# Computer Vision

권지혜, 김미라

# Index

1. Convolutional Neural Network (CNN)

2. Detection/Segmentation
   - Object Detection
   - Image Segmentation

3. Generative Model
   - VAE
   - GAN

# Index

1. Convolutional Neural Network (CNN)

2. Detection/Segmentation
   - Object Detection
   - Image Segmentation

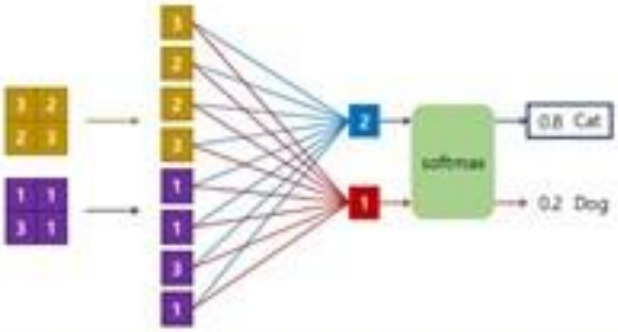
3. Generative Model
   - VAE
   - GAN

# 1. Fully Connected Layer의 한계

- Fully Connected Layer의 input data는 1차원 배열 형태
- input data가 한 장의 컬러 사진일 경우 3차원 → 1차원으로 평면화
- 3차원, 4차원 데이터의 '공간 정보' 손실

  ⇒ 이미지의 공간 정보 유실로 인한 정보 부족

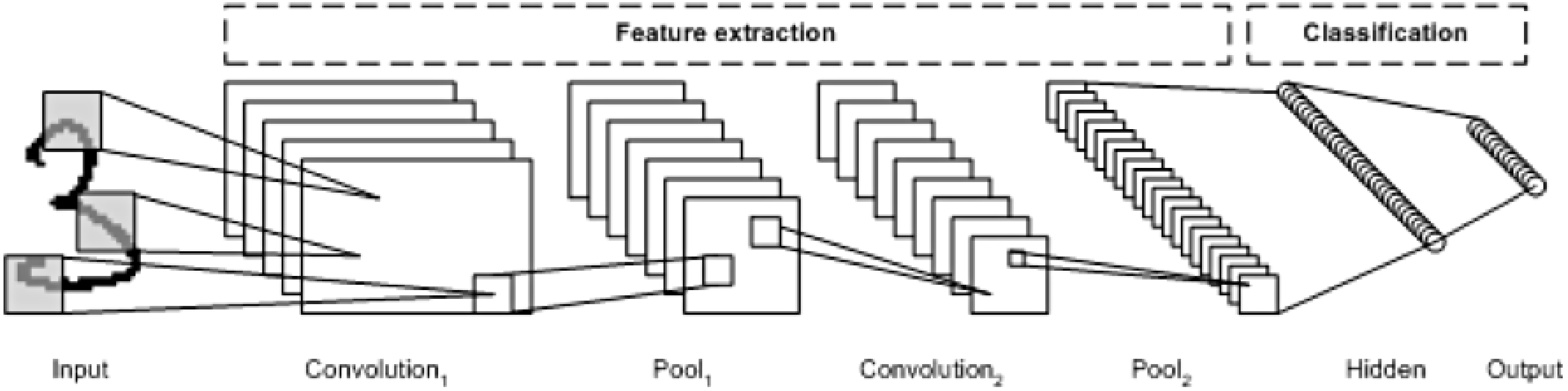
  ⇒ 특징 추출이 어렵고 비효율적인 학습, 낮은 정확도




https://www.slideshare.net/JinwonLee9/ss-70446412

∴ 이미지의 **공간 정보**를 유치한 채로 학습 가능: CNN

# 2. CNN, Convolutional Neural Network



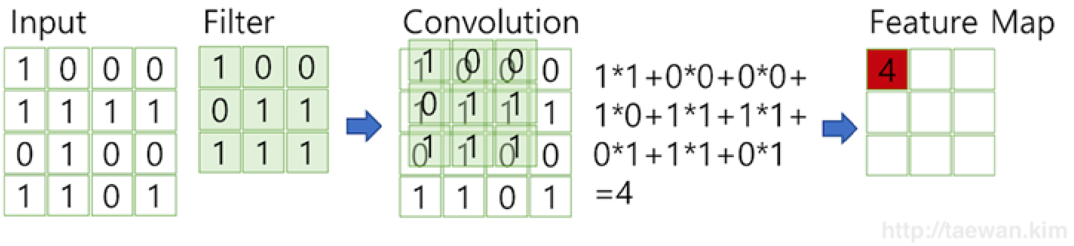
# 3. FC Layer 대비 CNN의 특징

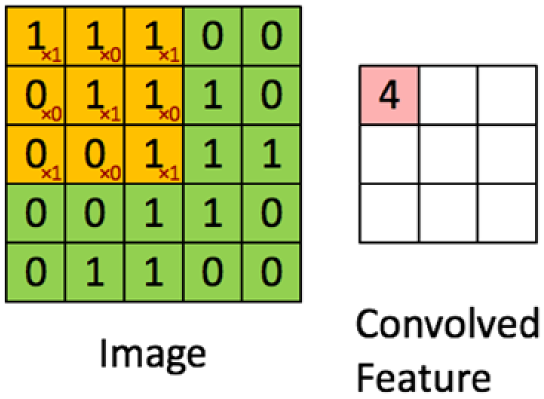
- 각 레이어의 입출력 데이터의 형상 유지
- **이미지의 공간 정보 유지**, 인접 이미지와의 특징을 효과적으로 인식
- 복수의 필터로 이미지의 특징 추출 및 학습
- 추출한 이미지의 **특징을 모으고 강화하는** Pooling Layer
- 필터를 공유 파라미터로 사용하기 때문에, 일반 인공 신경망과 비교하여 **적은 수의 학습 파라미터**

# 4. CNN main terms

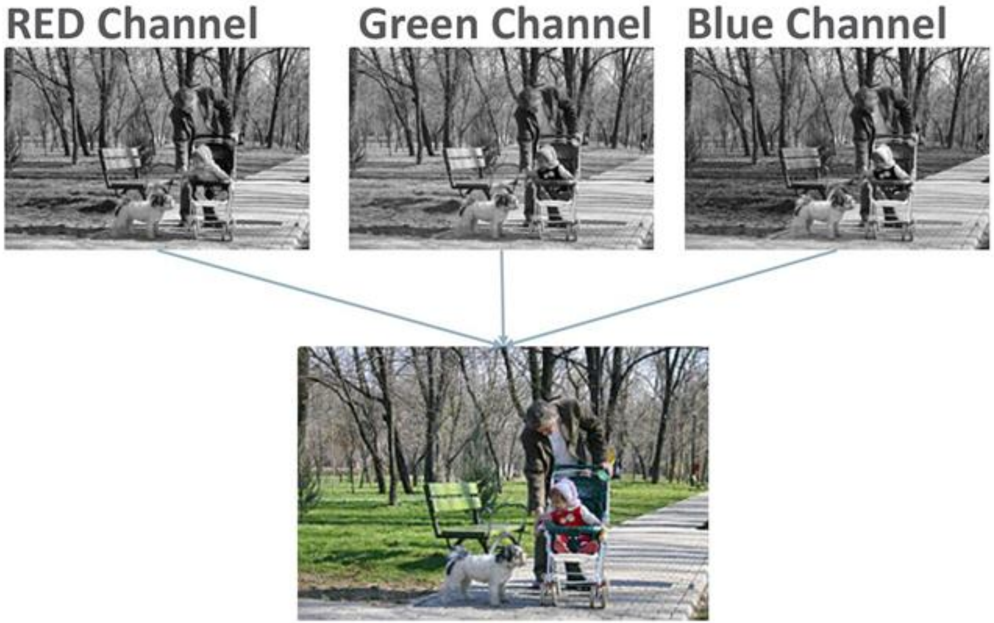
- Convolution (합성곱)
- Channel (채널)
- Filter (필터)
- Kernel (커널)
- Stride (스트라이드)
- Padding (패딩)
- Activation Map (액티베이션 맵)
- Pooling Layer (풀링 레이어)

# 4-1. Convolution (합성 곱)



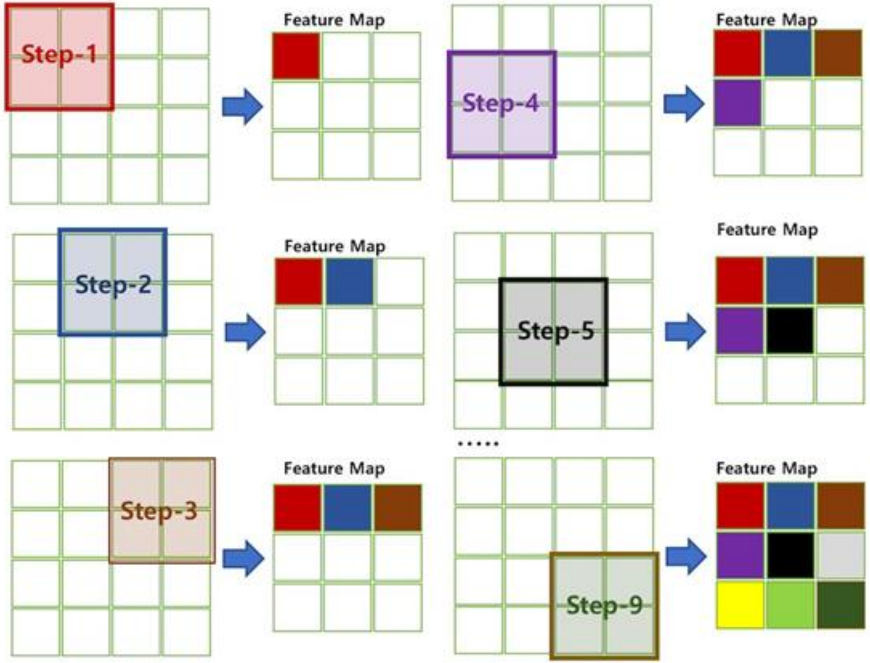



# 4-2. 채널 (Channel)

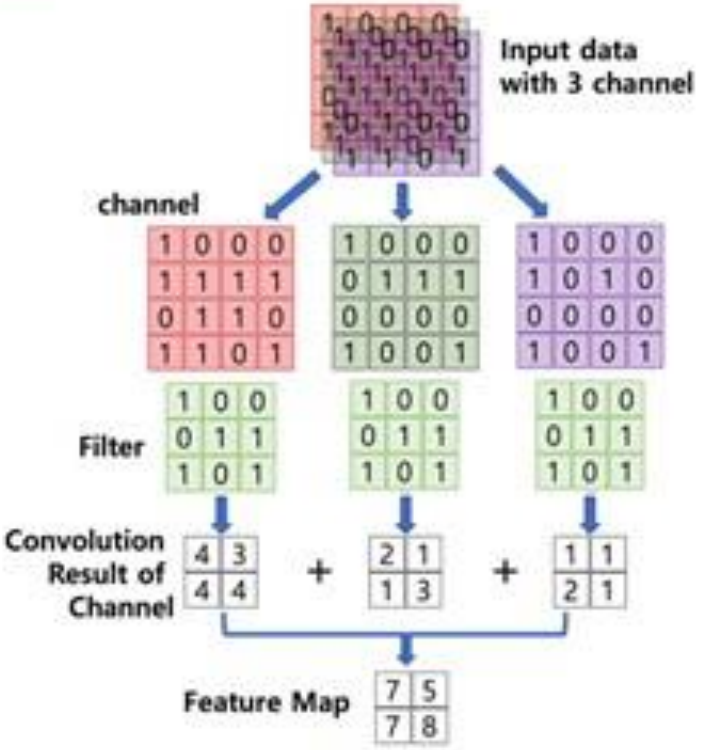



이미지 출처: https://en.wikipedia.org/wiki/Channel_(digital_image)

