Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Sanghyuck Na
April, 19, 2019
Dongguk University
Artificial Intelligence Laboratory
shna@Dongguk.edu

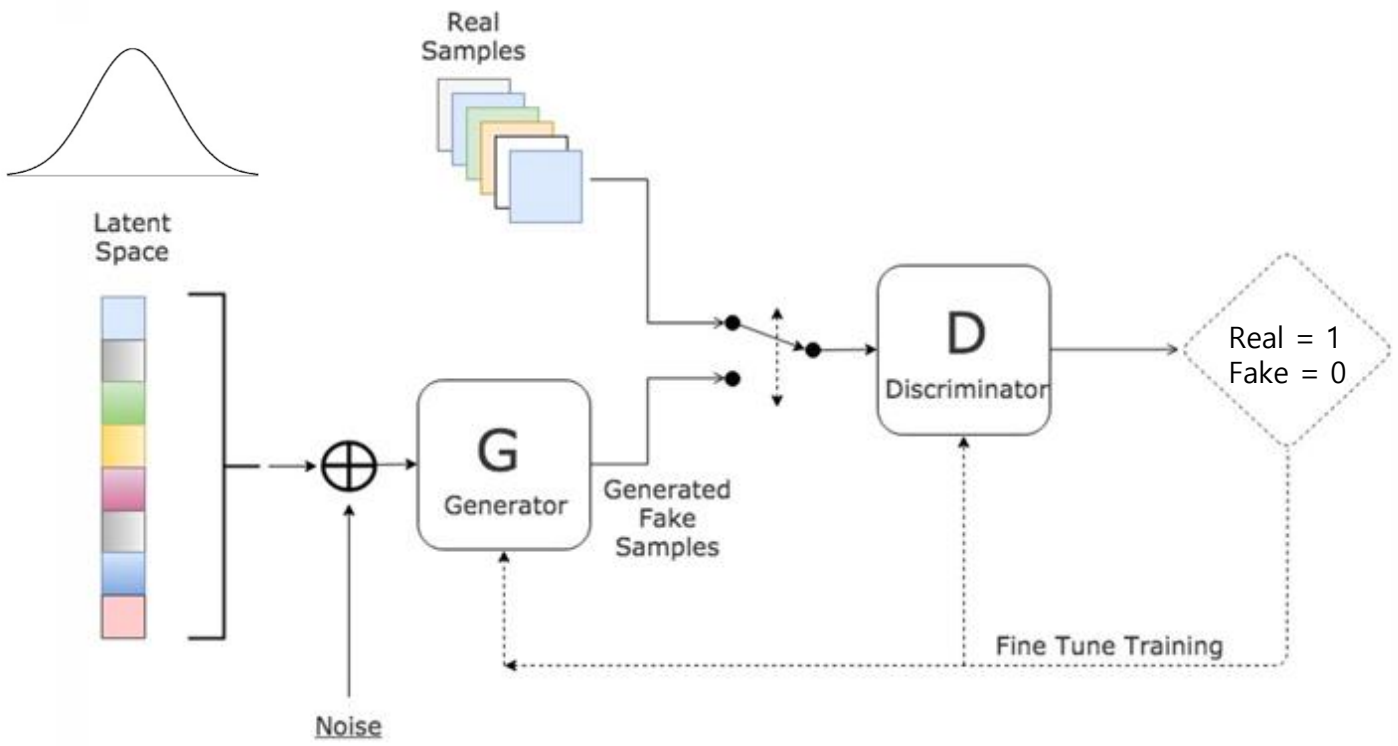
0

Contents

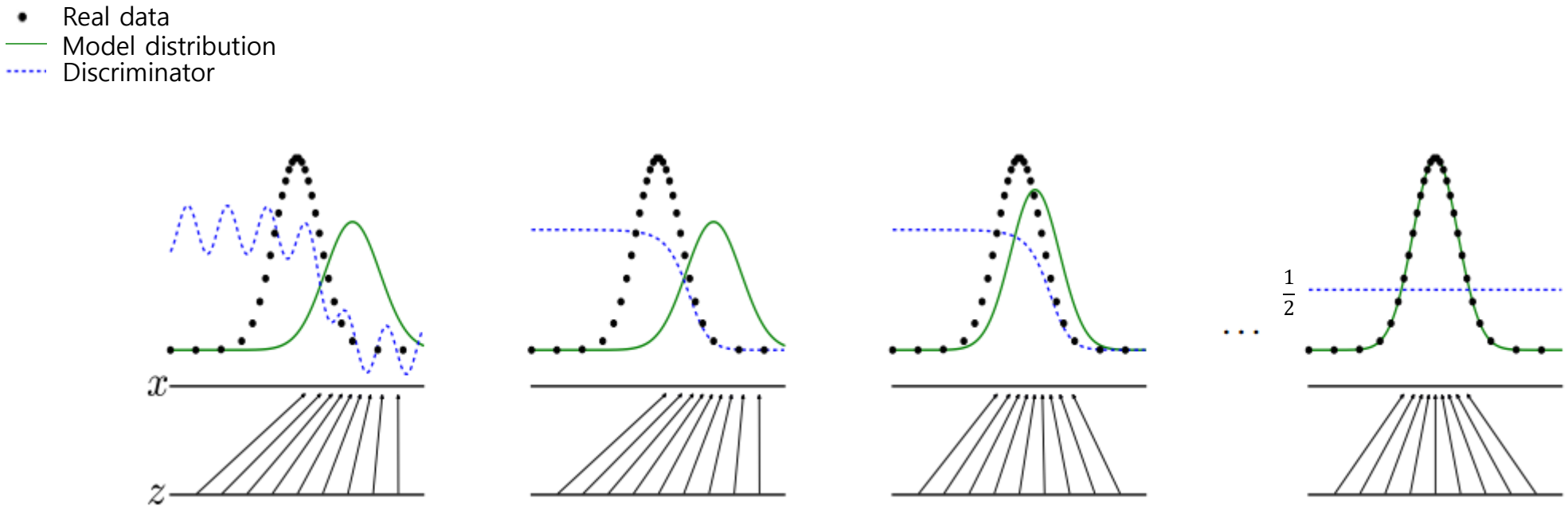
1. Vanilla GAN
2. Introduction
3. Model Architecture
4. Result and Discussion
5. Conclusion & Future work
6. Reference

Generative Adversarial Networks

