Boulder

Generative Networks; Conditional GANs

YouTube Playlist

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

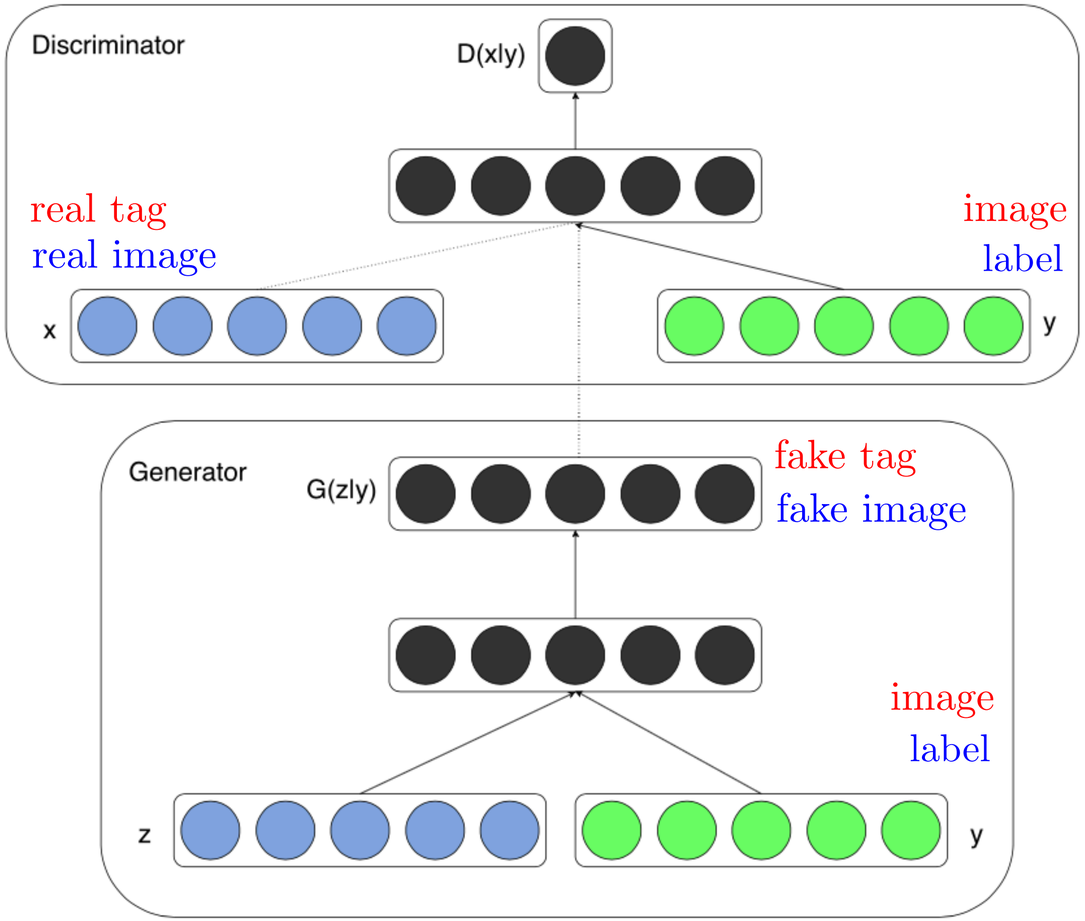
maziar.raissi@colorado.edu

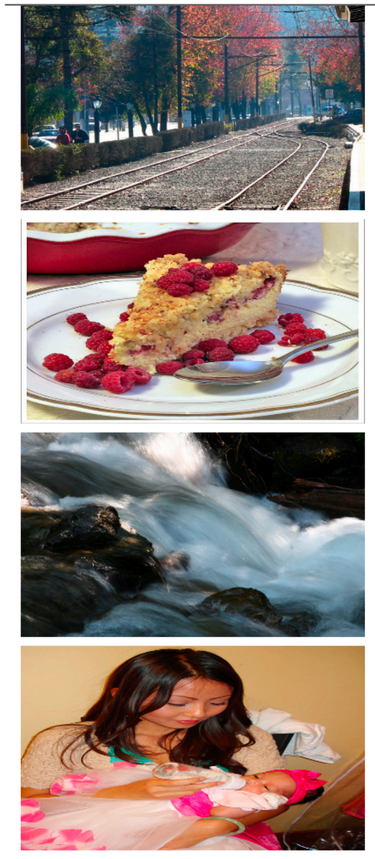
Conditional Generative Adversarial Nets

YouTube Playlist

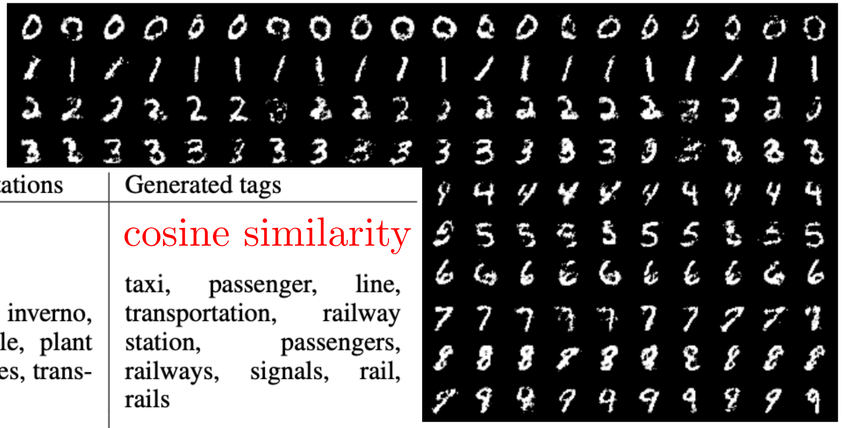
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log \underbrace{D(x)}_{D(x|y)}] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(\underbrace{G(z)}_{G(z|y)}))]$$

One-To-Many Mapping





User tags + annotations	Generated tags
	cosine similarity
montanha, trem, inverno, frio, people, male, plant life, tree, structures, transport, car	taxi, passenger, line, transportation, railway station, passengers, railways, signals, rail, rails
food, raspberry, delicious, homemade	chicken, fattening, cooked, peanut, cream, cookie, house made, bread, biscuit, bakes
water, river	creek, lake, along, near, river, rocky, treeline, valley, woods, waters
people, portrait, female, baby, indoor	love, people, posing, girl, young, strangers, pretty, women, happy, life



tag: word vector
label: one-hot vector

Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).

Context Encoders: Feature Learning by Inpainting

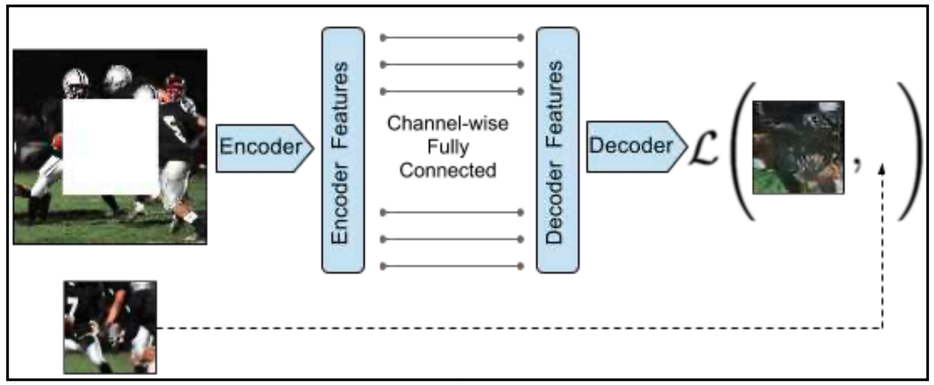
YouTube Playlist

(a) Input context

(b) Human artist

(c) Context Encoder ($L2$ loss)

(d) Context Encoder ($L2$ + Adversarial loss)



Encoder

$$\mathbb{R}^{227\times227\times3} \ni I \triangleright \overbrace{\text{AlexNet(pool5)}}^{\text{5 conv layers}} \in \mathbb{R}^{6\times6\times256}$$