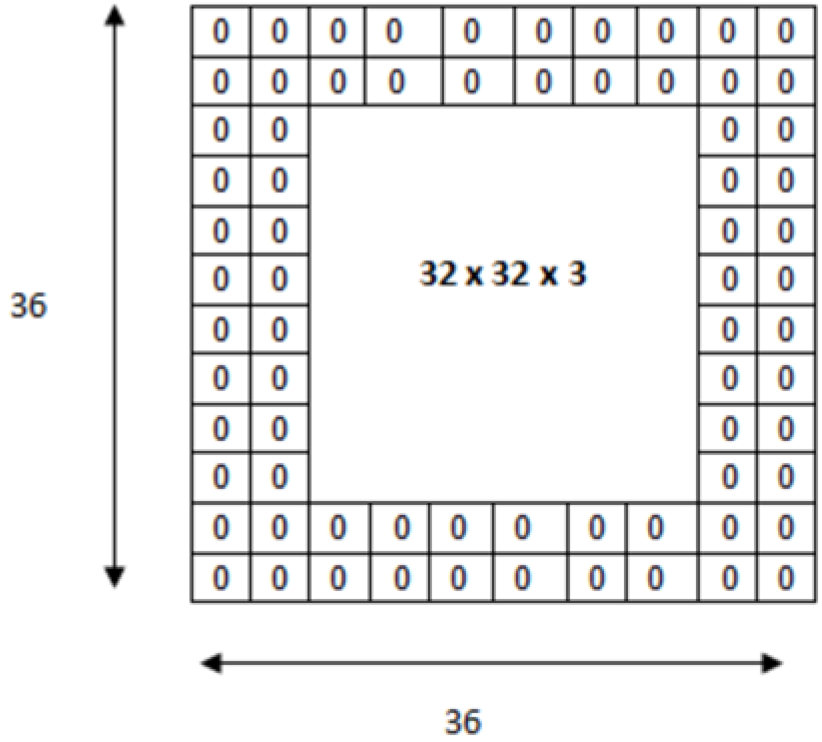
# 4-3. 필터 (Filter) & 스트라이드 (Stride)



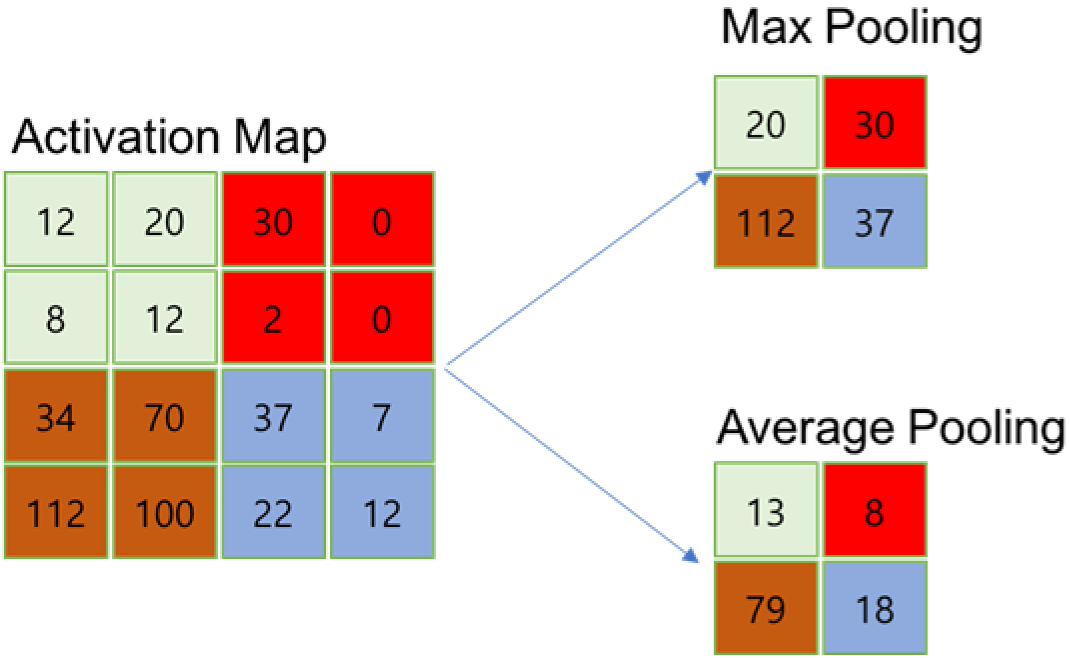
# 4-3. 필터(Filter) & 스트라이드(Stride)



# 4-4. 패딩 (Padding)



# 4-5. Pooling Layer



# 5. Convolution Layer 출력 데이터 산정

입력 데이터 높이: H

입력 데이터 폭: W

필터 높이: FH

필터 폭: FW

Stride 크기: S

패딩 사이즈: P

$$OutputHeight = OH = \frac{(H + 2P - FH)}{S} + 1$$

$$OutputWeight = OW = \frac{(W + 2P - FW)}{S} + 1$$

# 5. 레이어 별 출력 데이터 산정

$$OutputHeight = OH = \frac{(H + 2P - FH)}{S} + 1$$
$$OutputWeight = OW = \frac{(W + 2P - FW)}{S} + 1$$

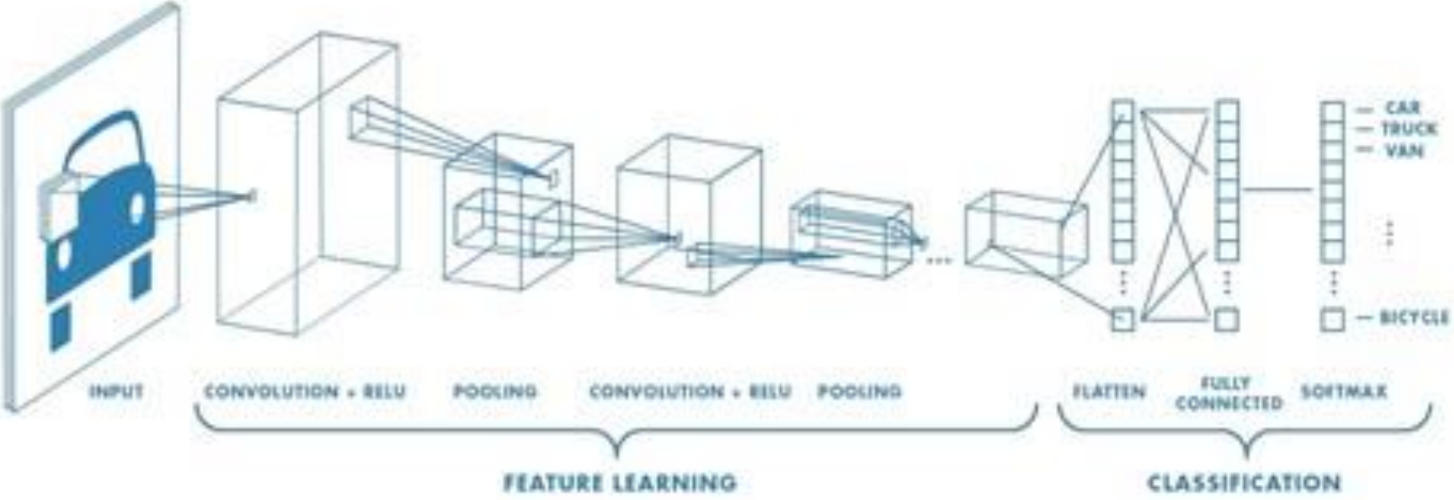
Convolution Layer 1의 기본 정보는 다음과 같습니다.

- 입력 데이터 Shape = (39, 31, 1)
- 입력 채널=1
- 필터=(4, 4)
- 출력 채널=20
- Stride = 1

**Activation Map의 Shape**

$$RowSize = \frac{N - F}{Strid} + 1 = \frac{39 - 4}{1} + 1 = 36$$
$$ColumnSize = \frac{N - F}{Strid} + 1 = \frac{31 - 4}{1} + 1 = 28$$

# 6. CNN Architecture




https://kr.mathworks.com/solutions/deep-learning/convolutional-neural-network.html

