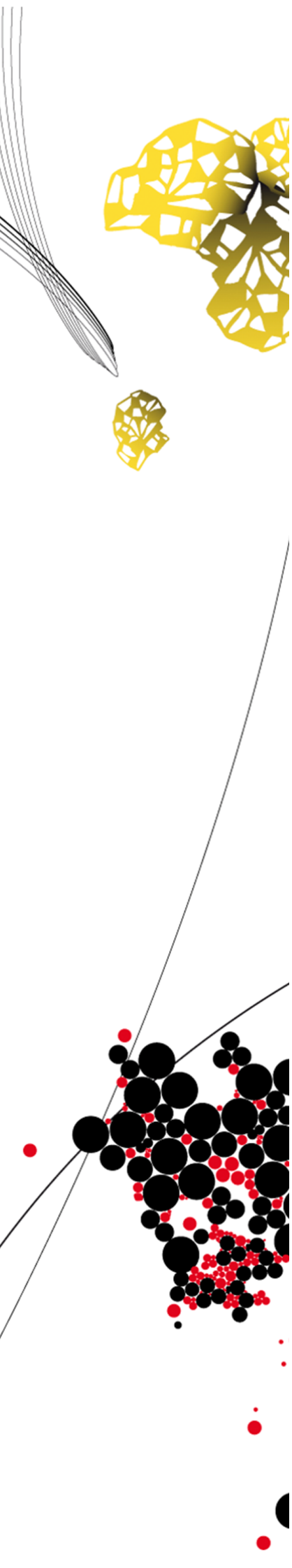

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Compositional Scene Generation using Generative Adversarial Networks

Mehmet Ozgur Turkoglu

M.Sc. Thesis

July 2018

Supervisors:
Prof.dr.ir. R.N.J. Veldhuis (UT)
Dr.ir. L.J. Spreeuwers (UT)
B. Kicanaoglu, MSc (UVA)
W.E. Thong, MSc (UVA)

Services, Cybersecurity and Safety Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Compositional Scene Generation using Generative Adversarial Networks

Mehmet Ozgur Turkoglu
University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
moturkoglu@gmail.com

Abstract

In this master's thesis, a novel sequential image generation model based on Generative Adversarial Networks (GANs) is proposed. Even though recent GAN-based approaches have been successful in generating for example faces, birds, flowers, street view images in a realistic manner, user control over the image is still limited. The proposed approach generates an image element-by-element (object-by-object) progressively and improves the controllability of the image generation process explicitly through an element-specific latent vector. Also, it improves the controllability by resolving affine transformation and occlusion issues existing conditional GANs models have. Experiments are carried out on the subset of the challenging and diverse MS-COCO dataset and the proposed model is compared with the state-of-the-art baselines. Both qualitative and quantitative results are provided to show the strength and the advantages of the proposed model.

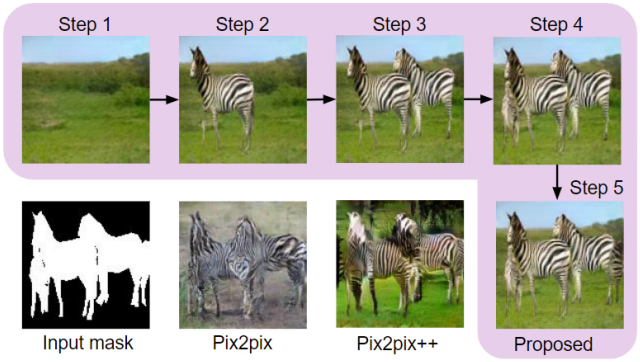

Figure 1: Example images generated by the state-of-the-art models and the proposed sequential model. The images are generated from the given semantic layout map that consists of four zebras. Pix2pix and Pix2pix++ generate a lot of artifacts and struggle to generate a realistic image while the proposed model can generate step-by-step a more realistic image.

1. Introduction

Image generation is an interesting problem in computer vision and machine learning. There has been huge progress in this research area after Generative Adversarial Networks (GANs) are introduced by Goodfellow et al. [1]. Recently, many GAN-based approaches have been proposed to generate photo-realistic natural images. Even though these approaches can generate very realistic images in some domains e.g face [52], birds images [50], generating complex scene images such as street view images is a still difficult problem. Because those images consist of many structural information and constraints but current models have an issue with geometric and structural patterns as pointed out in [48]. Some previous works based on conditional GANs e.g [21], [13], [14] have achieved to synthesize very complex images such as Cityscapes street images [49], or ADE20K scene images [51] by using the semantic layout as a prior knowledge to a model. This kind of supervision also increases the controllability of the image generation as it allows a user to control the scene layout. However, these existing models still have a fundamental limitation on control over the generated scene. Controllability over the elements of a scene e.g a specific object is quite hard since a single entangled latent vector is used for the entire scene and altering this vector usually makes a change in the entire scene. So, for instance, it is very hard to specify or edit a specific object color in the scene while keeping the rest of the image the same. The affine transformation issue is another drawback of the current conditional GAN models. Altering an object's position (or scale, rotation) in the image usually causes changes in other parts in the image. This feature would not be desirable for some applications such as an interior design application. For instance, when the user designs a bedroom, he/she may want to change the armchair position in the image while keeping the rest of the scene exactly the same; however, these conventional models may not allow the user to do this. In addition to those limitations, the occlusion is a problem because of the compact semantic map representation. When the object is occluded by another object or even sometimes when they are close each other, these models produce artifacts for instance, in some cases, they ignore the occluded object, merge them into a single object or produce meaningless objects. (See Figure 1.).

For many automatic image generation applications especially for the interactive applications, these kinds of element/object-level control mechanism would be very beneficial. The purpose of this master's thesis is to study element/object-level control and improve the controllability of the image generation process using GANs. The main research question for this work is twofold: (1) Can we generate an image in a realistic manner with a disentangled control over each foreground object and the background of the scene? (2) Can we generate higher quality and more diverse images compared to state-of-the-art GAN-based methods?

In order to realize a model that has a separate control mechanism over the elements of an image, a sequential image generation model is proposed. The proposed model adopts the layered structure modeling for images that is described in Section 2.7 and generates an image step-by-step starting with the background of the scene and forms the scene progressively by putting a single foreground object at each step. At each time step, a user has a control over an element through a specific latent vector. The generation process is similar to landscape painting where the painter usually first draws the general structure of the scene e.g background and then draws the smaller details e.g foreground objects one-by-one. So this kind of sequential scheme in an incremental fashion is a quite natural way of generating an image. The other motivation for generating an image step-by-step is to break down the generation problem into a sequence of easier problems and therefore to have a smaller task to deal with at each step.

The contributions of this work are the following:

- A new sequential image generation model is proposed which allows a user element/object-level control during the generation process.
- The proposed approach resolves the occlusion artifacts of the existing conditional GAN models.
- Also, it improves image quality and diversity.

This thesis is structured as follows. Preliminaries and closely related works are provided in Section 2 and 3, respectively. In Section 4, the proposed method is described in detail. Experimental setups and both qualitative and quantitative results are given in Section 5 and the limitation of the method is discussed in Section 6. Finally, the conclusion is drawn and possible future research direction is pointed out in Section 7.

