



조은진

iOS/AOS/AI 개발자

생년월일 1991.09.19

연락처 01027036630

이메일 jintazcho@gmail.com

희망연봉 5500만원

학력 Ursula Frayne Catholic College
RMIT University School of Engineering (중퇴)

이력 팀노바 수료
한국인공지능학회 논문 등재
Facebook Deepfake Detection Challenge 참여
Koiware 카드인식/신분증 인식 솔루션 개발

사용기술

Language

Java
Python
C++
PHP
Swift
Objective-C

Database

SQLite
MariaDB
MySQL

Platform

Android
Linux
Windows
iOS

Server/Framework

Wowza Media Server
Apache
Tensorflow
Pytorch

Protocol

RTSP
HTTP
TCP/IP

Learning Model / Algorithm

Object Detection(Faster R-CNN, SSD)
BEGAN, StarGAN
MWGAN, ALAE, StyleGAN , ProgressiveGAN
VAEGAN, HoloGAN, WGAN
DCGAN, LSGAN, SAGAN, CasualGAN
Pix2pix, Pix2pix-HD, FFTGAN

Library / API

OpenCV, MTCNN
Dlib, Numpy, Pillow
Pandas,
Transformers
Torchvision, Volley,
Picasso, Glide
Gson, SQLite,
ExoPlayer,
Scikit-Learn

포트폴리오

오픈소스 프로젝트

- MARS (Make Artificial Realistic Objects)

논문

- Deepfake detection model based on fake attributes shown in images/videos
(이미지, 동영상에서 나타나는 가짜 특징에 기반한 딥페이크 탐지모델)

안드로이드 어플리케이션

- Jinstagram

PHP 프로젝트

- JInside

프로젝트 소개

프로젝트 목적

- 저작권에 구매받지 않는 모든 객체 데이터셋을 제공하기 위한 오픈소스 프로젝트 입니다.

사용기술

- 라이브러리 OpenCV, Dlib
- 알고리즘 Pix2PixHD, Pix2Pix
- 프레임워크 Tensorflow, Pytorch

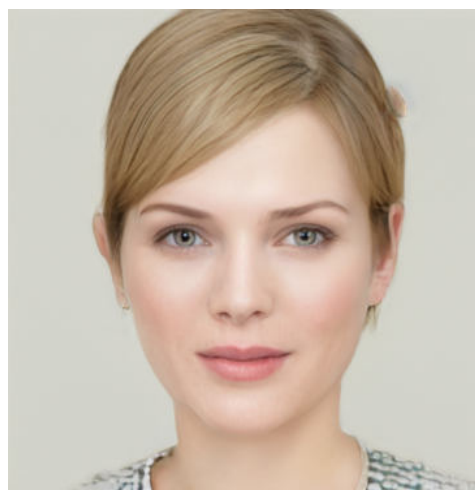
참여인원 3명

역할 및 기여도

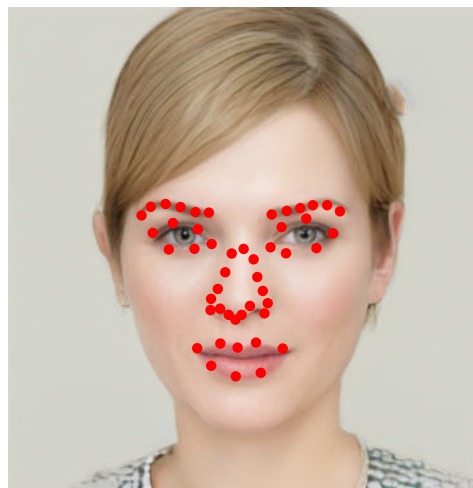
- 역할 팀장
- 기여도 40% (얼굴 이미지 생성 및 팀 리딩)



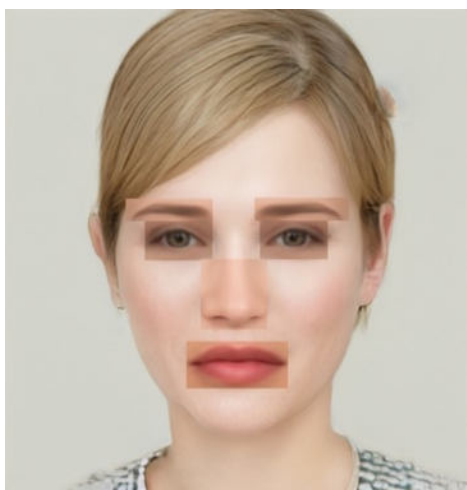
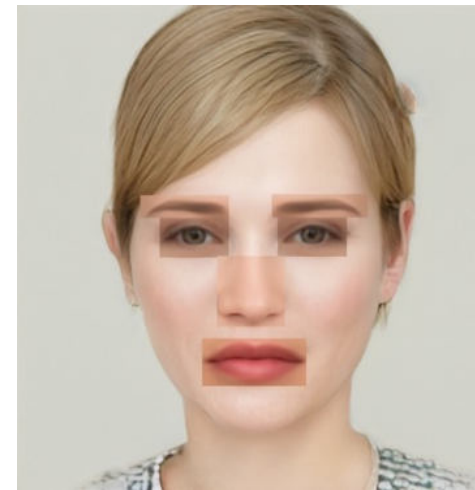
얼굴 이미지 생성 모델구조



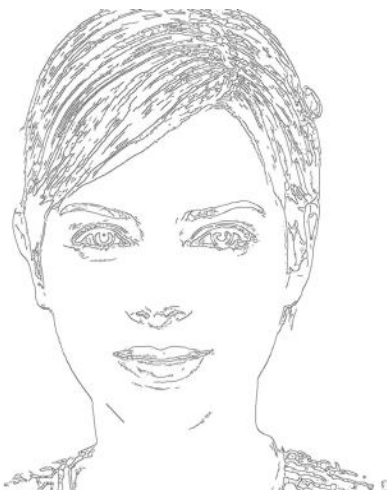
Dlib



눈, 눈썹,
코, 입



OpenCV
Filter



Pix2Pix
HD



논문소개

1. 연구 배경

- SNS 및 인터넷에 딥페이크 영상 증가
- 딥페이크 생성 알고리즘 다양화

2. 연구 목적

- 다양한 딥페이크 알고리즘을 탐지할 수 있는 모델 구축
- 고사양의 GPU가 아니더라도 탐지 모델 구축

3. 사용기술

- 가짜얼굴 생성 알고리즘(GAN)
- 가짜얼굴 탐지 알고리즘
 - Faster R-CNN, SSD
- 가짜얼굴 판별 알고리즘
 - BERT(Transformer)

