IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Special Issue, 7, March 2019 E-ISSN: 2347-2693

# A Survey of Text-to-Image Generative Adversarial Networks

**Siddhivinayak Kulkarni[1], Amol Dhondse[2], Anurag Katakkar[3*], Nitish Bannur[4], Trupti Deshpande[5]**

[1]Department of Computer Science, MIT College of Engineering, India
[2]Digital Business Group ISL, IBM, Pune, India

[*]*Corresponding Author: anuragkatakkar@gmail.com, Tel.: +91 9823198749*

**Available online at www.ijcseonline.org**

***Abstract*—**In recent years, generative models have gained alot of attention in the deep learning community. In particular,Generative Adversarial Networks (GANs), proposed by Ian Goodfellow et al. in 2014, and their variants have emerged as a powerful method which performs significantly better than other generative models such as Restricted Boltzmann Machines or Variational Auto-Encoders. In this paper, we focuson a specific type of GANs, the Text-to-Image GANs, and review some of the most seminal work which has been conducted in this area. We provide a high-level description of the architectural components of these models and also review their performance on variousdatasets. Further, we discuss how these architectures are suitedfor the particular use case of text-to-face image synthesis for generating images of human faces from text descriptions.

***Keywords*—GenerativeAdversarial Networks, Text-to-ImageGANs, Deep Learning**

## I. INTRODUCTION

In recent years, Deep Learning techniques have succeeded in a wide range of tasks including Computer Vision, Speech Recognition, and so on. This success has also extended to generative tasks such as image synthesis. The GAN framework [2] proposed by Goodfellow et al. has received a lot of attention for being able to solve many problems that require data synthesis and has emerged as a prominent method for addressing generative tasks. Generative models address the problem of learning the underlying data distribution without actually memorizing the data examples.

### A. Generative Adversarial Networks

In 2014 Ian Goodfellow et al. proposed a novel generative model called Generative Adversarial Networks [2] which overcame or sidestepped many problems with incumbent generative models such as Variational Auto-encoders [5], Deep Belief Networks and Deep Boltzmann Machines relating to image quality, learning the density estimation of data, and flexibility of defining the loss function and network, topology which can be found listed in this report [3], and this tutorial [26]. This paper [2] also described how the GAN framework can be extended to conditional generative models. Since then, the GAN framework has gained a lot of attention and popularity in the deep learning community and has been applied to a wide variety of generative task such as Image synthesis (as demonstrated by the original GAN by Goodfellow et al. [2]), Image super-resolution [27][28][29] , Text-to-Image Synthesis[3][7][14][17][18][19][20][21][22][23], 2D to 3D model generation [30], and so on. Recently, a "painting" which was the output of an imagesynthesis GAN was auctioned for the US $432,500 [38].

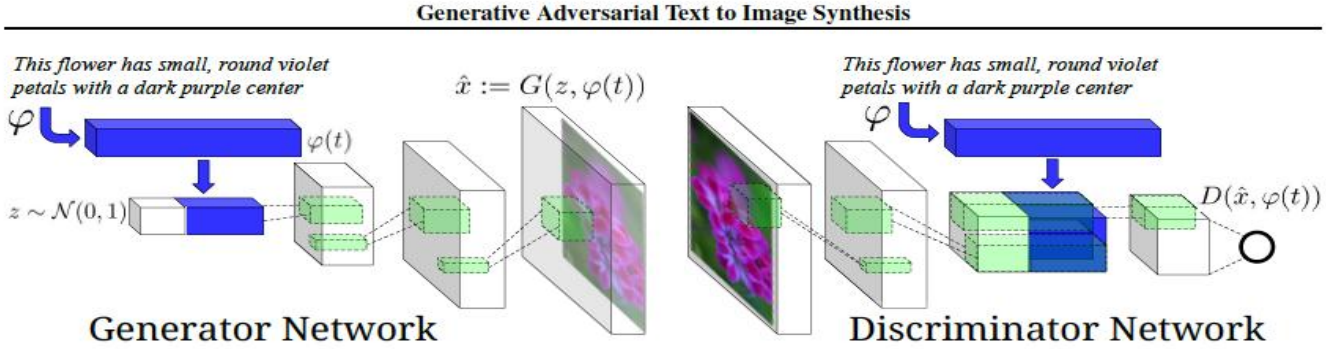


**Figure 1. The GAN-CLS architecture proposed by Reed et al. [7]. Image Source: [7]**

© 2019, IJCSE All Rights Reserved 

The basic model of the GAN [2] may be described as a minimax game between two multi-layer perceptrons called the Discriminator (D) and the Generator (G). The role of the Generator is to generate new data samples that capture the probability distribution of the training data, while that of the Discriminator is to determine whether a given sample is real (came from the data) or fake (was generated by G). These roles have been likened to those of a "team of counterfeiters" and the police [2]. The competition between the two adversaries accomplishes the task of optimizing both to achieve high accuracy in their respective roles. After sufficient time training the model, a point is reached where G becomes so good at generating fakes that D can no longer distinguish between fake and real samples.