2. Background

2.1. Deep Generative Models

Let us assume that the data we have $\mathbf{X} = \{\mathbf{x}^{(i)}\}$, where $i = 1, ..., N$, consisting of N samples comes from some unknown distribution P_{real}. Generative models can be simply considered as any model that takes the data $\mathbf{X}$ (i.e training data) and learns to represent an estimate of that distribution. The result is a probability distribution P_{model}. Some models estimate P_{model} explicitly; for instance through maximum likelihood estimation (MLE). In that case, the objective of the MLE is

$$\arg\max_{\theta \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \log P_\theta(\mathbf{x}^{(i)}) \tag{1}$$

where θ are the model parameters. The other models learn to sample from P_{model} with or without explicitly defining it.

Images are highly complex data; for instance, even very small-sized images; let's say 32-by-32 pixels RGB image lives in 32x32x3 = 3072-dimensional space and contains 32x32x3x8 = 24576-bits information, so it is quite challenging to estimate the probability distribution of this kind of complex data. Fortunately, recent advancements in deep learning techniques make this feasible and recent generative models have achieved great success in image synthesis tasks. These tasks span image generation from scratch, image-to-image translation, colorization, image super-resolution, image completion/inpainting and so on.

Three popular examples of generative models are: Autoregressive models (e.g PixelRNN [27], PixelCNN [28]), Variational Autoencoders (VAEs) [10] and Generative Adveserial Networks (GANs) [1]. PixelRNN/CNN are explicit density models which use chain rule to decompose likelihood of an image $\mathbf{x}$ into a product of 1-d distributions:

$$P_\theta(\mathbf{x}) = \prod_{i=1}^{n} P_\theta(x_i | x_1, x_2, ..., x_{i-1}) \tag{2}$$

where $P(x_i|x_1, x_2, ..., x_{i-1})$ is the probability of i^{th} pixel value given all previous pixels and $P(\mathbf{x})$ is the likelihood of image $\mathbf{x}$ which is computationally tractable. Then, they maximize likelihood of training data (Equation (1)). The advantages of this approach is that it can explicity compute likelihood, $P(\mathbf{x})$ and generate good samples; the disadvantage is that it is slow due to the sequential generation.

In VAEs, it is assumed that data are generated by some random process, involving an unobserved continuous random variable $\mathbf{z}$ which is unknown. The process consists of two steps: (1) a value $\mathbf{z}^{(i)}$ is generated from some prior distribution $P_{\theta^*}(\mathbf{z})$; (2) a value $\mathbf{x}^{(i)}$ is generated from some conditional distribution $P_{\theta^*}(\mathbf{x}|\mathbf{z})$ so the likelihood is:

$$P_\theta(\mathbf{x}) = \int P_\theta(\mathbf{x}|\mathbf{z}) P_\theta(\mathbf{z}) d\mathbf{z} \tag{3}$$

Unfortunately, this likelihood is computationally intractable; therefore, instead of directly maximizing log-likelihood, the lower bound on log-likelihood that is computationally tractable is derived and optimized.

$$\mathcal{L}(\mathbf{x}; \theta) \leq \log P_\theta(\mathbf{x}) \tag{4}$$

The advantages of VAEs are that they have a nice probabilistic formulation and allow inference of $P(\mathbf{z}|\mathbf{x})$ which can be useful representation for other tasks. The main disadvantages of VAEs are the gap between $\mathcal{L}$ and the true likelihood can result in P_{model} learning something other than the true data distribution, P_{real} [11] and secondly, they generate blurrier and lower quality samples compared to GANs.

Unlike previous approaches, GANs learn to sample from real data distribution, P_{real} without dealing with any explicit density function (P_{model} is defined implicitly.). Because there is no easy way to sample from high dimensional, complex real data distribution, they first sample from a simple distribution (random noise e.g multivariate Gaussian) and learn the transformation to real distribution. More details are given in the next section. The advantage of GANs is that they can generate very sharp, high-quality samples. The main disadvantage of GANs is the training instability which will be discussed in Section 2.5.

2.2. Generative Adversarial Networks

Generative adversarial networks are based on a game-theoretic scenario in which the generative model competes against an adversary. The generative model directly produces samples $\mathbf{x} = G(\mathbf{z}; \theta_G)$. Its adversary, the discriminative model, attempts to distinguish between samples drawn from the training dataset (real samples) and samples produced by the generative model (fake samples). The discriminator produces a probability for a given sample, $D(\mathbf{x}; \theta_D)$ which represents the likeliness of being real.

In the original article, the following analogy is given to explain the generative adversarial network framework more intuitively. The generative model is a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles.

The learning in generative adversarial networks is formulated as a two-player minimax game with value function $V(G, D)$. Discriminator tries to maximize the value; in contrast, the generator tries to minimize the value.

$$G^* = \arg\min_G \max_D V(G, D) \tag{5}$$

$$V(G, D) = \mathbb{E}_{\mathbf{x} \sim P_{real}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{Z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{6}$$

This drives the discriminator to attempt to learn to correctly classify samples as real or fake. Simultaneously, the generator attempts to fool the classifier into believing its samples are real. At convergence, the generators samples are indistinguishable from real data, and the discriminator's output is $\frac{1}{2}$ everywhere; this means $P_{model} = P_{real}$. This minimax game has a global optimum for $P_{model} = P_{real}$; see the original article [1] for the theoretical proof.

G and D are differentiable functions with respect to their parameters (θ_G and θ_D respectively) and their inputs ($\mathbf{z}$ and $\mathbf{x}$ respectively). They are typically represented by deep neural networks. Generator, G maps noise vector, $\mathbf{z}$ which lives in low dimensional space, $\mathcal{Z}$ (it is usually called latent space and its typical dimension is 100 or 128.) to an image that lives in higher dimensional image space, $\mathcal{X}$ ($G : \mathcal{Z} \to \mathcal{X}$). $\mathbf{z}$ is sampled from a simple prior distribution such as Gaussian or uniform distributions. Discriminator, D takes an image, $\mathbf{x}$ (both real and fake samples) as an input and produces a probability of being real, as an output ($D : \mathcal{X} \to [0, 1]$).

Training Process

The value function of the minimax game, $V(G, D)$ is used as a loss function (called adversarial loss) and parameters of G and D are optimized simultaneously using backpropagation algorithm. The training alternates between following two main steps (m is mini-batch size.).

- Update the discriminator, D by gradient ascend

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} [\log(D(\mathbf{x})) + \log(1 - D(G(\mathbf{z})))]$$

- Update the generator, G by gradient descend

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\mathbf{z})))$$

In practice, the second step is replaced with gradient ascend of $\nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \log(D(G(\mathbf{z})))$. This is because in the early stage of the training D can discriminate real and fake samples easily because they are quite different. In that case, $\log(1 - D(G(\mathbf{z})))$ saturates when $D(G(\mathbf{z}))$ goes to zero so $\log(D(G(\mathbf{z}))$ provides a stronger gradient for the generator.

