

# 생성형 모델을 활용한 다양한 서비스

- 랜덤 포켓몬 생성 - 얼굴 화장, 염색 - 가상 인물 생성

---

2023.10.26 이태형

1. 프로젝트 소개
2. 생성형 모델 소개
3. 포켓몬 생성 모델
4. 이미지 합성 모델
5. 가상인물 생성 모델
6. 웹/플라스크 구성
7. 웹페이지 소개
8. QnA

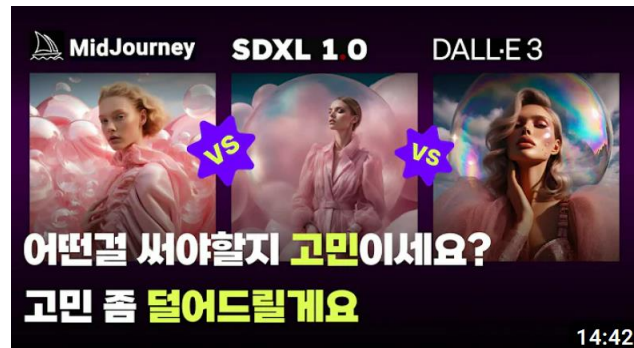
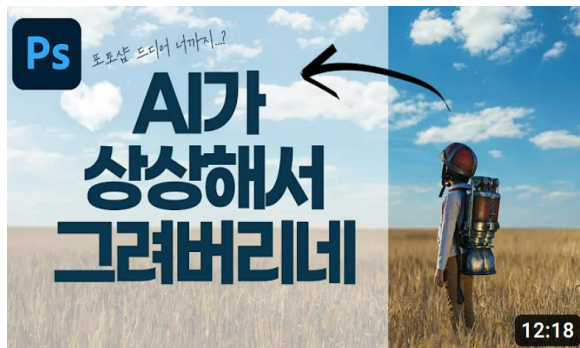
## ▣ 1. 프로젝트 소개

- 주제 소개

여러 생성형 ai 들의 각각의 장단점을 활용해 이미지 **생성/합성** 서비스

- 주제 선정 이유

Chat봇 이후 2023년 현재 ai 분야에서 뜨거운 감자로 떠오른 분야

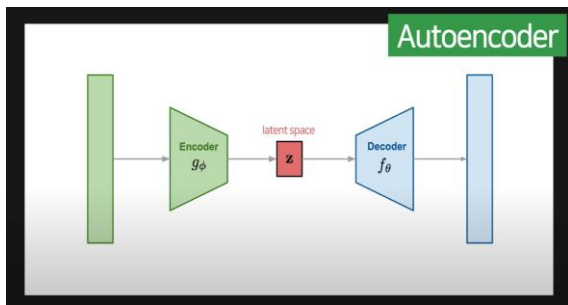


## ▣ 2. 생성형 모델 소개 - 생성형 AI란?

- 생성형 ai란?

대화, 이야기, 이미지, 동영상, 음악 등 새로운 콘텐츠와 아이디어를 만들 수 있는 인공지능의 일종

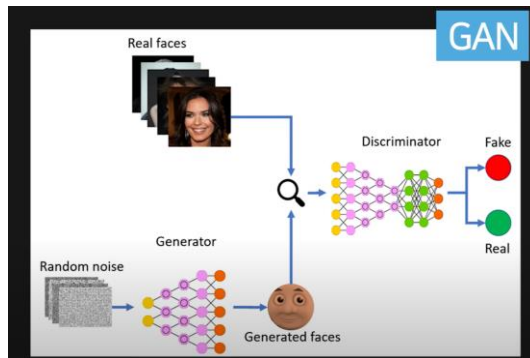
- 생성형 이미지 모델



학습속도 : 빠름

이미지 다양성 : 나쁨

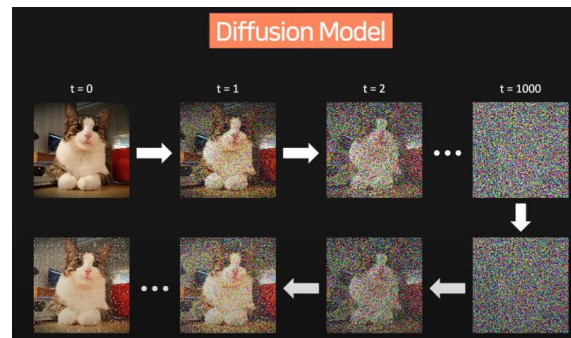
이미지 퀄리티 : 나쁨



학습속도 : 빠름

이미지 다양성 : 나쁨

이미지 퀄리티 : 좋음



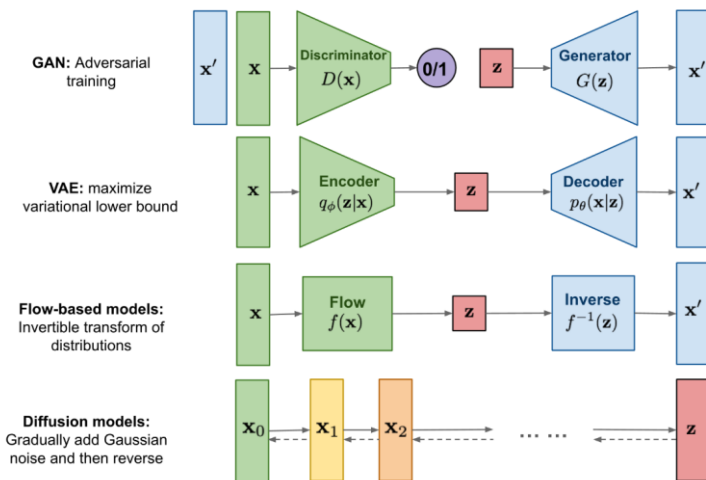
학습속도 : 느림

이미지 다양성 : 좋음

이미지 퀄리티 : 좋음

## ▣ 2. 생성형 모델 소개 - 의문

- 생성형 모델을 처음 배웠을 때 의문
  - 원본처럼 재구성만 하는데 왜 이게 생성형 이미지인가?
- 생성형 모델의 핵심 => 임베딩 공간, 잠재벡터, 잠재공간 **Z**



