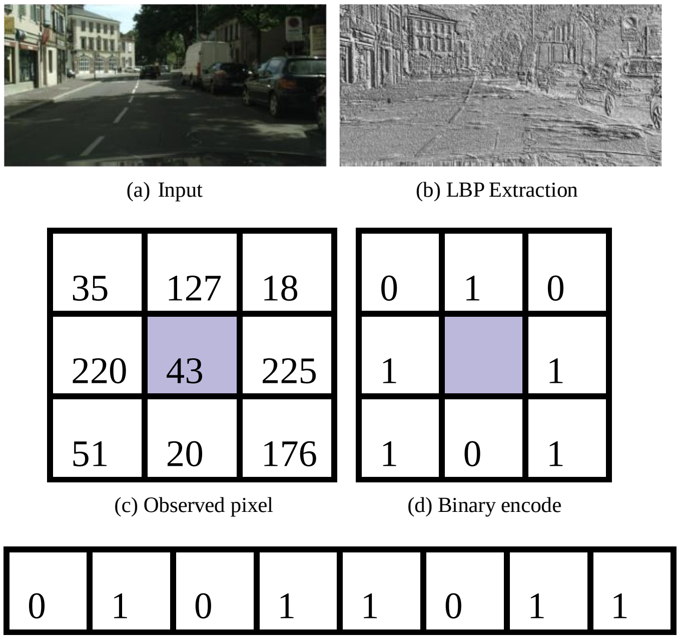
Recently a lot of researches used GAN to overcome the semantic image synthesis task which is to generate a photo-realistic image given the semantic segmentation map. Details are given in Table I. For instance, Isola et al. propose Pix2pix [1] employs a conditional GAN to learn to translate from one image domain to another, such as semantic labels to Street scene, semantic labels to Facade, and day to night. To synthesize high-resolution images with fine details and realistic features, Pix2pixHD [3] have been proposed. Pix2pixHD consists of a multi-scale discriminator and a coarse-to-fine generator. Also, Pix2pixHD applies LSGAN [35] to stabilize the training, and uses instance-level object semantic information.

Park et al. propose SPADE [6] which takes a noise map as input to the generator which leads to multi-model synthesis and produces spatially-adaptive normalization layers and uses hinge loss term. CC-FPSE [5] employs a weight prediction network for generator with conditional convolutions, it uses feature-pyramid semantic-embedding discriminator. L. Jiang et al. propose TSIT [4] which consists of multi-scale feature normalization (FADE and FAdaIN), and two-stream network design (content and style). Hao Tang et al. propose both EdgeGAN [9] and DAGAN [8] improve details of the synthesized images using the attention mechanisms. EdgeGAN contains a multi-modality discriminator and the generator consists of attention guided edge transfer module and edge generator, and image generator. DAGAN design two modules Spatial Attention Module (SAM) and Channel Attention Module (CAM).

In addition to GAN-based methods, Chen and Koltun [2] suggest that conditional GAN is unstable and has optimization issues for generating high-resolution images. Instead, they apply a cascaded refinement network and use objective function based on a perceptual loss. SIMS [7] is semi-parametric not a GAN-based method, and uses a memory bank and a refinement network.

III. LOCAL BINARY PATTERN (LBP)

Local Binary Pattern (LBP) [59] is one of the most common techniques for texture feature extraction and it can be helpful for semantic image synthesis task by preserving the spatial features. LBP is an operator used for describing the spatial features and texture of a rectangular block. Additionally, LBP is a simple but effective texture operator, which involves thresholding the neighborhood of each pixel using the window mean, window medium, or the actual value of the pixel, as thresholds, then extracts an LBP code which represents the spatial feature for that pixel. Fig 4 illustrates these steps which contains: (a) input RGB image is converted to Gray scale, (c) the observed pixel and pixel value of grayscale image, (d) binary encode for the observed pixel and its neighbours, (e) LBP code which represents spatial feature for the observed pixel, and finally (b) the LBP extraction of the input image.