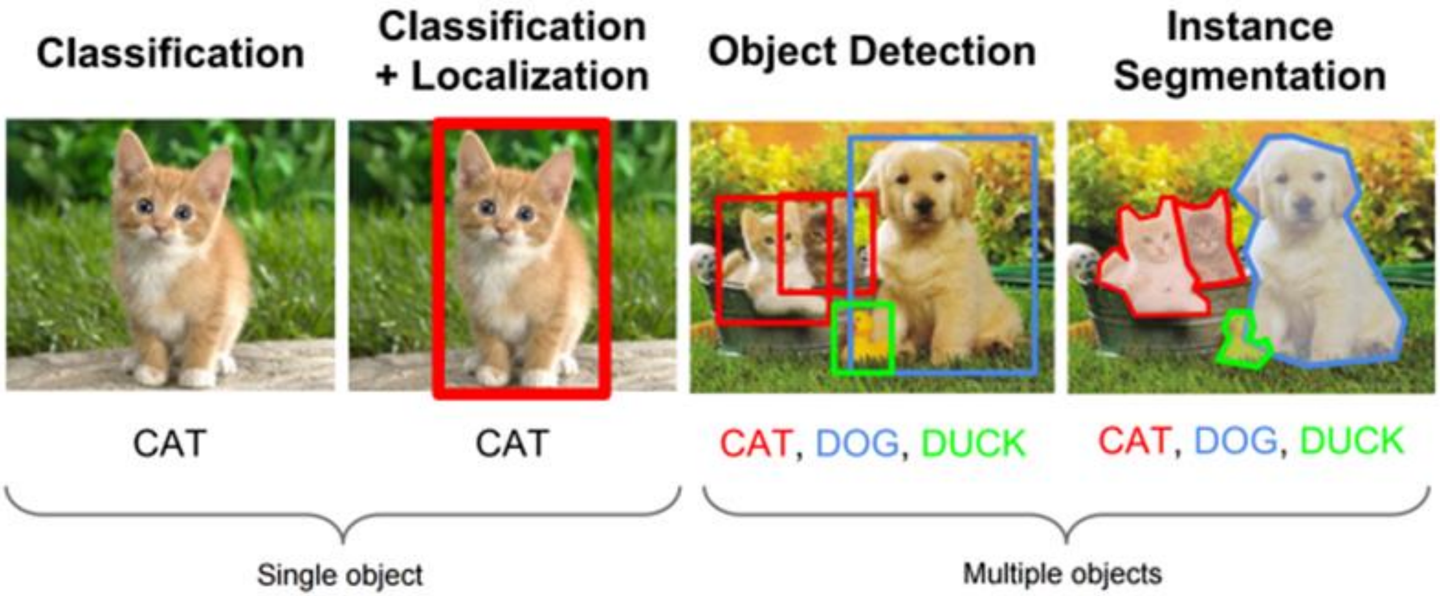
# Index

1. Convolutional Neural Network (CNN)

2. Detection/Segmentation
   - Object Detection
   - Image Segmentation

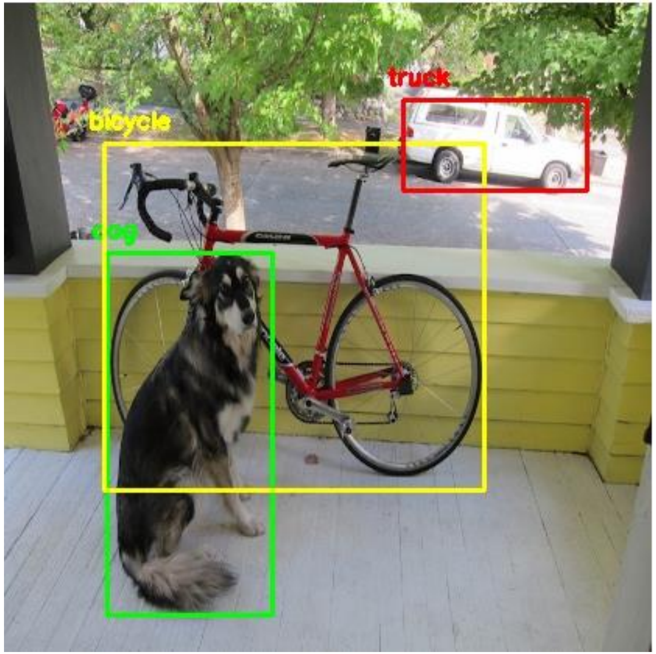
3. Generative Model
   - VAE
   - GAN

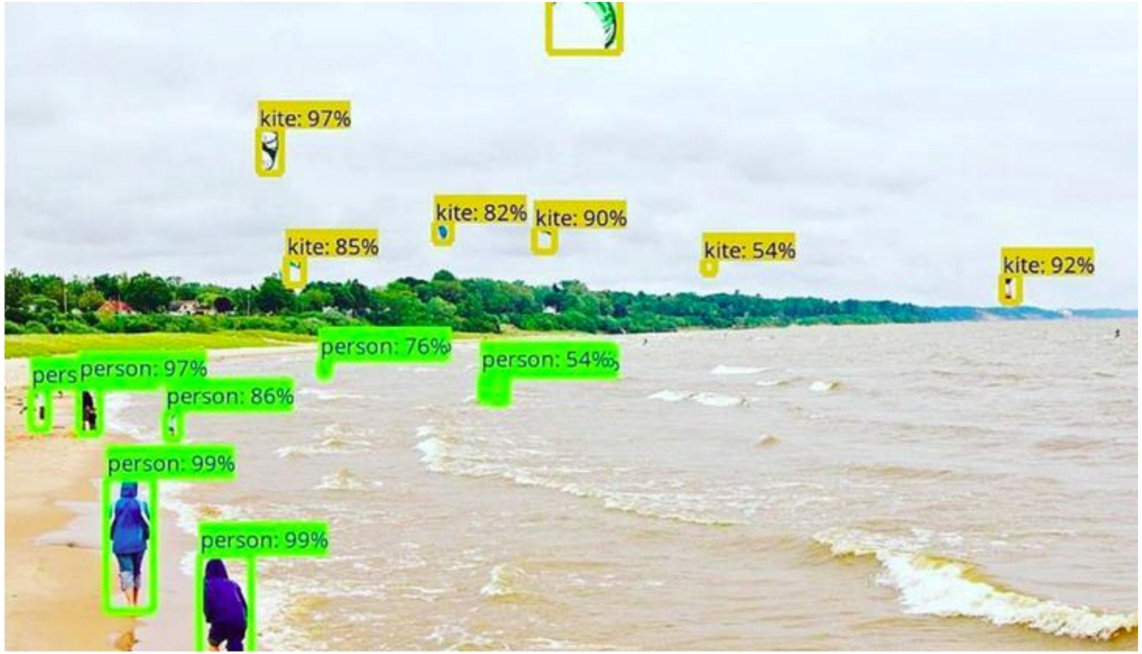
# 1. Object Detection이란?



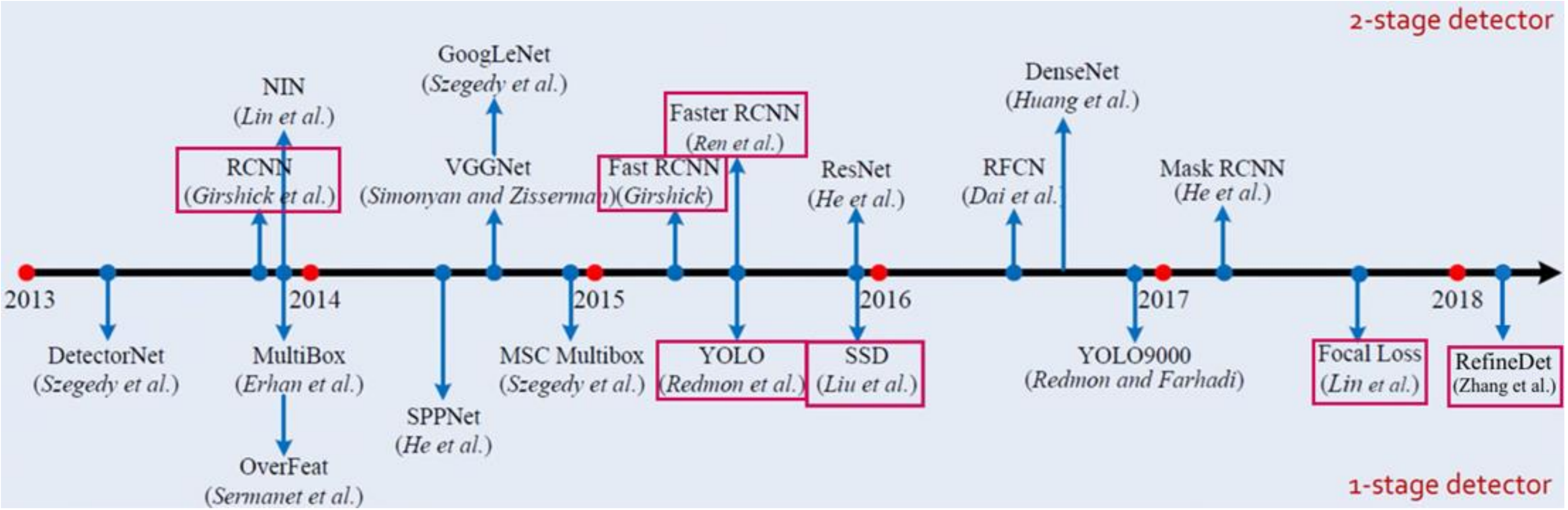

Object Detection = Multi-Labeled Classification + Bounding Box Regression(Localization)

# 1. Object Detection이란?



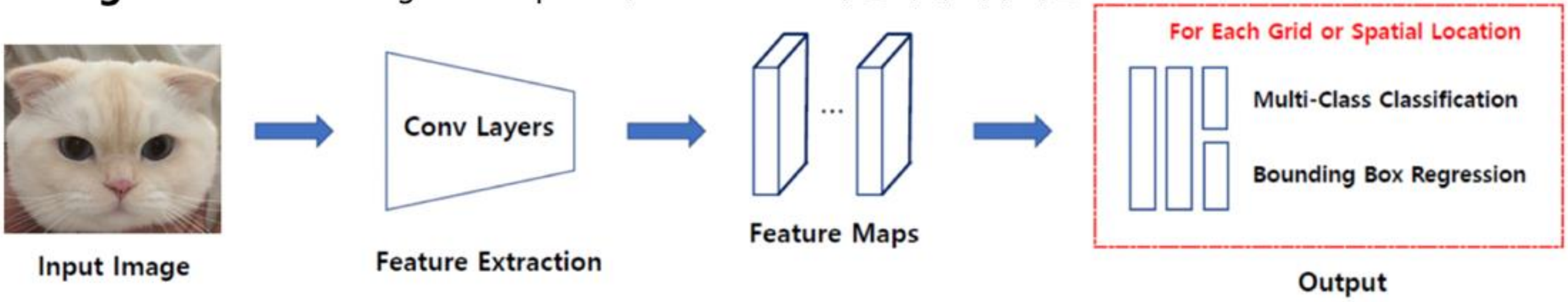



# 2. 2-Stage Detector VS 1-Stage Detector



# 2. 1-Stage Detector VS 2-Stage Detector

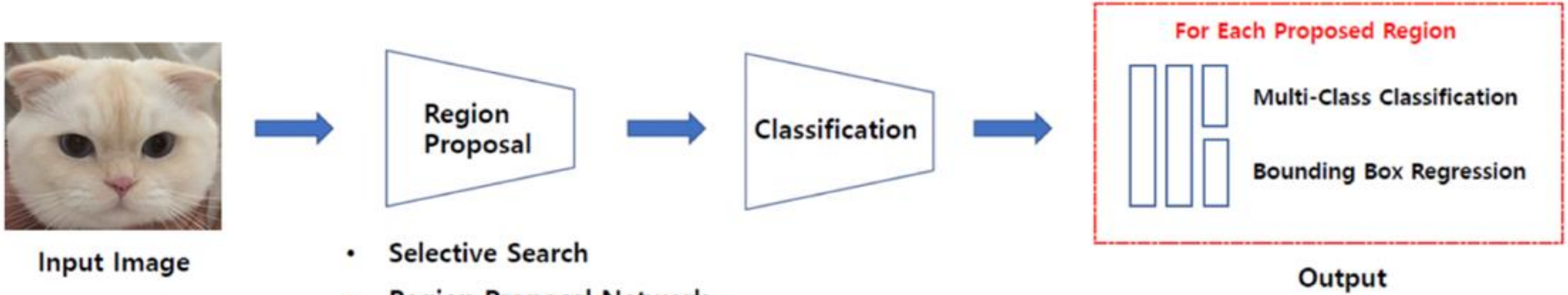
**1-Stage Detector -** Regional Proposal와 Classification이 동시에 이루어짐.




Ex) **YOLO 계열** (YOLO v1, v2, v3) , **SSD 계열** (SSD, DSSD, DSOD, RetinaNet, RefineDet ... )

# 2. 2-Stage Detector VS 1-Stage Detector

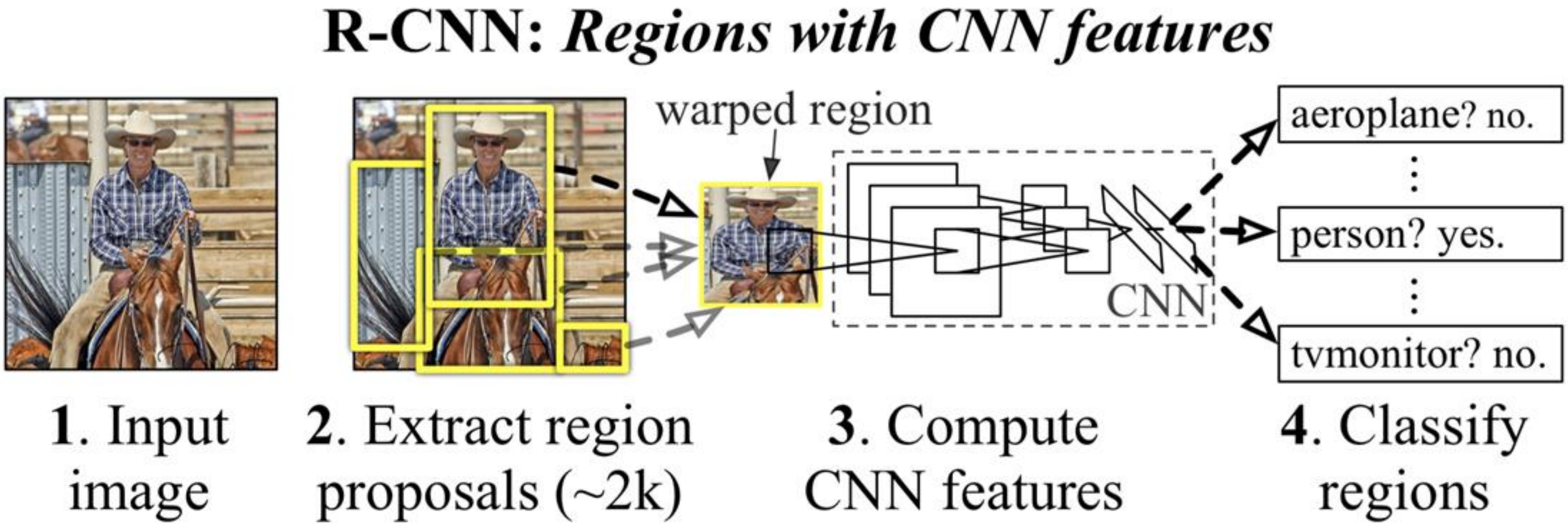
**2-Stage Detector** - Regional Proposal와 Classification이 순차적으로 이루어짐.