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_{z\,(z)}}[\log(1 - D(G(z)))]$$



1 Vanilla GAN



Global Optimality of $P_g = P_{data}$

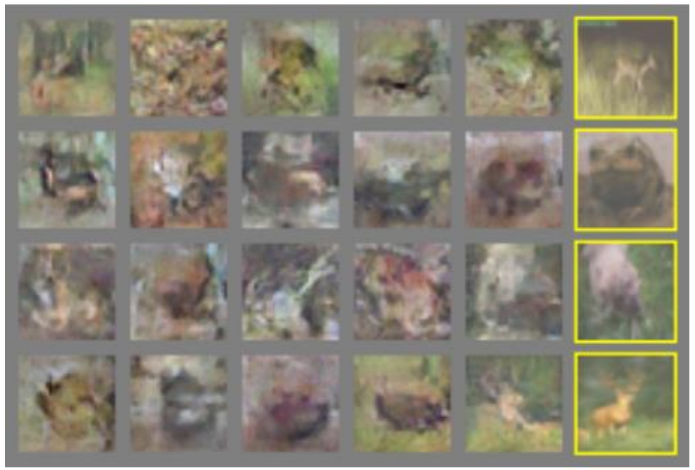
$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

1 Vanilla GAN

MNIST

TFD (Toronto Faces Dataset)

CIFAR-10(fully connected)



CIFAR-10(Convolution discriminator)

ABSTRACT

- Decrease the gap between the success of CNNs for supervised learning and unsupervised learning