Formally, this model may be defined by the following equation:

$$\min_G \max_D V(G,D) = E_{x\sim Pdata(x)}[\log(D(x))] + E_{z\sim pz(z)}[\log[1-D(G(z))]]$$

where $p_{data}$ is the probability distribution of the input data, and $p_z$ is the probability distribution of the input noise. From the above equation, we can see that the Discriminator tries to maximize the probability of correctly labelling the real and fake samples, while the Generator tries to maximize the probability of fooling the Discriminator and make it classify fake samples as real.

The remainder of this paper is organized as follows: In Section II we discuss Text to Image GANs including commonly used datasets and some of the most prominent Text-to-Image GAN models. Section III talks about how we believe these models can be used for the specific use case of Text-to-Face Image Synthesis,and finally, Section IV presents concluding thoughts.

## II. TEXT-TO IMAGE GAN

We have previously mentioned how the original GAN paper [2] introduced the extension of the GAN framework to conditional models. In such models, a conditional input vector c is concatenated to the noise vector z and this is jointly passed as an input to the Generator G. The vector c can have arbitrary meanings but for the task of Text-to-Image synthesis, it is usually the vector-space representation of text captions or text embeddings. This was illustrated by Mirza, Osindero, 2014 [8] in their model which was able to generate MNIST digits using class labels as conditional information. They made the following modifications to the objective function by introducing the conditional probability on the extra (conditional) information y:

$$\min_G \max_D V(G,D) = E_{x\sim Pdata(x)}[\log(D(x|y))] + E_{z\sim pz(z)}[\log[1-D(G(z|y))]]$$

Though generative models for text to image synthesis had already been proposed using alternative methods such as the DRAW model [4] (which extended the Variational - Autoencoder framework [5]) and its extension, the AlignDRAW model [6], Reed et al., 2016 [7] were the first to report success in this task using the GAN framework. Many novel frameworks for Text-to-Image Synthesis using GANs have emerged since. This section describes some of the most prominent and path-breaking text-to-image GAN models.

### A. Commonly used datasets

Naturally, the training of Text-to-Image GANs requires datasets of images along with text descriptions or captions. In the Text-to-Image GAN models discussed in this paper, the following datasets have been used:

- Caltech-UCSD Birds 200 (CUB-200) [15] containing images of 200 bird species,
- Oxford-102 [31], which contains flower images belonging to 102 classes,

**Table 1: Commonly used datasets for Text-To-GANs**

- Large-Scale Scene Understanding (LSUN) [32], containing images of scenes belonging to categories such as bedroom, bridge, classroom, and so on,
- Microsoft COCO: Common Objects in Context (MS COCO) [33],
- ImageNet: A large-scale hierarchical image database [34],
- MPII Human Pose dataset (MHP) [16]

Table 1

| Dataset Name | Details | #Classes | #Images | Annotations |
|---|---|---|---|---|
| CUB-200 [15] | Images of 200 Bird Species | 200 | 11,788 | 15 Part Locations 312 Binary Attributes 1 Bounding Box |
| Oxford-102 [31] | Images of 102 Flower Species | 102 | 8,153 | Segmentation, $Chi^2$ Distances, Image Labels |
| MPII Human Pose [16] | SOTA for Human Pose Estimation | 410 (human actions) | ~ 25,000 | Head, Joint, activity/category annotations |
| LSUN [32] | Contains Images of various scenes such as bedroom, kitchen, | 10 | 9.89mn(training) 3,000(Validation) 10,000 (Test) | Image content, Scene description, and room layer annotations |

© 2019, IJCSE All Rights Reserved 

| | | | | |
|---|---|---|---|---|
| | restaurant, bridge, tower, etc. | | | |
| MS COCO [33] | Dataset for object detection, segmentation and captioning | 80(Object) 91(Stuff) | 330,000 >200,000 labelled | Object Segmentation, 5 captions per image 250k people with key points |
| ImageNet [34] | Large-Scale object detection dataset organized according to the WordNet hierarchy. | 21,841 synsets | 14.19mn | Bounding Boxes (1.03mn images) SIFT features (1,000 synsets, 1.2mn images) |

**B. GAN-INT-CLS**

This model, by Reed et al. [7] combines the advantages of powerful RNN architectures and DCGANs [13]. The challenge of Text-to-Image synthesis is decomposed into the 2 sub-problems of generating text encodings that capture visual details correctly, and of using these text encodings to generate realistic images. Their main contribution is a simple yet effective GAN architecture and training method that enables synthesis of realistic images from hand-written text descriptions, which they train on the Oxford 102 and CUB-200 datasets [15].

