Image Generation from Text using Scene Graphs

Presented by Divya K Raman
Computer Vision Lab, IIT Madras

2 Aug 2019

Schedule

- Introduction
- Literature Survey
- Preliminaries
- Image Generation from Scene Graphs - Paper in detail
  - Image Generation Network
  - Scene Graphs
  - Graph Convolution Network
  - Scene Layout
  - Cascaded Refinement Network
  - Discriminators
  - Training - Loss terms
  - Results
- Conclusion
- References

Introduction

- Computers need to be able generate images in order to understand the visual world better.
- Recent progress combines RNNs and GANs.
- Complex sentences containing multiple objects - hard task
- Recent methods use scene graphs for image captioning, synthesis
- Applications - photo-editing, computer aided design, augmented reality
- Generative image modelling - variational autoencoders (probabilistic graphical models method), PixelRNN (Autoregressive method), GANs

Literature Survey

- Reed et al, ICML 2016 - generate images from text using a GAN
- Reed et al, NIPS 2016 - generate images conditioned on sentences and keypoints using GANs
- Reed et al, ICML 2017 - generate images conditioned on sentences and keypoints using multiscale autoregressive models; in addition to generating images they also predict locations of unobserved keypoints using a separate generator and discriminator operating on keypoint locations.
- Chen and Koltun, ICCV 2017 - generate high-resolution images of street scenes from ground-truth semantic segmentation using a cascaded refinement network (CRN) trained with a perceptual feature reconstruction loss
- Chang et al - have investigated text to 3D scene generation
- Zhang et al, ICCV 2017 - multistage generation, higher resolution images generated - proposed StackGANs - SOTA before scene graph method
- Current SOTA: uses scene graphs to generate images from text

Preliminaries

- Generative Image Models
  - GANs - jointly learn a generator for synthesizing images and a discriminator classifying images as real or fake
  - VAEs - use variational inference to jointly learn an encoder and decoder mapping between images and latent codes
  - Autoregressive approaches - model likelihoods by conditioning each pixel on all previous pixels
- Conditional Image Synthesis - condition generation on additional input like category labels

Preliminaries

- Scene Graphs
  - Scenes = directed graphs, nodes = objects, edges = relationships between objects
  - Can be used for image retrieval, image captioning
  - Visual Genome dataset - human annotated scene graphs
- Deep Learning on Graphs
  - Learn embeddings on graphs
  - Graph Neural Networks (GNN) - which generalize recursive neural networks to operate on arbitrary graphs

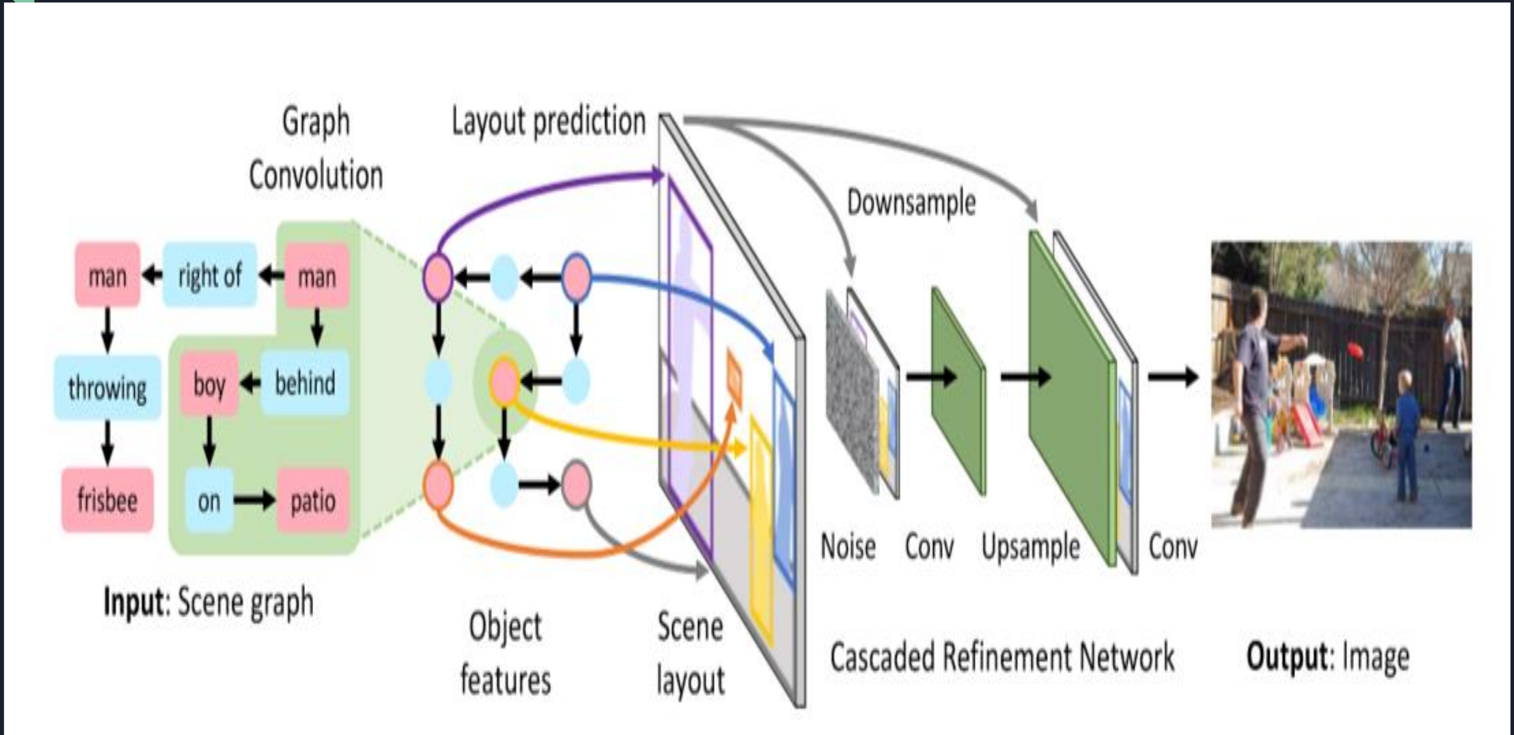
Image Generation from Scene Graphs: Justin Johnson et al, CVPR 2018

