

---

# 논문리뷰

# DualStyle GAN

카피바라팀 | 배누리, 김호정, 전사영, 박현아 | DATE. 2024.12.24

CONTENTS

# 목차

---

01

Introduction

02

StyleGAN

03

StyleGAN2

04

DualStyleGAN

05

코드 구현

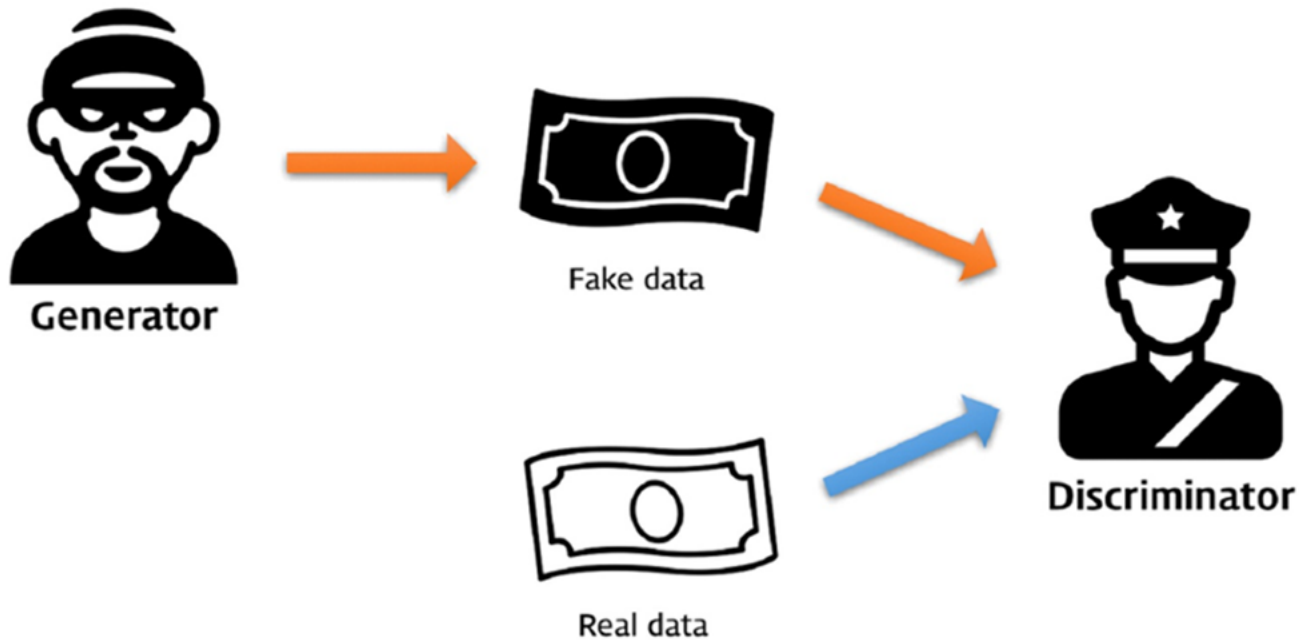
---

CHAPTER 01

# Introduction

## Introduction

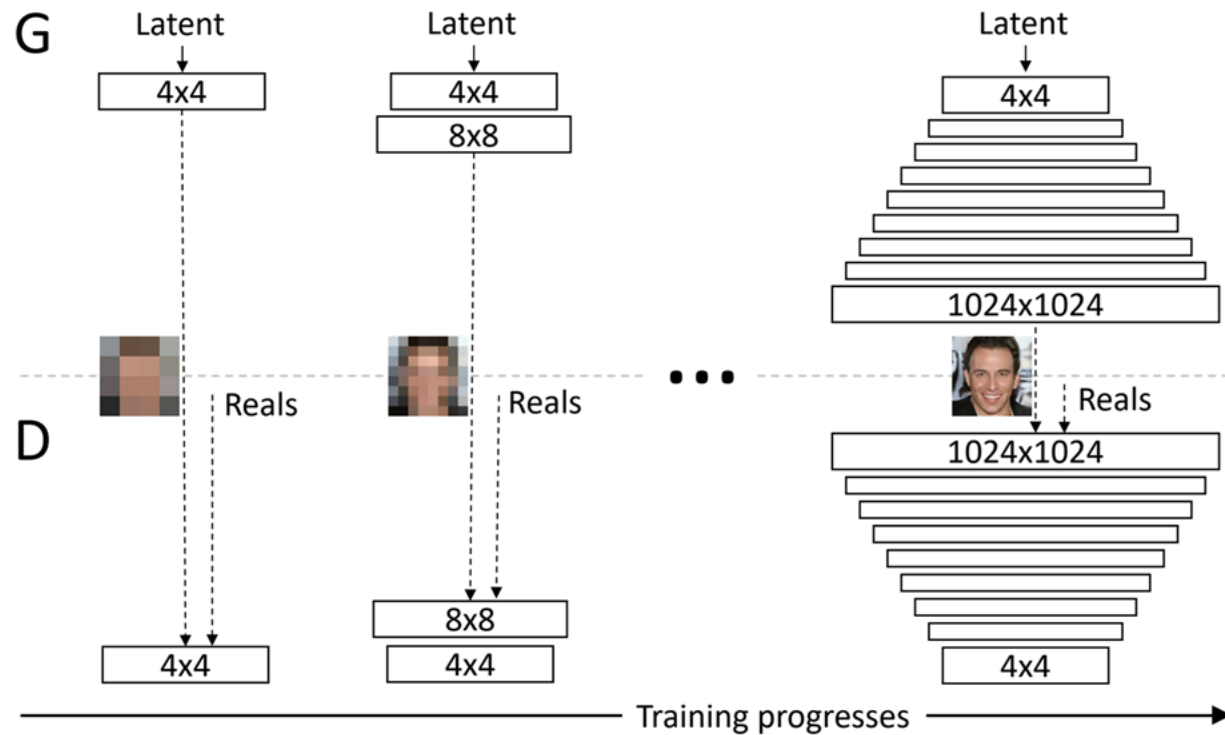
# GAN



- Generator : 진짜처럼 보이는 데이터 생성