The text description is converted to an encoding using a hybridcharacter-levelConvolutional-Recurrent neural network [9] which is pre-trained on 1,024-dimensional GoogLeNet image embeddings. This encoding is then projected to a 128-d space in both the D and the G, before depth concatenation.

In the Generator, the text encoding is concatenated to the noise and then feed-forwarded through the network to obtain the synthetic image. In the Discriminator, the concatenation occurs after several layers of stride-2 convolutions andspatial batch norm, followed by leaky ReLU until a spatial dimension of 4x4 is reached.

Finally, a 1x1 convolution is applied for rectification followed by a 4x4 convolution to obtain the final score from the D. Batch normalization is performed on all convolutional layers.

This model architecture is very similar to the DCGAN [13] architecture apart from the concatenation of the text encodings.

The work in [7] describes three methods of training their GAN architecture, namely, GAN-CLS, GAN-INT, and GAN-INT-CLS.

**GAN-CLS**: The GAN architecture must learn to identify not only whether the image and captions generated are both correct, but also if they correspond to each other. For this task, an additional input is added to the GAN during training, that of the real image with a fake caption. The algorithm below explains the training process:

**Algorithm:**

Input: mini-batch of images x, matching text t, mismatching text, number of steps for training S;

**For**{n = 1 to S}{
  Encode matching text description;
  Encode mismatching text description;
  Draw sample of random noise;
  Forward through generator;
  sr = D (real image, right text);
  sw = D (real image, wrong text);
  sf = D (fake image, right text);
  Loss D = log(sr) + (log (1 - sw) + log (1 - sf))/2\;
  Update Discriminator;
  Loss G = log(sf);
  Update Generator;}
**end**

{**Algorithm 1:GAN-CLS Algorithm as described in [7]**}

**GAN-INT**: The paper further proposes another method of generating more text embeddings based on the property of interpolation between text embeddings as described here [11] [12] which allows the gaps between training points to be filled by using these interpolations. Finally, the paper also proposes a combined GAN-INT-CLS architecture.

The authors used a mini batch size of 64 images and the model was trained for 600 epochs, both D and G had same base learning rate as 0.0002, and used the ADAM solver [10] with a momentum of 0.5. A 100-dimensional unit normal distribution was used by the authors for sampling Generator noise from. Alternating steps are taken to train the D and the G.

The authors report the performance of all 4 models: GAN, GAN-CLS, GAN-INT, and GAN-INT-CLS

Results on CUB-200 [15]: GAN and GAN-CLS display some accuracy of colour but the images look unrealistic overall. GAN-INT and GAN-INT-CLS show much more realistic images which correspond to the text caption to a great extent

Results on the Oxford-102 [31]: All four models produce plausible images which correspond to the caption but the

© 2019, IJCSE All Rights Reserved 

GAN model shows most morphological variation while others produce rather class-consistent images.
In comparison with the AlignDRAW [6] model, the GAN-CLS generates much clearer and higher-resolution images however AlignDRAW [6] is more sensitive to single word changes in the text captions.

The results of the GAN-CLS on the MS COCO [33] dataset show the ability of this model to generalize over generating images with multiple objects and various backgrounds.

### C. Generative Adversarial What- Where Network

This paper proposes a novel method of controlling not only what a GAN draws but also where it draws. To control the location of the object being drawn, the authors propose two models: one which is conditioned on bounding boxes and another which is conditioned on key-points. The authors also achieve generation of 128x128 resolution images whereas earlier models had only ever achieved generation of 64x64 images. They train their model on the CUB-200[15] dataset as well as the MHP [16] dataset. By training on the latter dataset, the authors also propose the first text-to-human image model. There is also a modification to the method of generating text encodings over the GAN-INT-CLS [7] by using a char-CNN-GRU [35] instead of a char-CNN-RNN [9].

