Carnegie Mellon University

Intermediate Deep Learning

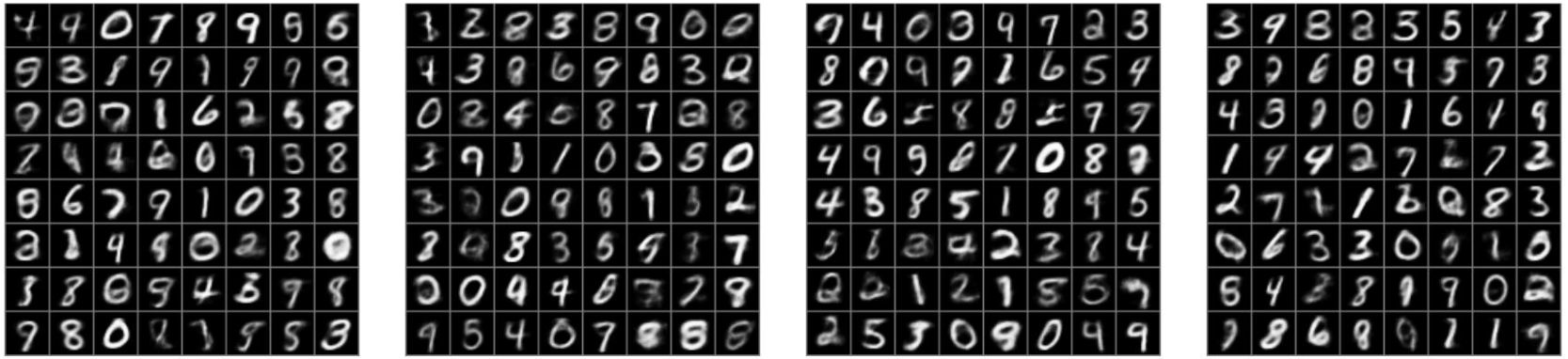
Spring 2025, Deep Learning for Engineers
March 13, 2025, 2nd Session

Amir Barati Farimani
Associate Professor of Mechanical Engineering and Bio-Engineering
Carnegie Mellon University

Variational Autoencoders

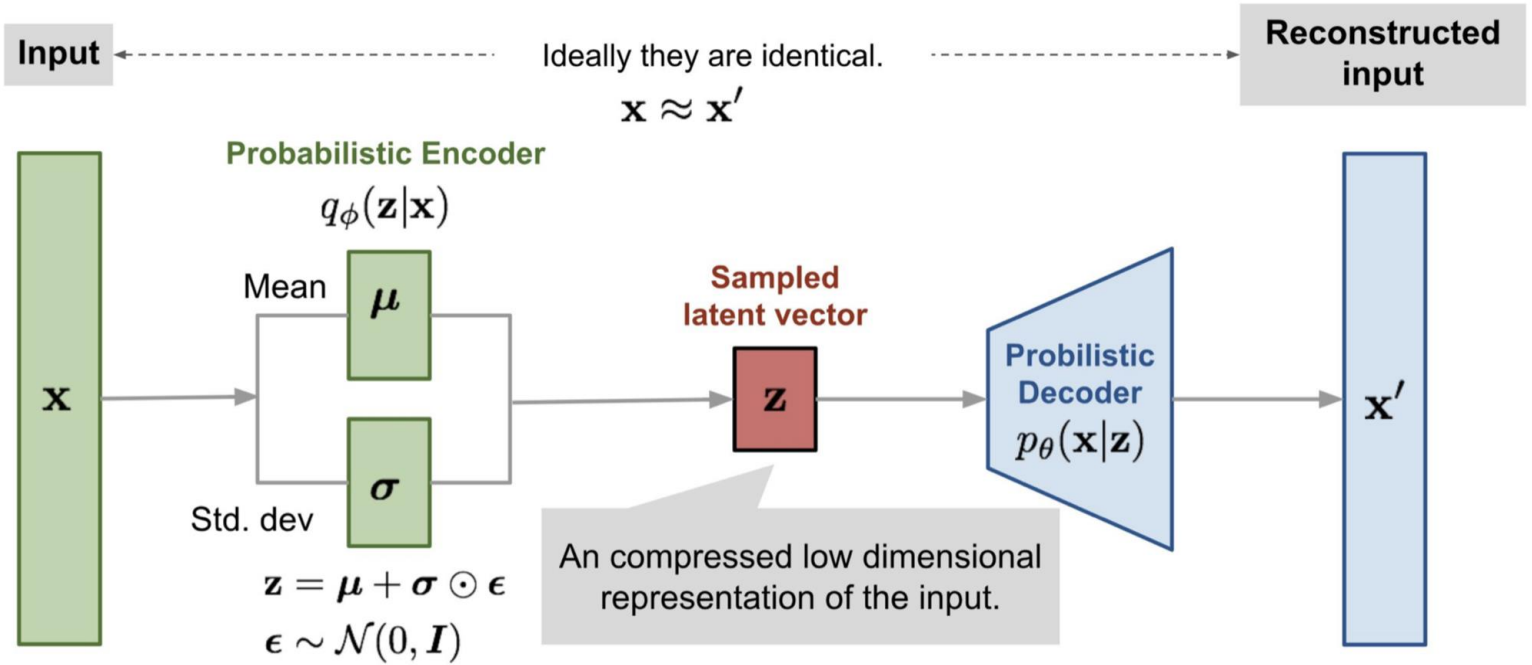
A PROBABILISTIC TWIST ON AUTOENCODERS THAT ENABLES:

- Novel image synthesis from random samples
- Transition from image to image or from mode to mode
- Aggregation of similar images to close locations in the latent space



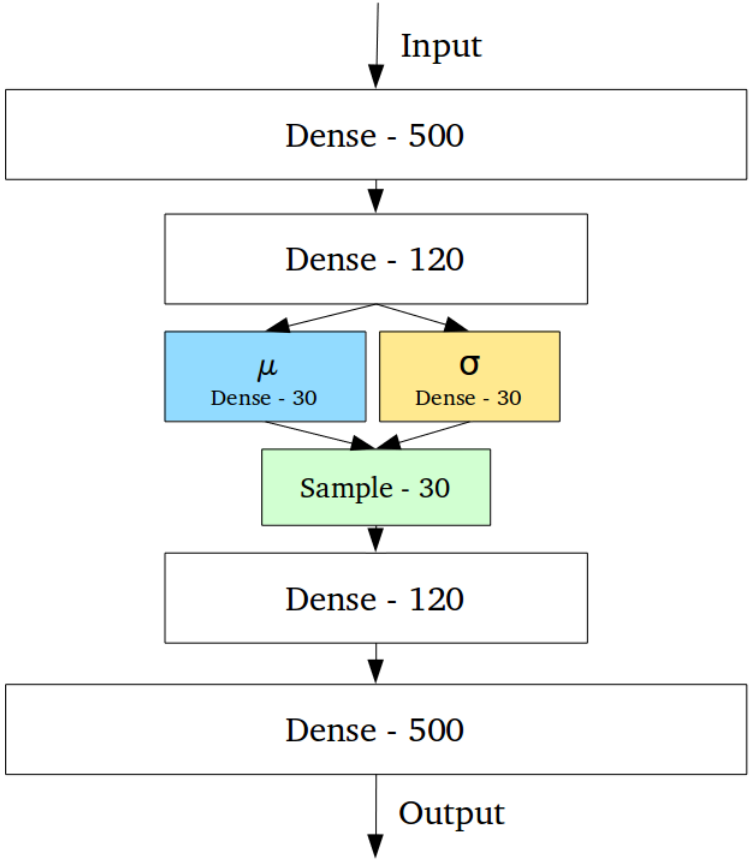
Carnegie Mellon University

Variational Auto-Encoder (VAE)



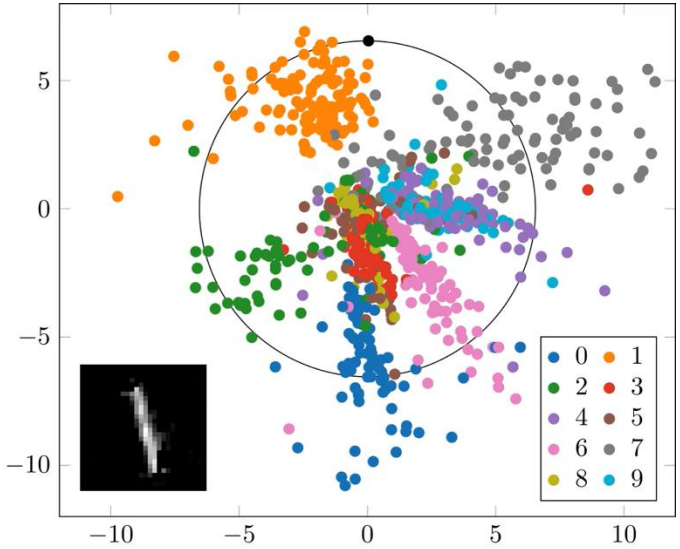
Architecture

- Very similar to the regular autoencoder
- The probabilistic nature of the VAE is enabled using a sampling layer

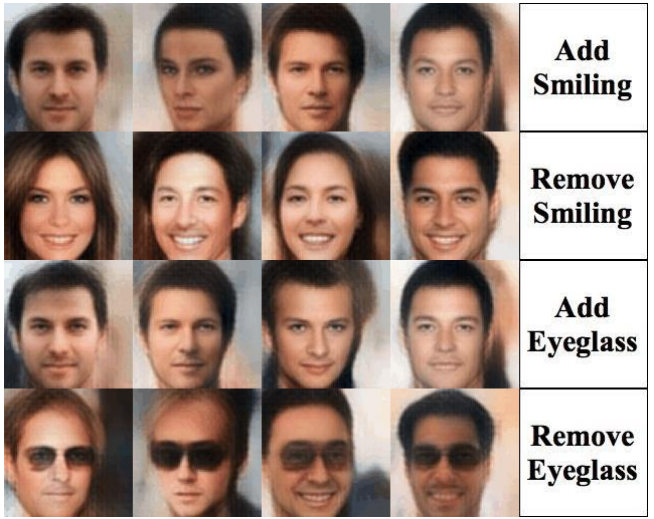


Motivation

VARIATIONAL AUTOENCODERS ARE A DEEP LEARNING TECHNIQUE FOR LEARNING USEFUL LATENT REPRESENTATIONS



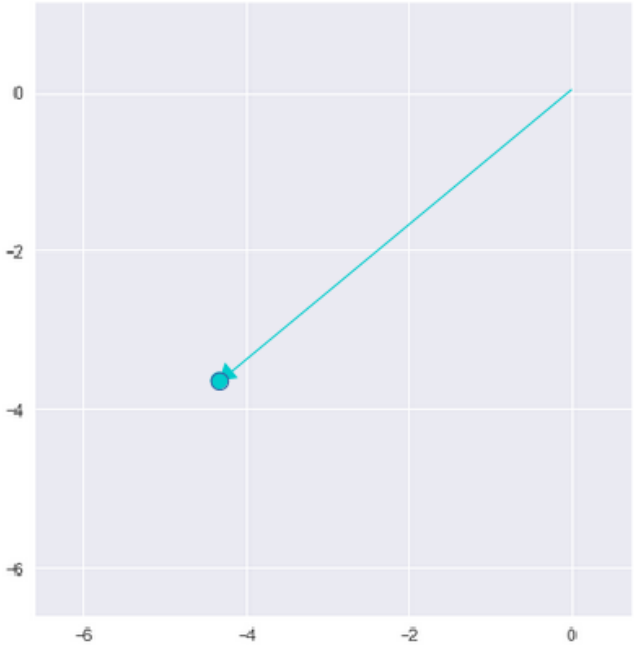
Regular Autoencoder on MNIST dataset



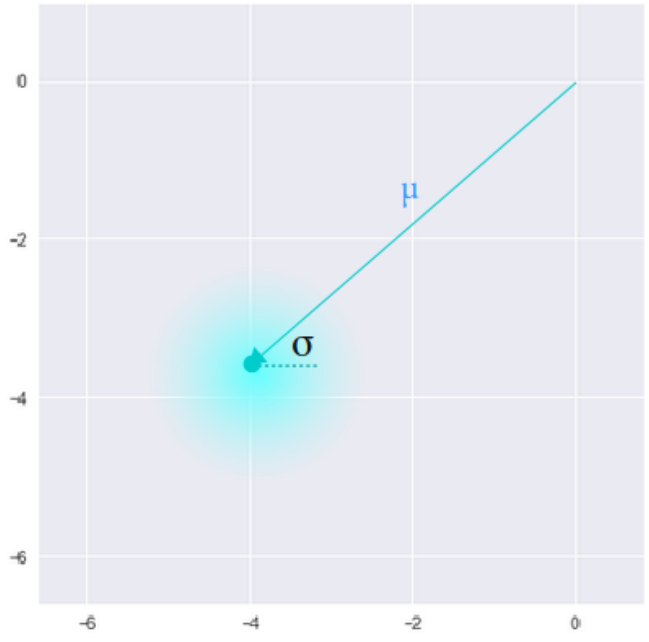
Variational Autoencoder on CelebA dataset

Carnegie Mellon University

VAE – Probabilistic Intuition



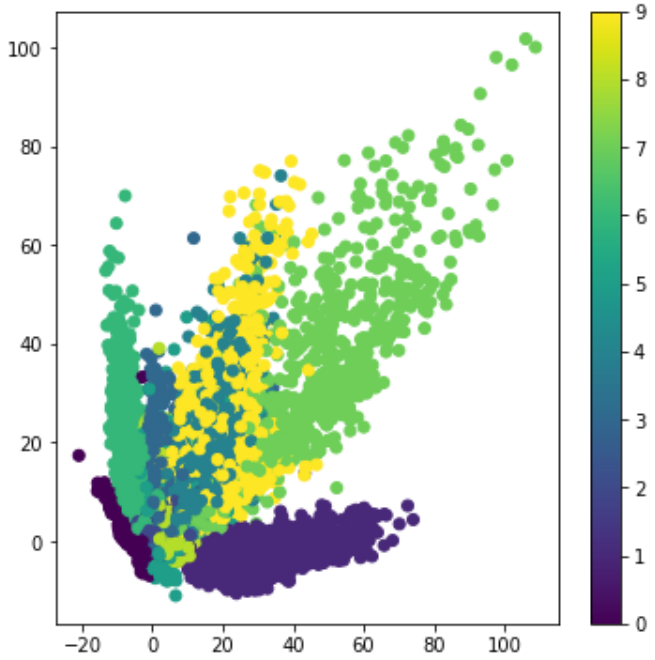
Standard Autoencoder
(direct encoding coordinates)



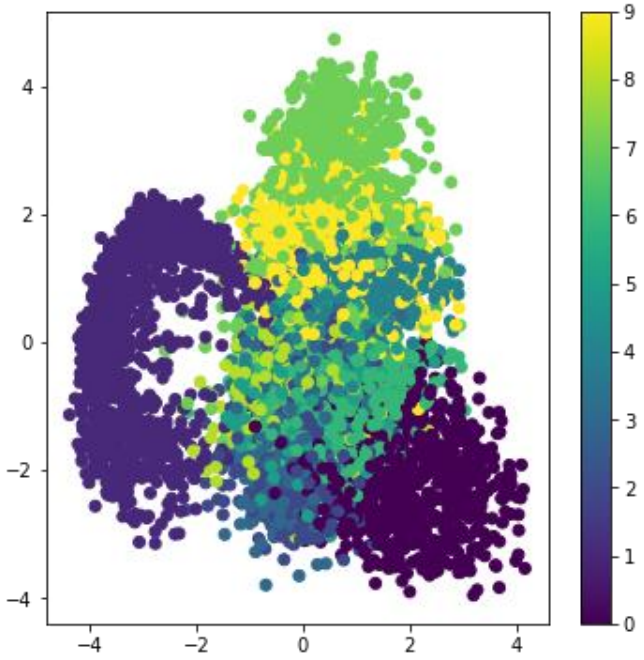
Variational Autoencoder
(μ and σ initialize a probability distribution)

Carnegie Mellon University

Latent Space Representation



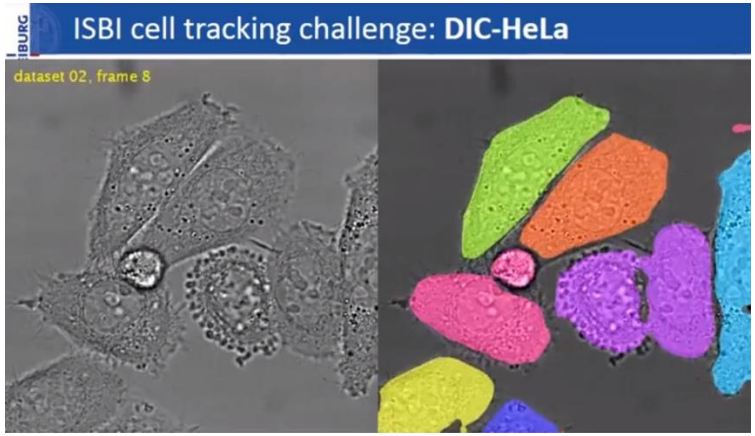
Autoencoder

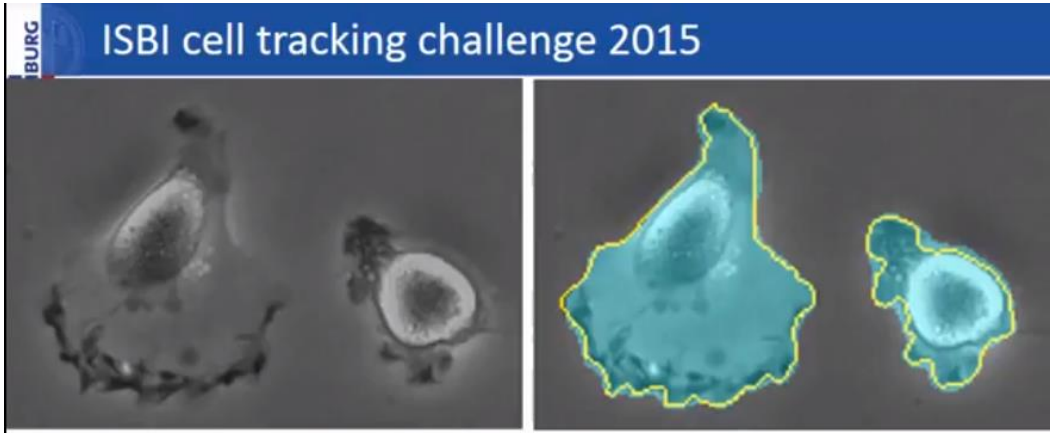


Variational Autoencoder

Carnegie Mellon University

What is Segmentation?