- Discriminator : 데이터가 진짜인지 가짜인지 구분하는 역할

## Introduction

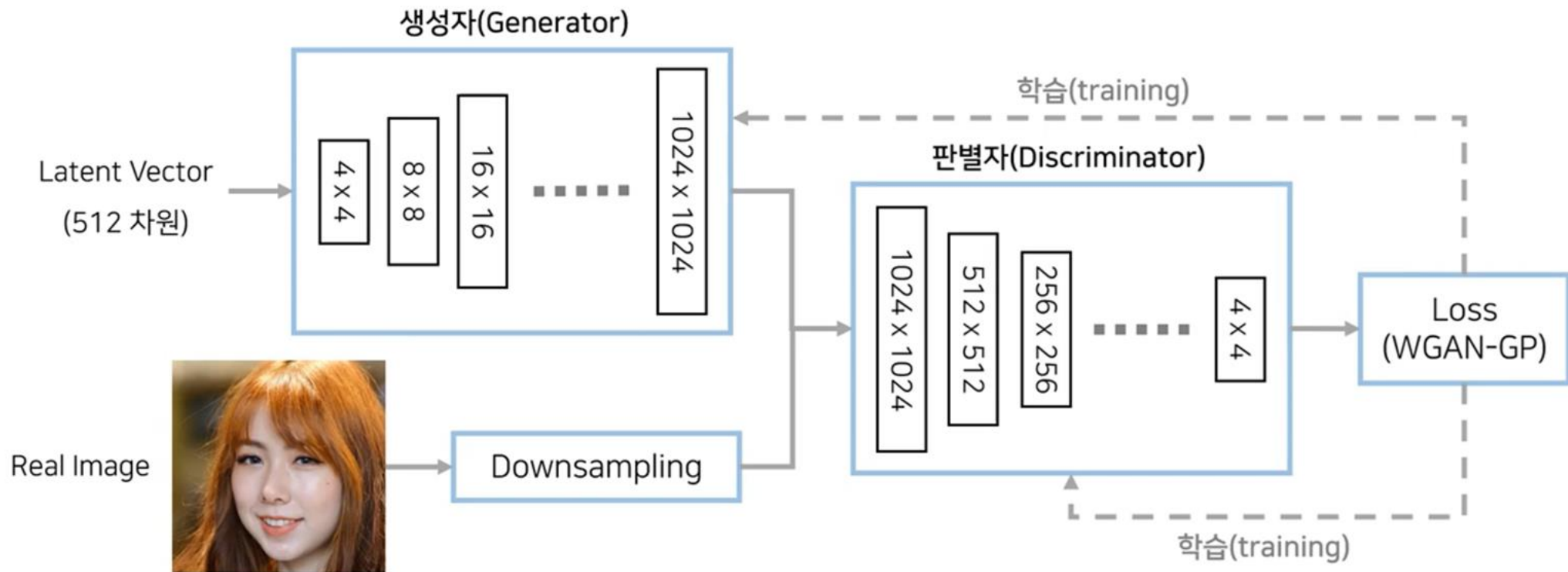
# PGGAN



- 학습 과정에 layer를 추가해 나감
- 이를 통해 점진적으로 해상도를 높여감
- 이미지 특징 제어가 어려운 한계가 있음

## Introduction

# PGGAN

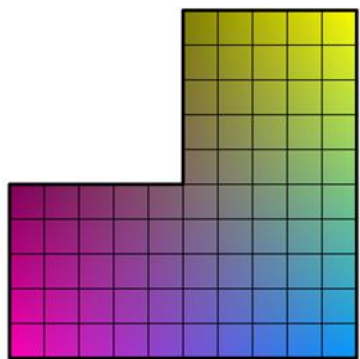


---

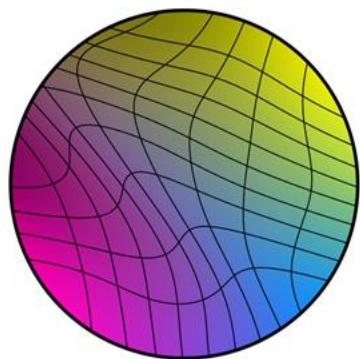
CHAPTER 02

# StyleGAN

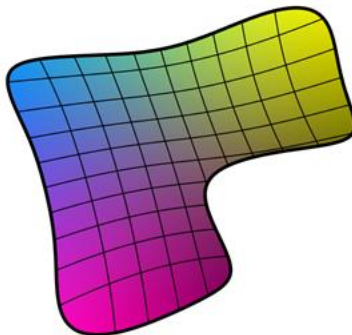
