
Synthesizing Images with Scene Graphs

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Creativity is always appreciated but the act of creation is rather demanding. If
2 a machine could faithfully comprehend the visual world, it should feature the
3 abilities to not only recognize images but also to produce them. Practically, image
4 synthesis from natural languages should achieve visual verisimilitude and semantic
5 consistency. However, many recent text-to-image systems may only be desirable
6 in limited domains such as flowers, food, and birds. If given more sophisticated
7 environments, they struggle to picture the word descriptions accurately. Our project
8 is a re-implementation of the paper *Image Generation from Scene Graphs* [7]. It
9 presents an approach to explicitly encoding relations between multiple entities and
10 integrates two adversarial discriminator networks to ensure that images respect
11 the true semantic relations. In this project, we experimented methods introduced
12 by this paper and validated our image generation model with the published work
13 quantitatively and qualitatively. Moreover, we explored various network config-
14 urations to enhance the effectiveness of scene graphs in reasoning about object
15 relations. We applied our model to a novel dataset and we will discuss some model
16 extensions for potential improvements.

17 1 Introduction

18 Generative models own huge promise in the future. If a machine could intelligently understand the
19 visual world, it may be convenient to customize images that cater to personal tastes and serve as
20 a source of inspiration for graphics designers and artists, as photographers do. In spite of recent
21 remarkable progress in image generation systems, semantic alignments between texts and images
22 remain challenging when the sentence structures are intricate. Our referenced paper [7] reports that
23 employing scene graphs to represent sentences overcomes some traditional limitations in portraying
24 texts with multiple objects. It applies a graph convolution network to extract semantic relations in
25 a scene graph, an object layout network to predict object bounding boxes and shapes, and finally
26 a cascaded refinement network to synthesize an image in a coarse-to-fine manner. In this project,
27 we adopted the model architecture proposed by the paper as our baseline framework and sought to
28 reinforce the advantages of scene graphs in explaining object relations. We focused on formulating
29 a variety of layer designs for the cascaded refinement network, and analyzed the contributions of
30 different activation functions to realizations of visually-consistent images.

31 We anticipate interactive image generation to be fascinating, however, implementation difficulties are
32 threefold. Firstly, we had to convert image annotations to scene graphs due to the unavailability of
33 datasets with all 3 ground-truth labels: scene graphs, object positions, and mask segmentation. Sec-
34 ondly, the complexity of real-life objects and predicates in the dataset demands training exhaustively
35 over all images and across a broad vocabulary. Hence, some simplifying assumptions were to be
36 considered to scope the data for a fair learning time. Lastly, it is hard to define evaluation metrics for
37 generative models and quantitatively measure how well the images comply with the graphs.

2 Related Work

2.1 Scene Graphs

Scene graph generation concerns the localization of objects and the detection of subject-predicate-object triples. As an abstraction of objects and their spatial or functional relations, a scene graph maintains rich semantic knowledge of an image by having object nodes and relation edges. This expressive structure can be consumed by a broad range of computer vision tasks, including image retrieval [8], image captioning [11], visual question answering, and image generation. Scene graphs bridge natural languages and image generation models, hence, they play a pivotal role for our project. Though the only accessible resource of human-annotated scene graphs is the Visual Genome dataset [9], diverse algorithms have been developed for sentence-graph translation [16] and image-graph prediction [9, 13, 18, 19]. We constructed scene graphs from two-dimensional object coordinates in the COCO dataset [10] as the principal inputs to the graph convolution network.

2.2 Cascaded Refinement Network

Low image resolution is analogous to nearsighted vision in that fine features are barely discernible. The cascaded refinement network (CRN) [3] strives to successively refine the feature maps by end-to-end convolutional networks and synthesize high-resolution images from pixel-wise layouts. It has shown success in visualizing street scenes trained with ground-truth segmentation and a reconstruction loss. We incorporated the CRN implementation as a tool to promote image resolution.

2.3 Generative Image models

Generative image models have been extensively exploited in deep learning and artificial intelligence. Apart from Variational Autoencoders and Autoregressive Models, the idea of Generative Adversarial Networks (GAN) [5, 14] has proven rewarding in generating images that appear superficially authentic to human observers. GAN trains two networks cooperatively: an image generator to fool the discriminator and a discriminator optimized to differentiate fabricated images from the genuine ones. We made use of two discriminating networks to criticize the inadequate model at train-time and motivate the generation of visually-reasonable images.