(a) Input

(b) LBP Extraction

35	127	18
220	43	225
51	20	176

(c) Observed pixel

0	1	0
1		1
1	0	1

(d) Binary encode

0	1	0	1	1	0	1	1

(e) LBP code

Fig. 4. Illustration of LBP extraction

LBP is widely used in image processing as it is easy and simple to be implemented, and it has been used for a various applications such as face recognition [60, 61], image classification [62–64], image reconstruction [65], and texture classification [66, 67]. Recently researches discussing the power of LBP, simplicity, and how it provides an accurate result. Dhingra et al. [68] discuss the five most prominent texture feature extraction techniques used in CBIR systems including LBP. And Wei et al. [64] talks about how can LBP lessen the workload of CNNs and improve the classification accuracy.

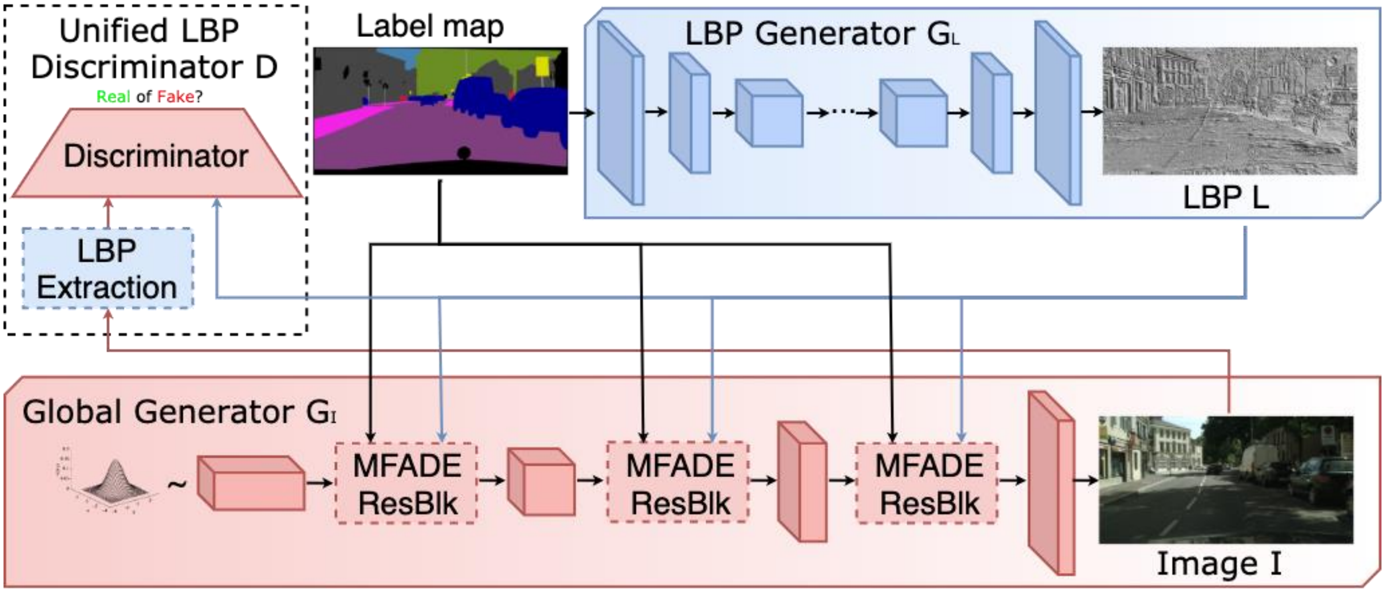


Fig. 5. Illustration of multiple methods to solve open issues

IV. Normalization Layers

Normalization Layers have been recently investigated in many tasks that aim to solve real-world problems, so a lot of modern deep networks use normalization layers or introduce a new normalization to produce fine details for samples such as Batch Normalization [69], Local Response Normalization [70], Instance Normalization [71], Layer Normalization [72], Adaptive Instance Normalization (AdaIn) [34]. Normalization layers are also widely used in Semantic Image Synthesis methods such as, SPADE [6] which propose a spatially-adaptive normalization layer, SEAN [10] takes SPADE as a start point and improve it by adding per-region style encoding. TSIT has been inspired by both SPADE and StyleGAN [34] which introduces a spatially-adaptive normalization layer and adaptive instance normalization layer, respectively. Moreover, most of the proposed semantic image synthesis techniques use both conventional batch normalization layers and instance normalization layers to improve the efficiency of the generated images.

