Explainable AI: Interpreting, Explaining and Visualizing Deep Learning: Part II

Sean Bin Yang
seany@cs.aau.dk

DPW, Computer Science
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

Contents

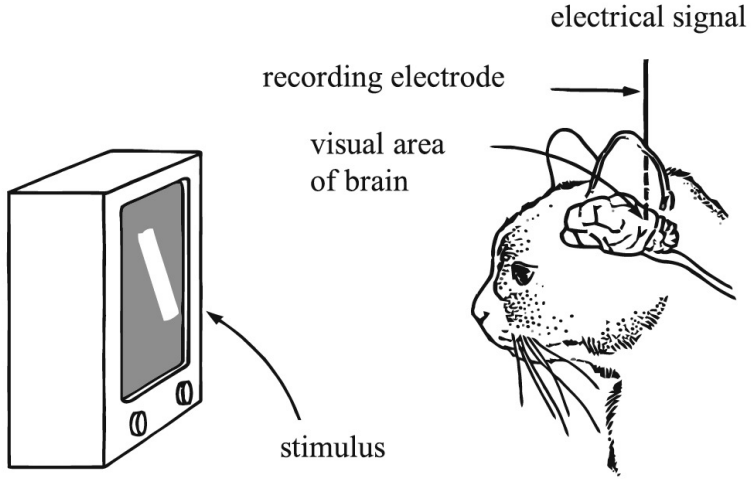
Section1: Neural Network Understanding via Feature Visualization: A survey

Anh Nguyen, Jason Yosinski, and Jeff Clune

Introduction

Human brain understanding

- What each neuron codes for?
- What information can activate the neuron?



Introduction

Neuron in Machine Learning

- Similarly, in machine learning (ML), visually inspecting the preferred stimuli of a unit can shed more light into what the neuron is doing.

Introduction

Neuron in Machine Learning

- Similarly, in machine learning (ML), visually inspecting the preferred stimuli of a unit can shed more light into what the neuron is doing.
- Intuitive Approaches
  - Possible method is finding such preferred inputs from existing, large image collection.
    - It requires testing each neuron on a large image set.
    - In such a dataset, many informative images that would activate the unit may not exist because the image space is vast and neural behaviors can be complex.
    - It is often ambiguous which visual features in an image are causing the neuron to fire.
  - Top 9 highest activating images for a unit.
    - Reflects only one among many types of features preferred by a unit.

Synthesize Visual Stimuli and Activation Maximization

Synthesize the visual stimuli from scratch

- Given a strong prior.
- More control over the types and contents of images to synthesize.

Synthesize Visual Stimuli and Activation Maximization

Synthesize the visual stimuli from scratch

- Given a strong prior.
- More control over the types and contents of images to synthesize.

Activation Maximization (AM)

- ***Definition:*** Finding an image $\mathbf{x}$ that maximizes the activation $a_i^l(\theta, \mathbf{x})$of a neuron indexed i in a given layer l of the classifier network can be formulated as an optimization problem:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \left(a_i^l(\theta, \mathbf{x}) \right) \tag{1}$$

where θ the parameters of a classifier that maps an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ onto a probability distribution over the output classes.

Synthesize Visual Stimuli and Activation Maximization

Examples

- Activating a single neuron and group neurons.

Synthesize Visual Stimuli and Activation Maximization

Examples

- Activating a single neuron and group neurons.


Optimization

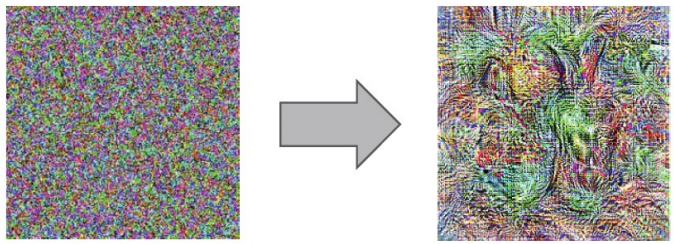
- AM is a non-convex optimization problem for which one can attempt to find a local minimum via gradient-based or non-gradient methods. In this case, a simple approach is to perform gradient ascent with an update rule such as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon_1 \frac{\partial a(\theta, \mathbf{x}_t)}{\partial \mathbf{x}_t} \tag{2}$$

Synthesize Visual Stimuli and Activation Maximization

Optimization

- Note that this gradient ascent process is similar to the gradient descent process used to train neural networks via back-propagation, except that here we are optimizing the network input instead of the network parameters , which are frozen.
- Optimization will be stopped when the neural activation has reached a desired threshold or a certain number of steps has passed.
- However, in practice, synthesizing an image from scratch to maximize the activation alone (i.e. an unconstrained optimization problem) often yields uninterpretable images.



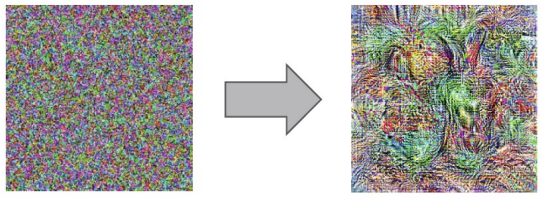
(a) Random initialization

(b) Synthesized rubbish example

Activation Maximization via Hand-Designed Priors

Human recognizable?

- Challenges
  - Examples like these images are not human recognizable.
  - The fact is that the network responds strongly to such images is intriguing and has strong implications for security.
  - If we can not interpret the images, it limits out ability to understand what the unit's purpose is.
- Solutions
  - To make it recognizable, the potential ways is to constrain the search to be within a distribution of images that we can interpret. That can be accomplished by incorporating natural image priors into the objective function.



(a) Random initialization

(b) Synthesized rubbish example

Activation Maximization via Hand-Designed Priors

Activation Maximization under Priors

- The priors often incorporated into the AM formulation as a *regularization* term $R(x)$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} (a(\mathbf{x}) - R(\mathbf{x})) \tag{3}$$

- To encourage the smoothness in AM images, $R : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}$ may compute the total variation across images. In each update, we follow the gradients to:
  - maximize the neural activation.
  - minimize the total variation loss.