3 Methods

Our goal is to develop an image generation model conditioned on scene graphs that capture the subject-predicate-object triples and output images that preserve the graph descriptions. To begin with, scene graphs are scrutinized by the graph convolution network that transmits relational information along the graph edges and learns the embedding vectors for all object nodes. Progression from scene graphs to images is managed by the object layout network that predicts the segmentation masks and bounding boxes for all objects. After we project the object layouts onto the scene layouts, the cascaded refinement network follows to generate images and upgrade their spatial resolution. The model will be trained adversarially against two discriminators that encourage images to appear realistic and recognizable. Figure 1 provides an outline of the full generative model and participants of the pipeline are elaborated below with more details.

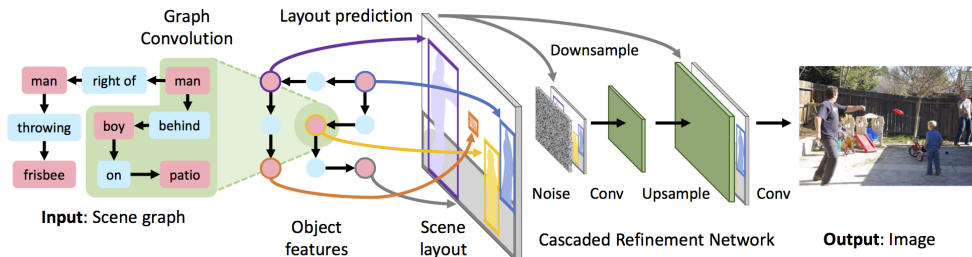


Figure 1: An overview of the image generation model.

3.1 Scene Graphs

Scene graphs provide a visually-grounded and interpretable representation of objects and their relations in an image. Provided an object domain \mathbb{O} and a relation domain \mathbb{R} , a scene graph G has a collection of nodes $o_i \in \mathbb{O}$ and directed edges $\langle o_i, r, o_j \rangle$ where o_i is a subject, $r \in \mathbb{R}$ is a predicate, and o_j is an object. Objects and predicates are discrete variables and they are embedded as dense features v_i, v_r, v_j as analogous to a word-to-vector layer in sentiment analysis. To scale the categories of predicates, only 6 geometric relations are taken into account. They are *left of*, *right of*, *above*, *below*, *inside*, and *surrounding*.

3.2 Graph Convolution Network

The graph convolution network contains a number of graph convolution layers, each of which propagates semantic information to the next layer with 3 functions. Given a subject-predicate-object triple $\langle v_i, v_r, v_j \rangle$, the predicate vector v_r is mapped to $v'_r = g_p(v_i, v_r, v_j)$. Nevertheless, the update is not equally evident for v_i and v_j in the light of that an object can be engaged in multiple relations and v'_i should retain dependencies on the adjacent neighbors that share a mutual relation with o_i . For example, the boy stays behind the man and the boy stands on the patio in the scene graph of Figure 1. For the graph edges that start with o_i and the graph edges that end with o_i , we consider the space of candidate vectors

$$V_i^s = \{g_s(v_i, v_r, v_j) : \langle o_i, r, o_j \rangle \text{ in } G\} \cup V_i^o = \{g_o(v_j, v_r, v_i) : \langle o_j, r, o_i \rangle \text{ in } G\}$$

and a pooling function $h(V_i^s \cup V_i^o)$ to be responsible for the next update v'_i . Empirically, g_p , g_s , and g_o are implemented with a multilayer fully-connected perceptron and h takes the average or the sum of all nominees in V_i^s and V_i^o . Iterations through the graph convolution layers learn relative positions from predicates and augment geometric information to the object vectors. We surveyed different numbers of graph convolution layers, activation functions, and pooling functions to process the scene graphs. Figure 2 shows a graph convolution layer on a scene graph with 3 nodes o_1, o_2 , and o_3 , and 2 edges $\langle o_1, r_1, o_2 \rangle$ and $\langle o_3, r_2, o_2 \rangle$.

