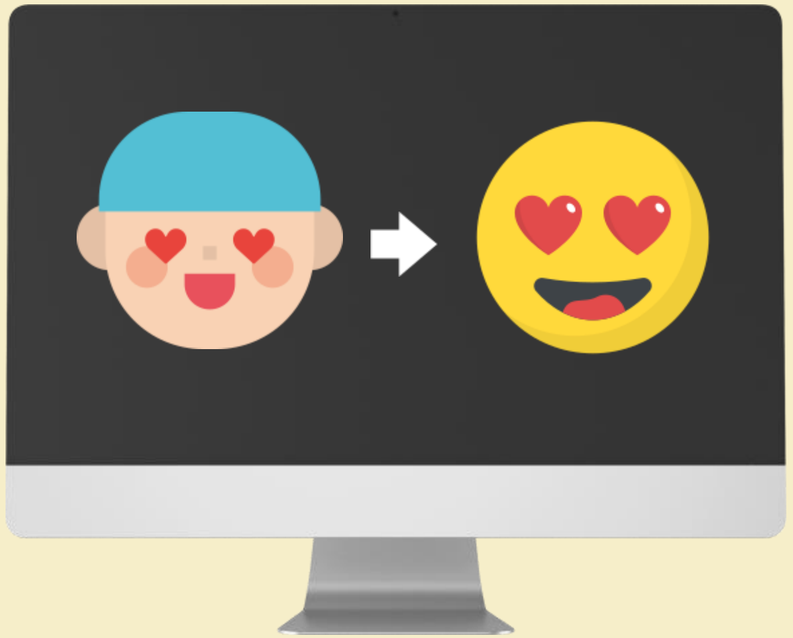


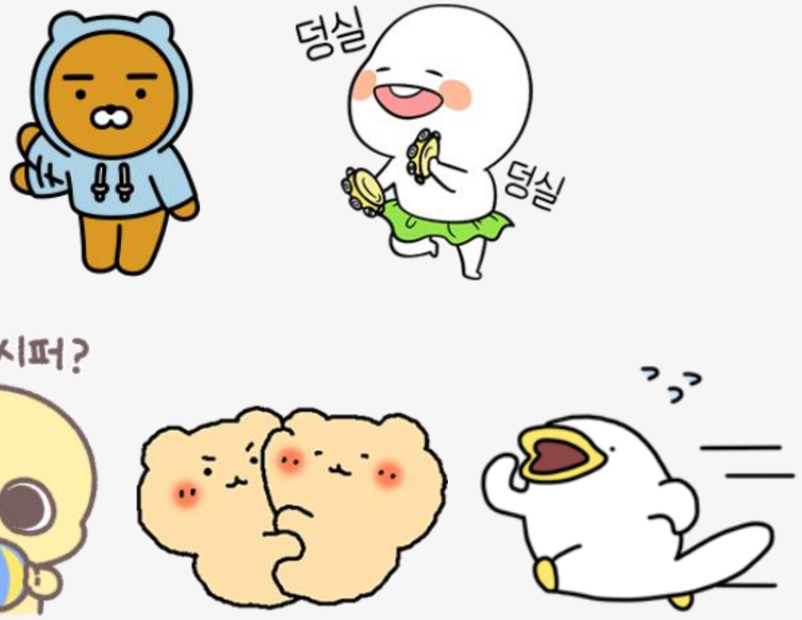
내가 이모티콘이 된다면?!

