西南科技大学

Southwest University of Science and Technology

Subject: Academic Report

Topic: Image Generation From Scene Graphs

Southwest University of Science and Technology

Department of Computer Science and Technology

Student Name: **MD AYNUL ISLAM**

Chinese Name: 叶子

Student ID: **4420190030**

Batch: **Undergraduate 2019**

Contents

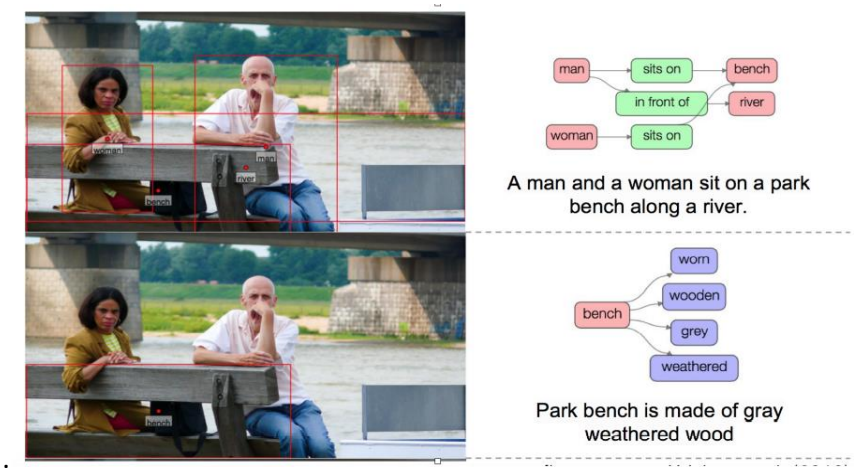
1 Abstract

Image understanding by computer is advancing exponentially these days due to the phenomenal success of deep learning, but there is still much work left for the computers to reach human level perception. Image classification (sometimes with localization) is one of the standard task, but this is far from the image understanding. The other tasks such as image caption generation or visual question answering have also reached to practical level of quality, but these are still far from the complete image understanding. Image caption generation, which is a task that generates a summary sentence from an image, cannot fully describe rich scenery in an image. In order to fully understand an image, we need to know; what objects are in the image, what are the characteristics of objects, and how these objects interact to each other.

2 Introduction

A scene graph is a data structure commonly used to represent hierarchical relationships between transformations applied to a set of objects in a three-dimensional scene. It finds applications in a variety of acceleration and rendering algorithms. A scene graph could also be used to organize visual attributes, bounding volumes, and animations as a hierarchy in a collection of objects. In the most general form, any scene related information that can be organized in a hierarchical fashion can be stored in a scene graph. It also provides a convenient way of representing logical groups of objects formed using their spatial positions or attributes. In this topic, we will outline the fundamental properties of scene graphs, look at some of the implementation aspects and consider a few applications.

Generating realistic images is a key task in computer vision research. Recently, a series of methods were presented for creating realistic-looking images of objects and faces. Despite this impressive progress, a key challenge remains: how can one control the content of images at multiple levels to generate images that have specific desired composition and attributes. “Controlling content can be particularly challenging when generating visual scenes that contain multiple interacting objects. One natural way of describing such scenes is via the structure of a Scene Graph (SG), which contains a set of objects as nodes and their attributes and relations as edges.”[1]



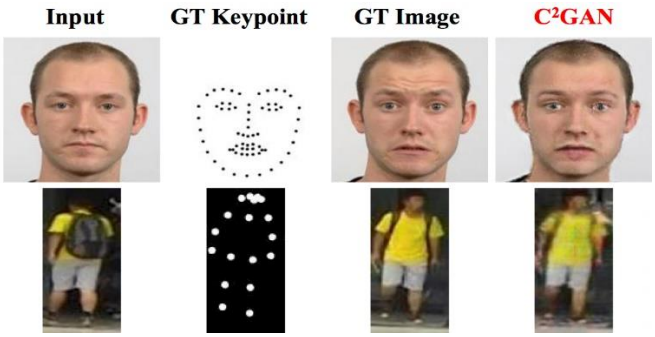
In here , The initial goal was to build a system to generate a scene graph from an arbitrary daily life image though we did not reach this level due to unexpected amount of time spent for environmental setup. Scene graph can be derived if we integrate objects, attributes of objects, and relations between objects. Part of scene graph is shown at the right side of Figure 1. The whole scene graph for the image in figure 1 is illustrated on figure 2. Scene graph is structured and more machine understandable than sentence descriptions or captions. It is actually shown to be more effective for image retrieval.[2]

3 Background and Related Work

In this Background and related work I describe some of related work and the background of the study provides context to the information that we are discussing in our paper. In this background of our study includes a review of the existing literature on the area of our research, leading up to the topic. Once we have discussed the contribution of other researchers in the field, we can identify gaps in understanding, that is, areas that have not been addressed in these studies. You can then explain how our study will address these gaps and how it will contribute to the existing knowledge in the field.