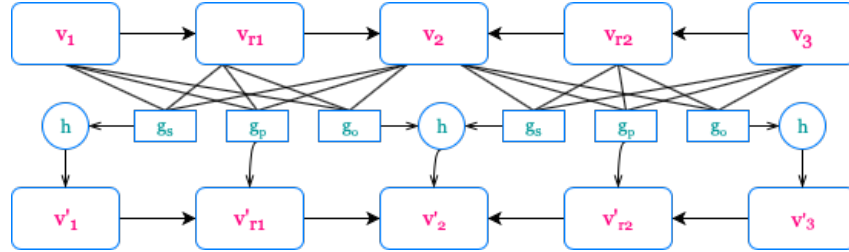


Figure 2: A computation graph of a graph convolution layer.

3.3 Object Layout Network

The object layout network functions as an intermediate between scene graphs and images. It sends a final object vector v_i from the graph convolution network to an internal box regression network and a mask regression network to estimate a bounding box $\hat{b}_i = (x_0, y_0, x_1, y_1)$ and a soft binary mask \hat{m}_i of size $M \times M$ for the object o_i . Afterwards, the element-wise product of v_i and \hat{m}_i is warped to the bounding box location \hat{b}_i by bi-linear interpolation [6] to construct an object layout. Aggregations of all individual object layouts yield a complete scene layout. Ground-truth bounding boxes b_i and masks m_i are leveraged to build the scene layout during train-time, while the network uses its predictions for novel scene graphs. Figure 3 illustrates the transformation from object vectors to a scene layout in the object layout network.

3.4 Cascaded Refinement Network

The cascaded refinement network is composed of a number of cascaded refinement modules, each of which collects the downsampled scene layout with average pooling and the upsampled output features from the preceding module with nearest-neighbor interpolation by a factor of 2. Inputs are

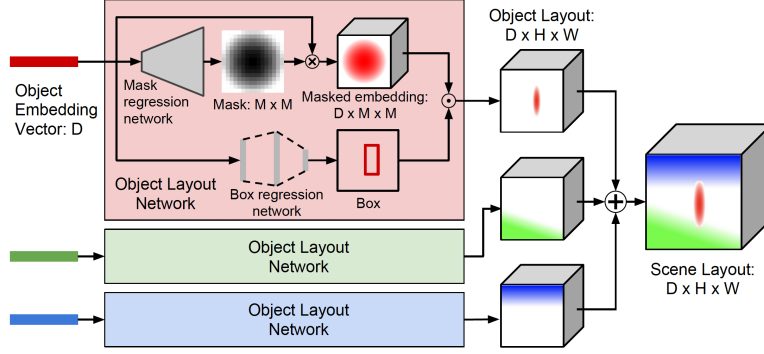


Figure 3: The workflow of the object layout network. $H \times W$ is the output resolution at which images are generated ($H = W = 64$ by default).

113 concatenated channel-wise and fed into a pair of convolution networks to double the spatial resolution.
 114 Gaussian noises are taken as the features to the first module and an image prediction with 3 channels
 115 is returned after the last module output is forwarded through 2 additional convolution networks.

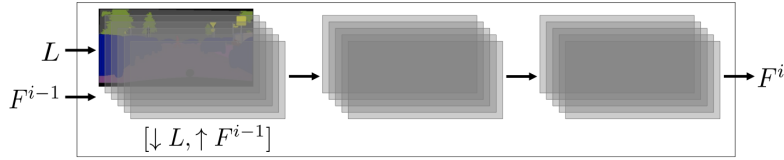


Figure 4: A single refinement module [3]. It receives a concatenation of the scene layout L (down-sampled) and the feature layer F^{i-1} (upsampled) as inputs, and produces the feature layer F^i .

116 3.5 Discriminators

117 The generative model will be trained adversarially against discriminator networks D_{img} and D_{obj} .
 118 The image discriminator D_{img} evaluates how realistic the images are by classifying image patches
 119 as either valid or fake, in the meantime, the object discriminator D_{obj} inspects the object crops and
 120 computes the truth scores. Furthermore, D_{obj} attempts to predict the categories that objects fall into
 121 with an auxiliary classifier and ameliorate the recognizability of objects. Both discriminators are
 122 primarily dependent on convolutional neural networks.