# Mapping Network



(a) Distribution of features in training set



(b) Mapping from  $\mathcal{Z}$  to features

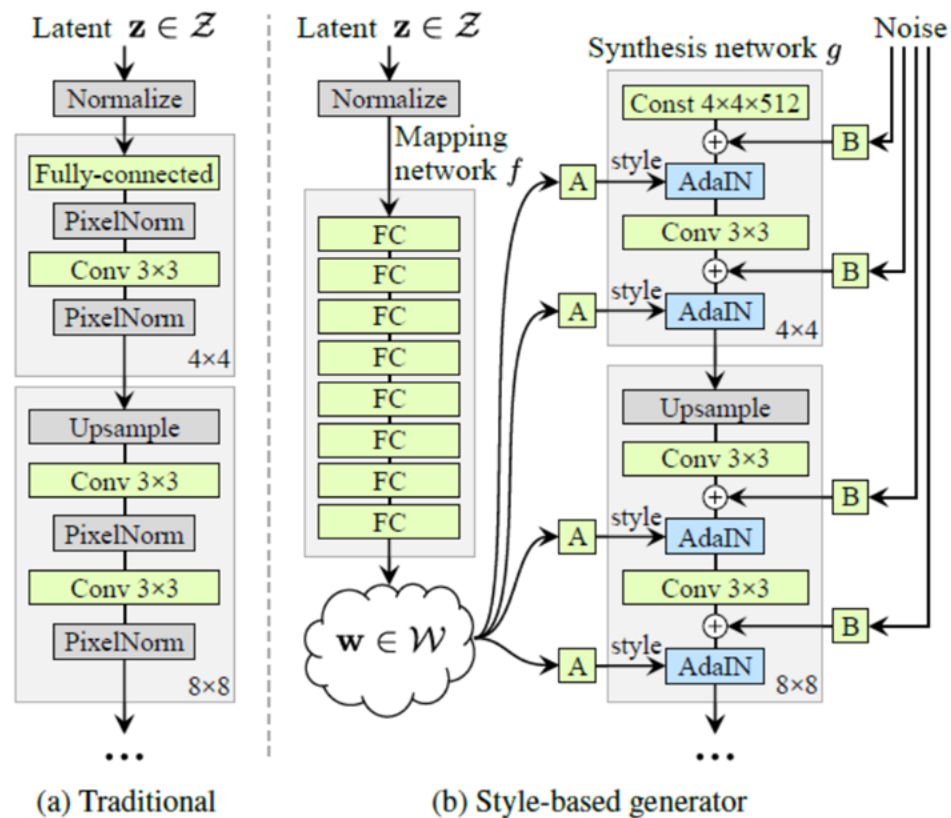


(c) Mapping from  $\mathcal{W}$  to features

- 512차원의  $z$  도메인에서  $w$  도메인으로 매핑
- 가우시안 분포에서 샘플링한  $z$  벡터를 직접 사용하지 않음
- StyleGAN의 생성자는 더욱 linear하며 덜 entangled 되어 있음.