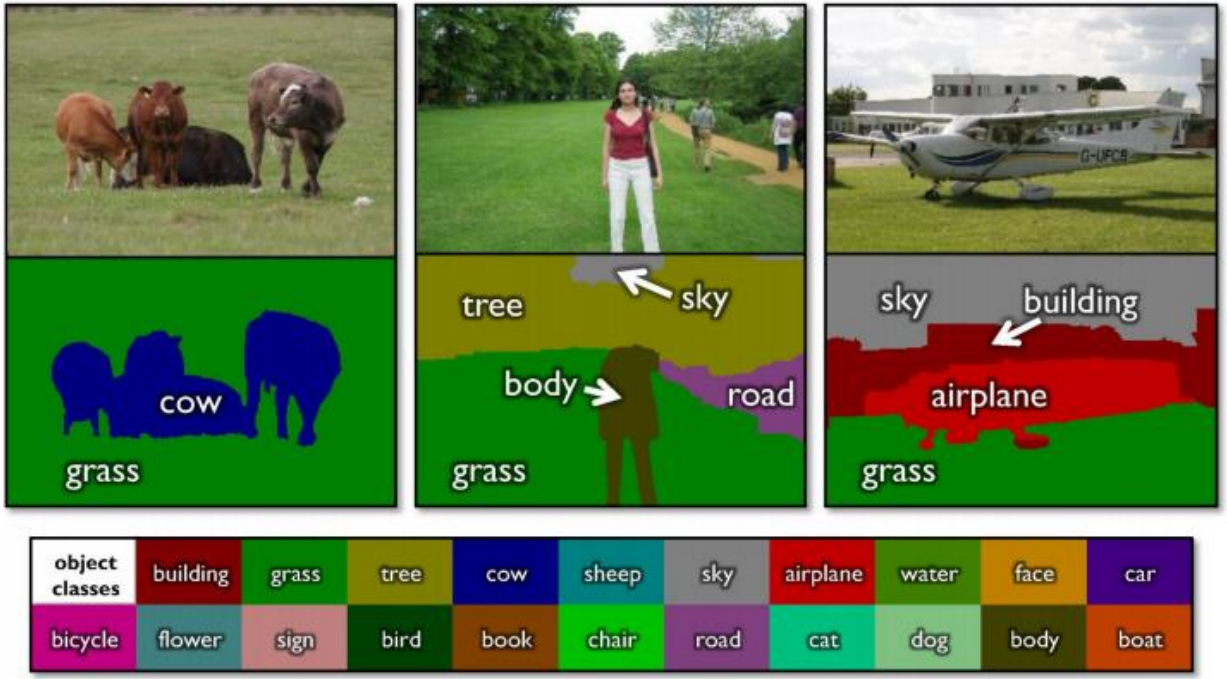
Carnegie Mellon University

Ronneberger et al. (2015) ISBI cell tracking challenge

Examples

- Label every pixel!
- Don't differentiate instances
- Classic computer vision problem

Pixel-wise Semantic Segmentation



http://cs231n.stanford.edu/slides/2016/winter1516_lecture13.pdf

Main Motivation

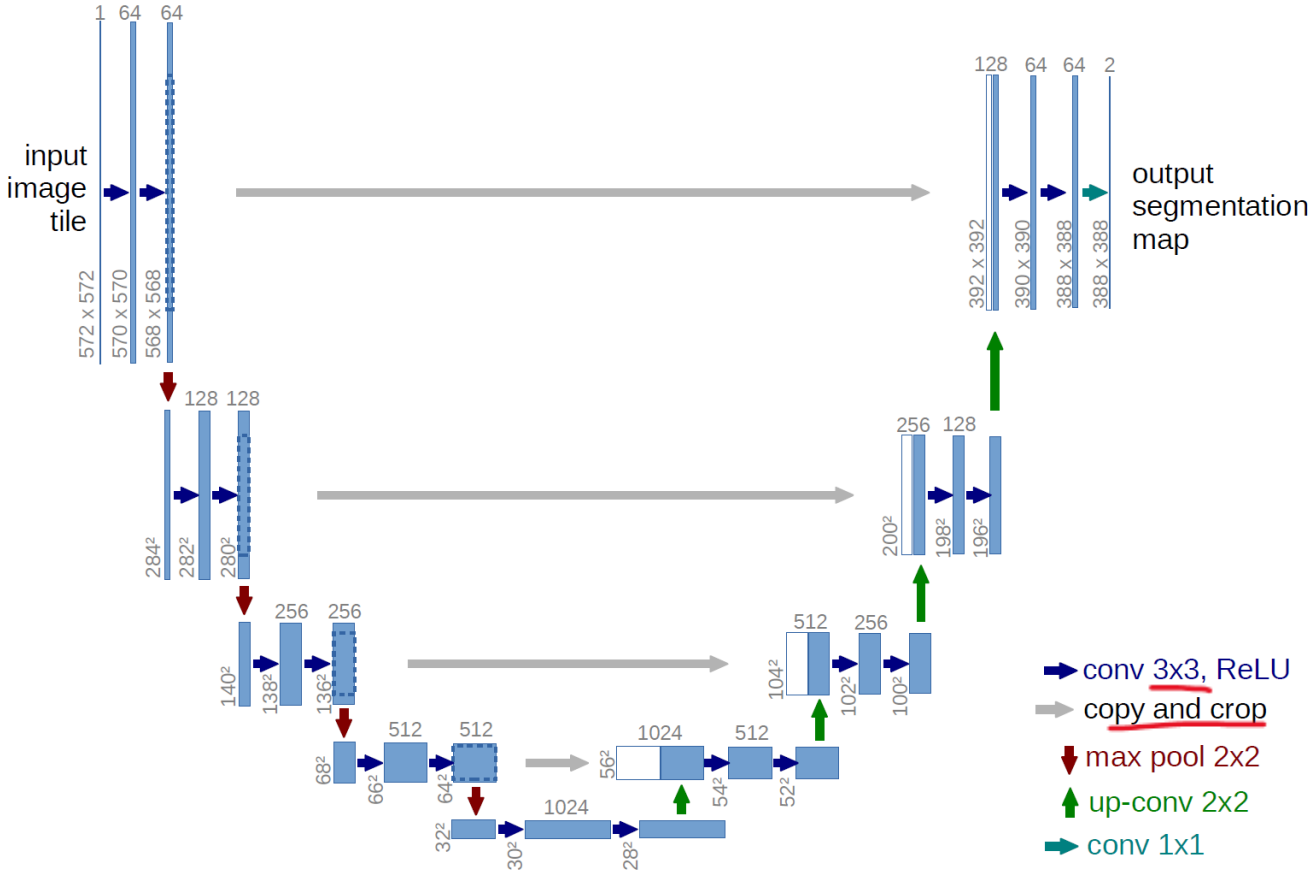
Biomedical Image Segmentation with U-net

- Thousands of training images are usually beyond reach in biomedical task
- For example, AlexNet:
  - 8 layers and millions of parameters on on the ImageNet dataset
  - 1 million training images
- The desired output should include localization

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Carnegie Mellon University

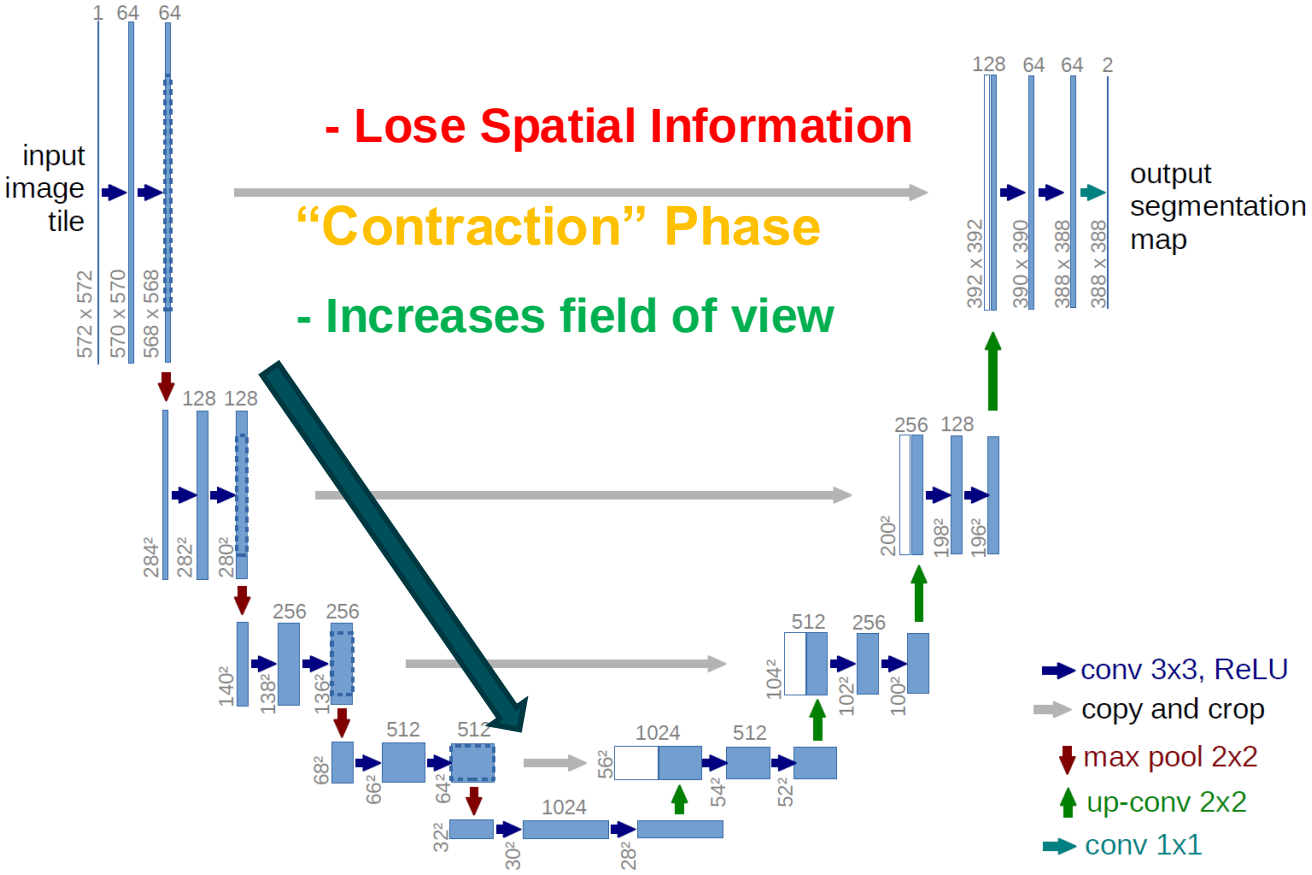
U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

Carnegie Mellon University

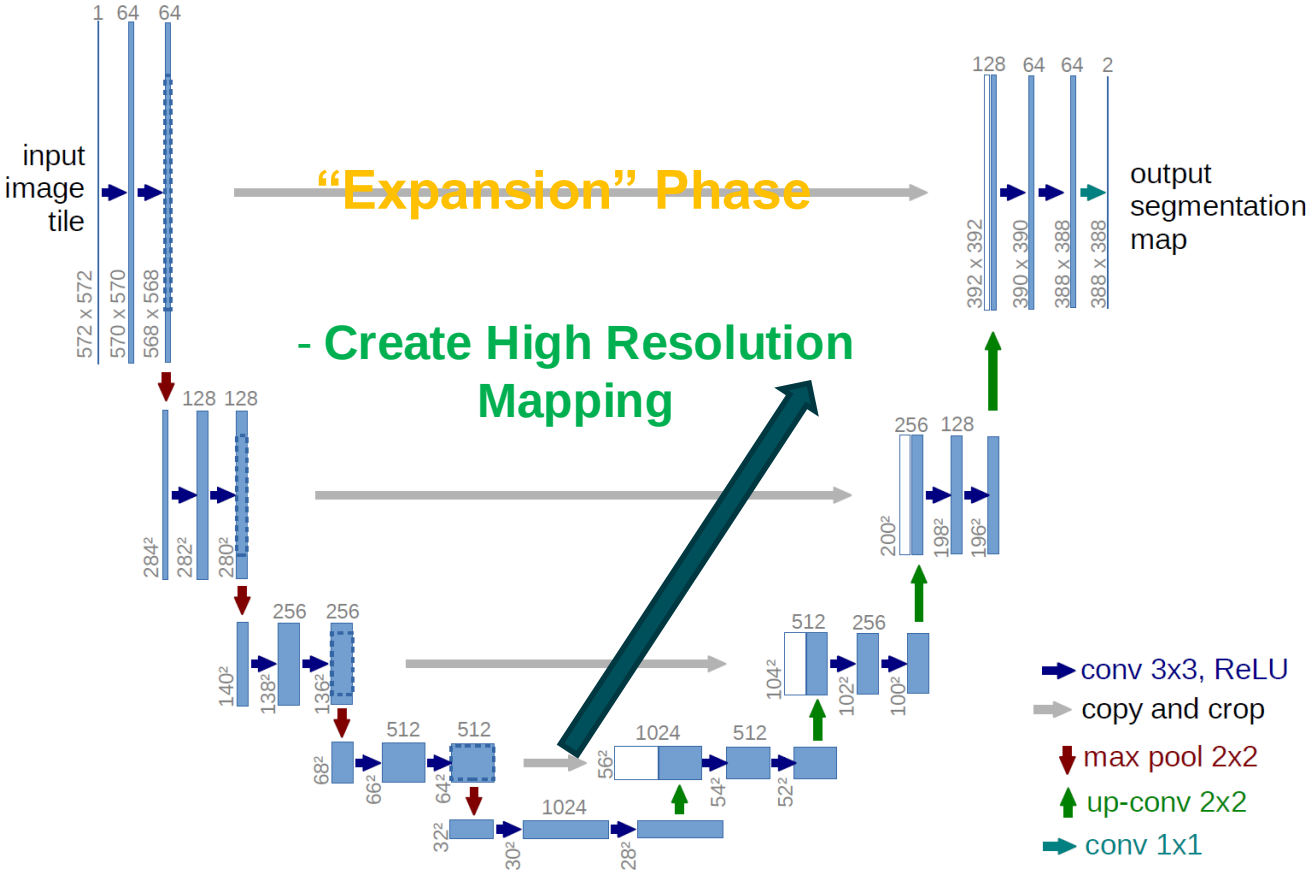
U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

Carnegie Mellon University

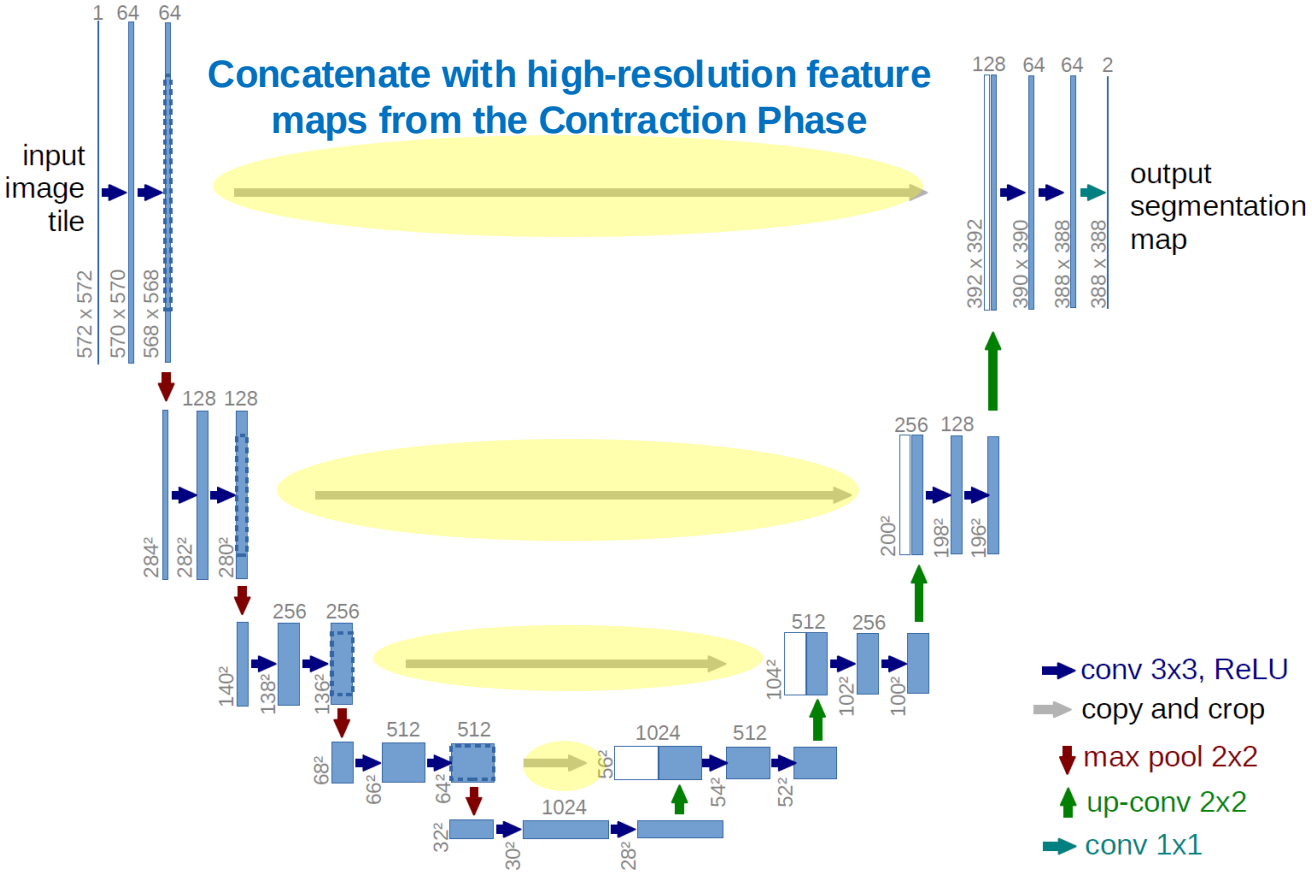
U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

Carnegie Mellon University

U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

Carnegie Mellon University

U-Net Summary

Contraction Phase

- Reduce spatial dimension, but increases the “what.”