2.3. Conditional Generative Adversarial Networks

Conditional GANs [3] are an extension to the original model. In this case, both the generator and discriminator are conditioned on some extra information, $\mathbf{y}$. $\mathbf{y}$ could be any kind of auxiliary information, such as class labels, semantic maps or data from other modalities. This conditioning can be done by putting $\mathbf{y}$ as additional inputs to both the generator and discriminator. In this case, value function can be written as follows.

$$\begin{aligned} V(G, D) = {} & \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim P_{real}(\mathbf{x}, \mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] \\ & + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{Z}}(\mathbf{z}), \mathbf{y} \sim P_{\mathbf{Y}}(\mathbf{y})}[\log(1 - D(G(\mathbf{z}, \mathbf{y}), \mathbf{y}))] \end{aligned} \tag{7}$$

2.4. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [47] are a type of artificial neural networks for processing data that has a known grid-like topology (e.g image, time-series data). They have proven to be very effective for many challenging computer vision problems; for instance image classification, object detection, and semantic segmentation. Unlike feedforward neural networks (i.e multilayer perceptrons) there are two important concepts adopted by CNNs: (1) sparse connectivity (interactions) and (2) weight sharing. Sparse connectivity refers to the inputs of any neuron in the i^{th} layer comes from a small subset of neurons in the $(i-1)^{th}$ layer. It reduces memory needs and computations as well as allows CNNs to exploit local correlations in data (e.g edges and blobs in the image). In CNNs rather than learning a separate set of parameters for every location, only one set is learned and applied for every location. This increases learning efficiency by reducing the number of parameters being learned significantly and achieves a translation-invariant capturing of patterns. Backpropagation algorithm can still be used to learn such shared parameters.

Convolution vs Transposed Convolution Operations

Convolution is the main operation in CNNs. A feature map is computed from an image (or the previous feature map) using this operation:

$$F(i,j) = \sum_m \sum_n x(i+m, j+n) K(m,n) \tag{8}$$

where F is the output feature map, x is the input (image or the previous feature map) and K is a kernel (weight matrix). Convolution maps from an input space to a feature space and transposed convolution (a.k.a. fractionally-strided convolution, deconvolution) is another operation that allows us to go the other way around, map from a feature space to an input space. It is quite useful for image generation tasks. We can simply consider that it enlarges the feature map in spatial dimensions by first contaminating the input with zeros (See [39] for better and detailed explanation.). These operations for a 2D image are illustrated in Figure 2.

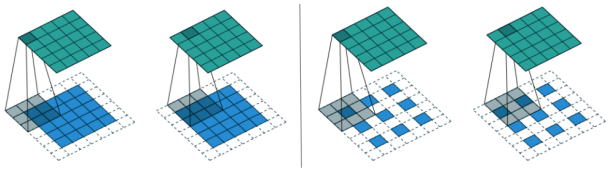

Figure 2: Left images: Convolution operations (for $i, j = 0, 0$ and $i, j = 0, 1$) on 5x5 image (blue) using 3x3 kernel (gray), the output feature map is 5x5 (green). Right images: Deconvolution operations (for $i, j = 0, 0$ and $i, j = 0, 1$) on 3x3 image (blue) using 3x3 kernel (gray), the output feature map is 5x5 (green). White grids in inputs are zeros padded. Courtesy Dumoulin et al. [39].

2.5. Deep Convolutional GANs (DCGAN)

DCGAN that is introduced by Radford et al. [2] makes generating high-quality images using GANs feasible. It is one of the most popular types of GAN used in literature. Convolutional neural networks are used in place of the multi-layer perceptrons and several architectural constraints are introduced. The discriminator has standard CNN architecture and the generator has transposed convolutional layers instead of convolutional layers; therefore, the representation at each layer of the generator is successively larger, as it maps from a low-dimensional noise vector to a high-dimensional image. The architectural constraints are: (1) replacing any pooling layers with strided convolutions for the discriminator and transposed convolutions for the generator, (2) using batch normalization layer in both the generator and the discriminator, (3) removing fully connected hidden layers, (4) using ReLU activation in the generator for all layers except for the output which uses Tanh and LeakyReLU activation in the discriminator for all layers.

2.6. Stability of GAN Training

Training GANs requires finding the Nash equilibrium of a minimax game described in Section 2.2 and it is a more difficult problem than optimizing a pre-defined fixed objective function as pointed out in [11]. Their training is unstable and requires a lot of tricks and choosing suitable hyper-parameters to get reasonable results. The other main issue about GANs is the mode collapse. In this case, the generator only outputs samples from a small set of modes; in extreme case, it could be even a single mode. This could happen because the generator can find a way to fool the discriminator into thinking that it is outputting realistic samples by generating a single sharp sample. So it is a quite active research area to stabilize the GANs' training and improving sample diversity (handling mode collapse). In order to deal with these problems, some works proposed different learning objectives and dynamics such as Wasserstein GAN [7], Least-square GAN [8], Loss-sensitive GAN [9], Optimal Transport GAN [32], Energy-based GAN [33], Unrolled GAN [34]. Some other works such as DCGAN [2] (described in the previous section) carefully design the network architectures. In addition to them, several other useful tricks used in literature (e.g mini-batch discrimination, feature matching, historical averaging) are described in [35].

Spectral Normalization for GAN

Spectral normalization is a new regularization technique recently proposed by Miyato et al. [4]. They argue that spectral normalization makes the discriminator network training more stable so in return, the generator receives a better gradient and learn better and whole training process is more stable. Observations in the early experiments confirmed its effectiveness; therefore, spectral normalization is applied to the discriminator networks. It simply normalizes network weights at each layer by the spectral norm of the weight matrix at that layer (Spectral norm of the matrix is equivalent to the largest singular value of the matrix.). Additional computational cost is small since the power iteration method is used for estimating the largest singular values of the matrices.

Feature Matching Loss

For image generation tasks, it is popular to add a reconstruction loss to an objective function in order to stabilize training and obtain higher quality samples. In this case, the generator has two tasks: (1) fool the discriminator and (2) generate samples similar to training images. Previous works ([37], [38]) showed that reconstruction loss in a feature space (called feature matching loss) is more effective than in a pixel space. The feature matching loss is used in the experiments. Early experiments showed that it is also effective to learn class conditioning. The feature matching loss for the model sample, $\hat{\mathbf{x}}$ is computed as follows.

$$L_{rec} = \sum_l ||\Phi_l(\hat{\mathbf{x}}) - \Phi_l(\mathbf{x})|| \tag{9}$$

where $\mathbf{x}$ is the corresponding training image (ground truth image), Φ_l is the feature extracted from the l^{th} layer of a CNN. VGG-19 network [36] pre-trained on ImageNet dataset [19] is used in the experiments.

2.7. Layered-Structure Modeling for Images

It is natural to model the 2D image of the 3D world in a layered structure in order to deal with a complex scene more easily. This modeling is already used in GAN literature such as in [5], [12] and in this work, the same idea is employed: the background and each foreground object are described in a separate layer. The image, **x** with the foreground object, **f** and the background, **b** can be modelled as:

$$\mathbf{x} = \mathbf{f} \odot \mathbf{m} + \mathbf{b} \odot (1 - \mathbf{m}) \tag{10}$$

where **m** is the foreground mask and $\odot$ is an element-wise multiplication.

