

FACE VERIFICATION

Whocares Korea

코드스테이츠 AI 한종현, 김혜수

벡터의 EUCLIDEAN DISTANCE를 이용한 본인 확인 알고리즘 구현 프로젝트

Contents

01

프로젝트
개요

02

프로젝트
팀 구성 및 역할

03

수행 절차 및
수행 방법

04

시연 영상

05

자체 평가 의견

프로젝트 개요

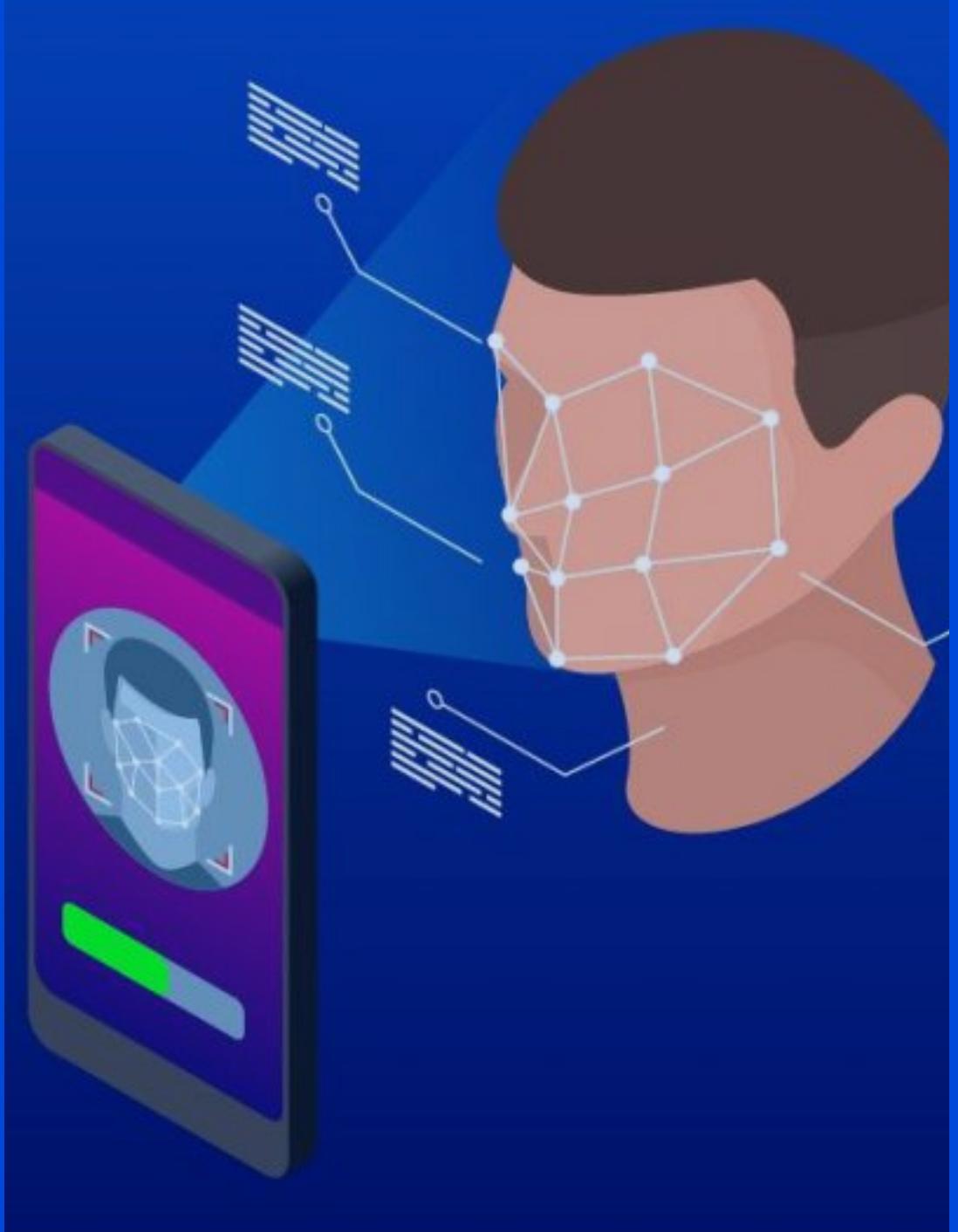
프로젝트 주제 및 선정 배경

개발 환경

프로젝트 개요

프로젝트 구조

01



프로젝트 주제 및 선정 배경



만남 앱에서 남의 얼굴 혹은 닮지 않은 얼굴을 올려 앱의 신뢰도가 떨어지는 일을 방지하고자 프로젝트를 진행

구성	번호
개발 환경	1
프로젝트 개요	2
프로젝트 구조	3

개발 환경

OS: Window 10 pro

개발 언어: Python 3.7

라이브러리: dlib, numpy, opencv, glob, math, tensorflow.compat.v1

프로젝트 개요

등록된 프로필 사진과 진위 확인을 위한 셀프 카메라 사진을 facenet 모델을 활용하여 벡터화 한 후 Euclidean distance 를 구해서 유사도를 구하고, 한계점을 이용해 프로필 사진의 얼굴이 본인인지 확인한다.

프로젝트 구조

1. 이미지 경로를 받아서 프로필 이미지 로드
2. 얼굴 쉐입(68 landmarks) 찾아 전체 사진 속 얼굴 위치 찾음
3. 찾은 얼굴을 자르고 회전 시킨 후 facenet모델에 삽입 후 벡터 추출
4. 셀프 카메라 이미지 1~3 과정 동일하게 진행
5. 프로필, 셀프 카메라 이미지 벡터 거리 비교 하며 0.95 이하 일 경우 동일인으로 판단



02 프로젝트 팀 구성 및 역할

팀 구성 안면인식팀

팀 역할 한종현

- 모델 이식과 이미지 로드, 크롭, alignment 기능 구현
