Title

# GAN Compression: Efficient Architectures for Interactive Conditional GANs

## CVPR 2020

석사과정 김 진용

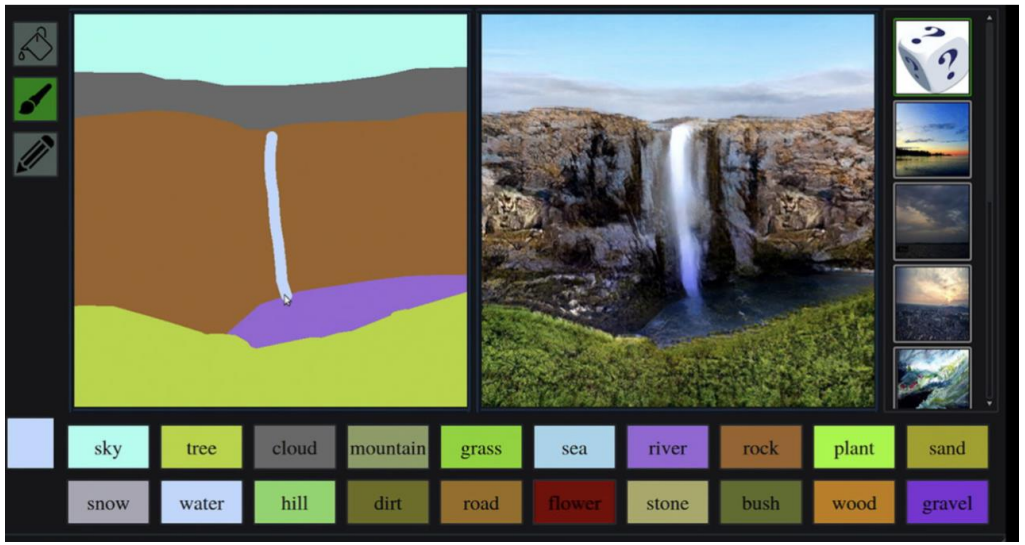
AI Lab. ARTIFICIAL INTELLIGENCE LAB.

# index

# Introduction

석사과정 김 진용

VR Facial Animation via Multiview Image Translation
- SIGGRAPH 2019




GauGAN : Semantic Image Synthesis with Spatially-Adaptive Normalization

Conditional GAN을 기반으로 많은 어플리케이션/논문들이 연구되고 있음

## Oculus Quest

| | |
|---|---|
| 프로세서 | 퀄컴 Snapdragon 835 |
| 디스플레이 | 2*1440*1600 72Hz OLED 펜타일 |
| 외부 카메라 | 4개 |
| 배터리 | 3.85v 3,648mAh 14.0Wh 리튬 이온 배터리 |
| 메모리 | 64GB, 128GB |
| 가변 IPD | O |

하지만, edge단의 모바일 디바이스에서 사용하기에는 resource constraint이 있음

Table 1. MobileNet Body Architecture

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | $3 \times 3 \times 3 \times 32$ | $224 \times 224 \times 3$ |
| Conv dw / s1 | $3 \times 3 \times 32$ dw | $112 \times 112 \times 32$ |
| Conv / s1 | $1 \times 1 \times 32 \times 64$ | $112 \times 112 \times 32$ |
| Conv dw / s2 | $3 \times 3 \times 64$ dw | $112 \times 112 \times 64$ |
| Conv / s1 | $1 \times 1 \times 64 \times 128$ | $56 \times 56 \times 64$ |
| Conv dw / s1 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 128$ | $56 \times 56 \times 128$ |
| Conv dw / s2 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 256$ | $28 \times 28 \times 128$ |
| Conv dw / s1 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 256$ | $28 \times 28 \times 256$ |
| Conv dw / s2 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 512$ | $14 \times 14 \times 256$ |
| $5\times$ Conv dw / s1 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| $5\times$ Conv / s1 | $1 \times 1 \times 512 \times 512$ | $14 \times 14 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| Conv / s1 | $1 \times 1 \times 512 \times 1024$ | $7 \times 7 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 1024$ dw | $7 \times 7 \times 1024$ |
| Conv / s1 | $1 \times 1 \times 1024 \times 1024$ | $7 \times 7 \times 1024$ |
| Avg Pool / s1 | Pool $7 \times 7$ | $7 \times 7 \times 1024$ |
| FC / s1 | $1024 \times 1000$ | $1 \times 1 \times 1024$ |
| Softmax / s1 | Classifier | $1 \times 1 \times 1000$ |

Table 2. Resource Per Layer Type

| Type | Mult-Adds | Parameters |
|---|---|---|
| Conv $1 \times 1$ | 94.86% | 74.59% |
| Conv DW $3 \times 3$ | 3.06% | 1.06% |
| Conv $3 \times 3$ | 1.19% | 0.02% |
| Fully Connected | 0.18% | 24.33% |

앞서 연구되었던 Efficient CNN들의 경우와 비교했을 때, CycleGAN은 Mobile Net의 100배 GauGAN은 Mobile Net의 500배의 차이가 난다는 것을 알 수 있음

*We use the number of Multiply-Accumulate Operations (MAC) to quantify the computation cost. Modern computer architectures use fused multiplyadd (FMA) instructions for tensor operations. These instructions compute $a = a + b \times c$ as one operation. 1 MAC=2 FLOPs.

또한, 본 논문에서는 MAC(Multiply-Accumulate Operation)이라는 새 지표를 제시함.

Conditional GAN에서 Number of Parameter와 MAC를 낮추기 위해 2가지를 제안

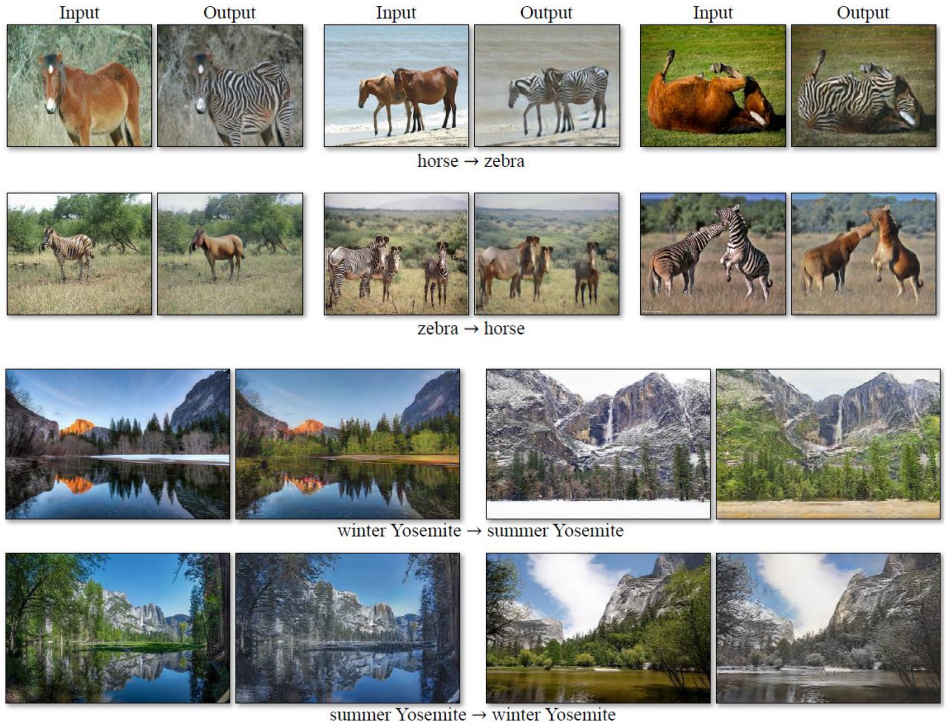
1. Knowledge Distillation(Knowledge transfer,지식 증류/전이)

