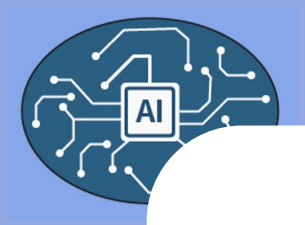


Guideline for Final Project

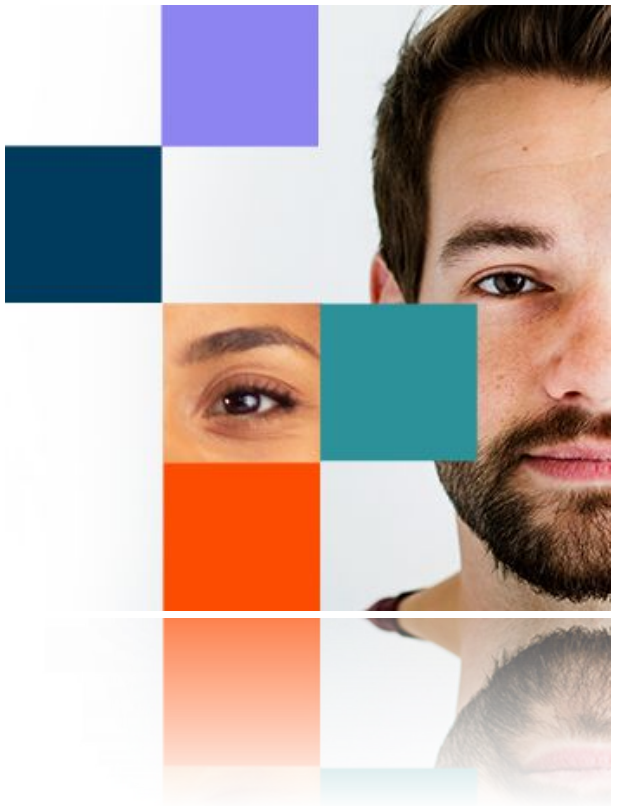
2025.08.01. (월)

작성자 : 전연주, 임수연, 주서영

OUTTA

Deep Fake Detection

- **딥페이크(Deepfake)**
  - 딥러닝(Deep Learning) + 가짜(Fake)의 합성어로 실제와 구분하기 어려운 가짜 이미지나 영상을 생성하는 기술



- **활용 사례**
  - 영화, 광고 등에서 얼굴을 합성하거나 영상 속 얼굴을 다른 얼굴로 바꿔 신원을 보호
  - 가짜 뉴스, 명예훼손, 사기 등 사회적 피해를 유발

→ 딥페이크 탐지(Deepfake Detection) 기술 필요

OUTTA

Task: Deep Fake Detection

Task

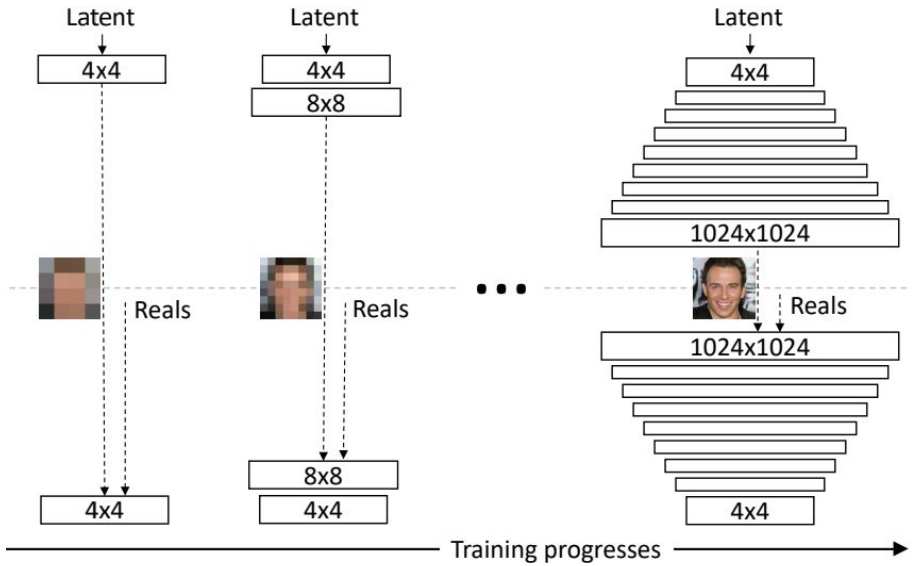
- 이번 프로젝트에서는 **ProGAN**과 **DDPM/DDIM**을 이용하여 높은 품질의 가짜 이미지를 생성해보고 해당 이미지를 포함해 다양한 데이터셋에서 가짜 이미지를 잘 탐지하는 탐지 모델을 구현하는 것을 목표로 합니다.

