Addressing GAN limitations: resolution, lack of novelty and control on generations

Camille Couprie

Facebook AI Research

Joint works with O. Sbai, M. Aubry, A. Bordes, M. Elhoseiny, M. Riviere, Y. LeCun, M. Mathieu, P. Luc, N. Neverova, J. Verbeek.
Why do we care about generative models

• Scene Understanding can be assessed by checking the ability to generate plausible new scenes.
• Generative models are interesting if they can be used to go beyond training data: data of higher resolution, data augmentation to help train better classifiers, use the learned representations in other tasks, or make prediction about uncertain events...
Outline

• 1/ Design inspiration from adversarial generative networks
• 2/ High resolution, decoupled generation
• 3/ Vector image generation by learning parametric layer decomposition
• 4/ Future frame prediction
Design inspiration from generative networks

Sbai, Elhoseiny, Bordes, LeCun, Couprie, ECCV workshop 17

Hedonic Value

Novelty
Generative Adversarial Networks

Random Numbers

0.3
0.7
0.1
0.8

Generator

Generated Image

Adversarial Network

Fake

Goodfellow et al, 2014
Generative Adversarial Networks

0.3
0.7
0.1
0.8

RANDOM NUMBERS

Generator

Real INPUT

GENERATED IMAGE

AdVERSARIAL NETWORK

Real

Goodfellow et al, 2014
Deep convolutional GANs

RADFORD ET AL : ICLR 2015
Training with pictures of about 2000 Clothing items
Texture and shape labels

- Floral
- Tiled
- Uniform
- Dotted
- Animal Print
- Graphical
- Striped

- Skirt
- Pullover
- T-Shirt
- Coat
- Top
- Jacket
- Dress
Class conditioned GAN

RANDOM NUMBERS
0.3
0.7
0.1
0.8

Generator

Real INPUT

GENERATED IMAGE

AdVERSARIAL NETWORK

Shape CLASS

0/1 (REAL/FAKE)

TEXTURE CLASS
GAN Optimization objectives

- **Generator’s loss**
  \[
  \min_{\theta_G} \mathcal{L}_{G_{\text{real/fake}}} = \min_{\theta_G} \sum_{z_i \in \mathbb{R}^n} \log(1 - D(G(z_i)))
  \]

- **Discriminator’s loss**
  \[
  \min_{\theta_D} \mathcal{L}_{D_{\text{real/fake}}} = \min_{\theta_D} \sum_{x_i \in \mathcal{D}, z_i \in \mathbb{R}^n} -\log D(x_i) - \log(1 - D(G(z_i)))
  \]

- **Auxiliary classifier discriminator:**
  \[
  \mathcal{L}_{D} = \lambda_{D_{r}} \mathcal{L}_{D_{\text{real/fake}}} + \lambda_{D_{b}} \mathcal{L}_{D_{\text{classif}}}
  \]

- **Additional loss for the generator:**
  \[
  \mathcal{L}_{G} = \lambda_{G_{r}} \mathcal{L}_{G_{\text{real/fake}}} + \lambda_{G_{c}} \mathcal{L}_{G_{\text{creativity}}}
  \]
Introduction of a Style Deviation criterion

0.3 0.7 0.1 0.8

RANDOM NUMBERS

Generator

Generated Image

AdVERSARIAL NETWORK

Dotted
Floral graphical
100% uniform
tiled
striped
Animal print
Introduction of a Style Deviation criterion

RANDOM NUMBERS

0.3
0.7
0.1
0.8

Generator

Generated Image

AdVERSARIAL NETWORK

Dotted
6%

Floral
7%

57%
graphical

6%
uniform

8%
tiled

11%
striped

9%
Animal print
With the Style Deviation criterion (CAN H)
Tested deviation objectives

Binary cross entropy loss:

\[ \mathcal{L}_{\text{CAN}} = -\sum_{k=1}^{K} \left( \frac{1}{K} \log(D_c(c_k | G(z)) + (1 - \frac{1}{K}) \log(1 - D_c(c_k | G(z)) \right) \]

Multi-class cross entropy loss:

\[ \mathcal{L}_{\text{CAN(H)}} = -\sum_{x_i \in \mathcal{D}} \frac{1}{K} \log \text{softmax}(D_b(x_i)) \]

\[ = -\sum_{x_i \in \mathcal{D}} \frac{1}{K} \log \left( \frac{e^{D_b, \hat{c}_i(x_i)}}{\sum_{k=1}^{K} e^{D_b,k(x_i)}} \right) \]
Human Evaluation Study

Overall Likability (%)

- GAN
- CAN texture
- CAN(h) texture
- Realistic Appearance

CAN: GAN with Creativity loss, (H) stands for the use of a holistic loss.
Models with texture deviation are Most Popular

Judged by humans and measured as a distance to similar training images.
Decoupled adversarial image generation

M. Riviere, C. Couprie, Y. LeCun

Motivation:
- Take advantage of white background clothing datasets
- Potentially avoid defaults in generated shapes
- Better enforce shape conditioning of generations
1024x1024 generations on the RTW dataset

Using Morgane’s pytorch “progressive growing of GANs” available online, Karras et al., ICLR’18
Decoupled architecture