123 3.6 Loss functions

124 The model is trained with respect to a weighted sum of 6 loss functions: the box loss $L_{box} =$
 125 $\sum_{i=1}^n ||b_i - \hat{b}_i||_1$ penalizing the L_1 difference between ground-truth and predicted boxes, the mask
 126 loss $L_{mask} = BCE(m_i, \hat{m}_i)$ penalizing mask differences with binary cross-entropy, the pixel loss
 127 $L_{pix} = ||I - \hat{I}||_1$ penalizing the L_1 difference between ground-truth and predicted pixels, the image
 128 adversarial loss L_{img} , the object adversarial loss L_{obj} , and the auxiliary classifier loss L_{aux} imposed
 129 by two discriminators. We undertook an exploration of the significance of some loss functions to the
 130 final images.

131 3.7 Challenges

132 Implementation challenges firstly come with the comprehension of the model workflow. Despite
 133 that the code of our reference paper is publicly available, pixel cropping, re-scaling, and some other
 134 spatial techniques are beyond the scope of our expertise so we will not claim our originality on
 135 those components. However, we made considerable efforts in delivering the pipeline coherently
 136 and decomposed the implementation into a graph-to-layout module, a layout-to-image module, a

discriminator module, and a loss function module. Due to the generative nature of our model, there is a fair amount of work in fine-tuning the hyper-parameters of the network configuration such as the activation functions, the learning rates, the hidden sizes, and the convolution kernels. It would not be feasible to examine all dimensions for an optimal system, therefore, we paid the greatest attention to advance the cascaded refinement network. In addition, the incompleteness of perfect scene graph annotations or mask segmentation imposes some difficulties in the decision of datasets and it will be highlighted in the following Dataset section. Our last challenge should be the limited GPU resource which prevents us from training our implement with the same iteration scale as the referenced work. As a result, we need to "downgrade" the referenced work in order to make a direct comparison, which adds additional difficulties in evaluation part.

4 Experiments

4.1 Dataset

Although there is an appreciable repository of images online for vision-related research, it is challenging to find an ideal dataset for this particular task. In the original study, the Visual Genome dataset [9] and the COCO dataset [2] have been explored but either the gold scene graphs or the gold object masks are absent. Visual Genome provides scene graphs with a variety of objects and predicates, unfortunately, some scene graphs do not truthfully reflect the images and the dataset is deficient in mask segmentation. Training without masks is viable but it is likely to degrade the image quality by granting bounding boxes absolute dominance in the loss functions associated with the scene layout. The COCO dataset has bounding boxes and mask segmentation, it does not contain scene graphs though. Therefore, we recovered the geometric relations from the object boxes.

Firstly, we filtered out objects whose occurrence counts are below a certain threshold. In our case, we found that a cut from 4000 reduces the number of classes to 109. Secondly, we went through the bounding boxes to remove objects occupying less than 2% of the area in the complete image. Finally, we kept images whose object counts are bounded between 3 and 8, and obtained a training set of 38,734 images from a total of 118,287 images. The paper uses 2D coordinates to determine the category of predicates between each subject-object pair. Specifically, if one bounding box lies strictly within the other, the relation is inside or surrounding. Other than that, the relation is governed by the direction of one centroid to the other. As the number of predicates will grow in second order under the assumption that every two entities should be related, only one random object is selected for each subject and it guarantees the attendance of all objects in the scene graph while preventing predicate redundancy.

Apart from the above datasets considered by the original paper, we actively looked for other suitable substitutes. The Pascal VOC [4] dataset contains bounding boxes, object segmentation, as well as the action labels. The major distinction between Pascal VOC and COCO is the size of object categories and the number of images available. There are only 20 classes of labeled objects and 2,913 images in Pascal VOC. We expected a dataset with specialized objects would lead to faster convergence and a comparable performance. However, even with significantly more training epochs, the images remain blurred and objects can be hardly distinguished by eyes given insufficient training samples.

4.2 Evaluation and Results

We conditioned on inception scores to quantitatively measure the quality of our model. The inception score is a traditional and widely adopted evaluation technique for generative networks [15]. A pre-trained model is migrated to predict a conditional class probability for each generated image. Images that are classified with high confidence to one class over the other classes tend to have high quality scores as it is easy to assign them memberships. Though it is not a perfect benchmark, in most cases notably higher inception scores indicate better image generation models. We will use the same Inception v1 model by Google.

