



# UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

## Abstract

- Computer Vision 분야에서 자주 이용되는 CNN의 지도 학습과 적게 이용되는 CNN의 비지도 학습 사이의 간극을 줄이고자 함
- Deep Convolution Generative Adversarial Network (DCGANs) 소개

## Introduction

- Convolutional GAN의 제약조건을 제안하고 평가하며 학습과정이 안정화되도록 함
- 이미지 분류에서 학습된 식별자를 이용해 다른 비지도 방법들과 비교
- 특정 물체를 그리기까지 학습된 필터들을 보임
- 생성자의 벡터 연산적 특성을 이용해 쉽게 조작 가능

## Related Work

1. 비지도 표현 학습의 여러 방법
  - Clustering

- Auto-encoder
  - Ladder Structure
2. 이미지 생성 모델 : Parametric & Non - Parametric
    1. Parametric : Noise, 품질 문제 등 성공적 X
    2. Non - Parametric : 성공적 O
      - Texture Synthesis, Super-Resolution, In-Painting
  3. CNNs 내부 과정/알고리즘 시각화
    - 비판 : black-box method
    - deconvolution & filtering maximal activation
      - 각 convolution filter의 목적 보임
    - input에 대한 gradient descent
      - 특정 filter를 활성화하는 이상적인 이미지 검사

## Approach and Model Architecture

효과 : allow stable training, higher resolution, deeper generative model

Architecture guidelines for stable Deep Convolutional GANs

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

1. pooling 대신 strided convolution로 downsampling, 생성자에서 upsampling 구현
2. convolutional features 맨 위 fully connected layer 삭제
  - global average pooling로 model stability 향상, convergence 속도 저하
3. batch Normalization로 input을 normalize함
 

모든 layer에 적용시 oscillation and model instability

  - 생성자 output layer과 식별자 input layer에는 적용 X

#### 4. ReLU activation 사용

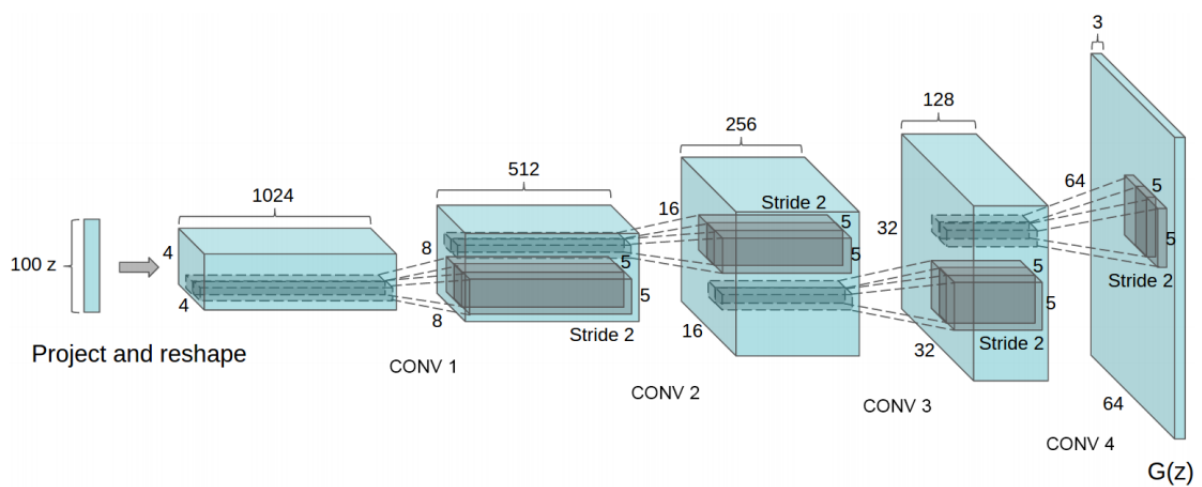
→ convergence 속도 향상, 공간에서 색깔 빠르게 커버

(예외적으로 output layer만 Tanh function 사용)

식별자에서는 leaky rectified activation 사용

→ higher resolution modeling 구현

#### Generator Model



- pre-processing 대신 scale tanh activation function  $[-1, 1]$
- mini batch size 128 + stochastic gradient descent
- weight initialized : zero-centered Normal distribution with standard deviation 0.02
- LeakyReLU : slope = 0.2
- Adam optimize : learning rate = 0.0002,  $\beta_1 = 0.5$