2. NAS를 통한 아키텍쳐 서치

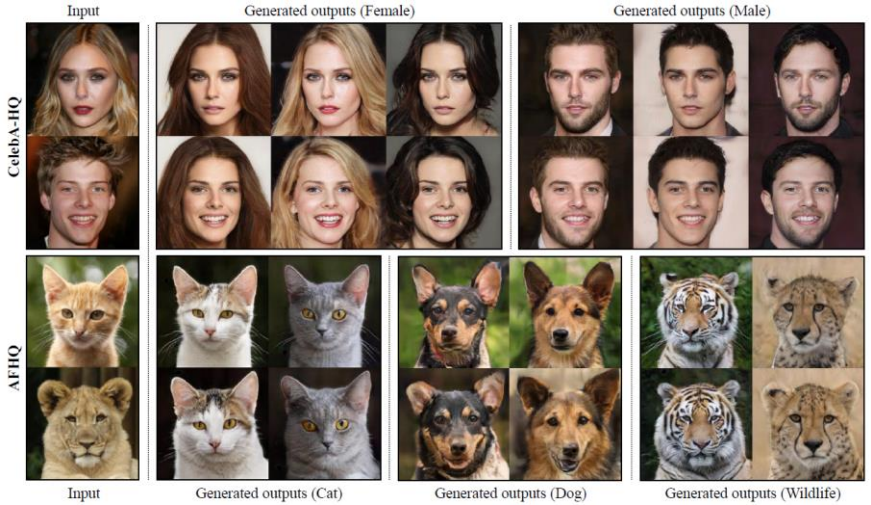
# Related Works

석사과정 김 진용

# Conditional GAN




CycleGAN, ICCV2017




StarGAN v2, CVPR2020


U-GAT-IT, ICLR 2020

Conditional GAN으로 다양한 연구들이 진행되고 있으며, 성과가 좋은 결과들이 많이 나타나고 있음.

# Conditional GAN

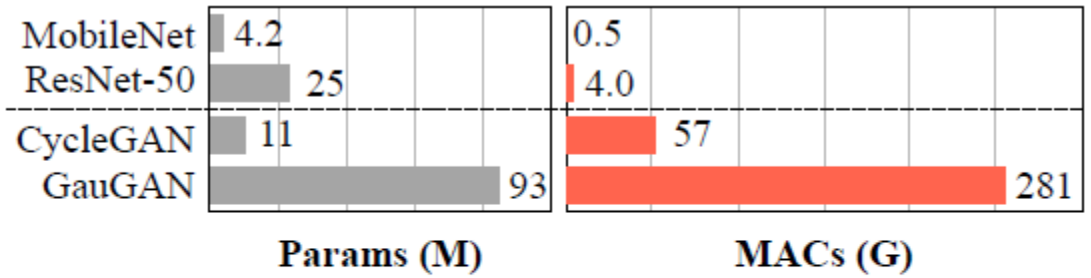
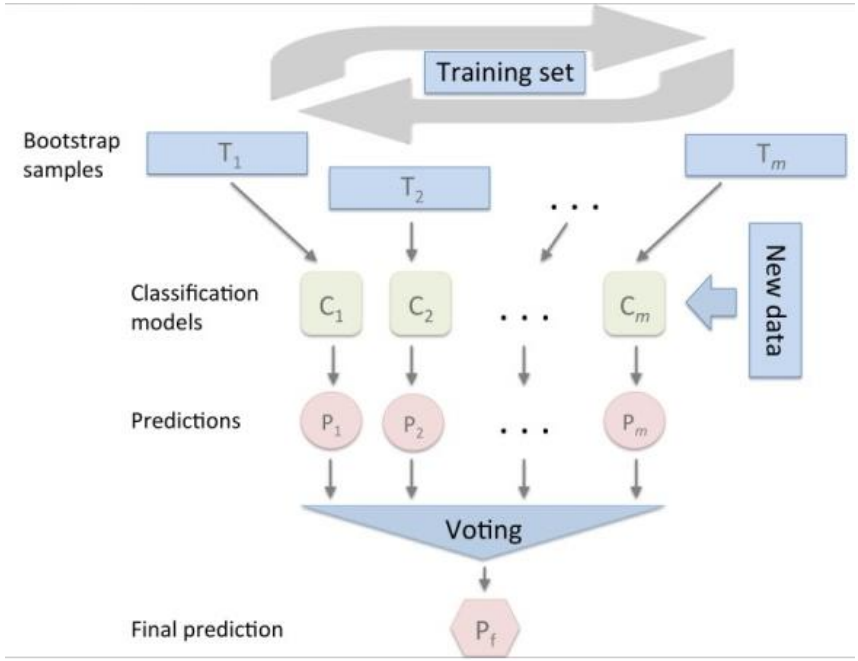
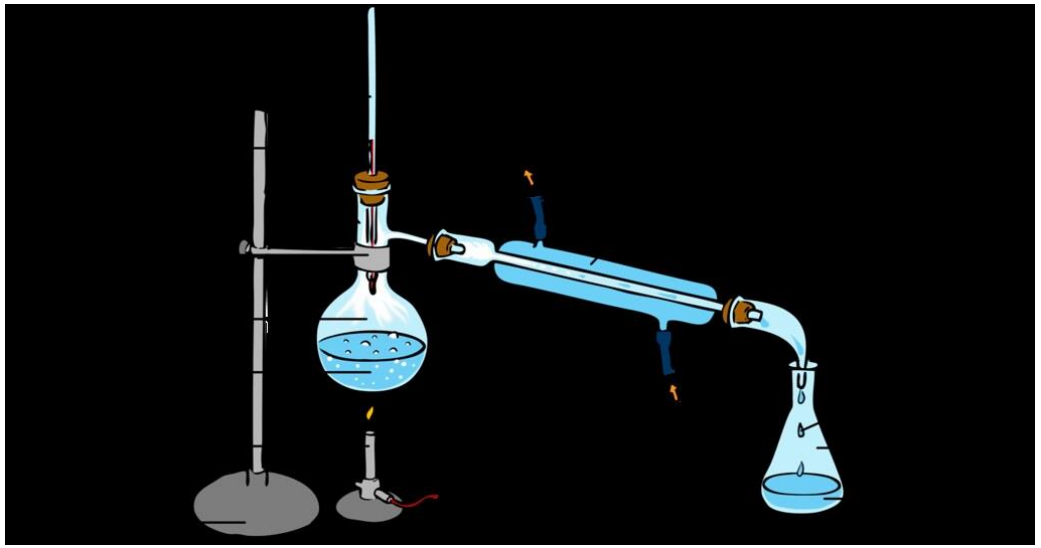