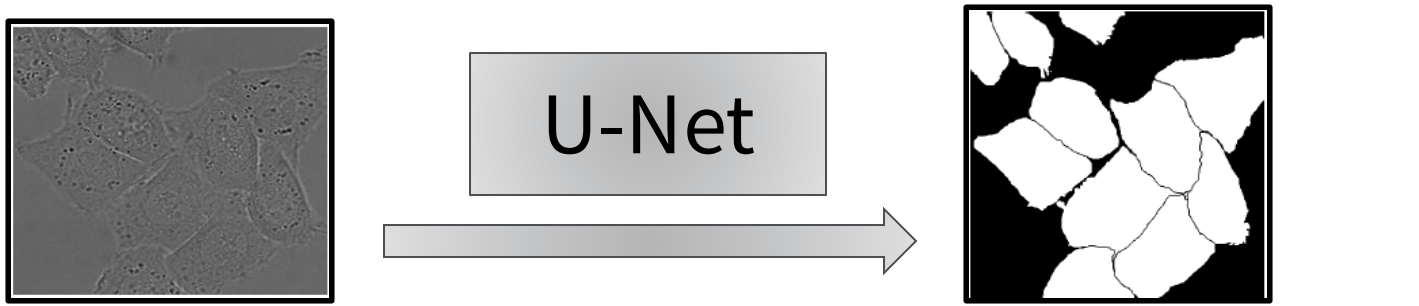
Expansion Phase

- Recovers object details and the dimensions, which is the “where.”

Concatenating feature maps from the Contraction phase helps the Expansion phase with recovering the “where” information.

Main Motivation

Biomedical Image Segmentation with U-net



- U-net learns segmentation in and end-to-end setting
- Touching objects of the same class
- Vary few annotated images (approx. 30 per application)

IEEE International Symposium on Biomedical Imaging (ISBI 2015)

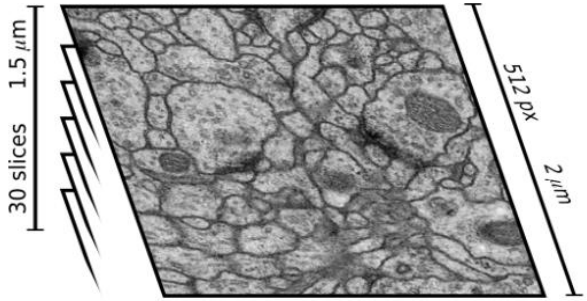
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Carnegie Mellon University

First Task

Predict the class label of each pixel

- Stacks of Electron microscopy (EM) images
- EM segmentation challenge at ISBI 2012
- 30 training images



Training stack

Black - neuron membranes
White - cells

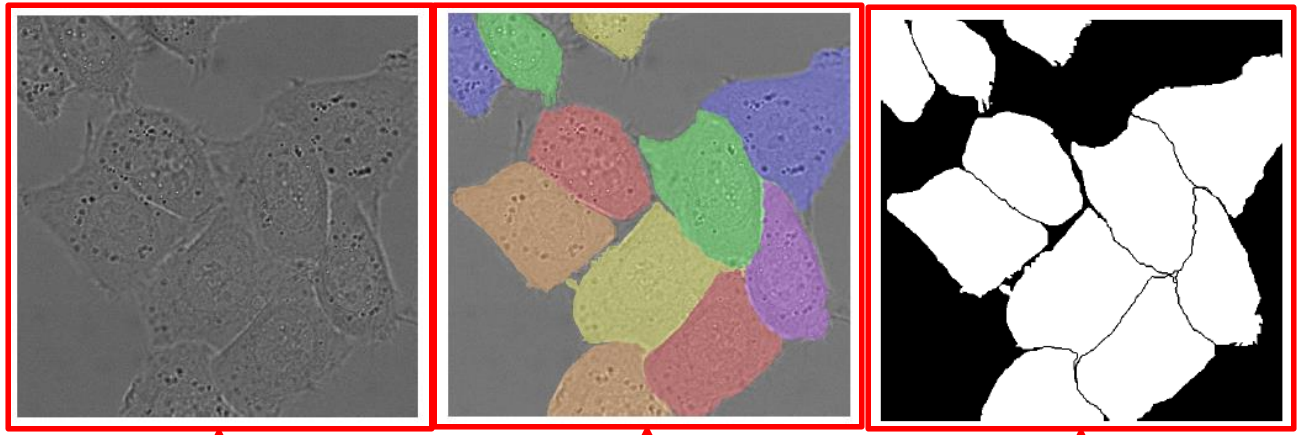
Ground truth

Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012)

Carnegie Mellon University

Second Task

ISBI 2015- separation of touching objects of the same class



Raw image (HeLa cells)

Ground truth segmentation

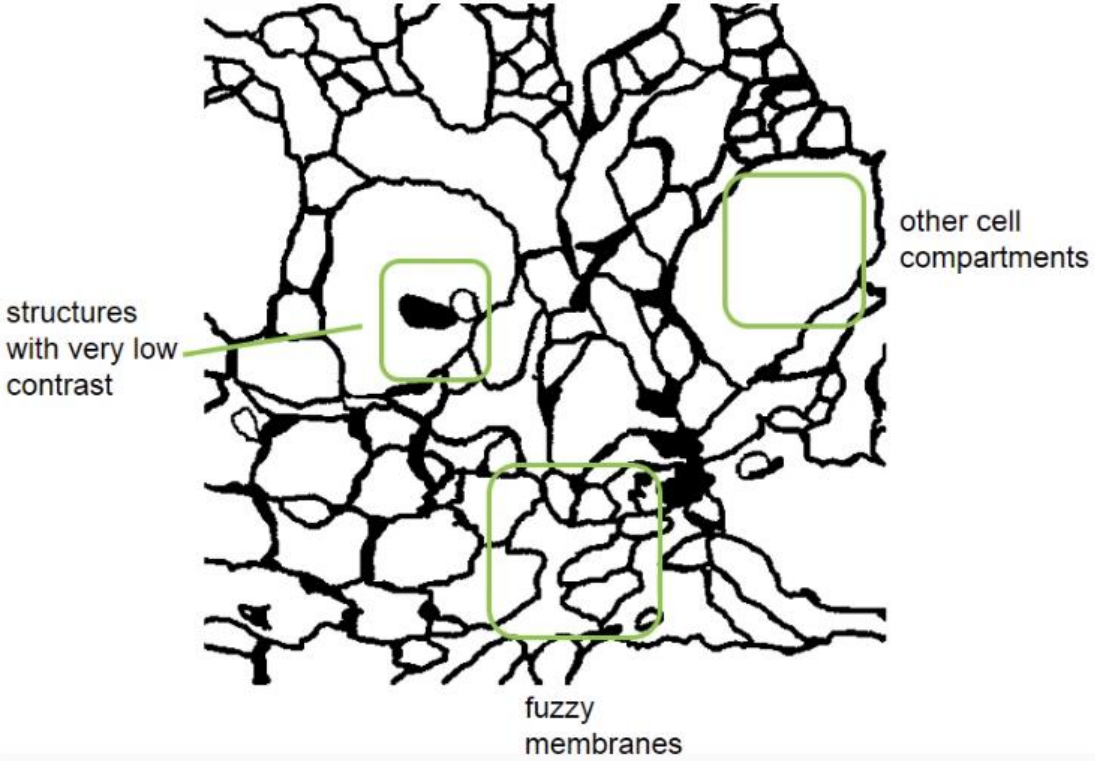
Generated segmentation mask (white: foreground, black: background)

- Light microscopic images (recorded by phase contrast microscopy). Part of the ISBI cell tracking challenge 2014 and 2015

Carnegie Mellon University

Challenges

Segmentation of Neuronal Structures in EM

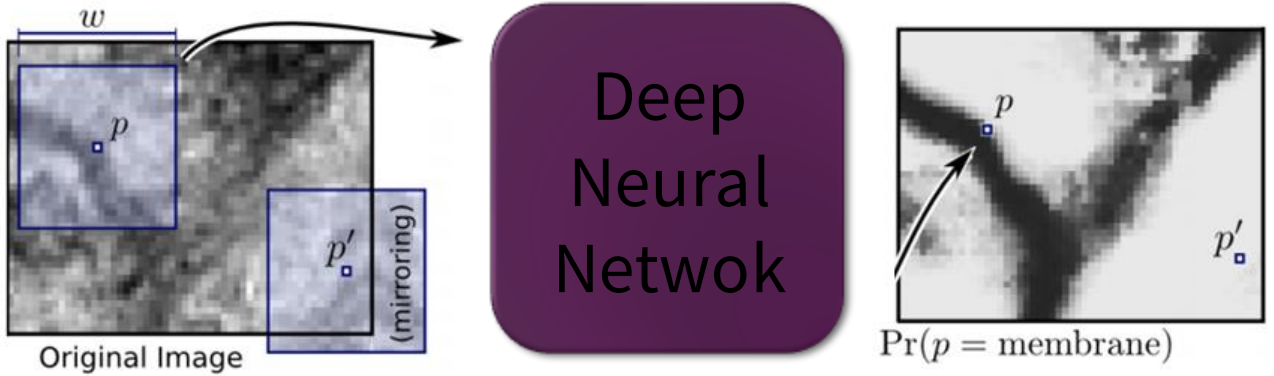


Carnegie Mellon University

Previous Work

The winner (ISBI 2012) (Ciresan et al.)

- Trained a network in a sliding-window (local region (patch) around that pixel)
  - ✓ **This network can localize**
  - ✓ **The training data in terms of patches is much larger than the number of training images**
  - x **Slow because the network must be run separately for each patch**
  - x **There is a lot of redundancy**



Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images.

Carnegie Mellon University

Previous Work

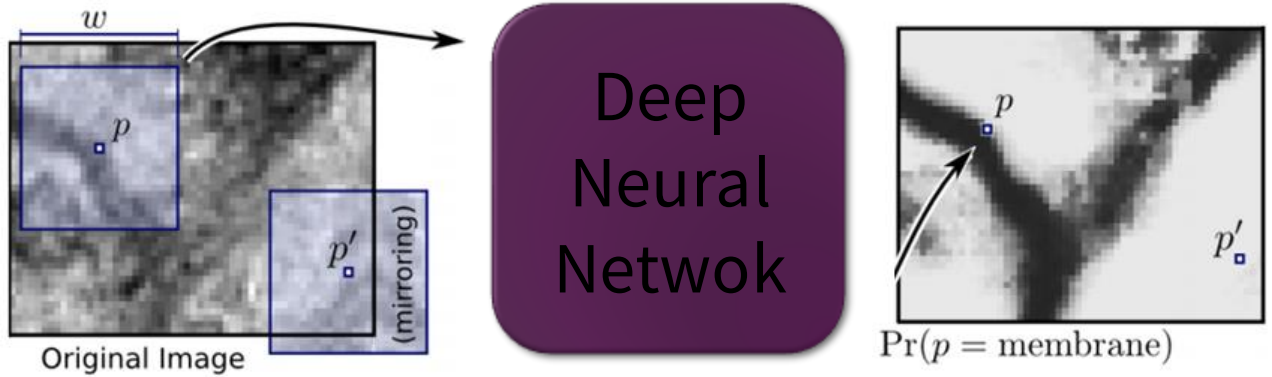
The winner (ISBI 2012)

- Trade-off between localization accuracy and the use of context.

Larger patches: Require more max-pooling layers → reduce the localization accuracy

Small patches: Allow the network to see only little context

We want a good localization and the use of context at the same time



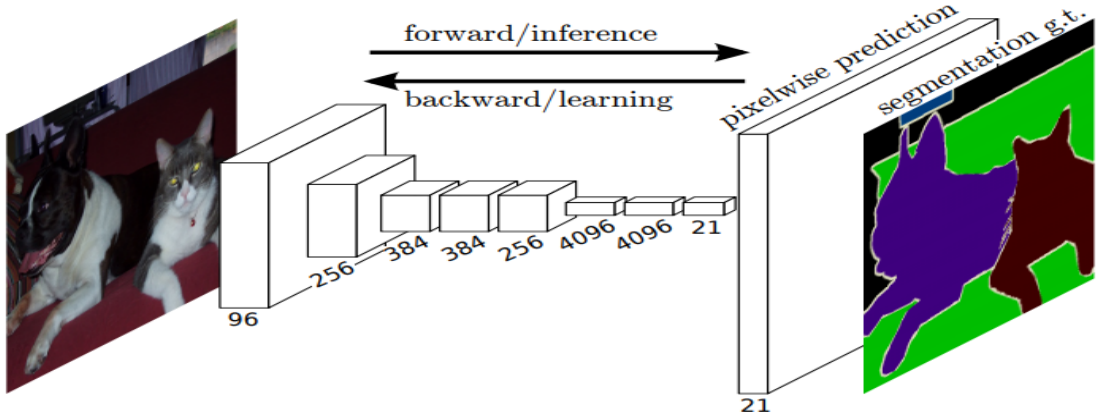
Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012)

Carnegie Mellon University

Previous Work (Inspiration)

Fully convolutional neural network (FCN) architecture for semantic segmentation

- **Input image with any size**
- **Added Simple Decoder (Upsampling + Conv)**
- **Removed Dense Layers**

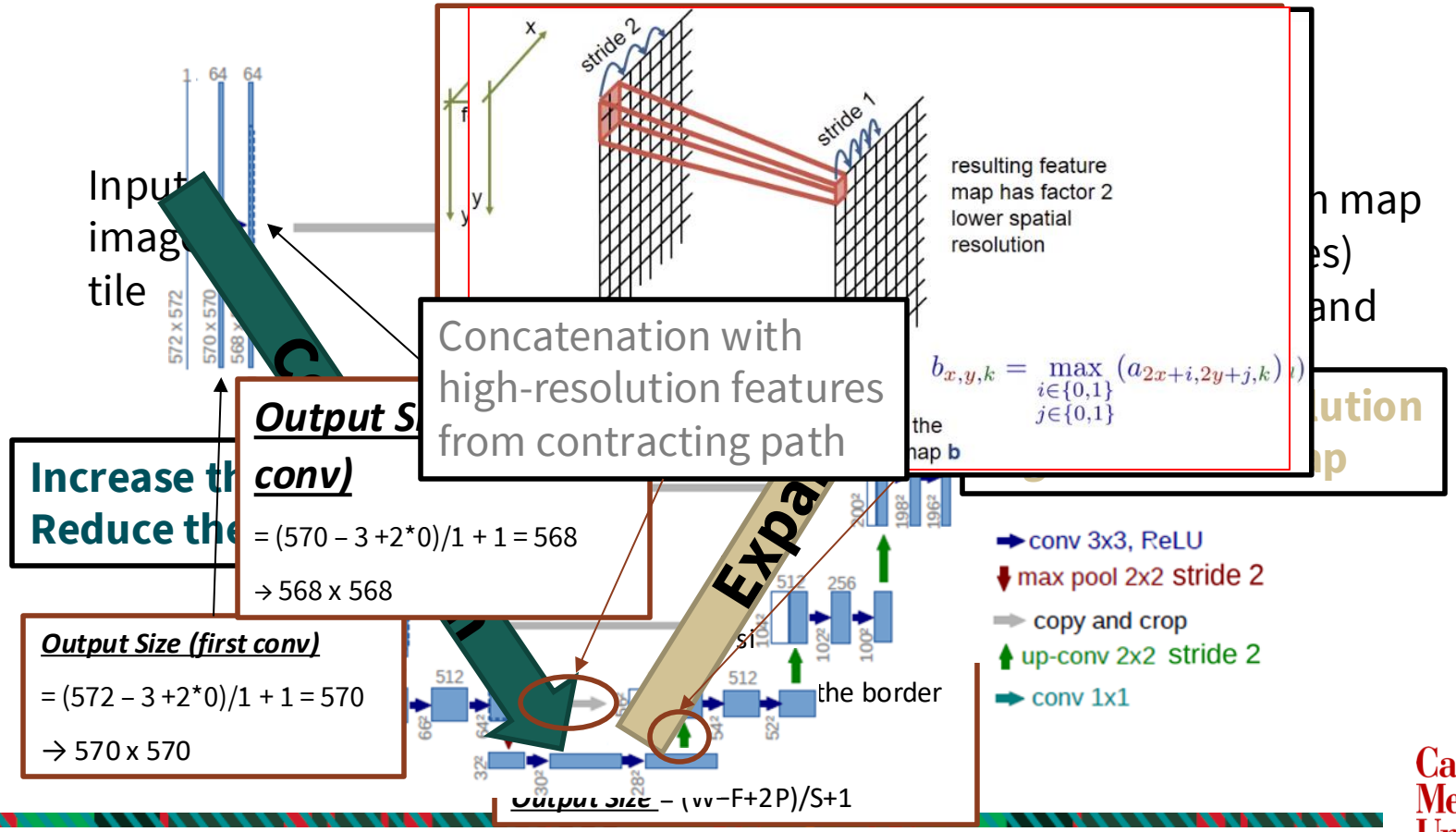


Localization and the use of context at the same time

https://www.azavea.com/blog/2017/05/30/deep-learning-on-aerial-imagery/

Carnegie Mellon University

U-NET Architecture

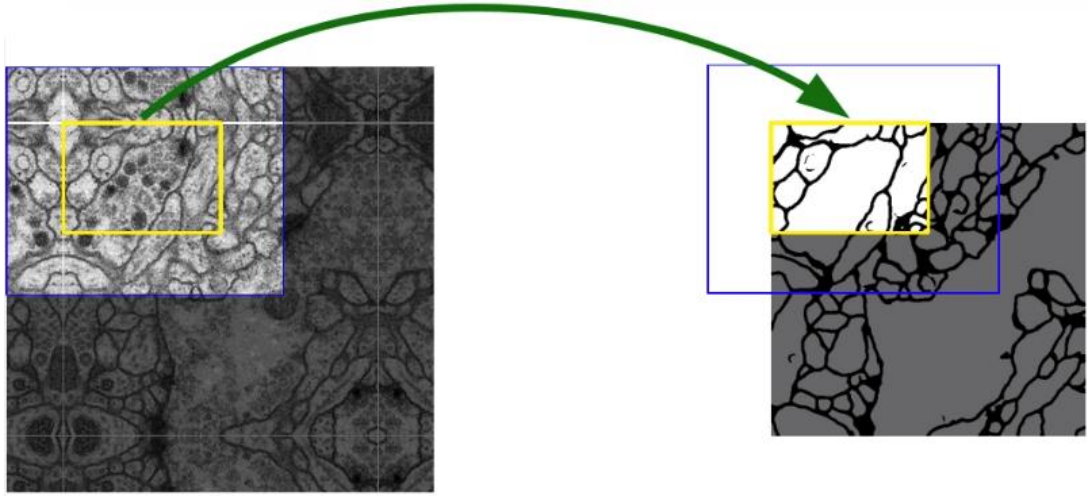


http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Carnegie Mellon University

U-NET Strategy

Over-tile strategy for arbitrary large images



- Segmentation of the yellow area uses input data of the blue area
- Raw data extrapolation by mirroring