![Diagram](image-url)
Random generations

Progressive growing

Progressive growing with decoupled architecture
Better class conditioning

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Class</th>
<th>GAN-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shoes</td>
<td>PGAN</td>
<td>Pose</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Category</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>Decoupled PGAN</td>
<td>Pose</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Category</td>
<td>0.5</td>
</tr>
<tr>
<td>Clothing</td>
<td>PGAN</td>
<td>Pose</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gender</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>Decoupled PGAN</td>
<td>Pose</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gender</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Accuracy of classifiers trained on FashionGen Clothing and FashionGen Shoes on our different models results (GAN-test metric)

Overall average improvement: 4.7%
Vector Image Generation by learning parametric layer decomposition

Sbai, Couprie, Aubry, arxiv dec18

Current deep generative models are great but... ... are limited in resolution, and control in generations
Related work

Kanan et al: Layered GANs (LR-GANs), ICLR’17
GANIN et al. SPIRAL, ICML’18
Our approach

Spoiler alert: yes, we can generate sharper images, this is just an example.
Our iterative pipeline for image reconstruction

Iterative generation: $I_t = g(I, I_{t-1})$
Vectorized mask generation $M_t(x, y) = g(x, y, p_t)$
Alpha-blending $I_t(x, y) = I_{t-1}(x, y) \cdot (1 - M_t(x, y)) + c_t M_t(x, y)$
Our iterative pipeline for image generation
Trainig criteria

Adversarial net criterion: Wasserstein loss with Gradient Penalty (WGAN-GP), Gulrajani et al. NIPS’17

\[ \mathcal{L}_D = - \mathbb{E}_{x \sim p_d}[D(x)] + \mathbb{E}_{\hat{x} \sim p_g}[D(\hat{x})] + \gamma \mathbb{E}_{\hat{x} \sim p_g}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \]

Our generator loss in the GAN setting:

\[ \mathcal{L}_G^{ADV} = -\mathbb{E}_{\hat{x} \sim p_g}[D(\hat{x})]. \]

Our generator loss in the image reconstruction setting:

\[ \mathcal{L}_G^{REC} = \mathcal{L}_1 + \lambda \mathcal{L}_G^{ADV} \]
Results using a l1 reconstruction loss
Results using a $l_1$ reconstruction loss

Target

Our iterative reconstruction
Applications

Editing the original image using extracted masks, by performing local modifications of luminosity (top), or color modification using a blending of masks (bottom).
Applications

Image editing using masks: Using chosen extracted mask(s) from image reconstruction, we apply object removal (top) and color modifications with object removal (bottom).
Image vectorization:
Reconstruction results on MNIST images. Our model learns a vectorized mask representation of digits that can be generated at any resolution without interpolation artifacts.
Baselines

1/ MLP-baseline
Baselines

1/ MLP-baseline
2/ ResNet baseline
Baselines

1/ MLP-baseline
2/ ResNet baseline
3/ MLP-xy baseline
Comparative results

- Using a l1 reconstruction loss
- Parameters for our approach:
  20 masks, p of size 10, c of size 3: 260 parameters
- Baselines: size of the latent code $z = 20 \times 13 = 260$
GAN results

CelebA generations trained on 64x64 images, sampled at 256x256

CIFAR10 generations trained on 32x32 images, sampled at 256x256
GAN
Results on ImageNet

trained on 64x64 images, sampled at 1024x1024
Results

Target

Perceptual loss

Result with perceptual loss

20 masks

Increasing adversarial weight

$\mathcal{L}_1$

Targets
Conclusion and Future work

Faster training
Use class conditioning
Texture image generation
Predicting next frames in videos

Michael Mathieu, Camille Couprie, Yann LeCun, ICLR16

4 input images

Our 2 predictions
Deep multiscale video prediction beyond Mean square error

• Result with a simple convolutional network trained minimizing an l2 loss

• Our result using
  • A multiscale architecture
  • an image gradient different loss
  • Use adversarial training
Predicting deeper into the future of semantic segmentations

P. Luc, N. Neverova, C. Couprie, J. Verbeek, Y. LeCun ECCV18

• Predictions in the RGB space quickly become blurry despite previous attempts
• Idea: predict in the space of semantic segmentation
Approach – predicting deeper into the future

Single time-step

Batch model

Autoregressive model

Autoregressive model is only possible for X2X, S2S, XS2XS

Autoregressive model is either:
- used for inference without additional training (w.r.t. to single time step model) AR
- Fine-tuned using BPTT AR fine-tune

Same color = shared weights
Some results

Baselines:
- Copy the last input frame to the output
- Estimate flow between the two last inputs, and project the last input forward using the flow

Flow baseline

Our AutoRegressive fine-tune result

Performance measure (mean IoU) of our approach and baselines on the Cityscapes dataset
Instance level segmentation: Mask RCNN

K. He G. Gkioxari P. Dollar R. Girshick’17

- Extends Faster RCNN [Ren et al.’15] by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition
Predicting future segmentation instances by forecasting convolutional features

P. Luc, C. Couprie, Y. LeCun, J. Verbeek, ECCV18

Luc, Neverova et al. ICCV17

F2F predictions

Predictions in feature space for future instance segmentation
Conclusions

Some open problems:
- Automatic metrics to evaluate generative models performances
- Non deterministic training losses for future prediction

Torch code online:
- For future video prediction:
  - Vector image generation: available soon on Othman Sbai’s github
    - of RGBs: on Michael Mathieu’s github
    - of semantic segmentations: on Pauline’s Luc github
    - of instance segmentation: on Pauline’s Luc github