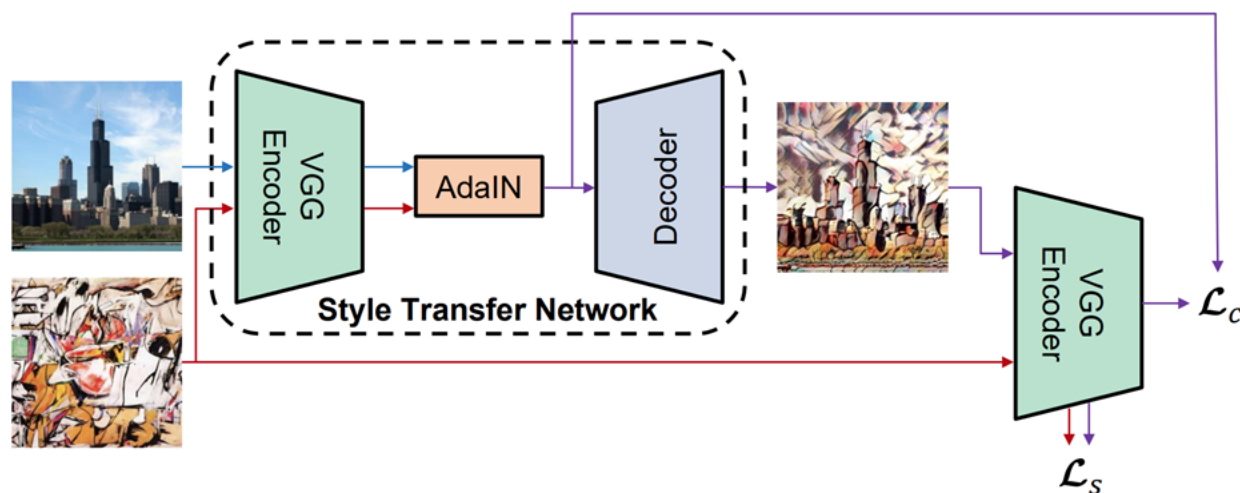


# StyleGAN Architecture



- Mapping된  $W$ 의 정보들을 AdaIN layer에 전달
- Noise를 Bias 형태로 더하여 줌

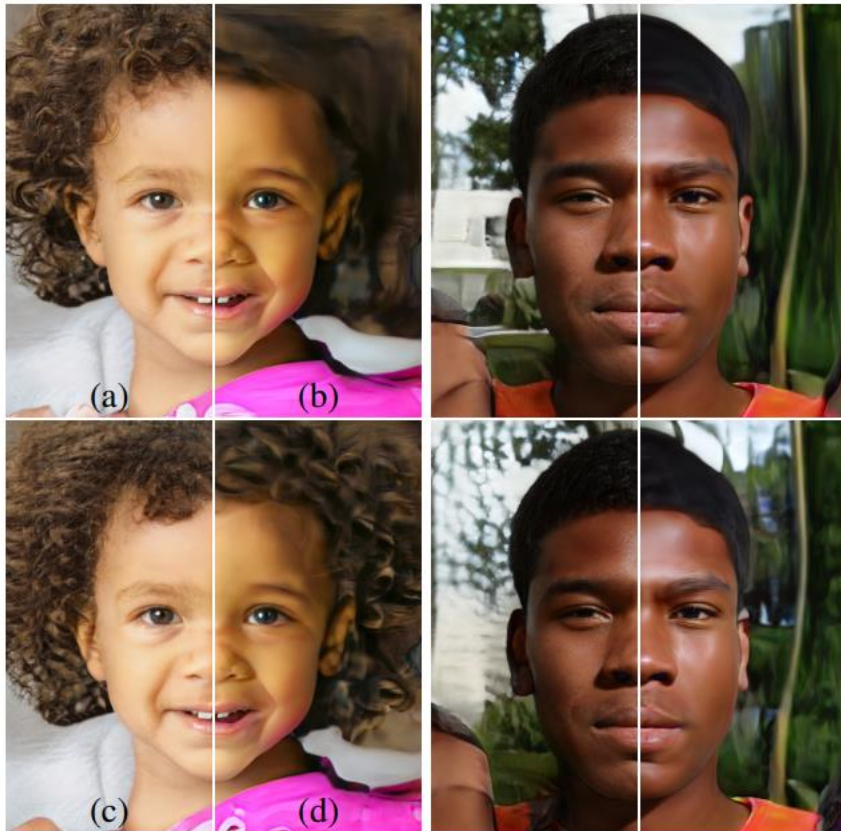
# StyleGAN ADaIN



$$\text{AdaIN}(x, y) = \sigma(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y)$$