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon_1 \frac{\partial a(\mathbf{x}_t)}{\partial \mathbf{x}_t} - \epsilon_2 \frac{\partial R(\mathbf{x}_t)}{\partial \mathbf{x}_t} \tag{4}$$

Activation Maximization via Hand-Designed Priors

Activation Maximization under Priors

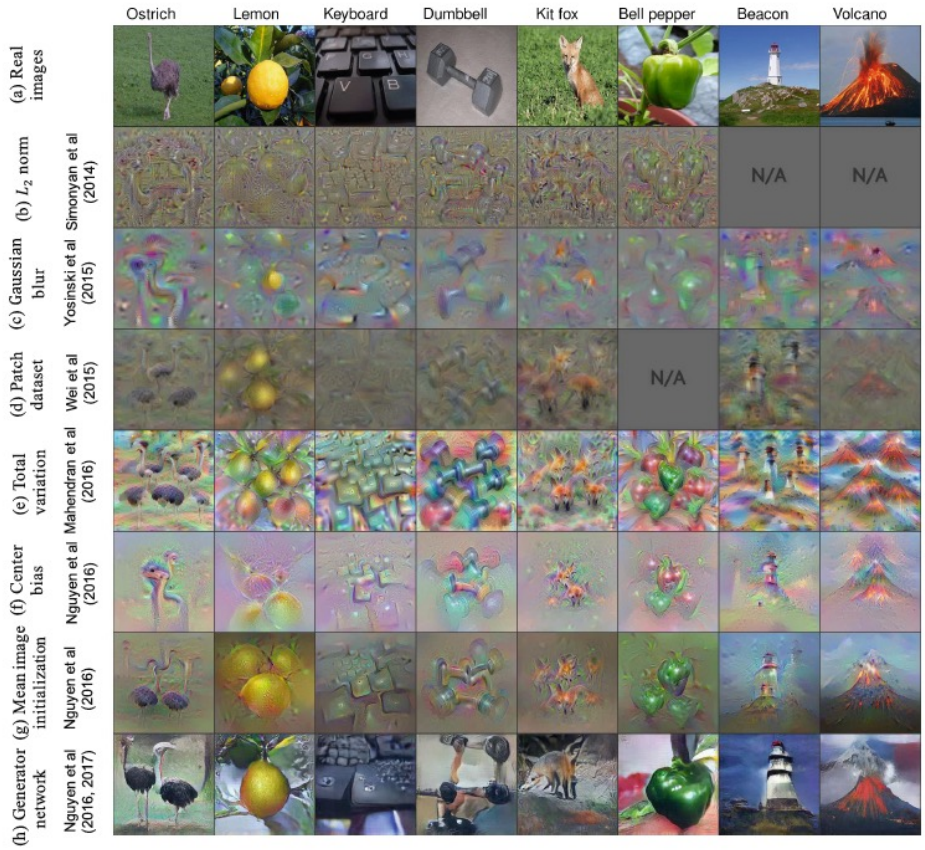
- Note that, we can not always compute the analytical gradient $\partial R(\mathbf{x}_t)/\partial \mathbf{x}_t$.
- Instead, we may define a regularization operator $r : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$ (e.g. a Guassian blur kernel) to map $\mathbf{x}$ to a more regularized version of itself in every step. So the update step becomes:

$$\mathbf{x}_{t+1} = r(\mathbf{x}_t) + \epsilon_1 \frac{\partial a(\mathbf{x}_t)}{\partial \mathbf{x}_t} \tag{5}$$

Activation Maximization via Hand-Designed Priors

Experiments

- Effects of different image prior



Activation Maximization via Hand-Designed Priors

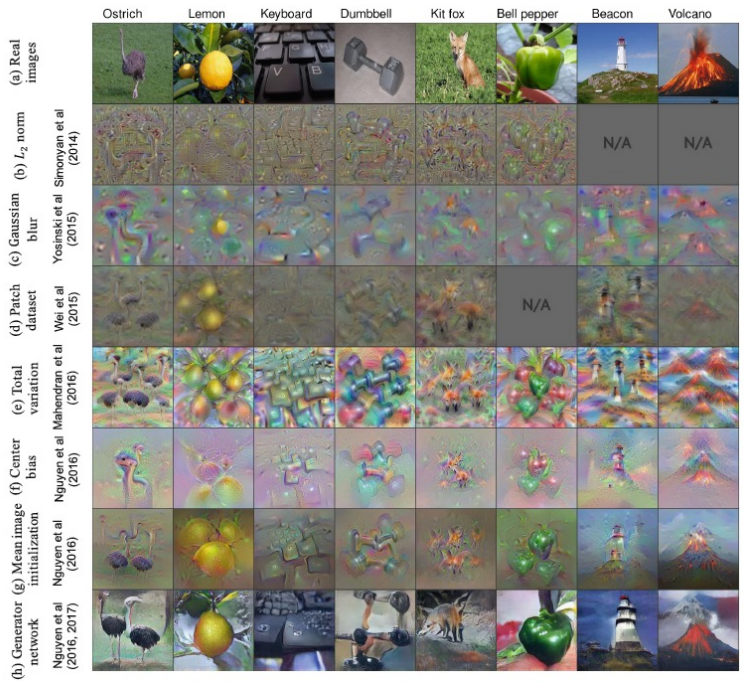
Discussions

- Local Statistics
  - Without priors, AM images often appear to have high-frequency pattern and unnatural colors. Many regularizers have been designed to AM, such as α-norm, Gaussian blurring, bilateral filter, etc..
- Global structures
  - Many AM images still lack global coherence.
  - An image synthesized to highly activate the "bell pepper" output neuron(Fig.b–e) may exhibit multiple bell-pepper segments scattered around the same image rather than a single bell pepper, which suggest the network has learned some local discriminative features.