Figure 2: Conditional GANs require two orders of magnitude more computation than image classification CNNs, making it prohibitive to be deployed on edge devices.

하지만, Conditional GAN에서 나타나는 Parameter의 수와 MAC에서
일반적인 Image Classification 모델보다 훨씬 많은 resource가 필요하다는 것을 알 수 있다.

# Knowledge Distillation



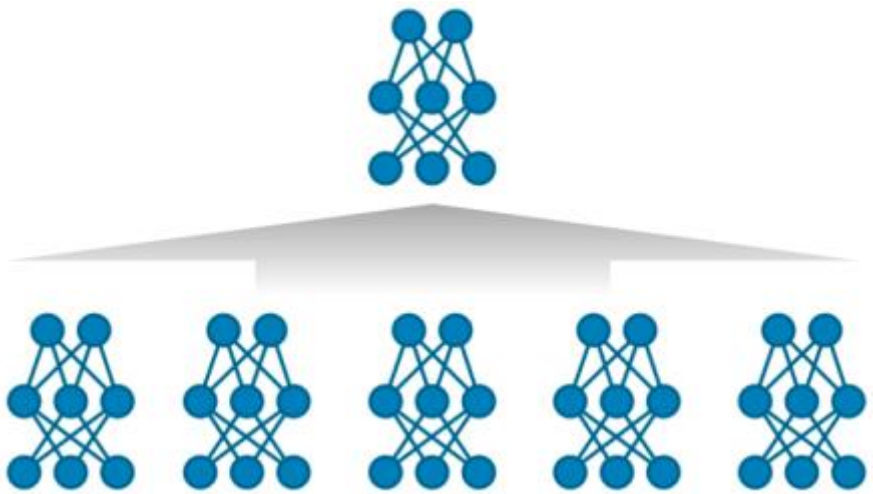



- Knowledge Distillation(지식 증류)는 Ensemble 모델 아이디어에서 착안하여 시작되었음
- Ensemble model은 주어진 자료로부터 여러 개의 예측모형(e.g. DL,ML)을 만들어 조합하여 하나의 최종 예측모형을 만드는 것을 말함



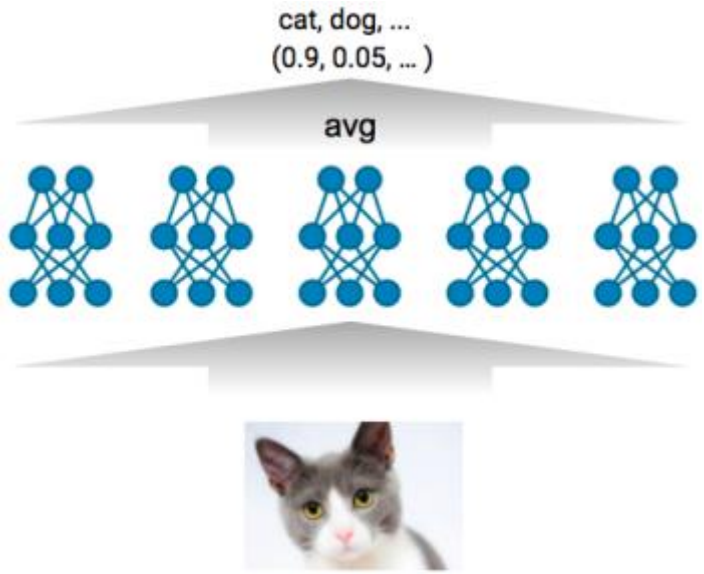
*Distilling the Knowledge in a Neural Network (NIPS 2014 Workshop)*

# Knowledge Distillation



- 즉, 모델의 Ensemble을 통하여 얻을 수 있는 부분을 Single Neural Network에 옮길 수 있을까라는 의문에서 시작함

# Knowledge Distillation




- Image Classification을 예로, Output은 여러 카테고리에 대한 합이 1이 되는 Softmax output을 주로 사용하는데, 이 부분이 모델이 지니고있는 Knowledge의 함축적인 부분이라고 가정



*Distilling the Knowledge in a Neural Network (NIPS 2014 Workshop)*

# Knowledge Distillation

dog

| cow | dog | cat | car | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | original hard targets |
| cow | dog | cat | car | |
| $10^{-6}$ | .9 | .1 | $10^{-9}$ | output of geometric ensemble |

Comparison with the 'hard label' and the 'soft label'

- Input data가 가지고 있는 hard label의 경우
  - 가지고 있는 Knowledge는 "이 사진은 강아지다" 라는 의미뿐
- Softmax output의 soft label의 경우
  - 가지고 있는 Knowledge는 "90%의 확률로 강아지, 10%의 확률로 고양이..."라는 의미
  - 이를 "Dark Knowledge"라고 칭함

# Knowledge Distillation

dog

| cow | dog | cat | car | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | original hard targets |
| cow | dog | cat | car | |
| $10^{-6}$ | .9 | .1 | $10^{-9}$ | output of geometric ensemble |

Comparison with the 'hard label' and the 'soft label'

- 이렇게 해서 Distillation 을 적용하게 되면 여러 모델의 성능을 한대로 합쳐서 Generalization을 잘 할 수 있는 성능좋은 모델이 나온다.