비 주 얼 어 벤 져 스 조

조문주 이경욱 신예주 윤경서 이태범

CONTENTS

배경



현존하는 이모티콘

작가가 이미 만들어 놓은 이모티콘 사용으로 표현의 한계

VS

우리의 Task

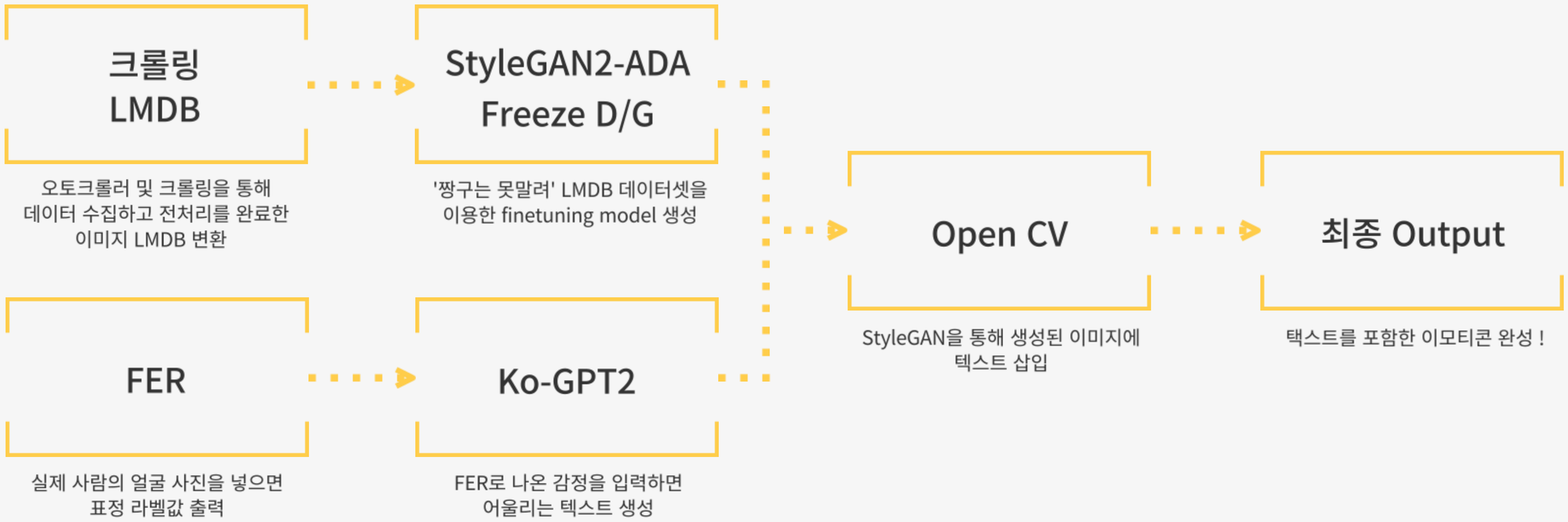
내가 원하는 캐릭터로 나의 표정을 담은 이모티콘을 만들 수 있다면 ?!

Pipeline_Task

- **Task Flow**

Pipeline_Model

- **Model Flow**



StyleGAN2-ADA_모델설명

StyleGAN

이미지를 style의 조합으로 보고 각 layer마다 style 정보를 입히는 방식으로 이미지 합성
→ style transfer 과정이 훨씬 안정적이고 높은 퀄리티의 이미지를 생성하는 모델

StyleGAN2

기존 모델에서 normalization 재구성, PPL지표 활용
→ 기존에 있던 문제점들을 보완한 모델

StyleGAN2 -ADA

ADA는 데이터 증강기법으로 데이터가 부족한 상황에서도
discriminator가 overfitting되지 않고
좋은 성능을 낼 수 있도록 개선한 모델