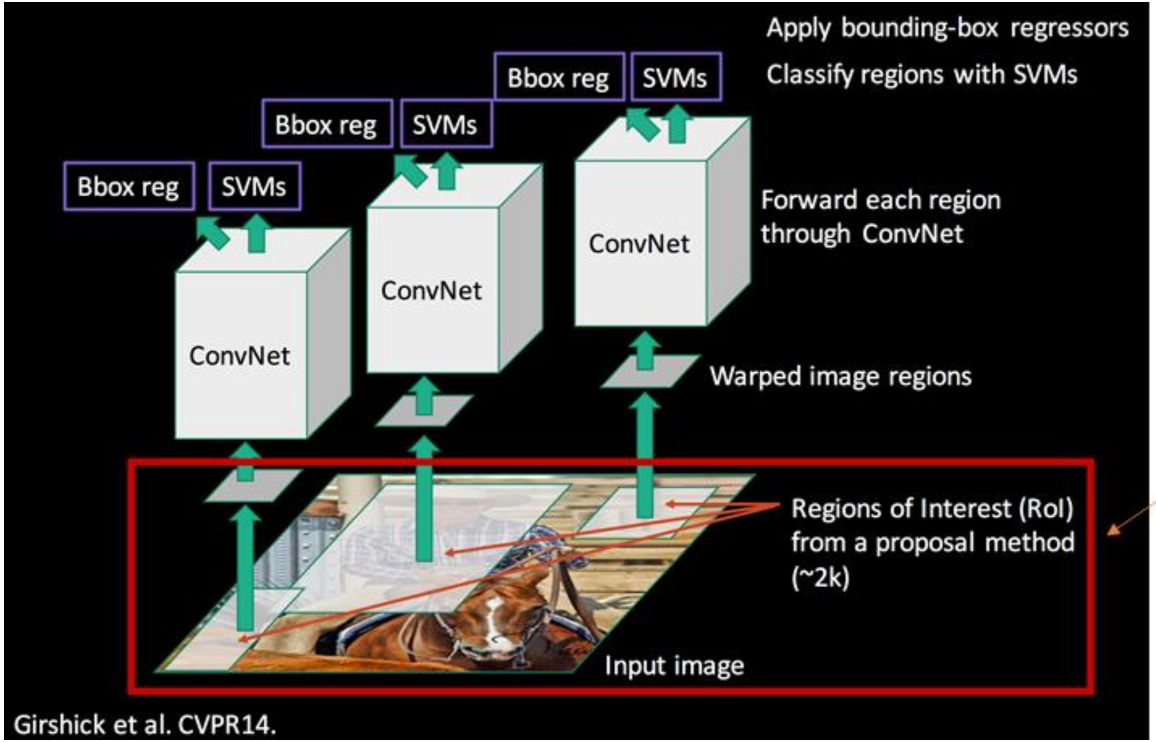

- Selective Search
- Region Proposal Network
- etc.

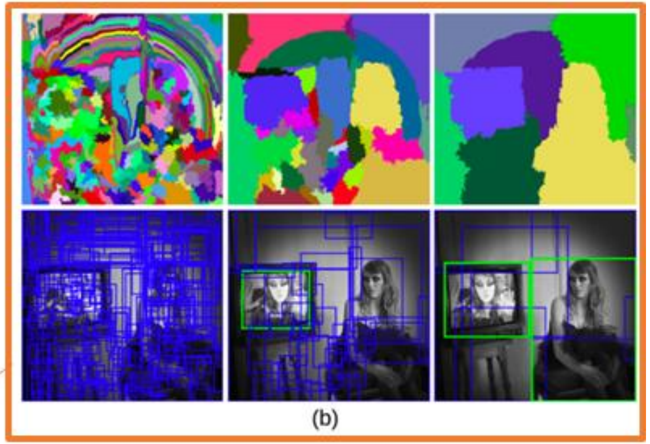
Ex) **R-CNN 계열** (R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, Mask R-CNN ... )

# 3. R-CNN

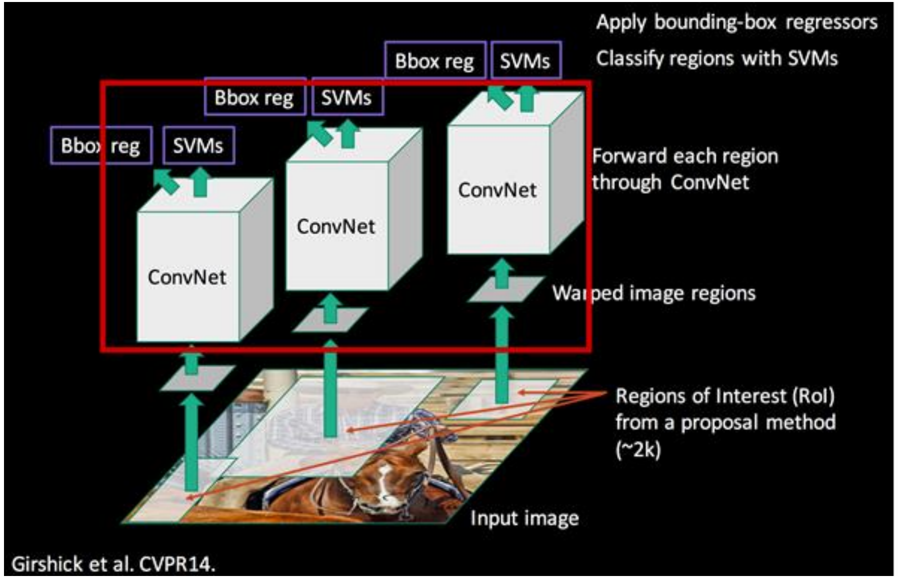

R-CNN: *Regions with CNN features*
warped region
aeroplane? no.
person? yes.
tvmonitor? no.
CNN
1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

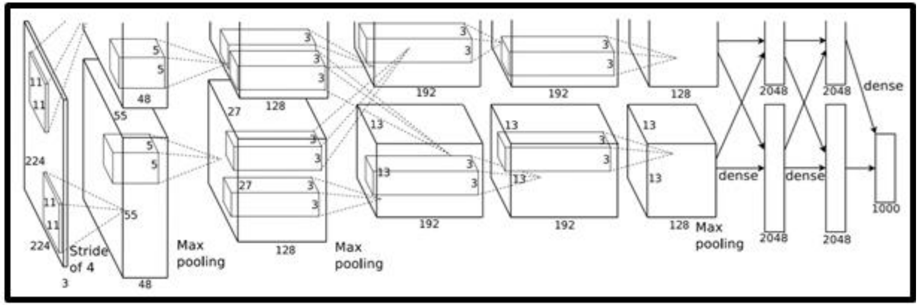
# 3-1. Region Proposal (영역 찾기)



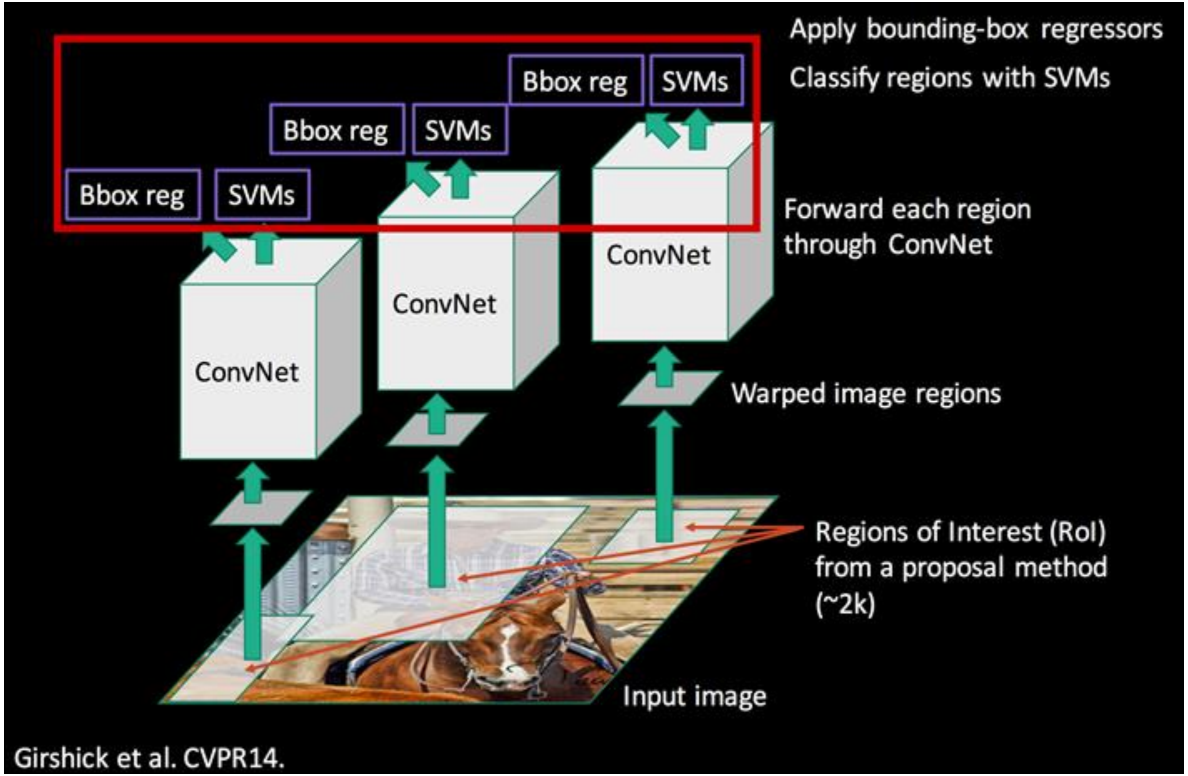



# 3-2. CNN (Convolutional Neural Network)

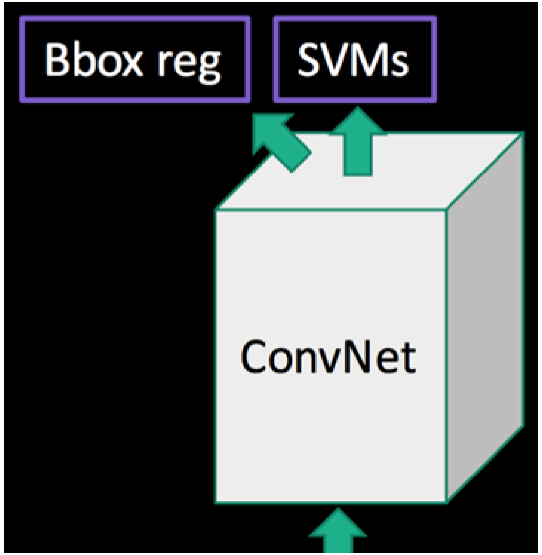


# 3. SVM (Support Vector Machine)



# 4. Bounding Box Regression




$$\{(P^i, G^i)\}_{i=1,\ldots,N}, \text{ where } P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$$

$$\hat{G}_x = P_w d_x(P) + P_x \quad (1) \qquad t_x = (G_x - P_x)/P_w \quad (6)$$

$$\hat{G}_y = P_h d_y(P) + P_y \quad (2) \qquad t_y = (G_y - P_y)/P_h \quad (7)$$

$$\hat{G}_w = P_w \exp(d_w(P)) \quad (3) \qquad t_w = \log(G_w/P_w) \quad (8)$$

$$\hat{G}_h = P_h \exp(d_h(P)). \quad (4) \qquad t_h = \log(G_h/P_h). \quad (9)$$

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}^{\mathrm{T}}_\star \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2. \quad (5)$$

$$d_\star(P) = \mathbf{w}^{\mathrm{T}}_\star \phi_5(P)$$

# 5. R-CNN 단점

1. 오래 걸린다
2. 복잡하다
3. Back Propagation이 안된다

Fast R-CNN, Faster R-CNN

But

1. 최초로 Object Detection에 CNN을 적용시켰다
2. 2-Stage detector들의 구조에 영향을 끼쳤다.