- Model: input - scene graph, output - realistic image
- Primary challenges addressed
  - a method for processing the graph-structured input
  - ensure that the generated images respect the objects and relationships specified by the graph
  - ensure that the synthesized images are realistic
- $I = f(G,z)$; f - image generation network; G - scene graph; z - noise; I - output image
- G -> Graph convolution network -> embedding vectors for each object
- Each layer of graph convolution mixes information along edges of the graph.

Image Generation Network

- Embedding vectors -> bounding boxes and segmentation masks for each object -> combine to get scene layout(intermediate between the graph and the image domains)
- Each module of cascaded refinement network (CRN) processes the layout at increasing spatial scales to generate the image I
- 2 discriminators – D_img and D_obj: encourage the image to appear realistic and to contain realistic, recognizable objects.

Image Generation Network



Scene Graphs

- Describes objects and relationships between objects
- C - set of object categories and R - set of object categories; 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.
- First stage of processing: use a learned embedding layer to convert each node and edge of the graph from a categorical label to a dense vector

Graph Convolution Network

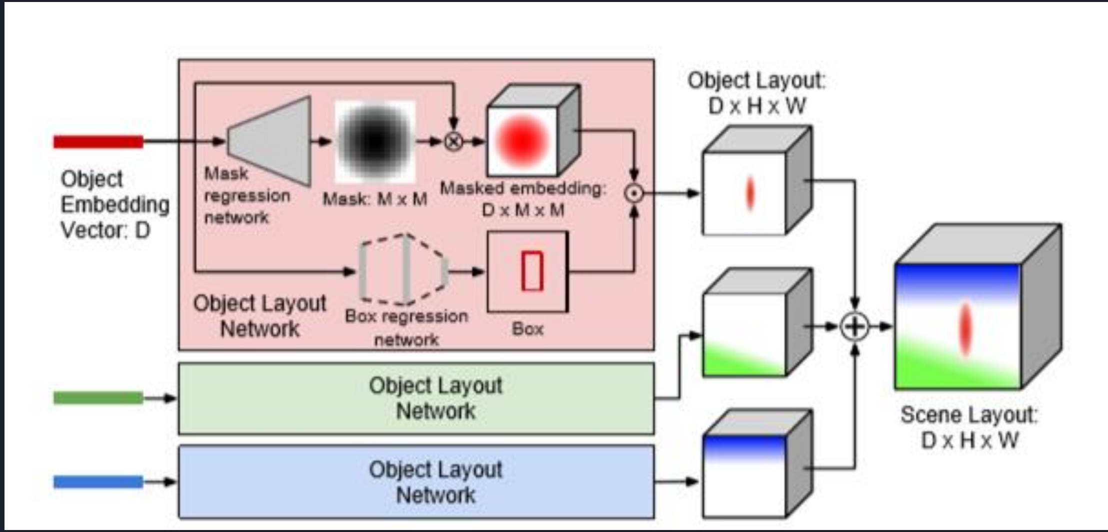
- Given an input graph with vectors of dimension D_in at each node and edge, each graph convolution layer computes new vectors of dimension D_out for each node and edge.
- Output vectors - function of a neighborhood of their corresponding inputs
- Each graph convolution layer propagates information along edges of the graph.
- A graph convolution layer applies the same function to all edges of the graph, allowing a single layer to operate on graphs of arbitrary shape