Activation Maximization via Deep Generator Networks

Existing Problem

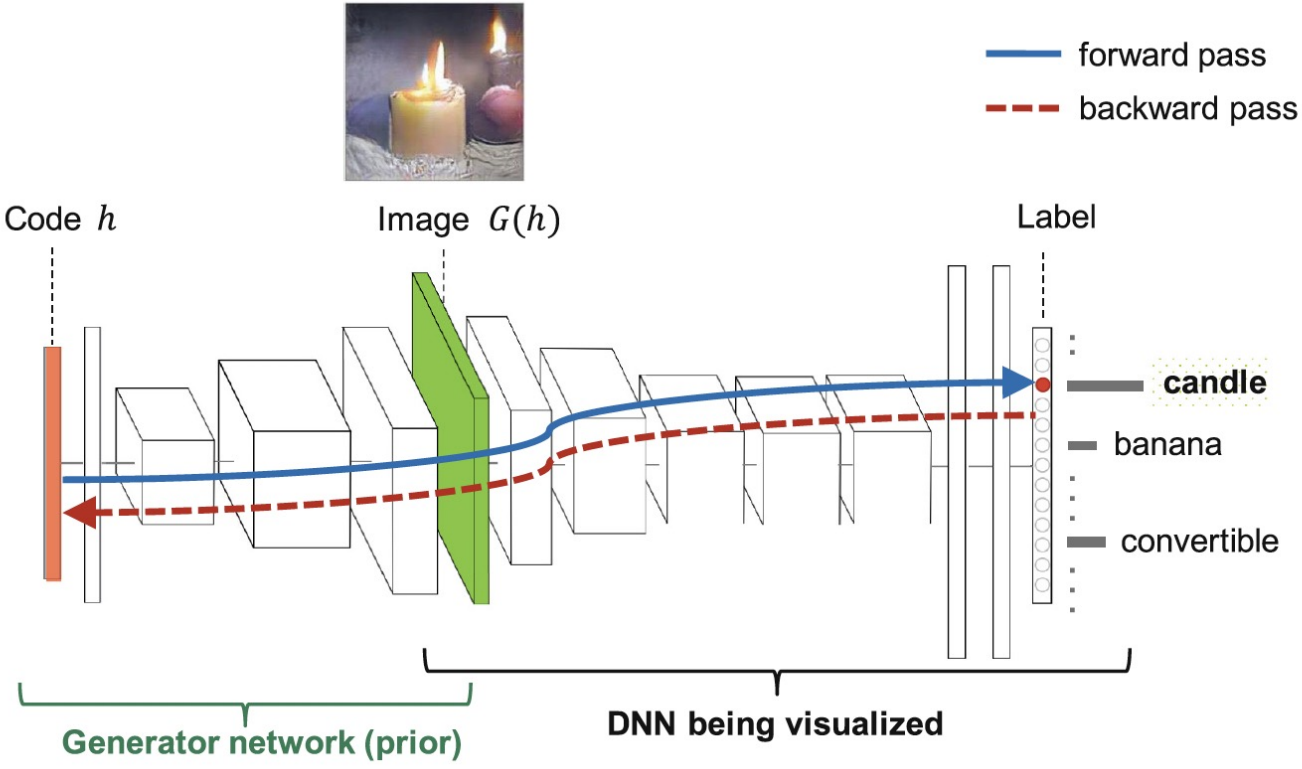
- Much previous AM research were optimizing the preferred stimuli directly in the high-dimensional image space where pixel-wise changes are often slow and uncorrelated, yielding high-frequency visualizations (Fig. b–e).



Activation Maximization via Deep Generator Networks

Solution

- Deep Generator Network Activation Maximization (DGN-AM)
  - Optimize in the low-dimensional latent space of a deep generator network.



Activation Maximization via Deep Generator Networks

Generator Networks

- The authors denote the sub-network of CaffeNet that maps images into 4096-D fc6 features as an encoder $E : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{4096}$.
- The generator network $E : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{4096}$ is trained to invert E i.e. $G(E(\mathbf{x})) \approx \mathbf{x}$

Activation Maximization via Deep Generator Networks

Generator Networks

- The authors denote the sub-network of CaffeNet that maps images into 4096-D fc6 features as an encoder $E : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{4096}$.
- The generator network $E : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{4096}$ is trained to invert E i.e. $G(E(\mathbf{x})) \approx \mathbf{x}$

Optimizing in the Latent Space

- Intuitively, we search in the input code space of the generator G to find a code $\mathbf{h} \in \mathbb{R}^{4096}$ such that the image $G(h)$ maximizes the neural activation $a(G(h))$. So, the AM problem in Eq.(3) now becomes:

$$\mathbf{h}^* = \arg\max_{\mathbf{h}} (a(G(\mathbf{h})) - R(\mathbf{h})) \tag{6}$$

Activation Maximization via Deep Generator Networks

Optimizing in the Latent Space

- Then take steps in the latent space following the below update rule:

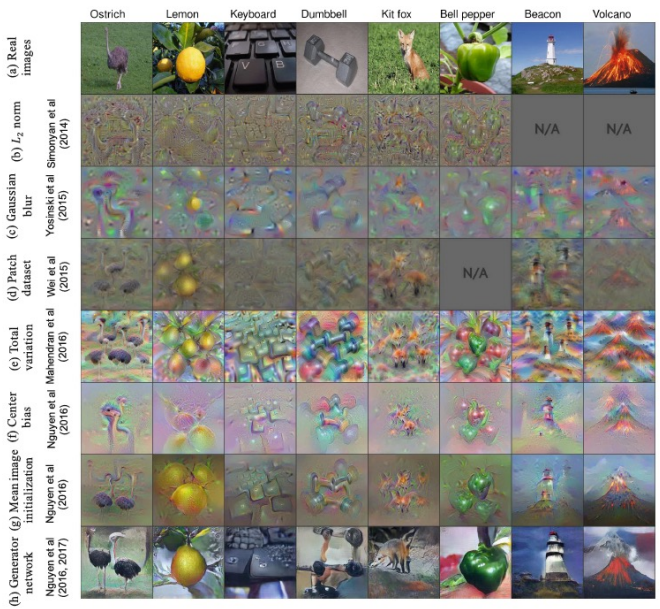
$$\mathbf{h}_{t+1} = \mathbf{h}_t + \epsilon_1 \frac{\partial a(G(\mathbf{h}_t))}{\partial \mathbf{h}_t} - \epsilon_2 \frac{\partial R(\mathbf{h}_t)}{\partial \mathbf{h}_t} \tag{7}$$

- Note that, here, the regularization term $R(.)$ is on the latent code h instead of the image $\mathbf{x}$.

Activation Maximization via Deep Generator Networks

Experiments

- Optimizing in the latent space of a deep generator network showed a great improvement in image quality.
- However, images synthesized by DGN-AM have limited diversity.
  - Learning realism prior for h via a denoising autoencoder.
  - Adding a small amount of Gaussian noise in every update step.



Probabilistic Interpretation for Activation Maximization

Synthesizing Selective Stimuli

- In the original AM formulation (Eq. 1), we only explicitly maximize the activation a_i^l of a unit indexed i in layer l; however, in practice, this objective may surprisingly also increase the activations $a_{j \neq i}^l$ of some other units j in the same layer and even higher than a_i^l.
- For example, maximizing the output activation for the "hartebeest" class is likely to yield an image that also strongly activates the "impala" unit because these two animals are visually similar.
- As the result, there is no guarantee that the target unit will be the highest activated across a layer. In that case, the resultant visualization may not portray what is unique about the target unit (l, i).

Probabilistic Interpretation for Activation Maximization

Selective Stimuli

- Only activate a_i^l, but not $a_{j \neq i}^l$
- The author wish to maximize a_i^l such that it is the highest single activation across the same layer l.
- To enforce that selectivity, we can either maximize the softmax or log of softmax of the raw activations across a layer where the softmax transformation for unit i across layer l is given as $s_i^l = \exp\left(a_i^l\right) / \sum_j \exp\left(a_j^l\right)$.
- Advantages
  - More interpretable and preferred in neuroscience because they contain only visual features exclusively for one unit of interest but not others;
  - Naturally fit the probabilistic interpretation discussed below.