# Index

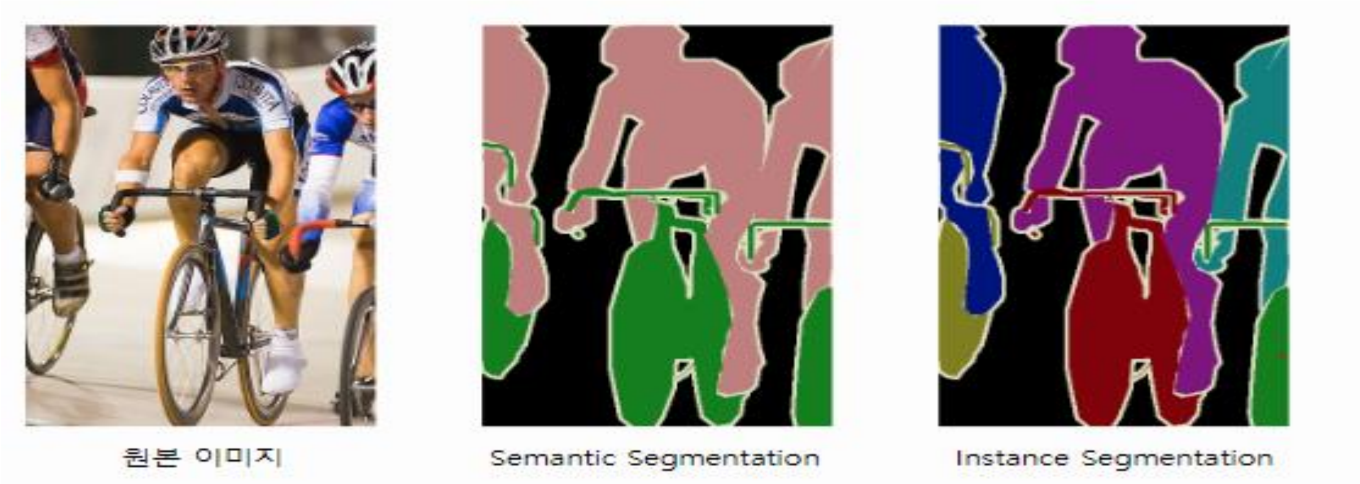
1. Convolutional Neural Network (CNN)

2. Detection/Segmentation
   - Object Detection
   - Image Segmentation

3. Generative Model
   - VAE
   - GAN

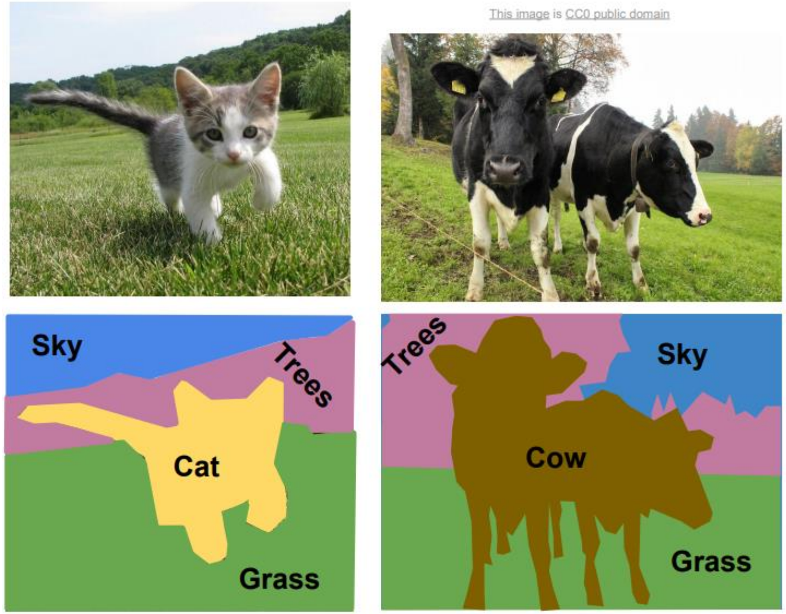
# 1. Image Segmentation

<u>Semantic Segmentation</u> <u>Instance Segmentation</u>



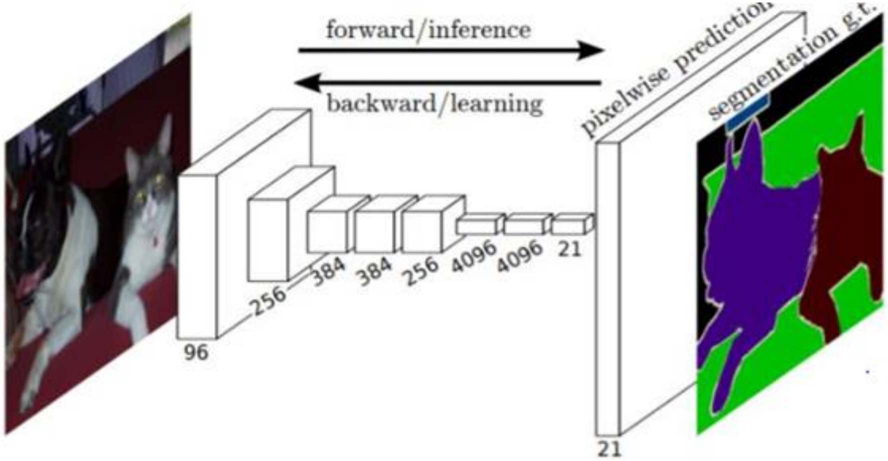
# 2. Semantic Segmentation

- Label each pixel in the image with a category label
- Don't differentiate instances, only care about pixels



# 2-1. Fully Convolutional Network (FCN)

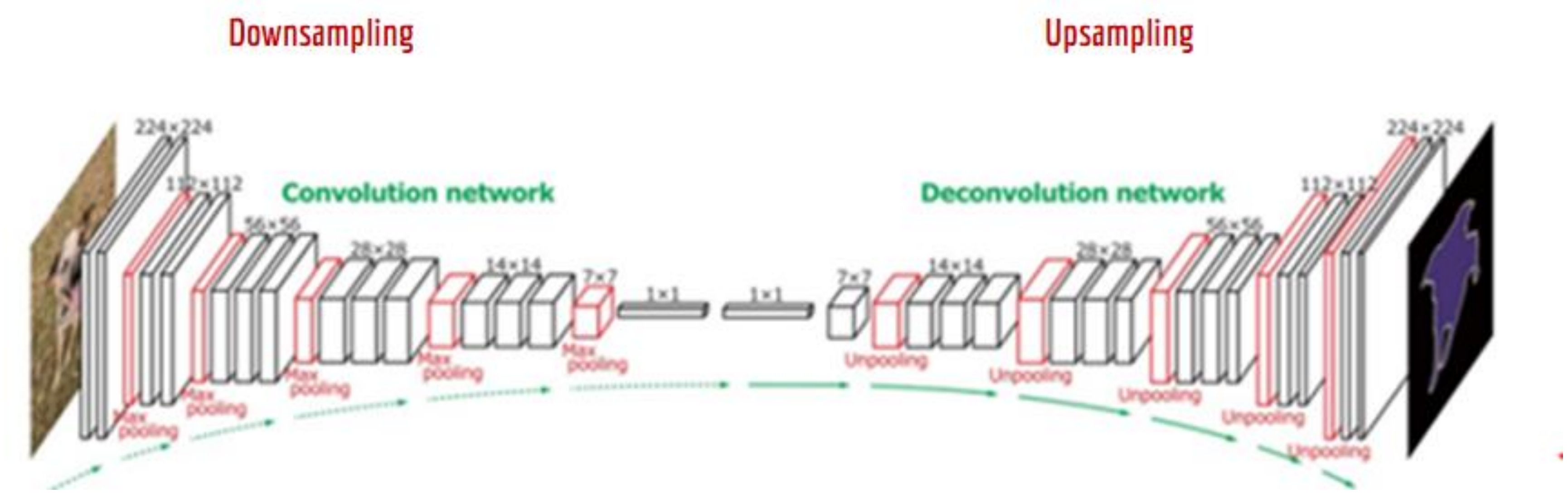
- Bunch of convolutional layers, with downsamplig and upsampling inside the network




- Convolution
- pixelwise prediction
- Upsampling

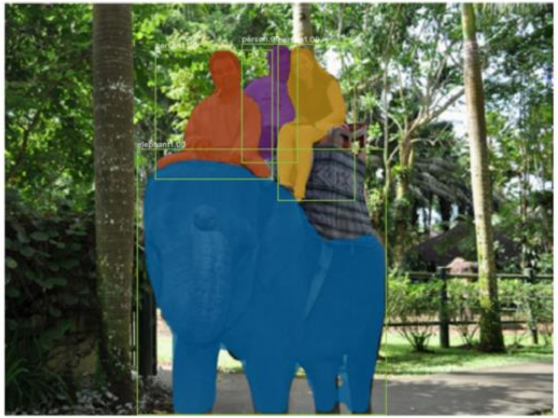
# 2-2. Deconvolutional Network

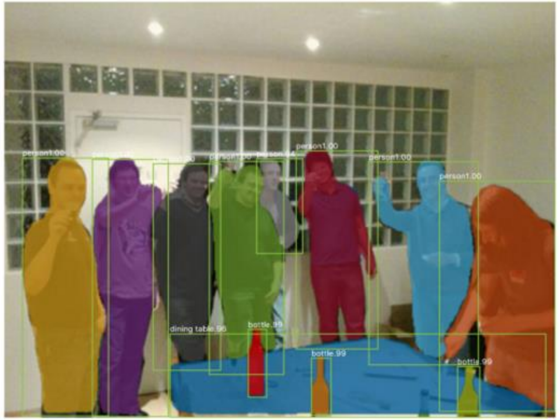
- Bunch of convolutional layers, with downsamplig and upsampling inside the network