3. Related Work

Image generation has been recently very active research area in computer vision community. Some recent approaches are based on auto-regressive models e.g [40], VAE e.g [41] and GAN e.g [2], [5], [29], [13]. GAN-based models are more popular compared to their competitors due to sampling efficiency and high image quality in spite of their difficult training. In GAN literature, in order to control the generated image somehow, various approaches have been proposed conditioned on class label e.g [31], attribute vector e.g [26], text description e.g [30], and semantic layout e.g [21] etc. Reed et al. [42] proposed to learn to control the foreground object position by conditioning on bounding-box and keypoint coordinates. Karacan et al. [21] succeeded at synthesizing realistic outdoor images from input semantic layout map and attributes. Isola et al. [13] proposed conditional adversarial networks as a general-purpose solution to image-to-image translation problems; their model can generate an image from the semantic layout map or the other way around, generating a semantic layout map from the natural image. Hong et al. [22] generates an image from text description. They have separate models for mask and image generations; they first generate semantic layout map from input text description and then generates the natural image from the semantic layout. These approaches can control the scene content up to a certain level but none of them can control different elements of the scene e.g different objects separately. The proposed model is similar to these works, it generates the image from semantic layout map; however, unlike these models, it can control every object in the scene and the background separately.

The basic idea of generating images sequentially is that it breaks down the original problem into a sequence of more manageable stages. In literature, sequential image generation has been studied in different ways. Some models e.g. [14], [29] generate images in course-to-fine fashion. Denton et al. [43] introduced a sequential model that has a series of generative models, each of which captures image structure at a particular scale of a Laplacian pyramid while Zhang et al. [29] improved the image quality by increasing image resolution with a two-stage GAN. Some models such as [5], [6], and [12] generate images part-by-part in order to deal with smaller problems at each generation step and disentangle the noise for different parts of the image which allows us to control the different parts separately. However, there is no supervision in these models; they are not conditioned on the semantic layout map. Therefore, control over each part is limited to a noise for that part. One drawback of these models is that generators do not necessarily learn to draw semantically meaningful part of the image since each sub-model (generator) learns its responsibility itself during training. For instance, for the face images, one generator may learn to draw both eyes and the background and the other one learns to draw the rest of the face. Their qualitative results support this argument. This issue can be resolved by appropriate supervision but even if these models are supervised accordingly, they still have an important limitation because their simultaneous or recurrent learning procedures make the GANs training even more difficult. In these models, the number of the generation steps is pre-defined and limited to 3 in their experiments. So they are not suitable for generating complex images which consist of an arbitrary number of parts.

Park et al. [44] quite recently introduced a new method to learn to generate foreground object image by conditioning on both the text description, the foreground object mask and the given background image while preserving the background image. This might be the most relevant work to the proposed method. But the method is able to generate only a single foreground object when the background image is provided.

The proposed method aims to improve the controllability of image generation by adopting advantages of both conditional GAN models and sequential models.

4. Methodology

4.1. Proposed Model

The proposed model is based on conditional GAN; it generates an image from a semantic layout map. The image is generated step-by-step in an incremental fashion. It can be formulated as following:

$$\mathbf{x} = G(\mathbf{z}_{bg}, \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{M}_1, ..., \mathbf{M}_n) \tag{11}$$

where **x** is the generated image, G is the generator function, $\mathbf{z}_{bg}$ is a noise vector associated with the background of the scene, $\mathbf{z}_i$ is the noise vector associated with i^{th} foreground object in the scene and $\mathbf{M}_i$ is the semantic layout map for the i^{th} foreground object. The proposed sequential model generates an image in n+1 steps where n is the number of foreground objects in the scene:

Step 0:

$$\mathbf{I}_0 = G_{bg}(\mathbf{z}_{bg}) \tag{12}$$

Step 1:

$$\mathbf{I}_1 = G^*(\mathbf{I}_0, \mathbf{z}_1, \mathbf{M}_1) \tag{13}$$

Step i:

$$\mathbf{I}_i = G^*(\mathbf{I}_{i-1}, \mathbf{z}_i, \mathbf{M}_i) \tag{14}$$

Step n:

$$\mathbf{x} = \mathbf{I}_n = G^*(\mathbf{I}_{n-1}, \mathbf{z}_n, \mathbf{M}_n) \tag{15}$$

where $\mathbf{I}_i$ is the image generated at i^{th} step. If we can model G^* as

$$G^*(\mathbf{I}, \mathbf{z}, \mathbf{M}) = \mathbf{I} + G_{fg}(\mathbf{I}, \mathbf{z}, \mathbf{M}) \tag{16}$$

then, equation (11) can be written as

$$\begin{aligned}\mathbf{x} &= G(\mathbf{z}_{bg}, \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{M}_1, ..., \mathbf{M}_n) \\ &= G_{bg}(\mathbf{z}_{bg}) + G_{fg}(\mathbf{I}_0, \mathbf{z}_1, \mathbf{M}_1) + ... + G_{fg}(\mathbf{I}_{n-1}, \mathbf{z}_n, \mathbf{M}_n)\end{aligned} \tag{17}$$

Thus, if we can find such G_{bg} and G^* functions; then, we get a separate noise vector for each object and the background. In the sequential generation process, at each time step, we have an explicit control over the object which is generated at that time step. However, control through the noise is constrained by the image generated so far (previous frame) since G^* is conditioned on the previous frame. This is a desirable feature because generated image should be semantically meaningful; for instance, if the background is dark, foreground object should not be shiny. So G* should be aware of what has been generated so far and put the new foreground object accordingly.

The proposed model consists of two generators: the background generator, G_{bg} and the foreground generator, G^*. Generating process starts with generating the background image, $\mathbf{I}_0$ by G_{bg} which takes a fixed-size noise vector, $\mathbf{z}_{bg}$ as an input and this image is fed to foreground generator, G^*. G^* takes also fixed-size noise ($\mathbf{z}$) vector and semantic layout map ($\mathbf{M}$) as an input. Then, G^* generates $\mathbf{I}_1$ in a way that preserves the background as much as possible while drawing the specified foreground object. This generation process continues until all the objects are drawn. The task of G^* is preserving the previous image and adding the current foreground object. The proposed image generation process is depicted in Figure 3.

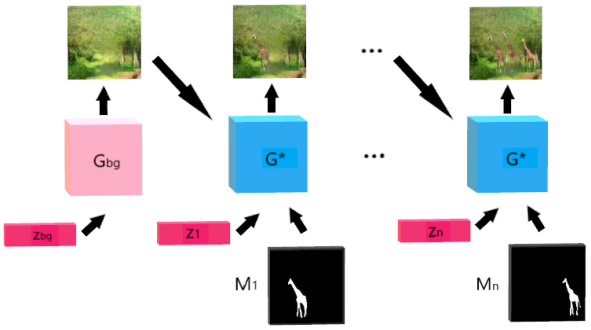

Figure 3: The proposed sequential image generation model. G_{bg}, G^* are the background and foreground generators, respectively.