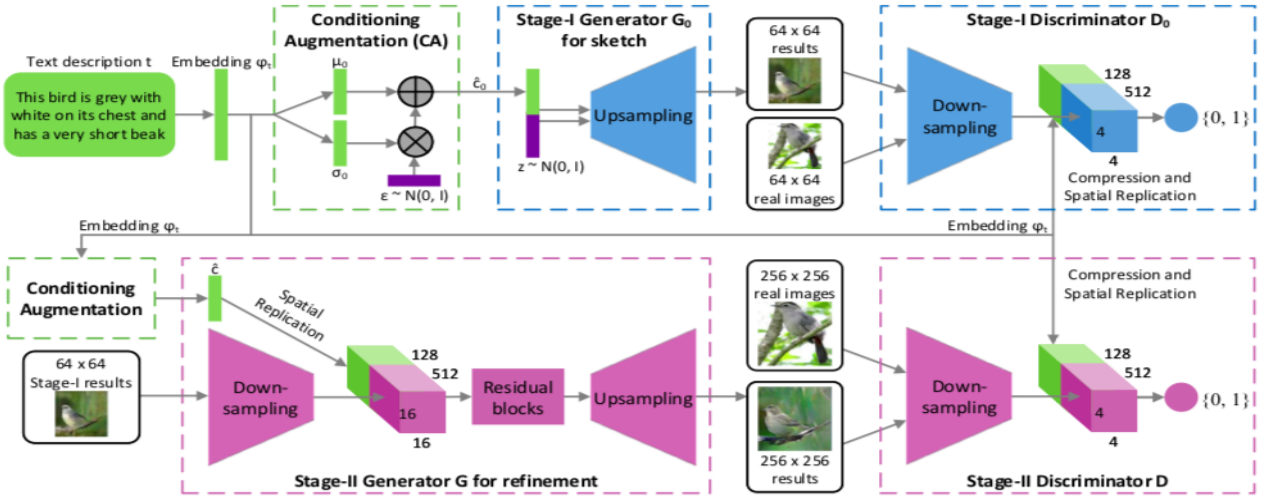


**Figure 2 StackGAN architecture by Hans et al. [19] Image Source: [19]**

**Bounding-box-conditional text-to-image model:**
Similar to the GAN-CLS [7], the text is converted to a text encoding using a pre-trained encoder.
The Generator and Discriminator are both divided into two branches: local and global. The local branch is responsible for ensuring that the object is drawn in the desired location. It does this by setting all the entries in the intermediate tensor obtained by convolution or deconvolution to zeros.
In the Generator, the Local and Global branches are joined using depth-concatenation of the MxMxT (T is the length of text encoding) dimensional vectors whereas,in the Discriminator, this task involves an additive combination of the 2 vectors.

**Key point-conditional text-to-image model:**
This is a slightly more complex method in which the key-point information which is initially encoded as an MxMxK tensor (where K is the number of key-points and each channel corresponds a particular key-point) is fed into multiple stages of the Generator.

In the Discriminator, the text encoding is also combined at 2 stages instead of 1. First, additively with the Global branch. Second, also with the MxMxT tensor of the local branch which is then gated with the MxMxK binary key-point tensor mask. After several convolutions, the local and global pathways are additively combined and fed into the final layer to produce the scalar Discriminator score.

The images in the CUB-200[15] dataset was complemented by 10 single-sentence descriptions of each bird image. Further, images from the MHP [16] dataset were also used with 3 sentence descriptions per image. 19,000 images were used from this dataset with multiple sets of key point coordinates for each of the 16 joints.Captions were encoded using the char-CNN-GRU model described in [35].For both the GAWWN architectures described above, the authors used the ADAM solver [10] with a batch size of 16 and a learning rate of 0.0002.

**Performance on CUB-200 [15]:** The bounding box technique shows the ability to shrink the bird with respect to the background while still maintaining its shape since the aspect ratio of the bounding box remains the same. However, the direction in which the bird is facing cannot be controlled.

© 2019, IJCSE All Rights Reserved 

The key-point technique,on the other hand, can warp the shape of the birds by changing the relative distance between key-points while still maintaining the overall shape of the bird. Interestingly, this approach can also control the direction in which the bird faces.

**Performance on MHP [16]:** This is a much more complex task owing to the greater number of permutations of human key-points as opposed to bird key-points. The images generated are blurrier than the ones generated for the birds. The model performs well on simple poses such as yoga, or golf but finds complex poses such as upside-down humans especially challenging.

The key point made by the authors here is that decomposing the problem into smaller sub-problems allows for the generation of clearer and higher resolution images.