Scene Layout

- Object embedding vectors -> scene layout which gives the coarse 2D structure of the image to generate.
- Object layout network - computes the scene layout by predicting a segmentation mask and bounding box for each object
- Object layout network
  - input: embedding vector v_i of shape D for object o_i
  - Mask regression network: output - soft binary mask m_i of shape $M \times M$
  - Box regression network: Output - bounding box $b_i = (x0,y0,x1,y1)$.
- Masked embedding of shape $D \times M \times M$ = elementwise multiplication of embedding vector v_i and mask m_i -> warped to the position of bounding boxes using bilinear interpolation - gives object layout
- Scene layout - sum of object layouts

Scene layout

- Training - use ground-truth bounding boxes to compute the scene layout
- Test-time - use predicted bounding boxes .



Cascaded Refinement Network

- CRN - series of convolutional refinement modules
- Spatial resolution doubling between modules
- Generation to proceed in a coarse-to-fine manner
- Input to each module: Scene layout (downsampled to the input resolution of the module) and the output from the previous module concatenated channel-wise -> pair of 3 x 3 convolution layers
- Output: upsampled using nearest-neighbor interpolation before being passed to the next module.
- First module input: Gaussian noise
- Output from the last module ->two final convolution layers -> output image.

Discriminators

- 2 discriminators: D_img and D_obj
- D_img: The patch-based image discriminator
  - ensures that the overall appearance of generated images is realistic
  - it classifies a regularly spaced, overlapping set of image patches as real or fake
  - implemented as a fully convolutional network
- D_obj: object discriminator
  - ensures that each object in the image appears realistic
  - its input are the pixels of an object, cropped and rescaled to a fixed size using bilinear interpolation
  - also ensures that each object is recognizable using an auxiliary classifier which predicts the object's category
  - both D_obj and f attempt to maximize the probability that D_obj correctly classifies objects.

Training - Loss terms

- Jointly train generator network and both discriminators
- 6 loss terms
  - Box loss: Penalizes the L1 difference between ground-truth and predicted boxes
  - Mask loss: Penalizes differences between groundtruth and predicted masks with pixelwise cross-entropy
  - Pixel loss: Penalizes the L1 difference between ground-truth generated images
  - Image adversarial loss: Patch based image discriminator loss
  - Object adversarial loss: Object discriminator loss
  - Auxiliary classifier loss: ensures that each generated object can be classified by D_obj

Results

Results


car on street
line on street
sky above street
bus on street
line on street
sky above street
car on street
bus on street
line on street
sky above street
car on street
bus on street
line on street
sky above street
kite in sky
car on street
bus on street
line on street
sky above street
kite in sky
car below kite
car on street
bus on street
line on street
sky above street
building behind street
car on street
bus on street
line on street
sky above street
building behind street
window on building
sky above grass
zebra standing on grass
sky above grass
sheep standing on grass
sky above grass
sheep standing on grass
sheep' by sheep
sky above grass
sheep standing on grass
sheep' by sheep
tree behind sheep
sky above grass
sheep standing on grass
tree behind sheep
sheep' by sheep
ocean by tree
sky above grass
sheep standing on grass
tree behind sheep
sheep' by sheep
ocean by tree
boat in ocean
sky above grass
sheep standing on grass
tree behind sheep
sheep' by sheep
ocean by tree
boat on grass

Conclusion

- Scene Graph method performs far better than previous SOTA StackGAN
- End to end method for generating images from scene graphs
- Scene Graphs can be used in image captioning, image retrieval too
- Generating images from structured scene graphs rather than unstructured text allows the model to reason explicitly about objects and relationships, and generate complex images with many recognizable objects.

References

1. https://sites.google.com/view/cvpr2018-recognition-tutorial/home
2. **Johnson, J., Gupta, A., & Fei-Fei, L. (2018). Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1219-1228).**
3. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5907-5915).
4. S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In NIPS, 2016.
5. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text-to-image synthesis. In ICML, 2016.
6. S. E. Reed, A. van den Oord, N. Kalchbrenner, S. G´omez, Z. Wang, D. Belov, and N. de Freitas. Parallel multiscale autoregressive density estimation. In ICML, 2017.