# Neural Architecture Search

- NAS는 두 개의 구성으로 이루어져있음
    1. RNN Controller(좌측)
    2. RNN에서 출력된 Architecture로 accuracy를 측정하고 그를 기반으로 Controller를 학습시키는 강화학습 모델

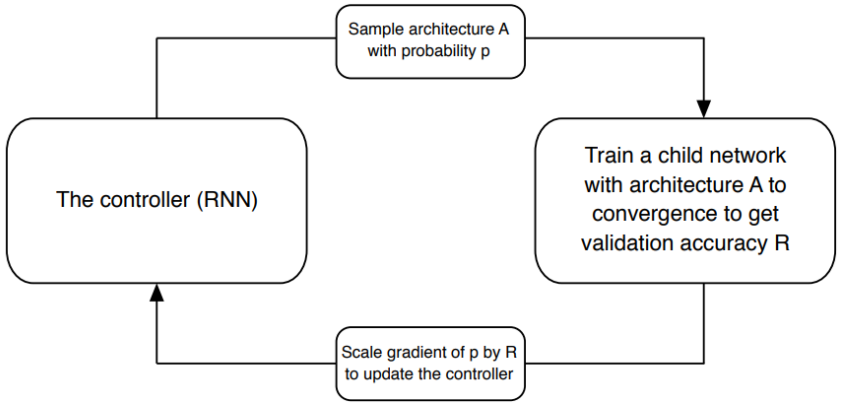



Figure 1. Overview of Neural Architecture Search [71]. A controller RNN predicts architecture $A$ from a search space with probability $p$. A child network with architecture $A$ is trained to convergence achieving accuracy $R$. Scale the gradients of $p$ by $R$ to update the RNN controller.

# Proposed Method

석사과정 김 진용

## Method

본 논문의 GAN Compression 모델은 다음 두 가지 이유 때문에 어렵다.

1. GAN의 트레이닝은 당연히 불안정하다. -> 이를 해결할 수 있는 Training Protocol을 제안하여 문제해결

2. CNN의 Compression 방식을 GAN에 적용하려는 부분 -> NAS(Neural Architecture Search)를 이용하여 해결

## 3.1. Training Objective

Unifying unpaired and paired learning

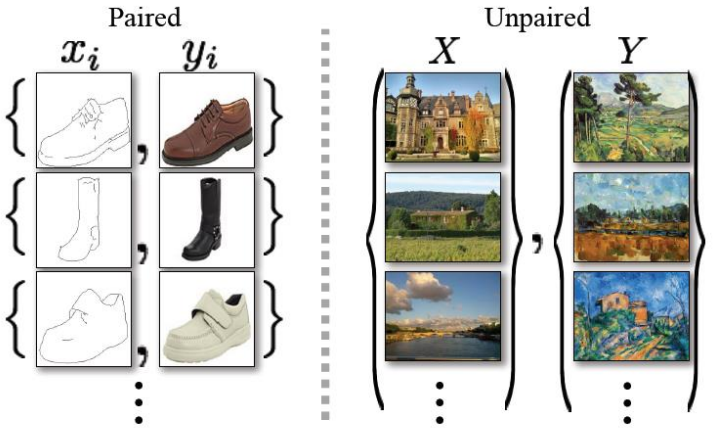



Figure 2: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^N$, where the correspondence between $x_i$ and $y_i$ exists [22]. We instead consider *unpaired* training data (right), consisting of a source set $\{x_i\}_{i=1}^N$ $(x_i \in X)$ and a target set $\{y_j\}_{j=1}$ $(y_j \in Y)$, with no information provided as to which $x_i$ matches which $y_j$.

Unpaired data와 Paired data로 나뉜 cGAN 에서 이것을 다 포괄해서 framework를 하나 만드는 것은 어렵다.

때문에, paired 와 unpaired를 그냥 통합해버렸다.

## 3.1. Training Objective

Unifying unpaired and paired learning

$$\mathcal{L}_{\text{recon}} = \begin{cases} \mathbb{E}_{\mathbf{x},\mathbf{y}} \|G(\mathbf{x}) - \mathbf{y}\|_1 & \text{if paired cGANs,} \\ \mathbb{E}_{\mathbf{x}} \|G(\mathbf{x}) - G'(\mathbf{x})\|_1 & \text{if unpaired cGANs.} \end{cases}$$

결국은 paired냐 unpaired냐는 loss에 대한 문제(여기서는 Reconstruction loss)이고,
이를 해결하기위해 위와 같은 조건을 삽입함.

이로인해, teacher generator가 뭐냐에 상관없이 student generator는 distillation이 가능함.

## 3.1. Training Objective

Inheriting the teacher discriminator

$$\mathcal{L}_{\text{cGAN}} = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x}}[\log(1 - D(\mathbf{x}, G(\mathbf{x})))]$$

Discriminator는 같은 구조의 네트워크로 사용하고, pre-trained의 네트워크를 꾸준히 사용한다

- discriminator에는 현재 상대하고 있는 Generator의 약점을 아는 것과 같은 좋은 Knowledge들이 많다.
- fine-tune 하기위해 협력

## 3.1. Training Objective

Intermediate feature distillation

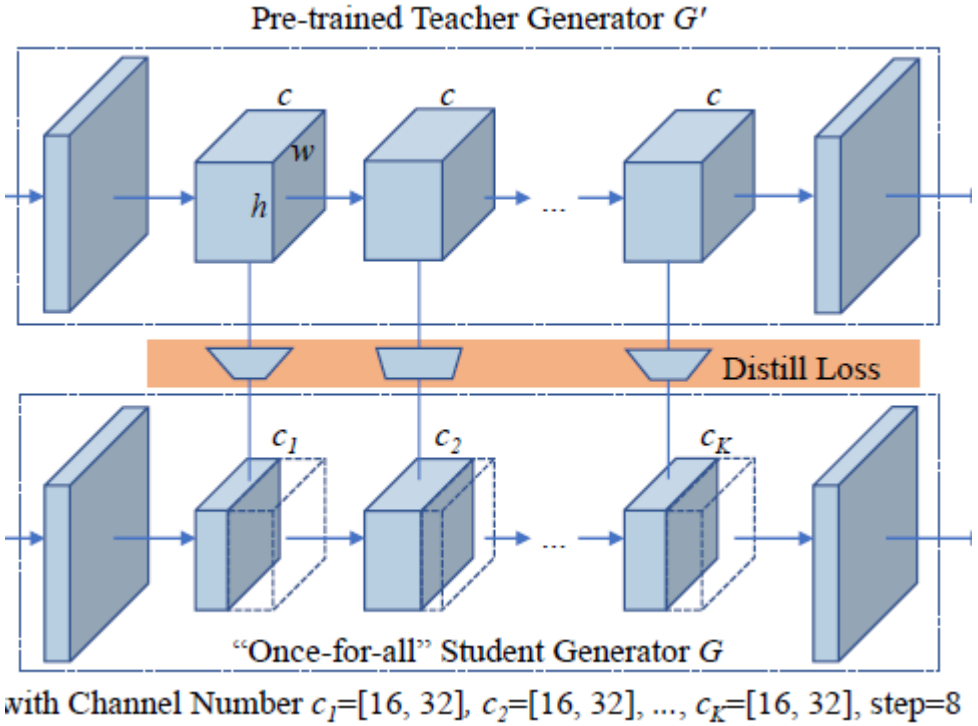
dog

| cow | dog | cat | car | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | original hard targets |
| cow | dog | cat | car | |
| $10^{-6}$ | .9 | .1 | $10^{-9}$ | output of geometric ensemble |