$6 \times 6 \times 256 = 9,216$ from scratch

Channel-wise fully-connected layer

input: m feature maps of size $n \times n$

output: m feature maps of size $n \times n$

$$\underbrace{Y_j}_{n^2\times1} = \underbrace{W_j}_{n^2\times n^2} \underbrace{X_j}_{n^2\times1}, j = 1, \ldots, m$$

$mn^4 \rightarrow$ number of parameters (rather than m^2n^4)

Decoder

5 up-conv layers

Loss function

$x \rightarrow$ ground truth image

$F \rightarrow$ context encoder

$F(x) \rightarrow$ output

$\hat{M} \rightarrow$ binary mask (dropped image region)

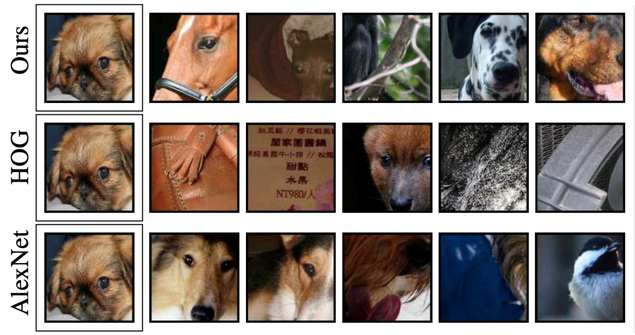
1 whenever a pixel is dropped and 0 for input pixels

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2$$

$$\mathcal{L}_{adv} = \max_D \quad \mathbb{E}_{x\in\mathcal{X}}[\log(D(x)) + \log(1 - D(F((1 - \hat{M}) \odot x)))]$$

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}$$

Method	Mean L1 Loss	Mean L2 Loss	PSNR (higher better)
NN-inpainting (HOG features)	19.92%	6.92%	12.79 dB
NN-inpainting (our features)	15.10%	4.30%	14.70 dB
Our Reconstruction (joint)	**10.33%**	**2.35%**	**17.59 dB**



Center patches whose context (not shown here) are close in the embedding space of different methods

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	**78.2%**	**56.8%**	**48.0%**
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Doersh *et al.* [7]	context	4 weeks	55.3%	**46.6%**	-
Wang *et al.* [39]	motion	1 week	**58.4%**	44.0%	-
Ours	context	14 hours	56.5%	44.5%	**29.7%**

Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Conditional Image Synthesis with Auxiliary Classifier GANs

YouTube Video

Generative Adversarial Network (GAN)

$$X_{fake} = G(z)$$

$$P(\underbrace{S}_{\text{source}} \mid X) = D(X)$$

$$L = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$

Auxiliary Classifier GAN (AC-GAN)

$$X_{fake} = G(\underbrace{c}_{\text{class label}}, \underbrace{z}_{\text{noise}})$$

$$P(S \mid X),\ P(C \mid X) = D(X)$$

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$

$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

$L_S \rightarrow$ log-likelihood of the correct source

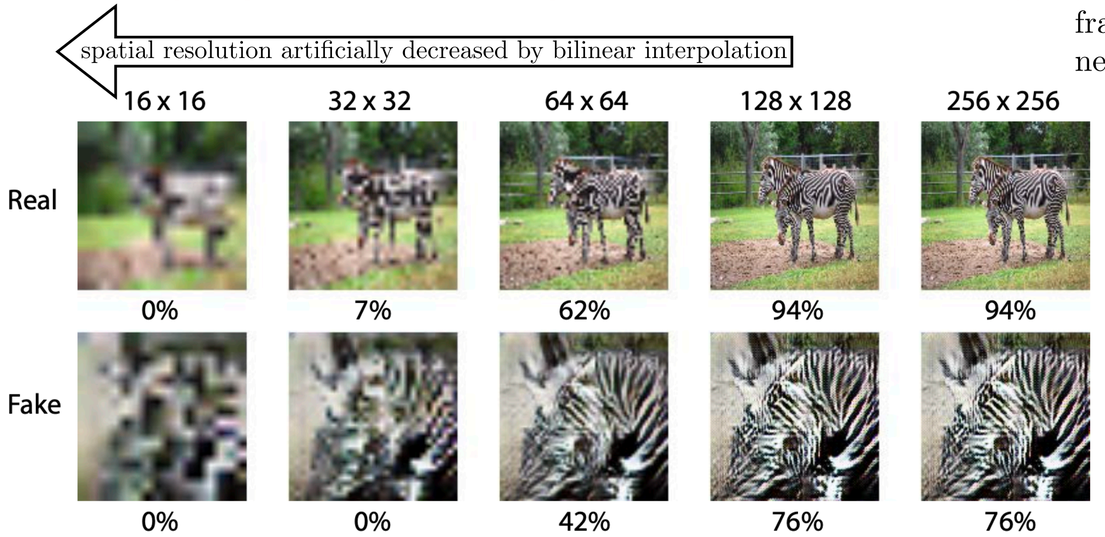
$L_C \rightarrow$ log-likelihood of the correct class

$$D^* = \arg\max_D L_S + L_C$$

$$G^* = \arg\max_G L_C - L_S$$

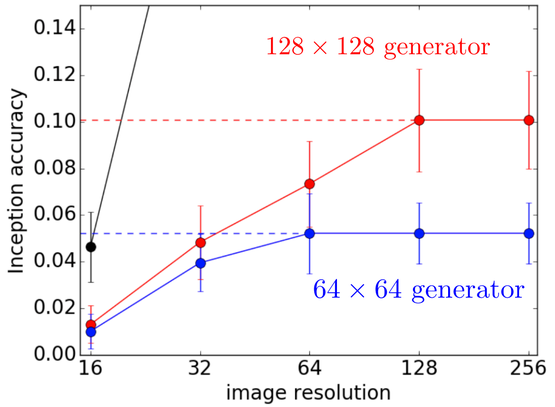
All ImageNet experiments are conducted using an ensemble of 100 AC-GANs, each trained on a 10-class split.