Probabilistic Interpretation for Activation Maximization

Probabilistic Framework

- Let us assume a joint probability distribution $p(x, y)$ where x denotes images, and y is a categorical variable for a given neuron indexed i in layer l. This model can be decomposed into an image density model and an image classifier model:

$$p(\mathbf{x}, y) = p(\mathbf{x})p(y \mid \mathbf{x}) \tag{8}$$

Note that, when l is the output layer of an ImageNet 1000-way classifier, y also represents the image category (e.g. "volcano"), and $p(y|x)$ is the classification probability distribution (often modeled via softmax).

Probabilistic Interpretation for Activation Maximization

Probabilistic Framework

- We can construct a Metropolis-adjusted Langevin (MALA) sampler for our $p(x, y)$ model. This variant of MALA does not have the accept/reject step, and uses the following transition operator:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon_{12} \nabla \log p\left(\mathbf{x}_t, y\right) + N\left(0, \epsilon_3^2\right) \tag{9}$$

- Since y is a categorical variable, and chosen to be a fixed neuron y_c outside the sampler, the above update rule can be re-written as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon_{12} \nabla \log p\left(y = y_c \mid \mathbf{x}_t\right) + \epsilon_{12} \nabla \log p\left(\mathbf{x}_t\right) + N\left(0, \epsilon_3^2\right) \tag{10}$$

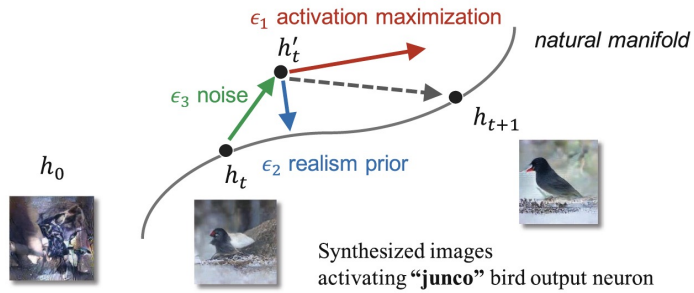
Probabilistic Interpretation for Activation Maximization

Probabilistic Framework

- Decoupling ϵ_{12} into explicit ϵ_1 and ϵ_2 multipliers, and expanding the ∇ into explicit partial derivatives, we arrive at the following update rule:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon_1 \frac{\partial \log p\left(y = y_c \mid \mathbf{x}_t\right)}{\partial \mathbf{x}_t} + \epsilon_2 \frac{\partial \log p\left(\mathbf{x}_t\right)}{\partial \mathbf{x}_t} + N\left(0, \epsilon_3^2\right) \quad (11)$$

where ϵ_1 term: take a step toward an image that causes the neuron y_c to be the *highest activated* across a layer (red arrow). ϵ_2 term: take a step toward a generic, realistic-looking image (blue arrow). ϵ_3 term: add a small amount of noise to jump around the search space to encourage image diversity (green arrow).



Probabilistic Interpretation for Activation Maximization

Maximizing Raw Activations vs. Softmax

- Note that the ϵ_1 term in Eq. 11 is not the same as the gradient of raw activation term in Eq. 2. The following table summarize the three variants of computing this ϵ_1 gradient term.

a. Derivative of raw activations. Worked well in practice [10, 27] but may produce *non-selective* stimuli and is not quite the right term under the probabilistic framework in Sect. 4.4.2	$\frac{\partial a_i^l}{\partial x}$
b. Derivative of softmax. Previously avoided due to poor performance [42, 48], but poor performance may have been due to ill-conditioned optimization rather than the inclusion of logits from other classes	$\frac{\partial s_i^l}{\partial x} = s_i^l \left(\frac{\partial a_i^l}{\partial x} - \sum_j s_j^l \frac{\partial a_j^l}{\partial x} \right)$
c. Derivative of log of softmax. Correct term under the sampler framework in Sect. 4.4.2. Well-behaved under optimization, perhaps due to the $\frac{\partial a_i^l}{\partial x}$ term untouched by the s_i^l multiplier	$\frac{\partial \log s_i^l}{\partial x} = \frac{\partial \log p(y = y_i \mid x_t)}{\partial x} = \frac{\partial a_i^l}{\partial x} - \frac{\partial}{\partial x} \log \sum_j \exp(a_j^l)$

Probabilistic Interpretation for Activation Maximization

Interpretation of Previous Algorithm

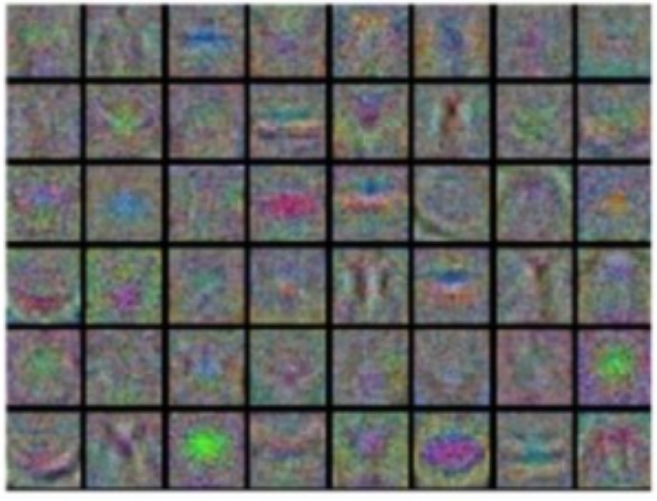
- AM with no priors;
  - $(\epsilon_1, \epsilon_2, \epsilon_3) = (1, 0, 0)$
- AM with a Gaussian prior;
  - $(\epsilon_1, \epsilon_2, \epsilon_3) = (1, \lambda, 0)$
- AM with hand-designed priors;
  - $(\epsilon_3 = 0)$
- AM in the latent space of generator networks.
  - $(\epsilon_1, \epsilon_2, \epsilon_3) = (1, 1, 0)$

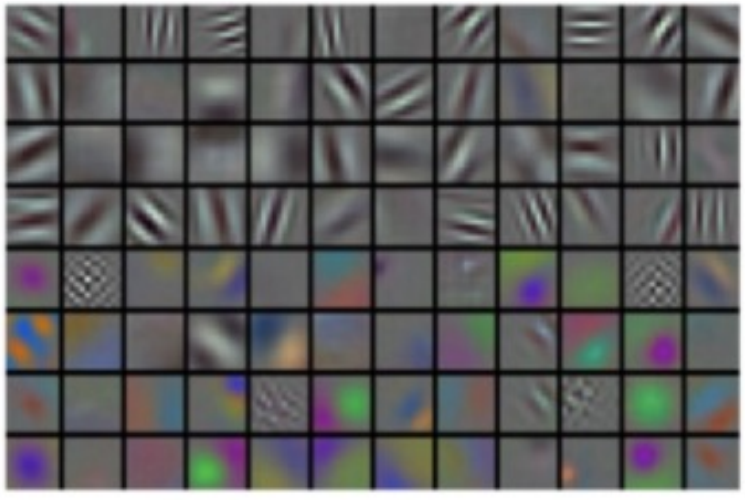
Application of Activation Maximization

- Visualize output units for new Tasks;
- Visualize Hidden Units;
- Synthesize Preferred Images Activating Multiple Neurons;
- Watch Feature Evolution During Training;
- Synthesizing Videos;
- Activation Maximization as a Debugging Tool;
- Synthesize Preferred Images conditioned on a Sentence;
- Synthesize Preferred Images Conditioned on a Semantic Segmentation Map;
- Synthesize Preferred Stimuli for Real, Biological Brains.