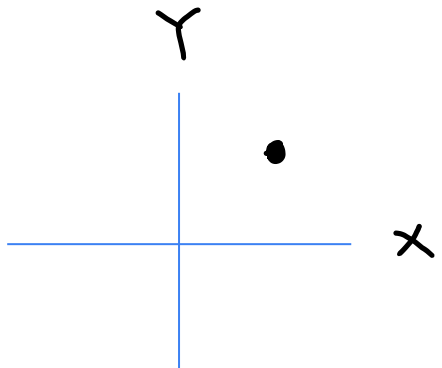
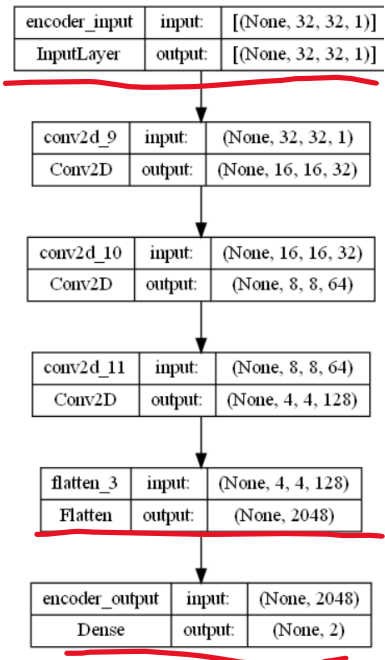
- 잠재공간
  - 비정형 데이터 속에 잠재해 있는 특성이 수치화 된 공간
  - 이 공간에서 비정형 데이터의 특성을 더하거나 빼거나 하는 계산을 할 수 있다