Generating High Resolution Images Improves Discriminability

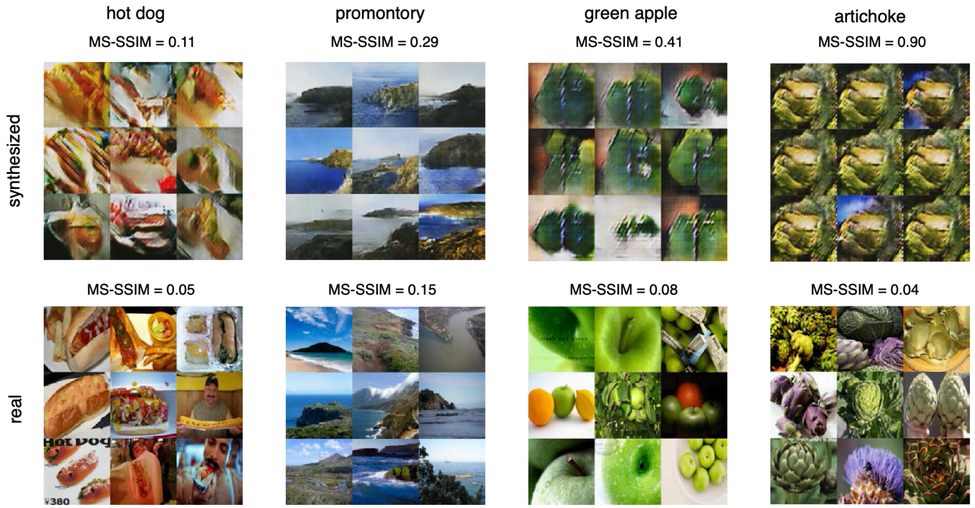


Synthesized Images ▷ Inception Model

fraction of images for which the Inception network assigns the correct label



Measuring the Diversity of Generated Images



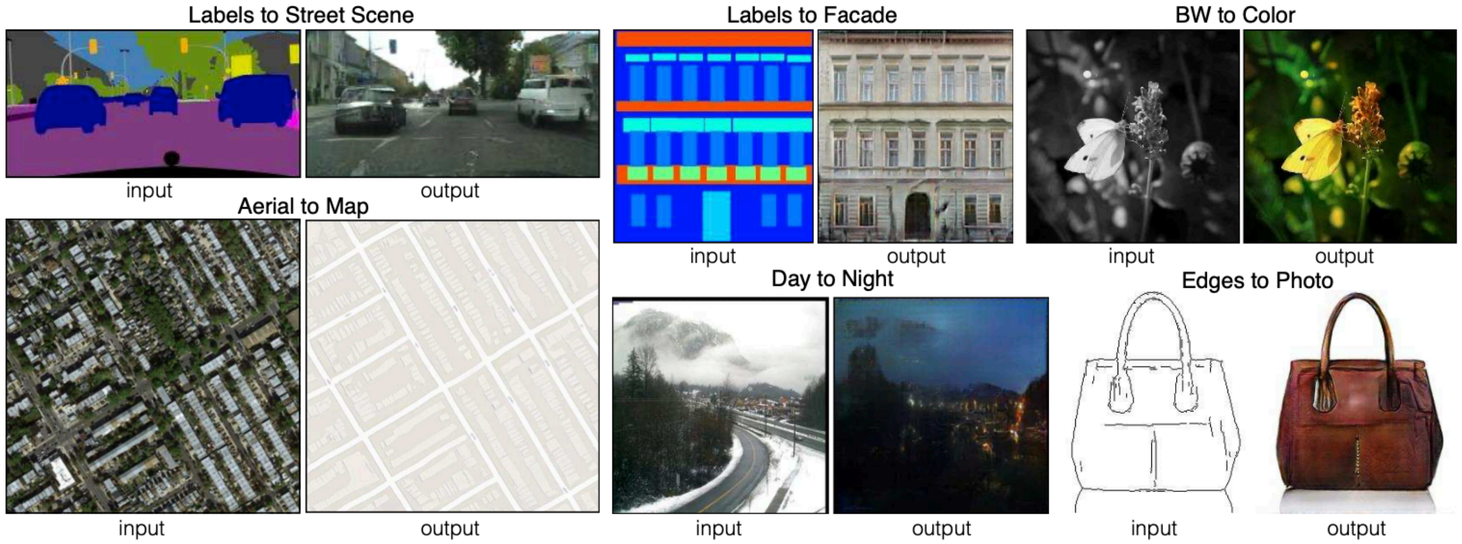
Multi-Scale Structural Similarity (MS-SSIM) attempts to discount aspects of an image that are not important for human perception. MS-SSIM values range between 0.0 and 1.0; higher MS-SSIM values correspond to perceptually more similar images.

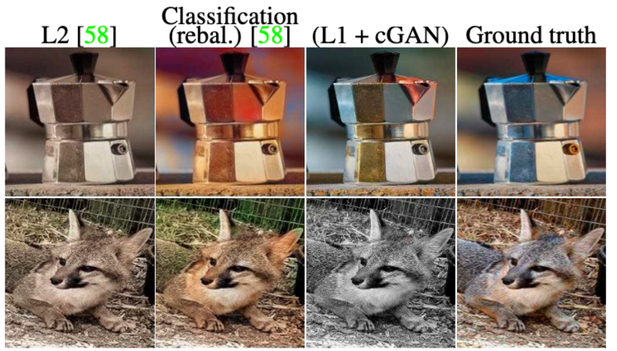
As a proxy for image diversity, measure the MS-SSIM scores between 100 randomly chosen pairs of images within a given class. Samples from classes that have higher diversity result in lower mean MS-SSIM scores.

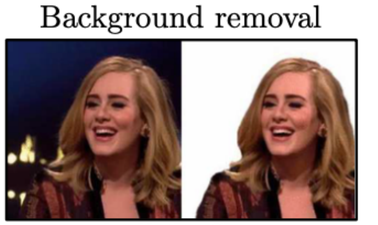
Odena, Augustus, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier gans." *International conference on machine learning*. PMLR, 2017.

Image-to-Image Translation with Conditional Adversarial Networks

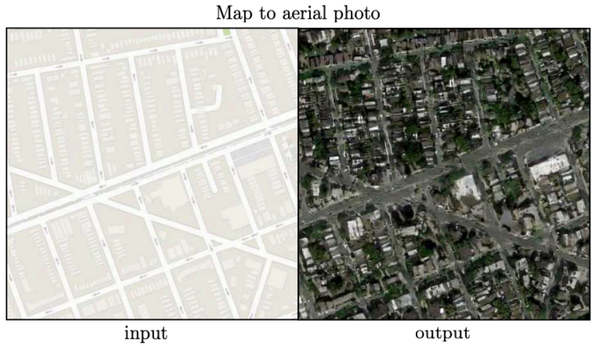
YouTube Playlist







by Kaihu Chen



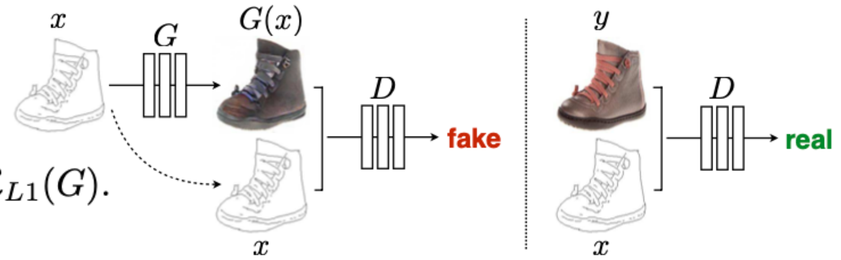
by Brannon Dorsey

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))], \quad \rightarrow \text{objective of a conditional GAN}$$

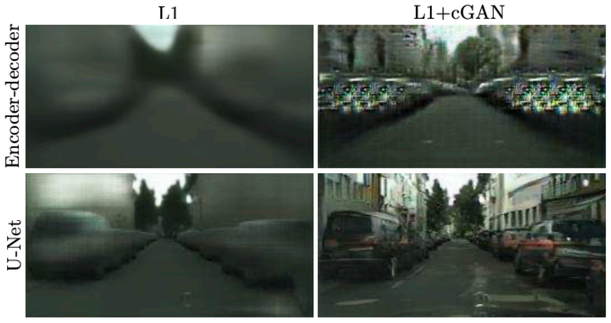
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D)$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1].$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G).$$



$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{y}[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,z))] \quad \rightarrow \text{an unconditional variant}$$

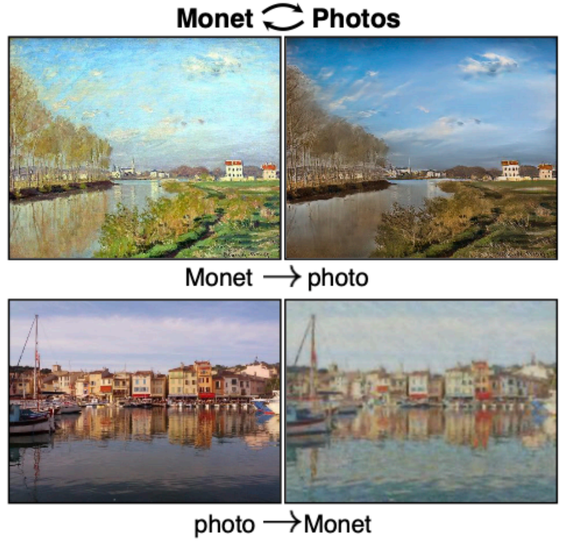


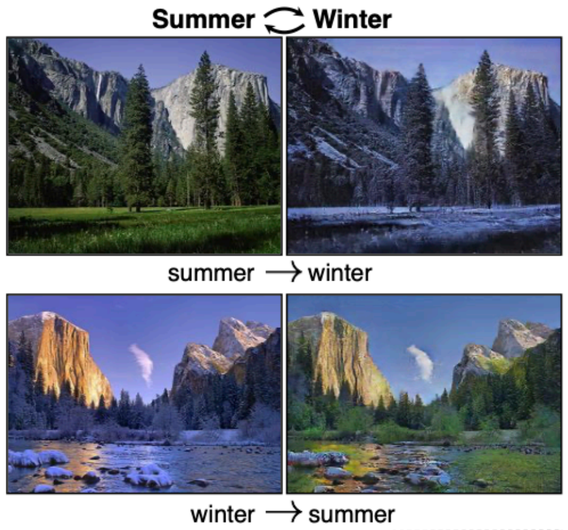
Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