- 모델 서빙 테스팅 진행

김혜수

- 전체적인 파이프라인을 만들어 모델을 연결
- mtcnn 기능 구현

03

프로젝트 수행 절차 및 결과

수행 절차

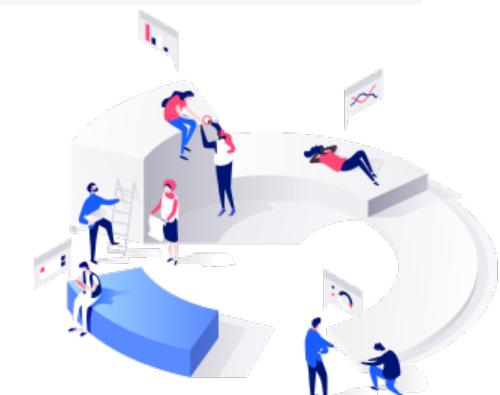
모델 선정 및 개요

프로젝트 수행 방법



프로젝트 수행 절차

구분	기간	활동	비고
사전 기획	22.05.16(월) ~ 22.05.20(금)	<ul style="list-style-type: none">- 프로젝트 팀배정- 기획안, 1주차 주간보고서 작성- 얼굴 인식 논문 서치 및 타사 앱 얼굴 인식 기능 자료조사- 사용할 모델: facenet 선정- 모델 파이프라인 구축- 이미지 저장 경로 확인 후 코드 초안 작성	<ul style="list-style-type: none">- 팀별 주간보고 실시- 기업협업 진행 관련 오프라인 미팅(17일)- 백엔드 개발자 기존 코드 관련 줌미팅(19일)
모델링	22.05.23(월) ~ 22.05.27(금)	<ul style="list-style-type: none">- 2주차 주간보고서 작성- Facenet 논문 분석 및 모델링 진행- 모델 성능 테스트 후 정확도 개선- 모델 정확도를 높이기 위한 정렬 기능 추가	<ul style="list-style-type: none">- 팀별 주간 보고 실시- 모델링 진행 관련 오프라인 미팅(23일)- 진행 사항 보고 관련 줌미팅(26일)
서비스 구축	22.05.30(월) ~ 22.05.31(화)	<ul style="list-style-type: none">- 백엔드 팀과 협업하여 url로 받은 이미지 모델과 연결- 최종 보고서 작성	<ul style="list-style-type: none">- 최종 보고 실시- 모델 서빙을 위한 오프라인 미팅(30일, 31일)
총 개발 기간	22.05.16(월) ~ 22.05.31(화) 총 3주		