4. 참여인원 및 기여

- 참여인원 2명
- 기여도 50%
 - 학습용 가짜얼굴 생성 [[구글드라이브](#)]
 - 영어논문 작성



학습용 가짜얼굴 생성

개요

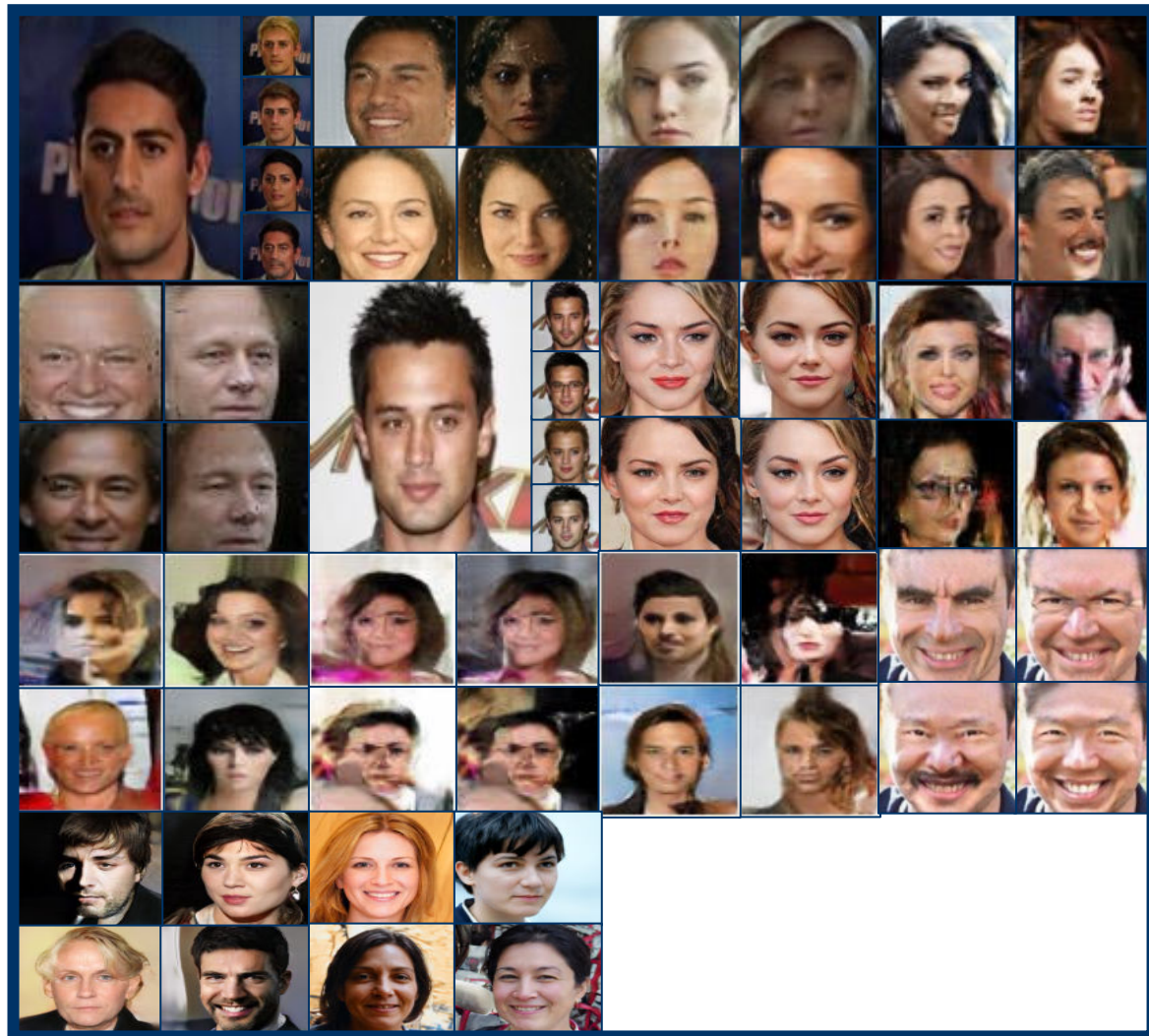
구현모델 StyleGAN, ALAE, ProgressiveGAN, FFT-GAN, MW-GAN, StarGAN, CasualGAN, VAE-GAN, HoloGAN SAGAN, BEGAN, WGAN, LSGAN, DCGAN