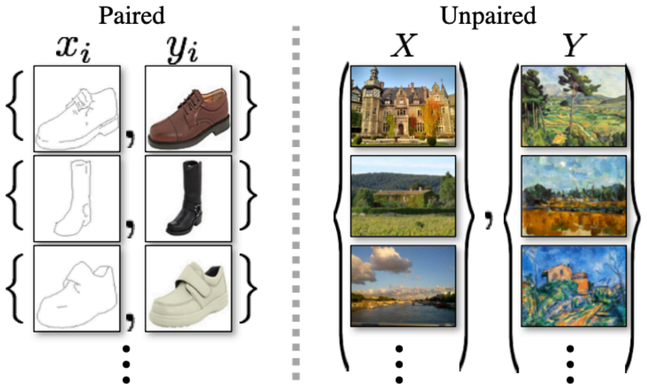
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Boulder

YouTube Playlist







$$G : X \rightarrow Y$$

$$F : Y \rightarrow X$$

$$F(G(x)) \approx x \text{ and } G(F(y)) \approx y.$$

$D_X \rightarrow$ aims to discriminate between $\{x\}$ and $\{F(y)\}$

$D_Y \rightarrow$ aims to discriminate between $\{y\}$ and $\{G(x)\}$

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

$$\min_G \max_{D_Y} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$$

$$\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X).$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

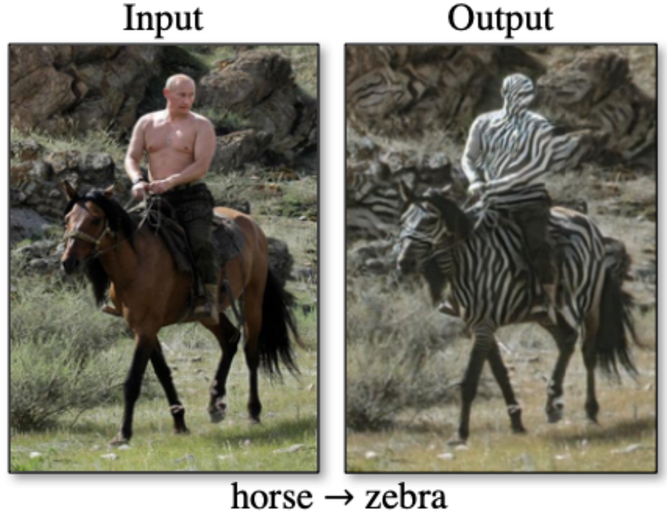
$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y).$$

Least-squares Loss

$$\min_G \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2]$$

$$\min_D \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(G(x))^2]$$

update the discriminators using a history of generated images



Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.

Unsupervised Image-to-Image Translation Networks

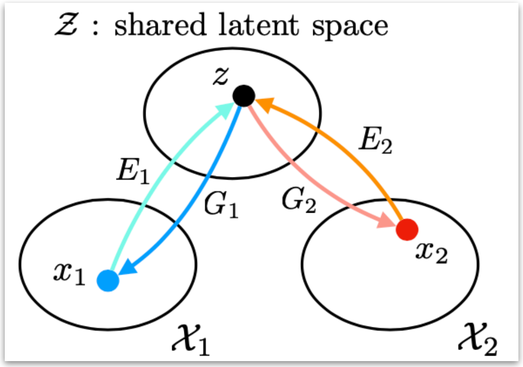
YouTube Playlist

UNIT: UNsupervised Image-to-image Translation Networks

Goal: To learn a joint distribution of images in different domains by using images from the marginal distributions in individual domains.

Since an infinite set of possible joint distributions can yield the given marginal distributions, we could infer nothing about the joint distribution from the marginal samples without additional assumptions.

Shared Latent Space Assumption



$z = E_1(x_1) = E_2(x_2)$

$x_1 = G_1(z) \quad x_2 = G_2(z)$

Shared latent space constraint implies cycle-consistency!

$F^{1\to2} := G_2 \circ E_1 \qquad F^{2\to1} := G_1 \circ E_2$

$x_1 = F^{2\to1}(F^{1\to2}(x_1))$

$x_2 = F^{1\to2}(F^{2\to1}(x_2))$

$\{E_1, E_2\} \to$ encoders, $\{G_1, G_2\} \to$ generators, $\{D_1, D_2\} \to$ discriminators

Variational Autoencoders (VAEs)

$\{E_1, G_1\} \to$ VAE for the $\mathcal{X}_1$ domain (VAE_1)

$q_1(z_1|x_1) := \mathcal{N}(z_1|E_1(x_1), I)$

$x_1^{1\to1} = G_1(\underbrace{z_1 \sim q_1(z_1|x_1)}_{\text{re-parametrization trick } z_1 = E_1(x_1) + \eta, \eta \sim \mathcal{N}(0,I)})$ $\to$ reconstructed image

$\mathcal{L}_{\text{VAE}_1}(E_1, G_1) = \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1\sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)]$

$p_\eta(z) = \mathcal{N}(z|0, I)$

$p_{G_1} \to$ Laplace distribution (equivalent to minimizing the absolute distance)

Weight Sharing

$H, L \to$ High, Low level features

$E_1 = E_H \circ E_{L,1}, E_2 = E_H \circ E_{L,2} \qquad G_1 = G_{L,1} \circ G_H, G_2 = G_{L,2} \circ G_H$

Note that the weight-sharing constraint alone does not guarantee that corresponding images in two domains will have the same latent code.

Generative Adversarial Networks (GANs)

$\{G_1, D_1\} \to \text{GAN}_1$

G_1 can generate two types of images:

– reconstruction stream (supervised training): $x_1^{1\to1} = G_1(z_1 \sim q_1(z_1|x_1))$

– translation stream (adversarial training): $x_2^{2\to1} = G_1(z_2 \sim q_2(z_2|x_2))$

$\mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) = \lambda_0 \mathbb{E}_{x_1\sim P_{\mathcal{X}_1}}[\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2\sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))]$

Cycle-consistency (CC)

$\mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) = \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1\to2}))||p_\eta(z)) - \lambda_4 \mathbb{E}_{z_2\sim q_2(z_2|x_1^{1\to2})}[\log p_{G_1}(x_1|z_2)]$

Learning

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{\text{VAE}_1}(E_1, G_1) + \mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) + \mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2)$$
$$\mathcal{L}_{\text{VAE}_2}(E_2, G_2) + \mathcal{L}_{\text{GAN}_2}(E_2, G_2, D_2) + \mathcal{L}_{\text{CC}_2}(E_2, G_2, E_1, G_1)$$