Application of Activation Maximization

- Examples





Application of Activation Maximization

- Examples

(a) Regular ImageNet training images

(b) ImageNet training images converted into the BRG color space

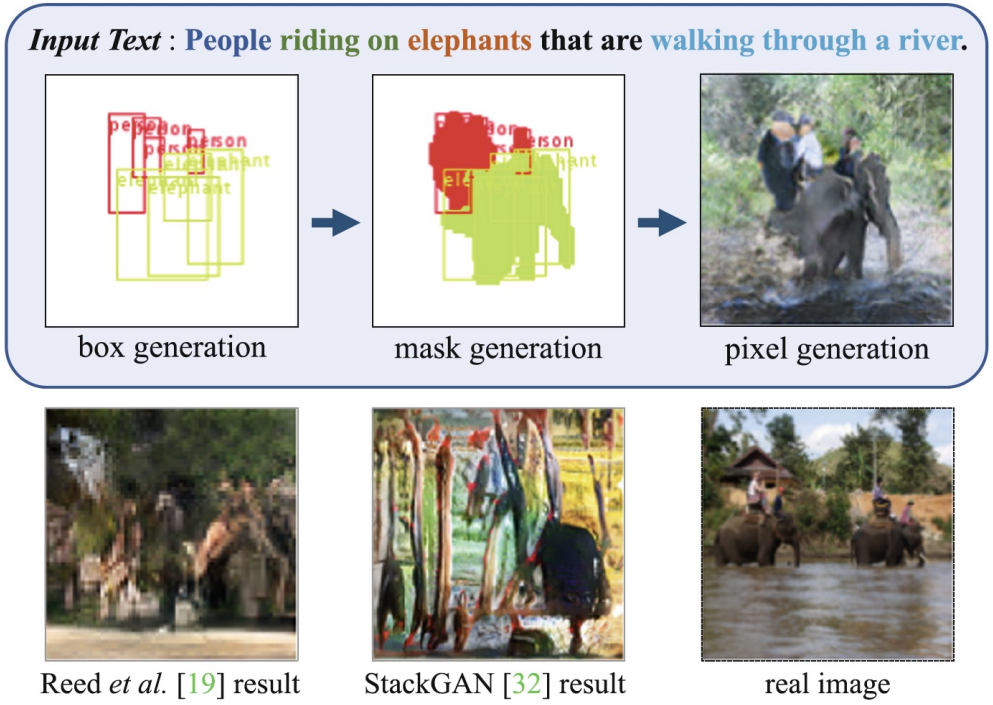
(c) Visualizations of the units that are trained on BRG ImageNet images above (b)

Section2: Interpretable Text-to-Image Synthesis with Hierarchical Semantic Layout Generation

Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee

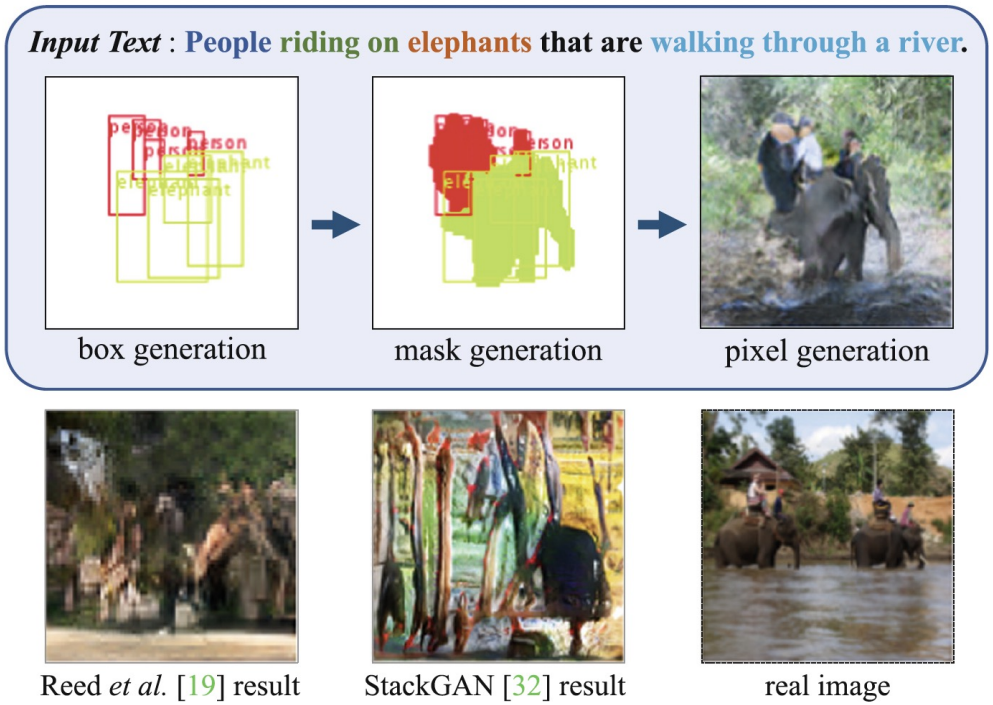
Introduction

- Generating images from natural language description has drawn a lot of attention.
  - Practical usefulness;
  - Describing the model's understanding of the visual concepts by synthesizing images matching the text description;
  - Allowing users to describe visual concepts in natural language.



Introduction

- Recently, conditional Generative Adversarial Network based method have shown promising results on this task.
  - Learning a direct mapping from text to image;
  - Limited to simple datasets, such as brids and flowers;
  - Generation of complicated real-world images, such as MS-COCO, remains as open challenge.



Related Work

- Various approaches have been proposed to formulate the task as a conditional image generation problem.
  - Auto-Encoders;
  - Auto-regressive models;
  - optimization techniques;
  - Conditional GANs.
- Recently, the problem of generating of images from pixel-wise semantic label has been studied, which the image generation task is formulated as translating semantic labels to pixels.
- Existing Problems
  - The first approaches are difficult to generate complicated images;
  - The second approaches require ground-truth layout for generation.