- VGG Encoder : Content image와 Style image를 input하여 특징을 추출
- 추출된 각 특징  $x, y$ 는 Adain을 통해 결합
- Decoder를 통해 Image 변환
- VGG Encoder를 통해 Loss 계산

# Stochastic Variation



- 스타일 : high – level global attributes
- 노이즈 : stochastic variation 주근깨, 피부 모공

Coarse noise : 큰 크기의 머리 곱슬거림, 배경 등

Fine noise : 세밀한 머리 곱슬거림, 배경 등

(a) 모든 레이어에 노이즈 적용 (b) 노이즈 없음

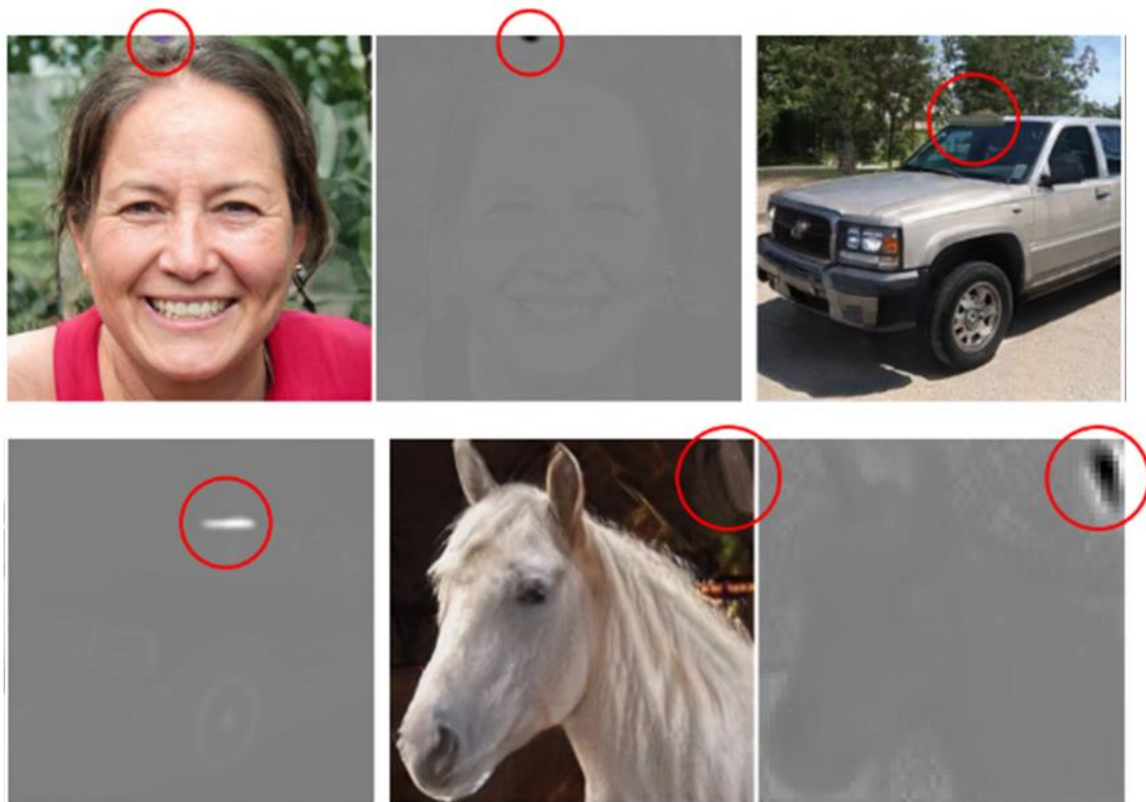
(c) Fine layer에 적용 (d) coarse layer에 적용

---

CHAPTER 03

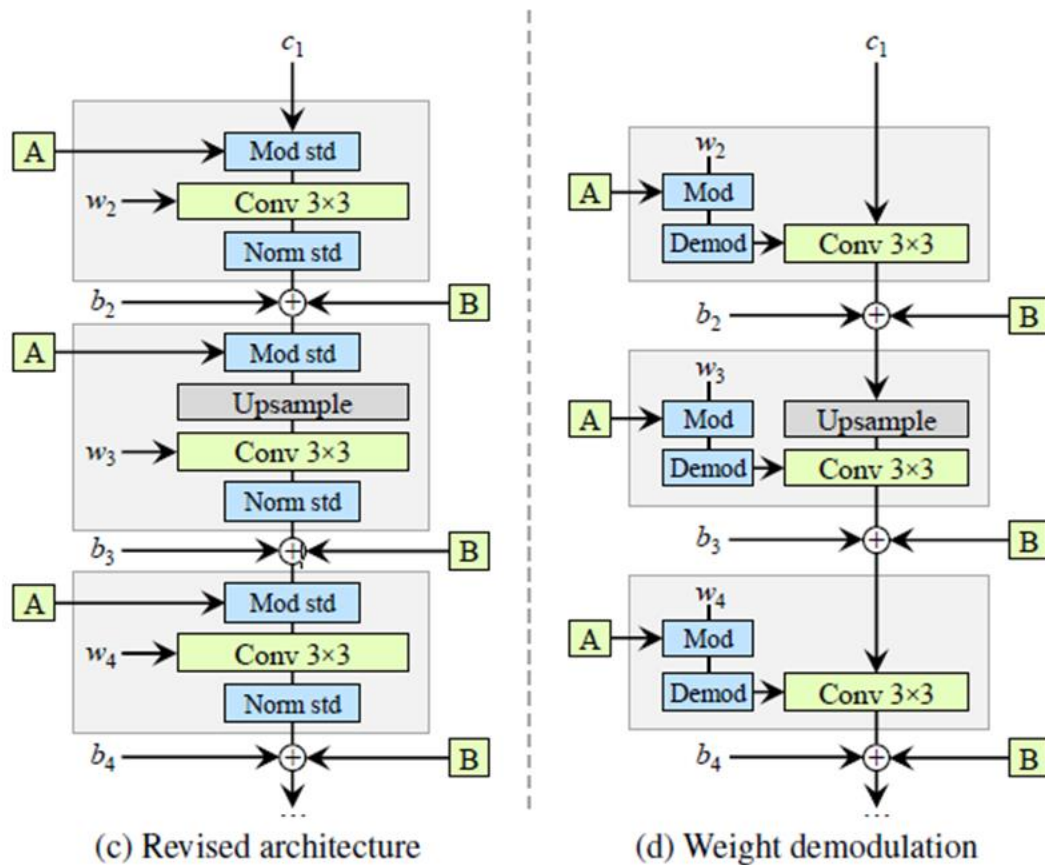
# StyleGAN2

# StyleGAN limitation



- Water droplet artifact : 생성된 이미지에 나타나는 물방울 모양의 노이즈

# Architecture



- Normalization을 제거하여 water droplet의 문제 해결
- 최종 아키텍처는 std로 나누지 않음으로써 A에서 넘어온 scaling factor를 conv weight에 곱하여 scaling을 진행