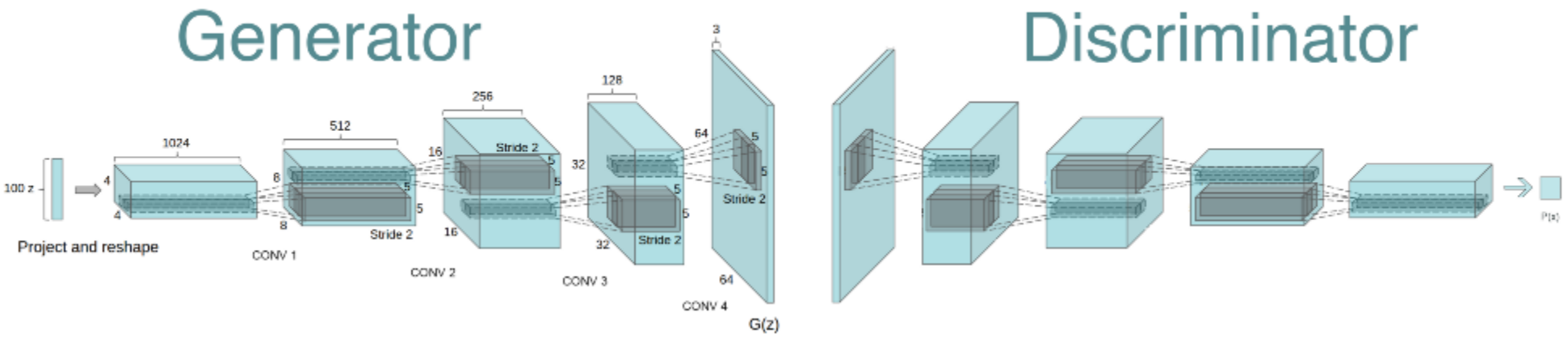
- Learn a hierarchy of representations from object parts to scenes in both the generator and discriminator

Paper

- Replace any pooling layer with strided convolutions (discriminator) and fractional-strided convolutions (generator)
- Use batch-norm in both the generator and the discriminator
- Remove fully connected hidden layer for deeper architectures
- Use ReLU activation in generator for all layers except for the output, which uses Tanh
- Use Leaky ReLU activation in the discriminator for all layer

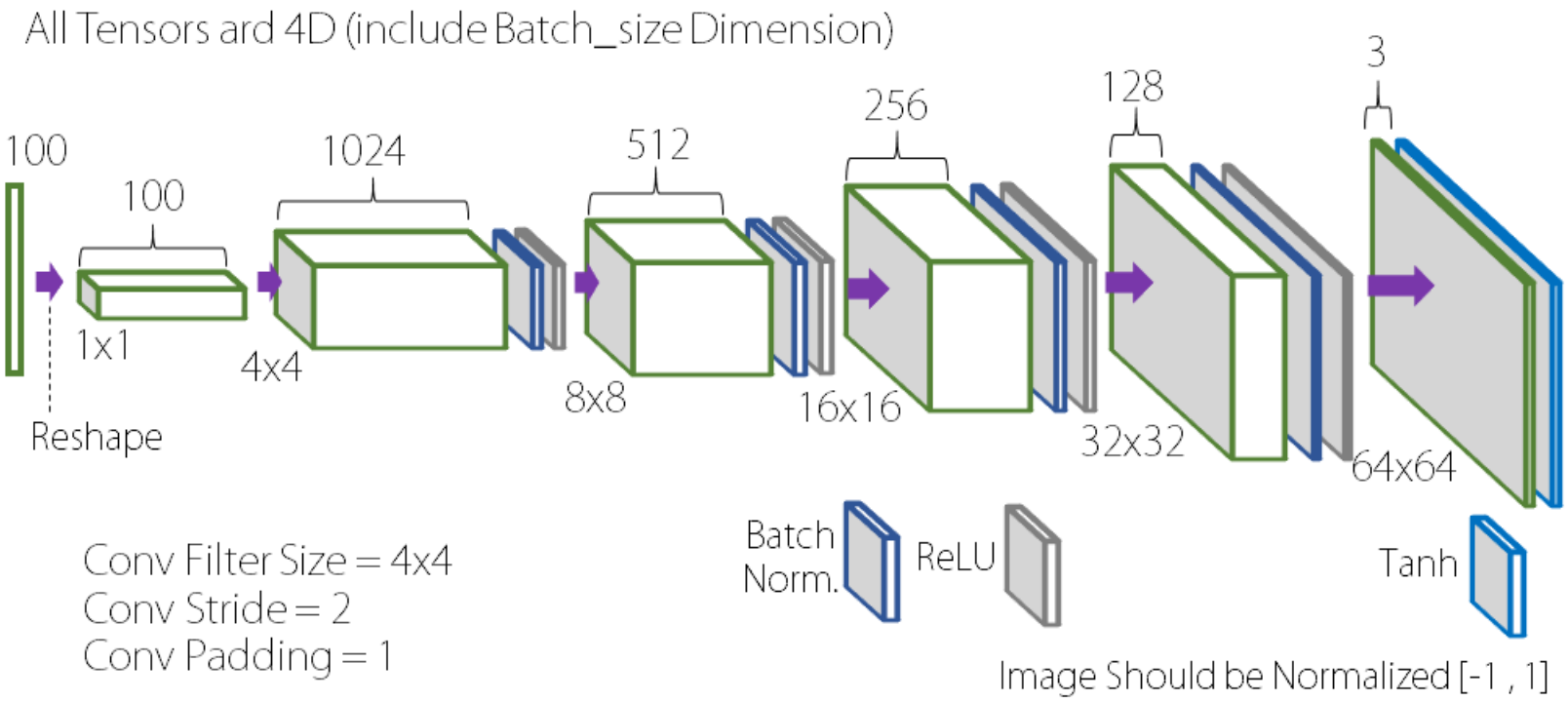
Deep Convolutional Generative adversarial Networks