Methodology

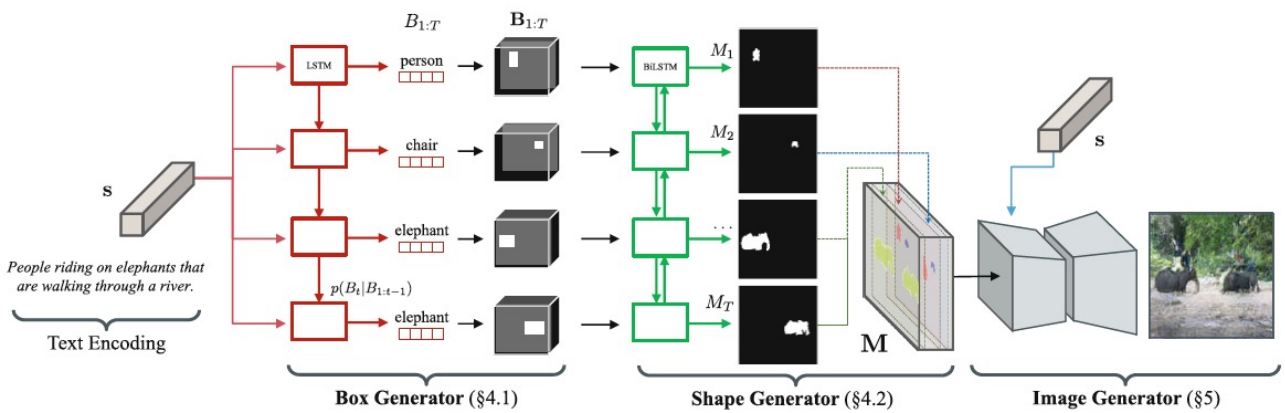
Contributions

- We propose a novel approach for synthesizing images from complicated text descriptions. Our model explicitly constructs a semantic layout from the text description and guides image generation using the inferred semantic layout.
- Conditioning image generation on explicit layout prediction allows our method to generate images that are semantically meaningful and well-aligned with input descriptions.
- We conduct extensive quantitative and qualitative evaluations on the challenging MS-COCO dataset and demonstrate substantial improvement on generation quality over existing works.

Methodology

Framework overview

- **The box generator** takes a text embedding s as input and generates a coarse layout by composing object instances in an image.
- **The shape generator** takes a set of bounding boxes generated from the box generator and predicts the shapes of the objects inside the boxes.
- **The image generator** takes the semantic label map M obtained by aggregating instance-wise masks and the text embedding as inputs and generates an image by translating a semantic layout to pixels matching the text description.



Methodology

Inferring Semantic Layout from text

- Bounding Box Generation
  - Given an input text embedding s, we first generate a coarse layout of the image in the form of object bounding boxes.
  - Associate each bounding box B_t with a class label to define which object class to place and where, which plays a critical role in determining the global layout of the scene.
  - The box generator G_{box} defines a stochastic mapping from the input text s to a set of T object bounding boxes $B_{1:T} = \{B_1, \ldots, B_T\}$

$$\widehat{B}_{1:T} \sim G_{\text{box}}(\text{s}) \tag{12}$$

  - Model
    - An auto-regressive decoder is used for the box generator by decomposing the conditional joint bounding box probability as: $p(B_{1:T} \mid \mathbf{s}) = \prod_{t=1}^{T} p(B_t \mid B_{1:t-1}, \mathbf{s})$, where the conditions are approximated by LSTM.

Methodology

Inferring Semantic Layout from text

- Bounding Box Generation
  - Model
    - In the generative process, we first sample a class label l_t for the t-th object and then generate the box coordinates b_t conditioned on l_t, i.e., $p(B_t \mid \cdot) = p(\mathbf{b}_t, \boldsymbol{l}_t \mid \cdot) = p(\boldsymbol{l}_t \mid \cdot)\, p(\mathbf{b}_t \mid \boldsymbol{l}_t, \cdot)$.
    - The two conditionals are modeled by a Gaussian Mixture Models (GMM) and a categorical distribution, respectively:

$$p(\boldsymbol{l}_t \mid B_{1:t-1}, \mathbf{s}) = \text{Softmax}(\mathbf{e}_t) \tag{13}$$

$$p(\mathbf{b}_t \mid l_t, B_{1:t-1}, \mathbf{s}) = \sum_{k=1}^{K} \pi_{t,k} \mathcal{N}(\mathbf{b}_t; \boldsymbol{\mu}_{t,k}, \boldsymbol{\Sigma}_{t,k}) \tag{14}$$

where K is the number of mixture components. e_t is parameter for and the softmax logit, $\pi_{t,k}$ is parameters for the Gaussian mixtures.

Methodology

Inferring Semantic Layout from text

- Bounding Box Generation
  - Training
    - The box generator is trained by minimizing the negative log-likelihood of ground-truth bounding boxes:

$$\mathcal{L}_{\text{box}} = -\lambda_l \frac{1}{T} \sum_{t=1}^{T} l_t^* \log p\left(\boldsymbol{l}_t\right) - \lambda_b \frac{1}{T} \sum_{t=1}^{T} \log p\left(\mathbf{b}_t^*\right) \tag{15}$$

where T is the number of objects in an image. and λ_l, λ_b are balancing hyperparameters. $\mathbf{b}_t^*$ and $\mathbf{l}_t^*$ are the ground-truth bounding box coordinates and a label of the t-th object.

Methodology

Inferring Semantic Layout from text

- Shape Generation
  - Given a set of bounding boxes obtained by the box generator, the shape generator predicts a more detailed image structure in the form of object masks.
  - In particular, for each object bounding box B_t obtained, we generate a binary mask $M_t \in \mathbb{R}^{H \times W}$ that defines the shape of the object inside the box.
  - To this end, we first convert the discrete bounding box outputs B_t to a binary tensor $\mathbf{B}_t \in \{0, 1\}^{H \times W \times L}$, whose element is 1 if and only if it is contained in the corresponding class-labeled box.
  - Model
    - The shape generator G_{mask} is defined as $\widehat{M}_{1:T} = G_{\text{mask}}(\mathbf{B}_{1:T}, \mathbf{z}_{1:T})$. The shape generator G_{mask} is built using a convolutional recurrent neural network.