## ▣ 2. 생성형 모델 소개 – 잠재공간1

- 잠재공간 예시
  - 오토 인코더



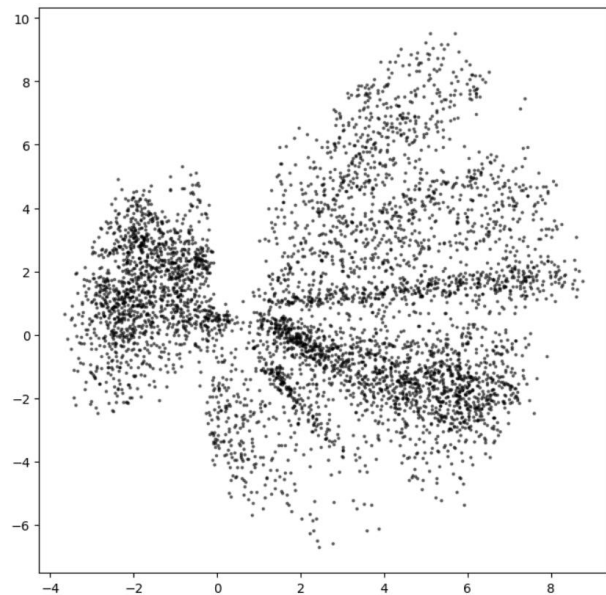
32 x 32 이미지 = 2차원 32x32 행렬



1 차원의 2개의 숫자 데이터 =>  
X, Y로 지정해서 좌표로 나타 낼 수  
있다

## ▣ 2. 생성형 모델 소개 – 잠재공간2

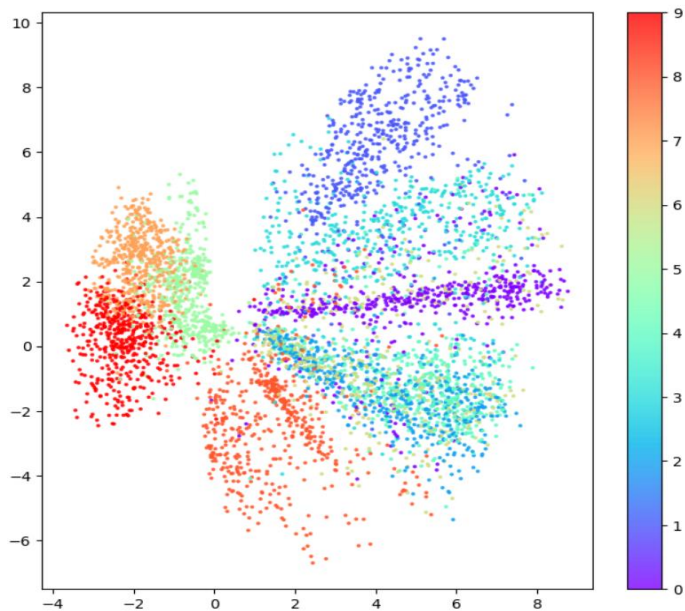
- 잠재공간 예시
  - 오토 인코더



- 모든 이미지 TRAIN 데이터를 인코더로 2차원의 X좌표 Y좌표로 임베딩 시키면 잠재공간 표현 가능
- 각 점들이 이미지 데이터

## ▣ 2. 생성형 모델 소개 – 잠재공간3

- 잠재공간 예시
  - 오토 인코더

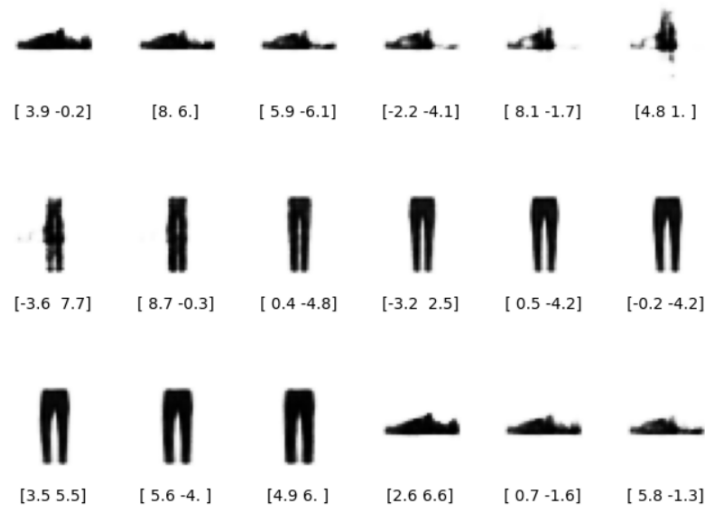
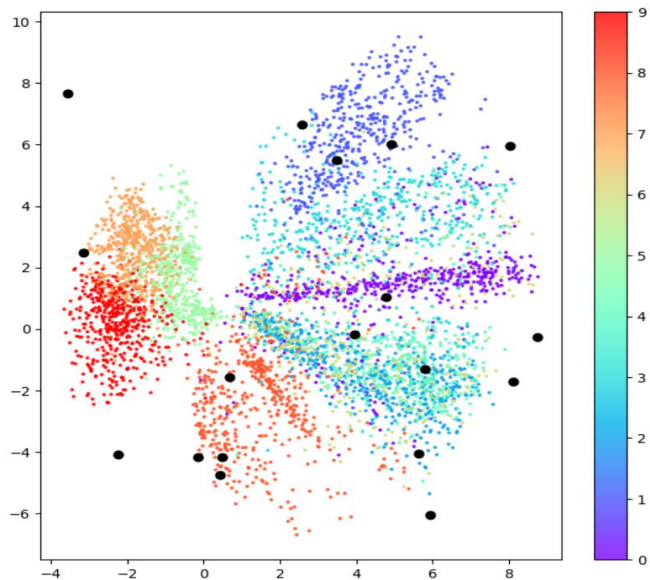


Label을 넣으면 각 분류들이 어떤 방향 즉 어떤 벡터로 뺏어 나가는지 볼 수 있음



## ▣ 1. 생성형 모델 소개 - 잠재공간4

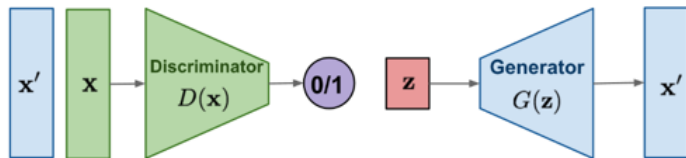
- 잠재공간 예시  
- 오토 인코더



잠재공간 내에서 무작위 좌표를 입력해 디코딩을 하면 기존에 없던 데이터로 생긴 새로운 이미지가 생성

## ▣ 1. 생성형 모델 소개 – 잠재공간5

- 잠재공간 예시
  - GAN



$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

- $E_{x \sim p_{data}(x)}$  : 실제 데이터에 대한 확률분포에서 샘플링한 데이터
- $E_{z \sim p_z(z)}$  : 일반적으로 가우시안분포를 사용하는 임의의 노이즈에서 샘플링한 데이터
- $\max V(D)$  = 앞 항과 뒤 항이 최대가 되어야 함  $\Rightarrow D(x) = 1$  (실제 데이터),  $D(G(z)) = 0$  (가짜데이터)
- $\min V(G)$  = 뒤 항이 최소가 되어야 함  $\Rightarrow D(G(z)) = 1$  (가짜 데이터)

## ▣ 1. 생성형 모델 소개 - 잠재공간6

- 잠재공간 예시  
- GAN
- 생성자가 학습한 딥러닝 매핑 ( $z \rightarrow G(z)$ )는 단순한 불연속적인 1:1매칭이 아니라 데이터의 확률분포를 학습하므로써 입력에서의 약간의 변화가 출력에서도 부드러운 변화로 표현 가능!



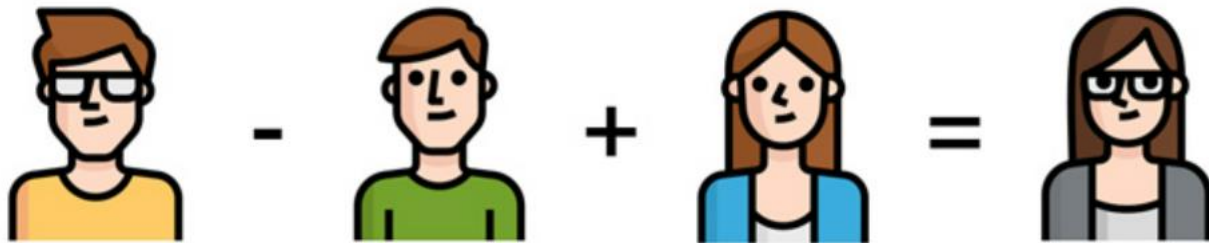
- G의 출력이 사람의 얼굴이라고 했을 때, 왼쪽을 바라보는 얼굴을 만들어 내는  $z$ (왼쪽그림)들의 평균벡터와 오른쪽을 보고있는  $z$ (오른쪽그림)들의 평균을 계산하고 이 두 벡터사이의 축을 중간에서 보간해 새로운 이미지 생성

## ▣ 1. 생성형 모델 소개 – 잠재공간7

- 잠재공간 예시  
- GAN

- GAN의 산술연산!!!

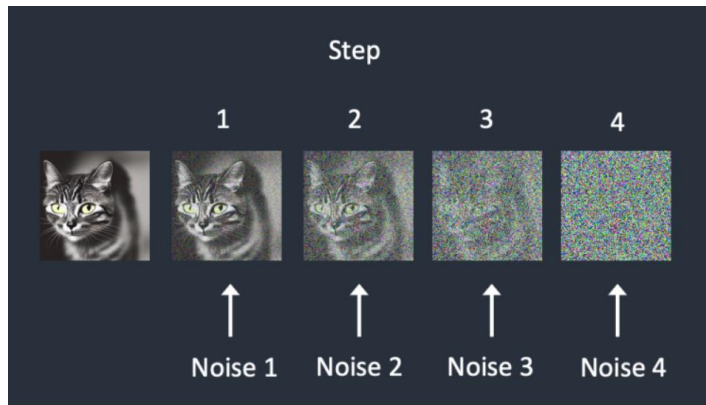
‘안경을 쓴 남자’ 이미지를 생성하는 z벡터에서 ‘안경을 쓰지 않은 남자’ 이미지의 입력인 z벡터를 빼고  
‘안경을 쓰지 않은 여자’ 이미지에 해당하는 z벡터를 더해 생성자 G에 넣어주면 ‘안경을 쓴 여자’ 이미지가 나타남