프로젝트 수행 방법



01. 모델 선정 및 개요

FaceNet

각각의 얼굴 이미지를 128차원으로 임베딩하여 유클리드 공간에서 이미지간의 거리를 통해 분류하는 모델

Triplet loss

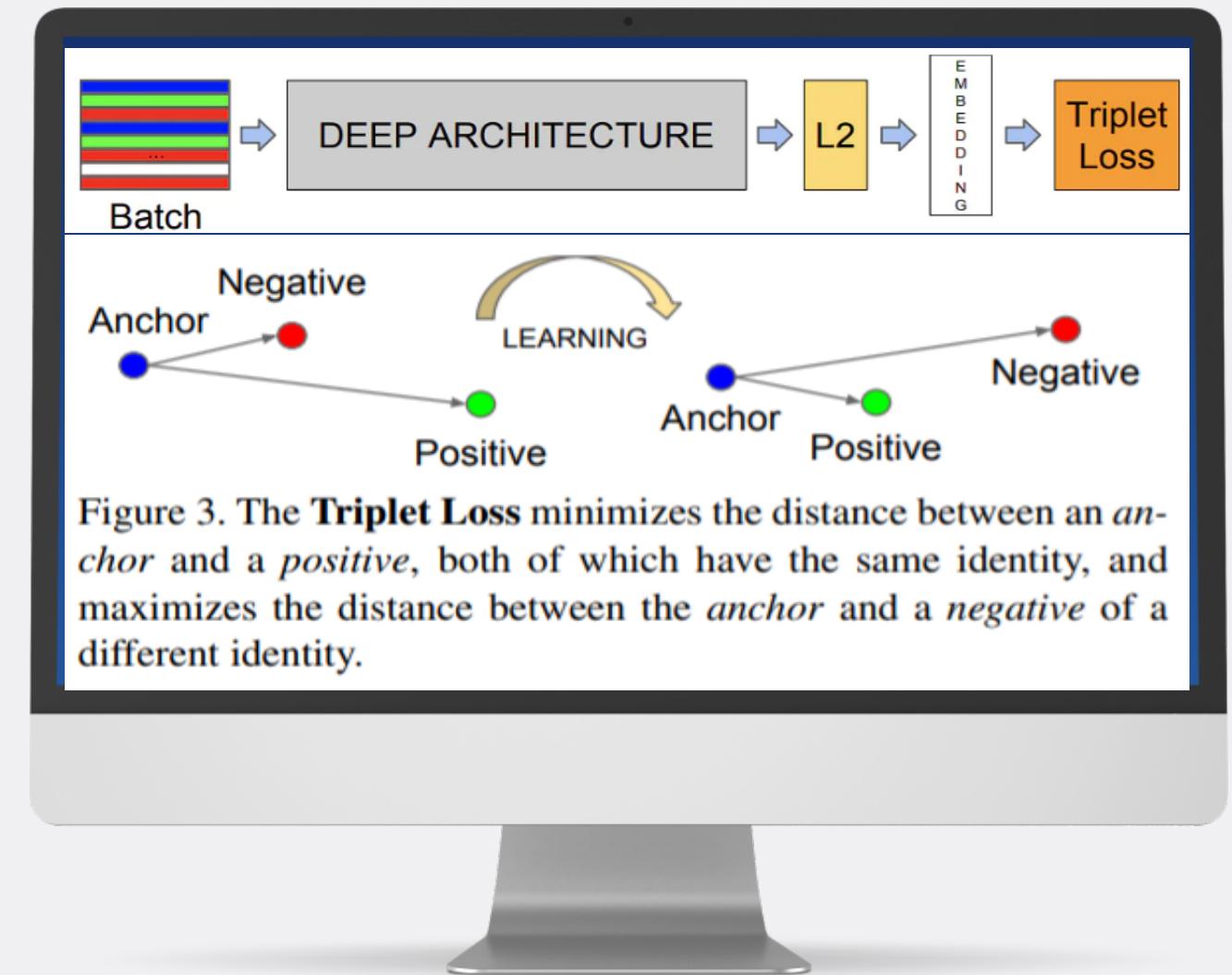
triplet은 2개의 일치하는 얼굴과 하나의 다른 얼굴로 이루어져 있으며 loss는 distance margin으로 일치하는 쪽을 불일치하는 쪽으로부터 분리

