Text to Image Generation with Semantic-Spatial Aware GAN and BERT

Min Khant Soe
School of Engineering and and Technology
Asian Institute of Technology
Klong Luang, Pathum Thani 12120
st122277@ait.asia

Ayush Koirala
School of Engineering and Technology
Asian Institute of Technology
Klong Luang, Pathum Thani 12120
st122802@ait.asia

***Abstract*—The main objective of a text to image generation (T2I) model is to generate realistic images from the descriptions of text. To focus on the local semantics, we used Semantic-Spatial Aware GAN [1]; it is trained from beginning to end so that the text encoder may take use of more text information. As in the paper of semantic-Spatial Aware GAN [1], we used a novel Semantic-Spatial Aware Convolution Network. To direct the transformation spatially, it learns a mask map in a weakly-supervised approach that relies on the present text-image fusion process. In addition, to efficiently merge text and picture information, it learns semantic-adaptive transformations conditioned on text. Moreover, we trained text encoder together with image generator to acquire improved text representations for image generation. For efficient text encoding, Bidirectional Encoder Representations from Transformers (BERT) [22] is a good encoder and beat the state of art model in many natural language processing tasks. In this project, we fine tuned original SSA-GAN and also implemented the SSA-GAN with BERT as text encoder and trained on our custom dataset. we proved that using BERT as text encoder is a good alternative to bidirectional LSTM with the close model performance result in Inception Score (IS) and Frechet inception distance (FID). Moreover, we also proved that we can reduce the training time with BERT.**

I. Introduction

There has been much evolution in GAN [9, 10] in integrating realistic images with the diverse conditions. In the field of computer vision and Natural language processing, it has been challenging to generate fine high resolution images from the text description. It is challenging because of text to image transformation and the ability to maintain semantic consistency between the created image and the given text. The most recent T2I methods improve visual quality and resolution by stacking a series of generator discriminator pairs to create images ranging from coarse to fine. However, many generator-discriminator pairs lead to a high computation and also increase instability during training processes. SSA-GAN has one pair of generator-discriminators and it can be trained in end-to-end fashion. So, a pre-trained text encoder can be fine tuned to learn better text representation for generating images. The main element of the framework is the semantic-spatial Aware convolution network which consists of a Condition Batch Normalization (CBN) module called Semantic-Spatial Condition Batch Normalization (SSCBN), and a mask predictor, as shown in *Fig. 1*. SSCBN learns semantic-aware affine parameters based on the text feature vector it has learned. Based on the present text-image fusion process, the mask map is predicted as output of the last SSCBN block. This affine transformation effectively and deeply combines the text and image features, as well as encouraging the image features to be semantically coherent with the text. The residual block in SSA-GAN ensures that the text-insensitive regions of the computed picture characteristics remain unchanged.

In this project, we followed the SSA-GAN architecture and tried to improve it. First, we reduced the number of SSACN blocks for a lesser computational and memory usage for training. Next, we fine tuned the original to train on our custom dataset. Then, we replaced bidirectional LSTM with BERT for text encoding. As a result, we found that our model can be trained faster and got higher score in FID evaluation than the model with bidirectiona LSTM.

II. Related Work

As for related work to our project, Generative Adversarial Network (GAN) [17]is the most popular model for text to image generation tasks. Reed et al. [18] are the first researchers to use conditional GANs (cGANs) to create plausible images from text descriptions. The StackGAN structure is introduced in [18, 19] to improve the resolution of generated images by stacking multiple generators in order to generate images from coarse to fine. Each generator has its own adversarial training discriminator for training. Ming et al. [20] propose a one-stage structure with only one generator-discriminator pair for T2I generation to overcome the training difficulties in the stacked GANs structure. Their generator is made up of a series of UPBlocks that are designed to upsample image features in order to produce high-resolution images. To avoid the issues with the stacked structure, we use a framework (SSA-GAN) that uses a one-stage structure.