사용언어 Python

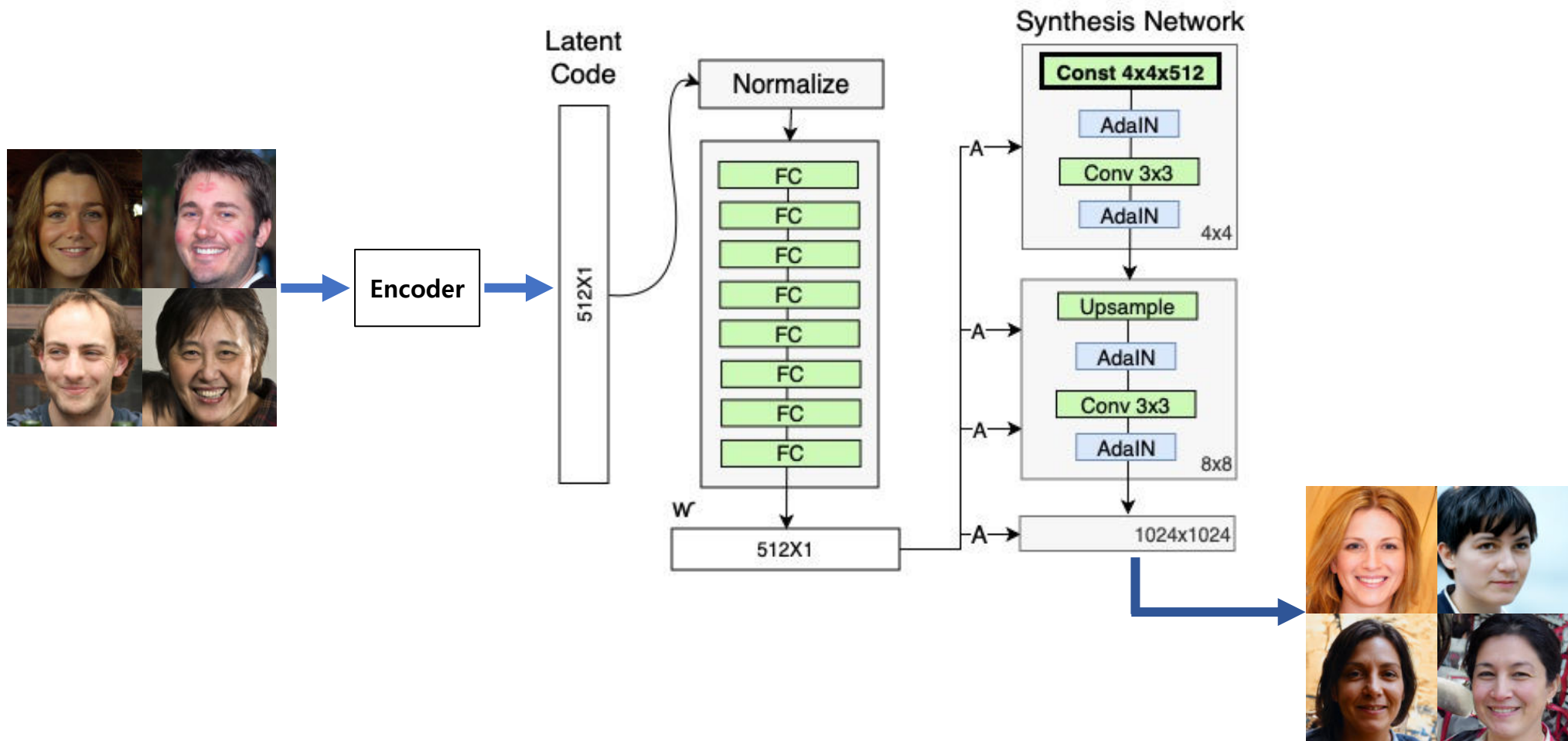
프레임워크 Tensorflow, Pytorch

개발기간 2weeks

학습데이터 CelebA, VGG-Face 2, CelebA-HQ, FFHQ

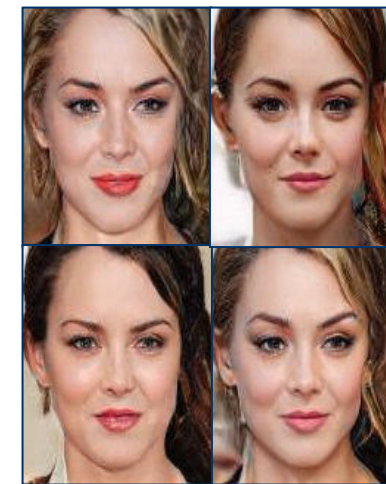
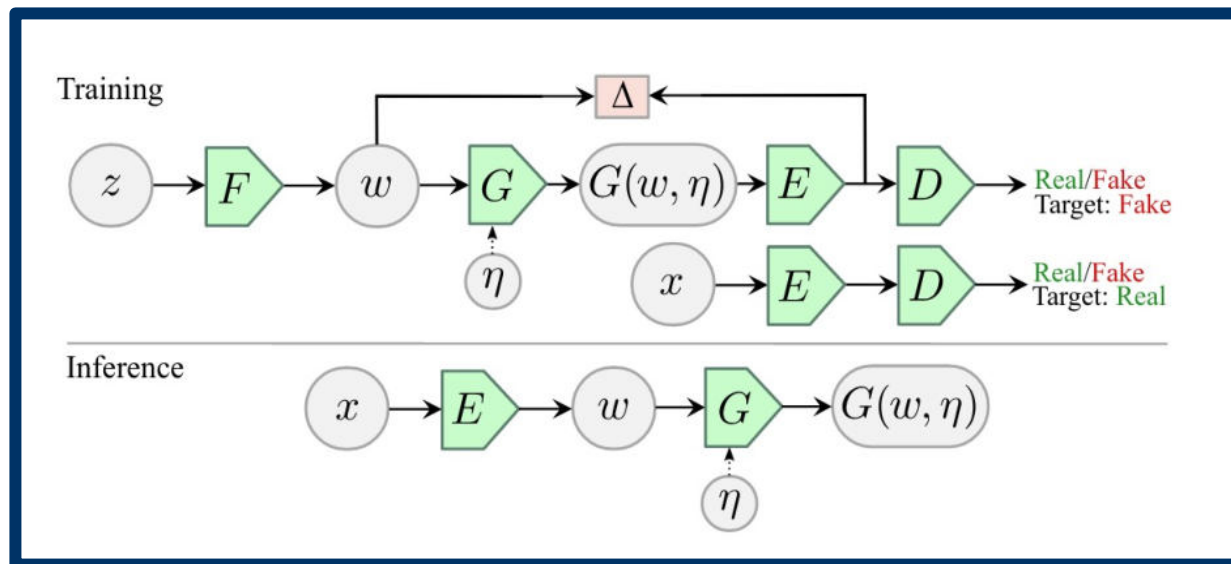
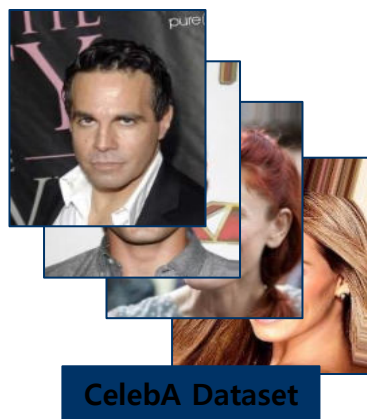


StyleGAN 구조



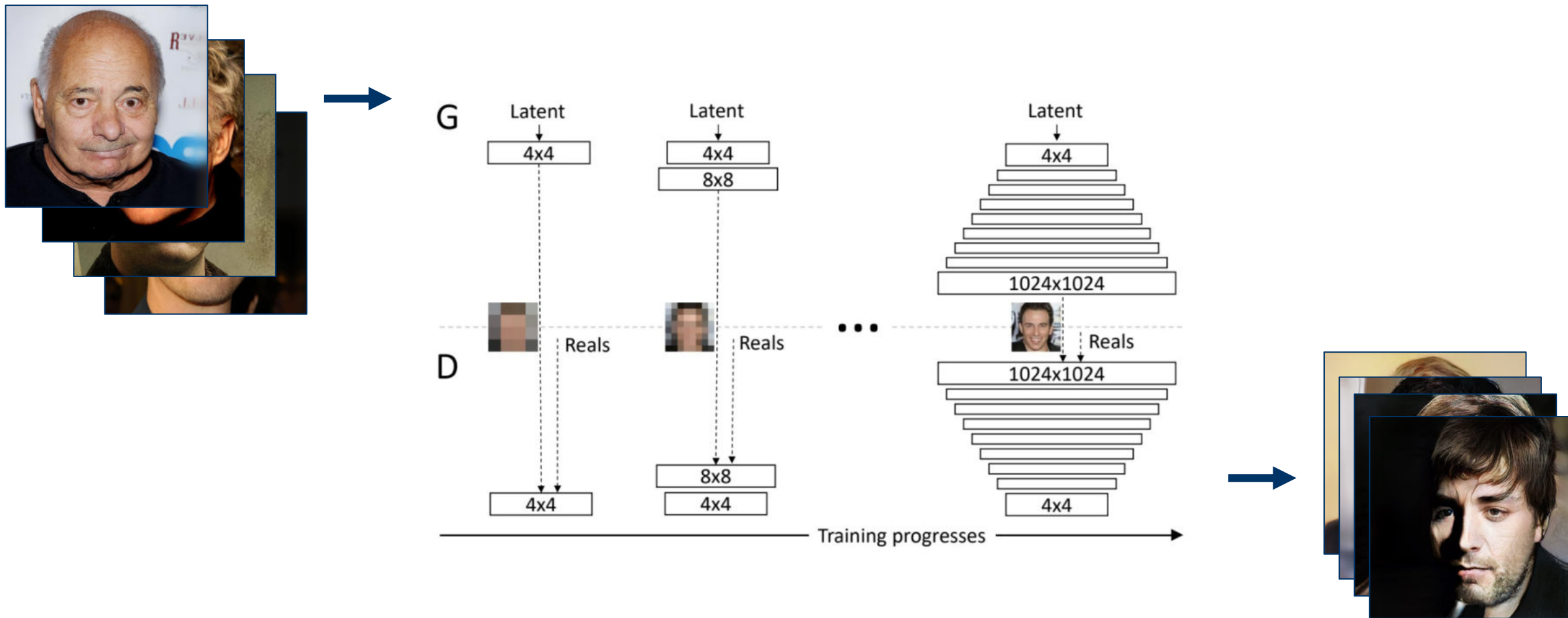
잠재공간이 Normalize 된 이후로 신경망을 거쳐 추출된 가중치를 각 합성네트워크에 하나의 스타일의 형태로 학습된다.

ALAE 구조



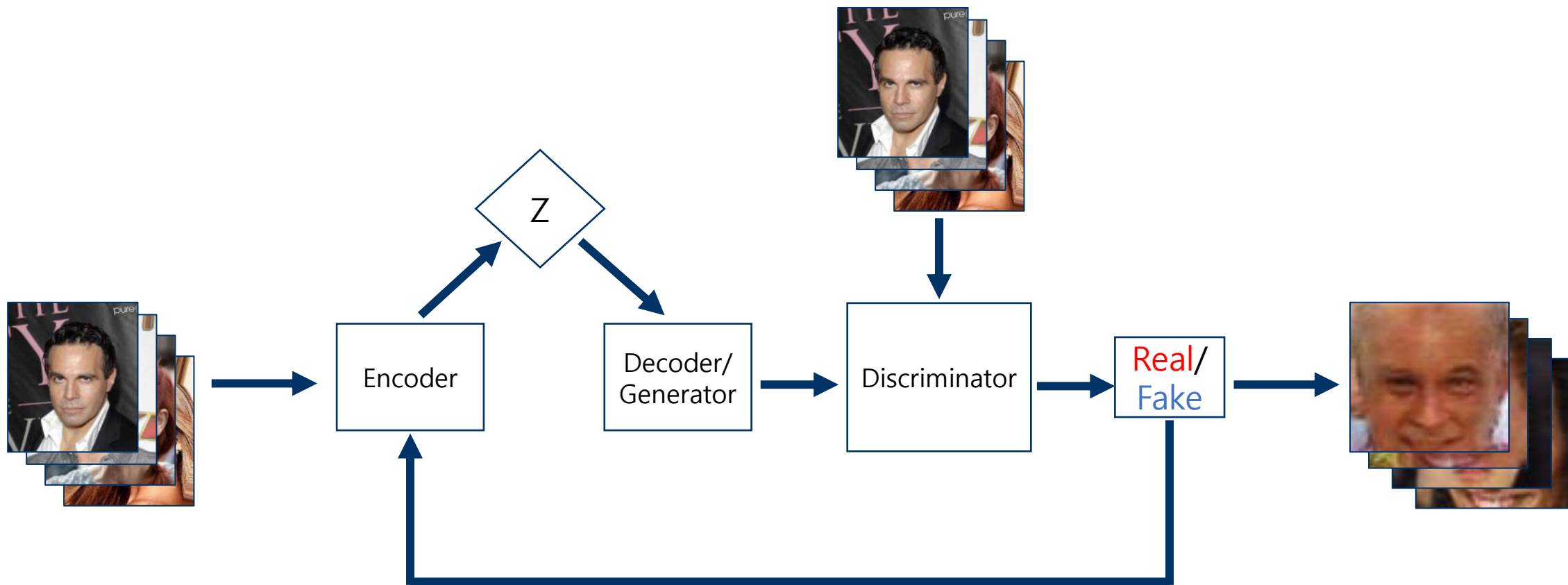
인코더와 디코더(제너레이터)는 각각 $F \circ G$ 그리고 $E \circ D$ 로 나뉜다. F 는 추출한 잠재공간으로부터 가중치를 추출한다 그리고 G 는 이미지를 합성하기 위한 레이블 값을 가져온다. E 는 이렇게 합성된 이미지를 다시 인코딩하고 이 값을 앞서 추출한 가중치와의 간격을 최소화 한다 이를 델타값 이라고 한다. Discriminator는 진짜/가짜를 판별하는데 이는 원본 이미지 x 를 인코딩 했을 때의 값과 앞서 추출했던 델타값의 간격을 토대로 판별을 한다.