Methodology

Inferring Semantic Layout from text

- Shape Generation
  - Training
    - We encouraging each object mask to be compatible with class and location information encoded by the object bounding box. The instance-wise discriminator D_{inst} is trained by optimizing the following instance-wise adversarial loss:

$$\begin{aligned}\mathcal{L}_{\text{inst}}^{(t)} =& \mathbb{E}_{(\mathbf{B}_t, M_t)}\left[\log D_{\text{inst}}\left(\mathbf{B}_t, M_t\right)\right] \\ &+ \mathbb{E}_{\mathbf{B}_t, \mathbf{z}_t}\left[\log\left(1 - D_{\text{inst}}\left(\mathbf{B}_t, G_{\text{mask}}^{(t)}\left(\mathbf{B}_{1:T}, \mathbf{z}_{1:T}\right)\right)\right)\right]\end{aligned} \tag{16}$$

    - On the other hand, the global loss encourages all the instance-wise masks to form a globally coherent context. To consider the relation between different objects, we aggregate them into a global mask $G_{\text{global}}\left(\mathbf{B}_{1:T}, \mathbf{z}_{1:T}\right) = \sum_t G_{\text{mask}}^{(t)}\left(\mathbf{B}_{1:t}, \mathbf{z}_{1:t}\right)$, and compute a global adversarial loss analogous to Eq. (16):

Methodology

Inferring Semantic Layout from text

- Shape Generation
  - Training

$$\mathcal{L}_{\text{global}} = \mathbb{E}_{(\mathbf{B}_{1:T}, M_{1:T})}\left[\log D_{\text{global}}\left(\mathbf{B}_{\text{global}}, M_{\text{global}}\right)\right] + \mathbb{E}_{\mathbf{B}_{1:T}, \mathbf{z}_{1:T}}\left[\log\left(1 - D_{\text{global}}\left(\mathbf{B}_{\text{global}}, G_{\text{global}}\left(\mathbf{B}_{1:T}, \mathbf{z}_{1:T}\right)\right)\right)\right] \tag{17}$$

- Finally, we impose a reconstruction loss L_{rec} that encourages the predicted instance masks to be similar to the ground-truths.

$$\mathcal{L}_{\text{rec}} = \sum_{l} \left\| \Phi_l\left(G_{\text{global}}\right) - \Phi_l\left(M_{\text{global}}\right) \right\| \tag{18}$$

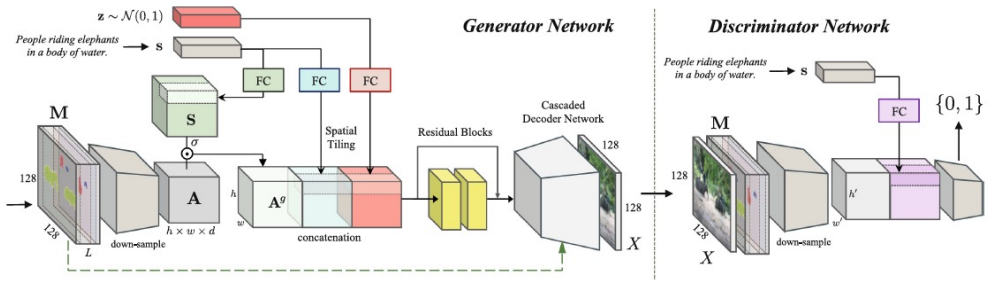
- Combining Eqs. (16), (17) and (18) allows the overall training objective for the shape generator to become:

$$\mathcal{L}_{\text{shape}} = \lambda_i \mathcal{L}_{\text{inst}} + \lambda_g \mathcal{L}_{\text{global}} + \lambda_r \mathcal{L}_{\text{rec}} \tag{19}$$

Methodology

Synthesizing Images from Text and Layout

- Image Generator
  - Model



  - Training
    - Conditioned on both the semantic layout M and the text embeddings, the image generator G_{img} is jointly trained with the discriminator D_{img}. We define the objective function by $\mathcal{L}_{\text{img}} = \lambda_a \mathcal{L}_{\text{adv}} + \lambda_r \mathcal{L}_{\text{rec}}$

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{(\mathbf{M},\mathbf{s},X)} \left[\log D_{\mathbf{img}}(\mathbf{M}, \mathbf{s}, X)\right]$$

$$+\mathrm{E}_{(\mathbf{M},\mathbf{s}),\mathbf{z}} \left[\log \left(1 - D_{\mathbf{img}}\left(\mathbf{M}, \mathbf{s}, G_{\text{img}}(\mathbf{M}, \mathbf{s}, \mathbf{z})\right)\right)\right] \quad (20)$$

$$\mathcal{L}_{\text{rec}} = \sum_{l} \left\| \Phi_l \left(G_{\text{img}}(\mathbf{M}, \mathbf{s}, \mathbf{z})\right) - \Phi_l(X) \right\| \quad (21)$$

Experiments

Experiments Setup

- **DataSet**. The MS-COCO dataset is used to evaluate the proposed model. It contains 164,000 training images over 80 semantic classes, where each images is associated with instance-wise annotations (i.e. object bounding boxes and segmentation masks).
- **Evaluation Metrics**.
  - *Inception Score*–It measures the recognizability and the diversity of generated images and is correlated with human perceptions on visual quality.
  - *Caption Generation*–It measures the relevance of the generated images to the input texts. The three standard language similarity metrics: BLEU, METEOR, and CIDEr.
  - *Human Evaluation*–Asking users to rank the methods based on the relevance of generated images to text.

Experiments

Quantitative Analysis

- Comparison to other methods

Table 5.1. Quantitative evaluation results. Two evaluation metrics based on caption generation and the Inception score are presented. The second and third columns indicate types of bounding box or mask layouts used in image generation, where "GT" indicates ground-truth and "Pred." indicates the predicted layouts by our model. The last row presents the caption generation performance on real images, which corresponds to the upper-bound of the caption generation metric. Higher values are more accurate in all columns.

Method	Box	Mask	Caption generation						Inception [26]
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	
Reed *et al.* [23]	-	-	0.470	0.253	0.136	0.077	0.122	0.160	7.88 ± 0.07
StackGAN [39]	-	-	0.492	0.272	0.152	0.089	0.128	0.195	8.45 ± 0.03
Ours	Pred.	Pred.	**0.541**	**0.332**	**0.199**	**0.122**	**0.154**	**0.367**	**11.46 ± 0.09**
Ours (control experiment)	GT	Pred.	0.556	0.353	0.219	0.139	0.162	0.400	11.94 ± 0.09
	GT	GT	0.573	0.373	0.239	0.156	0.169	0.440	12.40 ± 0.08
Real images (upper bound)	-	-	0.678	0.496	0.349	0.243	0.228	0.802	-

Table 5.2. Human evaluation results.