- Dark knowledge는 Teacher model에서 Student model로 전이 될 때, 많은 지식을 함유하고 있어 performance를 향상시킴
- 그러나 Conditional GAN은 확률적인 분포가 아닌, 결정적인(Deterministic) 이미지를 출력함.
- 그러므로 teacher model의 output에서 dark knowledge를 추출하기 어려움

## 3.1. Training Objective

Intermediate feature distillation




$$\mathcal{L}_{\text{distill}} = \sum_{t=1}^{T} \|G_t(\mathbf{x}) - f_t(G'_t(\mathbf{x}))\|_2,$$

- 이 문제를 해결하기위해, intermediate representation을 매칭함
- 이미지화 되지 않은 분포들을 사용하면 더 풍부한 정보들을 student model이 학습할 수 있음.
- 공식에서 T는 number of layer

## 3.1. Training Objective

Intermediate feature distillation

$$\mathcal{L} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}},$$

- Full objective 는 위와 같음

## 3.2. Efficient Generator Design Space

Convolution decomposition and layer sensitivity

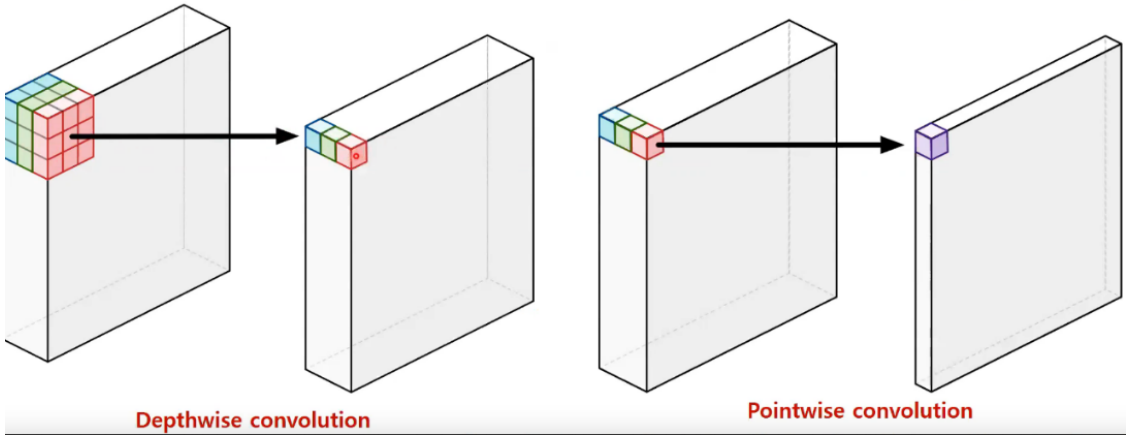



Fig. Depthwise Separable Convolution

- Decomposed version of convolution(위와 같은)을 적용하면 이미지의 퀄리티가 degrade되는 것을 실험에서 발견
  - 레이어의 sensitivity는 기존 recognition model과는 다름
  - 다른 레이어가 상대적으로 robust 해짐
- 따라서 ResBlock에만 적용함

## 3.2. Efficient Generator Design Space

Automated channel reduction with NAS

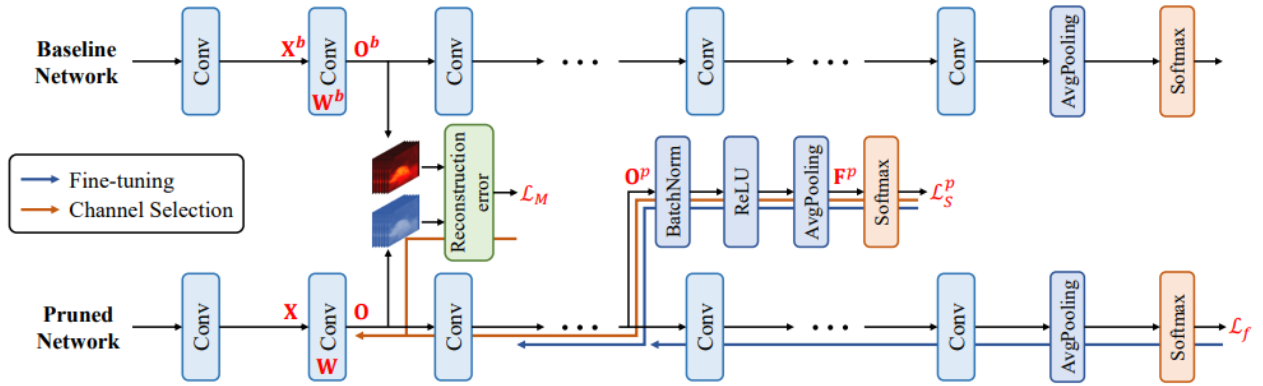



Figure 1: Illustration of discrimination-aware channel pruning. Here, $\mathcal{L}_S^p$ denotes the discrimination-aware loss (e.g., cross-entropy loss) in the $L_p$-th layer, $\mathcal{L}_M$ denotes the reconstruction loss, and $\mathcal{L}_f$ denotes the final loss. For the $p$-th stage, we first fine-tune the pruned model by $\mathcal{L}_S^p$ and $\mathcal{L}_f$, then conduct the channel selection for each layer in $\{L_{p-1}+1, \ldots, L_p\}$ with $\mathcal{L}_S^p$ and $\mathcal{L}_M$.

Fig. Channel Pruning Example
Ref : Discrimination-aware Channel Pruning for Deep Neural Networks

Channel reduction을 위해 기존 연구되었던 channel pruning을 사용한다.

$$\{c_1, c_2, \ldots, c_K\} \longrightarrow \{c_1^*, c_2^*, \ldots, c_K^*\} = \arg\min_{c_1, c_2, \ldots, c_K} \mathcal{L}$$

그러나, $K$ 값이 증가하게되면 가능한(찾을 수 있는 모든)것은 기하급수적으로 증가하게되고 이는 서치하기에 너무 많은 시간이 소요된다.