# 3. Instance Segmentation

- Combination of everything explained

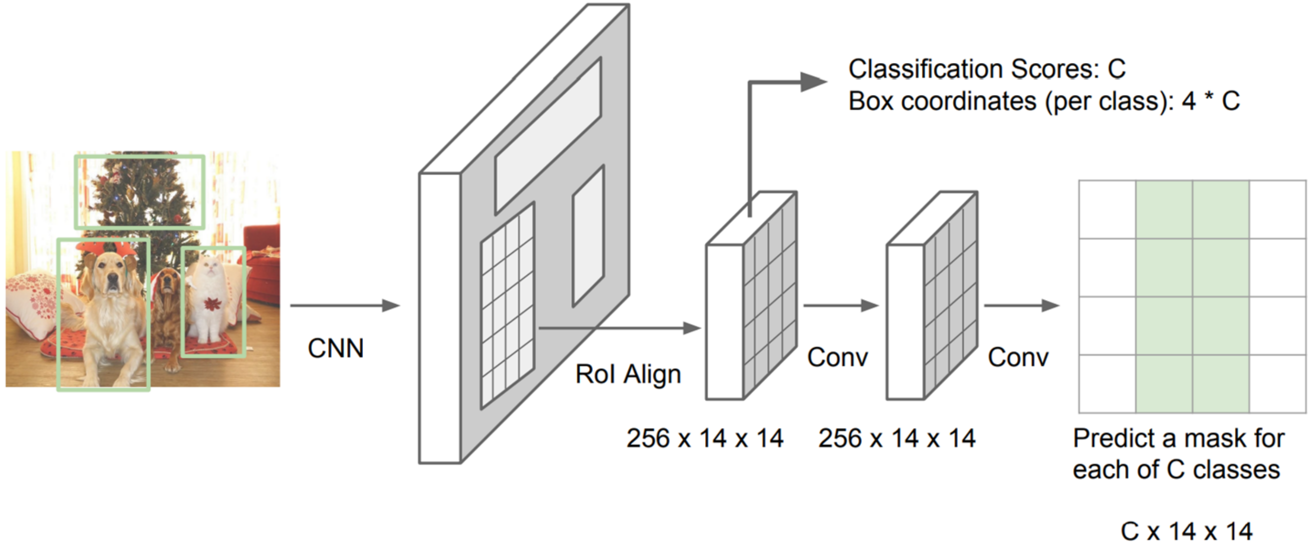




1. Locate & Identify different objects
2. Predict segmentation masks

# 3-1. Mask R-CNN

- Bounding box + Classify the box + Classify each pixel for predicting masks



# Index

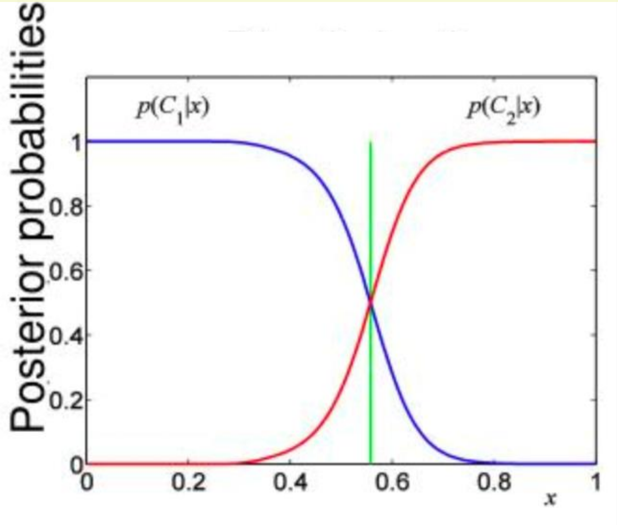
1. Convolutional Neural Network (CNN)

2. Detection/Segmentation
    - Fast R-CNN
    - Mask R-CNN

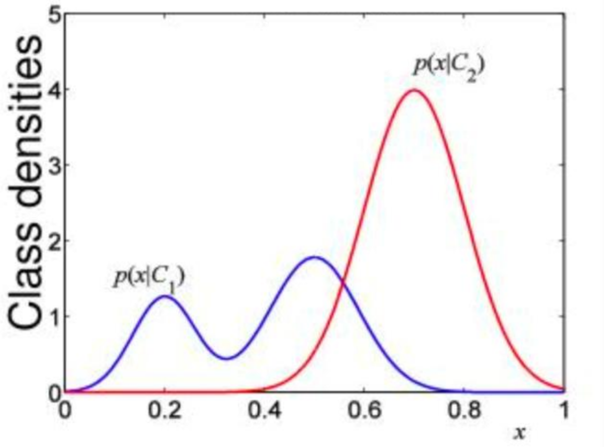
3. Generative Model
    - VAE
    - GAN

# 1. Discriminative Model vs. Generative Model

| Discriminative model | Generative model |
| --- | --- |



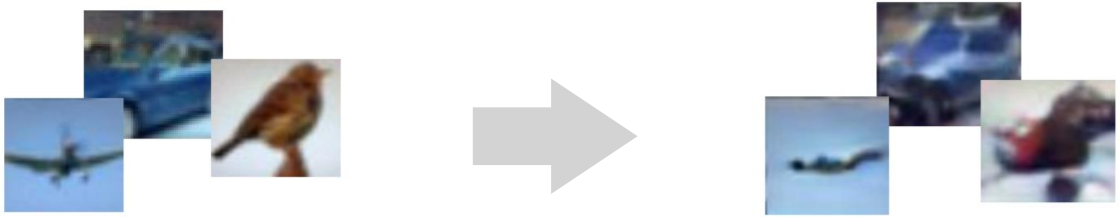

Learns **conditional probability distribution** P(y|x)
Label을 구분할 수 있는 모델




Learns **joint probability distribution** P(x,y)
각 Label의 분포를 제대로 이해하고 있으며,
새로운 label을 만들 수 있는 모델

# 2. Generative Models

- Given training data, generate new samples from same distribution

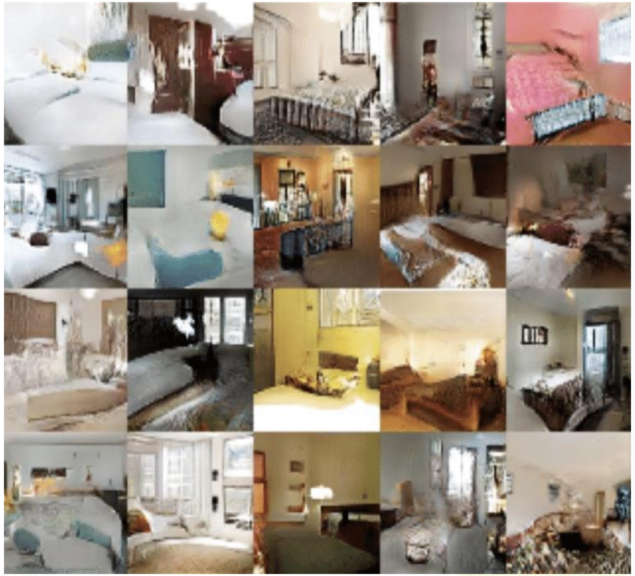


Training data ~ $p_{data}(x)$ Generated samples ~ $p_{model}(x)$

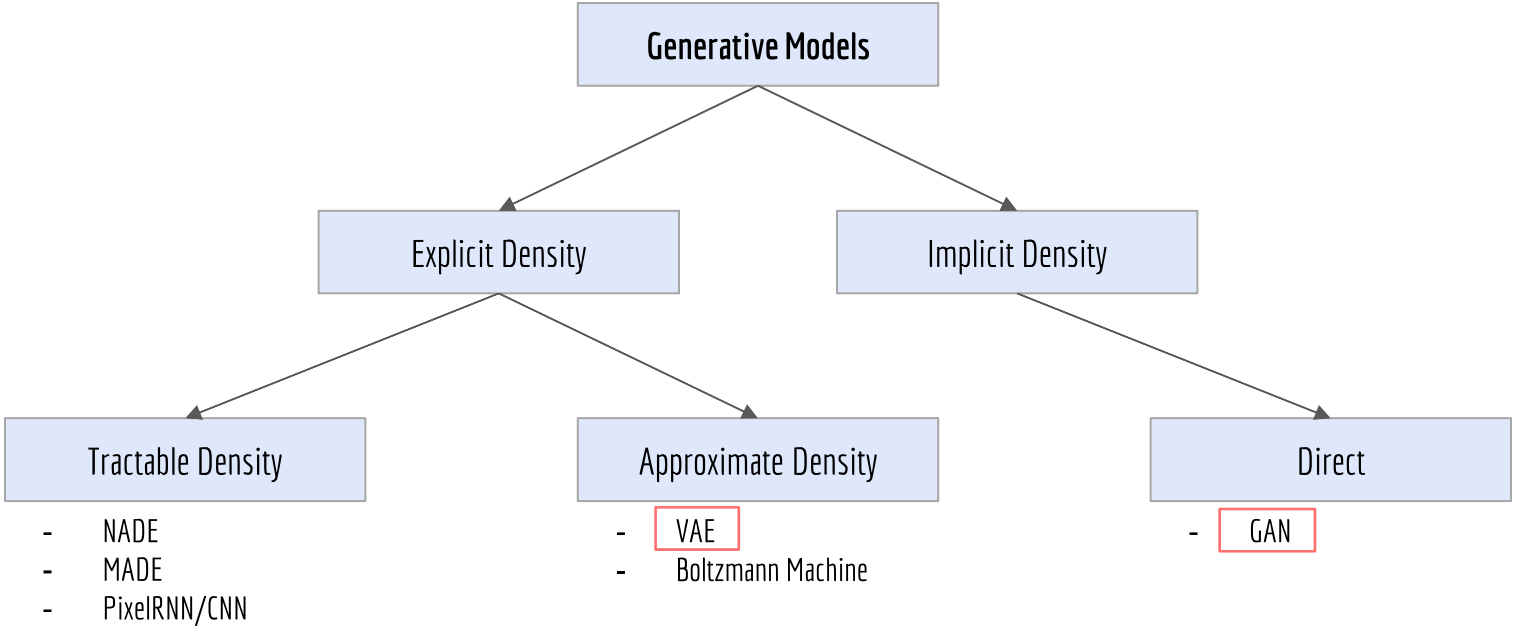
**Want to learn $p_{model}(x)$ similar to $p_{data}(x)$**

# 2. Generative Models

- Given training data, generate new samples from same distribution

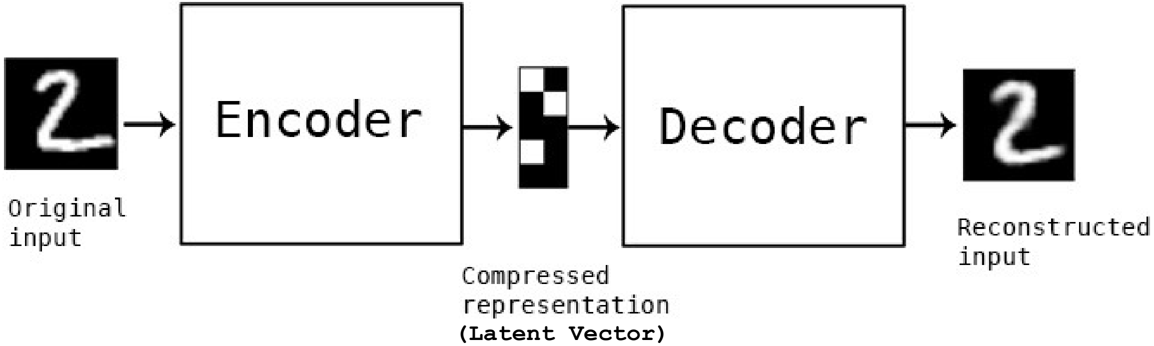


# 3. Taxotomy of Generative Models



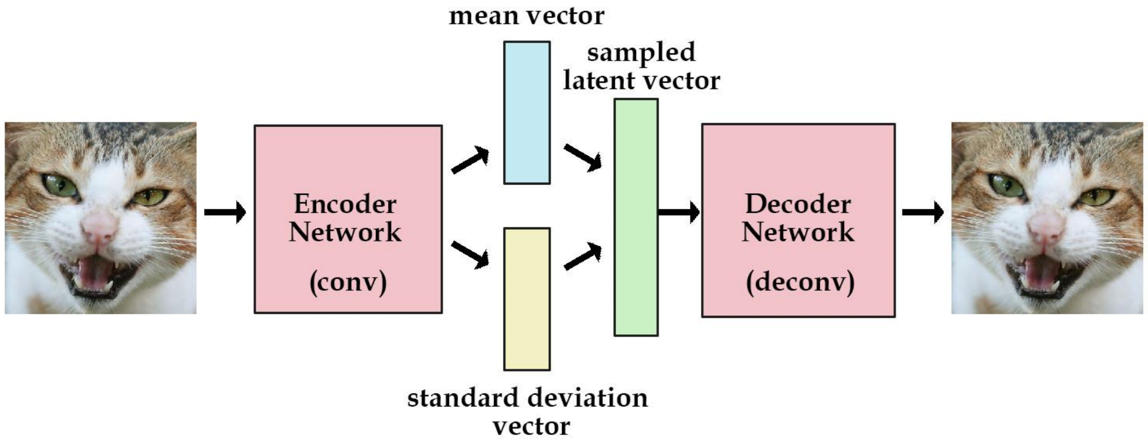
# 4. Variational Auto-Encoder (VAE)

- What is Auto-Encoder?