Task

- **ProGAN**과 **DDPM/DDIM**을 이용하여 가짜 이미지를 생성해보고 다양한 데이터셋에서 이를 진짜인지 가짜인지 판별해보는 것을 목표로 합니다.
- 우선 Face Image를 generation하는 Task를 수행합니다.
  - Generation에는 ProGAN 및 DDPM/DDIM을 사용합니다.
  - Train data로는 Celeb-A, FFHQ를 활용합니다.
  - Metric: LPIPS, FID
- ProGAN에서 학습된 Discriminator 구조를 재사용하여 Deepfake Detection으로 활용합니다.

High-level Architecture: Progressive Growing GAN

- (1) Progressive Growing Generator
  - 1단계: 4x4 -> 8x8
  - 2단계: 8x8 -> 16x16
  - 3단계: 16x16 -> 32x32
  - 4단계: 32x32 -> 64x64

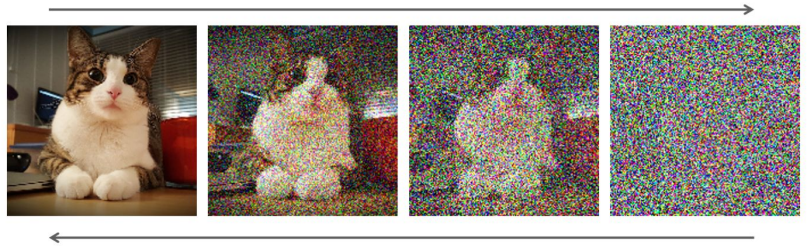


- (2) Fade-In 메커니즘을 적용하여 resolution이 증가할 때 모델이 안정적으로 적응할 수 있도록 합니다.
- (3) Discriminator는 고해상도 입력을 받아 점차 해상도를 감소시키며 step 수에 따라 resolution을 점진적으로 조절합니다.

OUTTA

High-level Architecture: DDPM/DDIM

- (1) Forward Process
  - 1단계: 원본 이미지 x_0에서 타임스텝 t에 따라 점진적으로 가우시안 노이즈 추가
  - 2단계: 최종적으로 순수 노이즈 x_T생성
  - 3단계: 각 스텝에서 β_t스케줄에 따라 노이즈 강도 조절



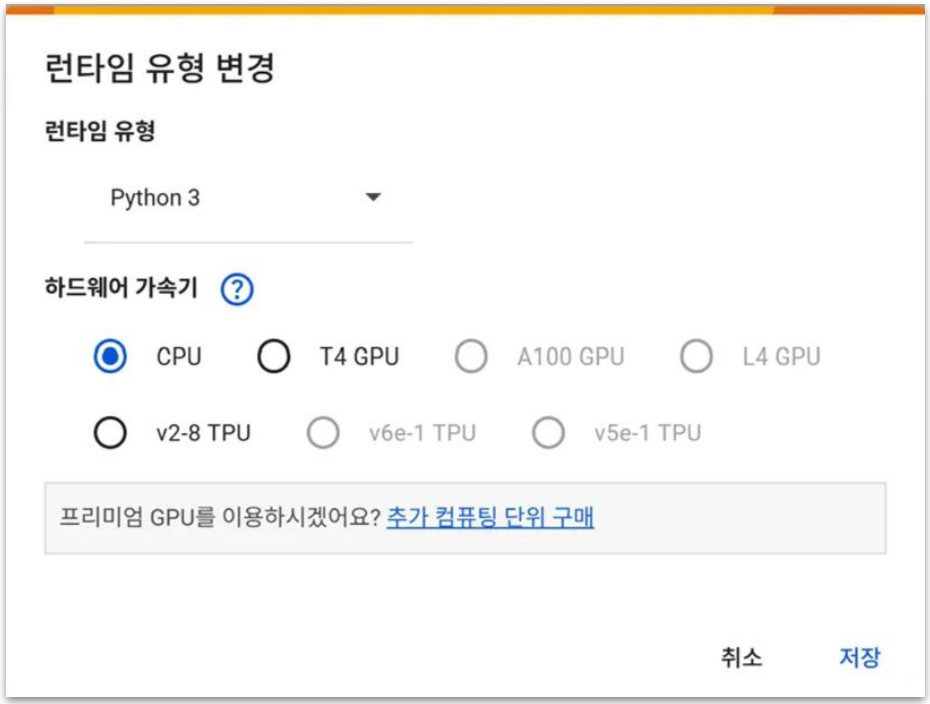
- (2) Reverse Process
  - 1단계: 순수 노이즈 x_T에서 타임스텝 t에 따라 점진적으로 노이즈 제거
  - 2단계: 학습된 모델을 통해 각 스텝의 평균 추정
  - 3단계: 최종적으로 원본 이미지 x_0복원
- (3) 샘플링 방식은 DDIM을 활용하여 더 적은 스텝에 이미지를 생성할 수 있도록 합니다.