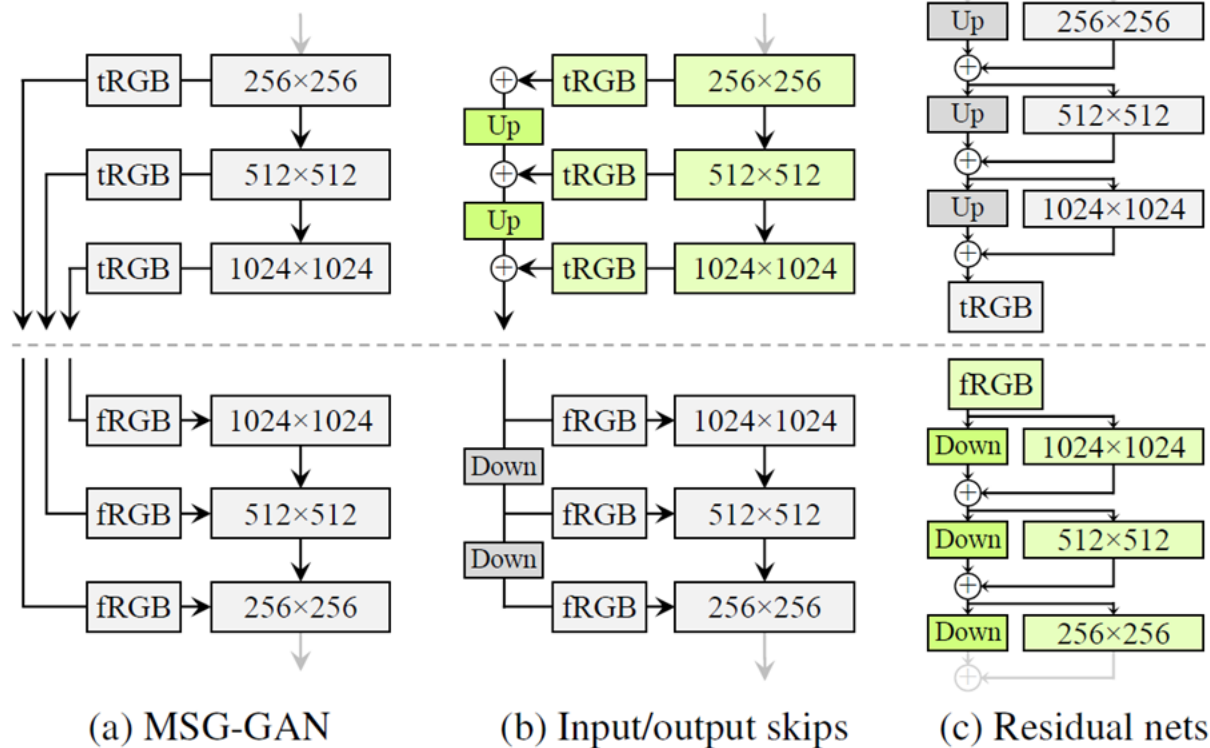


# StyleGAN limitation



- Phase artifact : 일부 feature의 중심이 얼굴의 움직임을 따르지 않음

# Progressive growing revisited



- MSG-GAN : generator와 discriminator의 resolution을 맞춰서 skip connection
- Input/output skips : 다른 resolution의 RGB output을 upsampling/downsampling하여 sum.
- Residual nets : residual block