ProgressiveGAN 구조



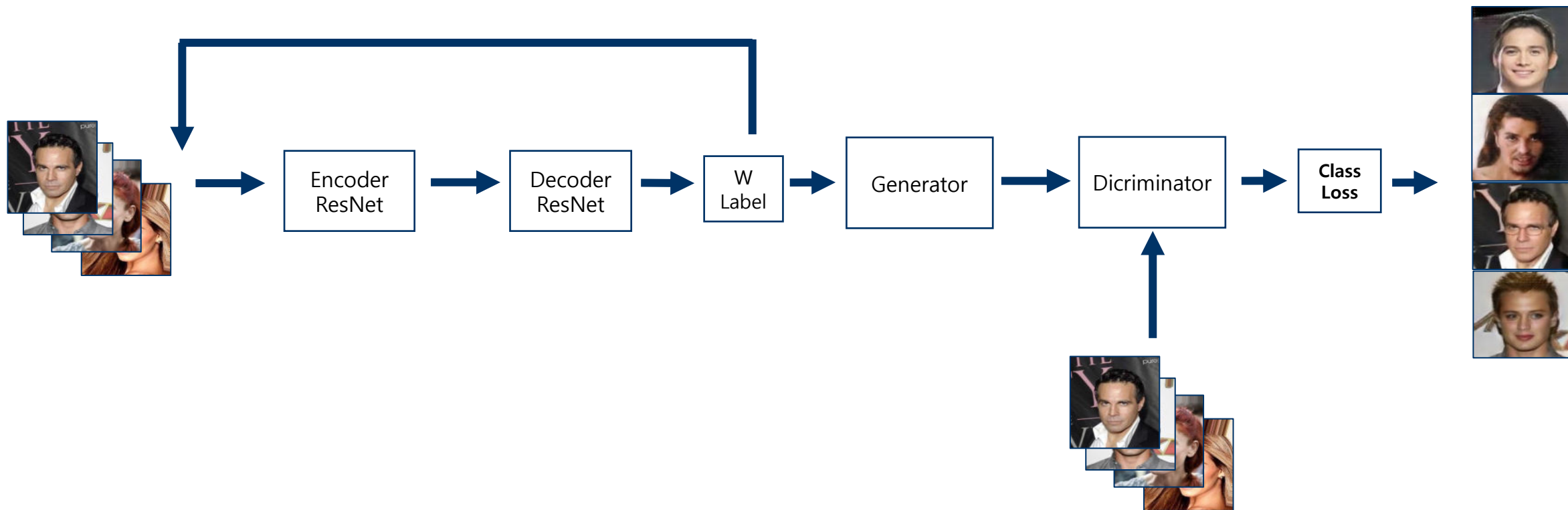
낮은 해상도의 이미지로부터 고해상도의 이미지 합성까지 순차적으로 학습한다.

VAE-GAN 구조



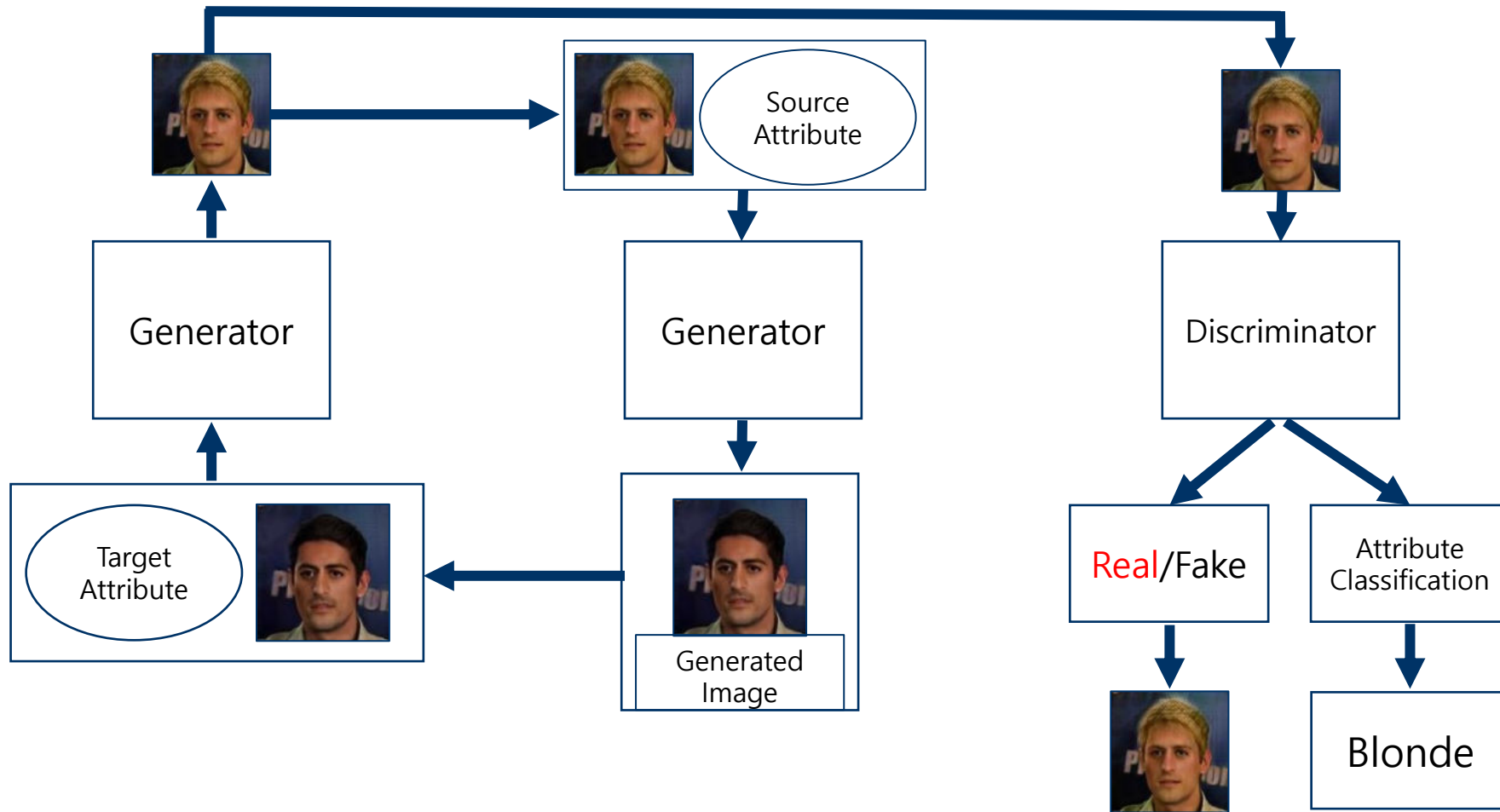
Encoder는 원본 이미지로부터 잠재공간(Z)을 추출한다 그리고 디코더는 generator의 역할을 수행한다. Discriminator는 생성된 이미지의 진짜/가짜를 판별한다.