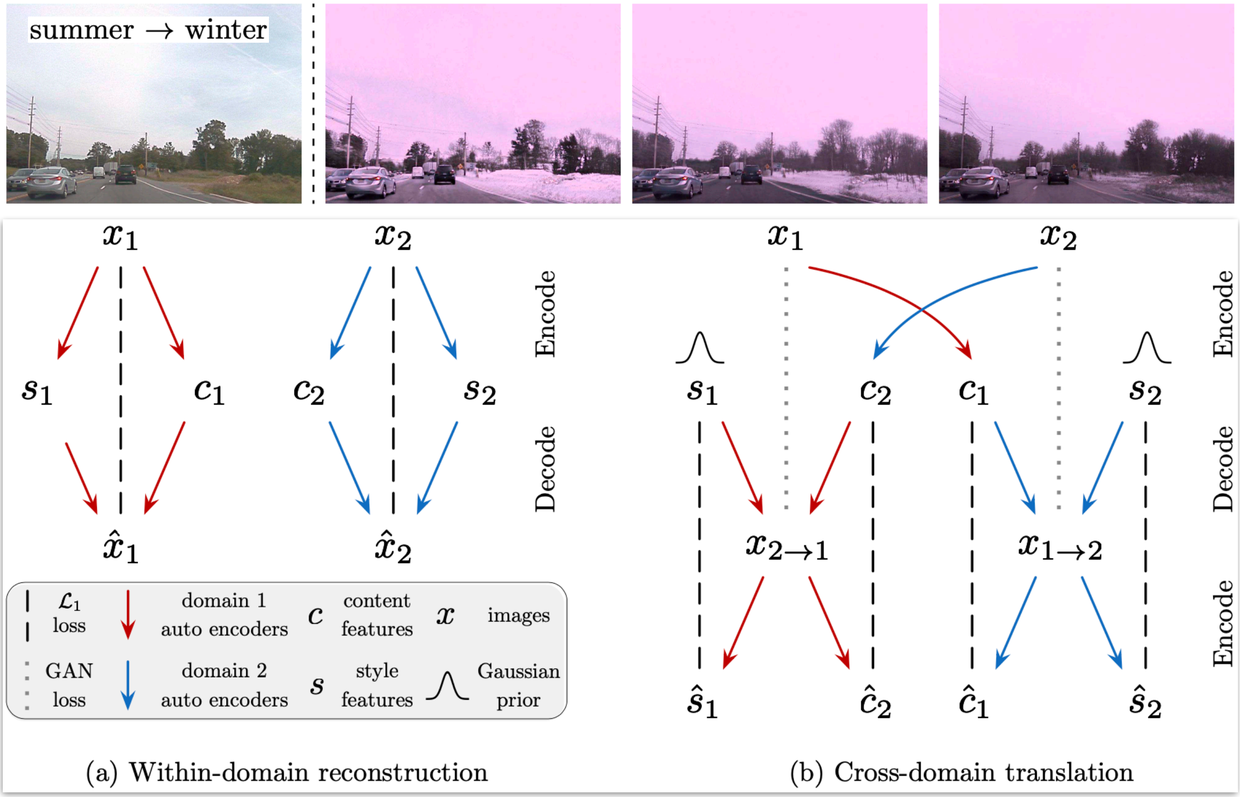
Translation

$F_{1\to2}(x_1) = G_2(z_1 \sim q_1(z_1|x_1)) \qquad F_{2\to1}(x_2) = G_1(z_2 \sim q_2(z_2|x_2))$

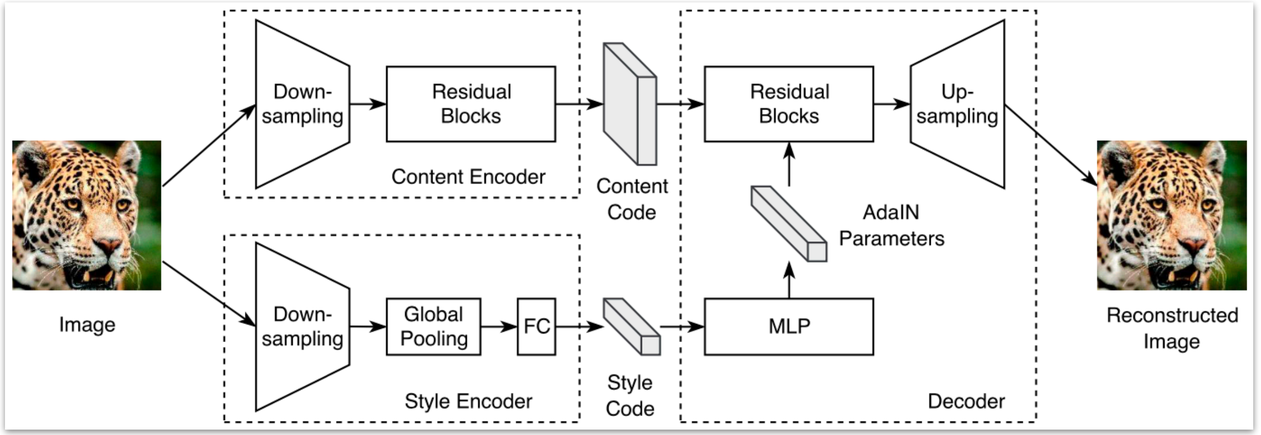
Liu, Ming-Yu, Thomas Breuel, and Jan Kautz. "Unsupervised image-to-image translation networks." *Advances in neural information processing systems*. 2017.

Multimodal Unsupervised Image-to-Image Translation

YouTube Video



(a) Within-domain reconstruction (b) Cross-domain translation



$\text{AdaIN}(z, \gamma, \beta) = \gamma \left(\frac{z - \mu(z)}{\sigma(z)} \right) + \beta \rightarrow$ Adaptive Instance Normalization

γ and $\beta \rightarrow$ parameters generated by the MLP

Loss

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}(E_1, E_2, G_1, G_2, D_1, D_2) = \mathcal{L}_{\text{GAN}}^{x_1} + \mathcal{L}_{\text{GAN}}^{x_2} + \lambda_x(\mathcal{L}_{\text{recon}}^{x_1} + \mathcal{L}_{\text{recon}}^{x_2}) + \lambda_c(\mathcal{L}_{\text{recon}}^{c_1} + \mathcal{L}_{\text{recon}}^{c_2}) + \lambda_s(\mathcal{L}_{\text{recon}}^{s_1} + \mathcal{L}_{\text{recon}}^{s_2})$$

E_i and $G_i \rightarrow$ encoder and decoder for each domain $\mathcal{X}_i$ $(i = 1, 2)$

$(c_i, s_i) = (E_i^c(x_i), E_i^s(x_i)) = E_i(x_i) \rightarrow$ content and style codes

Translate image $x_1 \in \mathcal{X}_1$ to $\mathcal{X}_2$:

$c_1 = E_1^c(x_1)$

$s_2 \sim q(s_2) = \mathcal{N}(0, I)$

$x_{1\rightarrow 2} = G_2(c_1, s_2)$

(Conditional) Inception Score

$$\text{CIS} = \mathbb{E}_{x_1 \sim p(x_1)}[\mathbb{E}_{x_{1\rightarrow 2} \sim p(x_{2\rightarrow 1}|x_1)}[\text{KL}(p(y_2|x_{1\rightarrow 2})||p(y_2|x_1))]]$$

$$\text{IS} = \mathbb{E}_{x_1 \sim p(x_1)}[\mathbb{E}_{x_{1\rightarrow 2} \sim p(x_{2\rightarrow 1}|x_1)}[\text{KL}(p(y_2|x_{1\rightarrow 2})||p(y_2))]]$$

$$\mathcal{L}_{\text{recon}}^{x_1} = \mathbb{E}_{x_1 \sim p(x_1)}[||G_1(E_1^c(x_1), E_1^s(x_1)) - x_1||_1]$$

$$\mathcal{L}_{\text{recon}}^{c_1} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[||E_2^c(G_2(c_1, s_2)) - c_1||_1]$$

$$\mathcal{L}_{\text{recon}}^{s_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[||E_2^s(G_2(c_1, s_2)) - s_2||_1]$$

$p(c_1)$ is given by $c_1 = E_1^c(x_1)$ and $x_1 \sim p(x_1)$

$$\mathcal{L}_{\text{GAN}}^{x_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[\log(1 - D_2(G_2(c_1, s_2)))] + \mathbb{E}_{x_2 \sim p(x_2)}[\log D_2(x_2)]$$

Huang, Xun, et al. "Multimodal unsupervised image-to-image translation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

YouTube Playlist

$$l^{SR} = \underbrace{\underbrace{l^{SR}_{\text{X}}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

$$l^{SR}_{MSE} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2$$

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

$I^{SR} \rightarrow$ high-resolution, super-resolved image

$I^{LR} \rightarrow$ low-resolution input image

$I^{HR} \rightarrow$ high-resolution counterpart of I^{LR}

$I^{HR} \triangleright$ Gaussian Filter $\triangleright$ downsampling(downsampling_factor$= r$)