Method	Ratio of ranking 1st	*vs.* Ours
StackGAN [39]	18.4%	29.5%
Reed *et al.* [23]	23.3%	32.3%
Ours	**58.3**%	-

Experiments

Quantitative Analysis

- Comparison to other methods

Ground Truth				
	(GT) ***A kid in wet-suit on surfboard in the ocean.***	**(GT)** ***a lady that is on some skies on some snow***	**(GT)** ***A young man playing fris-bee while people watch.***	**(GT)** ***A bus that is sitting in the street.***
	generated image and caption	generated image and caption	generated image and caption	generated image and caption
StackGAN 256x256		a person flying a kite on a beach .		a man is walking on a beach with a surfboard .		a man is standing next to a cow .		a city street with a traffic light and a green light .
Reed *et al.* 64x64		a man is flying a kite in the sky		a person is riding a snowboard on a snowy slope .		a group of people standing around a field with kites .		a large boat is in the water near a city .
Ours 128x128		a man is surfing in the ocean with a surfboard .		a man is skiing down a hill with a snowboard .		a man is playing with a frisbee in a field .		a red and white bus parked on a city street .

Experiments

Qualitative Analysis

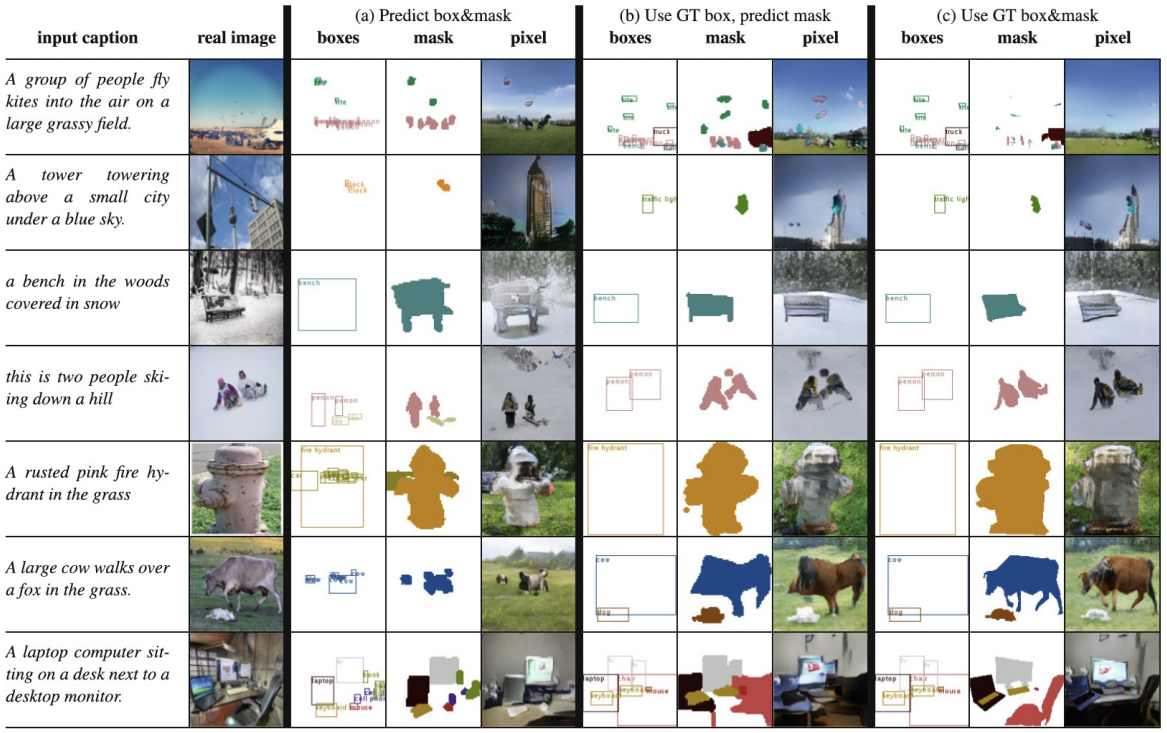


Fig. 5.5. Image generation results of our method. Each column corresponds to generation results conditioned on (a) predicted box and mask layouts, (b) ground-truth box and predicted mask layouts, and (c) ground-truth box and mask layouts. Classes are color-coded for illustration purpose. Best viewed in color.

Experiments

Qualitative Analysis

- Diversity of Samples

Fig. 5.6. Multiple samples generated from a text description.

Experiments

Qualitative Analysis

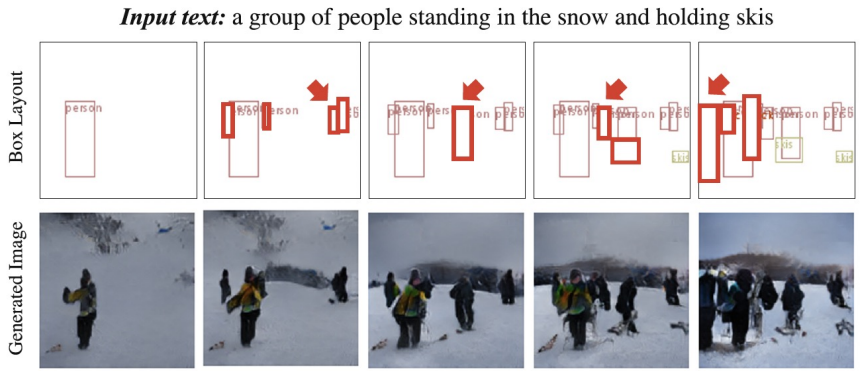
- Text-Conditional Generation

Fig. 5.7. Generation results by manipulating captions. The manipulated portions of texts are highlighted in **bold** characters, where the types of manipulation is indicated by different colors. **Blue**: scene context, **Magenta**: spatial location, **Red**: the number of objects, ahd **Green**: object category.

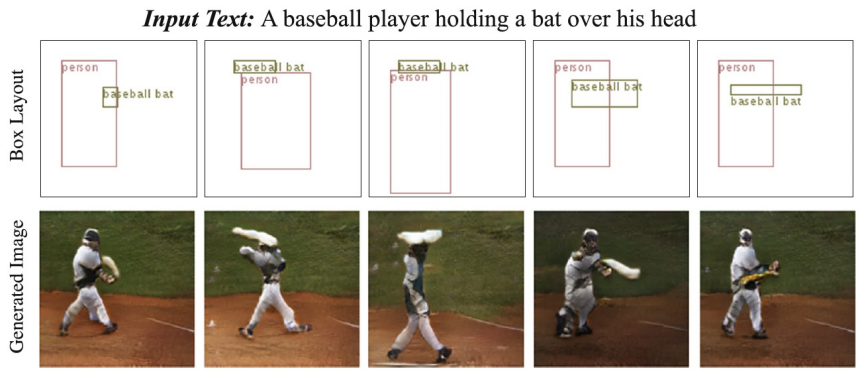
Experiments

Qualitative Analysis

- Controllable Image Generation



(a) Generation results by adding new objects.



(b) Generation results by changing spatial configuration of objects.

Fig. 5.8. Examples of controllable image generation.

Thank you for your Attentions!
Q & A

AALBORG UNIVERSITY
DENMARK