## 3.2. Efficient Generator Design Space

Automated channel reduction with NAS

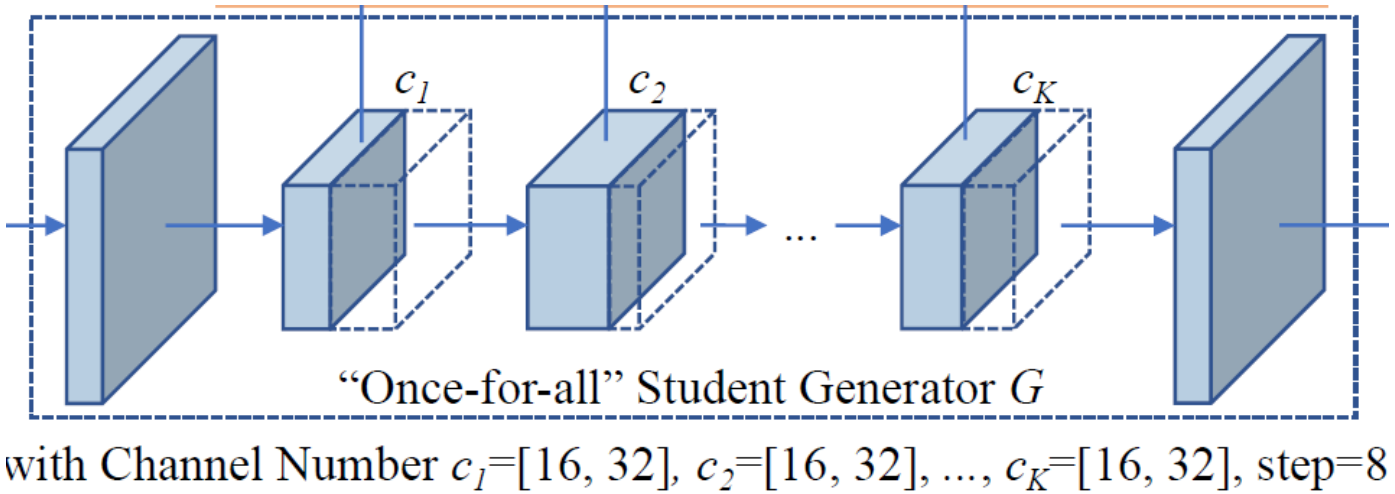



Fig. Channel Pruning Example
Ref : Discrimination-aware Channel Pruning for Deep Neural Networks

그렇기 때문에 매번 전체다 K만큼의 채널을 갖고 시작하는 것이 아님.
최대값 C_K를 두고 그것보다 작은 값에서 Random하게 설정하여 탐색함
어차피 Candidate를 염두하고 만들기 때문에 상관없고 적은값의 k를 가지게되면 그만큼 여러번 update하게 됨

# Summary of proposed model

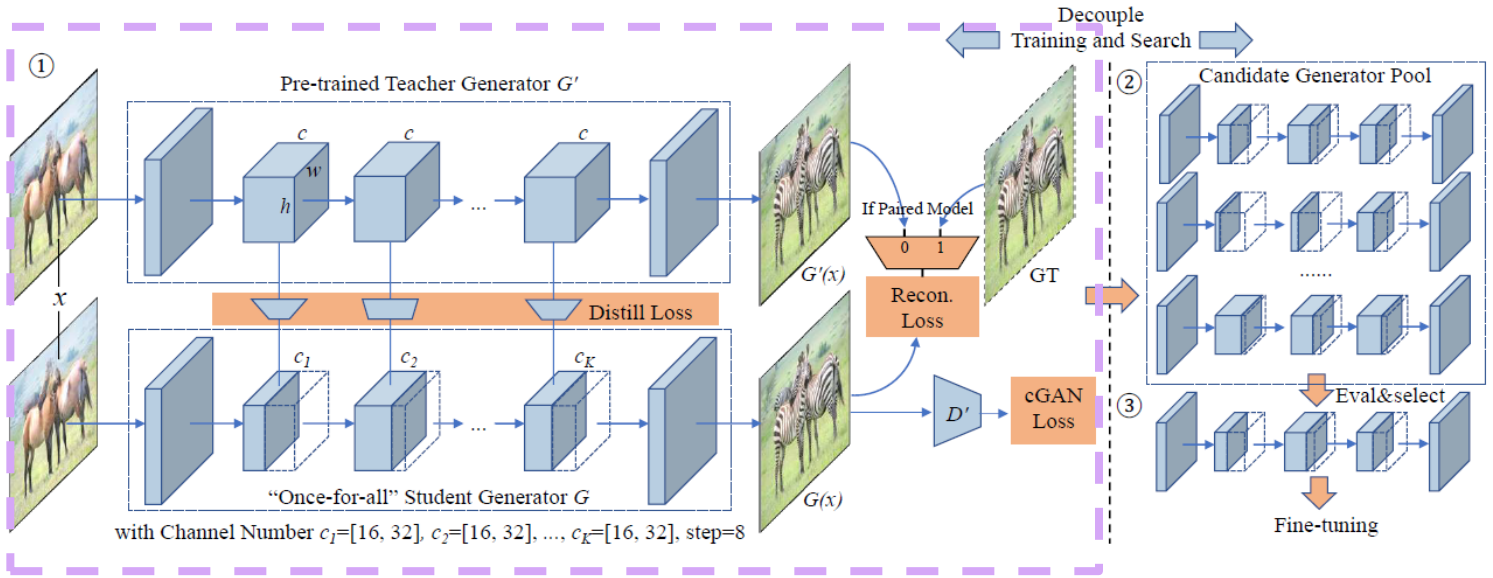



Figure 3: GAN Compression framework: ① Given a pre-trained teacher generator $G'$, we distill a smaller "once-for-all" student generator $G$ that contains all possible channel numbers through weight sharing. We choose different channel numbers $\{c_k\}_{k=1}^{K}$ for the student generator $G$ at each training step. ② We then extract many sub-generators from the "once-for-all" generator and evaluate their performance. No retraining is needed, which is the advantage of the "once-for-all" generator. ③ Finally, we choose the best sub-generator given the compression ratio target and performance target (FID or mAP), perform fine-tuning, and obtain the final compressed model.

① Teacher Generator가 학습을 거쳐 distillation을 진행하게되면 "once-for-all" student generator가 channel 수를 줄인 채로 distillation 받아 학습

$$\mathcal{L} = \mathcal{L}_{\text{cGAN}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}},$$