Moreover, cross-modal attention is used by AttnGAN [2] to compute a word-context vector for each sub-region of the image and concatenates it to the image feature maps for further text-image fusion. It also introduces the Deep Attentional Multimodal Similarity Model (DAMSM), which computes a

fine-grained loss for image generation by measuring image-text similarity at both the word and sentence levels. When compared to segmentation maps, text is much more abstract and has a larger gap with the image and it does not have precise spatial information. Hence, in this project, we focused on text to image.

III. Methodology

A. Text Encoder

For text encoder, we use bidirectional LSTM as in the paper. LSTM[15] networks are a type of recurrent neural network that learns the importance of order in sequence prediction problems. LSTM has three main gates which make it more efficient. First is forget gate which is in charge of selecting which information should be saved for calculating cell state and which should be discarded. Next is the input gate which modifies the cell's state and decides which data is critical and which is not. The input gate, like the forget gate, helps in finding important information and storing it in memory. The last gate is the output gate determines the next hidden state. Each training sequence is presented forward and backward in bidirectional LSTMs to independent recurrent networks. The output layer for both sequences is the same. Bidirectional LSTMs have complete information of every point in a sequence, including everything that comes before and after it. These advantages make LSTM more suitable for this project. As a text encoder, LSTM convert the given text description into a 256-dimensional sentence feature vector (LSTM's last hidden states) and 18-word features with a dimension of 256 (the hidden states on each step of the LSTM).

We also use the Deep Attentional Multimodal Similarity Model (DAMSM) loss which was introduced in AttnGAN [2], aims to optimize the similarity score between images and their text descriptions (ground truth). The original architecture of DAMSM is shown in *Fig 2* For training the generator, the DAMSM loss is a fine-grained word-region matching loss. for a batch of image-sentence pairings $(Q_i, D_i)_M^{i=1}$, the posterior probability of captions D_i matching image Q_i is determined as in equation (1).

$$P(D_i|Q_i) = \frac{exp(\gamma_3 R(Q_i, D_i))}{\sum_{i=1}^{M} exp(\gamma_3 R(Q_i, D_j))} \tag{1}$$

where γ_3 is an experimentally determined smoothing factor and M is the number of training pair. Only D_i matches the image Q_i in each batch of sentences; all other M-1 sentences are treated as mismatching descriptions. Then DAMSM loss can be computed as in equation (2).

$$L_{DAMSM} = -\sum_{i=1}^{M} LogP(D_i|Q_i) \tag{2}$$

Moreover, we replace bidirectional LSTM with Bidirectional Encoder Representations from Transformers (BERT) pretrained transformer. It uses self-attention to efficiently analyze long text and can understand the meaning of each word by looking at the entire text passage. The original transformer had both an encoder and a decoder, whereas BERT only has an encoder. BERT is a transformer-based model because it employs an encoder that is extremely close to the original encoder of the transformer.

B. Image Encoder

We used pre-trained inception v3 [6] model which is a 48-layer deep pre-trained convolutional neural network model. It's a network that's been trained on over a million images from the ImageNet [7] dataset. It's the third version of Google's Inception CNN model, which was first developed for the ImageNet Recognition Challenge. It has been demonstrated to achieve higher than 78.1 percent accuracy on the ImageNet dataset. The model is based on Szegedy, et aloriginal .'s publication, "Rethinking the Inception Architecture for Computer Vision." This network has an image input size of 299-by-299. The model extracts general features from input images in the first part and classifies them based on those features in the second part. During training, the model's feature extraction section is reused but classification portion is then re-trained using our dataset. The reason that we reused the feature extraction section is to train the model with fewer computational resources and training time.

C. Semantic-Spatial Aware Convolutional Network

In the core SSA-GAN we have, SSACN block as shown in *Fig. 3*. It takes the encoded text feature vector $\bar{e}$ and image feature maps from last SSACN block as input, and outputs the image feature maps which are further fused with the text features. Next, the input image feature map of the first SSACN block (no upsampling) are in shape of 4 × 4 × 512 which are achieved by projecting the noise vector z to visual domain using a fully-connected (FC) layer and then reshaping it. As the output from SSACN, the image feature maps have 256 × 256 resolution. Each SSACN block consists of an upsample block, a mask predictor, a Semantic-Spatial Condition Batch Normalization (SSCBN), and a residual block. Upsample is used for doubling the width and height of the image feature map by bilinear interpolation operation. The residual block is used to maintain the main contents of the image features to prevent text-irrelevant parts from being changed and the image information is overwhelmed by the text information.