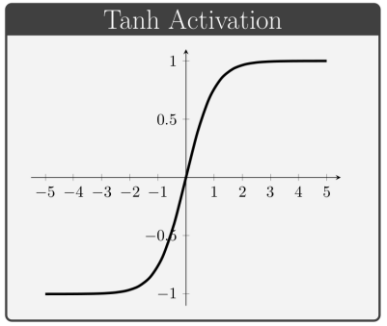
$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_{z\,(z)}}[\log(1 - D(G(z)))]$$



Model Architecture

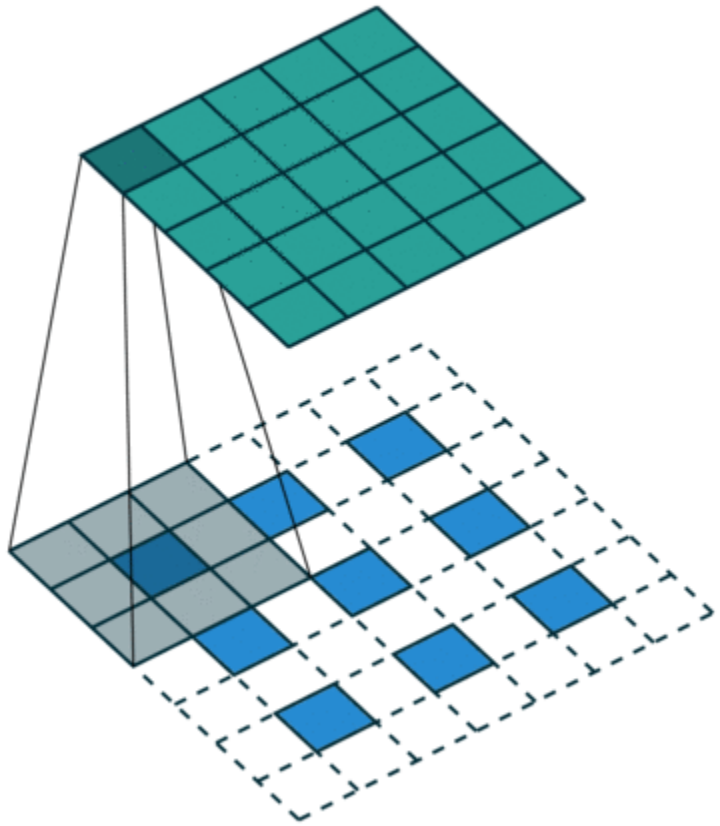
Generator





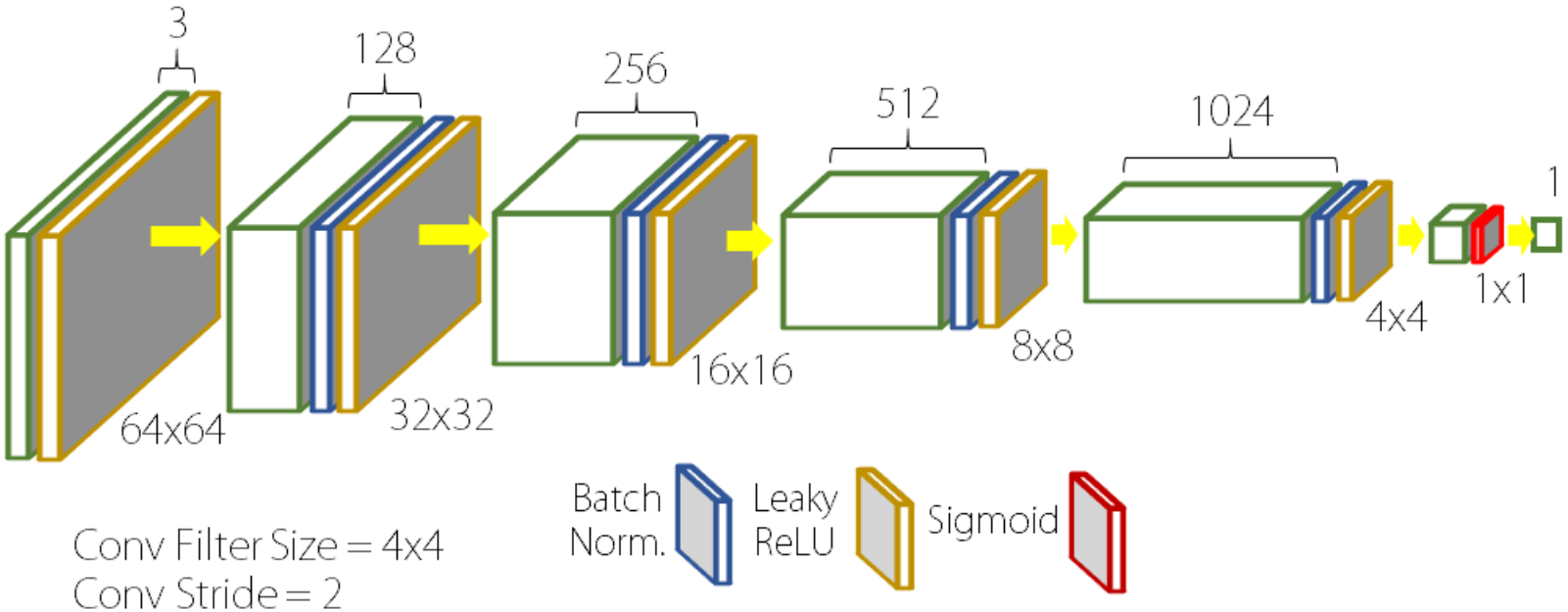
Deconvolutional Neural Network

- Up Sampling
- Fractional-Strided Convolutions
- Zero Padding
- Activation function (Leaky ReLU, Sigmoid)

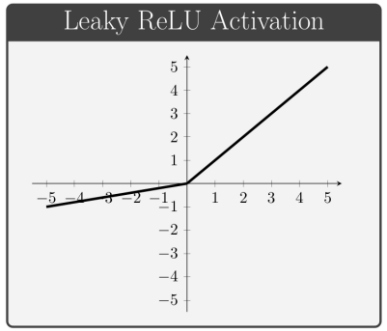


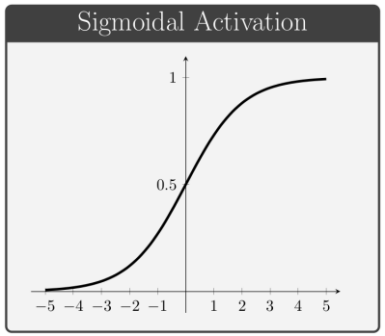
Discriminator

All Tensors ard 4D (include Batch_size Dimension)



