BachGAN: High-Resolution Image Synthesis From Salient Object Layout

Ayush Kumar - 2017B5A70761P
Saurabh Bhansali - 2017A3PS0377P
Vinay U Pai - 2017A3PS0131P

September 30, 2020

Abstract

In this paper, we explore a fundamental frontier of computer vision research, that is, image generation. More specifically, our aim is to generate high resolution images by taking the object layout of the desired image as an input. This greatly simplifies the input compared to previous image generating models, and also achieves better quality at the same time. To do this, we use a Background Hallucination Generative Adversarial Network (BachGAN) that selects a pool of backgrounds that fits the input object layout, hallucinates/fuses all the backgrounds into one, and then generates the image. We further propose some improvements to this model, including pushing the boundaries of the image resolution further, considering an alternate and better loss function, and to disentangle the foreground and background objects representations to give us more control and improve the generated images' quality.

1 Introduction

Generating realistic images by specifying the details of the image is a cornerstone in computer vision research. Previous methods of generating images from text or data structures such as scene graphs have been widely studied. The limitations of these methods can be overcome by using object layouts, which consists of bounding boxes of objects and their labels. These are easy to use and give great flexibility in specifying what we want in the image.

The Generative Adversarial Network (GAN) [1] was introduced as a framework for generative modelling as an unsupervised task corresponding to a two player mini-max game. GAN uses two neural networks - a generative model G to generate a new example in the given domain which is pitted against an adversary discriminative model D that tries to classify examples as real (from the domain) or fake (generated). Thus D provides guidance to G about the results to be generated. Competition between these two improves the model until the generated fake results can't be distinguished from the real domain data.

Since its proposal in 2014, GANs have been used exhaustively for image synthesis. Image synthesis or generating new images from an existing dataset can be broadly classified into two parts:

- *Unconditional Generation*: It refers to generation of samples similar to that of dataset without any imposed condition [2].
- *Conditional generation*: It refers to generation of samples corresponding to a particular dataset based on some conditions.

Conditional image generation has been on a rise with image synthesis models working on varied form of conditional inputs such as generation of image based on the given text [3] which saw some exciting progress. However due to the inability of text based image generator to generate images depicting many objects of a complex sentence, scene graph was introduced [4] which enables explicit reasoning between objects and their relationship using graph convolution. Another research field has been image generation based on semantic segmentation. [5] has generated high resolution photo-realistic images based on semantic label maps using conditional GANs. Though previous work in this field has shown favourable results but collecting the data for semantic segmentation map can be time-consuming.

So to overcome the challenges posed by the previously discussed methods, this paper proposed Background Hallucination Generative Adversarial Network (BachGAN). In [6], the authors train BachGAN to construct a high resolution realistic image from a salient object layout consisting of foreground objects' bounding boxes and labels. g

2 Related work

High Resolution Image Generation Adversarial learning techniques are commonly used for transforming images from one domain to another based on training data (in the form of input-output image pairs). In [7], the authors conclude that instability in training and issues of optimization may make it difficult for conditional GANs to produce high resolution images. They overcome this problem, by using a novel perceptual loss and generate high resolution images, albeit ones that are poor in quality in terms of finer details and textures.

Further, Wang et al. [5] solve the issues of training stability by using novel objective functions and multi-scale generators and discriminators. They achieve better results than Chen and Koltun [7], and also add additional features of user interaction. This allows the users to add and remove objects from the semantic layout to generate different images.

Semantic Image Synthesis with Spatially-Adaptive Normalization Some methods have already been developed to take semantic layout as input for the model and then pass it through the layers of convolutions, non-linearity layers and normalization layers for the synthesis of photo-realistic images[5]. However, the issue with them is that the normalisation layer tends to wash-off the data contained in the segmentation map which is fed as input. Thus these methods don't deliver the desired output.