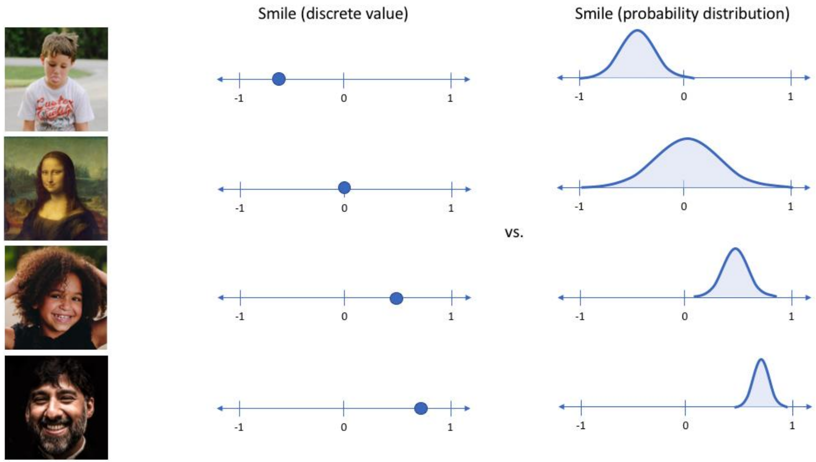

목표: dimension reduction

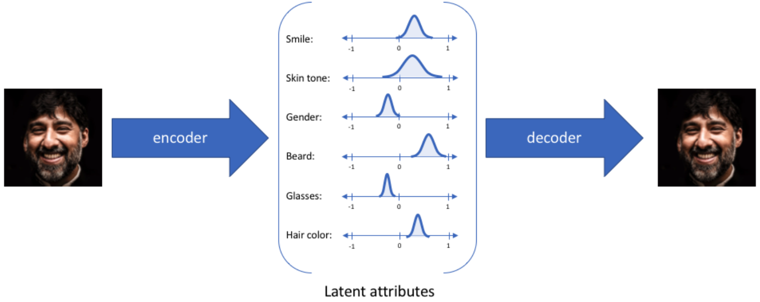
# 4. Variational Auto-Encoder (VAE)




- Formulate encoder to describe a **probability distribution for each attribute** from the input image
- Two latent vectors : 1. mean, 2. standard deviation

# 4. Variational Auto-Encoder (VAE)

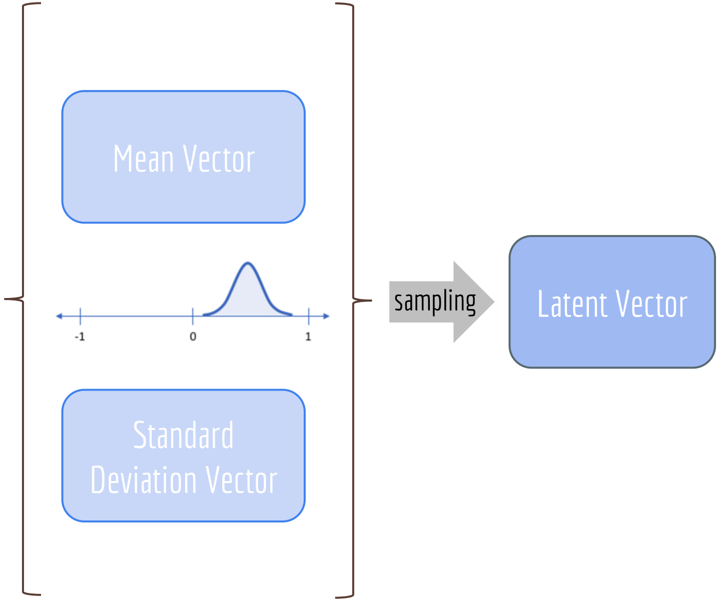


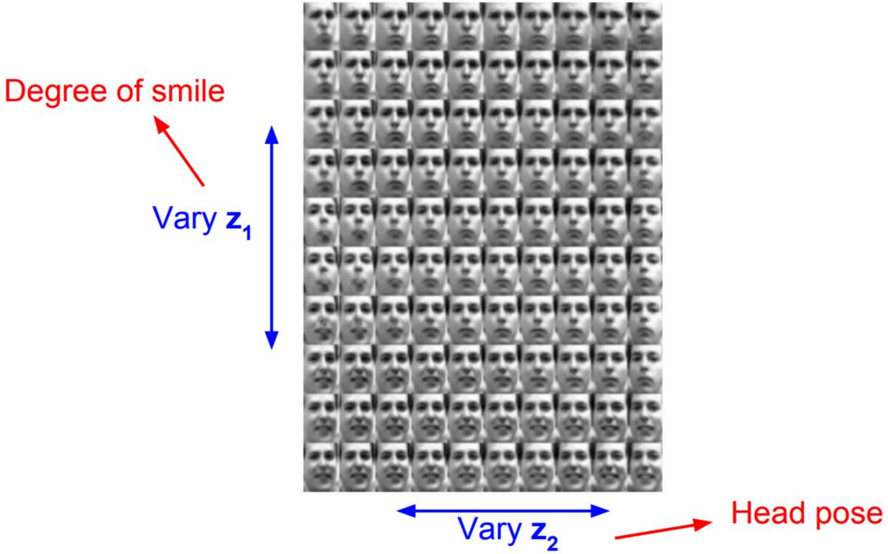

AE : Each attribute in the latent vector has a single value

VAE: Each attribute in the latent vector has a probability distribution

# 4. Variational Auto-Encoder (VAE)




result

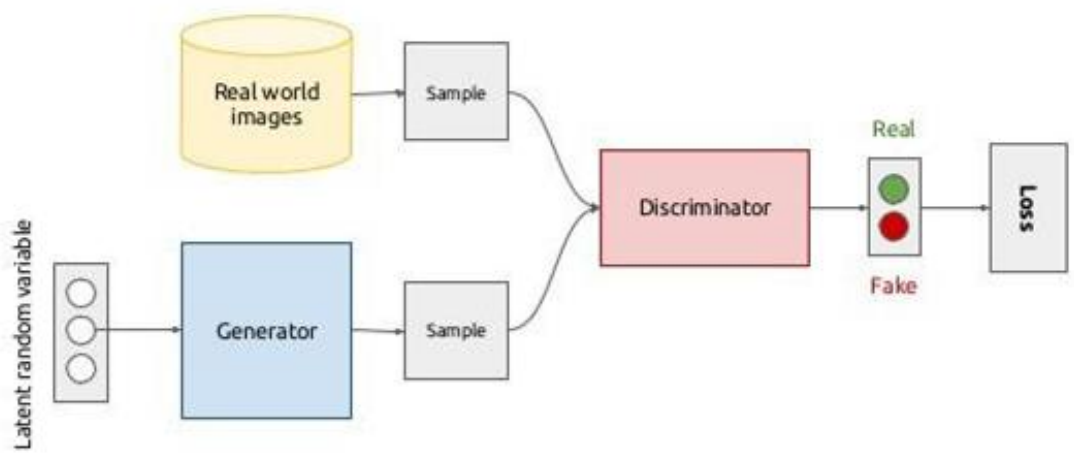


# 5. Generative Adversarial Network (GAN)

**Generator Network**
try to fool the discriminator by generating real-looking images

VS

**Discriminator Network**
try to distinguish between real and fake images

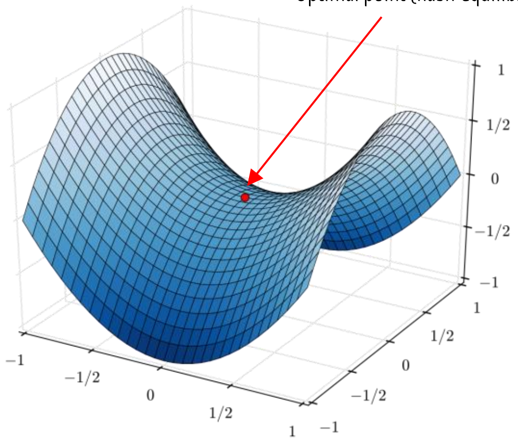


# 5. Generative Adversarial Network (GAN)

## Minimax Objective function

$$\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Optimal point (nash equilibrium)

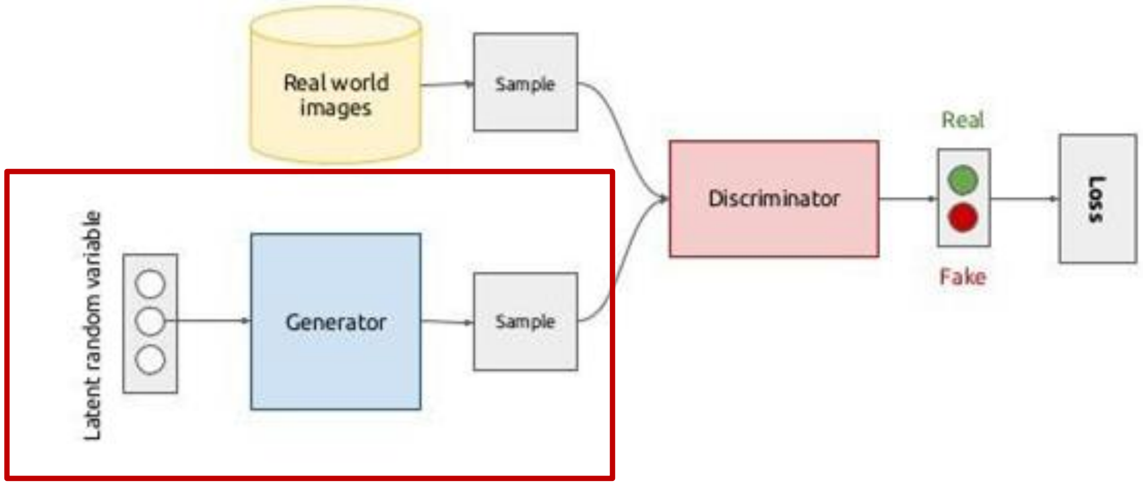



- **Discriminator** wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
  Gradient ascent on discriminator

- **Generator** wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)
  Gradient descent on generator

# 5. Generative Adversarial Network (GAN)

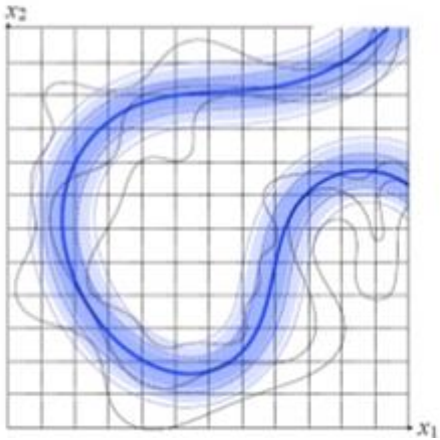
- Eventually, generator ends up generating high-quality images to fool discriminator

