Applications of GANs

- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
- Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
- Generative Adversarial Text to Image Synthesis
Using GANs for Single Image Super-Resolution

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi
Problem

How do we get a high resolution (HR) image from just one (LR) lower resolution image?

Answer: We use super-resolution (SR) techniques.
Previous Attempts

original
bicubic (21.59dB/0.6423)
SRResNet (23.44dB/0.7777)
SRGAN (20.34dB/0.6562)
SRGAN

original

bicubic
(21.59dB/0.6423)

SRRResNet
(23.44dB/0.7777)

SRGAN
(20.34dB/0.6562)
SRGAN - Generator

- $G$: generator that takes a low-res image $I^{LR}$ and outputs its high-res counterpart $I^{SR}$
- $\theta_G$: parameters of $G$, $\{W_{1:L}, b_{1:L}\}$
- $l^{SR}$: loss function measures the difference between the 2 high-res images

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR})$$
SRGAN - Discriminator

- $D$: discriminator that classifies whether a high-res image is $I_{HR}$ or $I_{SR}$
- $\theta_D$: parameters of $D$

\[
\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I_{HR} \sim p_{\text{train}}(I_{HR})} [\log D_{\theta_D}(I_{HR})] + \\
\mathbb{E}_{I_{LR} \sim p_{G}(I_{LR})} [\log(1 - D_{\theta_D}(G_{\theta_G}(I_{LR})))]
\]
SRGAN - Perceptual Loss Function

Loss is calculated as weighted combination of:

➔ Content loss
➔ Adversarial loss
➔ Regularization loss
SRGAN - Content Loss

Instead of MSE, use loss function based on ReLU layers of pre-trained VGG network. Ensures similarity of content.

- $\Phi_{i,j}$: feature map of $j^{th}$ convolution before $i^{th}$ maxpooling
- $W_{i,j}$ and $H_{i,j}$: dimensions of feature maps in the VGG

$$
\ell_{\text{VGG}/i,j}^{SR} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I_{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I_{LR}))_{x,y})^2
$$
SRGAN - Adversarial Loss

Encourages network to favour images that reside in manifold of natural images.

\[ l_{Gen}^{SR} = \sum_{n=1}^{N} - \log D_{\theta_D}(G_{\theta_G}(I_{LR}^n)) \]
SRGAN - Regularization Loss

Encourages spatially coherent solutions based on total variations.

\[ l^{SR}_{TV} = \frac{1}{r^2WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} ||\nabla G_{\theta_G}(I^{LR})_{x,y}|| \]
SRGAN - Examples

Figure 5: Reference HR image (left: a,b) with corresponding SRRResNet (middle left: c,d), SRGAN-MSE (middle: e,f), SRGAN-VGG22 (middle right: g,h) and SRGAN-VGG54 (right: i,j) reconstruction results.
SRGAN - Examples

original HR image

SRResNet

SRGAN-VGG54

(a)  (c)  (e)

(b)  (d)  (f)
Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

Work by Emily Denton, Soumith Chintala, Arthur Szlam, Rob Fergus
Short Background
Conditional Generative Adversarial Nets (CGAN)

Mirza and Osindero (2014)

GAN
\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{z}(z)}[\log (1 - D(G(z)))]
\]

CGAN
\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_{z}(z)}[\log (1 - D(G(z|y))))]
\]
Laplacian pyramid

\[
\hat{h}_k = L_k(I) = G_k(I) - u(G_{k+1}(I)) = I_k - u(I_{k+1})
\]

\[
I_k = u(I_{k+1}) + h_k
\]

Burt and Adelson (1983)
Laplacian pyramid

\[ I_k \rightarrow \begin{array}{c}
\text{\textbf{L}}_k(I) = \mathcal{G}_k(I) - u(G_{k+1}(I)) = I_k - u(I_{k+1}) \\
I_k = u(I_{k+1}) + h_k
\end{array} \]
Laplacian Pyramid Generative Adversarial Network (LAPGAN)
Image Generation

\[ \tilde{I}_k = u(\tilde{I}_{k+1}) + \tilde{h}_k = u(\tilde{I}_{k+1}) + G_k(z_k, u(\tilde{I}_{k+1})) \]
Generation: Coarse to fine
Different draws, starting from the same initial 4x4 image
Some thoughts on the method

- The Laplacian Pyramid Framework is independent of the Generative Model

Possible to use a completely different model like Pixel RNN
Some thoughts on the method

- The **Generative Models** at each step can be **totally different**!
Some thoughts on the method

- The Generative Models at each step can be totally different!
Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

Author’s code available at: https://github.com/reedscot/icml2016
Motivation

Current deep learning models enable us to...