1) Semantic-Spatial Condition Batch Normalization (SSCBN): SSCBN is done after adding the text and image information. As in equation (3), mi(h,w) does not only decide where to add the text information but also plays the role of weight to decide how much text information needs to be reinforced on the image feature map.

$$\tilde{x}_{nchw} = m_{i,(h,w)}(\gamma_c(\bar{e})\hat{x}_{nchw} + \beta_c(\bar{e})) \tag{3}$$

The modulation parameters λ and β are learned conditioned on the text information, and the predicted mask maps control the affine transformation spatially. Thus, the Semantic-Spatial CBN is realized in order to fuse the text and image features.

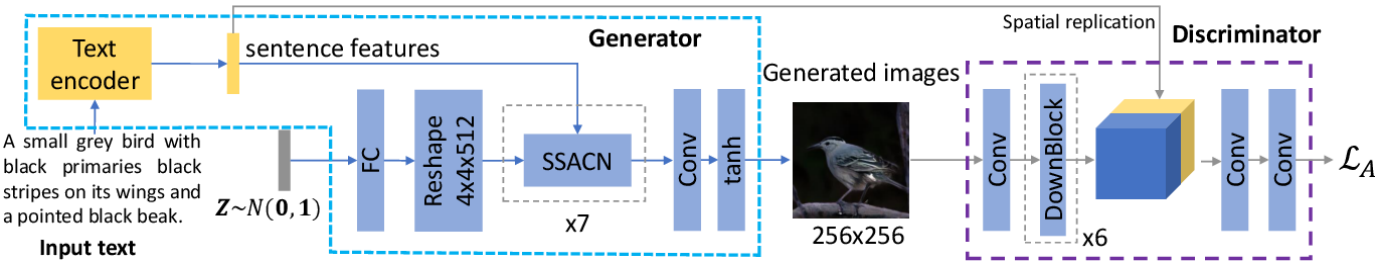

Fig. 1. Architecture of Original Semantic-Spatial Aware GAN

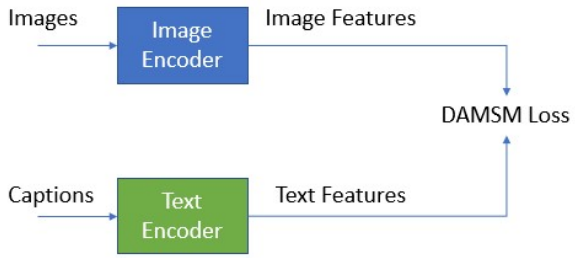

Fig. 2. Architectures of original DAMSM

D. Generator

Generator take text as an input and convert it to the encoded text feature vector. Then it pass text feature vector to the SSACN block. In the first SSACN block input image feature map which is obtained by projecting the noise vector to the visual domain using a fully-connected (FC) layer. It generates the image of 256*256. The adversarial loss of generator can be computed as in eq (4).

$$L_G^{adv} = -E_{x \sim P_G}[D(\hat{x}, s)] \tag{4}$$

where s is given text caption and $\hat{x}$ is generated image.

Loss function of generator is the combination of adversarial loss and a DAMSM loss as in eq (5).