$C \rightarrow$ number of color channels

$I^{LR} \in \mathbb{R}^{W \times H \times C}$

$I^{SR}, I^{HR} \in \mathbb{R}^{rW \times rH \times C}$

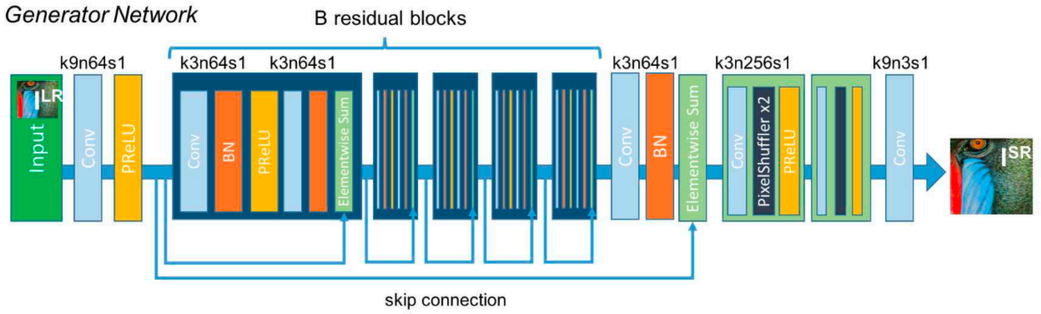
evaluation metrics $\rightarrow$
- PSNR: peak signal-to-noise ratio
- MOS: mean opinion score
- SSIM: structural similarity

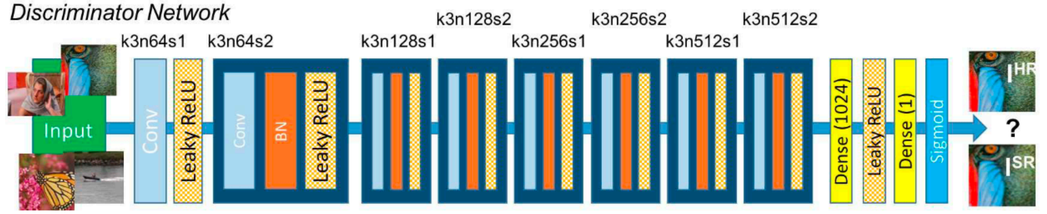
$G_{\theta_G} : I^{LR} \mapsto I^{HR} \rightarrow$ generator

$\phi_{i,j} \rightarrow$ feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network

$$\hat{\theta}^G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} \ell^{SR}(G_{\theta_G}(I^{LR}_n), I^{HR}_n) \rightarrow \text{loss function}$$

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))]$$





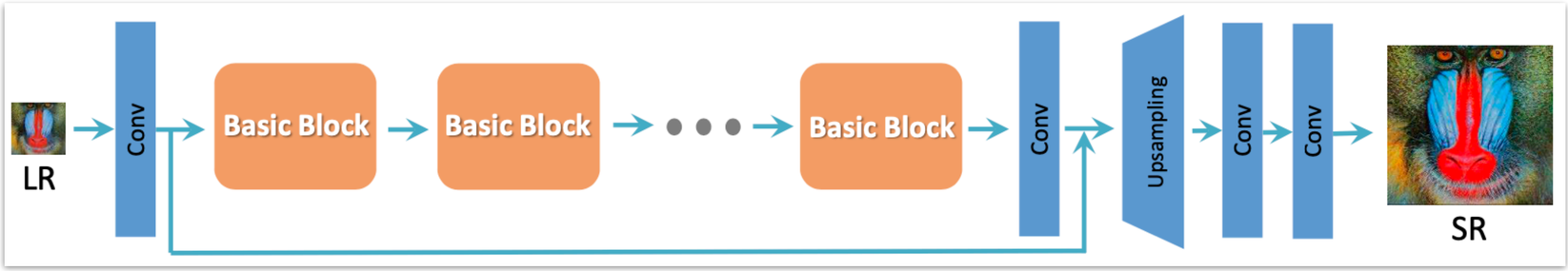
Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

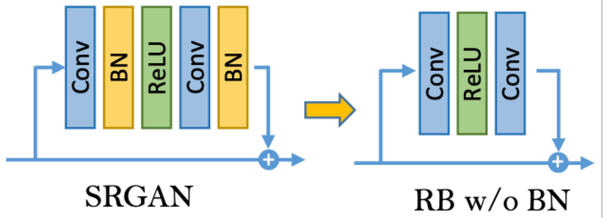
Boulder

YouTube Video

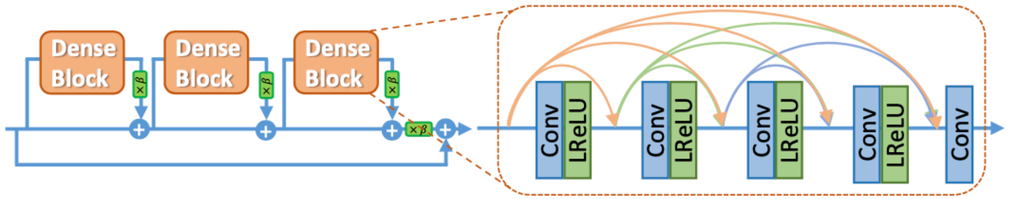
Network Architecture



Residual Block (RB)



Residual in Residual Dense Block (RRDB)



Relativistic Discriminator

$D(x_r) = \sigma(C(\text{Real})) \rightarrow 1$ Real?

$D(x_f) = \sigma(C(\text{Fake})) \rightarrow 0$ Fake?

a) Standard GAN

$D_{Ra}(x_r, x_f) = \sigma(C(\text{Real}) - \mathbb{E}[C(\text{Fake})]) \rightarrow 1$ More realistic than fake data?

$D_{Ra}(x_f, x_r) = \sigma(C(\text{Fake}) - \mathbb{E}[C(\text{Real})]) \rightarrow 0$ Less realistic than real data?

b) Relativistic GAN

Judge "whether one image is more realistic than the other" rather than "whether one image is real or fake"

$D(x) = \sigma(C(x)) \rightarrow$ standard discriminator

$C(x) \rightarrow$ non-transformed discriminator output

$D_{\text{Ra}} \rightarrow$ relativistic average discriminator (RaD)

$D_{\text{Ra}}(x_r, x_f) = \sigma(C(x_r) - \mathbb{E}_{x_f}[C(x_f)])$

all fake data in the mini-batch

$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_f, x_r))]$

$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))]$

$x_f = G(x_i)$ and x_i stands for the input LR image

Perceptual Loss

Features before activation rather than after activation!

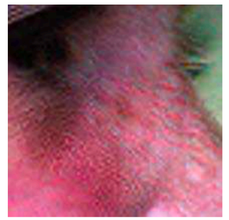
$L_G = L_{\text{percep}} + \lambda L_G^{Ra} + \eta L_1$

$L_1 = \mathbb{E}_{x_i} ||G(x_i) - y||_1 \rightarrow$ content loss

Network Interpolation

$\theta_G^{\text{INTERP}} = (1 - \alpha)\ \theta_G^{\text{PSNR}} + \alpha\ \theta_G^{\text{GAN}}$

face from Set14
(PSNR / Percpetual Index)



SRGAN
(30.28 / 4.47)

ESRGAN(ours)
(30.50 / 3.64)

artifacts

Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks."
Proceedings of the European conference on computer vision (ECCV) workshops. 2018.