V. Discussion

There is a huge gap between the quality of photo-realistic images and the quality of synthesized images because of the lack of structured information and spatial features. To address these limitations, we can use the Local Binary Pattern (LBP) Generator as illustrated in Fig 5. The semantic label map is passed to LBP Generator as an input, then the LBP generator outputs the estimated LBP feature map L. Meanwhile, the global generator is designed to synthesize photo-realistic images from the input labels. So, we use both semantic label map and generated LBP feature map L to guide the process of generating the photorealistic image by the global generator.

Besides, in Fig 5, the global generator is a deep network designed to output the photo-realistic image. since the network needs to simultaneously learn appearance and structure information from the input labels. So, we should benefit from the power of normalization layers to prevent semantic information from overloading.

As illustrated in Fig 5, we can develop a discriminator to simultaneously distinguish outputs from two generators (LBP generator and the global generator) to facilitate the training process, which is capable of solving the diversity of generated image and computational time problems, and It opens a way to use a consistent loss function to further tackle the challenges of training stability. Also, it transforms the generated photorealistic output from the global generator to LBP features by an LBP extraction module.

VI. Conclusion

Despite the significant successes achieved to date, applying GAN to the semantic image synthesis task still poses significant challenges such as the diversity of image generation, computational time, lack of semantic information, and training stability. To tackle these problems we need new models to be accurate, stable, and flexible to work on a wide range of datasets and applications. In addition, implement a consistent objective function to the new models to facilities the training phase. Moreover, diving into normalization layers can be very promising like the previous state-of-the-art methods which introduce a new normalization layer.

References

[1] 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.

[2] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1511–1520.

[3] 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, pp. 8798–8807.

[4] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, “Tsit: A simple and versatile framework for image-to-image translation,” *arXiv preprint arXiv:2007.12072*, 2020.

[5] X. Liu, G. Yin, J. Shao, X. Wang *et al.*, “Learning to predict layout- to-image conditional convolutions for semantic image synthesis,” in *Advances in Neural Information Processing Systems*, 2019, pp. 570–580.

[6] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[7] X. Qi, Q. Chen, J. Jia, and V. Koltun, “Semi-parametric image synthesis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8808–8816.

[8] H. Tang, S. Bai, and N. Sebe, “Dual attention gans for semantic image synthesis,” *arXiv preprint arXiv:2008.13024*, 2020.

[9] H. Tang, X. Qi, D. Xu, P. H. Torr, and N. Sebe, “Edge guided gans with semantic preserving for semantic image synthesis,” *arXiv preprint arXiv:2003.13898*, 2020.

[10] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, “Sean: Image synthesis with semantic region-adaptive normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5104–5113.

[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*, 2014, pp. 2672–2680.

[12] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[13] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in *International conference on machine learning*, 2017, pp. 2642–2651.

[14] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in neural information processing systems*, 2016, pp. 2172–2180.

[15] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” *arXiv preprint arXiv:1605.09782*, 2016.

[16] N. Souly, C. Spampinato, and M. Shah, “Semi supervised semantic segmentation using generative adversarial network,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5688–5696.

[17] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *arXiv preprint arXiv:1701.07875*, 2017.

[18] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” *arXiv preprint arXiv:1609.03126*, 2016.

[19] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[21] D. Berthelot, T. Schumm, and L. Metz, “Began: Boundary equilibrium generative adversarial networks,” *arXiv preprint arXiv:1703.10717*, 2017.