4.2. Foreground Model Learning

In order to deal with the smaller problem at once, the foreground and the background models are learned separately rather than simultaneously. For convenience, the G^* function is reformulated as following:

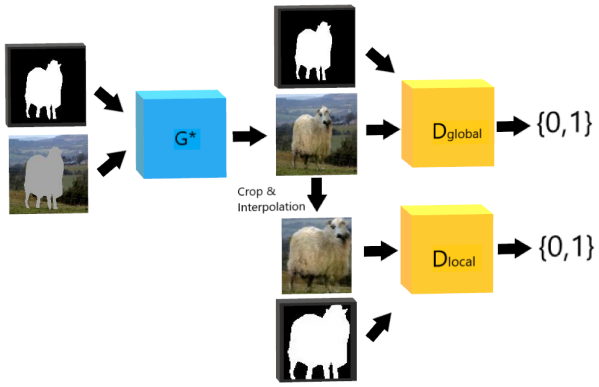

Figure 4: The foreground model training procedure.

$$\begin{aligned}G^*((1-\mathbf{M}) \odot \mathbf{I}, \mathbf{z}, \mathbf{M}) = (1-\mathbf{M}) \odot \mathbf{I} \\ + G_{fg}((1-\mathbf{M}) \odot \mathbf{I}, \mathbf{z}, \mathbf{M})\end{aligned} \tag{18}$$

Here $\odot$ is a pixel-wise multiplication. In this formulation, the foreground object generated by G^* is constrained by the scene outside the object mask which makes more sense because what previously drawn in the mask region is irrelevant for the new foreground so that region should not constrain it. This new formulation allows us to approach this problem as an image inpainting, completion problem. So the foreground model can be trained similar to GAN-based image inpainting models e.g [45], [46] instead of recurrent training. During training, real dataset images are used as an input to the generator. For each forward pass, one foreground in the dataset image is randomly selected and used. Since dataset consists of both images which comprise of single or multiple foreground objects, this training procedure leads the generator to learn to draw foreground object while reconstructing either only background or background with other foreground objects. The foreground model, G^* is trained in an adversarial scheme. So there are two discriminators which are jointly trained with G^*. The overall objective function is defined as follows.

$$L = L_{global} + \lambda_l L_{local} + \lambda_r L_{rec} + \lambda_{FM} L_{FM} \tag{19}$$

where L_{global} and L_{local} are adversarial losses, L_{rec} is a L2 reconstruction loss and L_{FM} is a feature matching loss. λ's are trade-off parameters that are determined emprically.

$$\begin{aligned}L_{global} = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D_{global}(\mathbf{x}, \mathbf{M})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})}[\log(1 \\ - D_{global}(G((1-\mathbf{M}) \odot \mathbf{x}, \mathbf{z}, \mathbf{M}), \mathbf{M}))]\end{aligned} \tag{20}$$

where $\mathbf{x}$ is a ground-truth image associated with semantic map, $\mathbf{M}$. L_{global} encourages the generator to generate an image looks like training images.

$$\begin{aligned}L_{local} = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D_{local}(S(\mathbf{x}), S(\mathbf{M}))] \\ + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})}[\log(1 \\ - D_{local}(S(G((1-\mathbf{M}) \odot \mathbf{x}, \mathbf{z}, \mathbf{M})), S(\mathbf{M})))]\end{aligned} \tag{21}$$

Here S is a bi-linear function that crops the region of interest (object bounding box) and interpolates in a fully differentiable way. This loss encourages the generator to generate foreground object looks like real foreground objects by focusing only foreground region.

$$L_{rec} = \mathbb{E}[((1 - \mathbf{BB}) \odot (G((1 - \mathbf{M}) \odot \mathbf{x}, \mathbf{z}, \mathbf{M}) - \mathbf{x}))^2] \quad (22)$$

L_{rec} encourages the generator to reconstruct the input image outside bounding-box, **BB**. The idea for using a bounding-box, **BB** instead of a mask, **M** is to give the generator, G^* more flexibility to modify the surrounding of the object accordingly. Lastly, L_{FM} is a feature matching loss used for stabilizing the training which is explained in Section 2.6. The training procedure is depicted in Figure 4.

The foreground generator design is inspired from text-to-image models such as [30] and [22]. The main difference is that the previous frame is encoded instead of text embedding. Propagating the noise vector **z** to the output more, features come from **z** are concatenated in another intermediate layer in the network; a similar technique is used in [53]. Besides, 'U-Net'[55]-like skip connection is added for the feature map obtained from the previous image. It is useful for reconstructing the previous image. The foreground generator architecture is illustrated in Figure 5.

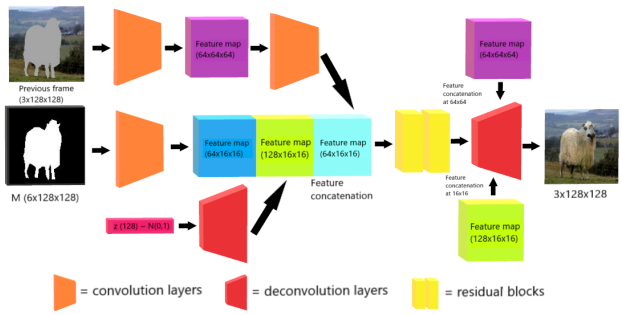

Figure 5: The foreground model network architecture. It generates an image by conditioning on both the previously generated image and the semantic layout map of the foreground object.

4.3. Background Model Learning

The purpose of the background model is to learn to generate background image (without any foreground objects) similar to backgrounds of the training images. Because, the real background images are not provided separately, the simple trick in the generator architecture is done in order to learn the background model in a classical GAN training scheme. The background model is conditioned on semantic layout map and it has two branches: (1) the first branch maps the noise vector to the background feature map and it is further processed to output $\mathbf{x}_{bg}$, the background image without any foreground objects. (2) The second branch maps the semantic layout map to another feature map and it is concatenated with background features then, they are further processed to generate **x**, the normal image with foreground objects which is similar to dataset images. The model architecture is illustrated in Figure 6. The overall training objective is defined as follows.

$$L = L_{global} + \lambda_r L_{rec} + \lambda_{FM} L_{FM} \quad (23)$$

where L_{global} is an adveserial loss as given in equation (20); however, in this case, **M** is a semantic layout map for the entire scene instead of a single foreground object. The local discriminator is not used for the backgroud model. L_{rec} is a reconstraction loss:

$$L_{rec} = \mathbb{E}[((1 - \mathbf{M}) \odot (\mathbf{x}_{bg} - \mathbf{x}))^2] \quad (24)$$

This loss encourages $\mathbf{x}_{bg}$ to be similar to **x** but without the foreground objects. L_{FM} is a feature matching loss same as the one in the equation (19).

During test time, the second branch of the generator can be discarded.

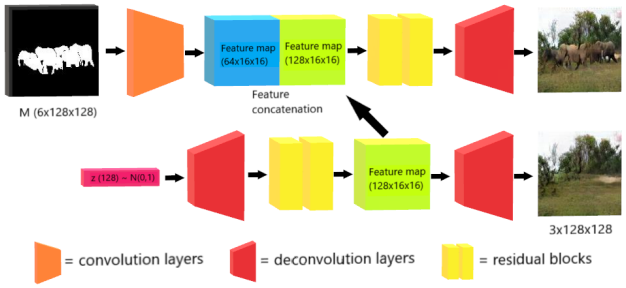

Figure 6: The background model network architecture. It generates two images: (1) the background image, (2) the normal image (background+foreground) that is conditioned on the semantic layout map.

5. Experiments & Results

5.1. Dataset

Microsoft Common Objects in Context (MS COCO) dataset[16] is used to evaluate the model performance. The dataset contains 164K training images over 80 semantic classes. Images are annotated with (foreground) object semantic masks and bounding boxes as well as 5 image captions. The dataset is very diverse and complex that contains images of multiple objects in natural environments and varied viewpoints. To ease the problem, semantically related 6 classes are chosen: Cow, sheep, giraffe, zebra, bear, and elephant. These classes have similar background image distribution and there are approximately 11K images in total for these classes.