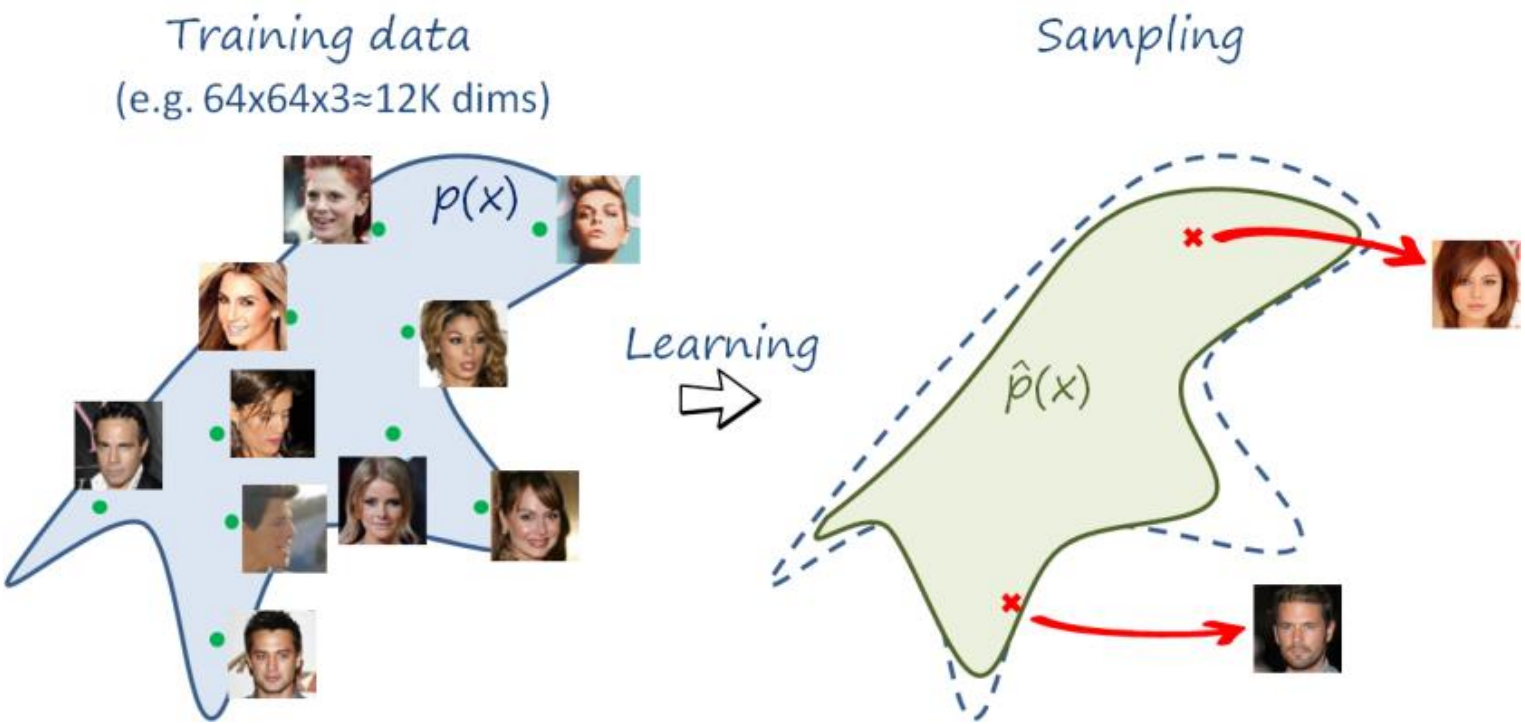
Carnegie Mellon University

Generative Adversarial Networks

Overview

- GANs Quick Recap
- GAN Training
- Issues with GAN Training
- Remedies for GAN Training Issues
- GAN Architectures & Recent Progress

Density Estimation P(x): on high dimensions



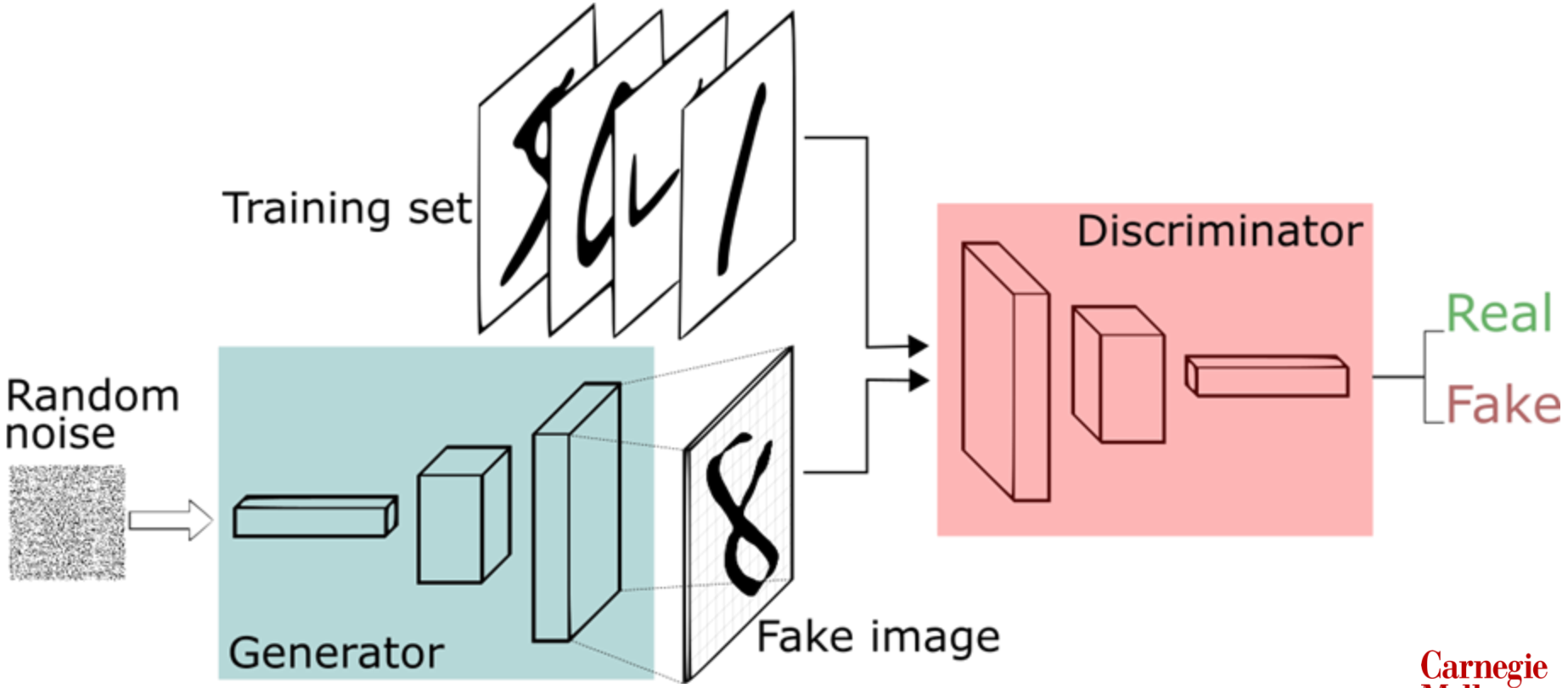
Getting Better And Better

Ian Goodfellow

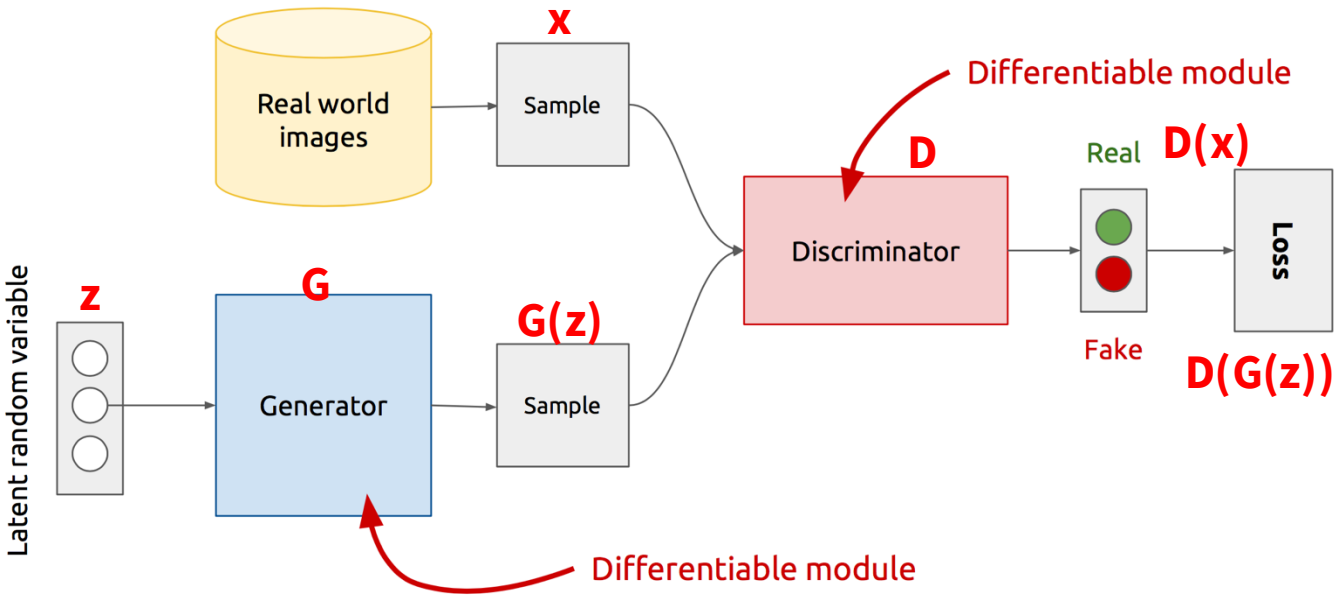
- Wrote our textbook (*Deep Learning*) with his PhD advisor, Yoshua Bengio
- Created GANs while at Google Brain
- Part of the founding team at OpenAI

Carnegie Mellon University

Generator and Discriminator



GAN's Architecture

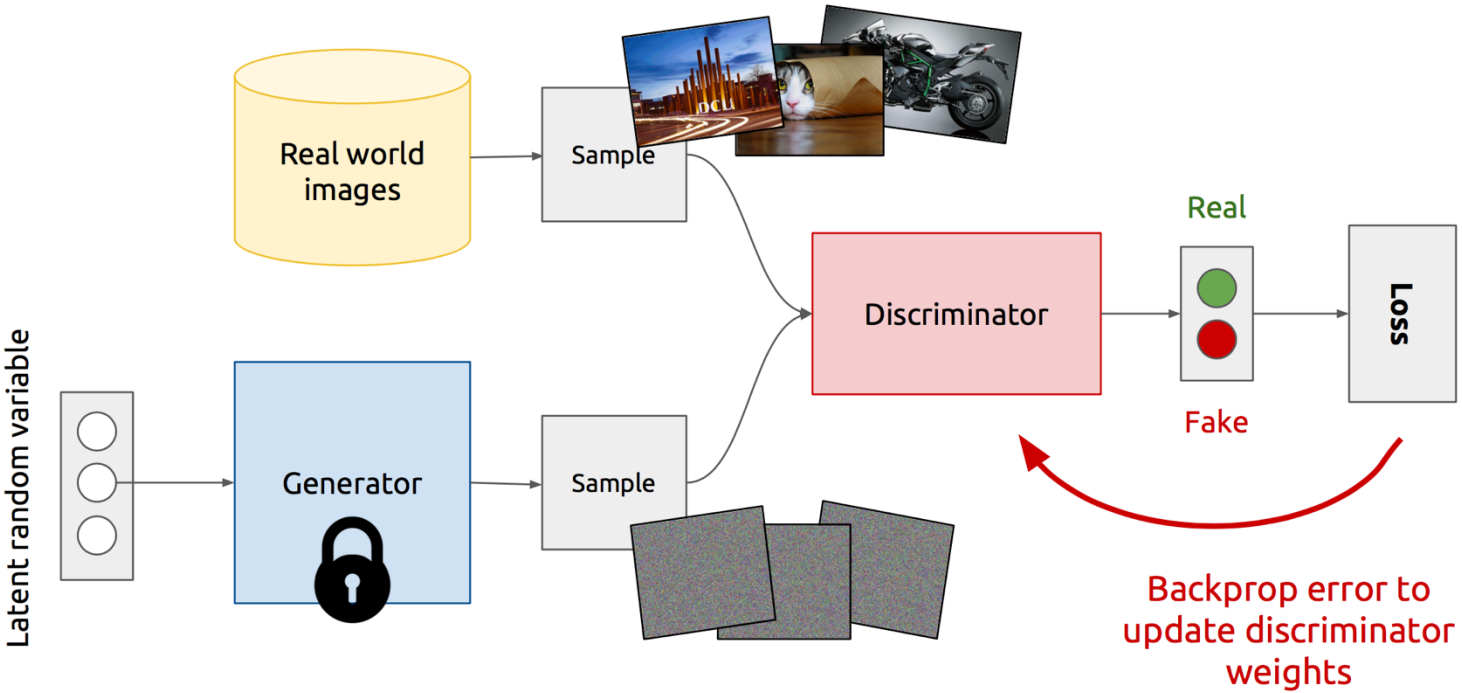


- **Z** is some random noise (Gaussian/Uniform).
- **Z** can be thought as the latent representation of the image.

https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Carnegie Mellon University

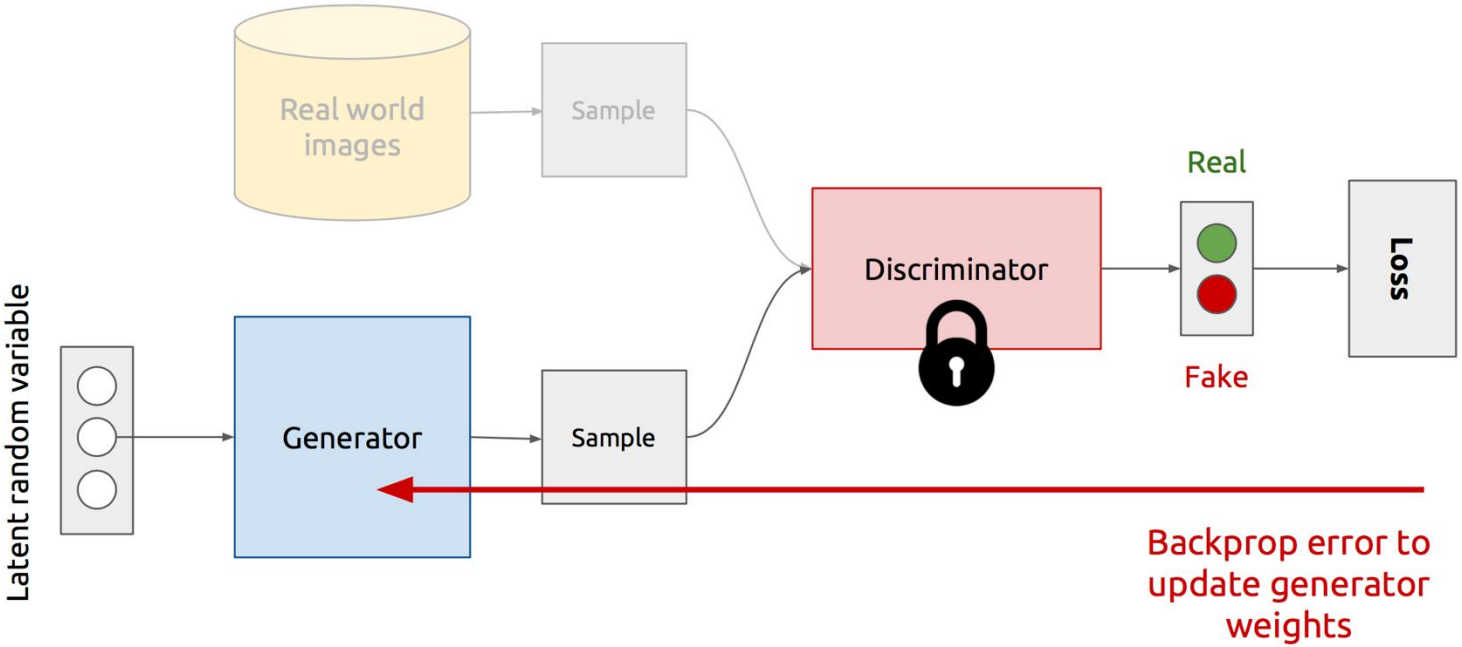
Training Discriminator



https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Carnegie Mellon University

Training Generator



https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Carnegie Mellon University

The Dual Loss Functions (attempt 1)

$$J^{(D)} = \underbrace{-\frac{1}{m_{real}}\sum_{i=1}^{m_{real}} y_{real}^{(i)}.\log(D(x^{(i)}))}_{\text{cross-entropy 1: "D should correctly label real data as 1"}} \underbrace{-\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}(1-y_{gen}^{(i)}).\log(1-D(G(z^{(i)})))}_{\text{cross-entropy 2: "D should correctly label generated data as 0"}}$$

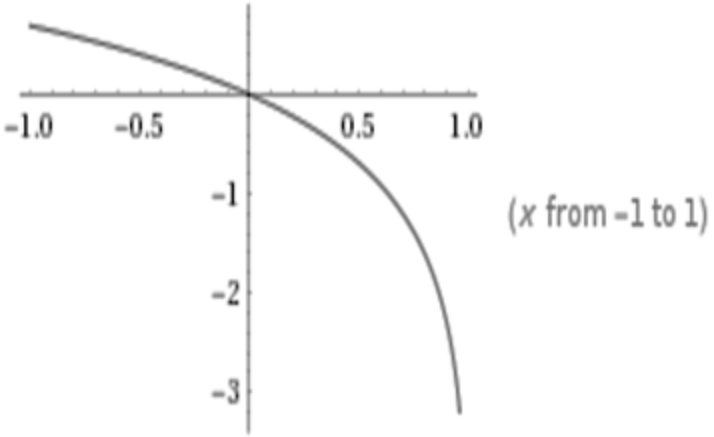
$$J^{(G)} = -J^{(D)} = \frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(1-D(G(z^{(i)}))) \quad \text{"G should try to fool D: by minimizing the opposite of what D is trying to minimize"}$$

This Is A Bad Choice Of Loss For G. Why?

$$J^{(G)} = -J^{(D)} = \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} \log(1 - D(G(z^{(i)})))$$

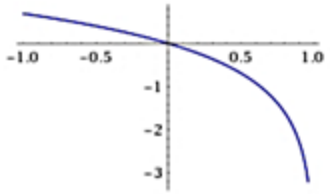
"G should try to fool D: by minimizing the opposite of what D is trying to minimize"