**D.STACKGAN**

In 2017 Han Zhang et al. proposed the StackGAN [19] which was the first technique to successfully synthesize 256x256 resolution images from a text description. As the name

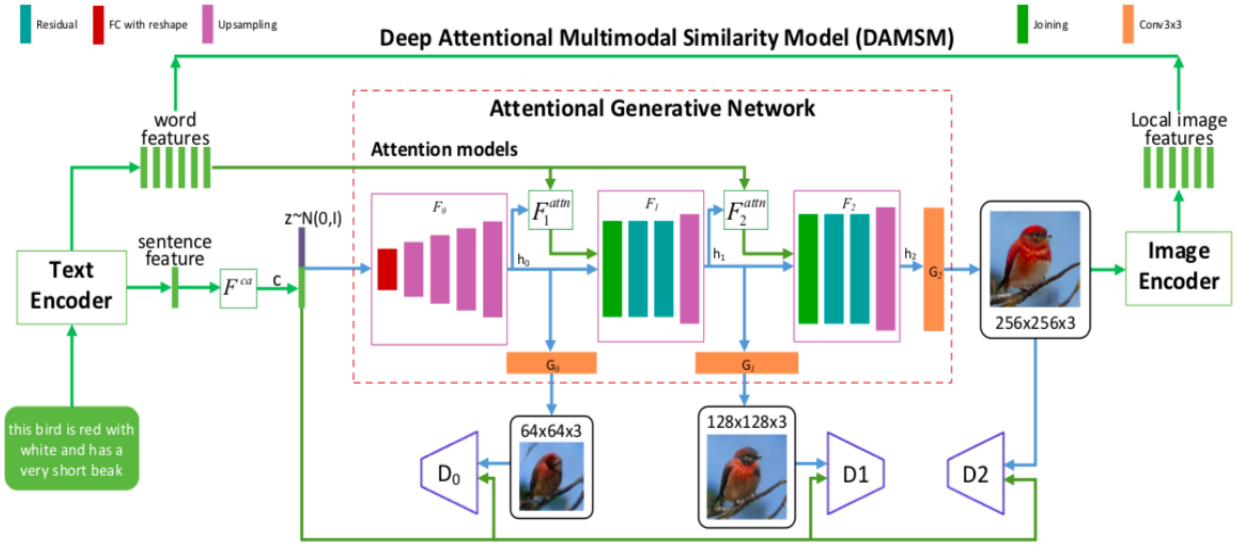


**Figure 3. The AttnGAN architecture proposed by Tao et al. [20]. Image Source: [20]**

suggests, StackGAN achieves this by stacking 2 GANs together. It tackles the problem of text-to-image synthesis in two stages. One may naively assume that adding more upsampling layers in the Generator network ought to help generate higher resolution images however this usually leads to nonsensical images and training instability [19]. The authors of this paper draw an analogy between their StackGAN model and human painters: the model first generates a rough sketch (64x64) in Stage I, then stacking the Stage II GAN on top of this to generate high-resolution 256x256 images. The authors state that by conditioning the Stage II GAN on the Stage I result as well as on the text encodings, Stage II learns to capture text information that is omitted by the Stage I GAN and draws a more detailed object. Thus, an attempt to draw the final image from the rough sketch produced by Stage I is easier than directly attempting to draw the final 256x256 image from the image distribution.

Thus, the Stage I GAN draws a crude image capturing only basic colour and primitive shape of the object. The Stage II GAN refines this image by using the text description again and generating a more detailed image.

The StackGAN was evaluated on the CUB-200 [15] (preprocessed to have at least 0.75 object-image ratio by cropping), Oxford-102 [31] and MS COCO datasets, and benchmarked against the GAN-INT-CLS [7] and GAWWN [14] using the Inception Score and Human Rank metrics.

StackGAN beats the former two networks on all 3 datasets and both evaluation metrics.

The major conclusions that can be drawn about this model are:

- The StackGAN does not simply memorize examples from the dataset as is proved by the authors in [StackGAN] by visually evaluating the extracted visual features from their generated images which are similar but significantly different from training images,
- The Stacked nature of the architecture along with the Conditioned Augmenting using the text information again and the input from Stage I are crucial to improved performance. The Stage I GAN with more upsampling layers fails to generate superior quality images. Conditional Augmentation stabilizes the training and leads to improved sample diversity in generated samples[StackGAN],

© 2019, IJCSE All Rights Reserved 

- The StackGAN model learns a smooth latent data manifold; the authors of [19] demonstrate that by using a fixed noise vector and linearly interpolated sentence embeddings.

**E. AttnGAN**

Most text-to-image GANs including those discussed so far convert the text descriptions into a global sentence vector as a whole. The authors of AttnGAN [20] argue that this omits important fine-grained information at the word level. The AttnGAN [20] introduces two novel mechanisms to produce high-resolution images: The Attentional Generative Network and the Deep Attentional Multimodal Similarity Model(DAMSM). The authors are the first to show that a layered conditional GAN can generate conditional data by automatically attending to the relevant words in the text descriptions.