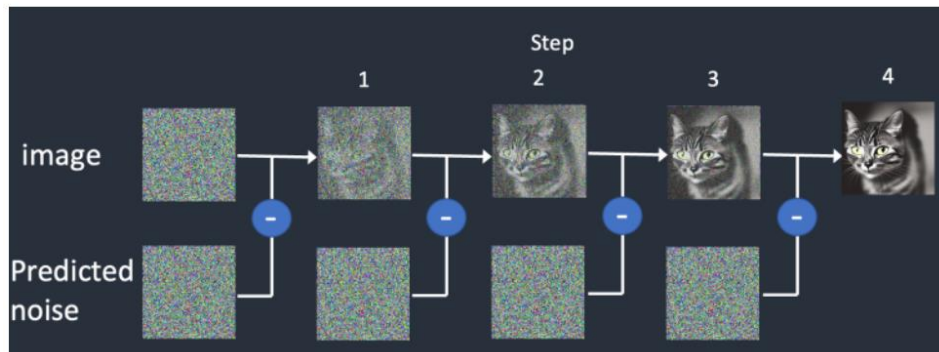
- 따라서 GAN은 생성자의 결과물을 우리가 원하는 데로 마음껏 조작할 수 있음!

## ▣ 1. 생성형 모델 소개 – 잠재공간8

- 잠재공간 예시
  - DIFFUSION



각 단계마다 잡음이 추가된다. 잡음 예측기는 총 잡음을 추정한다.



역방향 확산은 이미지에서 예측된 잡음을 순차적으로 제거한다.

확산을 뒤집기 위해 잡음 예측기인 U-NET 모델 활용

1. 학습용 이미지를 선택
2. 무작위 잡음 이미지 생성
3. 학습용 이미지에 정해진 단계만큼 무작위 잡음 이미지를 추가하여 손상
4. 잡음 예측기에게 잡음 추가량을 학습, 가중치를 조정하고 정답을 보여주는 방식으로 작업 수행

## ▣ 1. 생성형 모델 소개 – 잠재공간9

- 잠재공간 예시
  - DIFFUSION

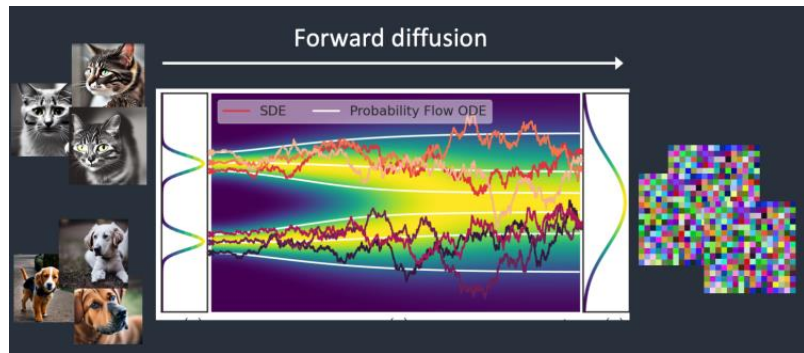
Diffusion models:  
Gradually add Gaussian  
noise and then reverse



- Overview

- $q(X_t | X_{t-1}) = N(X_t; \mu_{X_{t-1}}, \Sigma_{X_{t-1}}) = N(X_t; \sqrt{1 - \beta_t} X_{t-1}, \beta_t * I)$ 
  - Conditional gaussian distribution
  - $0 < \beta_1 < \beta_2 < \dots < \beta_T < 1$
  - $0.0001 \sim 0.02$

mean      variance

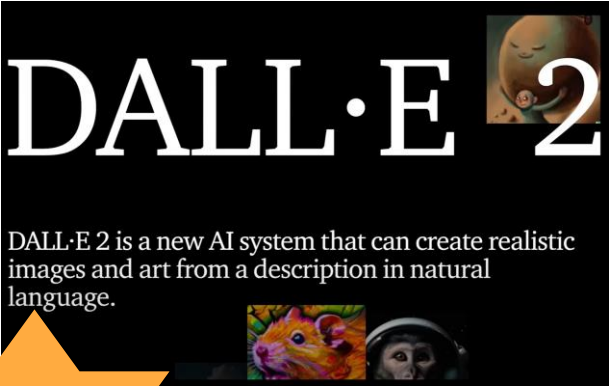


단순한 불연속적인 1:1매칭이 아니라 데이터의 확률분포를 학습하므로

학습한 모델에 무작위 이미지인 z 넣으면 잡음을 제거하면서 새로운 이미지를 생성!!

## ▣ 2. 생성형 모델 소개 – latent diffusion model 1

- 왜 갑자기 이미지 생성 ai가 부상했나?



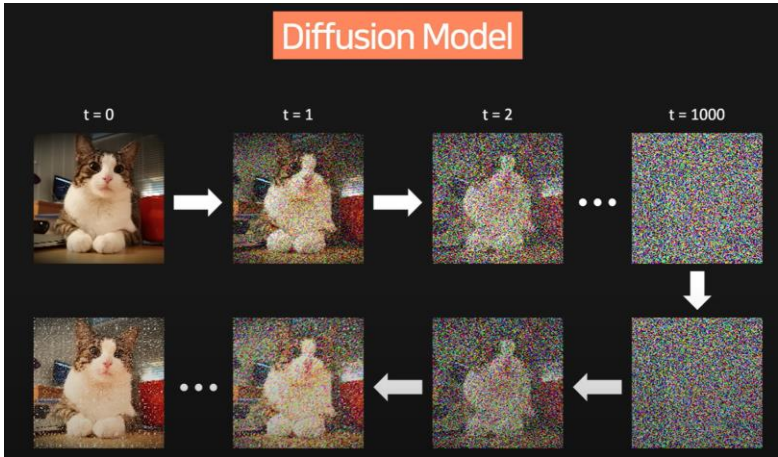
old

뭐가  
새롭지 ?