- Learn feature representations of images & text
- Generate realistic images & text

- pull out images based on captions
- generate descriptions based on images
- answer questions about image content

“Two pizzas sitting on top of a stove top oven”
Problem - Multimodal distribution

• Many plausible image can be associated with one single text description

• Previous attempt uses Variational Recurrent Autoencoders to generate image from text caption but the images were not realistic enough. (Mansimov et al. 2016)
What GANs can do

• CGAN: Use side information (e.g. classes) to guide the learning process

• Minimax game: Adaptive loss function

➤ Multi-modality is a very well suited property for GANs to learn.
The Model - Basic CGAN

Pre-trained char-CNN-RNN

Learns a compatibility function of images and text -> joint embedding

This flower has small, round violet petals with a dark purple center

\[ \varphi \quad \varphi(t) \]

\[ z \sim N(0, 1) \]

\[ z \in \mathbb{R}^Z \]

\[ \hat{x} := G(z, \varphi(t)) \]

This flower has small, round violet petals with a dark purple center

\[ D(\hat{x}, \varphi(t)) \]

Generator Network

Discriminator Network

\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_z(z)}[\log(1 - D(G(z)))]
\]
The Model - Variations

GAN-CLS

In order to distinguish different error sources:

Present to the discriminator network 3 different types of input. (instead of 2)

Algorithm

1: **Input:** minibatch images $x$, matching text $t$, mis-matching $\hat{t}$, number of training batch steps $S$
2: **for** $n = 1$ **to** $S$ **do**
3: $h \leftarrow \varphi(t)$ \{Encode matching text description\}
4: $\hat{h} \leftarrow \varphi(\hat{t})$ \{Encode mis-matching text description\}
5: $z \sim \mathcal{N}(0, 1)^Z$ \{Draw sample of random noise\}
6: $\hat{x} \leftarrow G(z, h)$ \{Forward through generator\}
7: $s_r \leftarrow D(x, h)$ \{real image, right text\}
8: $s_w \leftarrow D(x, \hat{h})$ \{real image, wrong text\}
9: $s_f \leftarrow D(\hat{x}, h)$ \{fake image, right text\}
10: $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
11: $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ \{Update discriminator\}
12: $\mathcal{L}_G \leftarrow \log(s_f)$
13: $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ \{Update generator\}
14: **end for**
GAN-INT

In order to generalize the output of G:

Interpolate between training set embeddings to generate new text and hence fill the gaps on the image data manifold.

Updated Equation

\[
\min_G \max_D V(D, G) = \\
= \mathbb{E}_{x \sim p_{data}}(x)[\log D(x)] \\
+ \mathbb{E}_{x \sim p_z(z)}[\log(1 - D(G(z)))] + \\
\mathbb{E}_{t_1, t_2 \sim p_{data}}[\log(1 - D(G(z, \beta t_1 + (1 - \beta)t_2)))]
\]

\{fake image, fake text\}

GAN-INT-CLS: Combination of both previous variations
Disentangling

❖ Style is background, position & orientation of the object, etc.
❖ Content is shape, size & colour of the object, etc.

● Introduce $S(x)$, a style encoder with a squared loss function:

$$
\mathcal{L}_{style} = \mathbb{E}_{t, z \sim \mathcal{N}(0,1)} \| z - S(G(z, \varphi(t))) \|^2_2
$$

● Useful in generalization: encoding style and content separately allows for different new combinations
Training - Data (separated into class-disjoint train and test sets)

**Caltech-UCSD Birds**

<table>
<thead>
<tr>
<th>Caption</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>this vibrant red bird has a pointed black beak</td>
<td></td>
</tr>
<tr>
<td>this bird is yellowish orange with black wings</td>
<td></td>
</tr>
<tr>
<td>the bright blue bird has a white colored belly</td>
<td></td>
</tr>
</tbody>
</table>

**MS COCO**

<table>
<thead>
<tr>
<th>Caption</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>a pitcher is about to throw the ball to the batter</td>
<td></td>
</tr>
<tr>
<td>a group of people on skis stand in the snow</td>
<td></td>
</tr>
<tr>
<td>a man in a wet suit riding a surfboard on a wave</td>
<td></td>
</tr>
</tbody>
</table>

**Oxford Flowers**

<table>
<thead>
<tr>
<th>Caption</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>this flower has white petals and a yellow stamen</td>
<td></td>
</tr>
<tr>
<td>the center is yellow surrounded by wavy dark purple petals</td>
<td></td>
</tr>
<tr>
<td>this flower has lots of small round pink petals</td>
<td></td>
</tr>
</tbody>
</table>
Training – Results: Flower & Bird

these flowers have petals that start off white in color and end in a dark purple towards the tips.

a tiny bird, with a tiny beak, tarsus and feet, a blue crown, blue coverts, and black cheek patch
Training Results: MS COCO

A large blue octopus kite flies above the people having fun at the beach.

A toilet in a small room with a window and unfinished walls.

A man in a wet suit riding a surfboard on a wave.

Mansimov et al.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand, clad vast desert.
Training – Results Style disentangling

Text descriptions (content) Images (style)

The bird has a yellow breast with grey features and a small beak.

This is a large white bird with black wings and a red head.

A small bird with a black head and wings and features grey wings.

This bird has a white breast, brown and white coloring on its head and wings, and a thin pointy beak.

A small bird with white base and black stripes throughout its belly, head, and feathers.

A small sized bird that has a cream belly and a short pointed bill.

This bird is completely red.

This bird is completely white.

This is a yellow bird. The wings are bright blue.
Thoughts on the paper

• Image quality

• Generalization

• Future work