So to overcome this issue, [8] proposes *spatially adaptive normalization* (SPADE). This introduced normalization modulates the activation using spatially adaptive, learned transformations so that it doesn't fade off while traversing through the layers of the network. Similar to BatchNorm [9], SPADE also modulates the activation in a channel-wise manner. However the key difference is that the

modulation parameter *gamma* and *beta* depend on input and vary with respect to the location. SPADE is a generalization to several existing normalization layers and since the parameters adapt to the input mask, it preserves the semantic information in the layers better than InstanceNorm [10] which has been used in most of the state-of-the-art models for conditional image synthesis.

To measure the performance image generated through the model is run through a semantic segmentation model to generate a segmentation mask which is compared to the original one. The accuracy of the model is evaluated using mean Intersection-over-union(mIoU), pixel accuracy(accu) and Frechet Inception Distance (FID). SPADE was able to outperform the leading methods in semantic segmentation, namely CRN[7], SIMS[11] and pix2pixHD[5] on each dataset: COCO-Stuff, ADE20K, ADE20K-outdoor and Cityscapes on all the performance parameters.

In both [5] and [8], the images are generated from semantic segmentation layouts or maps. However interpreting and creating these segmentation maps are often difficult tasks in themselves. Hence an easier input in the form of object layouts is used in [12], [6].

References	**Input**	**Technique**	**Datasets used**
[3]	Text	Attentional GAN	COCO, CUB
[4]	Scene Graphs	Cascaded Refinement Network	COCO-Stuff, Visual Genome
[8]	Semantic segmentation masks/maps	SPADE	Cityscapes, COCO-Stuff, ADE20K
[5]	Semantic segmentation maps	Conditional GANs (with novel adversarial loss)	Cityscapes, NYU Indoor RGBD, Helen Face, ADE20K
[12]	Holistic objects layout	SN-GAN with batch normalization and LSTM at fuser	COCO-Stuff, Visual Genome
[6]	Salient objects layout	BachGAN	Cityscapes, ADE20K

Table 1: Comparison of different image generation techniques

Layout Based Image Generation The concept of object layouts has always been an intermediate step in scene graphs/text to image generation systems. In [12], Zhao et al. have proposed a system that generates images from holistic object layouts. These object layouts specify the bounding box and categories of both foreground and background objects. The model is able to generate multiple different images, which all fit the layout. They achieve this by disentangling the representations of objects into two: categories (labels) and appearances. The generators minimize the sum of 6 different losses, and the authors have also performed ablation studies to find out the impact of each of these.

The layouts in [12] need to specify details for both background and foreground objects. Moreover, the generated images are of low resolution. Hence in [6], the authors propose a BachGAN that can generate high resolution images

from layouts consisting of only foreground objects.

Image Generation with BachGANs BachGAN [6] aims to generate high quality images from salient object layout(i.e. information about the foreground only in terms of bounding boxes and corresponding labels). The two main problems tackled are a) how to do high resolution image synthesis without the segmentation map; b)how to create background without any inputs and merge it with the foreground objects.

To address these problems BachGAN consists of three major modules:

1. *Background Retrieval Module* – This module helps to retrieve a segmentation map given a particular object layout. This is based on assumption that objects which share similar foreground objects can possess similar background. Now, given an input object layout L and a memory bank B containing Images I and their respective segmentation map S, we select the pair of image I and map S which has a layout similar to that of input L. This layout is chosen on the basis of layout-similarity score, a variant of Intersection-over-Union(IOU) metric. This way we can hallucinate a background for a given input layout.

2. *Background Fusion Module* – Since a particular segmentation map can't ensure to include all the objects in the given input layout, we use this module to choose Top-m segmentation map and encode it to include all the objects in the layout and synthesize a smoother background. In this module, m segmentation maps with corresponding background label maps. These maps are then concatenated with query label maps of salient object layout. A convolutional network is then used to label maps to feature maps. The final feature map obtained from this module contains information about both salient object layout and its corresponding hallucinated background.

3. *Image Generator* – The generator takes final feature map from Background Fusion Module as the input and produces high-quality photo-realistic images. This module uses spatially-adaptive normalization (SPADE)[8] layer instead of Instance normalization[10] which is used by most state-of-the-art conditional image synthesis model.

   Reliability of the proposed model has been established by experiments and comparison with state-of-the-art approaches.