3.1 Image Generation

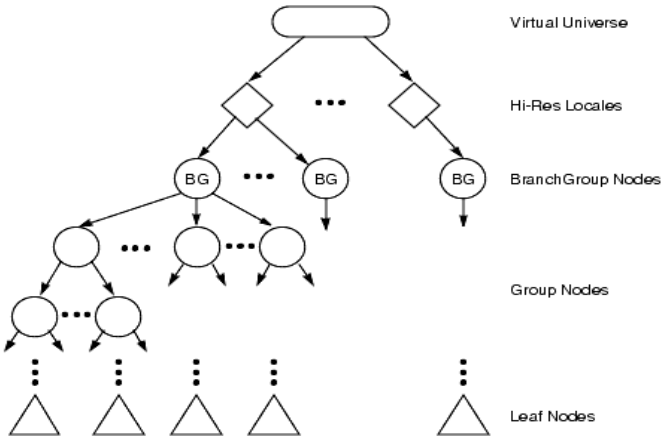
Earlier work on image generation used autoregressive networks to model pixel conditional distributions. Recently, GANs and VAEs emerged as models of choice for this task. Specifically, generation techniques based on GANs were proposed for generating sharper, more diverse and better realistic images in a series of works.



Click here to find the Image Source

3.2 Scene Graph

A scene graph is a general data structure commonly used by vector-based graphics editing applications and modern computer games, which arranges the logical and often spatial representation of a graphical scene. It is a collection of nodes in a graph or tree structure.

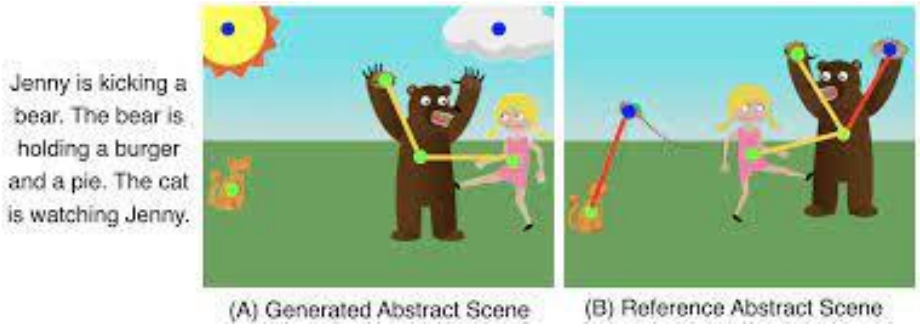


Click here to Find The Image Source

This representation is easier to understand for computers than sentence descriptions. For example, it is shown that scene graph helps computer to retrieve images more efficiently than sentence description. Aiming for better image retrieval, sense graph generator from image descriptions. However, to our knowledge, no previous work tries to generate scene graph directly from images.

3.3 Conditional Image Synthesis

Multiple works have explored approaches for generating images with a given desired content. Conditioning inputs may include class labels source images model interventions and text. Other studies focused on image manipulation using language descriptions while disentangling the semantics of both input images and text descriptions.



generate images conditioned on sentences and key-points using both GANs and multiscale auto regressive models in addition to generating images they also predict locations of unobserved key points using a separate generator and discriminator operating on key point location.[3]