Why The Naïve Generator Loss Has A Saturation Issue



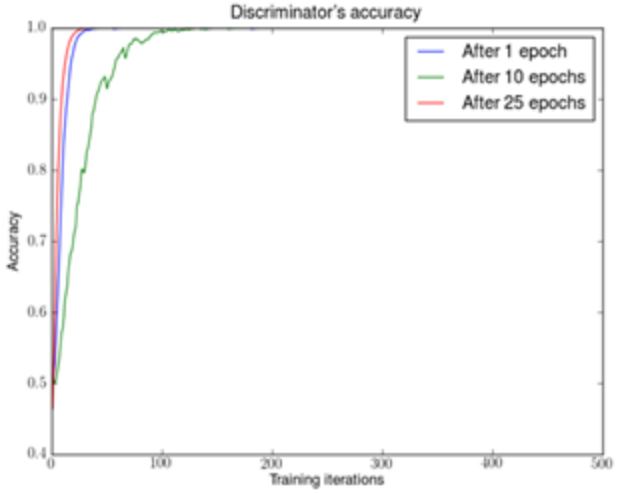
y=log (1-x)

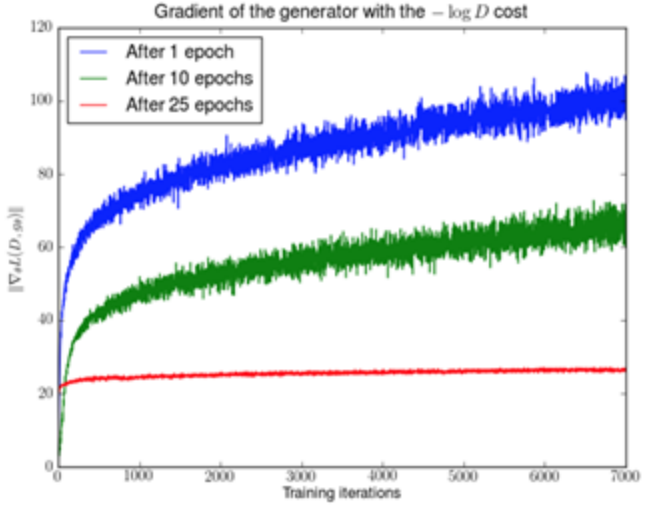
Why The Naïve Generator Loss Has A Saturation Issue



(x from -1 to 1)

y=log (1-x)





Carnegie Mellon University

The Dual Loss Functions (attempt 2)

$$J^{(D)} = \underbrace{-\frac{1}{m_{real}}\sum_{i=1}^{m_{real}} y_{real}^{(i)}.\log(D(x^{(i)}))}_{\text{cross-entropy 1: "D should correctly label real data as 1"}} \underbrace{-\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}(1-y_{gen}^{(i)}).\log(1-D(G(z^{(i)})))}_{\text{cross-entropy 2: "D should correctly label generated data as 0"}}$$

$$J^{(G)} = -\frac{1}{m_{gen}}\sum_{i=1}^{m_{gen}}\log(D(G(z^{(i)})))$$

Carnegie Mellon University

GANs In PyTorch: Architecture

```python
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()

        def block(in_feat, out_feat, normalize=True):
            layers = [nn.Linear(in_feat, out_feat)]
            if normalize:
                layers.append(nn.BatchNorm1d(out_feat, 0.8))
            layers.append(nn.LeakyReLU(0.2, inplace=True))
            return layers

        self.model = nn.Sequential(
            *block(opt.latent_dim, 128, normalize=False),
            *block(128, 256),
            *block(256, 512),
            *block(512, 1024),
            nn.Linear(1024, int(np.prod(img_shape))),
            nn.Tanh()
        )

    def forward(self, z):
        img = self.model(z)
        img = img.view(img.size(0), *img_shape)
        return img
```

```python
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()

        self.model = nn.Sequential(
            nn.Linear(int(np.prod(img_shape)), 512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(512, 256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(256, 1),
            nn.Sigmoid()
        )

    def forward(self, img):
        img_flat = img.view(img.size(0), -1)
        validity = self.model(img_flat)

        return validity
```

Carnegie Mellon University

GANs In PyTorch: Loss

```
# Generate a batch of images
gen_imgs = generator(z)

# Loss measures generator's ability to fool the discriminator
g_loss = adversarial_loss(discriminator(gen_imgs), valid)

g_loss.backward()
optimizer_G.step()

# Measure discriminator's ability to classify real from generated samples
real_loss = adversarial_loss(discriminator(real_imgs), valid)
fake_loss = adversarial_loss(discriminator(gen_imgs.detach()), fake)
d_loss = (real_loss + fake_loss) / 2

d_loss.backward()
optimizer_D.step()
```

Carnegie Mellon University

GANs Training Issues

1. Mode Collapse
2. Vanishing Gradients
3. Convergence and Oscillation

GANs Can Easily Fail To Train: Mode Collapse

NVIDIA

Mode collapse

$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

- D in inner loop: convergence to correct distribution
- G in inner loop: place all mass on most likely point

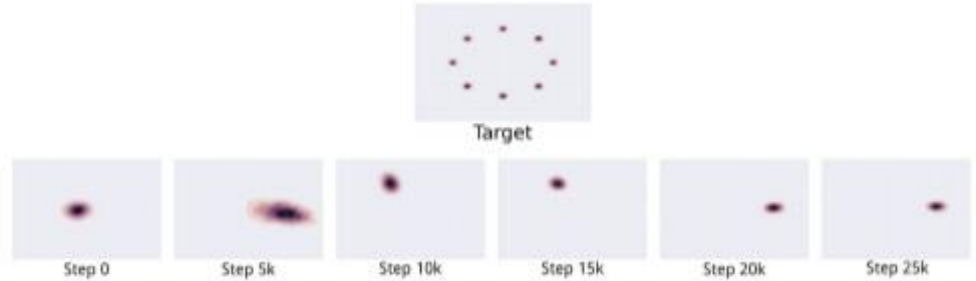


Figure credit, Metz et al, 2016
Slide credit Goodfellow 2016

Carnegie Mellon University

Problems With Counting

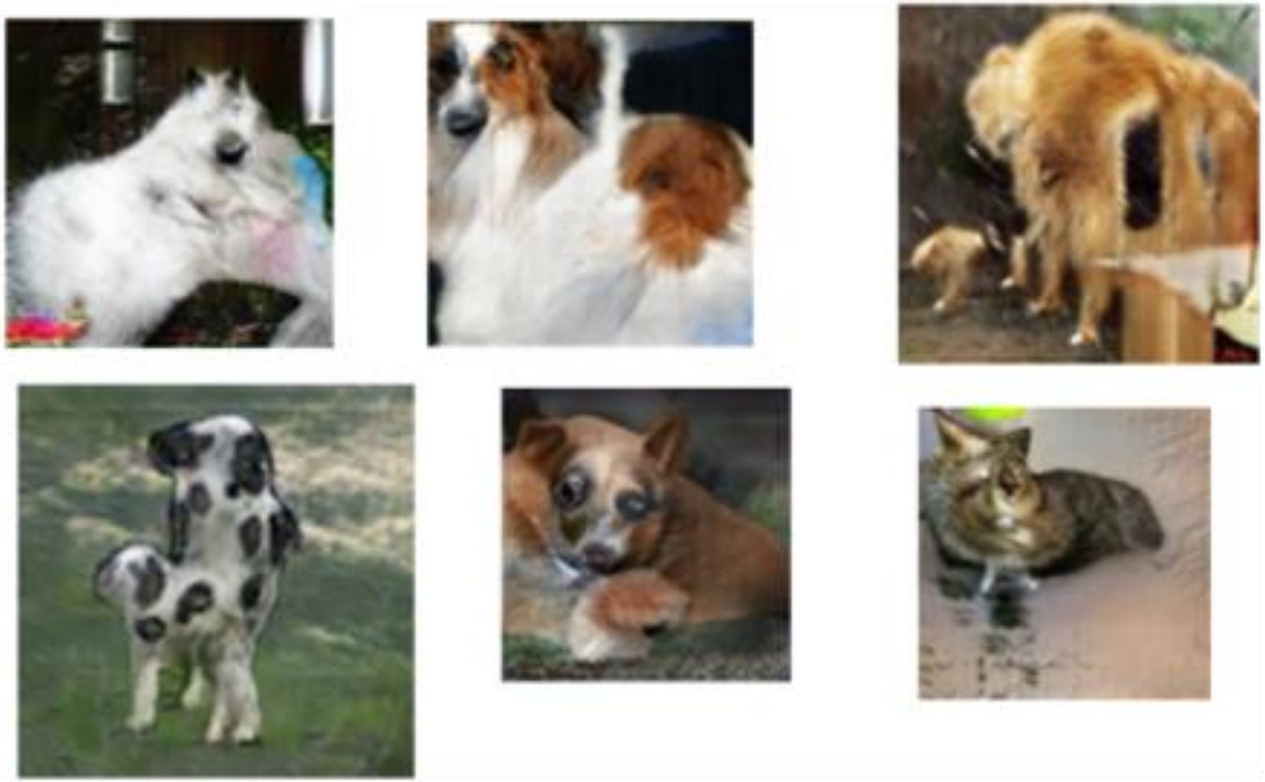





(Goodfellow 201

Carnegie Mellon University

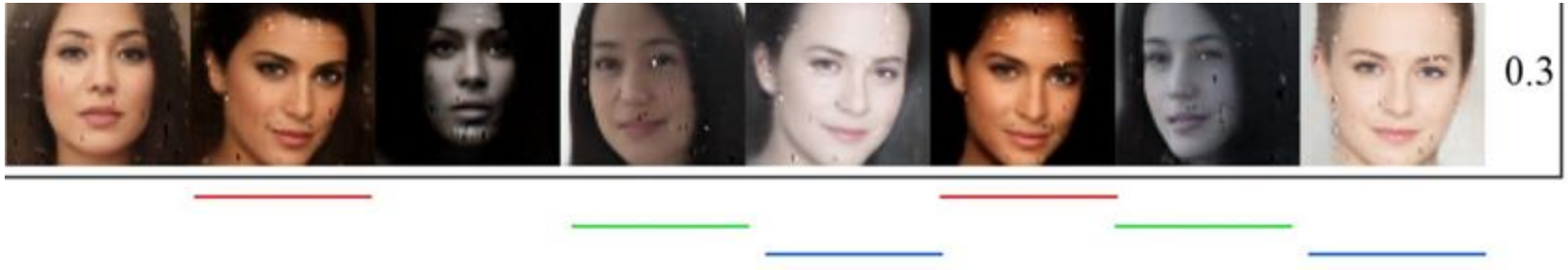
Fallout Cow



Carnegie Mellon University

GANs Training Issues – Mode Collapse

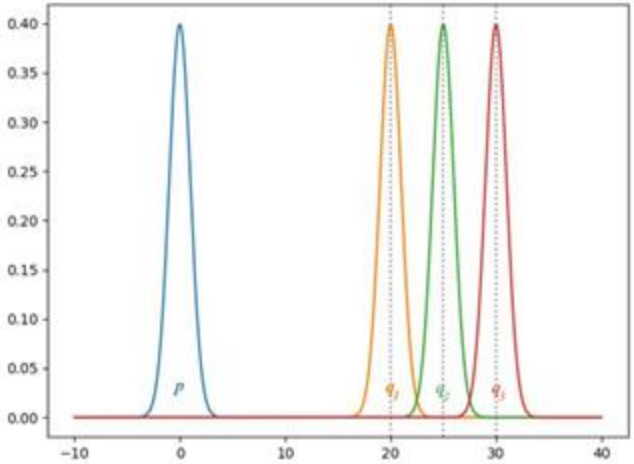
- If a generator produces an especially plausible output, the generator may learn to produce only that output
- The **Generator gets stuck** at a point where it only produces a limited variety of samples or one sample repeatedly during or after training
- Each iteration of Generator over-optimizes for a particular Discriminator, and the Discriminator never manages to learn its way out of the trap

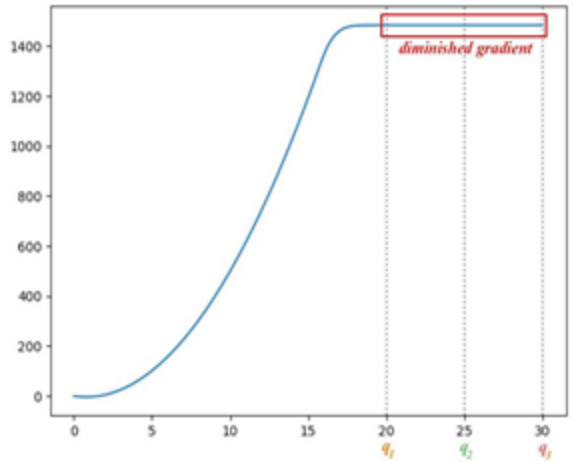


https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b

GANs Training Issues – Vanishing Gradients

- If the Discriminator is too good, then the Generator training can fail due to vanishing gradients. An optimal Discriminator doesn't provide enough information for the Generator to make progress
- Below image shows how gradients will vanish if the distribution of generated images (*p*) is too different than the distribution of real images (*q1, q2, q3*)





https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b

Carnegie Mellon University

GANs Training Issues – Convergence and Oscillation

- GAN training is based on a zero-sum, non-cooperative, minmax game. In short, if one wins the other loses
- In game theory, the GAN model converges when the discriminator and the generator reach a Nash equilibrium. This is the optimal point for the GAN objective
- Simplified example: Consider two player A and B which control the value of *x* and *y* respectively. Player A wants to maximize the value *xy* while B wants to minimize it

GANs Training Issues – Convergence and Oscillation

We update the parameter *x* and *y* based on the gradient of the functions:

$f(x) = xy$, $f(y) = -xy$

$\partial f/\partial x = y$ and $\partial f/\partial y = -x$

$x \rightarrow x - \alpha \cdot y$ and $y \rightarrow y + \alpha \cdot x$ (α is learning rate)

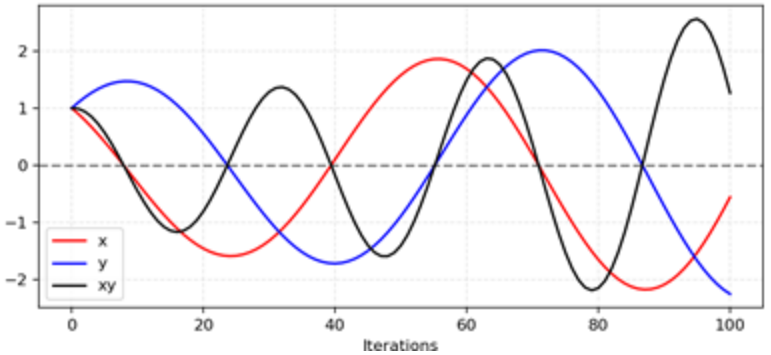
GANs Training Issues – Convergence and Oscillation

- We update the parameter *x* and *y* based on the gradient of the functions

  f(x) = xy, f(y) = -xy

$\partial f/\partial x = y$ and $\partial f/\partial y = -x$

$x \rightarrow x - \alpha \cdot y$ and $y \rightarrow y + \alpha \cdot x$ (α is learning rate)



- The Nash equilibrium is **x = y = 0**. This is the only state where the action of your opponent does not matter. It is the only state that any opponents' actions will not change the game outcome

https://medium.com/deep-math-machine-learning-ai/ch-14-general-adversarial-networks-gans-with-math-1318faf46b43

Carnegie Mellon University

Feature Matching

Statistics of generated images should match statistics of real images.

Discriminator **produces multidimensional output, a "statistic" of data**.

Generator trained to minimize L2 between real and generated data.

Discriminator trained to maximize L2 between real and generated data.

Goal: matching features in real images

$$\|E_X D(X) - E_Z D(G(Z))\|_2^2$$

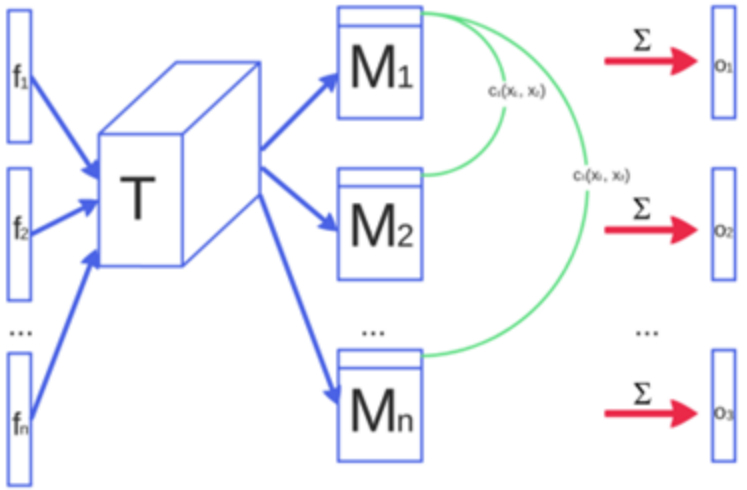
$$\|E_X f(X) - E_Z f(G(Z))\|_2^2$$

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, Improved techniques for training gans, CoRR abs/1606.03498 (2016).

Minibatch Discrimination

Discriminator can look at multiple inputs at once and decide if those inputs come from the real or generated distribution.

- GANs frequently collapse to a single point
- Discriminator needs to differentiate between two distributions
- Easier task if looking at multiple samples



Append the **similarity between the image and other images in the same batch** in **one of the dense layers in the discriminator** to classify whether this image is real or generated

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, Improved techniques for training gans, CoRR abs/1606.03498 (2016).

Historical Averaging

Dampen oscillations by encouraging updates to converge to a mean.

- GANs frequently create a cycle or experience oscillations
- **Add a term** to reduce oscillations that encourage the current parameters to be near a **moving average of the parameters**

$$\left\| \theta - \frac{1}{t} \sum_{i}^{t} \theta_i \right\|_2^2$$

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, Improved techniques for trainingganss, CoRR abs/1606.03498 (2016).

One-sided Label Smoothing

Don’t over-penalize generated images

- Label smoothing is a common and easy technique that improves performance across many domains

- Sigmoid tries hard to saturate to 0 or 1 but can never quite reach that goal

- Provide targets that are epsilon or 1- epsilon so the sigmoid doesn’t saturate and overtrain

  - Experimentally, **smooth the real targets** but **do not smooth the generated targets** when training the discriminator.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, Improved techniques for training gans, CoRR abs/1606.03498 (2016).

Virtual Batch Normalization

Use batch normalization to accelerate convergence

- Batch normalization accelerates convergence
- However, hard to apply in adversarial setting
- Collect statistics on **fixed batch** of real data and use to normalize other data.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, Improved techniques for training gans, CoRR abs/1606.03498 (2016).

GAN Architectures

There are many variations of GANs for modeling different tasks. This is not meant to be exhaustive but a sample of the possibilities.