Two data sets were used by the authors for image generation and for comparing six models:

1. *Cityscape* – this data set contains street scenes of over 50 cities with images taken on the various times of the year it also contains some generated images of segmentation models.

2. *ADE20K* – this data set contain around 20000 images of both indoor and outdoor scenes.

Pixel Accuracy and Frechet Inception Distance (FID) were used to measure the performance and compare synthesized and real data for a certain model.

Pixel accuracy measures how much data has been classified correctly. However, it is not a reliable parameter because high accuracy does not imply that the model will work fine with test cases that are negative. Hence FID, which compares statistics of generated and real images, is also used. The BachGAN performs better compared to all the other 6 models including ones presented in [8] and [12].

Table 1 compares the different image generation techniques explained so far.

3 Implementation

The code for the implementation was acquired [13], as instructed by Li et al. in their publication [6]. The authors trained the model on Nvidia DGX1 with 8 V100 GPUs. Since the model is based on Pix2Pix, it loads the whole model in the memory. Thus, it is computationally expensive to generate results even with the pre-trained models. Google Colab was a must, as it offers us to run these codes on the virtual GPU. PyTorch and torchvision modules were used for computer vision tasks.

3.1 Setup

All the code was downloaded from the GitHub repository [13], and then uploaded to Google Colab. The pre-trained model weights for the ADE20K dataset were downloaded from the OneDrive link provided in the GitHub repository. These were then uploaded to Google Colab too. To run the code in the instructed manner, the hierarchies of the directories had to be managed and minor changes were made. After the setup and several iterations of debugging, the results were generated.

3.2 Challenges Faced

1. Setup: A lot of challenges were faced in understanding the hierarchies of the directories, and setting it up so that the code would work. We overcame this by careful management of the directories, and slight changes to the code. The directories were stored on Google Drive and then mounted to the Google Colab environment.

2. Limited resources on Google Colab: Even on Google Colab, issues of computational resources were faced. Firstly, an older Google Colab Notebook was used as this gives us more RAM size. Also with the original code, with a batch size of 14, the RAM provided by Google Colab was getting exceeded. Hence after a trial and error process, results were generated by using a batch size of 2. The code still took around 10 minutes to run.

3. Google Colab Sessions: Since Colab times out periodically, we had to mount the directories using Google Colab and make sure we were able to run the code before the time out.

4. Size of files: The Cityscape dataset was 18 GB, and hence too much for us to download and then upload to the Colab. We hence went for the ADE20K dataset, which is much smaller in size.

4 Innovation

Improving the resolution further We found that though the images are of high resolution compared to earlier state-of-the-art image generation models, compared to the ground truths, the images were still of a lower quality. We therefore identified this as an area that can be improved upon, to generate actual high resolution images. For this we took inspiration from architectures for super resolution CNNs (SRCNNs) in [14].

We have used Image Super Resolution based on [15] to further upscale and improve the quality of images by using residual blocks to extract features via DCNNs. This paper proposed a novel idea of using residual dense block. This model was already trained on DIV2K dataset [16]. This addition to the already existing model helped us achieve a high quality image with a resolution of 512x512. The results can be seen Figure 3.

Disentanglement of foreground from background Another area of improvement that we thought of (and was proposed by the authors in the original paper), was to disentangle the representations of foreground and background objects. This essentially means that we build the architecture of the model in such a way that different parts of the model control the foreground and background object, giving us more control over the objects and allowing us to change the foreground objects given a background or vice versa. This in turn, helps in constructing more realistic images as we can seamlessly integrate foreground objects into the generated pictures.

The current model inputs the hallucinated background generated through the background fusion module into the generator. This might create some challenge on more complex problem such as using the model on face or animated character due to different image factor such as foreground, background and pose. Thus instead of passing just the background information, the fusion module needs to pass learned representation of of all three entities separately. For this we need to incorporate a new architecture in the background fusion module similar to [17] so that it can disentangle the information and pass the encoded embedding features. This proposed network can help learn the different mapping functions and thus help us manipulate the foreground, background and pose during the generation and thus gives us more control over the process of generation.