As the first step to assess the correctness of our implementation, we trained the model proposed by the referenced work [7] with the provided code and our own implementation under default network settings. Concretely, the activation function in the graph convolution network is ReLU, the activation function in the cascaded refinement network is LeakyReLU with a negative slope of -0.2 , and the learning rate is 2×10^{-4} . Since training generative models is computationally expensive, it is difficult

Scene Graph	Original Image ¹	Reduced Reference ²	Our Model ³	Reference Model ⁴
Inception Score: ⁵	13.02	2.19	3.31	4.72

¹ Ground truth image in test set.

² Reference model trained through 50 epochs.

³ Our baseline model trained through 50 epochs.

⁴ Original reference model trained 1 million iterations (32 batch size).

⁵ The Inception Score is calculated by Inception v1 model from Google.

Table 1: A comparison between results from our baseline model and the referenced model.

189 to replicate trials with one million iterations with a batch size of 32, which was done in the referenced
190 work. The estimated time for training an equivalent number of iterations with our limited GPU
191 resource is approximately 162 hours, which turns out to be infeasible within the project timeframe.
192 In order to expedite the speed and perform adequate analysis for various parameters, we decided to
193 train the model with 50 epochs (32000 iterations) and use a batch size of 50 for an optimal usage of
194 the GPU memory. The averaged training time lowers to 7.2 hours.¹

195 Table 1 presents some evaluation results on the test images for our baseline model and the referenced
196 model. The test set is a hold-out set consisting of 1,550 images from the COCO validation set, and
197 it was processed with the same filters applied to the training samples. As shown in Table 1, an
198 outstanding reduction in training time results in a significant decrease in image qualities. However,
199 the objective here is to verify if our model is comparable to the referenced model in generating images
200 that reflect geometric information from the scene graphs. It can be observed that images generated
201 from our baseline model manifest some blurred contours of the central objects in the scene graph.
202 Even though it may not be as optimal as the original referenced model, it is significantly better than
203 the blurred images generated by the referenced model with reduced iterations. Therefore, within
204 limited training iterations, our model is capable of extracting some important features in the scene
205 graph. In some cases, our model could achieve similar performances to the original referenced model.
206 For example, in the image of a person surfing in the sea (third row in the table), the person and the
207 beaches can be detected in both models. This is also reflected by the inception scores: our model and
208 the original reference model both have higher inception scores than the iteration-reduced model.

209 The advantage of our model in comparison to the referenced model within a limited number of
210 iterations (fast training scenario) is likely to originate from the differences in the training methods.
211 The training losses can be partitioned into two disjoint components: the losses from object layout
212 predictions, and the losses from image generation. For our model, we train the two parts in a rather
213 decoupled way. The second half of the model is trained based on the true object layout. Therefore, a
214 deviated layout prediction would not tamper the training process for the second half. However, for
215 the referenced model, it trains the model as a whole for the most of the iterations. We believe training
216 the model as a whole would introduce instability.

217 After confirming the feasibility of our baseline model, the other question is how to extend the model
218 to achieve better performance. It appeared to us that the cascaded refinement network is a crucial step
219 in image generation, since it is the component that transforms the scene layout to an actual image.
220 We noticed that the activation function used in this network is LeakyReLU, which is a modification
221 of most widely used ReLU activation. Since this is not well justified in the studies [7, 3], we decided
222 to use different activations and examine the corresponding effects. We selected plain ReLU, PReLU,
223 CELU, and Softplus as candidate activation functions. These functions have similar behaviors: they
224 are less or equal to zero when $x \leq 0$ and increase almost linearly with x when $x > 0$. It turns out that
225 the Softplus is obviously not capable for this generating task due to a significantly lower inception
226 score (mean: 2.55), and a visually-unpleasant image.

227 Table 2 shows some image outputs from the models with different activation functions used in the
228 cascaded refinement network. Based on observations, it is decided that the performance of CELU is
229 not as favorable as other three because of its more blurred boundaries, which can also be confirmed
230 by the evaluation of inception scores. However, it remains difficult to visually distinguish between
231 the performances of the other different activation functions. One possible approach is to conduct a
232 survey among a group of reviews to judge the quality of two generated images, but it is not achievable
233 within the timeframe. Thus we used the inception score as an objective criterion and determined that
234 PReLU and ReLU have achieved best performances.