$$L_G = L_G^{adv} + \lambda_{DA} L_{DAMSM} \tag{5}$$

where λ_{DA} is the weight of DAMSM loss

E. Discriminator

In SSA-GAN, one-way discriminator [16] is used. As shown in *Fig 1*, discriminator concatenates the features extracted from the generated image and the encoded text vector for computing the adversarial loss through two convolution layers. It works in combination with the Matching-Aware zero-centered Gradient Penalty (MA-GP) to help our generator to synthesize more realistic images with improved text-image semantic consistency. We add the widely used DAMSM [13] to our framework to improve the quality of generated images and the text-image consistency, as well as to help train the text encoder alongside the generator. The discriminator loss functions is composed of adversarial loss and MA-GP loss as in eq (6);

$$\begin{aligned} L_D^{adv} &= E_{x \sim P_{data}}[max(0, 1 - D(x, s))] \\ &+ \frac{1}{2} E_{x \sim P_G}[max(0, 1 + D(\hat{x}, s))] \\ &+ \frac{1}{2} E_{x \sim P_{data}}[max(0, 1 + D(x, \hat{s}))] \\ &+ \lambda_{MA} E_{x \sim P_{data}}[(|| \bigtriangledown_x D(x, s)||_2 \\ &+ || \bigtriangledown_s D(x, s)||_2)^p] \end{aligned} \tag{6}$$

where s is given text caption, $\hat{s}$ is a mismatched text caption. x corresponds to the real image of s. $\hat{s}$ is generated image. D() is the discriminator's judgement on whether or not the input image fits the input sentence. The hyper-parameters for MA-GP loss are λ_{MA} and p.

IV. Experiment

A. Datasets

We collected our dataset from Stanford car [8]. We selected 8 brand names of the cars, then we selected 4 models of each brand. From each model of each brand, we collected 20 images as training images and 5 images as validation images. Thus, we have the total of 640 training images and 160 validation images in training and validation datasets respectively. We renamed all the images in order starting from "00000" to "00639" and "00000" to "00159" for training images and validation images respectively. Then we wrote 5 captions for each images and created the caption folders for both training and validation. After that, we created the "$filenames\_car.pickle$" for both training and validation datasets to match the images and captions in training and validations process.

B. Implementation and Training

In this project, we trained our models with batch size of 16 on 4 GPUs provided by Asian Institute of Technology (AIT). We used text embedding dimensions of 256 in bidirectional LSTM as in original paper, so we have to change the BERT embedding dimension to 256 from 768 which is used in original pre-trained BERT. Thus, we have to change number of attention heads used in BERT to 8. As for optimizer, we use Adam optimizer[4] with β_1 of 0.0 and β_2 of 0.9. As in the original paper, the generator and discriminator have learning rates of 0.0001 and 0.0004 respectively, and p, γ_{MA}, and γ_{DA} hyper-parameters are set to 6, 2, and 0.1, respectively. We trained the model with bidirectional LSTM text encoder

(as in original paper) and our model with BERT text encoder for 150 and 50 epochs respectively on our car dataset The average training time taken for each epoch of the model with bidirectional LSTM text encoder is 2 min and 20 sec while our model only took 1 min and 52 sec in average. The reason we only trained 50 epochs for our model is that we analyze the loss plot from the model with bidirectional LSTM and found that 50 epochs is enough for model improvement.

C. Inference

We use the same batch size as in training. We evaluate our trained model on our validation set. Then, we used evaluation metrics such as Inception Scores and Frechet Inception Distance on our result to check the performance of our SSA-GAN and to compare the model performance of our SSA-GAN model and original SSA-GAN model fine tuned by us.

D. Evaluation Metrics

1) Inception Score: The Inception Score [10] is a measure for evaluating the quality of image generative models automatically (Salimans et al., 2016). The IS takes a list of images and returns the score as a single floating point value. The IS was shown to have a good correlation with human realism evaluation of produced pictures from the CIFAR-10 dataset. It is a popular metric for judging the image outputs of Generative Adversarial Networks (GANs). The score considers two factors at once: the diversity of the pictures and the distinct appearance of each image. If both of these statements are correct, the score will be high. The score will be low if any or both of the statements are untrue. The greater the score, the better. It means that your GAN may create a variety of diverse pictures. The lowest possible score is zero.

2) Frechet Inception Distance: The Fréchet Inception Distance (FID)[14] is a metric that measures the distance between two images distributions. Given the distance's importance as a ranking measure in data-driven generative modeling research, it appears critical that it be calculated using general, "vision-related" attributes. Frechet Distance is a measure of similarity between curves that takes the placement and ordering of points along the curves into consideration. The lower FID score represents that the quality of images generated by the generator is higher and similar to the real ones. FID is based on the feature vectors of images.