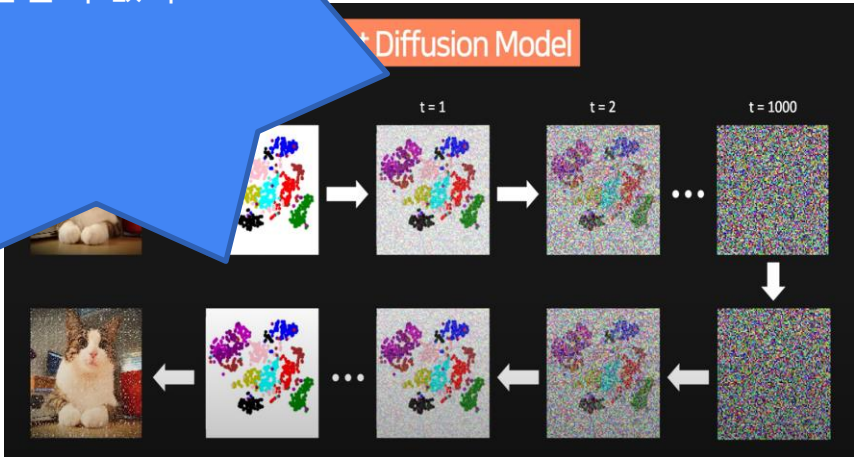
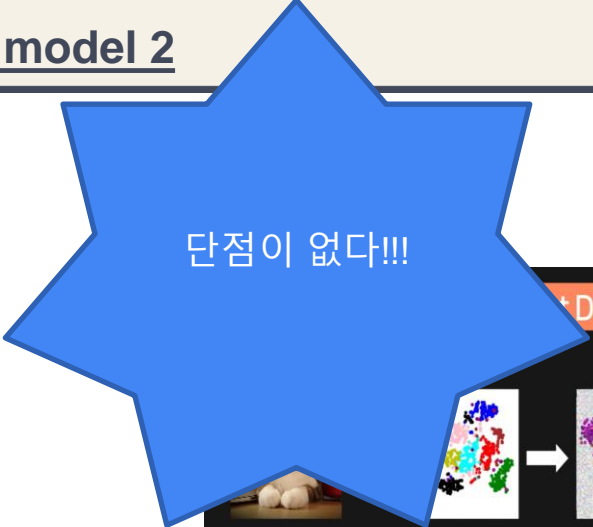


새롭다

2. 생성형 모델 소개 – latent diffusion model 2



학습속도 : 느림  
이미지 다양성 : 좋음  
이미지 퀄리티 : 좋음

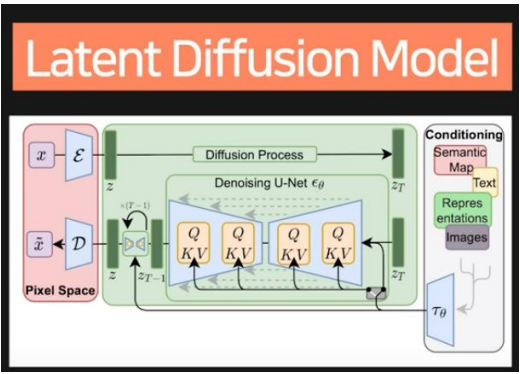


학습속도 : 빠름  
이미지 다양성 : 좋음  
이미지 퀄리티 : 좋음

= 비용 감소



## ▣ 2. 생성형 모델 소개 - 스테이블 디퓨전



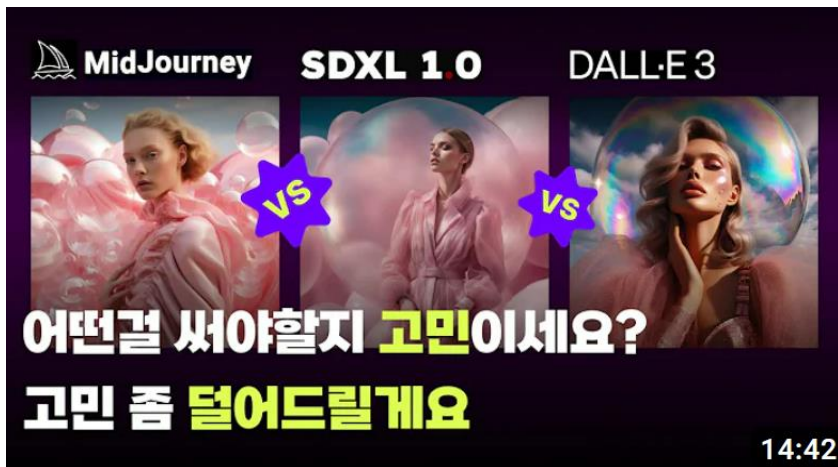
Stability.ai 에서 LDM모델과 엄청난 수의 고퀄리티 이미지 데이터 셋으로 stable diffusion 모델 구현  
-> 오픈 소스로 제공, 상업적 이용 가능