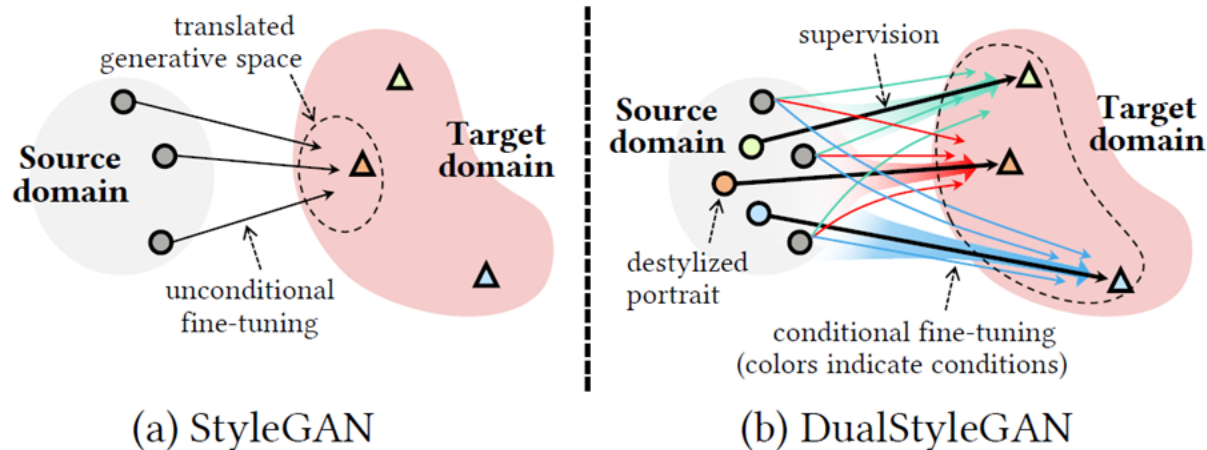


---

CHAPTER 04

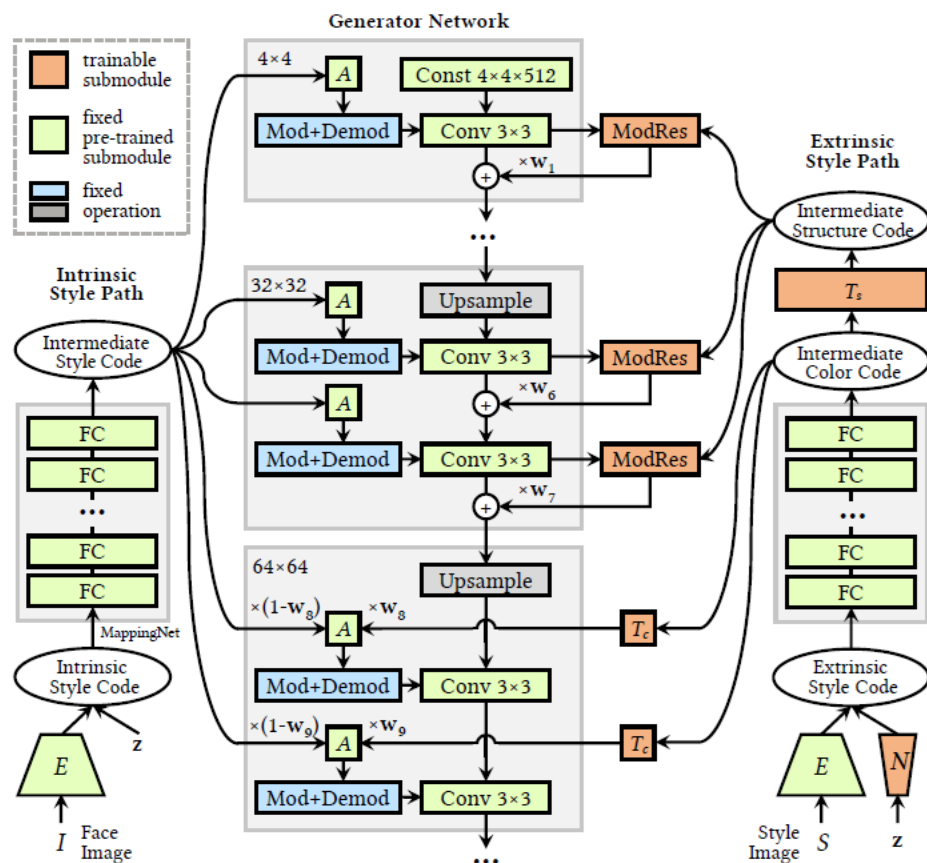
# DualStyleGAN

# Image to image translation



- 기존의 fine-tuning은 공간을 전체적으로 변환하여 스타일의 다양성을 잃음
- DualStyleGAN은 외부 스타일 네트워크를 이용하여 기존의 도메인을 해치지 않음과 동시에 다양한 스타일을 도메인에 매핑

# DualStyleGAN Architecture



- $E$  : 입력 이미지 인코더
- $N$  : 스타일 이미지 인코더
- **Mod + Demod** : 스타일 조정 모듈, 주어진 스타일 코드를 레이어별로 조정
- **ModRes** : Modulation Residual 블록, 스타일 정보와 기본 구조 결합

---

CHAPTER 05

# 코드구현

# 코드 구현

```
import torch
from torchvision import transforms
from PIL import Image

# 모델 로드 (paprika 스타일 사용)
model_path = "paprika.pt"
model = torch.hub.load("AK391/animegan2-pytorch:main", "generator")
model.load_state_dict(torch.load(model_path))