- GAN
- Conditional GAN
- LapGAN
- Recurrent Adversarial Network
- Categorical GAN
- InfoGAN
- AAE
- BiGAN
- CycleGAN

GAN

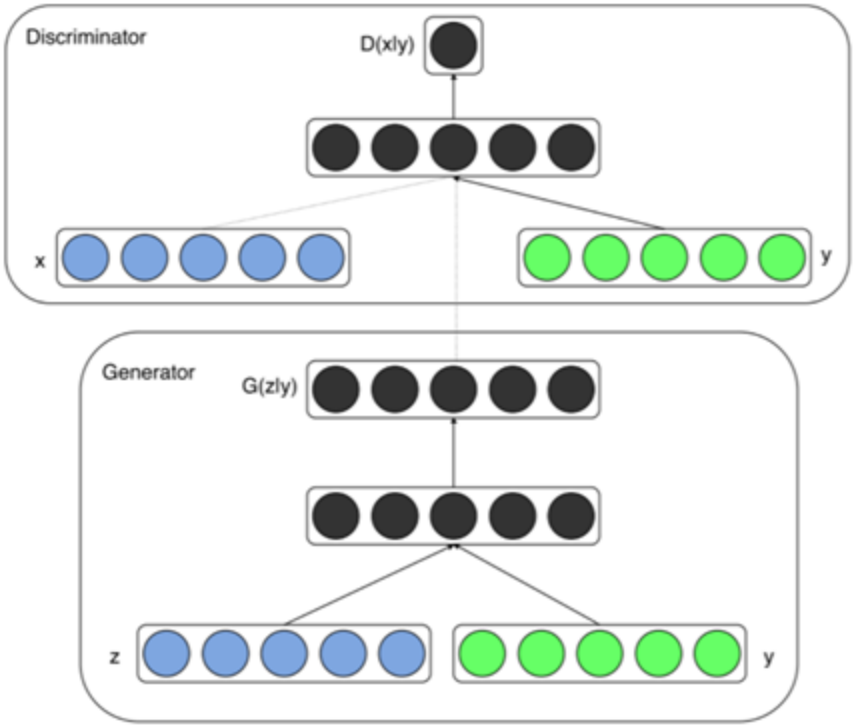
Unqualified, "GAN" typically refers to a simple model of $P(X)$ [GPM+14]. This is a vanilla GAN. Think unsupervised generation of unlabeled images, video, etc.

Conditional GANs

A conditional GAN models $P(X \mid Y)$. For example, generate samples of MNIST conditioned on the digit you are generating. [MO14]. The model is constructed by adding the labels Y as an input to both generator and discriminator.

$$\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X, Y) + \mathbb{E}_Z \log D(G(Z, Y), Y)$$

Conditional GAN Architecture



Conditional GAN Results

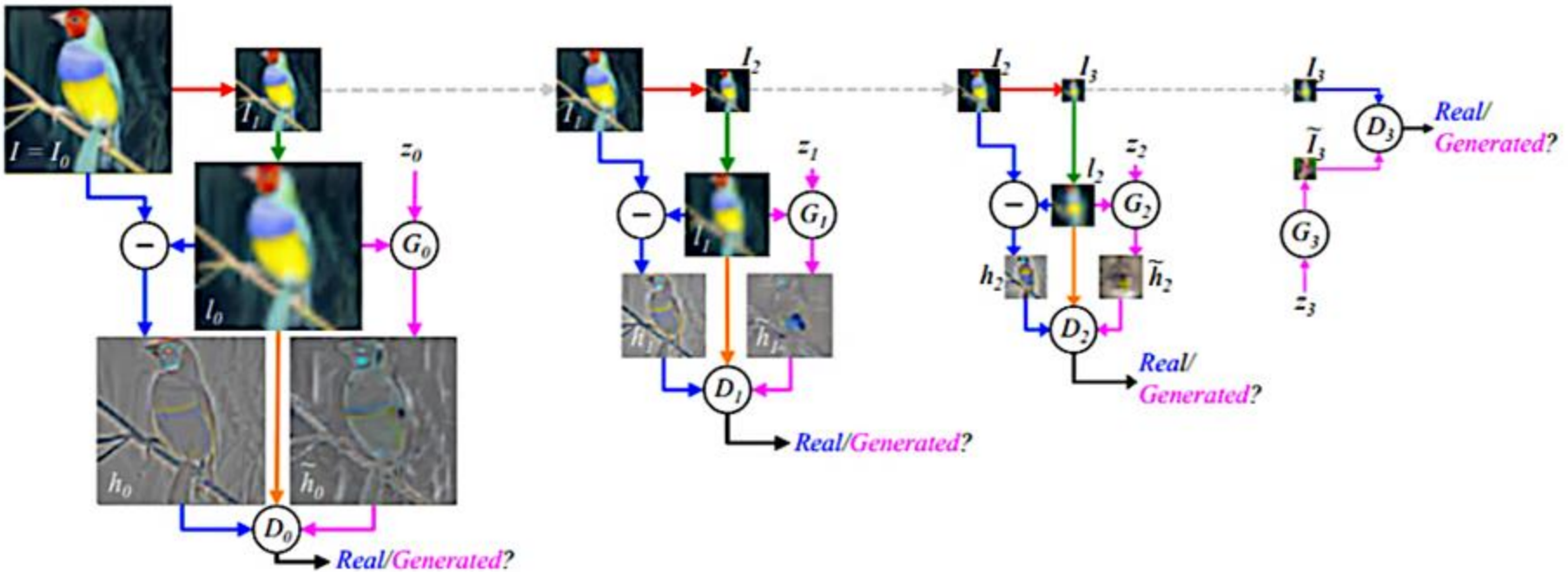
Figure 2: Generated MNIST digits, each row conditioned on one label

Carnegie Mellon University

LapGAN

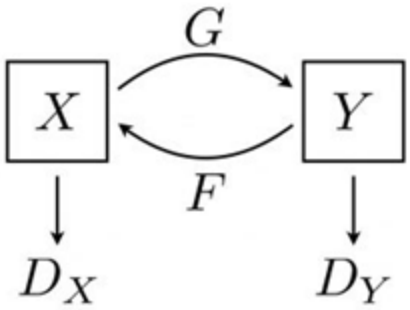
A Laplacian GAN is constructed of a chain of conditional GANs, to generate progressively larger images. A GAN generates small, blurry images. A conditional GAN generates larger images conditioned on the smaller image, repeated until you reach the desired size. [DCSF15]

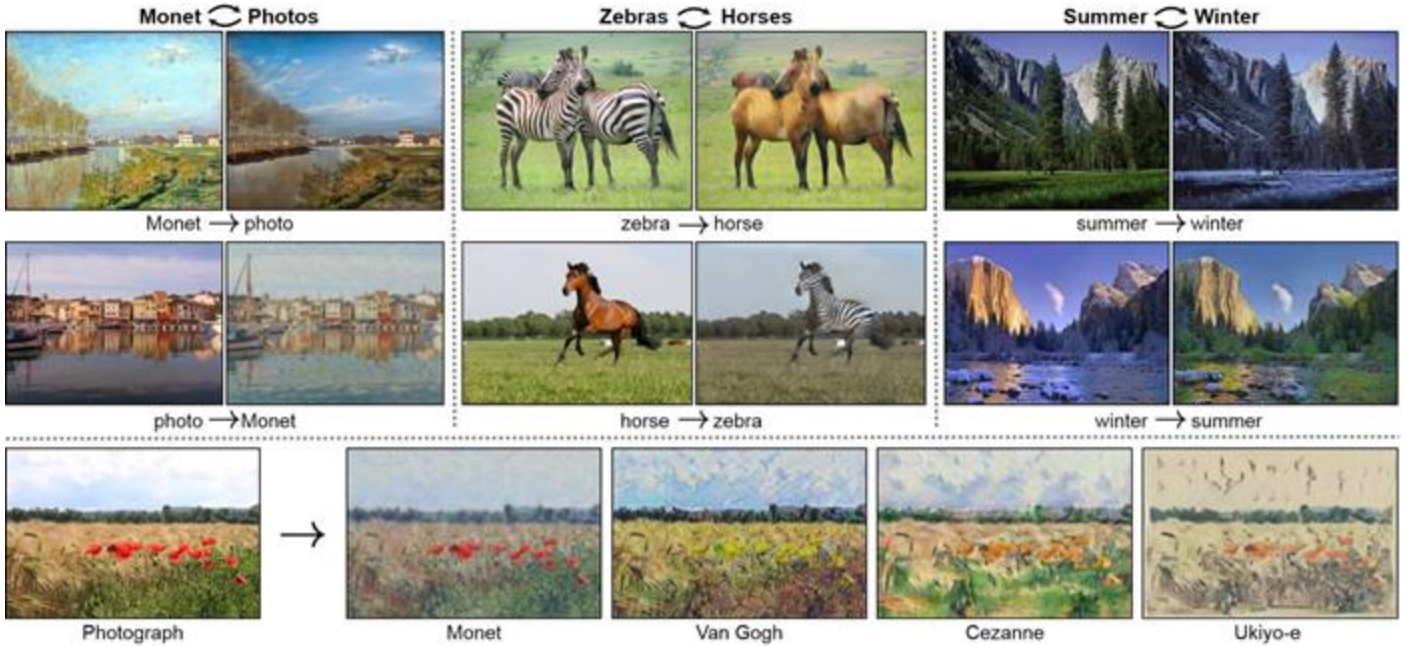
LapGAN Architecture



Carnegie Mellon University

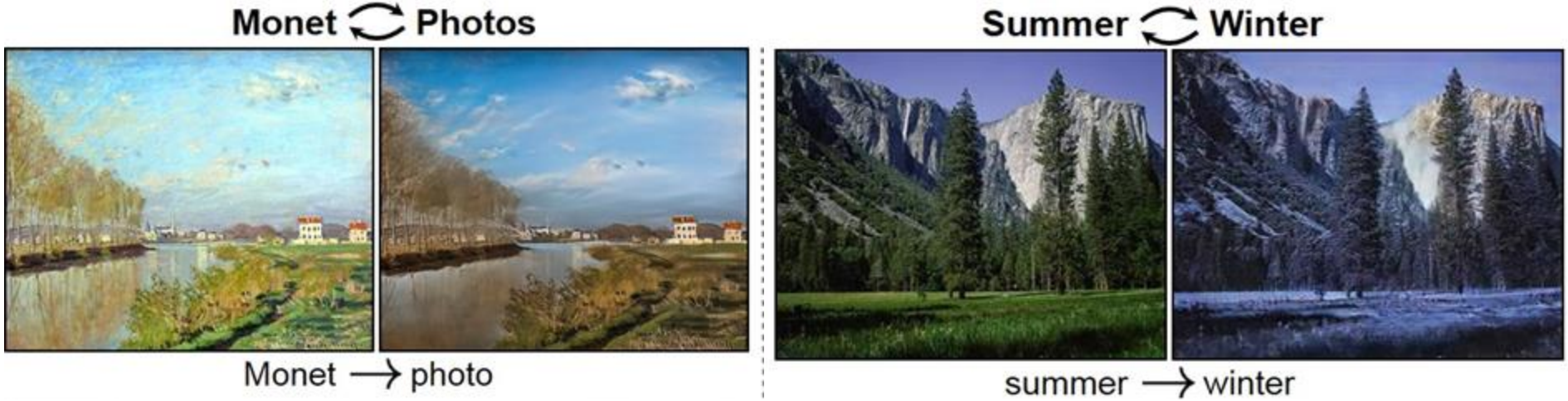
CycleGANs





Carnegie Mellon University

CycleGANs

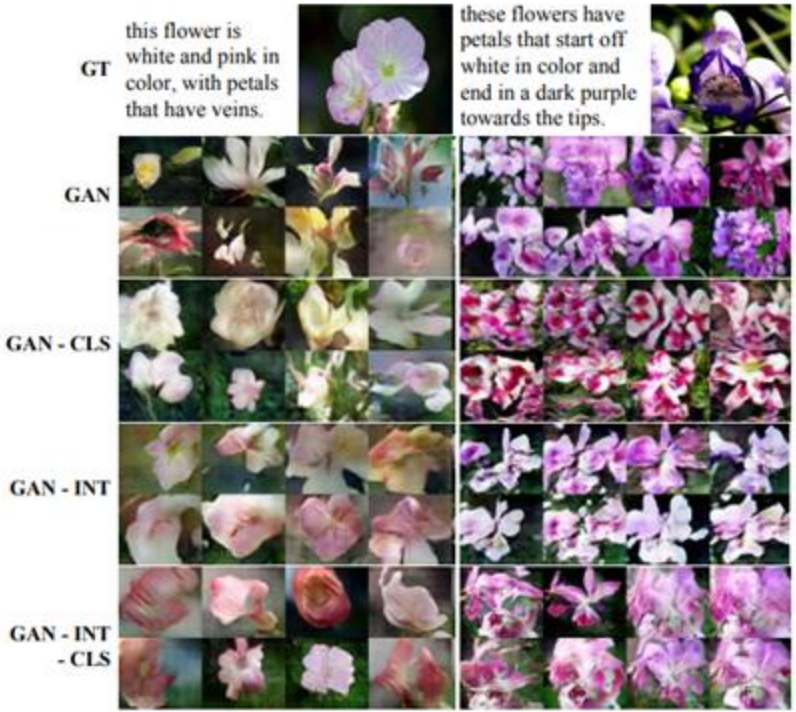


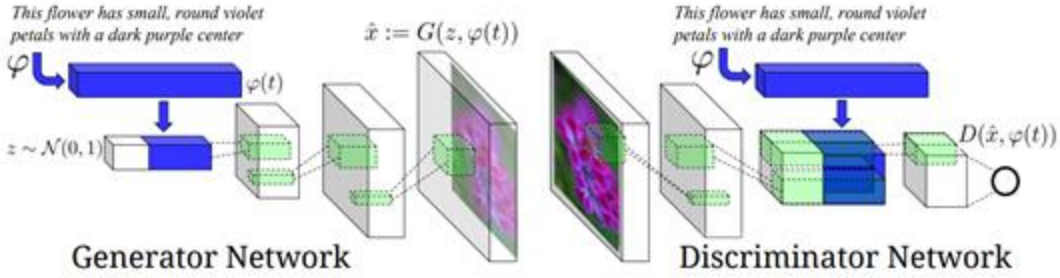
D2GAN

$$\min_{G} \max_{D_1, D_2} \mathcal{J}(G, D_1, D_2) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log D_1(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [-D_1(G(\mathbf{z}))]$$
$$+ \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [-D_2(\mathbf{x})] + \beta \times \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [\log D_2(G(\mathbf{z}))]$$

Carnegie Mellon University

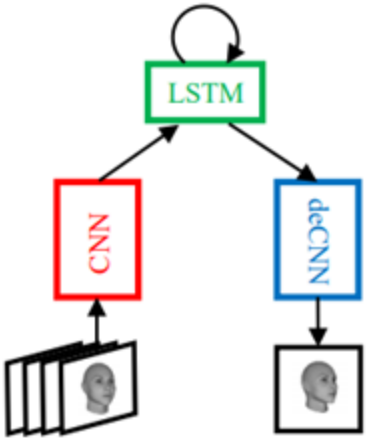
GATIS



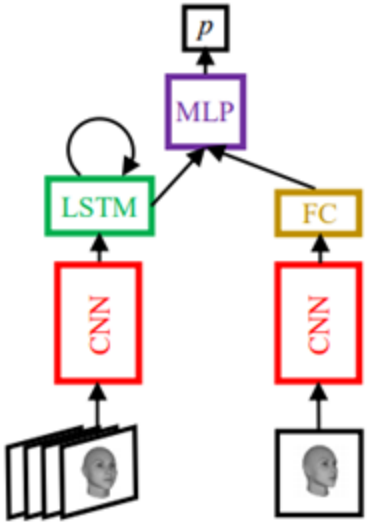


Carnegie Mellon University

Conditional Recurrent GAN



Predictive Generative Network



Adversarial Discriminator

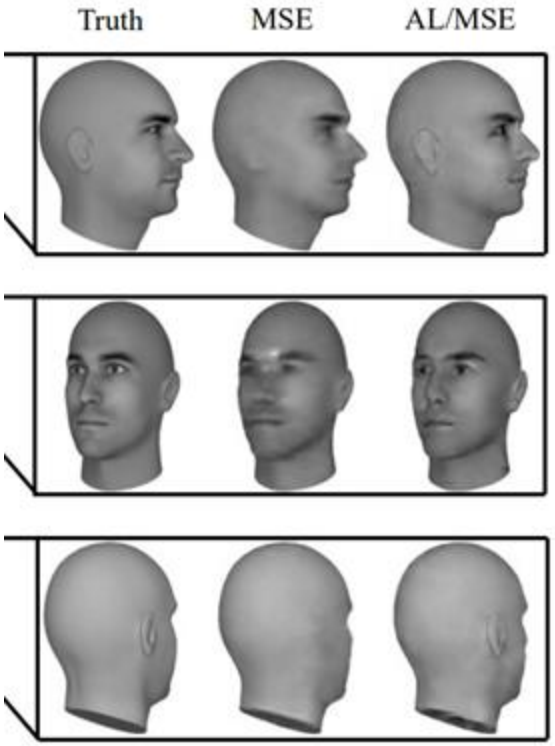


Figure 1: Predictive Generative Network (PGN)

Nvidia’s Growing GAN: Which Is Fake?