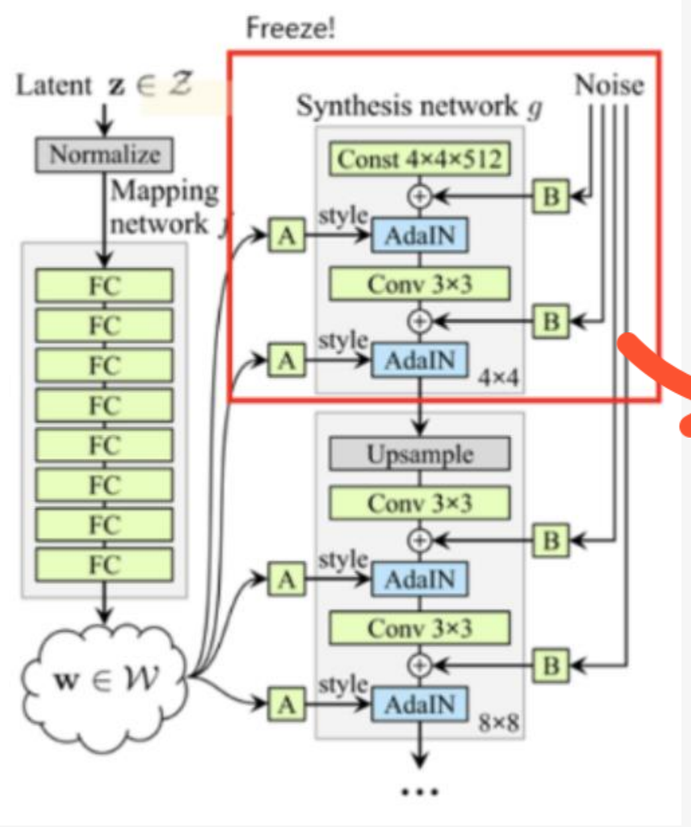
Discriminator(Generator)의 특정 layer를 학습하지 않고 freezing하여 이전에 학습된 네트워크를 최대한 활용
→ 전체를 finetunning하는 방식보다 더 우수한 성능 / 데이터 부족현상 극복 / 빠른 학습속도

Freeze G/D

최종 모델 ?

StyleGAN2-ADA_모델설명

StyleGAN2-ADA + Freeze G/D + Freeze Style



이미지의 특징을 잘 뽑아내는 것도 중요하지만,
원본 이미지의 구조를 유지하는 것도 중요 !!

generator와 스타일 벡터의 저해상도 부분

짱구 이미지의 특성상 캐릭터의 생김새가 거의 비슷하기 때문에
생성 이미지 또한 비슷비슷하지 않을까?

→ 원본 이미지의 구조를 잘 유지할 수 있도록 스타일 벡터도 freezing

StyleGAN2-ADA_데이터셋

크롤링

네이버, 구글, 인스티즈, 핀터레스트에서
크롤링을 통해 짱구 및 엑스트라 데이터 수집

+

데이터 정제

화질, 사진에 얼굴이 50%이상 차지하는지,
얼굴이 다른 물체에 의해 가려지진 않았는지,
중복 여부 등을 고려하여 데이터 정제

최종적으로
2096개의
학습데이터
구축

LMDB 변환

- StyleGAN2-ADA는
LMDB 형식으로 변환 필요

* LMDB는 메모리 효율이 좋아
대용량 데이터를 load할 때 용이

StyleGAN2-ADA_학습과정

학습횟수에 따른 성능

각각 10000, 230000, 430000번째 모델로 학습했을때 나온 Output

StyleGAN2-ADA_loss / R1 regularization

```python
def d_logistic_loss(real_pred, fake_pred):
    real_loss = F.softplus(-real_pred)
    fake_loss = F.softplus(fake_pred)

    return real_loss.mean() + fake_loss.mean()


def d_r1_loss(real_pred, real_img):
    with conv2d_gradfix.no_weight_gradients():
        grad_real, = autograd.grad(
            outputs=real_pred.sum(), inputs=real_img, create_graph=True
        )
    grad_penalty = grad_real.pow(2).reshape(grad_real.shape[0], -1).sum(1).mean()

    return grad_penalty


def g_nonsaturating_loss(fake_pred):
    loss = F.softplus(-fake_pred).mean()

    return loss
```

StyleGAN Score

- **real_score** (≒discriminator 성능)