5.2. Baseline

The proposed method is compared against three different baselines both quantitatively and qualitatively. All baseline models are non-sequential; they generate images at once.

5.2.1 Standard CNN

The first baseline is a standard CNN model which translates a semantic layout into a natural image. This model is trained without GAN loss; only L1 reconstruction loss is minimized during training. So it is a deterministic model which basically learns this mapping by heart without any stochasticity. There is no discriminator used during training, everything else (e.g. generator architecture, hyper-parameters) are the same as Pix2pix which is described in the next section. The purpose of this baseline model is to emphasize the correlation between the qualitative and quantitative results.

5.2.2 Pix2pix

In literature, Pix2pix [13] is considered as the state of the art baseline for the image-to-image translation problem (The problem in this work is the semantic map to natural image translation.); therefore, Pix2pix is adopted as the baseline model. It is a conventional GAN model which is conditioned to a semantic layout. In order to stabilize the training, the L1 reconstruction loss is utilized. The one of the most important issue about this model is that the stochasticity is very limited, the noise is provided only in the form of dropout, applied on several layers of the generator at both training and test time. The model is trained exactly the same as described in the article [13].

5.2.3 Pix2pix++

In the proposed sequential model, the generator and the discriminator architectures and also the training tricks differ compared to Pix2pix. To emphasize the advantage of the sequential model more, another baseline (called Pix2pix++) is used as well. Pix2pix++ is a conventional GAN model which conditioned to a semantic layout similar to Pix2pix. However, in this case, the same discriminator networks and the similar generator network with the sequential case are used. Also, exactly the same training tricks (e.g. spectral normalization, feature matching loss) and hyper-parameters are used during training.

5.3. Evaluation Metrics

Experimental results are presented in terms of Frechet Inception Distance (FID) and the sementic segmentation accuracy.

5.3.1 Frechet Inception Distance (FID)

FID is a recently proposed metric by Heusel et al.[17] which measures the distance between two different image data distribution (real image dataset vs generated image dataset) in feature space. [17] empirically shows that FID is consistent with human judgment in terms of visual fidelity and with increasing disturbances (e.g Gaussian noise, salt and pepper noise, Gaussian blur, swirl, black rectangles). In default settings, it extracts 2048-dimensional features from Inception V3 network[18] pre-trained on ImageNet dataset[19] (3th max-pooling layer is used.). It assumes that features are of multivariate Gaussian distribution. The Frechet distance[23] between these two Gaussian distributions is then used to quantify the quality of generated samples as given by the following formula.

Method	FID
Standard CNN	120.7
Pix2pix	34.0
Pix2pix++	24.0
Sequential (proposed)	28.7
Sequential (Bg from Pix2pix++)	**23.2**

Table 1: Quantitative evaluation results: Frechet Inception Distance (FID). Lower score means generated image samples are similar to real images in terms of visual quality and content. Bg stands for background.

$$d(p_1, p_2) = ||m_1 - m_2||^2 + Tr(C_1 + C_2 - 2(C_1 C_2)^{1/2}) \tag{25}$$

Here m_i is a mean and C_i is a co-variance matrix of the observed distribution, p_i. In the extreme case, $\lim_{p_1 \to p_2} d(p_1, p_2) = 0$. Intiutively, FID is lower, if generated image samples are similar to real images in terms of visual quality and semantic content.

5.3.2 Semantic Segmentation Accuracy

Semantic segmentation accuracy is a recently used measure to estimate the synthesized image quality (e.g. used in [13], [14], [15]). The intuition is that if the generated image is realistic; then, an off-the-shelf semantic segmentation model should be able to segment the image correctly. In this work, deep learning semantic segmentation model, Deeplab [25] which is pre-trained on MS-COCO dataset is used as a segmentation model. Intersection-over-Union (IoU) is used to evaluate the segmentation performance. The IoU is the standard performance measure that is commonly used for the semantic segmentation problem. Given a predicted and ground-truth semantic maps, the IoU score gives the similarity between the predicted region and the ground-truth region for an object present in the image. It is defined as the size of the intersection divided by the union of the two regions. The IoU score is high if the semantic segmentation model prediction is accurate.

5.4. Experimental Setup

5.4.1 Experimental Objectives

The main purpose of the experiment is to show that if the proposed sequential model can generate an image similar to dataset images in an incremental fashion (object-by-object) and if there is an improvement in user control over the generated image. In order to evaluate that generated images are similar to dataset images, FID score that is described in the

Method	Mean IoU Training-set	Mean IoU Validation-set
Standard CNN	0.298	0.272
Pix2pix	0.504	0.480
Pix2pix++	0.608	0.605
Sequential (proposed)	**0.650**	**0.650**
Ground-truth	0.803	0.770

Table 2: Quantitative evaluation results: Semantic segmentation performance, Mean IoU (Intersection-over-Union). Ground-truth corresponds to model performance on the real dataset images, so it can be considered as an upper bound.

Section 5.3.1 is computed and compared with the baseline cases. Qualitative results are provided to illustrate the improvement in the user control (both the control through the disentangled noise and the control on the affine transformation of the object mask). The second purpose of the experiment is to evaluate if the sequential model generates better quality and more diverse images compared to conventional GAN models. To evaluate the quality and diversity, FID and mean IoU scores are computed and compared with the baseline cases as well as qualitative results are provided. Another purpose of the experiment is to evaluate whether the sequential model can deal with occlusion and near objects better compared to the baseline model. Qualitative results are provided to assess that, too.

Figure 7: Qualitative results: 128x128 example images generated sequentialy in 3 steps by the proposed model. Object classes are giraffe, bear, elephant, sheep, cow and zebra, cow, elephant, respectively.

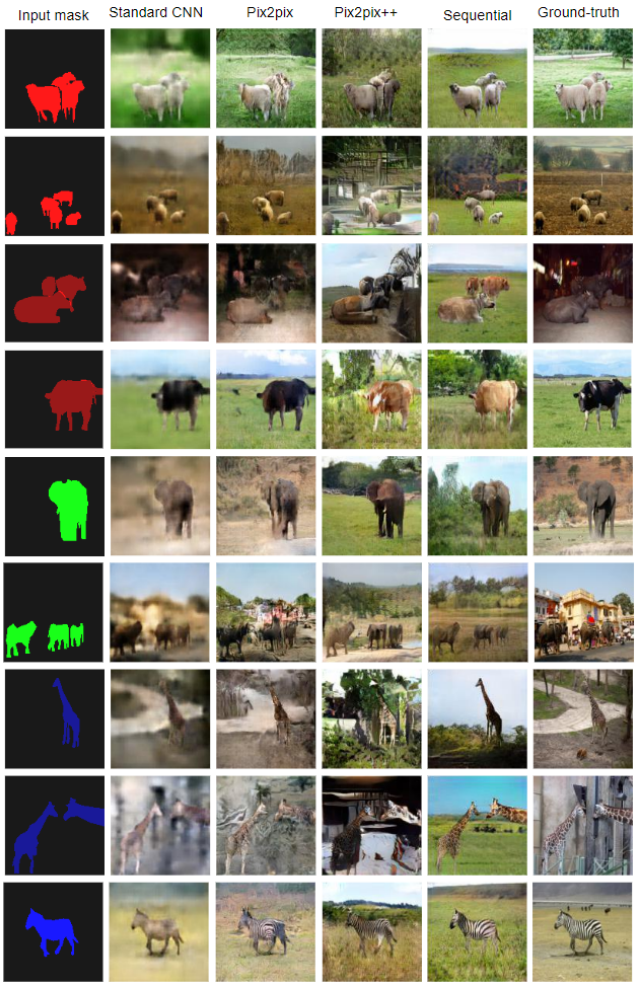

Figure 8: Qualitative results: 128x128 example images generated by different models using train-set object masks. Ground-truth corresponds to real dataset images.