OUTTA

Preliminary setup: GPU Usage & Crawling

Google Colab 사용법 (T4 GPU)

- Colab 소개: 클라우드 기반의 무료 Jupyter 노트북 서비스입니다.
- GPU 활성화
  - 런타임 설정-> 런타임 유형 변경에서 GPU를 선택하여 활성화합니다.
- 사용 제약
  - 무료 버전은 최대 12시간 제공됩니다.
  - 리소스는 유동적으로 제공되며 보장되지 않습니다.



OUTTA

이미지 크롤링 코드 설명

- Bing 검색
  - Bing에서 대상 이미지를 검색합니다.
- 원본 다운로드
  - 대상 이미지를 다운받습니다.
- 얼굴 검출
  - [1]MTCNN을 사용하여 이미지에서 얼굴을 검출합니다.
- 얼굴 크롭
  - 정면 얼굴만 잘라내어 저장합니다.
- 리사이즈
  - 64x 64 크기로 리사이즈하여 최종 저장합니다.

원본 이미지

검출 후
크롭된 이미지

64 x 64로
리사이즈된 이미지

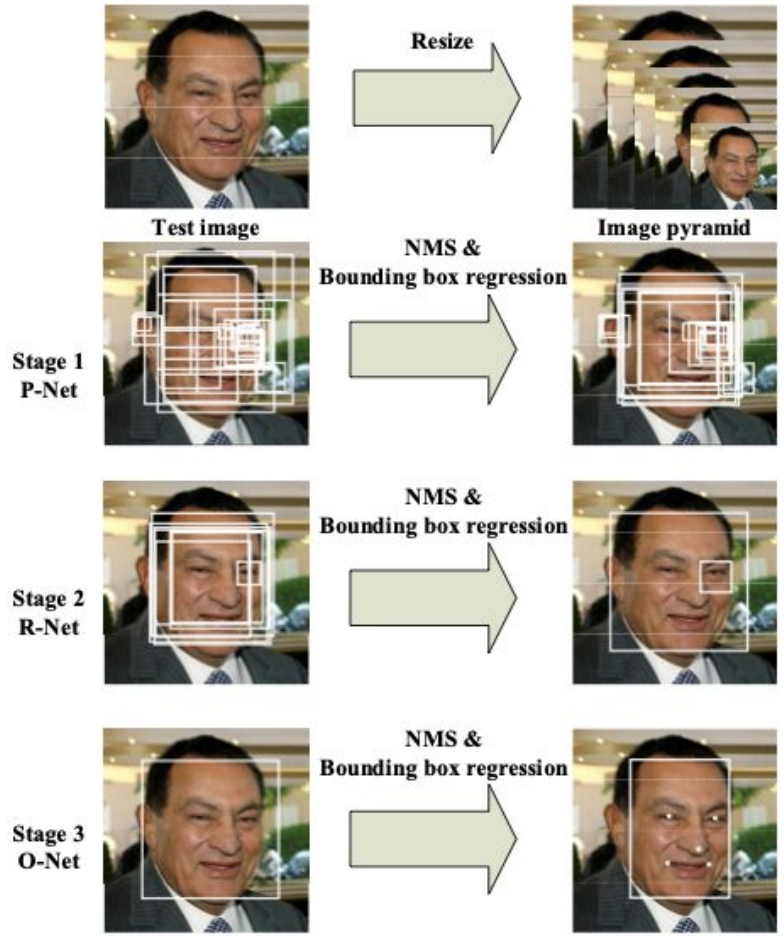
OUTTA

[1]MTCNN(Multi-task Cascaded Convolutional Networks)은 얼굴 검출 및 랜드마크 추출에 널리 사용되는 딥러닝 모델

이미지 크롤링 모델 설명

MTCNN: Multi-task Cascaded Convolutional Networks

- 구조:
  - **P-Net** (Proposal Network): 얼굴 후보 영역을 빠르게 제안
  - **R-Net** (Refine Network): 후보 영역을 정제 및 필터링
  - **O-Net** (Output Network): 최종 얼굴 영역 결정 및 5개 랜드마크 예측
- 장점:
  - 실시간 처리 가능
  - 얼굴 검출과 정렬을 동시에 수행
  - 다양한 얼굴 방향, 표정에 견고함



이미지 크롤링 코드 설명

get_undetectable_driver()

Selenium 드라이버를 초기화합니다.

resize_for_detection()

얼굴 검출을 위해 이미지 크기를 조정합니다.

is_frontal_strict()

이미지 내 얼굴이 정면인지 판단합니다.

download_images_bing()

Bing에서 이미지를 다운로드 합니다.