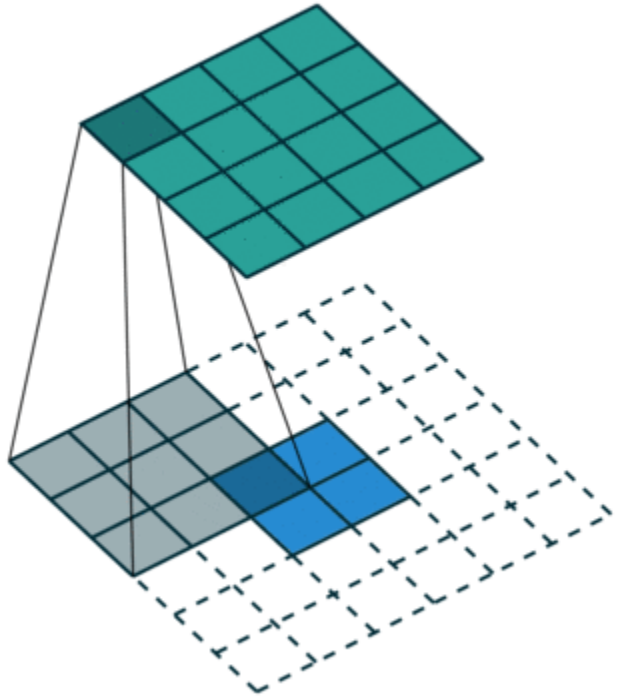
Conv Filter Size = 4x4
Conv Stride = 2
Conv Padding = 1





Convolutional Neural Network

- Down Sampling
- Zero Padding
- Strided Convolutions
- Fully connected networks
- Activation function(ReLU, Tanh)