235 During our experiments, we also examined the images generated from ground-truth bounding boxes
236 and masks, or from the "cheat" mode – neglecting the effects of scene layout predictions and
237 only focusing on the CRN's performance. As shown in Table 3, the performances experienced a
238 significantly boost given the real layouts (The "standard solution" of our Object Layout Network).
239 Previously, the truck (second row) and the bicycle (fourth row) are located around the center. With
240 additional information captured in the scene layout, the locations of objects are adjusted to match
241 the original image. However, such adjustment may not be necessary. Taking the bicycle image as an
242 example, the plain PReLU model is able to generate a figure in the image. Moreover, since the effect
243 of layout predictions is eliminated, "cheating" is more sensitive to the performances of CRN and

¹The code including our baseline model and the reduced reference model is attached in the submission.

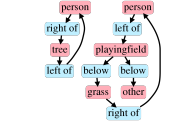
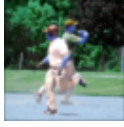
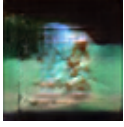
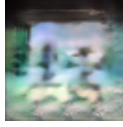
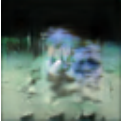

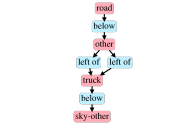








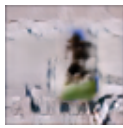


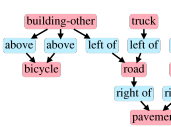
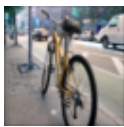
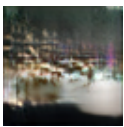
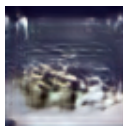

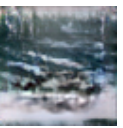


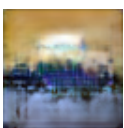

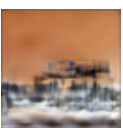
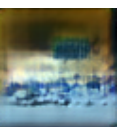

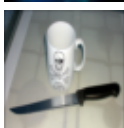
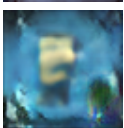
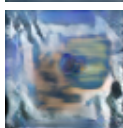

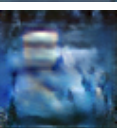


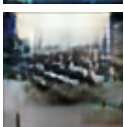
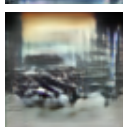
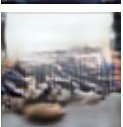
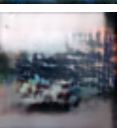
Scene Graph	Original Image	LeakyReLU	ReLU	PReLU	CELU
					
					
					
					
					
					
					
Inception Score Mean:	13.02	3.31	3.64	3.75	3.16

Table 2: A comparison between models with different activation functions.

discriminators, which in turn leads to better indications of the effects of different activation functions. It is observed that the images from "cheat" mode of PReLU have more detailed features and better qualities, both in colors and texture. It is in agreement with the conclusion drawn earlier from the inception scores that PReLU is the most desirable activation function for CRN.

5 Future Work

The task of graph-based image generation faces some empirical impediments. Firstly, the Visual Genome dataset has no ground-truth masks and some defective scene graphs, on the other hand, the COCO dataset can only yield scene graphs with spatial relations. Seeing that data reliability is an influential factor of the model performance, heuristic-oriented approaches to accomplish scene graph annotations from images may be applicable. Secondly, evaluation metrics such as inception scores allow us to quantitatively measure the overall image quality, but it may not be informative in terms of how strongly the images agree with the scene graphs. Relation Scores and Mean Opinion Relation Scores [17, 1], the fraction of true spatial relations achieved by the scene layout prediction and the fraction of true spatial relations perceived in the image prediction, may collaborate to show supportive evidence in the degree of graph-image compliance. Moreover, the model may be extended with a recurrent architecture and refines image generation by incrementally expanding scene graphs [12]. As it is intuitive for an artist to sketch the underlying layout with a few objects and incrementally append more objects when painting some real-world scenes, stacking multiple graph-to-image pipelines in