In Figure 1 we have proposed a novel architecture, which gets the learned representation of foreground and background from retrieval module. They are then passed into separate fusion module. More complexity can later be introduced. The representation are fed as an input to the image generator which can now produce the result with more control. A distinct loss function should be used as disentangled loss to handle the difference.

Using alternate loss function Upon doing substantial research into how GANs in general can be improved, we came upon the conclusion that an alternate loss function could be used. In the current model, the authors use a minimax loss function. This means that the generator generates an output

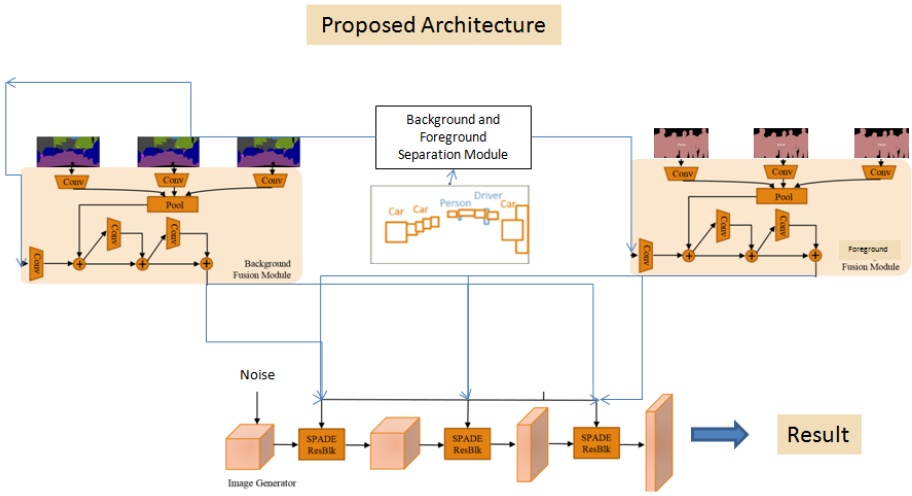

Figure 1: Proposed architecture with disentangled foreground and background representations

image, and the discriminator classifies images as real or fake (i.e. generated by the generator). However this is prone to getting stuck and facing problems with vanishing gradients.

An improvement would be to use the Wasserstein loss function instead. In this, the discriminator is more of a 'critic' and outputs a real number (larger values for real images, and small value for fake ones). The intuition is that this uses the earth-mover's distance metric between real and generated distributions compared to the cross-entropy that is considered in minimax loss function.

Generating animated characters One potential application we identified was to generated animated characters from input object layouts or sketches. Once we achieve the objective of disentangling the foreground and background representations, then we can use the BachGAN as a tool for artists to generate animated characters by changing individual features such as nose shape, eyes, etc. given a basic background face and body structure. We can individually define the different feature representation and after training on suitable dataset, we can get more control over the generation of animated characters.

Incorporating text to object layout converter in the existing model Another improvement that we identified is to improve the interface of the model and make it more convenient to use in practical applications. Text-to-object-layout synthesis is not explore to the best of our knowledge. This could improve the ease-of-use for designers, architects and artists who want to generate images and would like to use text to specify the objects and their locations.

5 Results

Figure 2 shows three representative pictures that display the BachGAN in work. The generated images can be seen to have a close resemblance to the input object layout. They appear in the Results directory, and are exactly as expected and specified in [6].

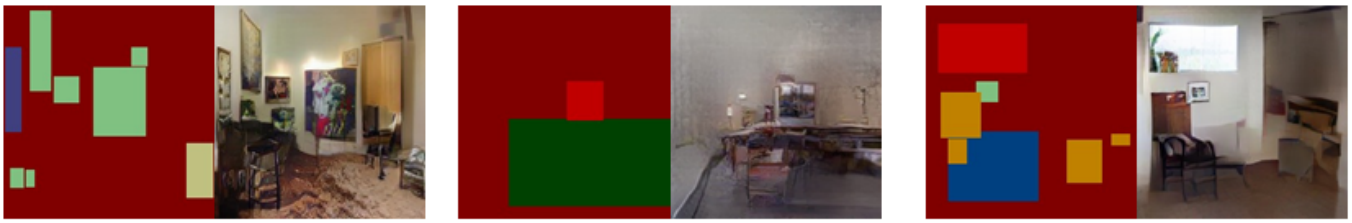

Figure 2: High Resolution Image Generated from the Object Layout

5.1 Proposals from innovations

As mentioned in Section 4, we tried different approaches to improve the model. The improvement of the images' resolution can be seen clearly in Figure 3.