## LSUN

- produce high quality sample :
  - not using overfitting/memorizing example
  - no data augmentation applied to image

- simple image de-duplication process 거침  
생성자의 input 암기 방지  
convert image to code by using autoencoder and eliminate similar code

## INVESTIGATE & VISUALIZE THE INTERNALS OF THE NETWORKS

- DCGAN의 error rate가 제일 낮음

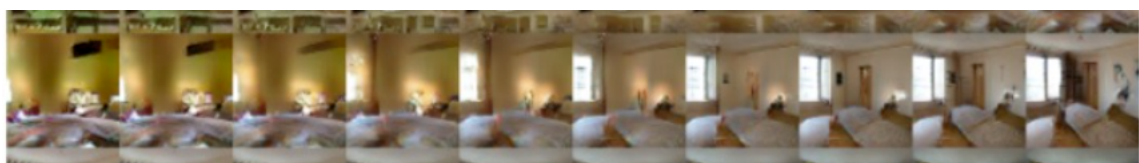
Table 2: SVHN classification with 1000 labels

| 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) |

### 1. walking in the latent space



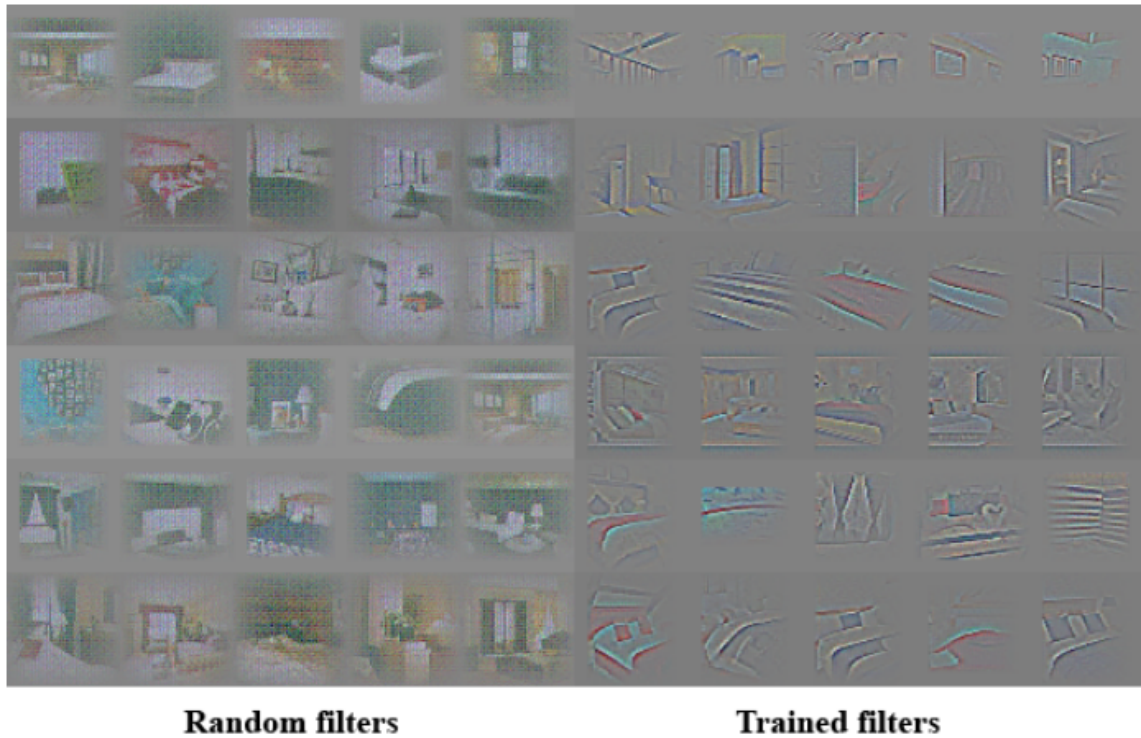
→moving slowly in Latent Space



→ chang in that situation is significant meaningful for creating image

## 2. visualizing the discriminator features

large image에 대한 비지도 학습에서도 interesting feature를 hierarchy에 따라 학습 가능



guided backpropagation 과정

## 3. manipulating the generator representation

장면에서 특정 요소 학습



→ 창문을 제거함

scale, rotation, position 이해하는 네트워크



→ 왼쪽을 보는 것을 오른쪽을 보는 것으로 바꿈