High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs

YouTube Playlist

The pix2pix baseline

$G \rightarrow$ generator

$D \rightarrow$ discriminator

$G \rightarrow$ translate semantic label maps to realistic looking images

$D \rightarrow$ distinguish real images from the translated ones

$\{(s_i, x_i)\} \rightarrow$ training data

$s_i \rightarrow$ semantic label map

$x_i \rightarrow$ corresponding natural photo

$$\min_G \max_D \mathcal{L}_{\text{GAN}}(G, D)$$

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(s,x)}[\log D(s, x)] + \mathbb{E}_s[\log(1 - D(s, G(s)))]$$

U-Net as the generator

patch-based fully convolutional network as the discriminator

256×256

Improving photorealism and resolution 2048×1024

– coarse-to-fine generator

– multi-scale discriminator

– improved adversarial loss

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right)$$

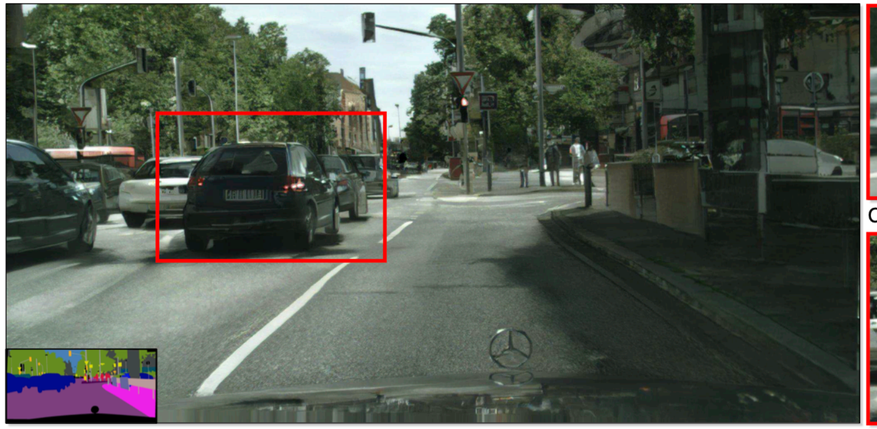
feature matching

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))||_1]$$

$D_k^{(i)} \rightarrow i$-th layer feature extractor of discriminator D_k

$T \rightarrow$ total number of layers

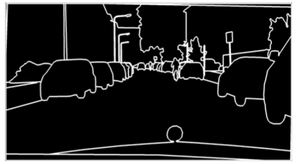
$N_i \rightarrow$ number of elements in each layer



(a) Synthesized result

Cascaded refinement network [5]

Our result



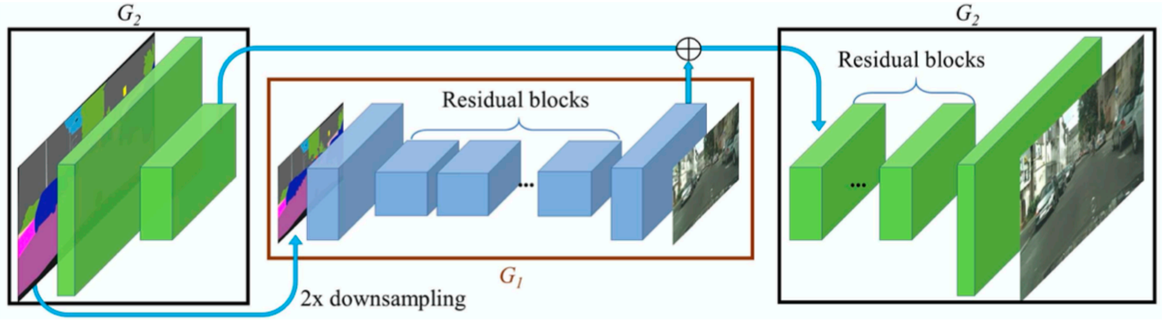
(a) Boundary map

(b) w/o boundary map

(c) w/ boundary map

(b) Application: Change label types

(c) Application: Edit object appearance



Semantic segmentation scores

	pix2pix [21]
Pixel acc	78.34
Mean IoU	0.3948

CRN [5]	Ours	Oracle
70.55	**83.78**	84.29
0.3483	**0.6389**	0.6857

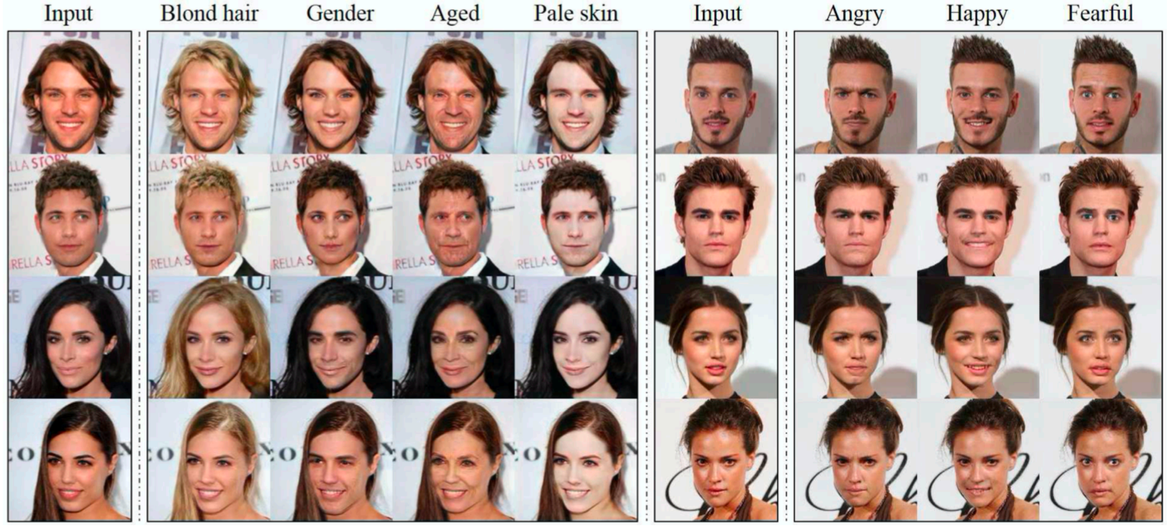
original images

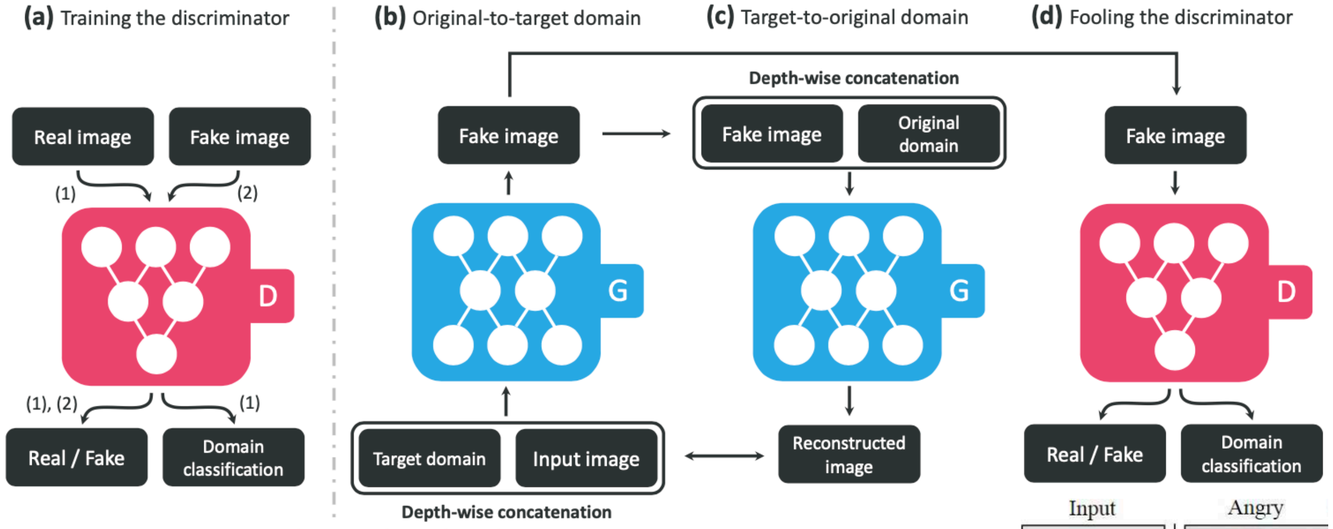
Wang, Ting-Chun, et al. "High-resolution image synthesis and semantic manipulation with conditional gans." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

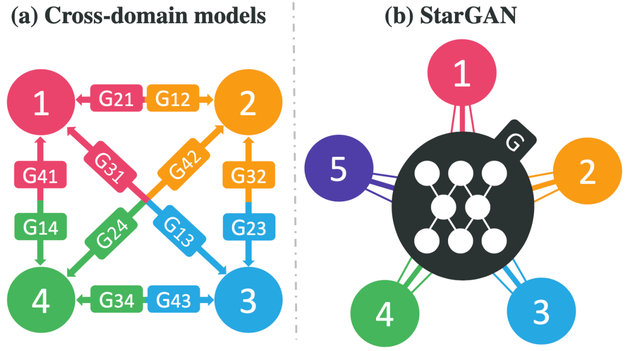
StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