model.eval()

# 이미지 로드 및 전처리
def load_image(path):
    img = Image.open(path).convert("RGB")
    return img

transform = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
])
```

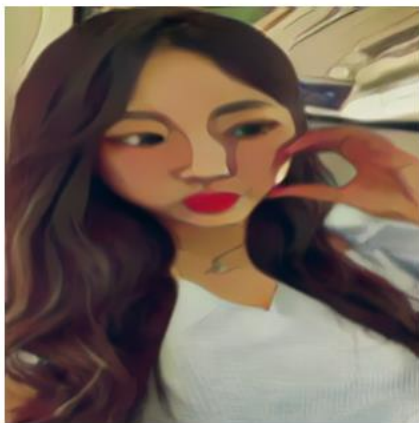
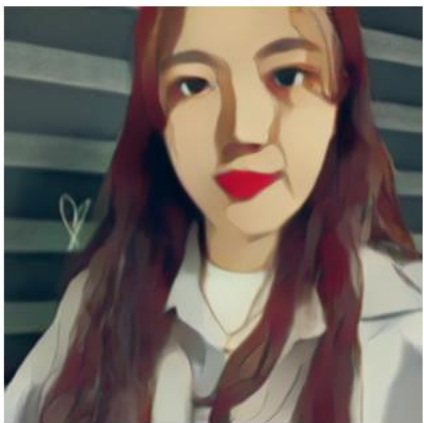
# 코드 구현

```
img = load_image("C:/Users/noori/Pictures/사진/KakaoTalk_20241126_114702173.jpg")
img = transform(img).unsqueeze(0)

# 모델에 이미지 전달
with torch.no_grad():
    output = model(img)

# 후처리 및 저장
output = output.squeeze().permute(1, 2, 0).numpy()
output = (output * 0.5 + 0.5) * 255
output = output.astype("uint8")
output_img = Image.fromarray(output)
output_img.save("anime_style_image2_8.png")
```

# 구현 결과



# Q&A