V. Result and Discussions

According to the loss plots of both models in *Fig 3 and 4*, discriminator is working well but generator cannot generate images very well to make that the discriminator cannot distinguish between the generated images and real images. In *Fig 5, 6, 7 and 8*, we can clearly see that both models can generate better result around at epoch 20; at 50 epochs the results is poorer. The generated result at epoch 20 will be more clear if we use 7 SSACN blocks. From this result, we can also conclude that SSA-GAN can perform better with less epochs; too many training epochs will lead the model to overfit and result in poor quality. In *Fig 9*, we showed the generated images by the SSA-GANs from the captions given by a user.

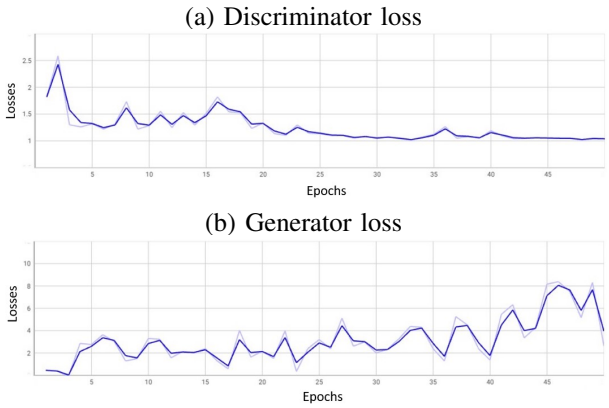

Fig. 3. Losses of SSA-GAN with Bidirectional LSTM for 50 epochs

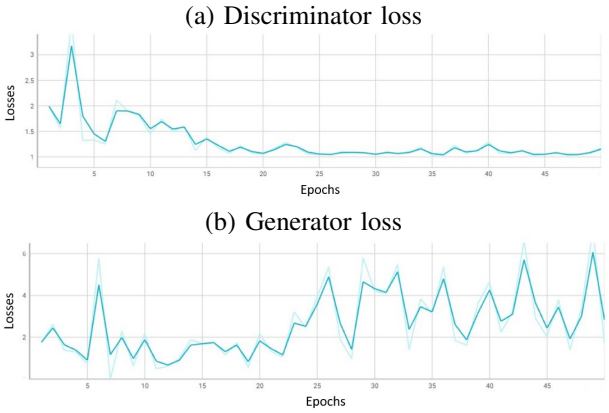

Fig. 4. Losses of SSA-GAN with BERT for 50 epochs

According to *Table 1*, SSA-GAN with BERT has higher score in FID while SSA-GAN with bidirectional LSTM has higher score in IS. Since FID measures similarity between two image sets (generated image set and validation image set) and our model gets higher score in FID, we can include that SSA-GAN generates better image with BERT. Although BERT does not noticeably beat bidirectional LSTM in GAN model performance, it boosts the training time.

We believe that with all the layer 7 layers of SSACN block as in original paper, we could generate the good resolution images. To improve the performance, we can dd more sample for each class of images and also adjust hyper-parameters by trials and errors method.

TABLE I
Performance of the Models

Methods	*Epochs*	*IS*	*FID*
SSA-GAN with bidirectional LSTM	20	**2.56**	310.0
	50	2.51	312.60
SSA-GAN with BERT	20	2.38	**295.96**
	50	1.50	340.03