LSUN

Small learning late and minibatch SGD is not memorization

Repeated noise texture and visual under-fitting

One train LSUN

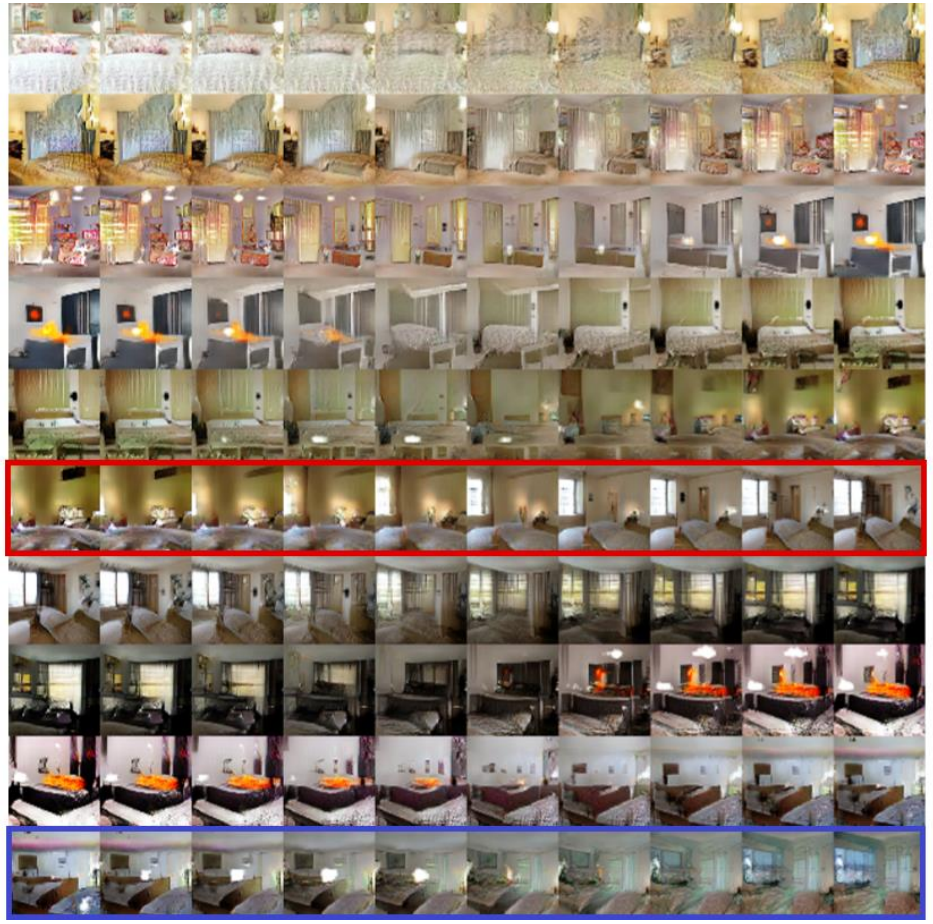
Five train LSUN

Result & Discussion

LSUN

Red bonding box : Create window

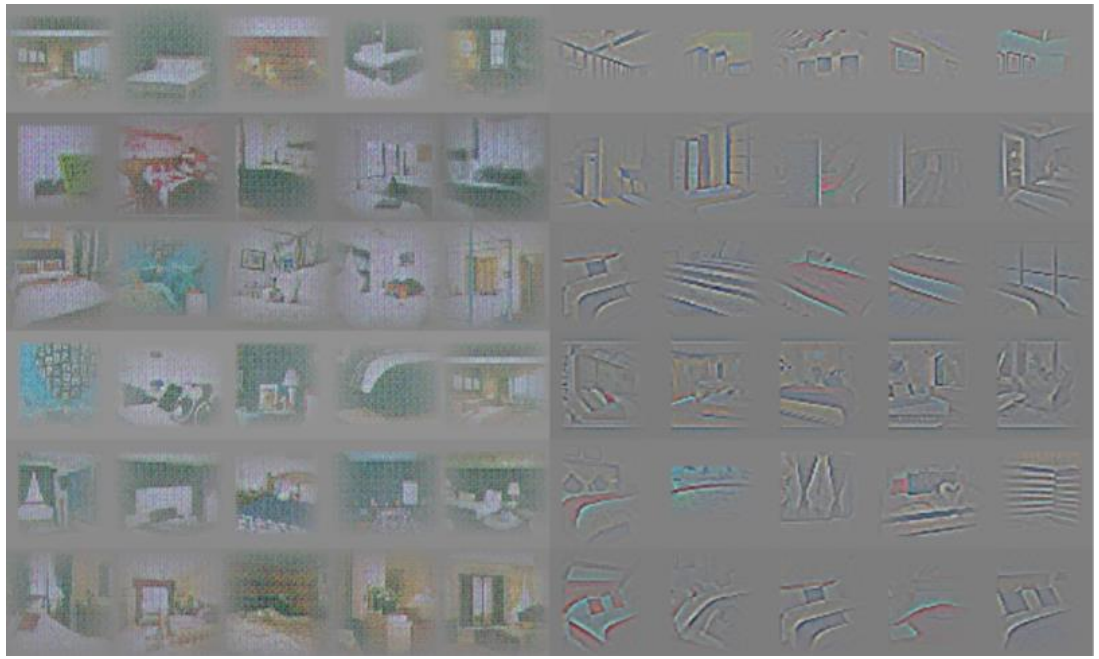