MW-GAN 구조



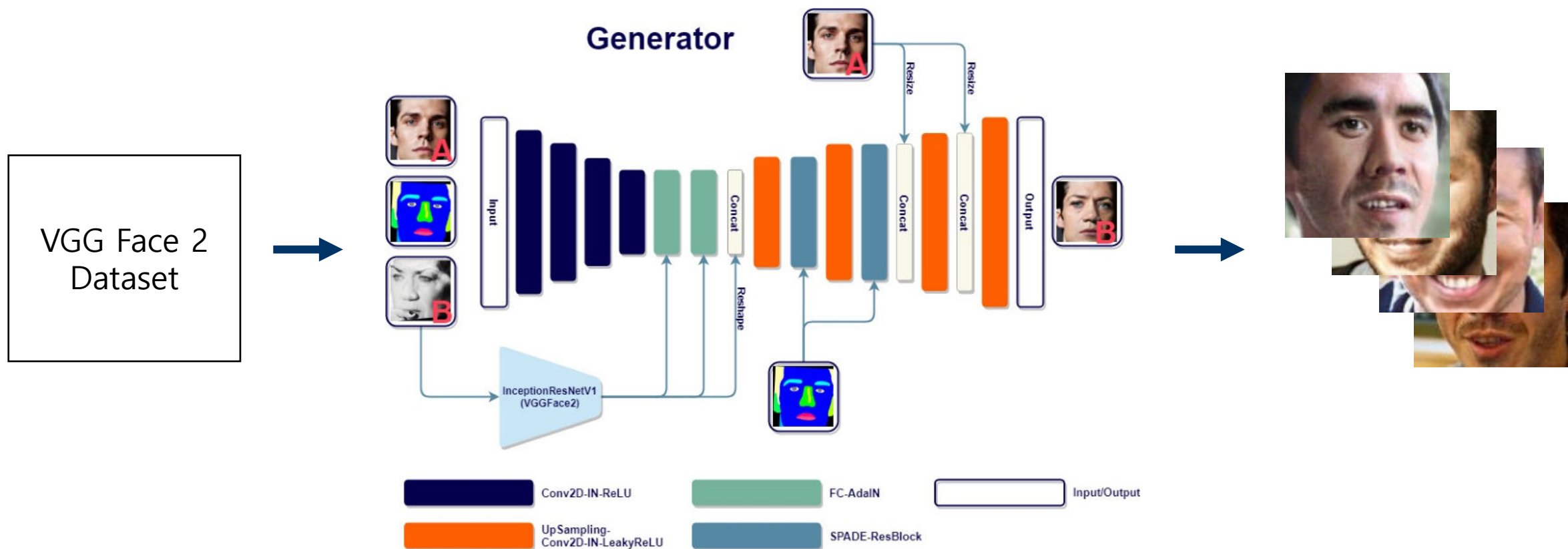
원본 이미지는 인코딩/디코딩 과정을 통해 레이블 값과 W값을 얻게 된다. 이러한 값들을 토대로 이미지를 생성하고 Discriminator가 생성된 이미지를 각각의 Classification loss를 토대로 진짜/가짜를 판별하는데 이는 각 레이블별 원본/가짜 이미지의 Wasserstein distance loss를 의미한다.

StarGAN 구조

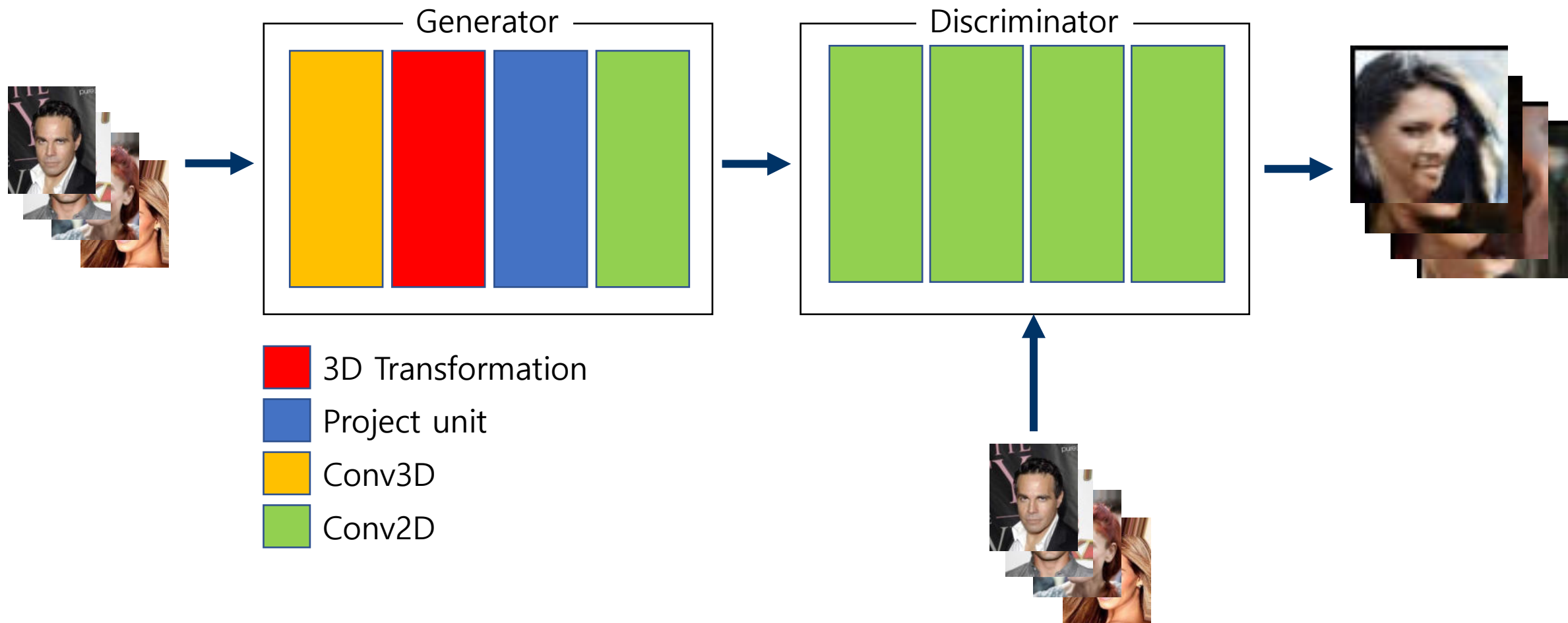


원본 이미지는 다양한 특징이 적용된 얼굴 이미지를 생성하기 위해 재활용 된다.