## ▣ 2. 생성형 모델 소개 – 생성형 이미지 ai가 부상한 이유

학습이 완료된 모델을 오픈 소스로 제공, 상업적 이용 가능으로

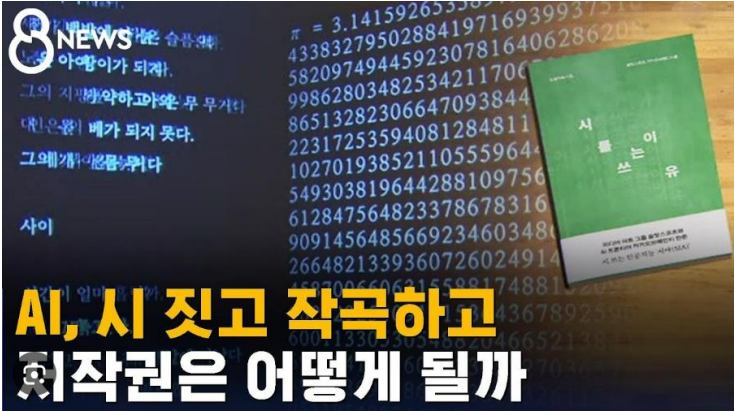
학습자가 주제에 맞는 이미지 데이터를 추가로 학습

**fine-tuning**을 거쳐 새로운 서비스를 상업적으로 이용하기 시작



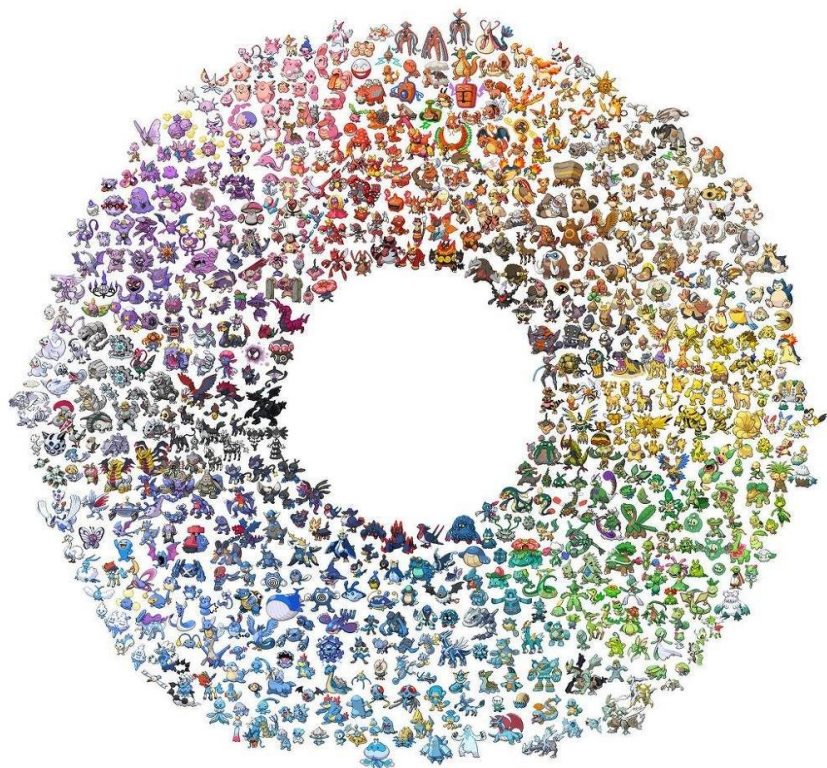
## 2. 생성형 모델 소개 – 생성형 ai의 해결 과제

- 저작권 문제



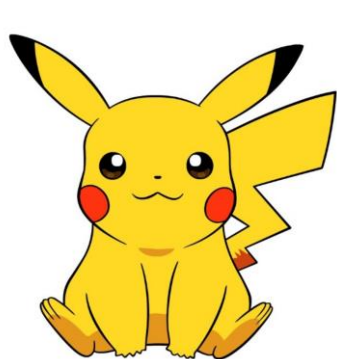
### ▣ 3. 포켓몬 생성 모델 소개 - 1

- 데이터 : Kaggle의 **Complete Pokemon Image Dataset**  
898종의 각 3~4장, 총 2503장의 이미지 데이터
- 모델 : diffusion 모델
- 환경 : 구글 코랩, 실증랩 사용



### ▣ 3. 포켓몬 생성 모델 소개 - 2

- 데이터 선정 이유 : 포켓몬의 디자인은 전체적으로 '3색의 법칙'을 따름  
=> 학습할 때 하나의 이미지에 필요한 색상 수가 적으므로 학습을 잘 할 것이라 판단