# Summary of proposed model

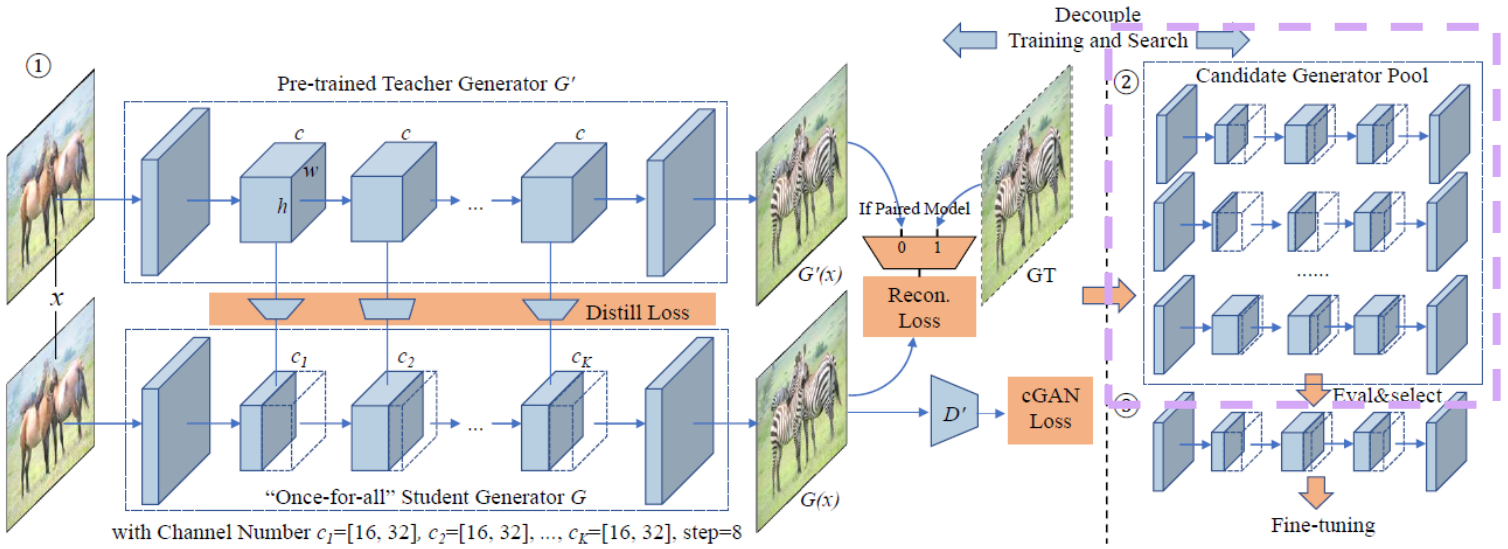



Figure 3: GAN Compression framework: ① Given a pre-trained teacher generator $G'$, we distill a smaller "once-for-all" student generator $G$ that contains all possible channel numbers through weight sharing. We choose different channel numbers $\{c_k\}_{k=1}^{K}$ for the student generator $G$ at each training step. ② We then extract many sub-generators from the "once-for-all" generator and evaluate their performance. No retraining is needed, which is the advantage of the "once-for-all" generator. ③ Finally, we choose the best sub-generator given the compression ratio target and performance target (FID or mAP), perform fine-tuning, and obtain the final compressed model.

② Candidate의 pool에서 성능에 따른 평가로 가장 최고인 것을 하나 선택

# Summary of proposed model

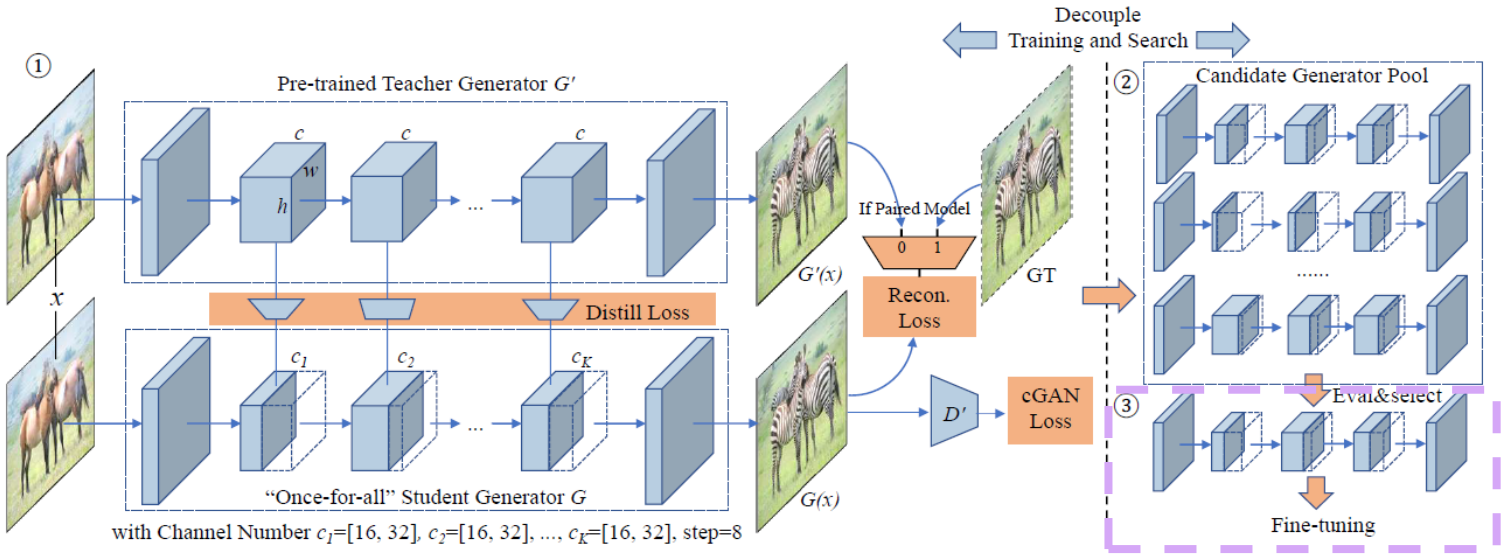



Figure 3: GAN Compression framework: ① Given a pre-trained teacher generator $G'$, we distill a smaller "once-for-all" student generator $G$ that contains all possible channel numbers through weight sharing. We choose different channel numbers $\{c_k\}_{k=1}^{K}$ for the student generator $G$ at each training step. ② We then extract many sub-generators from the "once-for-all" generator and evaluate their performance. No retraining is needed, which is the advantage of the "once-for-all" generator. ③ Finally, we choose the best sub-generator given the compression ratio target and performance target (FID or mAP), perform fine-tuning, and obtain the final compressed model.

③ 완성 + Fine-tuning

# 4. Experiments
Setup

## 실험 모델

- CycleGAN [76], an unpaired image-to-image translation model, uses a ResNet-based generator [19, 28] to transform an image from a source domain to a target domain, without using pairs.
- Pix2pix [27] is a conditional-GAN based paired image-to-image translation model. For this model, we replace the original U-Net generator [52] by the ResNet-based generator [28] as we observe that the ResNet-based generator achieves better results with less computation cost, given the same learning objective. See Appendix 6.2 for a detailed U-Net vs. ResNet benchmark.
- GauGAN [49] is a state-of-the-art paired image-to-image translation model. It can generate a high-fidelity image given a semantic label map.