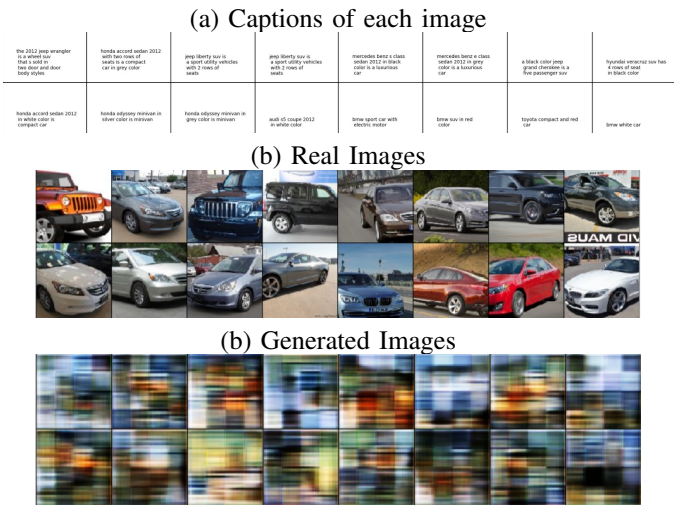

Fig. 5. Captions, Real and Generated Images of SSA-GAN with Bidirectional LSTM at epoch 20

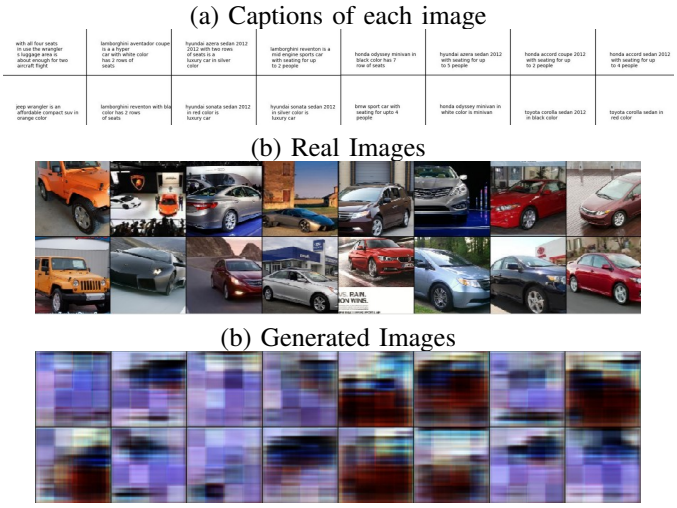

Fig. 6. Captions, Real and Generated Images of SSA-GAN with BERT at epoch 20

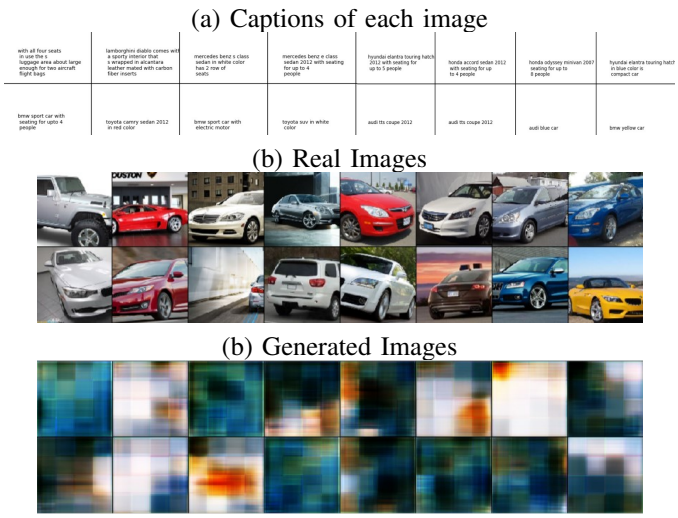

Fig. 7. Captions, Real and Generated Images of SSA-GAN with Bidirectional LSTM at epoch 50