### ▣ 3. 포켓몬 생성 모델 소개 - 3

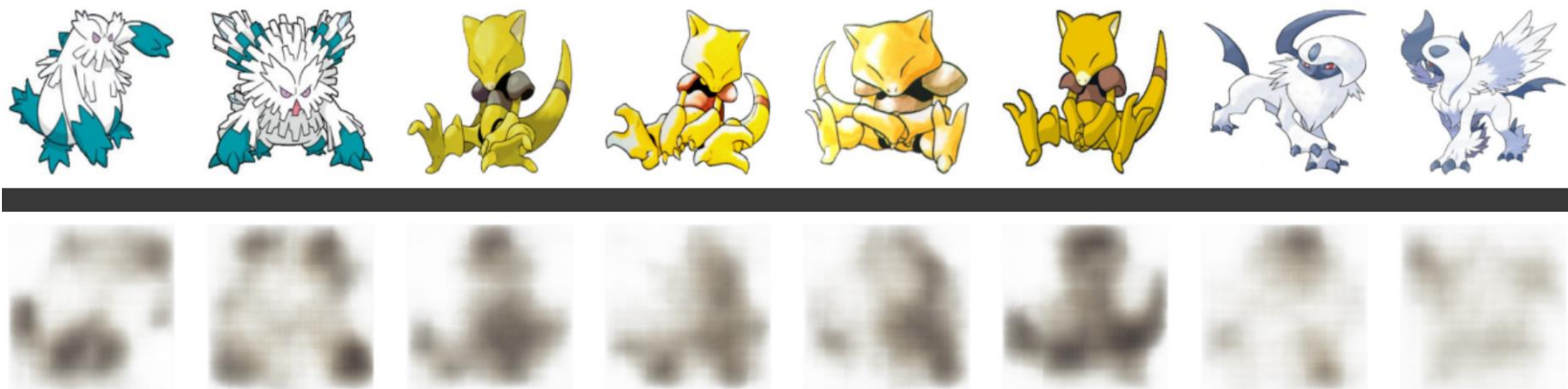
- 모델 선정 이유

변이형 오토인코더를 사용해 보았는데 앞서 설명한 대로 이미지 퀄리티가 너무 좋지 않음

Gan 모델을 사용해 보았는데 형태는 잘 잡지만 색상을 잘 표현하지 못함

따라서 이미지 다양성과 퀄리티가 높은 디퓨전 모델 선정

ex) 오토인코더



### ▣ 3. 포켓몬 생성 모델 소개 - 3

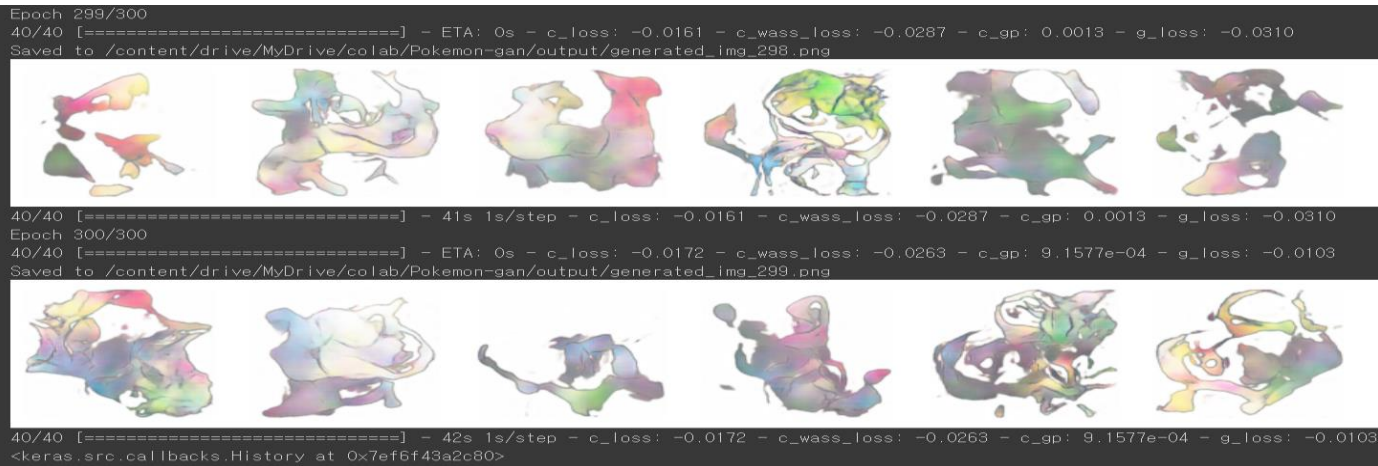
- 모델 선정 이유

변이형 오토인코더를 사용해 보았는데 앞서 설명한 대로 이미지 퀄리티가 너무 좋지 않음

Gan 모델을 사용해 보았는데 형태는 잘 잡지만 색상을 잘 표현하지 못함

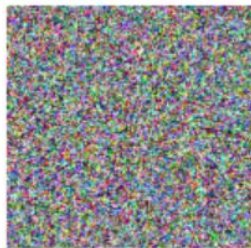
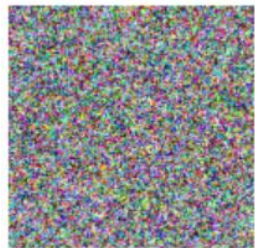
따라서 이미지 다양성과 퀄리티가 높은 디퓨전 모델 선정

ex>) Gan



### ▣ 3. 포켓몬 생성 모델 소개 - 4

- 이미지 생성 과정



1 에포크



6 에포크



12 에포크

- 아쉬운 점

- 1) 학습데이터 수가 너무 적었다 > 좌우반전, 30도 회전 등을 이용해 데이터 수를 늘렸어야 했다.
  - 2) 색상 수가 적다고 포켓몬을 고른것이 패착! 오히려 형태를 고려 했어야 했다!!!
    - > 포켓몬들의 형태가 정형화 되지 않아서 구조를 잡지 못한다
- 이후 꽃 이미지를 따로 학습해 보니 겨우 50 에포크 만에 형상을 실사와 비슷하게 잡음



## ▣ 4. 이미지 합성 모델 소개 - 1

- 데이터 : Kaggle의 **CelebFaces Attributes (CelebA) Dataset**

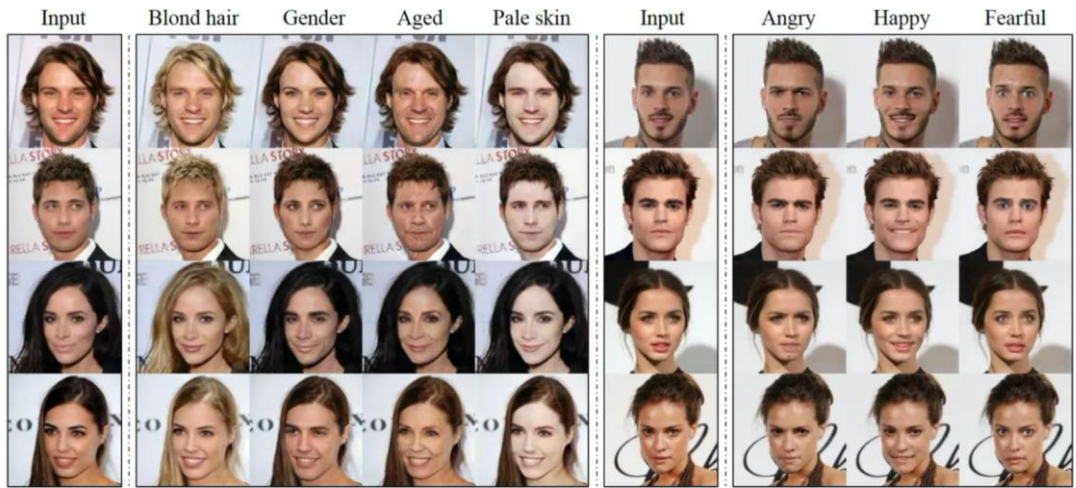
인물사진 약 20만장

- 모델 : 위 데이터를 학습한 stargan 모델을 fine-tuning한 모델
- 모델 선정 이유 : 이미지 합성 시키는 stylegan3, stargan-v2, 등등 여러 모델의 api를 읽어 봤지만 지식 부족으로 학습불가... 그나마 초창기에 만들어진 모델이라 이해하기 쉬웠음
- 환경 : 구글 코랩, 실증랩 사용

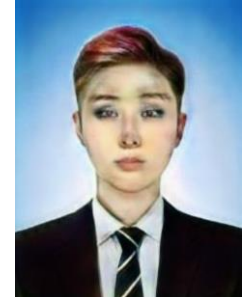
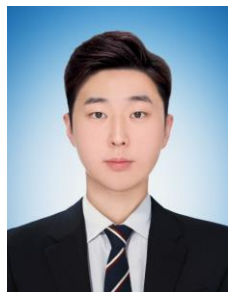