Face verification

유클리드 공간에 직접 임베딩 하는 학습 방법으로 진행해 Face verification이 가능

Alignment

Facenet은 얼굴 영역을 정확히 자르는게 중요하기 때문에 커스텀 alignment 진행 해 성능을 향상 시킴



02.

얼굴인식 진행 과정

Load images & get shape of face

프로필 사진과 셀프 카메라 사진의 얼굴 이미지를 `dlib.get_frontal_face_detector`를 이용하여 크로핑 한 후 `68 landmark model` 모델을 이용해 각 얼굴들의 shape 정보를 얻습니다.

Image rotation

Shape 정보에서 눈 사이의 좌표값을 꺼내온 두 좌표에 직각 삼각형을 그립니다. 코사인 법칙을 활용하여 삼각형의 각도를 알아낸 후 기울어진 각도와 방향을 구합니다. 이렇게 얻은 각도와 방향을 이용해 `rotate matrix`를 구하고 이미지에 적용해서 적절하게 이미지를 회전시킵니다.

01

02

03

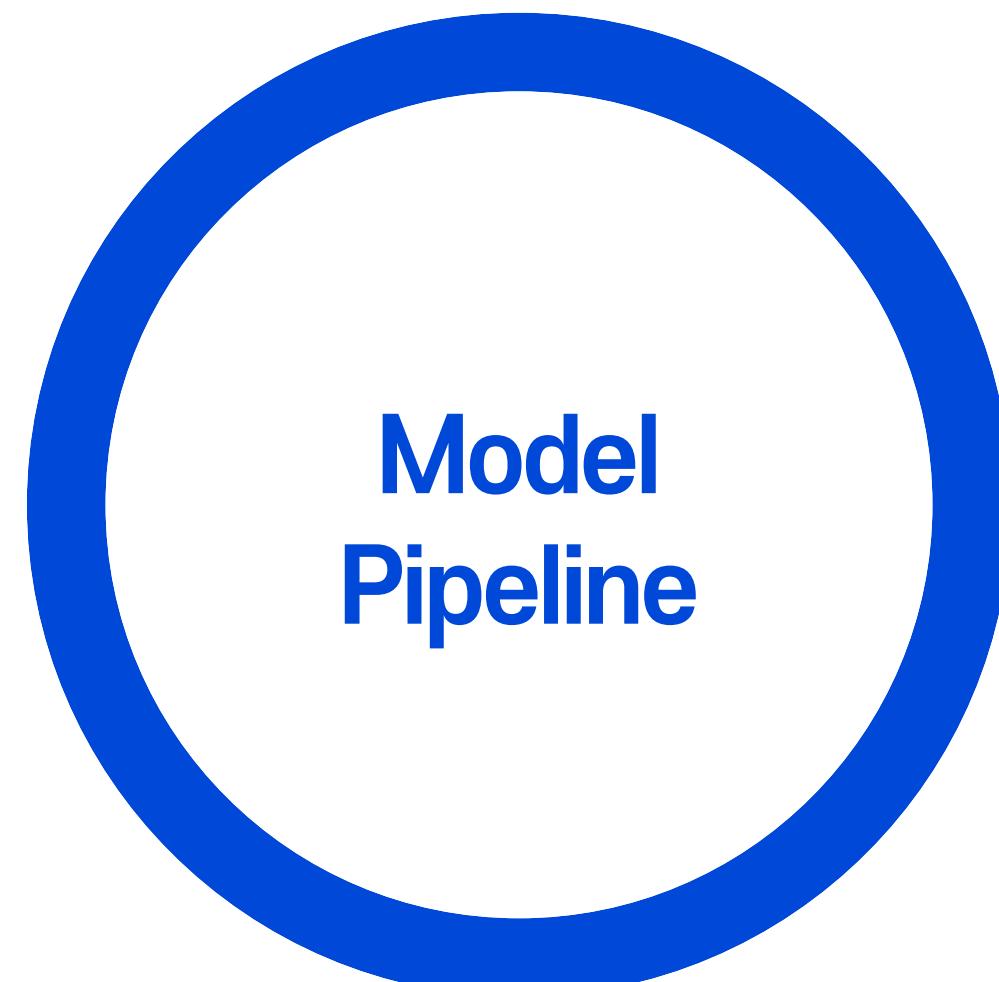
04

Identify individual face

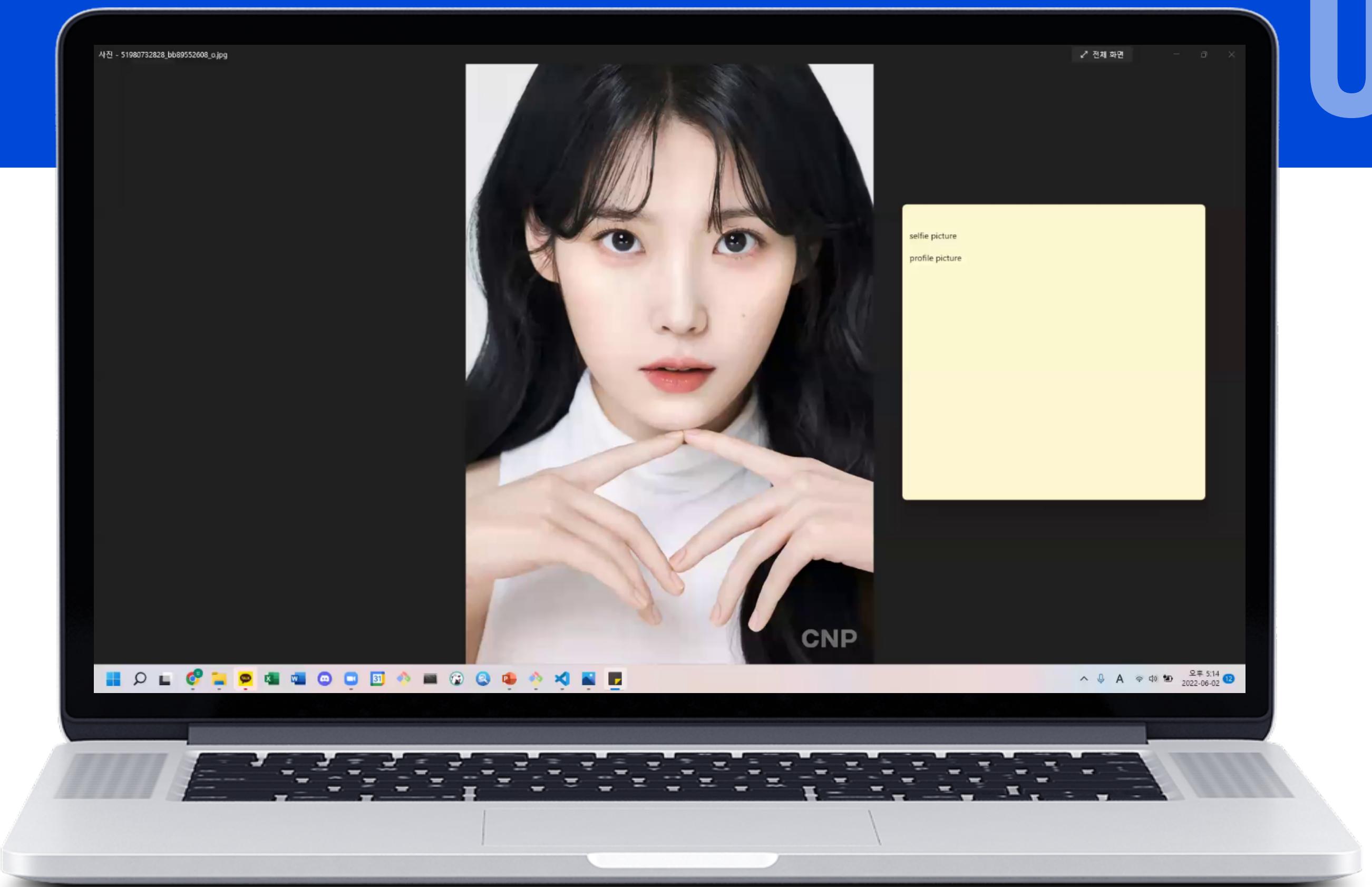
회전시킨 이미지들을 변형된 Facenet 모델에 넣어 512 차원의 텐서를 구한 후 $(n, 1, 512)$ numpy 형태로 변경해 프로필, 셀프 이미지 간의 결과들을 유클리디안 거리를 통해 유사도를 구합니다.