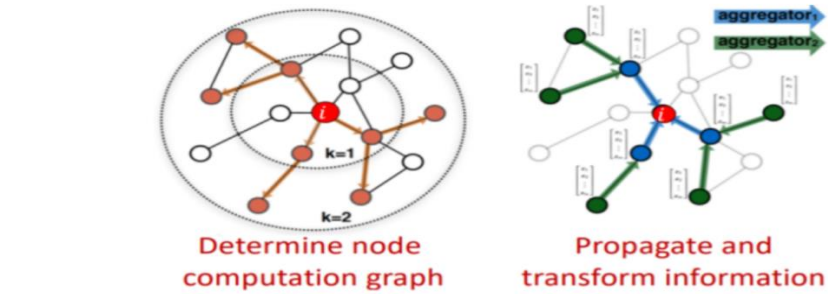
3.4 Scene-graph-to-image Generation

First to propose an end-to-end method for generating images from scene graphs. However, as we note above, the current SG-to-image models show degraded performance on complex SGs with many objects. To mitigate this, the authors in have utilized stronger supervision in the form of a coarse grid, where attributes of location and size are specified for each object. The focus of our work is to alleviate this difficulty by directly modeling some of the invariances in SG representation.

3.5 Deep Learning on Graphs

In graph theory, we implement the concept of Node Embedding. It means mapping nodes to a d-dimensional embedding space (low dimensional space rather than the actual dimension of the graph), so that similar nodes in the graph are embedded close to each other. We have to map nodes so that

similarity in the embedding space approximates similarity in the network.



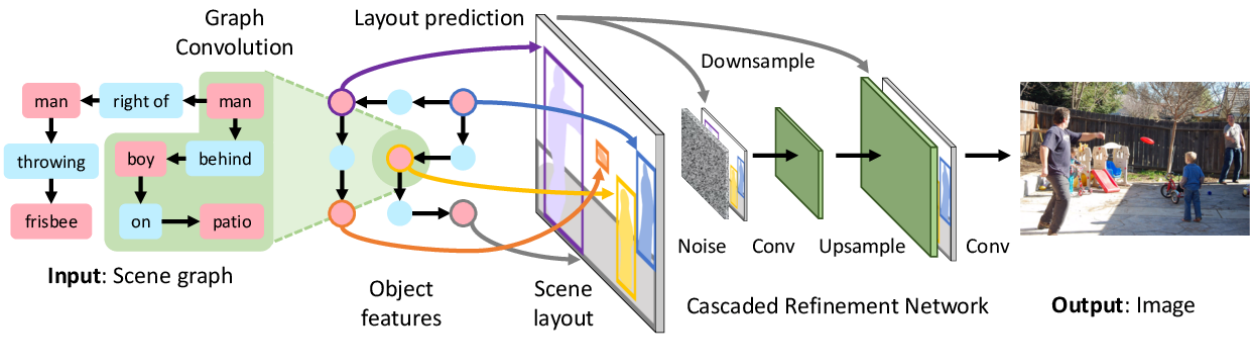
Neighborhood exploration and information sharing | Source

Neural Networks are presented in grey boxes. They require aggregations to be order-invariant, like sum, average, maximum, because they are permutation-invariant functions. This property enables the aggregations to be performed.[4]

4 Method

In this report , Our goal is to develop a model which takes as input a scene graph describing objects and their relationships, and which generates a realistic image corresponding to the graph. The primary challenges are threefold: first, we must develop a method for processing the graph-structured input; second, we must ensure that the generated images respect the objects and relationships specified by the graph; third, we must ensure that the synthesized images are realistic.

Here we can see ,converted scene graphs to images with an image generation network f, which inputs a scene graph G and noise z and outputs an image $\hat{I}$=f(G,z).The scene graph G is processed by a graph convolution network which gives embedding vectors for each object; each layer of graph convolution mixes information along edges of the graph. We respect the objects and relationships from G by using the object embedding vectors from the graph convolution network to predict bounding boxes and segmentation masks for each object, these are combined to form a scene layout, shown in the center of Figure 2, which acts as an intermediate between the graph and the image domain s. The output image $\hat{I}$ is generated from the layout using a cascaded refinement network (CRN).[5]



Overview of image generation network f for generating images from scene graphs. The input to the model is a scene graph specifying objects and relationships; it is processed with a graph convolution network which passes information along edges to compute embedding vectors for all objects. These vectors are used to predict bounding boxes and segmentation masks for objects, which are combined to form a scene layout. The layout is converted to an image using a cascaded refinement network (CRN). The model is trained adversarial against a pair of discriminator networks. During training the model observes ground-truth object bounding boxes and (optionally) segmentation masks, but these are predicted by the model at test-time.