Carnegie Mellon University

Nvidia's Growing GAN: Training Methodology

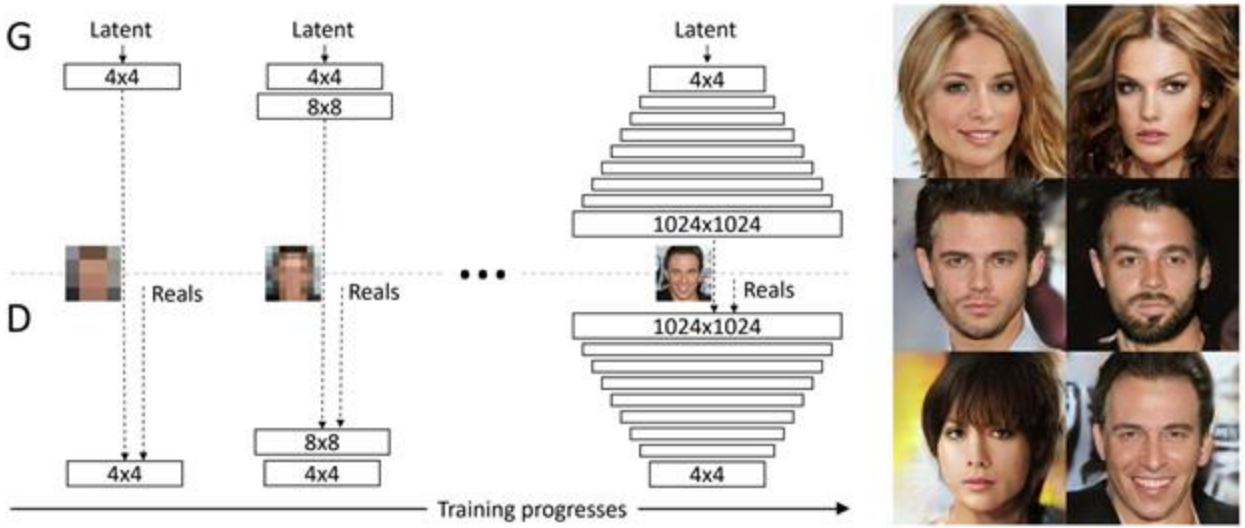


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $N \times N$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

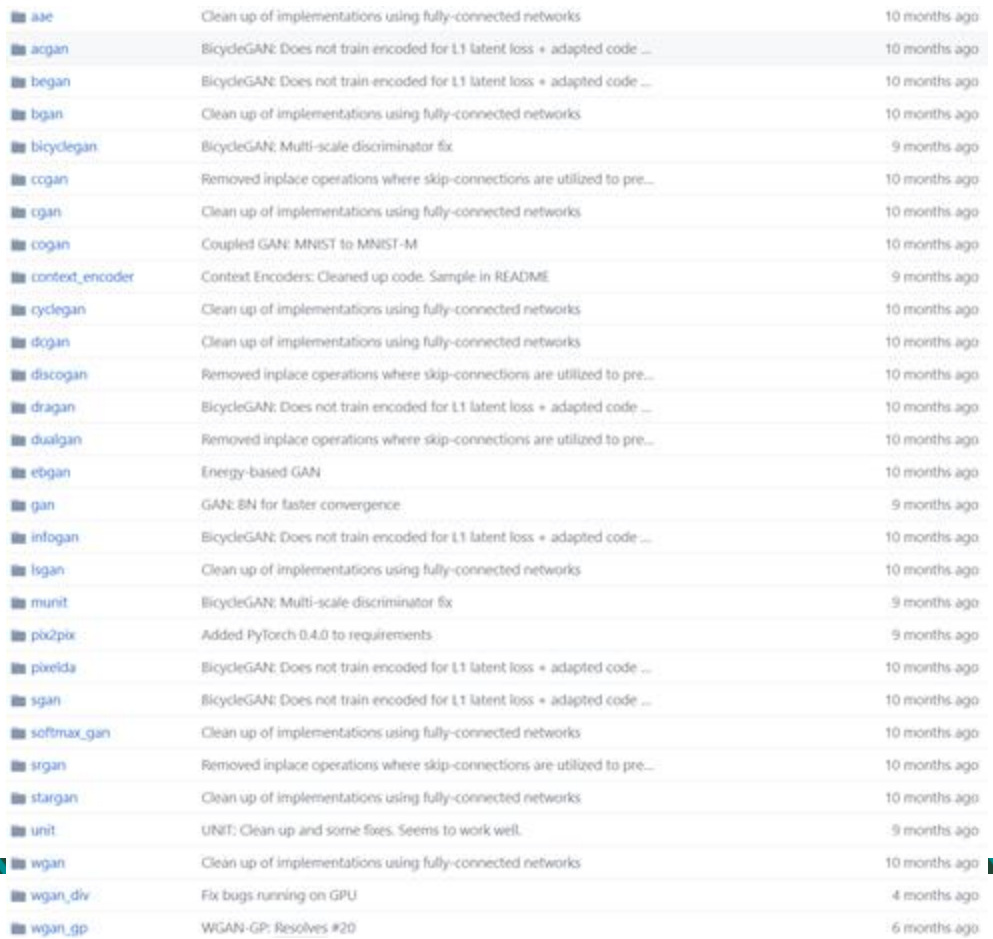
HTTPS://MEDIUM.COM/SYNCEDREVIEW/GAN-2-0-NVIDIAS-HYPERREALISTIC-FACE-GENERATOR-E3439D33EBAF

Carnegie Mellon University

There Are So, So Many Types Of GANs

GAN	Discriminator Loss	Generator Loss
MM GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{GAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
NS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{NSGAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{NSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
WGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} = -\mathbb{E}_{x\sim p_d}[D(x)] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
WGAN GP	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGANGP}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\|\nabla D(\alpha x + (1-\alpha\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGANGP}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{LSGAN}} = -\mathbb{E}_{x\sim p_d}[(D(x)-1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{LSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x}-1)^2]$
DRAGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{DRAGAN}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_d+\mathcal{N}(0,c)}[(\|\nabla D(\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{DRAGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
BEGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{BEGAN}} = \mathbb{E}_{x\sim p_d}[\|\|x-\mathrm{AE}(x)\|\|_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x}-\mathrm{AE}(\hat{x})\|\|_1]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{BEGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x}-\mathrm{AE}(\hat{x})\|\|_1]$

No, Really. There Are A Lot



aae	Clean up of implementations using fully-connected networks	10 months ago
acgan	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
began	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
bgan	Clean up of implementations using fully-connected networks	10 months ago
bicyclegan	BicycleGAN: Multi-scale discriminator fix	9 months ago
ccgan	Removed inplace operations where skip-connections are utilized to pre...	10 months ago
cgan	Clean up of implementations using fully-connected networks	10 months ago
cogan	Coupled GAN: MNIST to MNIST-M	10 months ago
context_encoder	Context Encoders: Cleaned up code. Sample in README	9 months ago
cyclegan	Clean up of implementations using fully-connected networks	10 months ago
dcgan	Clean up of implementations using fully-connected networks	10 months ago
discogan	Removed inplace operations where skip-connections are utilized to pre...	10 months ago
dragan	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
dualgan	Removed inplace operations where skip-connections are utilized to pre...	10 months ago
ebgan	Energy-based GAN	10 months ago
gan	GAN: BN for faster convergence	9 months ago
infogan	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
lsgan	Clean up of implementations using fully-connected networks	10 months ago
munit	BicycleGAN: Multi-scale discriminator fix	9 months ago
pix2pix	Added PyTorch 0.4.0 to requirements	9 months ago
pixelda	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
sgan	BicycleGAN: Does not train encoded for L1 latent loss + adapted code ...	10 months ago
softmax_gan	Clean up of implementations using fully-connected networks	10 months ago
srgan	Removed inplace operations where skip-connections are utilized to pre...	10 months ago
stargan	Clean up of implementations using fully-connected networks	10 months ago
unit	UNIT: Clean up and some fixes. Seems to work well.	9 months ago
wgan	Clean up of implementations using fully-connected networks	10 months ago
wgan_div	Fix bugs running on GPU	4 months ago
wgan_gp	WGAN-GP: Resolves #20	6 months ago

HTTPS://GITHUB.COM/ERIKLINDERNOREN/PYTORCH-GAN/TREE/MASTER/IMPLEMENTATIONS

Carnegie Mellon University

Ethics Of Being Able To Create Reality-like Fiction

“This is a big deal,” Hany Farid, computer science professor at Dartmouth College, told The Wall Street Journal. “You can literally put into a person’s mouth anything you want.”

Reddit bans 'deepfakes,' pornography using the faces of celebrities such as Taylor Swift and Gal Gadot

Carnegie Mellon University

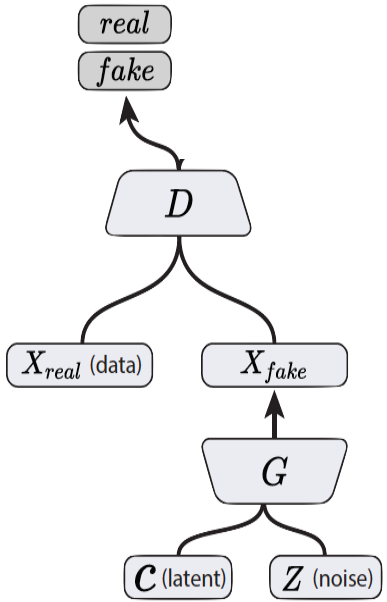
InfoGAN

We want to maximize the mutual information I between $\boldsymbol{c}$ and

$\mathbf{x} = \boldsymbol{G}(\boldsymbol{z}, \boldsymbol{c})$

Incorporate in the value function of the minimax game.

$$\min_G \max_D V_I(D,G) = V(D,G) \boxed{- \lambda\, I\big(c; G(z,c)\big)}$$



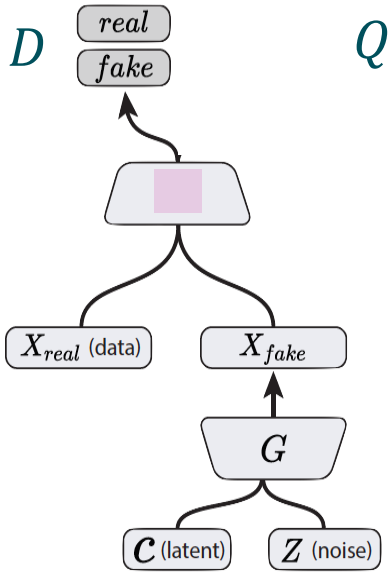
Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

InfoGAN

Mutual Information's Variational Lower bound

$$I\big(c; G(z,c)\big) = H(c) - H(c|G(z,c))$$
$$= \mathbb{E}_{x \sim G(z,c)}\Big[\mathbb{E}_{c' \sim P(c|x)}[\log P(c'|x)]\Big] + H(c)$$
$$= \mathbb{E}_{x \sim G(z,c)}\Big[D_{KL}(P||Q) + \mathbb{E}_{c' \sim P(c|x)}[\log Q(c'|x)]\Big] + H(c)$$
$$\geq \mathbb{E}_{x \sim G(z,c)}\Big[\mathbb{E}_{c' \sim P(c|x)}[\log Q(c'|x)]\Big] + H(c)$$

$$\geq \mathbb{E}_{c \sim P(c),\ x \sim G(z,c)}[\log Q(c|x)] + H(c)$$



Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

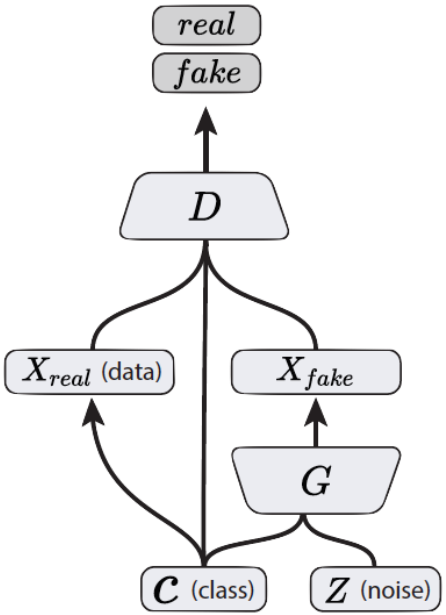
Part 3

- **Conditional GANs**
- **Applications**
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- **Advanced GAN Extensions**
  - Coupled GAN
  - LAPGAN – Laplacian Pyramid of Adversarial Networks
  - Adversarially Learned Inference
- **Summary**

Conditional GANs

Simple modification to the original GAN framework that conditions the model on *additional information* for better multi-modal learning.

Lends to many practical applications of GANs when we have explicit *supervision* available.



Conditional GAN
(Mirza & Osindero, 2014)

Carnegie Mellon University

Mirza, Mehdi, and Simon Osindero. "**Conditional generative adversarial nets**". arXiv preprint arXiv:1411.1784 (2014).
Image Credit: Figure 2 in Odena, A., Olah, C. and Shlens, J., 2016. Conditional image synthesis with auxiliary classifier GANs. *arXiv preprint arXiv:1610.09585*.

Conditional GANs

MNIST digits generated conditioned on their class label.

$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] \rightarrow$

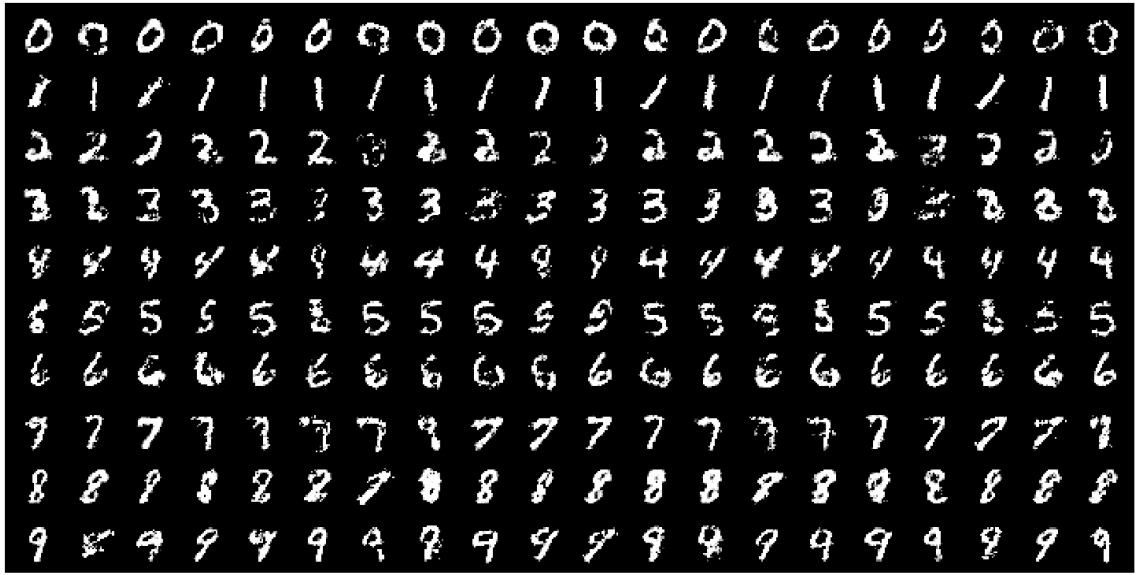
$[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] \rightarrow$

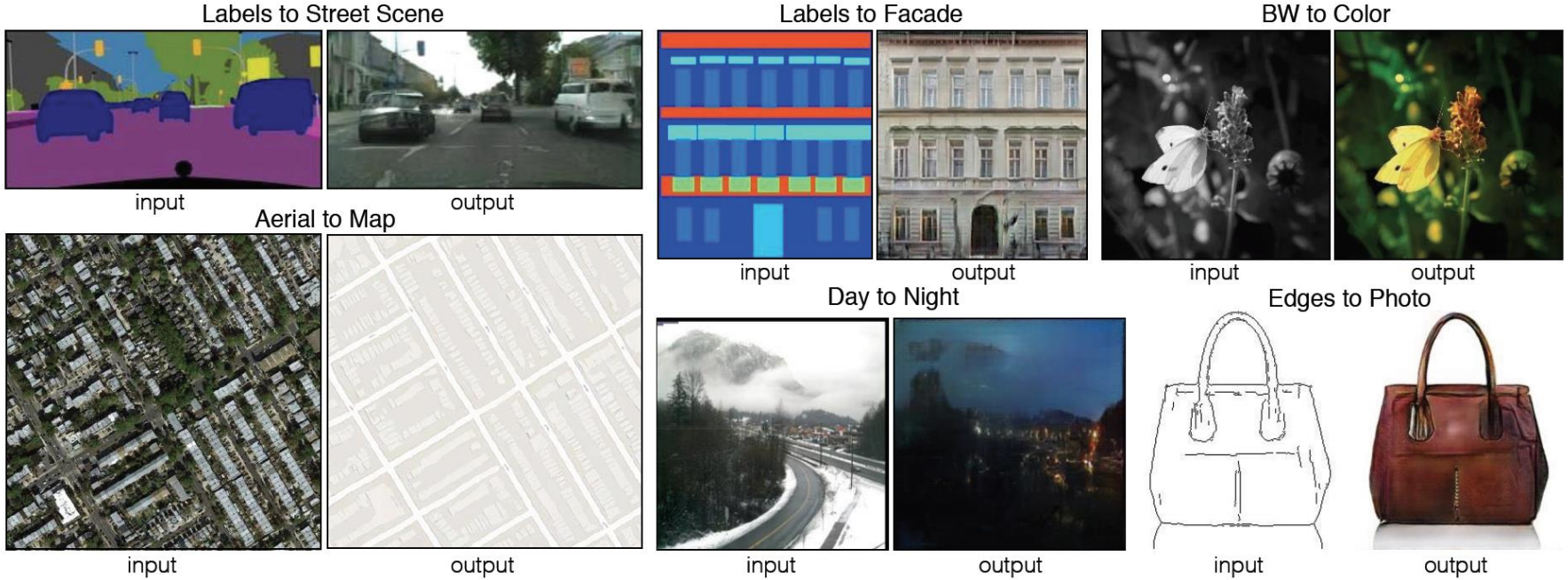
$[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \rightarrow$



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

Carnegie Mellon University

Image-To-Image Translation



Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. "**Image-to-image translation with conditional adversarial networks**". arXiv preprint arXiv:1611.07004. (2016).

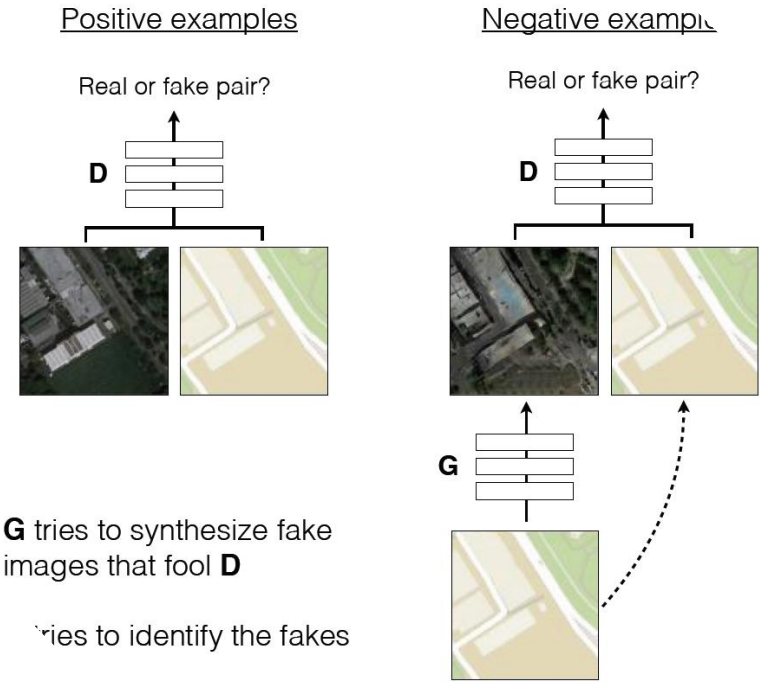
Carnegie Mellon University

Image-To-Image Translation

Architecture: *DCGAN*-based architecture

Training is conditioned on the images from the source domain.

Conditional GANs provide an effective way to handle many complex domains without worrying about designing *structured loss* functions explicitly.



Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. "**Image-to-image translation with conditional adversarial networks**". arXiv preprint arXiv:1611.07004. (2016).

Text-To-Image Synthesis

Motivation

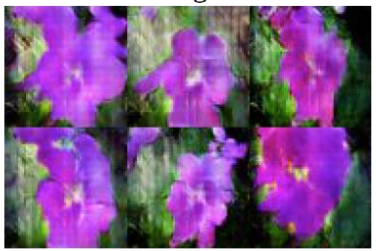
Given a text description, generate images closely associated.