Criterion of face verification

약 0.95 이하의 유사도를 가지면 본인이라고 판단합니다.



04 프로젝트 시연영상



YouTube



QR코드를 스캔하면
유튜브에서 확인 가능합니다.

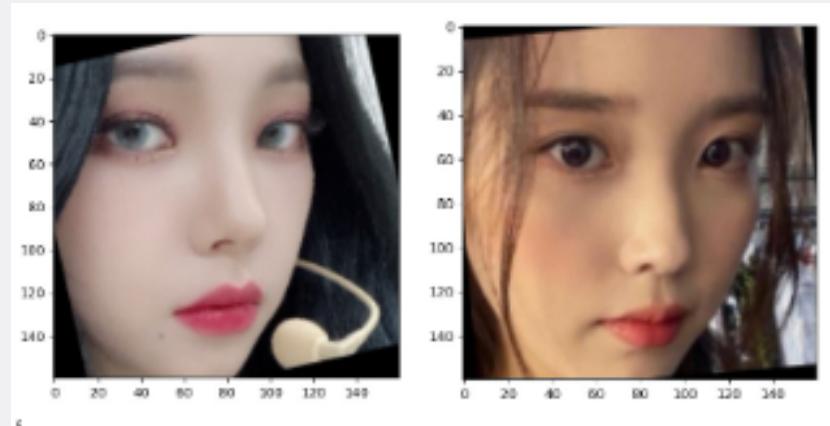
프로젝트 기대효과

추후

개선 사항

현재 모델이 LFW 데이터 기준 0.9965의 높은 정확도를 보이지만, 동양인의 얼굴 학습이 부족해 분위기가 유사한 동양인들을 동일인이라고 판단하는 경우가 있습니다.

향후 딥러닝에 적절한 컴퓨팅 환경을 구성하여 동양인의 얼굴을 재학습 시켜 모델을 개선 시 더 정확하게 구분이 가능할 것이라고 예상됩니다.