[22] R. Wang, A. Cully, H. J. Chang, and Y. Demiris, “Magan: Margin adaptation for generative adversarial networks,” *arXiv preprint arXiv:1704.03817*, 2017.

[23] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *arXiv preprint arXiv:1710.10196*, 2017.

[25] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 7354–7363.

[26] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[28] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[29] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” *arXiv preprint arXiv:1703.05192*, 2017.

[30] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[31] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[32] ——, “Stackgan++: Realistic image synthesis with stacked generative adversarial networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.

[33] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, “Esrgan: Enhanced super-resolution generative adversarial networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[34] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[35] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[36] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in neural information processing systems*, 2014, pp. 3581–3589.

[37] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” in *Advances in neural information processing systems*, 2016, pp. 4790–4798.

[38] R. Salakhutdinov and G. Hinton, “Deep boltzmann machines,” in *Artificial intelligence and statistics*, 2009, pp. 448–455.

[39] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[40] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski, “Deep generative stochastic networks trainable by backprop,” in *International Conference on Machine Learning*, 2014, pp. 226–234.

[41] F. Shama, R. Mechrez, A. Shoshan, and L. Zelnik-Manor, “Adversarial feedback loop,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3205–3214.

[42] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, “Few-shot unsupervised image-to-image translation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 551–10 560.

[43] T. R. Shaham, T. Dekel, and T. Michaeli, “Singan: Learning a generative model from a single natural image,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4570–4580.

[44] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197.

[45] X. Gong, S. Chang, Y. Jiang, and Z. Wang, “Autogan: Neural architecture search for generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3224–3234.

[46] A. Jahanian, L. Chai, and P. Isola, “On the”steerability” of generative adversarial networks,” *arXiv preprint arXiv:1907.07171*, 2019.

[47] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “Cnn-generated images are surprisingly easy to spot... for now,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 7, 2020.

[48] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[49] A. Dash, J. C. B. Gamboa, S. Ahmed, M. Liwicki, and M. Z. Afzal, "Tac-gan-text conditioned auxiliary classifier generative adversarial network," *arXiv preprint arXiv:1703.06412*, 2017.

[50] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5802–5810.

[51] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1316–1324.

[52] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 2089–2093.

[53] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5810–5818.

[54] H. Bin, C. Weihai, W. Xingming, and L. Chun-Liang, "High-quality face image sr using conditional generative adversarial networks," *arXiv preprint arXiv:1707.00737*, 2017.

[55] S. Vasu, N. Thekke Madam, and A. Rajagopalan, "Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[56] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in neural information processing systems*, 2016, pp. 613–621.

[57] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[59] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[60] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European conference on computer vision*. Springer, 2004, pp. 469–481.

[61] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[62] S. Jia, B. Deng, J. Zhu, X. Jia, and Q. Li, "Local binary pattern-based hyperspectral image classification with superpixel guidance," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 749–759, 2017.

[63] S. Jia, J. Hu, J. Zhu, X. Jia, and Q. Li, "Three-dimensional local binary patterns for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 4, pp. 2399–2413, 2017.

[64] X. Wei, X. Yu, B. Liu, and L. Zhi, "Convolutional neural networks and local binary patterns for hyperspectral image classification," *European Journal of Remote Sensing*, vol. 52, no. 1, pp. 448–462, 2019.

[65] H. Wu and J. Zhou, "Privacy leakage of sift features via deep generative model based image reconstruction," *arXiv preprint arXiv:2009.01030*, 2020.

[66] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[67] Z. Guo, X. Wang, J. Zhou, and J. You, "Robust texture image representation by scale selective local binary patterns," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 687–699, 2015.

[68] S. Dhingra and P. Bansal, "Experimental analogy of different texture feature extraction techniques in image retrieval systems," *Multimedia Tools and Applications*, vol. 79, no. 37, pp. 27 391–27 406, 2020.

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[71] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[72] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.