Boulder

YouTube Playlist







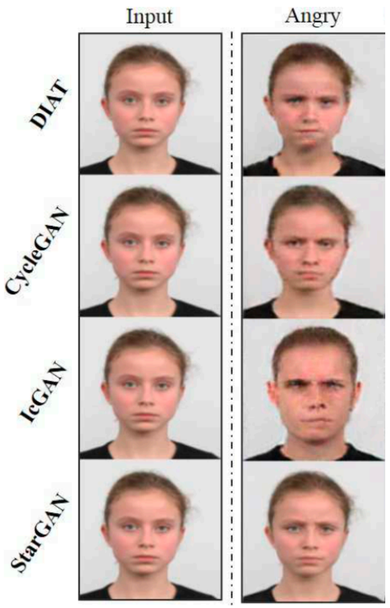
$G \rightarrow$ single generator
$x \rightarrow$ input image
$y \rightarrow$ output image
$c \rightarrow$ target domain label
$G(x, c) \rightarrow$ generated image
$D : x \longmapsto \{D_{src}(x), D_{cls}(x)\}$
$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^r_{cls}$
$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}_{rec}$

$\mathcal{L}_{adv} = \mathbb{E}_y[\log D_{src}(y)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x, c)))] \rightarrow$ adversarial loss
$\mathcal{L}^r_{cls} = \mathbb{E}_{y,c}[-\log D_{cls}(c|y)] \rightarrow$ domain classification loss of real images
$\mathcal{L}^f_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x, c))] \rightarrow$ fake images
$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[\|x - G(G(x, c), c')\|_1] \rightarrow$ reconstruction loss
$\lambda_{cls} = 1$ & $\lambda_{rec} = 10$

Training with multiple datasets
$n \rightarrow$ number of datasets (e.g., $n = 2$)
CelebA and RaFD
$m \rightarrow$ mask vector (n-dimensional one-hot vector)
$\tilde{c} := [c_1, \ldots, c_n, m] \rightarrow$ concatenate
$c_i \rightarrow$ vector for labels of the i-th dataset
assign zero values to the remaining $n - 1$
Improved GAN training unknown labels

$$\mathcal{L}_{adv} = \mathbb{E}_y[D_{src}(y)] - \mathbb{E}_{x,c}[D_{src}(G(x, c))] - \lambda_{gp}\mathbb{E}_{\hat{y}}[(\|\nabla_{\hat{y}} D_{src}(\hat{y})\|_2 - 1)^2]$$

$\hat{y} \rightarrow$ sampled uniformly along a straight line between y and $G(x, c)$
$\lambda_{gp} = 10 \rightarrow$ gradient penalty



Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Semantic Image Synthesis with Spatially-Adaptive Normalization

Boulder

YouTube Video

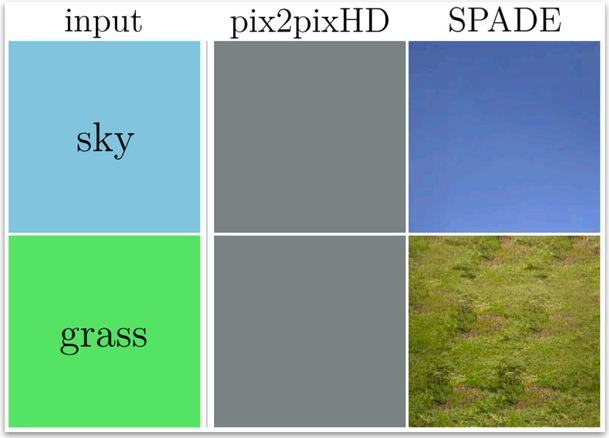
$$\mu_c^i = \frac{1}{NH^iW^i}\sum_{n,y,x} h^i_{n,c,y,x}$$

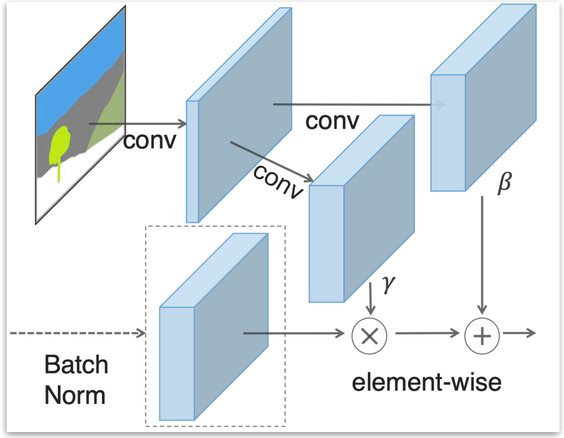
$$\sigma_c^i = \sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}\left((h^i_{n,c,y,x})^2 - (\mu_c^i)^2\right)}$$

SPatially-Adaptive (DE)normalization (SPADE)

Normalization layers tend to "wash away" semantic information!

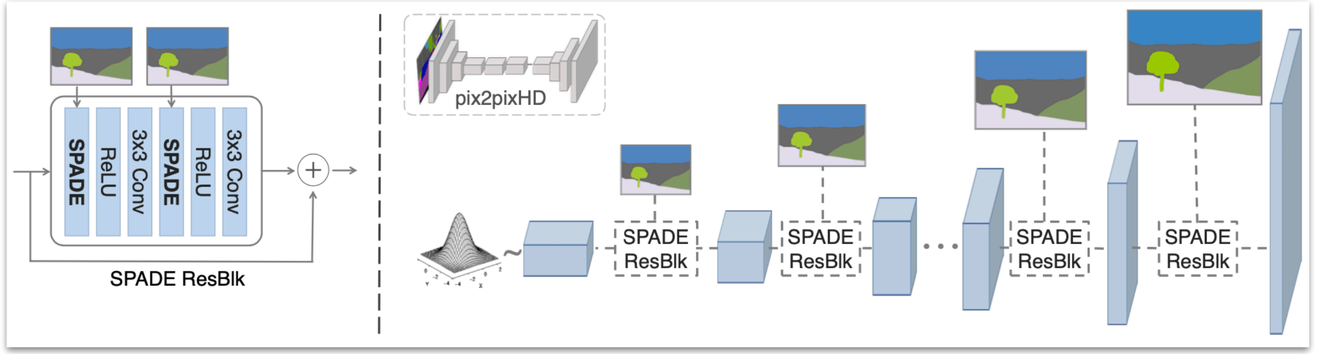
$\gamma^i_{c,y,x}(\mathbf{m})$ and $\beta^i_{c,y,x}(\mathbf{m})$ → learned modulation parameters of the normalization layer (functions of $\mathbf{m}, y, x$)





Conditional BatchNorm: Replacing the segmentation mask $\mathbf{m}$ with the image class label and making the modulation parameters spatially-invariant

AdaIN: Replacing $\mathbf{m}$ with a real image, making the modulation parameters spatially-invariant, and setting $N = 1$



$\mathbf{m} \in \mathbb{L}^{H\times W}$ → semantic segmentation mask

$\mathbb{L}$ → set of integers denoting the semantic labels

$\mathbf{h}^i \in \mathbb{R}^{N\times C^i\times H^i\times W^i}$ → activations of the i-th layer of a deep convolutional network for a batch of N samples

$\gamma^i_{c,y,x}(\mathbf{m})\dfrac{h^i_{n,c,y,x}-\mu^i_c}{\sigma^i_c} + \beta^i_{c,y,x}(\mathbf{m})$ → activation value at site $(n \in N, c \in C^i, y \in H^i, x \in W^i)$

$h^i_{n,c,y,x}$ → activation at the site before normalization

Same multi-scale discriminator and loss function used in pix2pixHD except replace the least squared loss term with the hinge loss term

Multi-modal synthesis

style ▷ encoder → random vector

For training, add a KL-Divergence loss term.

Park, Taesung, et al. "Semantic image synthesis with spatially-adaptive normalization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

Boulder

Questions?

YouTube Playlist