FFT-GAN 구조

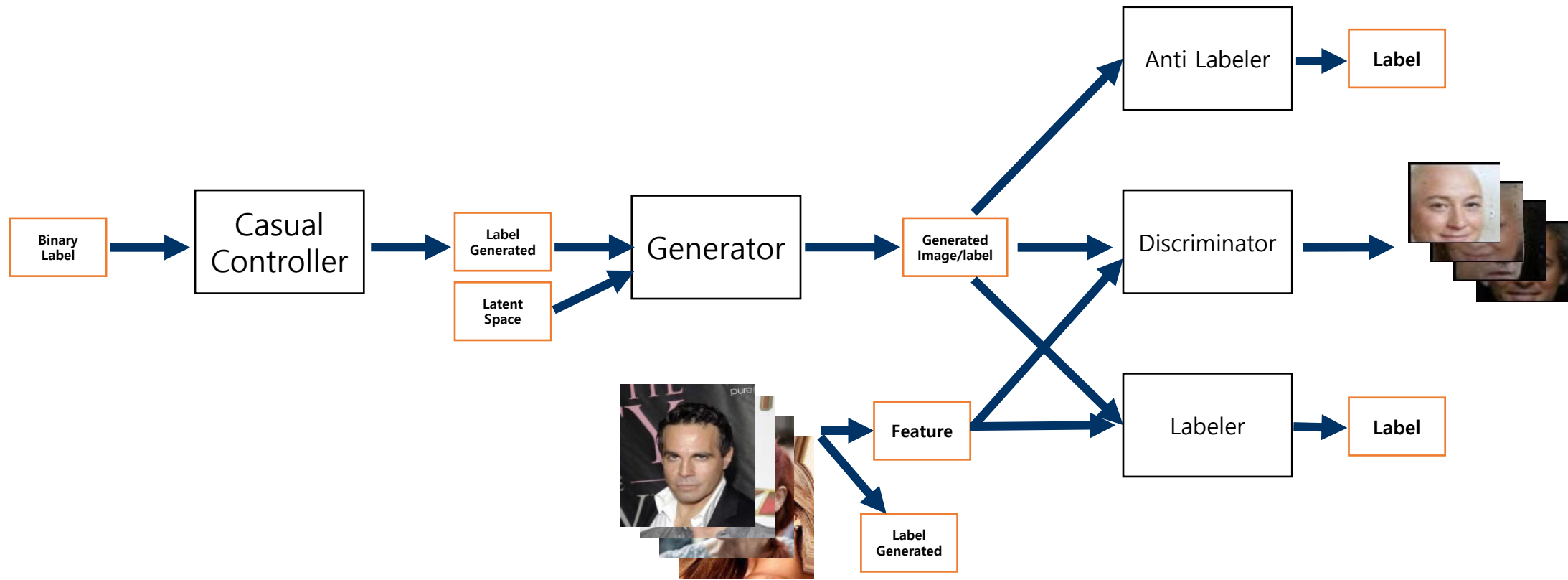


HOLOGAN 구조



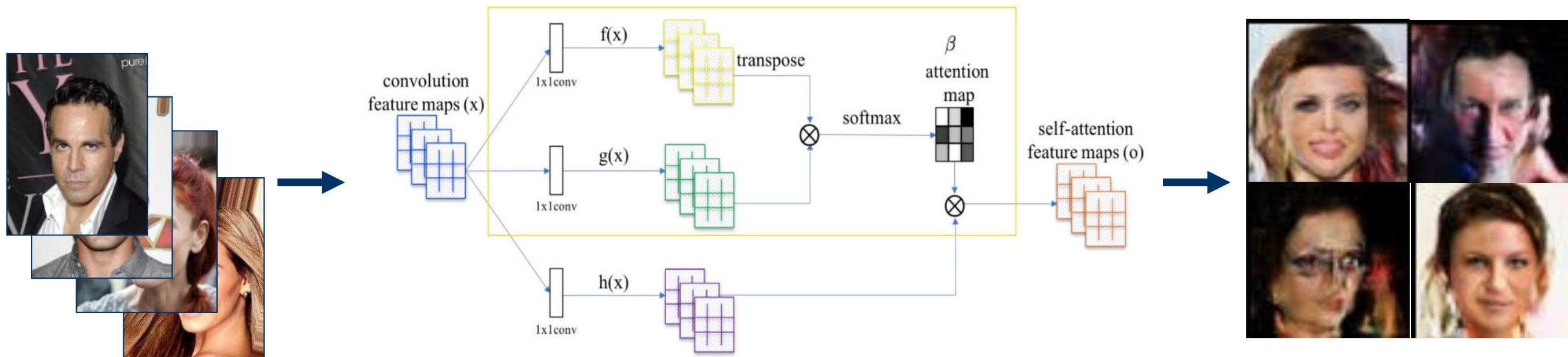
Generator에 3D transformation layer 그리고 project unit을 3D conv layer 그리고 2D conv layer사이에 배치한다. 이는 원본 얼굴데이터를 다른 방향을 바라보고 있는 상태로 재구성하기 위함이다.

CasualGAN 구조



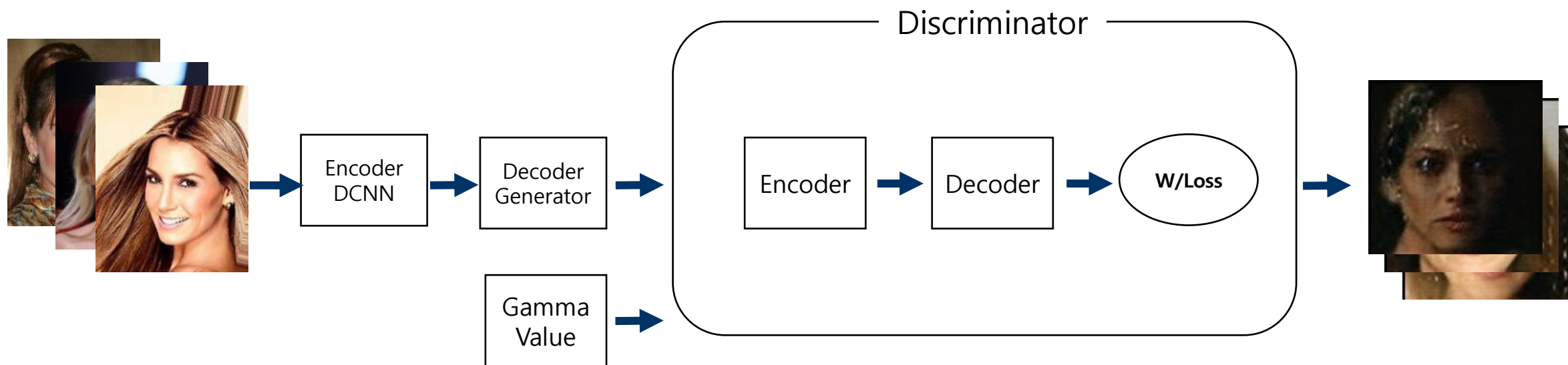
Casual controller는 label을 생성하고 generator 그 label과 latent space를 가지고 이미지를 생성한다. Discriminator는 생성된 이미지의 진짜/가짜 여부를 판별한다. 그리고 생성된 이미지의 레이블은 labeler 그리고 anti-labeler 전달된다. Labeler는 보다 정확한 이미지 라벨링을 하기위해 원본 이미지로부터 label값을 받아온다.

SAGAN 구조



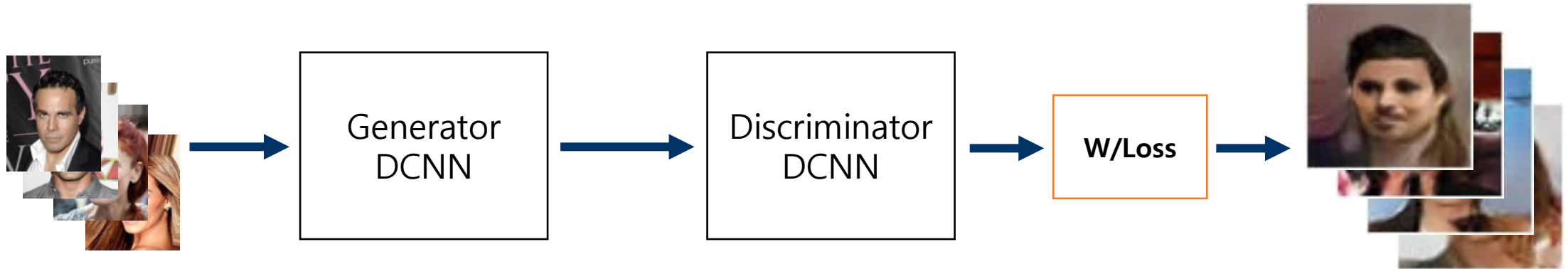
각 1×1 convolution map은 각 영역별로 특징을 보다 상세하게 추출하고 feature map을 형성하는데 이를 attention map이라 한다.