Figure 9: Qualitative results: 128x128 example images generated by different models using validation-set object masks.

5.4.2 Experimental Pipeline

The proposed sequential model has two sub-models: the background generator and the foreground generator. These models are trained separately. During the test time, the baseline generators take the entire semantic map as an input and generate an image. In the sequential case, first, the background is generated and the foreground objects are generated one-by-one in a random order until all the foreground objects are drawn. There is no user observation and control in this process. Afterward, FID and the mean IoU scores are computed using the

10K images. According to [54], 10K is a reasonable number for FID score. In order to have an idea about models' generalization performances, mean IoU scores are also computed on the images generated from the validation-set masks (contains 450 images).

5.4.3 Training Details

For the proposed sequential model and the Pix2pix++, the Adam optimizer [20] with $\beta_1 = 0$ and $\beta_2 = 0.9$ are used for training. By default, the learning rate both for the discriminators and the generator is 0.0002. The learning rate is decayed to $\frac{1}{2}$ of its previous value every 80 epochs. All the models are trained for 480 epochs. The parameters of the generators are updated once in every 5 discriminators' updates. The trade-off parameters in the loss function given in the equation (19) are set $\lambda_l = 0.1$, $\lambda_r = 0.00001$, $\lambda_{FM} = 1$ and in the equation (23) $\lambda_r = 100$, $\lambda_{FM} = 1$. All the implementation are done using Pytorch deep learning framework [24]. The code will be available at: `https://github.com/0zgur0/scene_generation_using_GANs`

5.4.4 Expected Outcomes

As a result of the experiment, for the sequential model, higher mean IoU score is expected since the improvement in the image quality is expected. Because in the experiment, images are generated without any user observation and control, it is possible that the sequential model generates an image which is not similar to dataset images e.g it might even be flying sheep. So this might cause higher FID score. Even though there is an improvement in the image quality, FID score might be higher for the sequential model. But still, similar FID scores are expected for the sequential model and Pix2pix++. Higher image diversity is expected since the sequential model has more flexibility and stochasticity. The sequential model should improve the user control over the scene as well as deal with occlusion and the near objects better.

5.5. Quantitative Results

FID scores for the baseline models and the proposed model are given in Table 1. For a better interpretation, FID score is computed for images that are created by copying foreground objects from the image generated by the sequential model and pasting on the corresponding image generated by Pix2pix++. So these images are more similar to training images in terms of content (This guarantees that there are no flying sheep etc.); however, boundary artifacts occur due to copy-paste process which causes higher FID, too. Pix2pix++ achieves better FID score, 24.0 then the proposed model does (28.7); however, images generated by copy-paste give the best FID score, 23.2 in spite of boundary artifacts. Mean IoU scores are given in Table 2. It is computed for the real dataset images (ground-truth) as well, so the corresponding scores can be considered as an upper bound for mean IoU. The best scores are achieved by the proposed model with a significant difference.

5.6. Qualitative Results

Various qualitative results are presented in order to show the strength and the advantage of the proposed model. The original size of the images is 128x128 pixels. In Figure 7, the sequential generation process is illustrated in 3 steps. The generation process is not limited to 3 steps, it is chosen just for convenience. Some example images generated by the baseline models and the proposed model are given in Figure 8 and 9 to emphasize the quality and the diversity. The enhancements in the user control are shown in Figure 10 and 11. In Figure 10, images are generated in 2 steps. For each subset of images, the latent (noise) vector at the first step (It determines the background.) is identical and the second latent vector (It determines the foreground.) is different; therefore, these images should have the same backgrounds and the different foregrounds. This is the results that show explicit user control through an object-specific latent vector. In Figure 11, the baseline model (Pix2pix++) and the proposed model responses are compared when the user transforms (translation, rotation, scaling) the input object mask (Same noise vector is used for each subset of images.). Finally, the behaviors of the near objects and the occlusion are compared in Figure 12.

Figure 10: Qualitative results: 128x128 example images generated sequentialy in 2 steps. For each consecutive three images, the latent vector for the background is identical and the one for the foreground is different so they should have the same backgrounds and the different foregrounds.

5.7. Analysis

The sequential model achieves similar FID score with the strong baselines, Pix2pix and Pix2pix++. This shows that the proposed sequential model learns to generate images faithfully as the conventional models do; at the same time, it provides more user control compared to conventional models. Mean IoU scores, FID score for the data created by copy-paste (23.2) and the qualitative results suggest that the sequential model improves the image quality. Although the image quality is higher, FID score is worse for the sequential model compared to Pix2pix++. Probably, the reason is the fact that the

Figure 11: Qualitative results: The affine transformation behaviors of the proposed model (Sequential) and the baseline model (Pix2pix++). The same latent vectors are used for each consecutive three images.

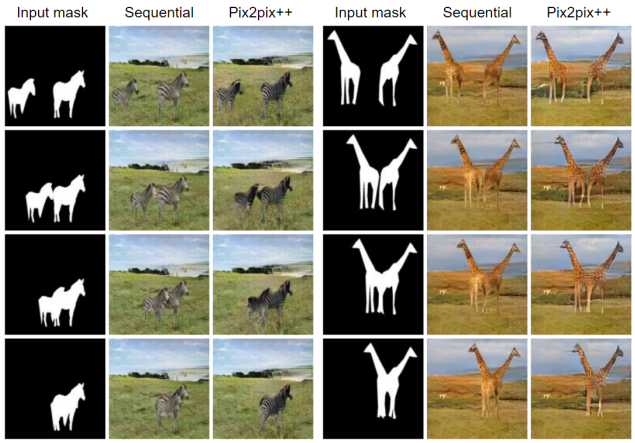

Figure 12: Qualitative results: The occlusion and the near objects behaviors of the proposed model (Sequential) and the baseline model (Pix2pix++). The same latent vectors are used for the images at the same column.

sequential model can generate images out of the training data distribution. So this result supports that the sequential model also increases the diversity.

The proposed model also contributes to the controllability by resolving the affine transformation issue conventional models have. If you examine the Figure 11 carefully, even when the object is translated a little, both the foreground and the background is changed for Pix2pix++. In contrast, for the sequential model only small changes occur in the foreground. When the object is rotated, the change in the background can be very severe; it can even replace the part of the sky with the ground. Apparently, the conventional model learns the correlation between the ground and the legs of the zebra and when the zebra is rotated, it confuses and draws a meaningless image. Moreover, the proposed model resolves the occlusion issue. Figure 12 compares the responses of the baseline and the proposed models when the object masks get closer. In the giraffe example, when the masks get closer (second row), the colors of the giraffes become similar in the Pix2pix++ case. When they touch each other slightly (third row), Pix2pix++ merges giraffes and draws continuous pattern. In the full occlusion case (fourth row), it draws a giraffe with two heads. In the zebra example, when the masks touch each other slightly (second row), Pix2pix++ splits the zebras at the wrong place. In the full occlusion case (fourth row), the only single zebra is drawn. Those artifacts are not occurred in the proposed model.