  높은 값일 수록 실제 입력 이미지를 진짜 이미지라고 판단

- **fake_score**

  높은 값일 수록 가짜 입력 이미지를 진짜 이미지라고 판단

StyleGAN2-ADA_loss / R1 regularization

Loss / R1 regularization

GAN의 Standard GAN Loss Function

- Minimax GAN Loss
- Non-saturating GAN Loss ✓

→ Non-saturating GAN Loss + R1 regularization

R1 regularization

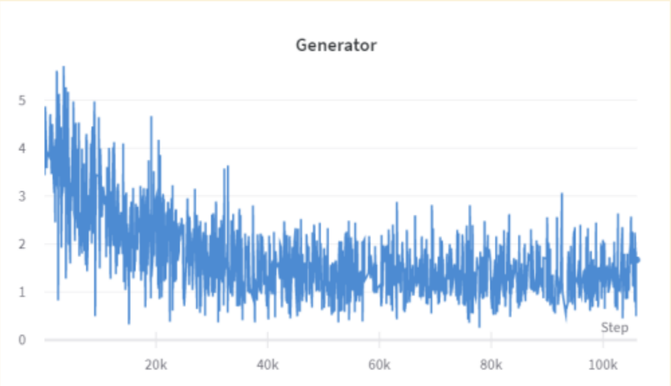
generator loss & discriminator loss

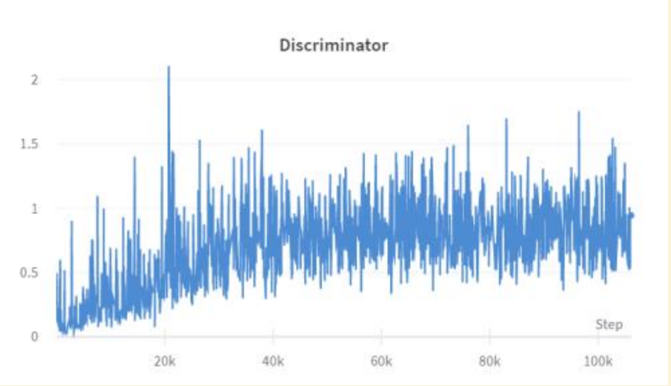
softplus 함수 이용

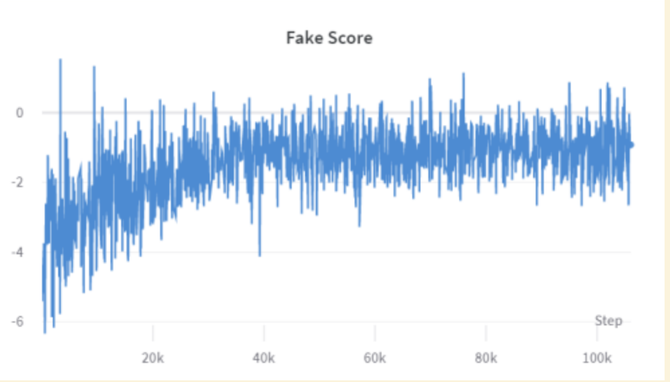
StyleGAN2-ADA_loss / Iteration

Loss

- Generator loss
- Discriminator loss
- Fake score







FER_패키지 설명

VGG16과 같은 분류 모델을 활용해 감정을 분류하려고
하였으나 시간적으로 무리가 있어,
텍스트 생성 task에 집중하는 것이 좋을 것 같다고 판단

FER(Facial Expression Recognition)

FER는 fer+라는 데이터셋을 감정에 따라 7가지의 범주 (화남, 혐오, 두려움, 행복, 슬픔, 놀람, 보통)로 분류하여 표정을 인식하는 패키지

MTCNN - joint learning

- Face Detection
  얼굴 검출
- Face Alignment
  눈, 코, 입 좌표
- Bounding Box Regression
  박스의 위치 조정

FER_과정 및 결과

'box' : array([79, 54, 174, 174]),
'emotions' : {'angry' : 0.28, 'disgust' : 0.0,
'fear' : 0.01, 'happy' : 0.02,
'sad' : 0.03, 'surprise' : 0.01,
'neutral' : 0.66}

사진을 input으로 넣으면 face detection을 통해 얼굴인식
→ bounding box 출력
→ 얼굴 감정을 7개로 분류 & 각각의 확률
(angry, disgust, fear, happy, sad, surprise, neutral)