## 실험 데이터

- Edges→shoes. We use 50,025 images from UT Zappos50K dataset [69]. We split the dataset randomly so that the validation set has 2,048 images for a stable evaluation of Fréchet Inception Distance (FID) (see Section 4.2). We evaluate the pix2pix model on this dataset.
- Cityscapes. The dataset [12] contains the images of German street scenes. The training set and the validation set consists of 2975 and 500 images, respectively. We evaluate both the pix2pix and GauGAN model on this dataset.
- Horse↔zebra. The dataset consists of 1,187 horse images and 1,474 zebra images originally from ImageNet [13] and used in CycleGAN [76]. The validation set contains 120 horse images and 140 zebra images. We evaluate the CycleGAN model on this dataset.
- Map↔aerial photo. The dataset contains 2194 images scraped from Google Maps and used in pix2pix [27]. The training set and the validation set contains 1096 and 1098 images, respectively. We evaluate the pix2pix model on this dataset.

# 4. Experiments

## Quantitative Evaluation

| Model | Dataset | Method | #Parameters | | MACs | | Metric | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | FID (↓) | | mAP (↑) | |
| CycleGAN | horse→zebra | Original | 11.3M | – | 56.8G | – | 61.53 | – | – | |
| | | Shu *et al.* [56] | – | – | 13.4G | (4.2×) | 96.15 | (34.6 ☹) | – | |
| | | Ours (w/o fine-tuning) | 0.34M | (**33.3**×) | 2.67G | (**21.2**×) | 64.95 | (**3.42** ☹) | – | |
| | | Ours | 0.34M | (33.3×) | 2.67G | (21.2×) | 71.81 | (10.3 ☹) | – | |
| Pix2pix | edges→shoes | Original | 11.3M | – | 56.8G | – | 24.18 | – | – | |
| | | Ours (w/o fine-tuning) | 0.70M | (16.3×) | 4.81G | (11.8×) | 31.30 | (7.12 ☹) | – | |
| | | Ours | 0.70M | (**16.3**×) | 4.81G | (**11.8**×) | 26.60 | (**2.42** ☹) | – | |
| | cityscapes | Original | 11.3M | – | 56.8G | – | – | | 35.62 | – |
| | | Ours (w/o fine-tuning) | 0.71M | (16.0×) | 5.66G | (10×) | – | | 29.27 | (6.35 ☹) |
| | | Ours | 0.71M | (**16.0**×) | 5.66G | (**10.0**×) | – | | 34.34 | (**1.28** ☹) |
| | map→arial photo | Original | 11.3M | – | 56.8G | – | 47.76 | – | – | |
| | | Ours (w/o fine-tuning) | 0.75M | (15.1×) | 4.68G | (11.4×) | 71.82 | (24.1 ☹) | – | |
| | | Ours | 0.75M | (**15.1**×) | 4.68G | (**11.4**×) | 48.02 | (**0.26** ☹) | – | |
| GauGAN | cityscapes | Original | 93.0M | – | 281G | – | – | | 58.89 | – |
| | | Ours (w/o fine-tuning) | 20.4M | (4.6×) | 31.7G | (8.8×) | – | | 56.75 | (2.14 ☹) |
| | | Ours | 20.4M | (**4.6**×) | 31.7G | (**8.8**×) | – | | 58.41 | (**0.48** ☹) |

Table 1: Quantitative evaluation of GAN Compression: We use the mAP metric (the higher the better) for the Cityscapes dataset and FID (the lower the better) for other datasets. Our method can compress state-of-the-art conditional GANs by **9-21**× in MACs and **5-33**× in model size, with only minor performance degradation. For CycleGAN compression, our systematic approach outperforms previous CycleGAN-specific Co-Evolution method [56] by a large margin.

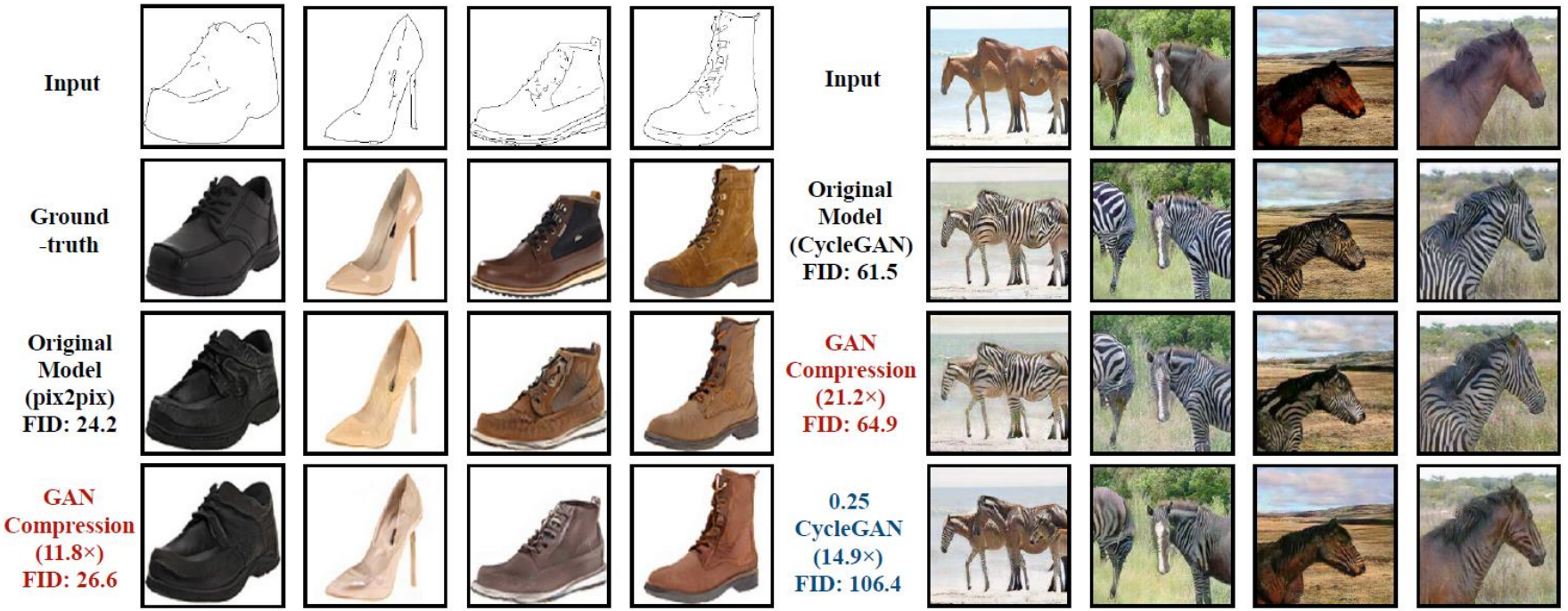
줄어든 파라미터, MAC(Multiply-Accumulation-Computation)를 확인 가능

# 4. Experiments

## Qualitative Evaluation


Segmentation 에 대한 mAP를 통해 정성적 평가 측정

# 4. Experiments
## Qualitative Evaluation




FID(Frechet Inception Distance)를 통한 이미지 퀄리티 정성평가