6. Discussion

The proposed sequential image generation approach can be applied in different image generation tasks and on different datasets. However, the learning scheme adopted in this work assumes that dataset images have a background and a couple of foreground objects. Some of them have single and some of them have multiple foreground objects. Therefore, this model can not be applied directly on any dataset which has a semantic annotation. In order to adopt this approach for the dataset which has a dense semantic annotation e.g Cityscapes dataset [49], the learning scheme needs to be modified.

7. Conclusion

In this master's thesis, a novel sequential image generation model based on Generative Adversarial Networks is proposed. This model adopts the layered structure modeling for images and generates an image step-by-step starting with the background of the scene and forms the scene progressively by putting a single foreground object at each step. The proposed approach improves the controllability of the image generation process through a element/object-level control mechanism. The experimental results suggest that the sequential generation scheme also improves the image quality and the diversity. In addition to them, it is shown that it resolves the occlusion artifacts of the existing conditional GAN models. As a future work, the mask generation can be studied. The current model requires an input mask but there are the only limited amount of pre-defined masks. In order to improve controllability over mask selection and to create a model that can generate images from scratch or from input text description, the mask generator is needed.

References

[1] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

[2] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks."arXiv preprint arXiv:1511.06434 (2015).

[3] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).

[4] Miyato, Takeru, et al. "Spectral normalization for generative adversarial networks."arXiv preprint arXiv:1802.05957 (2018).

[5] Yang, Jianwei, et al. "LR-GAN: Layered recursive generative adversarial networks for image generation." arXiv preprint arXiv:1703.01560 (2017).

[6] Kwak, Hanock, and Byoung-Tak Zhang. "Generating images part by part with composite generative adversarial networks." arXiv preprint arXiv:1607.05387 (2016).

[7] Arjovsky, Martin, Soumith Chintala, and Lon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

[8] Mao, Xudong, et al. "Least squares generative adversarial networks."2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017.

[9] Qi, Guo-Jun. "Loss-sensitive generative adversarial networks on lipschitz densities."arXiv preprint arXiv:1701.06264 (2017).

[10] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

[11] Goodfellow, Ian. "NIPS 2016 tutorial: Generative adversarial networks." arXiv preprint arXiv:1701.00160 (2016).

[12] Harn, Yeu-Chern, and Vladimir Jojic. "3C-GAN: AN CONDITION-CONTEXT-COMPOSITE GENERATIVE ADVERSARIAL NETWORKS FOR GENERATING IMAGES SEPARATELY." (2018).

[13] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." arXiv preprint (2017).

[14] Wang, Ting-Chun, et al. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs." arXiv preprint arXiv:1711.11585 (2017).

[15] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." arXiv preprint arXiv:1703.10593 (2017).

[16] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.

[17] Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium." Advances in Neural Information Processing Systems. 2017.

[18] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." " Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[19] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks."" Advances in neural information processing systems. 2012.

[20] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization."arXiv preprint arXiv:1412.6980 (2014).

[21] Karacan, Levent, et al. "Learning to generate images of outdoor scenes from attributes and semantic layouts."arXiv preprint arXiv:1612.00215 (2016).

[22] Hong, Seunghoon, et al. "Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis." arXiv preprint arXiv:1801.05091 (2018).

[23] M. Frchet. Sur la distance de deux lois de probabilit. C. R. Acad. Sci. Paris, 244:689692, 1957.

[24] Paszke, Adam, et al. "Automatic differentiation in PyTorch."(2017).

[25] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.

[26] Kaneko, Takuhiro, Kaoru Hiramatsu, and Kunio Kashino. "Generative attribute controller with conditional filtered generative adversarial networks." IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 2. 2017.

[27] Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks."arXiv preprint arXiv:1601.06759 (2016).

[28] van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders."Advances in Neural Information Processing Systems. 2016.

[29] Zhang, Han, et al. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks." IEEE Int. Conf. Comput. Vision (ICCV). 2017.

[30] Reed, Scott, et al. "Generative adversarial text to image synthesis." arXiv preprint arXiv:1605.05396 (2016).

[31] Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." arXiv preprint arXiv:1610.09585 (2016).

[32] Salimans, Tim, et al. "Improving GANs using optimal transport." arXiv preprint arXiv:1803.05573 (2018).

[33] Zhao, Junbo, Michael Mathieu, and Yann LeCun. "Energy-based generative adversarial network." arXiv preprint arXiv:1609.03126 (2016).

[34] Metz, Luke, et al. "Unrolled generative adversarial networks." arXiv preprint arXiv:1611.02163 (2016).

[35] Salimans, Tim, et al. "Improved techniques for training gans." Advances in Neural Information Processing Systems. 2016.

[36] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[37] Cha, Miriam, Youngjune Gwon, and H. T. Kung. "Adversarial nets with perceptual losses for text-to-image synthesis." arXiv preprint arXiv:1708.09321 (2017).

[38] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European Conference on Computer Vision. Springer, Cham, 2016.

[39] Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." arXiv preprint arXiv:1603.07285 (2016).

[40] Reed, Scott, et al. "Parallel Multiscale Autoregressive Density Estimation." International Conference on Machine Learning. 2017.

[41] Yan, Xinchen, et al. "Attribute2image: Conditional image generation from visual attributes." European Conference on Computer Vision. Springer, Cham, 2016.

[42] Reed, Scott E., et al. "Learning what and where to draw." Advances in Neural Information Processing Systems. 2016.

[43] Denton, Emily L., Soumith Chintala, and Rob Fergus. "Deep generative image models using a laplacian pyramid of adversarial networks." Advances in neural information processing systems. 2015.

[44] Park, Hyojin, Youngjoon Yoo, and Nojun Kwak. "MC-GAN: Multi-conditional Generative Adversarial Network for Image Synthesis." arXiv preprint arXiv:1805.01123 (2018).

[45] Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[46] Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion." ACM Transactions on Graphics (TOG) 36.4 (2017): 107.

[47] Ian Goodfellow, Yoshua Bengio and Aaron Courville. "Deep Learning." MIT Press, `http://www.deeplearningbook.org`, 2016, pp. 326-366.

[48] Zhang, Han, et al. "Self-Attention Generative Adversarial Networks." arXiv preprint arXiv:1805.08318 (2018).

[49] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[50] Wah, Catherine, et al. "The caltech-ucsd birds-200-2011 dataset." (2011).

[51] Zhou, Bolei, et al. "Scene parsing through ade20k dataset." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 1. No. 2. IEEE, 2017.

[52] Liu, Ziwei, et al. "Deep learning face attributes in the wild." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[53] Zhu, Jun-Yan, et al. "Toward multimodal image-to-image translation." Advances in Neural Information Processing Systems. 2017.

[54] Huang, Gao, et al. "An empirical study on evaluation metrics of generative adversarial networks." (2018).

[55] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.