**Attentional Generative Network**

This part of the AttnGAN architecture encodes the entire text description into one global vector representation which is used to generate a crude image in the first stage. It also encodes the individual words into vectors. It constructs a word-context vector for each image sub-region from the word features and image features from the previously hidden layer, which signifies the word that is most relevant to that sub-region. The model weights each word *i* with respect to sub-region *j* to signify how much attention the model should pay to *i* while drawing *j*

**Deep Attentional Multimodal Similarity Model**

The DAMSM consists of two neural networks: a text encoder, and an image encoder. The text encoder is a bi-directional LSTM [36]. This encoder provides a DxT matrix where D is the dimension of the word vector, and T the number of words. Each column in this matrix corresponds to the vector representation of the respective word. The final hidden states are concatenated to generate the global vector representation of the entire sentence.

The image encoder is a CNN which is built upon the inception v3 [37] model pre-trained on ImageNet[34]. The intermediate layers of this network learn the features of the image sub-regions while the later layers learn global features. Further, the authors compute an attention driven image text matching score which indicates the relatedness of an image and sentence based on the attention model.

**Evaluation**: Similar to previous text-to-image models, the AttnGAN was evaluated using the CUB-200[15] and COCO [33] datasets. Further, the authors used R-Precision for measuring how well the generated image captures the text description. The model also boasts great generalization ability, robustness to subtle changes in the input text and ability to generate images of non-realistic scenarios such as "a stop sign is floating on top of a lake" [20]. The AttnGAN achieves impressive results on both the CUB dataset as well as on the significantly more challenging COCO dataset by achieving inception scores of 4.36 and 25.89 respectively.

**F. ChatPainter**

The ChatPainter [21] architecture adds a dialogue box to iteratively refine the image and reports an improvement in the inception score over theStackGAN [19].The authors present the analogy to process followed by human sketch-artists which involves a continuous feedback loop in order to improve the quality of the sketch drawn.

**Architecture**:Similar to the StackGAN [19], the ChatPainter has two stages. In the first stage, a low-resolution 64x64 image is generated. Stage I is conditioned on the text encodings of the caption which are generated similar to the method used in GAN-INT-CLS [7]. For encoding the dialogue, two methods are used:

- Non-recurrent encoder: This encodes the entire dialogue with a pre-trained Skip-Thought encoder [39]
- Recurrent encoder: Skip-Thought vectors are generated for each turn of the dialogue and then encoded with a bi-directional LSTM [40] [41]

**Training details**: As introduced in GAN-INT-CLS [7] and further used in StackGAN [19] the authors use both 'real' pairs of matching images, caption and dialogue, and 'fake' pairs of the mismatched image, caption and dialogue. They use the Adam optimizer and train both stages of their model for 800 epochs. The initial learning rate which is 0.0002 is halved every 50 epochs. Stage I uses a mini-batch size of 384 while Stage II uses a mini-batch of 64. The recurrent version uses a 1024 hidden-dimensional RNN. Their implementation is based on PyTorch [42] and the authors used 4 NVIDIA Tesla P40 machines for training this model.

The model is evaluated on the MS COCO [33] dataset and the dialogues for the images are obtained from the VisDial dataset [43] using the Inception score metric. Both versions of the ChatPainter [21], which have additional conditioning on dialogue information, perform better than the StackGAN [19] which is conditioned only on the image caption. The recurrent version achieves a better Inception score than the non-recurrent version which the authors think is likely due to the inability of the Skip-Thought encoder used in the non-recurrent version of dealing with long sentences.

**G. StackGAN++**

In order to make the framework of Stack GAN version-1 [19] (original StackGAN) more general, Stack GAN version-2 [22] (StackGAN++) proposes an end-to-end network with a series of multi-scale image distributions. The network consists of multiple generators (Gs) and discriminators (Ds)

© 2019, IJCSE All Rights Reserved 

in a tree-like structure, where images of low-resolution and high-resolution are generated from different branches of the tree.

At each branch, the generator captures the image distribution (at a certain scale) and the discriminator estimates the probability that the sample came from training images rather than from the generator (at that scale). The generators and discriminators are trained in an alternating fashion. The motivation of the proposed StackGAN-v2 [22] is that, by modelling data distributions at multiple scales, if anyone of those model distributions shares supports with the real data distribution at that scale, the overlap could provide good gradient signal to expedite or stabilize training of the whole network at multiple scales.

**Modified GAN-CLS [23]**

Problems with GAN-CLS [7]: It can be proved that the global optimum of the objective function in GAN-CLS [7] algorithm is not the same as actual GANs. As a result, the generator is not able to generate samples which obey the same distribution with training data in GAN-CLS [7] algorithm theoretically.