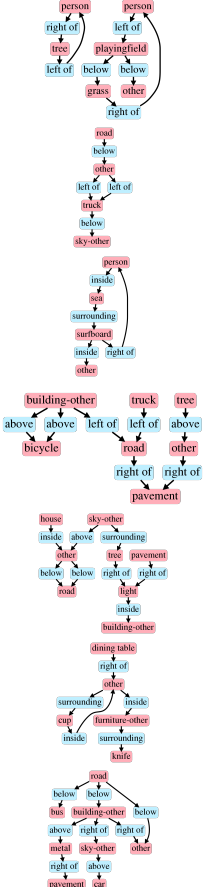
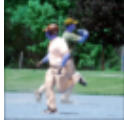
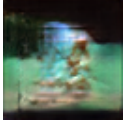
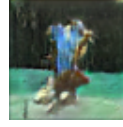
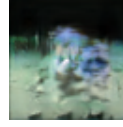

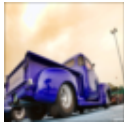
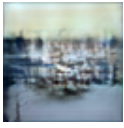
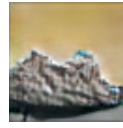
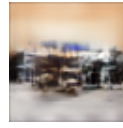
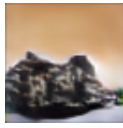


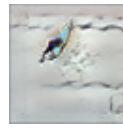
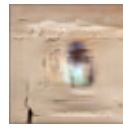

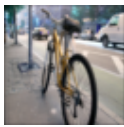
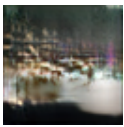




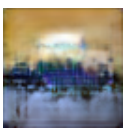
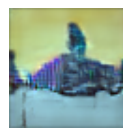
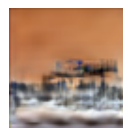
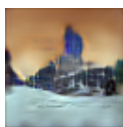
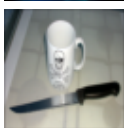
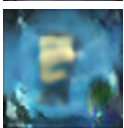




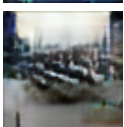
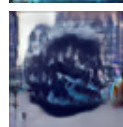
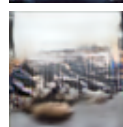
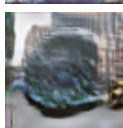
Scene Graph	Original Image	LeakyReLU	LeakyReLU*	PReLU	PReLU*
					
					
					
					
					
					
					
Inception Score Mean:	13.02	3.31	4.85	3.75	5.31

Table 3: LeakReLU v.s. PReLU in "cheat" mode*

Figure 1 may be exceptionally effective. During the first pass, the graph convolution network receives a partial scene graph. The next iteration sees an expanded scene graph and absorbs previous scene layouts and image predictions into its inputs to preserve visual contexts. The complete scene graph is released in the final pass. However, we will leave this implementation scheme to future work considering the timeframe of this project.

6 Conclusion

Our project is a re-implementation attempt of an image generation model. It adopts a graph convolution network to process the scene graphs, an object layout network to predict segmentation masks and bounding boxes, a cascaded refinement network to transform the scene layouts to realistic images, and finally two discriminators in cooperation to adversarially train the networks. Building generative models is a fruitful experience for all group members. Through literature reviews and hands-on experiments, we accomplished our initial objective to implement the model and we further explored various approaches to extend the baseline model. And it is worth mentioning that we also made several simple but effective improvements beyond the original work. Our decoupled training method achieves significantly better results in a limited iteration training scenario and we also concluded out that the PReLU is probably a better alternative for the leakyReLU in the cascaded refinement network. Due to the limitation of computational resources, we were not able to directly compete the referenced work in their training scale in this project, but we plan to leave it as future work.

References

- [1] David R. Bull. Chapter 10 - measuring and managing picture quality. In David R. Bull, editor, *Communicating Pictures*, pages 317 – 360. Academic Press, Oxford, 2014.
- [2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.
- [3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017.
- [4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.
- [5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [6] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.
- [7] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018.
- [8] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.
- [9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [11] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2017.
- [12] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743*, 2019.
- [13] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in neural information processing systems*, pages 2171–2180, 2017.
- [14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.
- [16] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015.
- [17] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. Heuristics for image generation from scene graphs. 2019.

- 327 [18] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative
328 message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*
329 *Recognition*, pages 5410–5419, 2017.
- 330 [19] Michael Ying Yang, Wentong Liao, Hanno Ackermann, and Bodo Rosenhahn. On support
331 relations and semantic scene graphs. *ISPRS journal of photogrammetry and remote sensing*,
332 131:15–25, 2017.