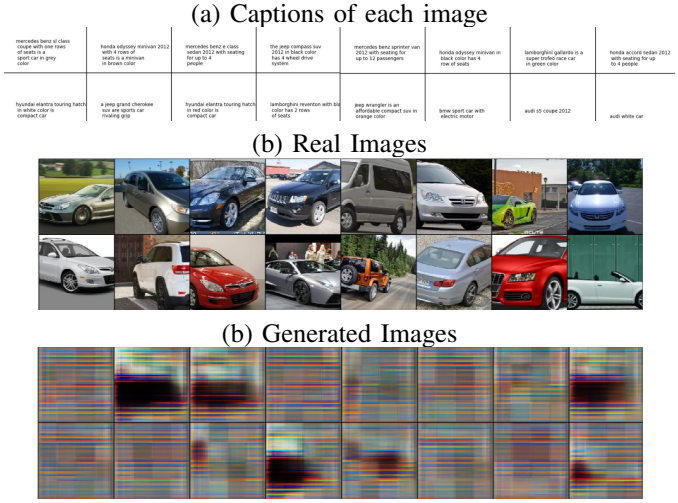

Fig. 8. Captions, Real and Generated Images of SSA-GAN with BERT at epoch 50

VI. Limitations and Challenges

We require a lot of memory for training, so it is difficult to train in our Puffer provided by AIT. So, We reduce the the numbers of SSACN blocks to 3 which makes model performance is poor. Since our model still takes quite a lot of computation space, we could not train the model every time after we implement the codes.

VII. Conclusion

In this project, we learnt a lot about text to image generation with SSA-GAN. Even though our results are not very impressive, we fine tuned the original SSA-GAN to train on our dataset and we also replaced bidirectional LSTM with BERT model. Next, we found that as text encoder in GAN, BERT beat LSTM in FID evaluation and boost the training speed. Therefore, BERT must be considered to use as text encoder in GAN for text to image generation tasks since it can be trained faster and have similar model performance with bidirectional LSTM.

Acknowledgment

First, we would like to thank the professor, Matthew N. Dailey for allowing us to do this project, and teaching us about GAN and basic knowledge of transformers in Recent Trend in Machine Learning class. Next, we want to thank Dr. Chaklam Silpasuwanchai for teaching us about various types of transformers in Natural Language Processing class. Lastly, we would also thank to Ms Alisa Kunapinun, our teacher assistant in Recend Trend in Machine Learning class for providing useful laboratory assignments.

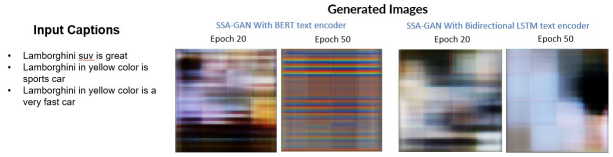

Fig. 9. Example of Generated Image conditioned on the given text descriptions

REFERENCES

[1] Hu, Kai, et al. "Text to image generation with semantic-spatial aware gan." arXiv preprint arXiv:2104.00567 (2021).

[2] Xu, Tao, et al. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[3] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In ICLR, 2017.

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

[5] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang,YiZhe Song, Bodo Rosenhahn, and Tao Xiang. Context-aware layout to image generation with enhanced object appearance. In CVPR, 2021.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, JonShlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In CVPR, pages 2818–2826,2016.

[7] Olga Russakovsky, Jia Deng, Hao Su, JonathanKrause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252,2015.

[8] https://www.kaggle.com/datasets/jutrera/stanford-car-dataset-by-classes-folder

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, BingXu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NeurIPS,2014.

[10] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[11] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2016.

[12] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang,Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Finegrained text to image generation with attentional generative adversarial networks. In CVPR, pages 1316–1324, 2018.

[13] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang,Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Finegrained text to image generation with attentional generative adversarial networks. In CVPR, pages 1316–1324, 2018.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter.Gans trained by a two time-scale update rule converge to a nash equilibrium. In NeuIPS, pages 6626–6637, 2017.

[15] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11):2673–2681, 1997.

[16] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Fei Wu, and Xiao-Yuan Jing. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. arXiv preprint arXiv:2008.05865, 2020. 1, 2, 3, 5, 6, 8, 10, 11

[17] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang,Yi-Zhe Song, Bodo Rosenhahn, and Tao Xiang. Context aware layout to image generation with enhanced object appearance. In CVPR, 2021

[18] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiao gang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stack gan: Text to photo-realistic image synthesis with stacked

[19] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In ICML, pages 1060–1069,2016.

[20] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiao gang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stack gan++: Realistic image synthesis with stacked generative adversarial networks. Transactions on pattern analysis and machine intelligence, 41(8):1947–1962, 2018.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In CVPR, pages 2337–2346, 2019.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" arXiv:1810.04805