But in practice, GAN-CLS [7] algorithm is able to synthesize the corresponding image for text description, the reason could be the distributions of P*d*(x) and P*d*(x) are similar.

GAN-CLS [7] algorithm with modified objective function [23] is observed to generate better images on various datasets including the Iris dataset and dataset of images of birds. In the case of the Iris dataset, the flower images generated are more plausible, with a precise number of petals, accurate shapes and better matching to the corresponding text description. GAN-CLS [7] also works efficiently on the bird dataset generating beaks, colours, shapes of birds more accurately using a modified version of the objective function. Defects in the modified version of GAN-CLS:Generation of shapeless (without a clear boundary) and less diverse images, sensitivity to the hyperparameters are some of the shortcomings of modified GAN-CLS [23].

The original GAN-CLS [7], as well as the modified version, is observed to give poor results in case of complex text description such as the position of the particular colour in a flower or the position of colours and posture of birds. The poor results can be attributed to the similarity between the two distributions- when text and image are matching when text and image are non-matching. To solve this problem, a simple experiment is conducted in which the original image is divided into 16 pieces and they are shuffled in order to make the two distributions dissimilar.

After the experiment, the modified version of GAN-CLS is still able to generate plausible results but original GAN-CLS algorithm stops generating images as it is observed to work efficiently only when the distribution of matching text and image, non-matching text and image are similar or same.

## III. Text-to-Face

Generating face images from text descriptions is possibly a much more complex problem than generic text to image synthesis problems due to the incredibly high number of features that may be used to describe a human face. As an illustration, consider the description of a flower: "A yellow flower with long thin petals that are concentrated around the centre". This is already a complicated problem as it leaves many features of the flower such as the petal length and shape open to interpretation by the GAN. Now, consider that a human face is being described. There are a significantly larger number of features that must be taken into accounts such as size, shape, and relative positioning of facial features like the eyes, the nose, the ears, the eyebrows, and so on. Further, there is a problem of counting where GAN architectures have been known to generate examples with more than the correct number of facial features for a single subject.

For these reasons, we believe that a hybrid model constructed from the combination of two or more of the Text-to-Image GAN architectures discussed in section 2 above could be particularly well suited to the task of Text-to-Face image synthesis. In particular, the GAN-INT-CLS [7] architecture was able to generalise well over various backgrounds and objects. The 2 GAWWN [14] models showed how the model could be taught to learn where to draw in addition to what to draw. The second approach mentioned in this paper for keypoint conditioned text-to-image synthesis could be very useful for Text-to-Face if the keypoints that are used for conditioning were facial landmarks.

Further, the inventors of the ChatPainter [21] architecture themselves draw an analogy of how their model draws an image in a manner similar to that of sketch artists, by continuously incorporating feedback from the user. This is a promising architecture based on the StackGAN [19], which could be strengthened by using the StackGAN++ [22] which outperforms the StackGAN [19] on multiple datasets. Most of all, we believe that the AttnGAN would provide the best results as it has the best inception score of all the Text-to-Image models discussed in this paper. Thus, we hope to incorporate the advantages of these architectures and construct a novel architecture to tackle the very specific task of text-to-face image synthesis in our future work.

## IV.CONCLUSION

In this paper, we have discussed what generative models are and which tasks they are helpful for. We discussed the most

© 2019, IJCSE All Rights Reserved 

popular generative models such as VAEs and RBMs and why GANs are believed to be superior to them. Further, we outlined the various problems that can be tackled using GANs. One such problem is that of Text-to-Image synthesis. We have described the architectures and the advantages and disadvantages of some of the most novel architectures that have been proposed for this task. Finally, we discussed how these architectures or a hybrid thereof may be particularly well suited and perform well for the task of Text-to-Face Image Synthesis.

## REFERENCES


[1] Ian Goodfellow and Yoshua Bengio and Aaron Courville, ``Deep Learning," MIT Press, 2016.

[2] Goodfellow, Ian J. et al. ``Generative Adversarial Nets," NIPS (2014).

[3] Bodnar, Cristian. "Text to Image Synthesis Using Generative Adversarial Networks." CoRR abs/1805.00676 (2018): n. pag.

[4] Gregor, Karol et al. "DRAW: A Recurrent Neural Network for Image Generation." ICML (2015).

[5] Kingma, Diederik P. and Max Welling. "Auto-Encoding Variational Bayes." CoRR abs/1312.6114 (2013): n. pag.

[6] Mansimov, Elman et al. "Generating Images from Captions with Attention." CoRR abs/1511.02793 (2015): n. pag.