KoGPT2_모델 선정

언어 생성 모델 중 한국어에 최적화된 모델 탐색
pre-trained model로 inference를 할 수 있는 모델
→ SKT-AI에서 제공하고 있는 KoGPT2

* PPL 지표 : 문장을 생성할 때, 다음에 올 여러 후보 단어 중
어떤 단어를 선택할 지 고민하는 정도 및 헷갈리는 정도

KoGPT2

KoGPT2 2.0은 한국어 성능을 극복하기 위해 40GB이상의 한국어 데이터로 Fine-Tuning한 한국어 언어 모델로, PPL 수치가 다른 버전에 비해 높음

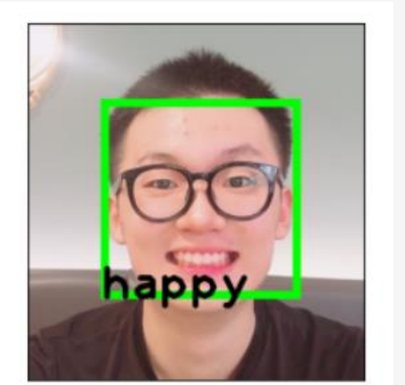
KoGPT2 Inference

챗봇 데이터로 pretrained된 모델 사용

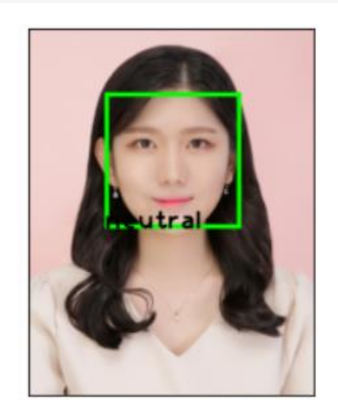
→ 마치 챗봇을 만드는 것처럼 감정분석을 거쳐 분류된 감정이 입력되었을 때 해당 감정과 관련된 문장을 생성.

\+ 감정에 따라 0(중립) / 1(부정) / 2(긍정)로 라벨링 진행

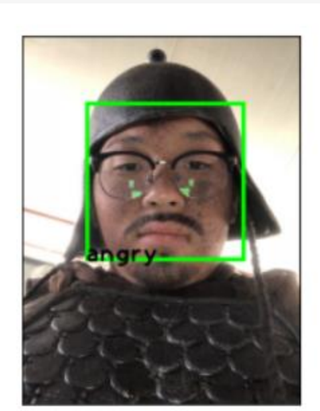
Ko-GPT2_학습결과



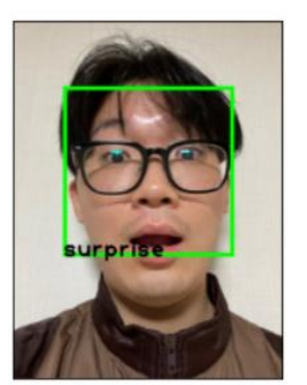
행복행복 액션가면~



흐음



킹받네



당신은 지금 놀라운 상태

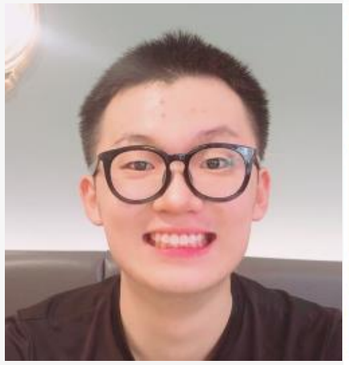
OpenCV로 텍스트 합성

'텍스트를 포함한 이모티콘'

최종결과

팀원 이미지





외국인 이미지





한계점 및 해결방안

데이터셋

한국인 데이터셋으로 모델을 finetuning 진행

단조로운 표정

짱구보다 더 다양한 표정을 가진 style의 그림체로 학습을 진행 및 데이터셋 확대

컴퓨팅 파워

짱구보다 더 다양한 표정을 가진 style의 그림체로 학습을 진행 및 데이터셋 확대

Q&A