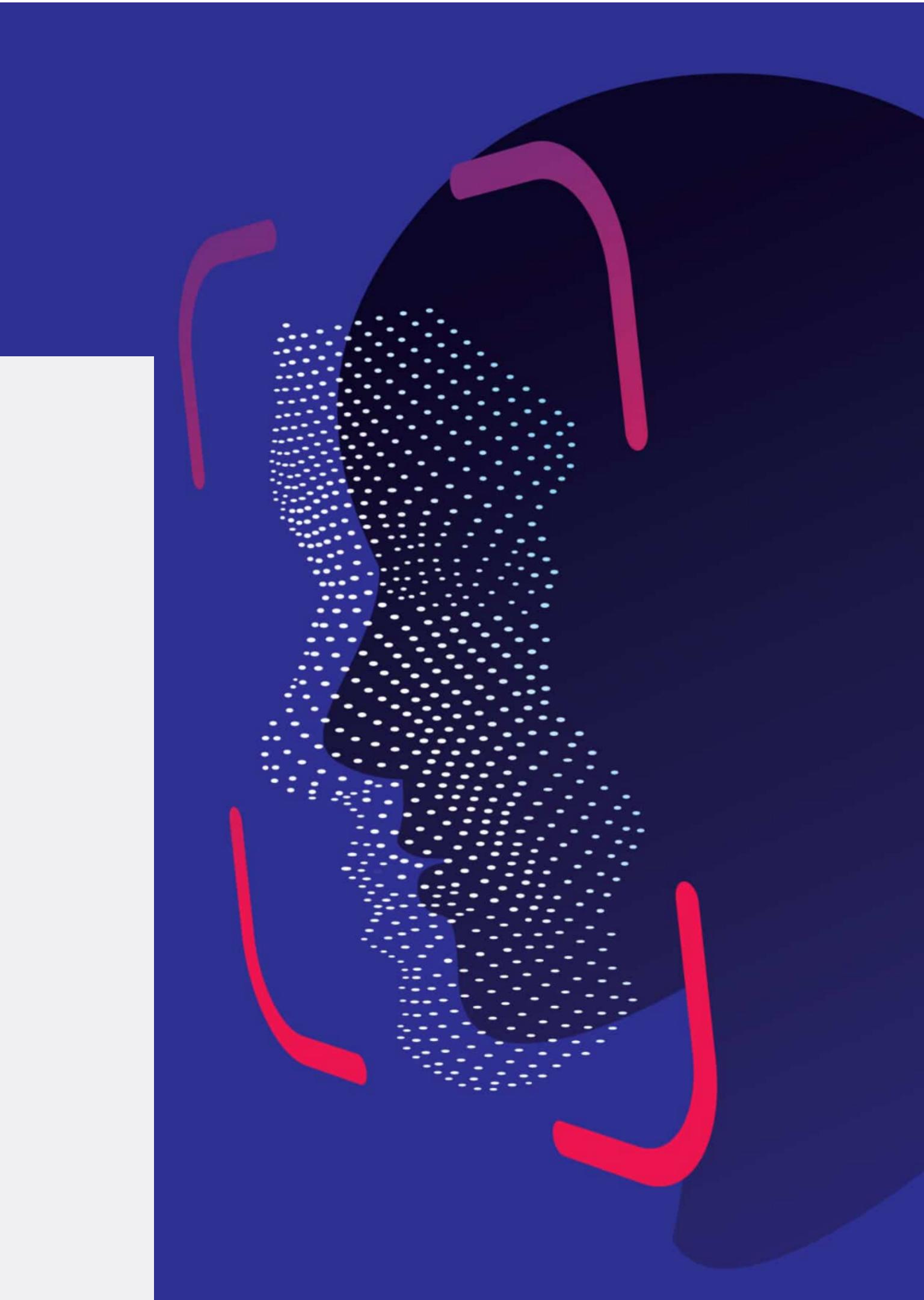


05

기대효과

facenet 모델의 얼굴 인식 알고리즘을 이용해 프로필 이미지와 다른 유저를 식별이 가능해 앱 신뢰도를 높여줍니다.

EXPECTED EFFECT
OF PROJECT



THANK YOU

Reference

<https://github.com/davidsandberg/facenet>