Blue bonding box : TV transform window



Visualizing discriminator feature

Guided backpropagation visualizations

Last convolution layer in the discriminator



Random filters Trained filters

DCGAN Classification

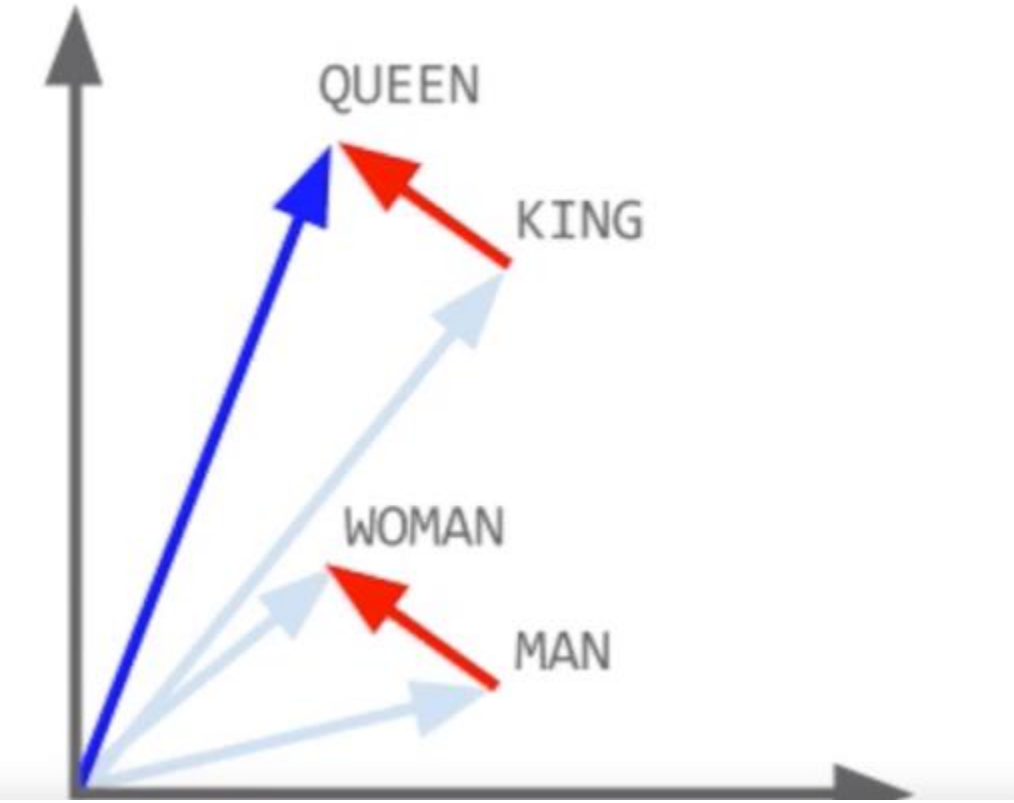
Table 2: SVHN classification with 1000 labels