핵심 파라미터는 MAX_IMAGES_PER_CATEGROY = 15, MIN_FACE_SIZE = 200, TARGET_FACE_SIZE = (224, 224)

입니다.

OUTTA

Details on dataset

OUTTA

Dataset

- Generation을 위해 고해상도 얼굴 이미지 데이터셋을 활용합니다.
- CelebA-HQ
  - https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
- FFHQ
  - https://huggingface.co/datasets/student/FFHQ
- 전체 이미지 수 :
  - Train data: 33,000장

OUTTA

Restriction

- 공정성을 위해 구현 및 학습 과정에서 다음과 같은 제한사항이 존재합니다.
- **(1) 모델의 성능은 FID와 LPIPS Metric으로 평가할 예정입니다.**
  - Metric을 계산하는 코드는 제공되며, 최종 성능을 평가하는 데 사용되는 Private data는 제공되지 않습니다.
- **(2) Random seed 는 42으로 고정되어 있으며, 이 값은 그대로 사용하셔야 합니다.**
  - Seed 고정은 이미 구현되어 있으므로, 추가 구현 및 수정은 불필요합니다.

제출해야 하는 것 [Essential]

- (1) ProGAN
- (2) DDPM
- (3) 모델의 weight 파일
  - epoch_{num}_Dis.pt
  - epoch_{num}_Gen.pt

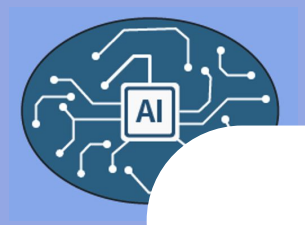
제출해야 하는 것 [Optional]

- (4) Fine Tuning
- (5) 멘토가 제공한 스켈레톤 코드가 아닌 본인들의 조에서 직접 제작한 모델
  - 모델을 만들 때 사용한 모든 코드
  - 가장 좋은 성능의 weight - 간단한 설명서 (5 page 내외)
- (6) 그 외 자체적으로 추가 및 수정한 코드 전부

유의 사항

- 과제 제출 마감일 : 8월 21일 오전 11시 59분
    - 다음의 메일로 과제를 제출 받을 것입니다. : admin@outta.ai
- <유의 사항>
    - 제출물 중 가장 높은 점수를 받은 제출물로 최종 평가를 진행합니다.
    - 평가는 구현(40), 성능(60)입니다.

OUTTA



감사합니다

OUTTA