4.1 Scene Graphs

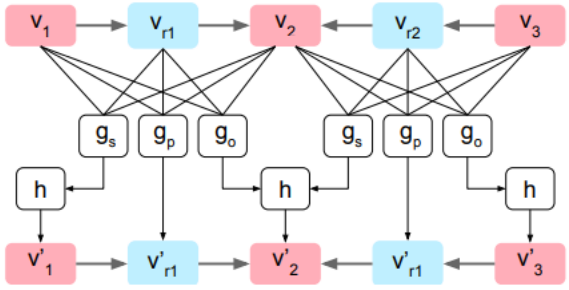
The input to our model is a scene graph describing objects and relationships between objects. Given a set of object categories C and a set of relationship categories R, a scene graph is a tuple (O, E) where O = {o1, . . . , on} is a set of objects with each oi ∈ C, and E ⊆ O × R × O is a set of directed edges of the form (oi, r, oj) where oi, oj ∈ O and r ∈ R. [5]

4.2 Image Generation from Scene Graphs

The input to our model is a scene graph describing objects and relationships between objects. Given a set of object categories C and a set of relationship categories R, a scene graph is a tuple(O,E)where O={o1,...,on}is a set of objects with each oi∈ C, and E⊆O× R ×O is a set of directed edges of the form(oi,r,oj) where oi,oj∈O and r∈R.As a first stage of processing, we use a learned embed-ding layer to convert each node and edge of the graph from a categorical label to a dense vector, analogous to the embedding layer typically used in neural language models.[6]

4.3 Graph Convolution Network

In order to process scene graphs in an end-to-end manner, we need a neural network module which can operate natively on graphs. To this end we use a graph convolution network composed of several graph convolution layers. The Graph Convolution Network (GCN) is composed of several graph convolution layers, and can operate natively on graphs. GCN takes an input graph and computes new vectors for each node and edge. Each graph convolution layer propagates information along edges of the graph. The same function is applied to all graph edges, which ensures that a single convolution layer can work with arbitrary shaped graphs.



To compute the output vectors v'r for edges we simply set v'r=gp(vi,vr,vj). Updating object vectors is more complex, since an object may participate in many relationships; as such the output vector v'I for an object oi should depend on all vectors vj for objects to which oi is connected via graph edges, as well as the vectors vr for those edges. To this end, for each edge starting at oi we use gs to compute a candidate vector, collecting all such candidates in the set Vsi; we similarly use go to compute a set of candidate vectors V oi for all edges terminating at oi. Concretely, [5]

Vsi={gs(vi,vr,vj) : (oi,r,oj)∈E}(1)

Voi={go(vj,vr,vi) : (oj,r,oi)∈E}(2)

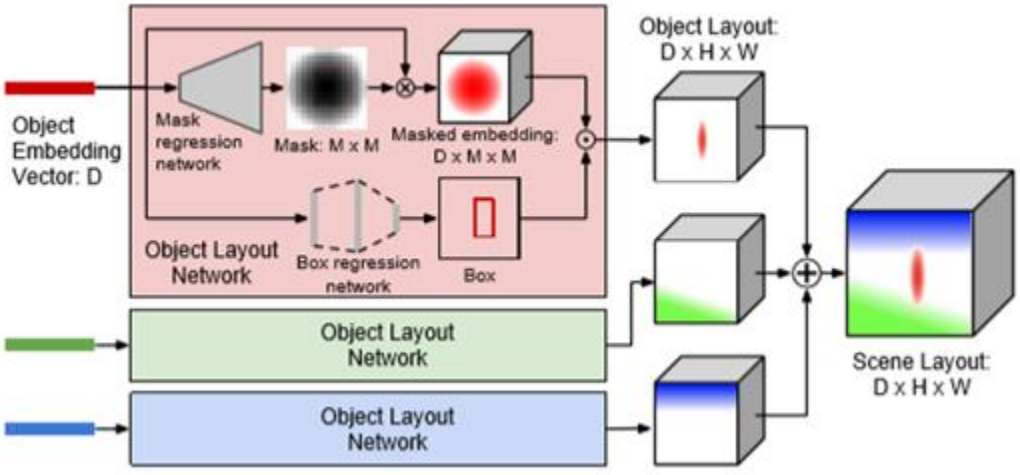
The output vector for v'I for object oi is then computed as v'i=h(VsiUVoi) where h is a symmetric function which pools an input set of vectors to a single output vector.