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

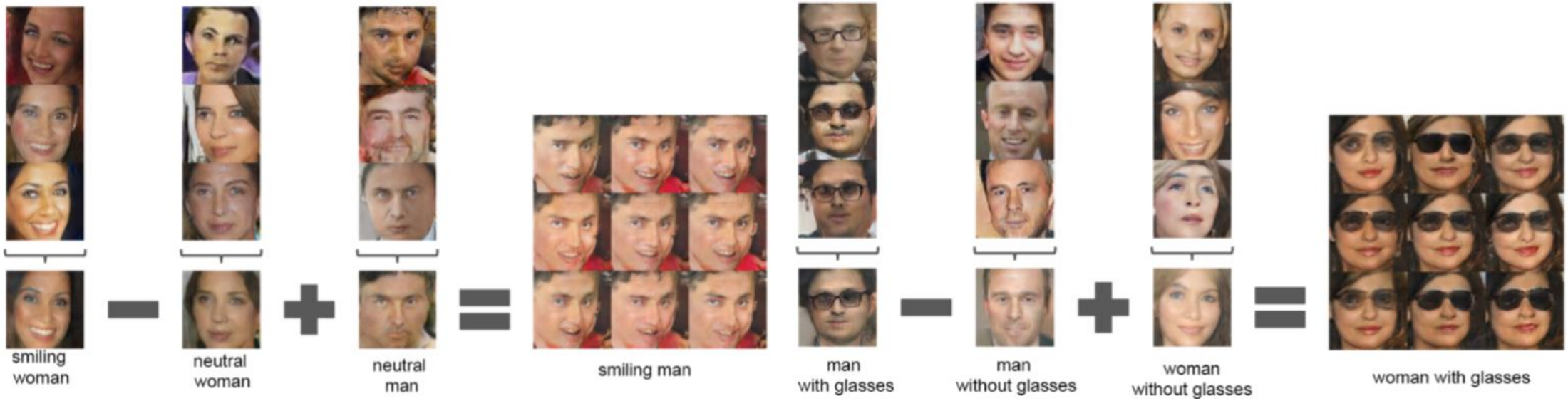
Remove windows

- Visual quality decreased but overall scene composition stayed similar

Vector arithmetic on face samples



Vector arithmetic on face samples



Vector arithmetic on face samples

- Applying arithmetic in the input space results in noisy overlap due to misalignment.

Vector arithmetic on face samples

Nearest Neighbor classification

Table 3: Nearest neighbor classification results.

Model	Test Error @50K samples	Test Error @10M samples
AlignMNIST	-	1.4%
InfiMNIST	-	2.6%
Real Data	3.1%	-
GAN	6.28%	5.65%
DCGAN (ours)	2.98%	1.48%

Groundtruth MNIST GAN DCGAN (ours)

5 Conclusion & Future work

- Trained longer they sometimes collapse a subset of filters to a single oscillating mode.

- Video (for frame prediction) and audio (pre-trained features for speech synthesis) should be very interesting.

Reference

- http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
- https://www.slideshare.net/carpedm20
- https://zzsza.github.io/data/2018/02/23/introduction-convolution/
- https://kakalabblog.wordpress.com/2017/06/04/unsupervised-representation-learning-with-dcgan-2016-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0/
- https://brunch.co.kr/@kakao-it/145

THANKYOU