BEGAN 구조



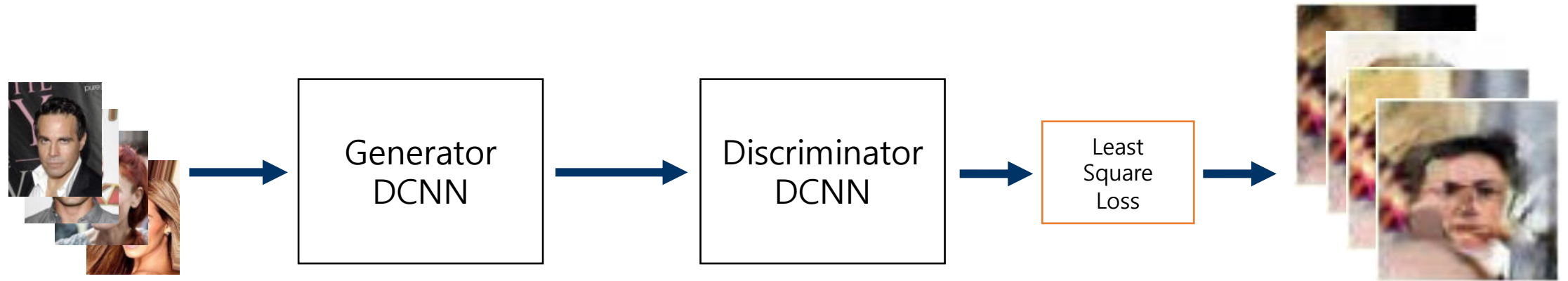
DCNN이 Generator 그리고 Discriminator를 형성하고 있고 Wasserstein loss값을 토대로 이미지의 진짜/가짜 여부를 판별한다. 그리고 감마값을 노이즈로 줌으로써 이미지의 판별값을 보다 엄격하게끔 해서 보다 정교한 이미지 생성이 가능하다.

WGAN Architecture



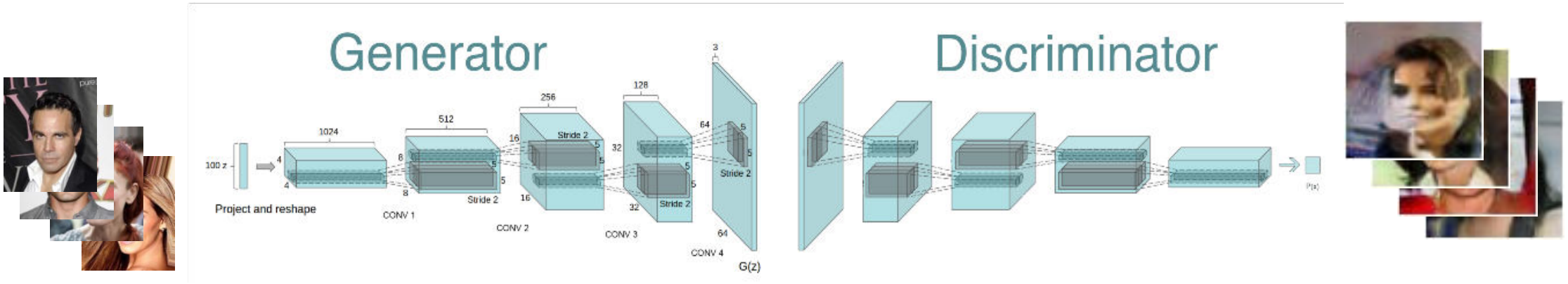
Wasserstein loss 값을 기반으로 이미지의 진짜/가짜 여부를 판별하여 보다 정교한 이미지 생성을 하고자 하였다.

LSGAN Architecture



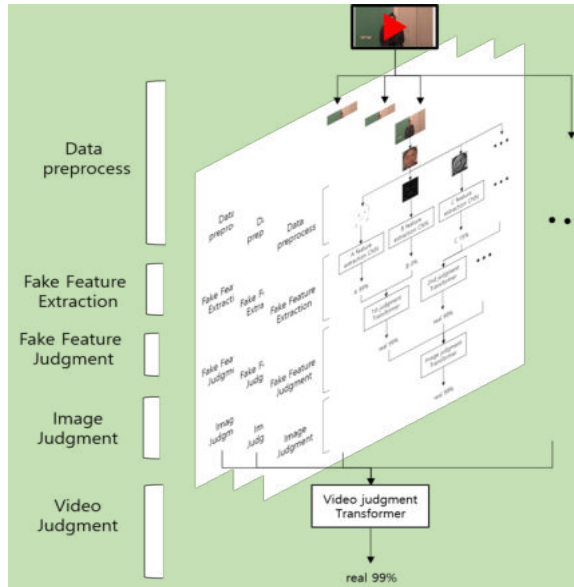
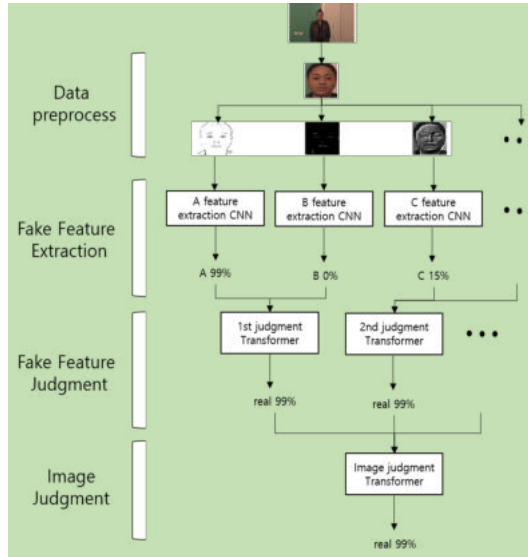
Least Square loss 값을 기준으로 이미지의 진짜/가짜 여부를 판별하여 discriminator가 보다 엄격하게 가짜 이미지 필터링을 하게끔 하였다.

DCGAN Architecture



Generator 그리고 Discriminator 각각 Deep convolutional neural network로 구성되어 있다.