4. 이미지 합성 모델 소개 - 2

- Stargan 활용 예시 =>



- 현실 =>



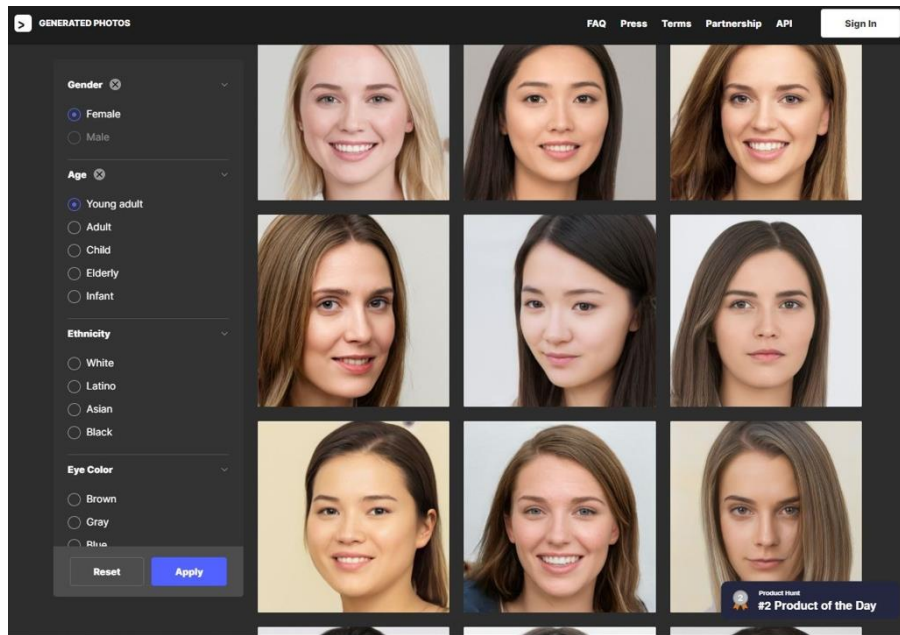
- 아쉬운 점 : celeba 데이터 셋에 한국인 데이터가 없어서 한국인 특화가 되지 않고 화장 진한 여자들이 너무 많아  
진한 화장이 된다

## ▣ 5. 가상인물 생성 모델 소개 - 1

- 데이터 : Kaggle의 **CelebFaces Attributes (CelebA) Dataset**

인물사진 약 20만장

- 모델 : wgan 모델
- 모델 선정 이유 : 더 좋은 모델들도 많았지만  
라벨을 붙여야 해서 이미지 데이터셋만으로도  
좋은 결과를 낼 수 있는 wgan모델 선정
- 환경 : 구글 코랩, 실증랩 사용



## ▣ 5. 가상인물 생성 모델 소개 - 2

- 이미지 생성 과정



1에포크



2시간



12시간

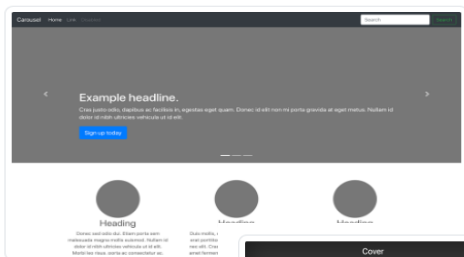


20시간

- 아쉬운 점 : 더 좋은 모델을 활용해 라벨까지 넣어서 학습 시켰다면 더 좋은 결과가 나왔을 것이다

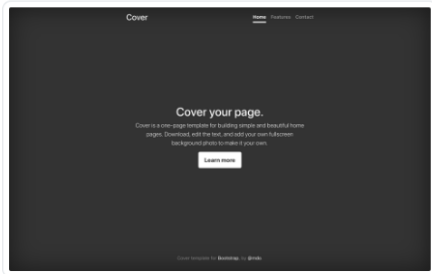
## ▣ 6. 웹/플라스크 구성 - 1

- 모델이 3개 이므로 메인 페이지, 각 서브 페이지 3개 구성
- 시간 단축을 위해 부트스트랩 템플릿 활용



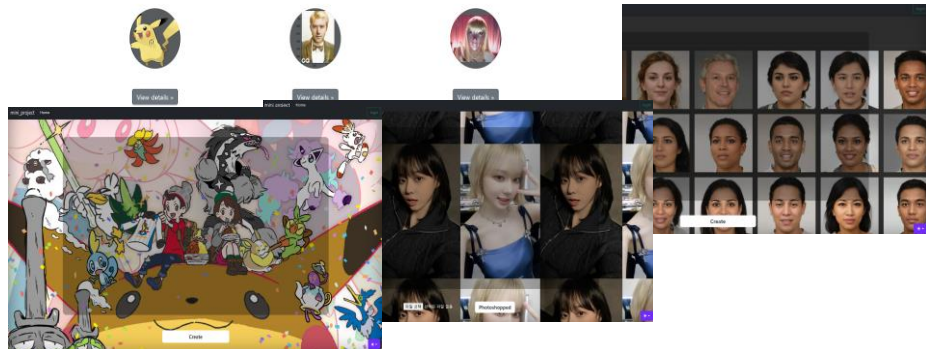
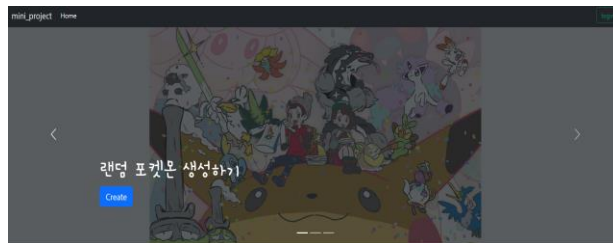
### Carousel

내비바와 캐러셀을  
몇 가지 새로운 컴포



### Cover

심플하고 아름다운 홈페이지 제작을 위한  
단일 페이지 템플릿입니다.



## ▣ 6. 웹/플라스크 구성 - 2

- 플라스크 구성

각 모델들을 py파일로 만들어 app.py에서 임포트 해  
각각 라우팅 함수에서 사용

```
from static.load_models.cover1.poke_models import get_diffusion
from static.load_models.cover2.stargan_models import get_stargan
from static.load_models.cover3.wgan_models import get_wgangp

# cover1 모델 함수
ddm=get_diffusion()

# cover2 모델 함수
solver=get_stargan()

# cover3 모델 함수
wgangp=get_wgangp()
```

- 아쉬운 점 : db연결, 회원가입 기능, 블루스크린, html상속을 이용하지 못해서 아쉽다

## ▣ 7. 웹페이지 소개

---

- 웹 페이지 : <http://127.0.0.1:5000>

QnA