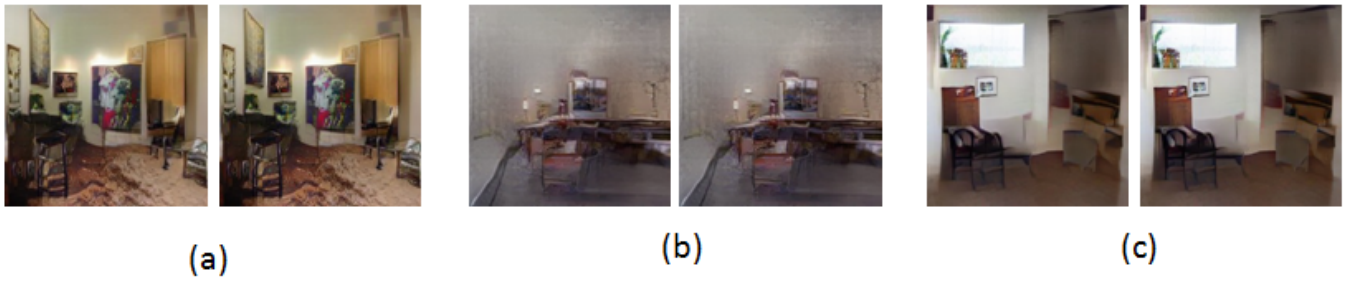

Figure 3: Improvement of generated images' resolution. The left image in each case has been generated by BachGAN and the right image has been processed further to improve quality and resolution

Regarding the alternate loss function, we sent an email to the authors of the original paper [6] to ask whether they faced any issues with minimax function and whether the Wasserstein loss function would fare better. However we did not receive any reply from them, as of now. Upon trying to change it ourselves and seeing the effects, we faced issues in training the model due to computational constraints (the number of GPUs allowed by GCP was not enough for our purposes).

For disentangling the foreground and background objects' representations we were able to do some research and propose an architecture. However implementing this architecture and training the new model will need significant changes to the code and much better computational resources than we have at our disposal right now.

6 Conclusion

In this paper, we studied and implemented a BachGAN for generating realistic high resolution images from object layouts. The generated images can be clearly seen to resemble the object layouts and to the naked eye, cannot be differentiated from real photos. We proposed a few additional improvements to this model. We were able to improve the resolution of the generated images by using a SRCNN. Further, after some research, we concluded that training the model with an alternate loss function may improve the performance. We also propose a text-to-object converter to be added to this model, as this hasn't been done before to the best of our knowledge. Disentangling the foreground and background objects' representation is another interesting approach that we tried to work on. The applications of the model are limited only by the imagination of the user. Animators can use this to build animated characters from mixing and matching objects in a character's appearance. Architects, designers, artists can use this model to make any visualisation task simpler. The work done in this project thus has promising applications in practical scenarios.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Jen-Yu Liu, Yu-Hua Chen, Yin-Cheng Yeh, and Yi-Hsuan Yang. Unconditional audio generation with generative adversarial networks and cycle regularization. *arXiv preprint arXiv:2005.08526*, 2020.

[3] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong 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*, pages 1316–1324, 2018.

[4] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[5] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[6] Y. Li, Y. Cheng, Z. Gan, L. Yu, L. Wang, and J. Liu. Bachgan: High-resolution image synthesis from salient object layout. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8362–8371, 2020.

[7] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1529, 2017.

[8] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[11] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018.

[12] B. Zhao, L. Meng, W. Yin, and L. Sigal. Image generation from layout. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8576–8585, 2019.

[13] BachGAN GitHub Repository. https://github.com/Cold-Winter/BachGAN. [Online; accessed 10-October-2020].

[14] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[15] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[16] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[17] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018.