4.4 Scene Layout

Processing the input scene graph with a series of graph convolution layers gives an embedding vector for each object which aggregates information across all objects and relationships in the graph. The object layout network receives an embedding vector vi of shape D for object oi and passes it to a mask regression network to predict a soft binary mask ˆ mi of shape M×M and a box regression network to predict a bounding box ˆ bi= (x0,y0,x1,y1). The mask regression network consists of several transpose convolutions terminating in a sigmoid non linearity so that elements of the mask lies in the range (0,1); the box regression network is a MLP. During training we use ground-truth bounding boxes bi to compute the scene layout; at test-time we instead use predicted bounding boxes ˆbi.[7]

4.5 Cascade Refinement Network

Given a scene layout, the Cascade Refinement Network (CRN) is responsible for generating an image which respects the object positions in the scene layout. The CRN consists of a series of convolutional refinement modules. The spatial resolution doubles between modules; which ensures that image generation is happening in a coarse-to-fine manner. The scene layout is first down sampled to the module input resolution and then fed to the module along with the previous module's output. Both these inputs are concatenated channel wise and then passed to a pairof3×3convolution layers. This output is up sampled using nearest-neighbor interpolation and then passed to the next module. The output from the last module is finally passed to 2 convolution layers to produce the output image.



As we can see, they move from the graph domain to the image domain by computing a scene layout. The embedding vector for each object is passed to an object layout network which predicts a layout for the object; summing all object layouts gives the scene layout. Internally the object layout network predicts a soft binary segmentation mask and a bounding box for the object; these are combined with the embedding vector using bilinear interpolation to produce the object layout.

4.6 Image Quality Enhancement

Image enhancement aims to improve image quality in terms of colors, brightness, and contrasts. Earlier methods are mainly based on histogram equalization and gamma correction. Although these methods are simple and fast, their performance are limited by the fact that individual pixels are enhanced without consideration to contextual information.

Here also focus on improving the quality of the images generated by the baseline model. By default, the images are generated at a 64x64 resolution. In experiments that recurrent network and modifications to the loss terms by itself lead to improvements in the image quality. However, we further seek to enhance the generated images by drawing on insights drawn from StackGAN. The authors if StackGAN proposed a two-stage coarse-to-fine generation framework that generated an image conditioned on a text embedding. They proposed a conditioning augmentation, which augmented the size of the training data and lead to a smoother manifold of the GANs.[8]

5 Conclusion

A scene graph could also be used to organize visual attributes, bounding volumes, and animations as a hierarchy in a collection of objects. In the most general form, any scene related information that can be organized in a hierarchical fashion can be stored in a scene graph. It also provides a convenient way of representing logical groups of objects formed using their spatial positions or attributes. In this topic, we will outline the fundamental properties of scene graphs, look at some of the implementation aspects and consider a few applications. Image understanding by computer is advancing exponentially these days due to the phenomenal success of deep learning, but there is still much work left for the computers to reach human level perception.

6 Reference

[1] "Kinematics (Advanced Methods in Computer Graphics) Part 1." http://what-when-how.com/advanced-methods-in-computer-graphics/kinematics-advanced-methods-in-computer-graphics-part-1/ (accessed Dec. 16, 2021).

[2] J. Johnson *et al.*, "Image Retrieval Using Scene Graphs." pp. 3668–3678, 2015.

[3] S. Mehta and T. Marwah, "Incremental Image Generation using Scene Graphs."

[4] "Graph Neural Network and Some of GNN Applications: Everything You Need to Know - neptune.ai." https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications (accessed Dec. 17, 2021).

[5] J. Johnson *et al.*, "2015_CVPR_[SG definition]_Image Retrieval using Scene Graphs," *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pp. 3668–3678, 2015.

[6] G. Mittal, S. Agrawal, A. Agarwal, S. Mehta, and T. Marwah, "Interactive image generation using scene graphs," *Deep Gener. Model. Highly Struct. Data, DGS@ICLR 2019 Work.*, 2019.

[7] J. Johnson, A. Gupta, and L. Fei-Fei, "Image Generation from Scene Graphs," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1219–1228, 2018, doi: 10.1109/CVPR.2018.00133.

[8] T. Vu, C. V. Nguyen, T. X. Pham, T. M. Luu, and C. D. Yoo, "Fast and efficient image quality enhancement via desubpixel convolutional neural networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11133 LNCS, pp. 243–259, 2019, doi: 10.1007/978-3-030-11021-5_16.