[7] Reed, Scott E. et al. "Generative Adversarial Text to Image Synthesis." ICML (2016).

[8] Mirza, Mehdi and Simon Osindero. "Conditional Generative Adversarial Nets." CoRR abs/1411.1784 (2014): n. pag.

[9] Reed, Scott E. et al. "Learning Deep Representations of Fine-Grained Visual Descriptions." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 49-58.

[10] Kingma, Diederik P. and Jimmy Ba. "Adam: A Method for Stochastic Optimization." CoRR abs/1412.6980 (2014): n. pag.

[11] Bengio, Yoshua et al. "Better Mixing via Deep Representations." ICML (2013).

[12] Reed, S., Sohn, K., Zhang, Y., and Lee, H. "Learning to disentangle factors of variation with manifold interaction," (ICML 2014).

[13] Radford, Alec et al. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." CoRR abs/1511.06434 (2015): n. pag.

[14] Reed, Scott E. et al. "Learning What and Where to Draw." NIPS (2016).

[15] Wah, Catherine et al. "The Caltech-UCSD Birds-200-2011 Dataset." (2011).

[16] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In CVPR, June 2014.

[17] Dash, Ayushman et al. "TAC-GAN - Text Conditioned Auxiliary Classifier Generative Adversarial Network." CoRR abs/1703.06412 (2017): n. pag.

[18] Dong, Hao et al. "I2T2I: Learning text to image synthesis with textual data augmentation." 2017 IEEE International Conference on Image Processing (ICIP) (2017): 2015-2019.

[19] Zhang, Han et al. "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks." 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 5908-5916.

[20] Xu, Tao et al. "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks." CoRR abs/1711.10485 (2017): n. pag.

[21] Sharma, Shikhar et al. "ChatPainter: Improving Text to Image Generation using Dialogue." CoRR abs/1802.08216 (2018): n. pag.

[22] Zhang, Han et al. "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks." IEEE transactions on pattern analysis and machine intelligence (2018): n. pag.

[23] Gong, Fuzhou and Zigeng Xia. "Generate the corresponding Image from Text Description using Modified GAN-CLS Algorithm." CoRR abs/1806.11302 (2018): n. pag.

[24] X. Wu, K. Xu and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," in Tsinghua Science and Technology, vol. 22, no. 6, pp. 660-674, December 2017.doi: 10.23919/TST.2017.8195348

[25] Huang, He et al. "An Introduction to Image Synthesis with Generative Adversarial Nets." CoRR abs/1803.04469 (2018): n. pag.

[26] Goodfellow, Ian J.. "NIPS 2016 Tutorial: Generative Adversarial Networks." CoRR abs/1701.00160 (2016): n. pag.

[27] Ledig, Christian et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 105-114.

[28]Sønderby, Casper Kaae et al. "Amortised MAP Inference for Image Super-resolution." CoRR abs/1610.04490 (2016): n. pag.

[29] Wang, Yifan et al. "A Fully Progressive Approach to Single-Image Super-Resolution." CoRR abs/1804.02900 (2018): n. pag.

[30] Gadelha, Matheus et al. "3D Shape Induction from 2D Views of Multiple Objects." 2017 International Conference on 3D Vision (3DV) (2017): 402-411.

[31] Nilsback, Maria-Elena and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes." 2008 Sixth Indian Conference on Computer Vision, Graphics \& Image Processing (2008): 722-729.

[32] Yu, Fisher et al. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop." CoRR abs/1506.03365 (2015): n. pag.

[33] Lin, Tsung-Yi et al. "Microsoft COCO: Common Objects in Context." ECCV (2014).

[34] Deng, Jia et al. "ImageNet: A large-scale hierarchical image database." 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009): 248-255.

[35] Cho, Kyunghyun et al. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches." SSST@EMNLP (2014).

[36] Schuster, Mike and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE Trans. Signal Processing 45 (1997): 2673-2681.

[37] Szegedy, Christian et al. "Rethinking the Inception Architecture for Computer Vision." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 2818-2826.

[38]Cohn, Gabe. "AI-Art at Christie's sells for \$432,500". The New York Times, October 25, 2018.

[39] Kiros, Ryan et al. "Skip-Thought Vectors." NIPS (2015).

[40] Graves, Alex and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." Neural networks: the official journal of the International Neural Network Society 18 5-6 (2005): 602-10.

[41] Hochreiter, Sepp and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation 9 (1997): 1735-1780.

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

[43] Das, Abhishek et al. "Visual Dialog." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 1080-1089.


© 2019, IJCSE All Rights Reserved 