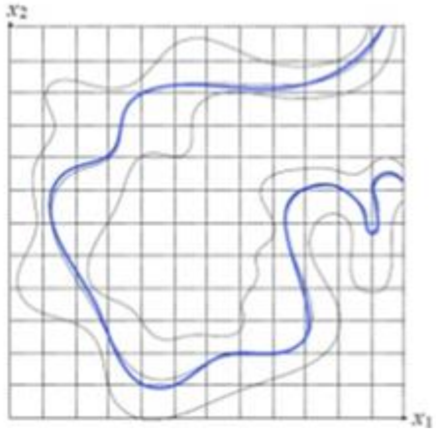



After training, use generator network to generate new images

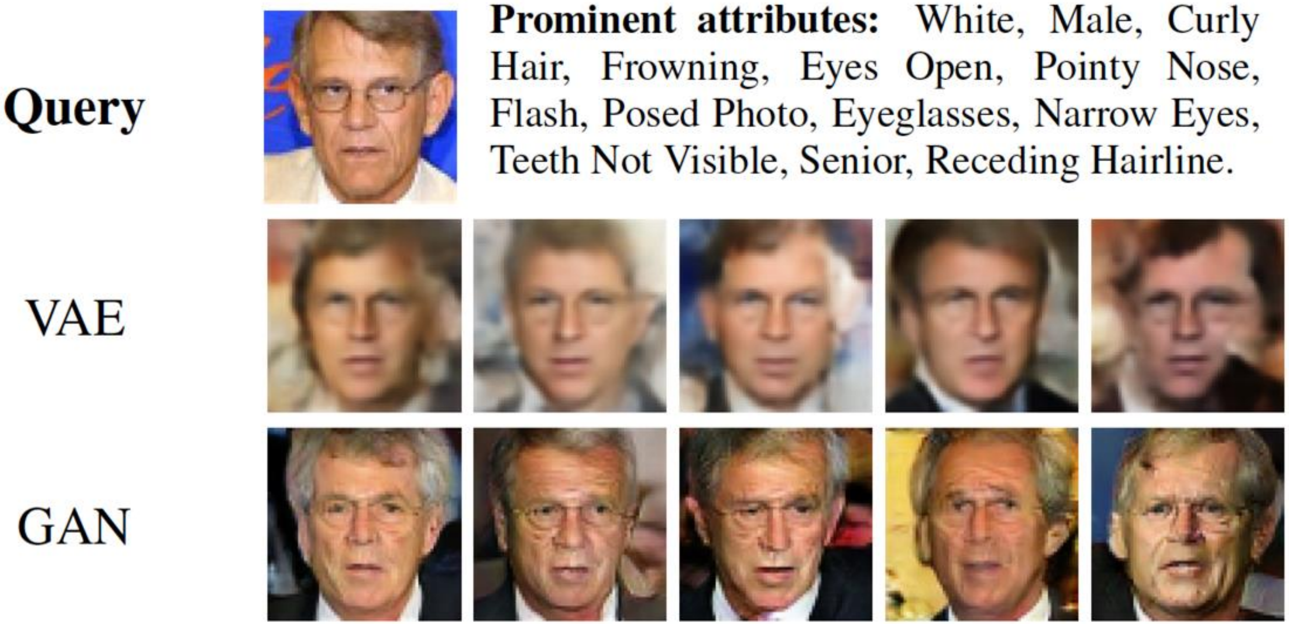
# 6. VAE vs. GAN

| VAE | GAN |
| --- | --- |
| • generates similar data from data distribution<br>• Stable to train, but outputs blurry images | • generates new sample from data distribution<br>• Can learn complex distribution<br>• Outputs sharp images, but unstable to train |



# 6. VAE vs. GAN




https://lh4.googleusercontent.com/aadzYmkduW9bf64AxBxWMB9r0rvR-1yc0BBxAYuu20cvUNa1PV77pWmpf07cpAQ48owGWLCy2nxb5BG7lfwmDHdc_2hoFWpYmqDGi6alhYfKVwn95WnJHylfgXPj2tFrDF8Dt4oewks

# 7. Types of GANs

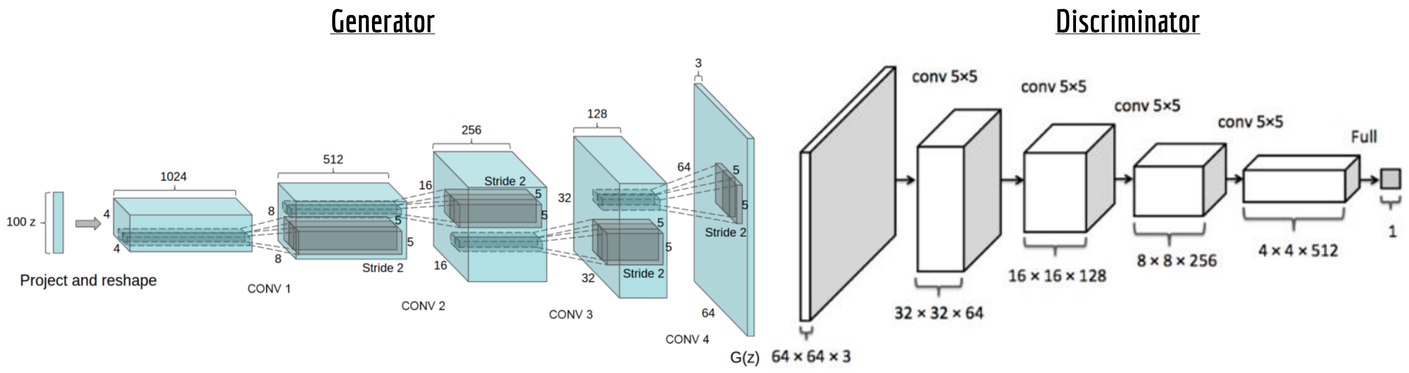
- "The GAN Zoo"

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

https://github.com/hindupuravinash/the-gan-zoo

# 7-1. Deep Convolutional GAN (DCGAN)

- Generator and Discriminator composed of convolution layers without max pooling or fully connected layers



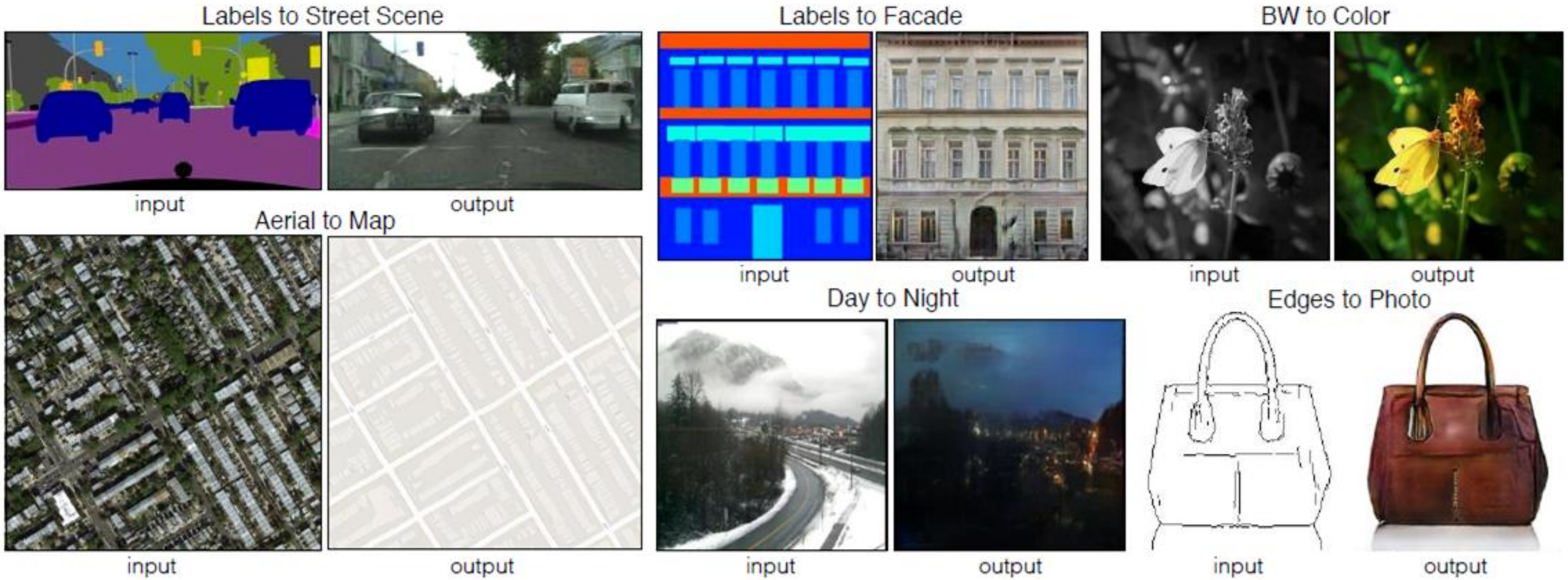
# 7-1. Deep Convolutional GAN (DCGAN)

- Samples created from the model (2016)


(Vector Arithmetic)

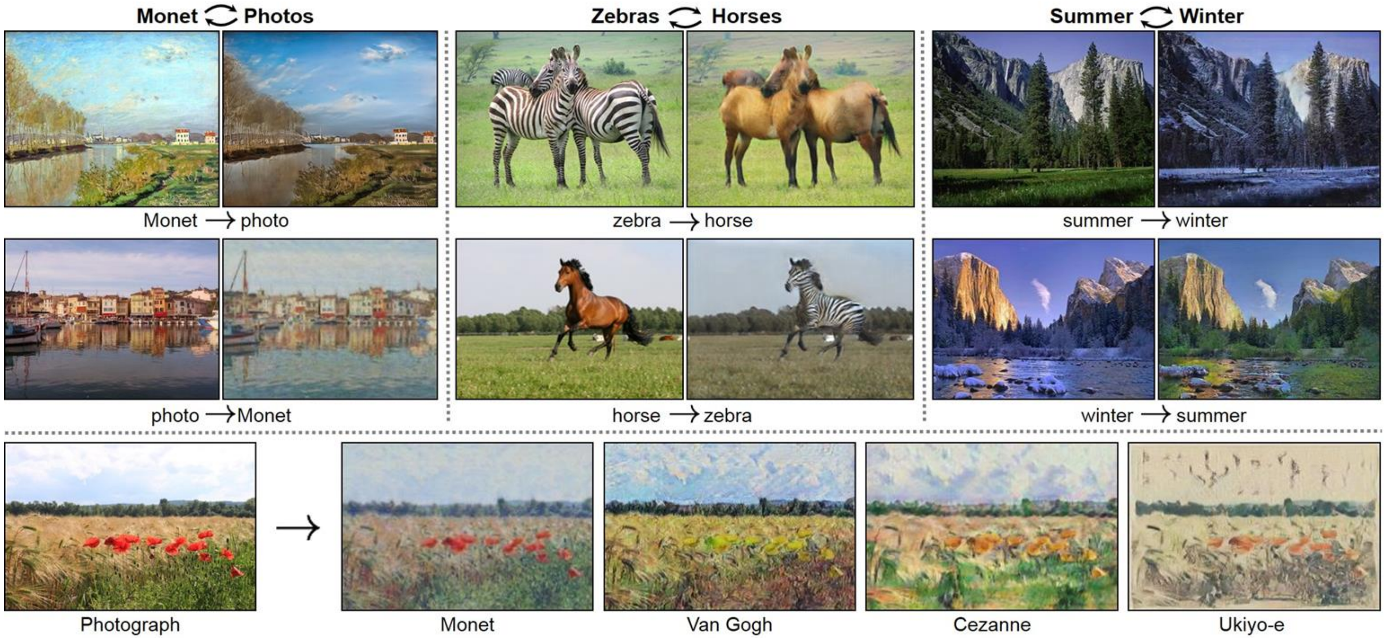
# 7-2. Conditional GAN (cGAN)

- Discriminator and Generator conditioned on some auxiliary information



# 7-3. CycleGAN

- Trains an image-to-image translation without paired examples (style transfer)




# Edmond De Belamy

The shadows of the demons of complexity awaken by family are haunting me.

Everything was so simple back then.

**EXHIBITION :** CHRISITE'S NEW YORK

**OWNER :** ANONYMOUS

**AUCTIONNED PRICE :** 432 000$

# THANK YOU