Uses a conditional GAN with the generator and discriminator being conditioned on “dense” text embedding.

this small bird has a pink breast and crown, and black primaries and secondaries.

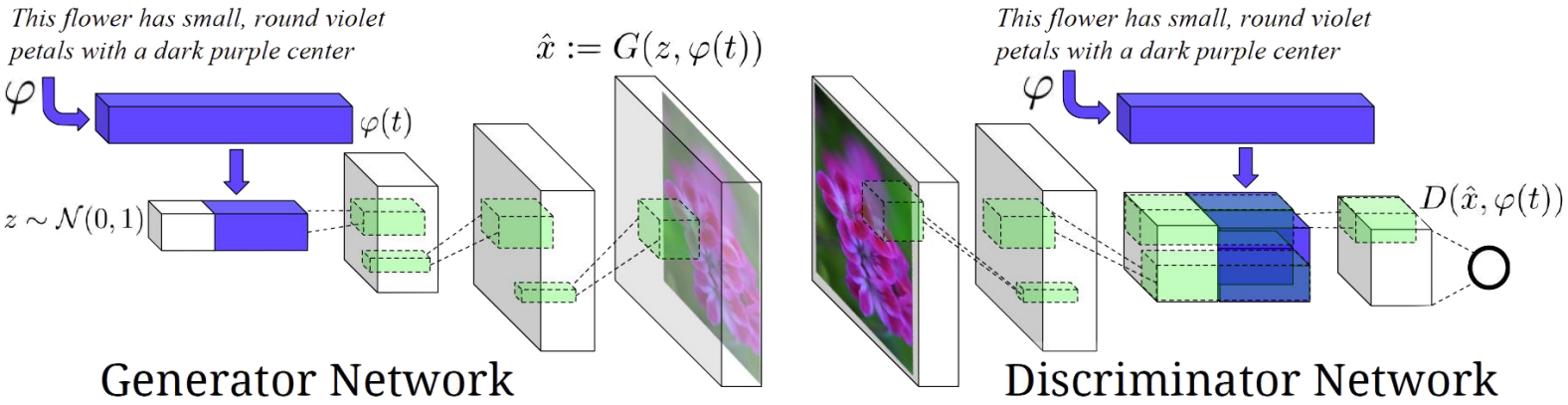
this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma



this white and yellow flower have thin white petals and a round yellow stamen

Text-To-Image Synthesis



Positive Example:
Real Image, Right Text

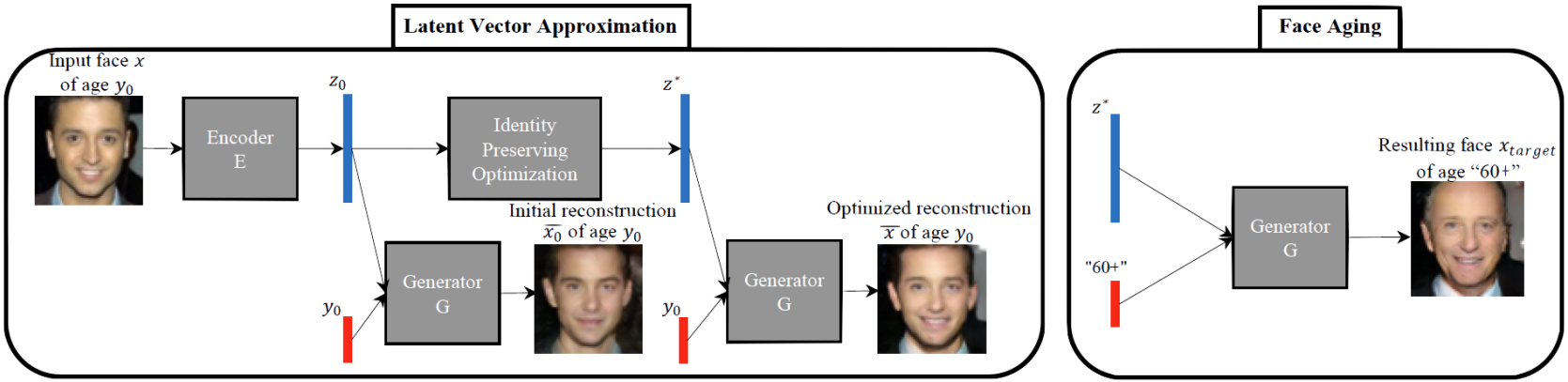
Negative Examples:
Real Image, Wrong Text
Fake Image, Right Text

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. "**Generative adversarial text to image synthesis**". ICML (2016).

Carnegie Mellon University

Face Aging With Conditional GANs

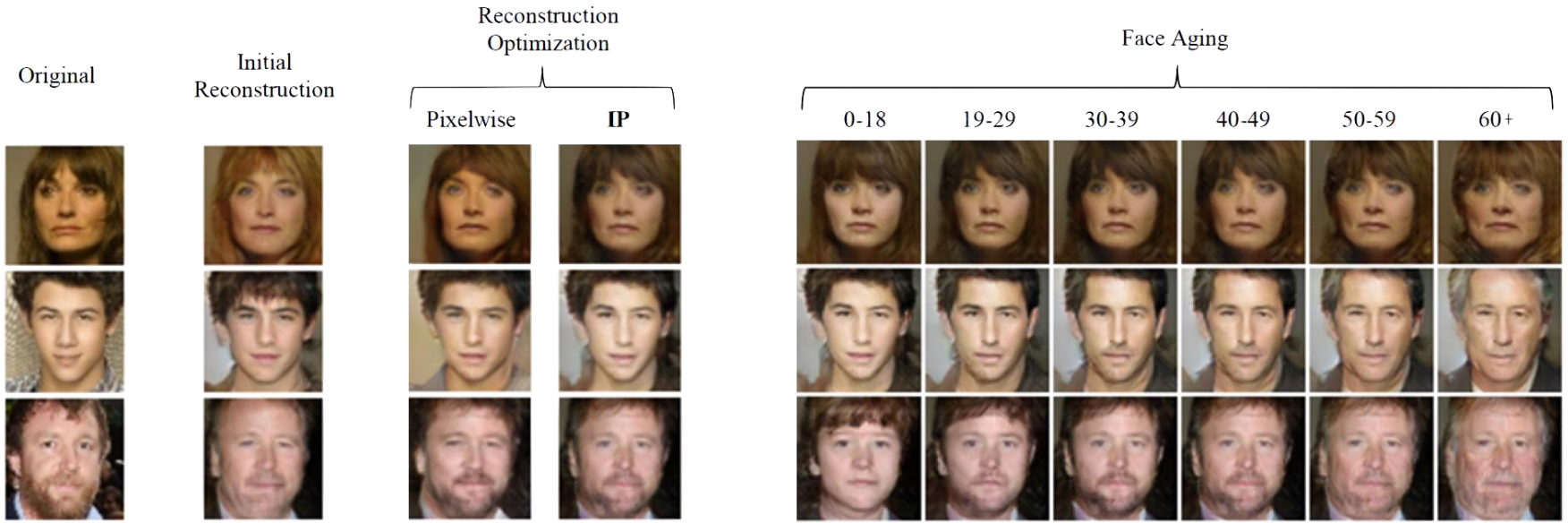
- Differentiating Feature: Uses an *Identity Preservation Optimization* using an auxiliary network to get a better approximation of the latent code (z*) for an input image.
- Latent code is then conditioned on a discrete (one-hot) embedding of age categories.



Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Carnegie Mellon University

Face Aging With Conditional GANs



Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

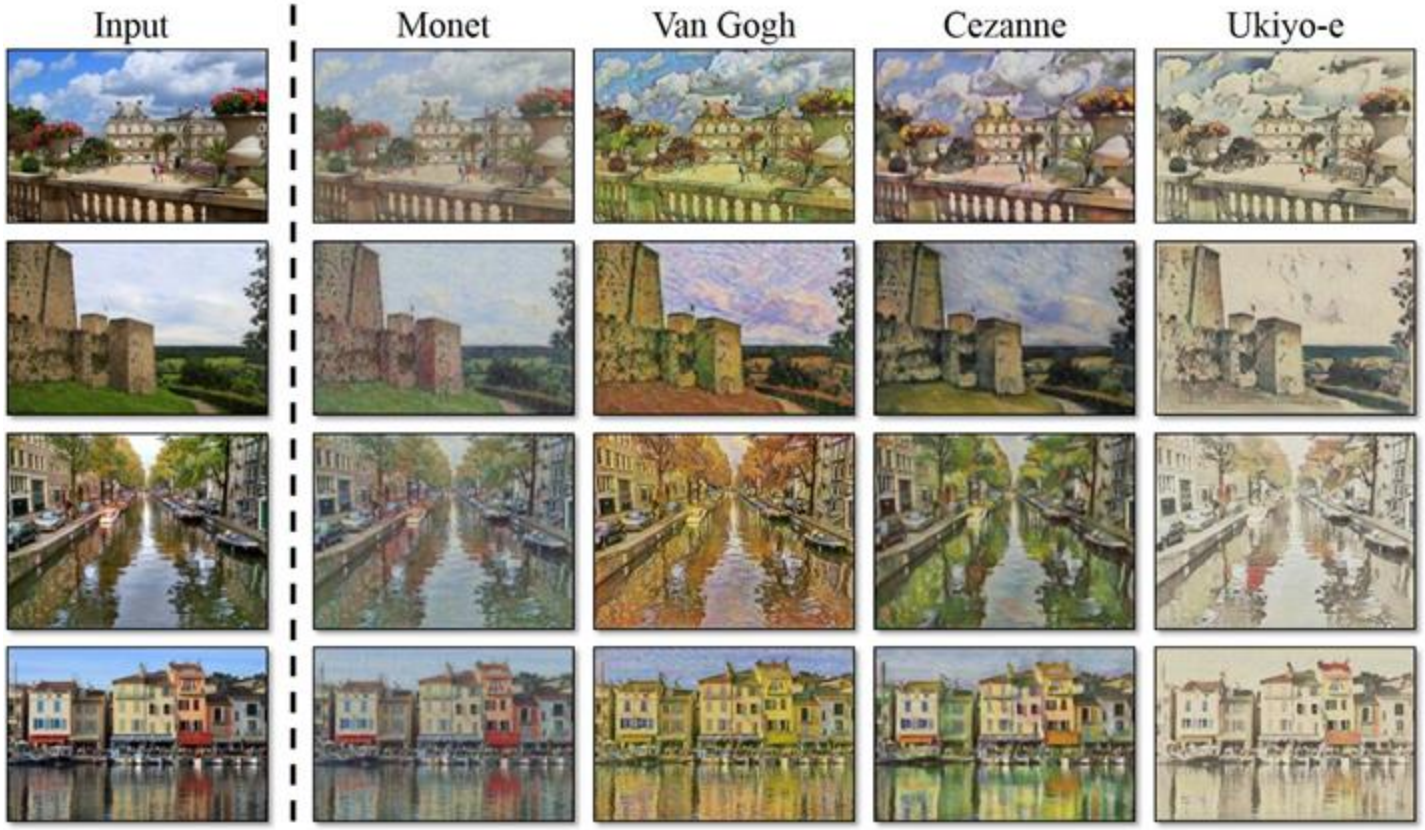
Carnegie Mellon University

Neural Style Transfer Results

Neural Style Transfer Results

Carnegie Mellon University

Style Transfer With CycleGAN



Carnegie Mellon University

Neural Style Transfer vs CycleGAN

- Neural Style Transfer
  - Need a *content* and *style* image
  - Specific & small number of images
  - More control
  - https://reiinakano.com/arbitrary-image-stylization-tfjs/
- CycleGAN
  - Just need 2 domains of images. No need for specific *content* or *style* images
  - Many similar pictures
  - Specificity of images doesn't really matter

CycleGAN For Monet Paintings

I'm Something of a Painter Myself

Introduction to CycleGAN - Monet paintings

https://www.kaggle.com/code/dimitreoliveira/introduction-to-cyclegan-monet-paintings

Carnegie Mellon University

Carnegie Mellon University

Part 3

- Conditional GANs
- Applications
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- **Advanced GAN Extensions**
  - LAPGAN – Laplacian Pyramid of Adversarial Networks
  - Adversarially Learned Inference
- Summary

Adversarial Learned Inference

Nash equilibrium yields

- **Joint:** $p(x, z) \sim q(x, z)$
- **Marginals:** $p(x) \sim q(x)$ and $p(z) \sim q(z)$
- **Conditionals:** $p(x|z) \sim q(x|z)$ and $p(z|x) \sim q(z|x)$

Inferred latent representation successfully reconstructed the original image.

Representation was useful in the downstream semi-supervised task.

Dumoulin, Vincent, et al. "**Adversarially learned inference**". *arXiv preprint arXiv:1606.00704* (2016).

Summary

GANs are generative models that are implemented using two stochastic neural network modules: **Generator** and **Discriminator**.

Generator tries to generate samples from random noise as input

Discriminator tries to distinguish the samples from Generator and samples from the real data distribution.

Both networks are trained adversarially (in tandem) to fool the other component.

In this process, both models become better at their respective tasks.

Why Use GANs For Generation?

- Can be trained using back-propagation for Neural Network based Generator/Discriminator functions.
- Sharper images can be generated.
- Faster to sample from the model distribution: *single* forward pass generates a *single* sample.

Reading List

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial nets, NIPS (2014).
- Goodfellow, Ian NIPS 2016 Tutorial: Generative Adversarial Networks, NIPS (2016).
- Radford, A., Metz, L. and Chintala, S., Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. (2015).
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. Improved techniques for training gans. NIPS (2016).
- Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).
- Zhao, Junbo, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016).
- Mirza, Mehdi, and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).
- Liu, Ming-Yu, and Oncel Tuzel. Coupled generative adversarial networks. NIPS (2016).
- Denton, E.L., Chintala, S. and Fergus, R., 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. NIPS (2015)
- Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016).

- **Applications:**

- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004. (2016).
- Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. Generative adversarial text to image synthesis. JMLR (2016).
- Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). Face Aging With Conditional Generative Adversarial Networks. arXiv preprint arXiv:1702.01983.

Carnegie Mellon University

Conditional GAN

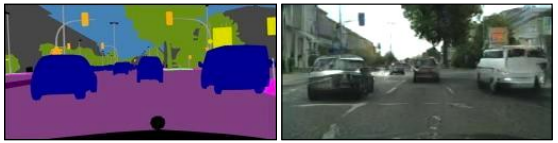
Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

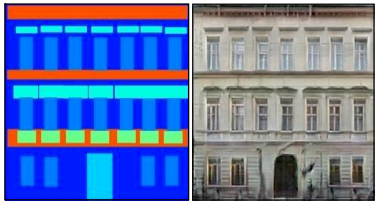
{isola,junyanz,tinghuiz,efros}@eecs.berkeley.edu

Labels to Street Scene



input output

Labels to Facade



input output

BW to Color

input output

Aerial to Map

input output

Day to Night

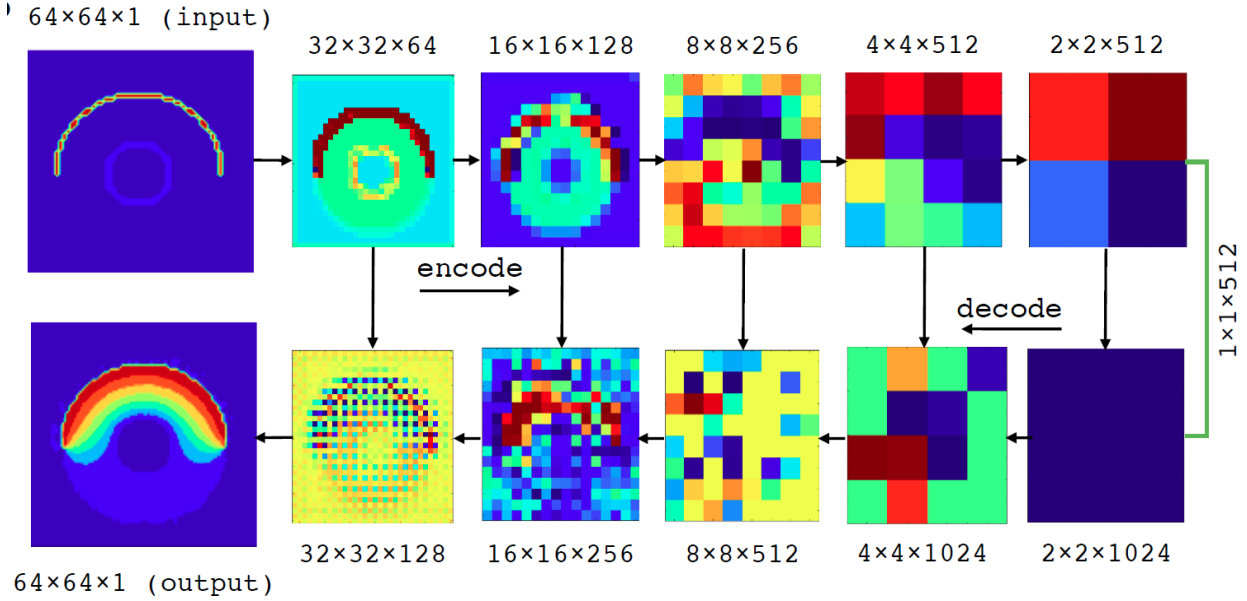
input output

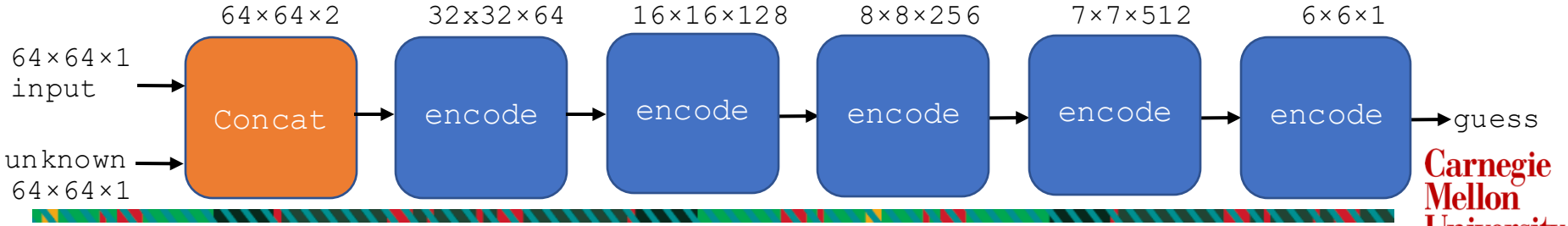
Edges to Photo

input output

Carnegie Mellon University

Conditional GAN





Carnegie Mellon University