가짜얼굴 이미지/동영상 판정



• 전처리

: 얼굴 크롭 및 각 특징에 최적화된 필터 적용

• 가짜 특징 검출

: CNN 모델을 사용하여 이미지/비디오에서 발견되는 가짜 특징 검출

• 가짜 특징 판정

: 검출된 가짜 특징 정보를 토대로 Transformer 모델이 Real/Fake를 판단

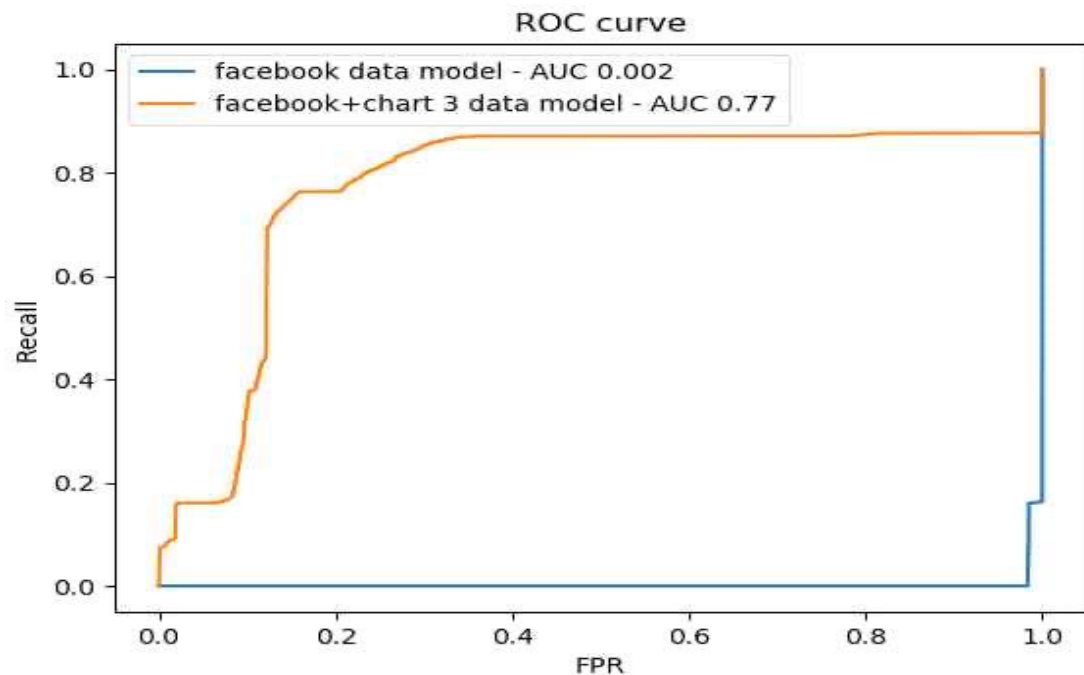
• 이미지 판정

: 가짜 특징 판정의 결과값을 입력으로 사용하여 이미지의 Real/Fake를 판단

• 비디오 판정

: 선택한 프레임 개수 만큼의 이미지 판정 결과값을 입력으로 사용하여 비디오의 Real/Fake를 판단

실험결과



Detection Model

AUC

Dang et al.

0.93

Wang et al.

0.906

Our model

0.77

실험방법

1. 페이스북 데이터로 학습된 모델로 테스트셋에 대한 탐지 진행
2. 다른 GAN 모델로 생성한 데이터로 모델을 보완해 같은 데이터셋에 대해 탐지 진행
3. 가짜 특징을 추가 하기 전후의 탐지율 비교

프로젝트 소개

라이브 스트리밍 서비스를 제공하는 SNS 앱

참여인원 1명

담당역할 개발(기여도 100%)

주요기능

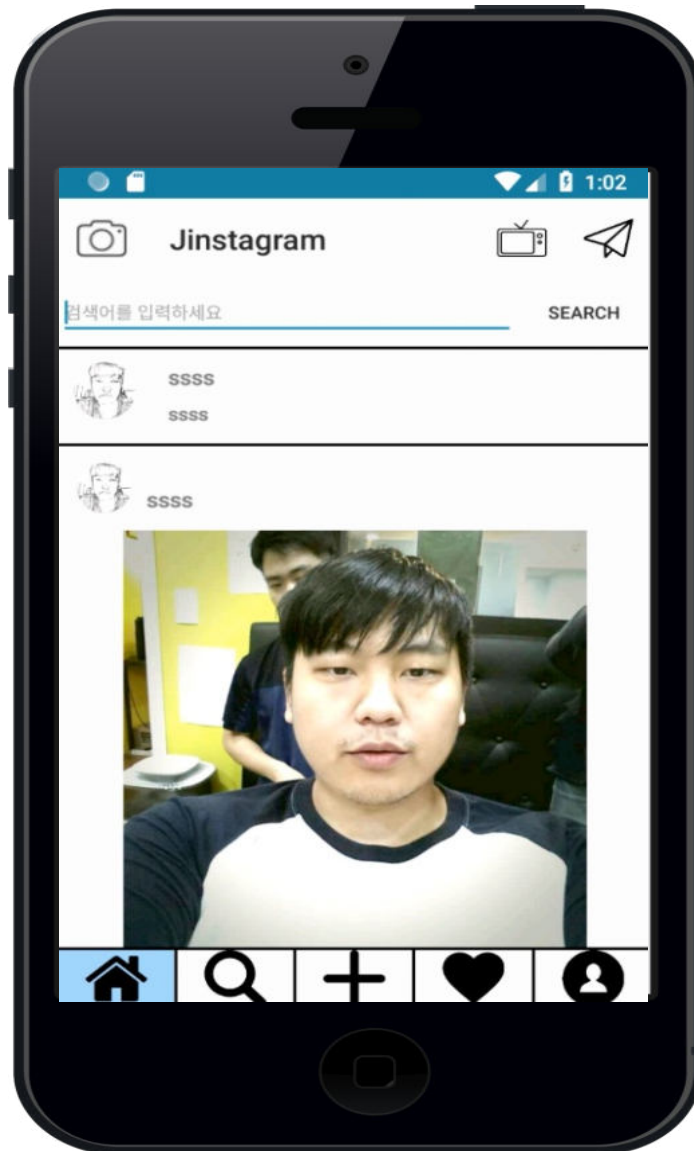
- 라이브 스트리밍
- 1대1 채팅,
- 얼굴인식,
- 이미지 필터,
- 게시물 업로드 및 좋아요.

개발기간 12주

시연영상 <https://cafe.naver.com/teamnovaopen/552>

사용기술

- **사용언어** Java, C++, PHP
- **IDE** Android Studio, Eclipse, Atom
- **프로토콜** TCP/IP, http, RTSP
- **웹서버** Apache
- **스트리밍 서버** Wowza Media Server
- **Database** MariaDB, MySQL, SQLite
- **API/Library** OpenCV, Volley, Picasso, Glide, Exoplayer



라이브 스트리밍



기능설명

스트리머(왼쪽)가 시청자(오른쪽)에게 보여주는 실시간 스트리밍 화면입니다.

주요기능

- 실시간 스트리밍
- 라이브 방송내 메시지 주고받기

얼굴인식



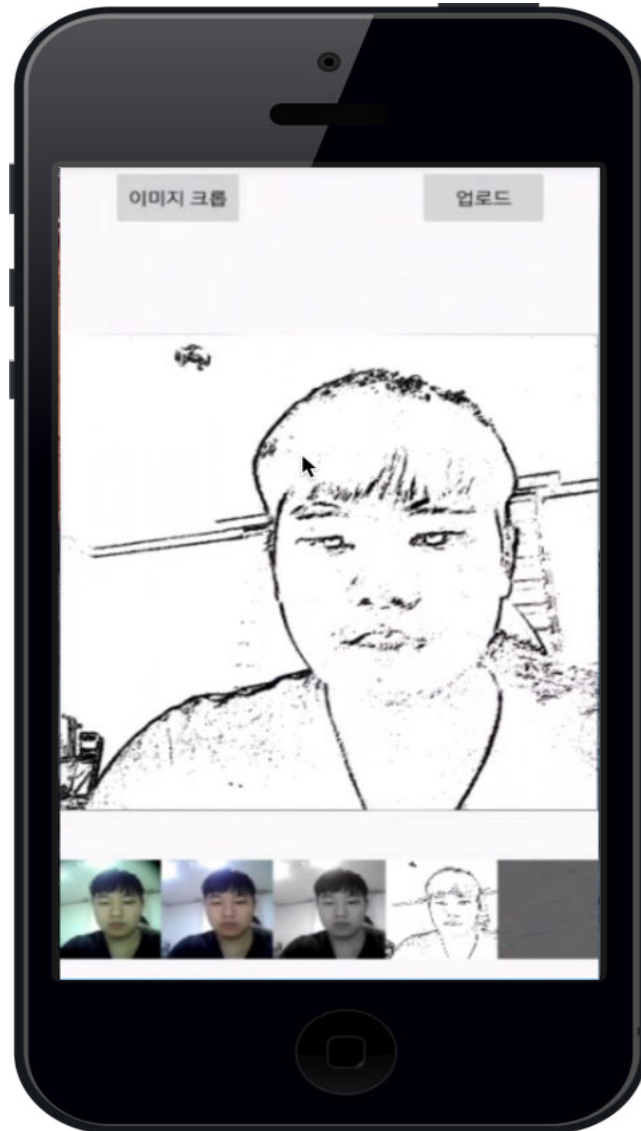
기능설명

전방/후방 카메라를 이용해서 사진을 촬영할 수 있고 프로필 사진에 사용하기 위한 얼굴사진을 촬영할 수 있습니다.

주요기능

- 사진 촬영
- 얼굴인식 모드 활성화/비활성화
- 이미지 저장

이미지 필터



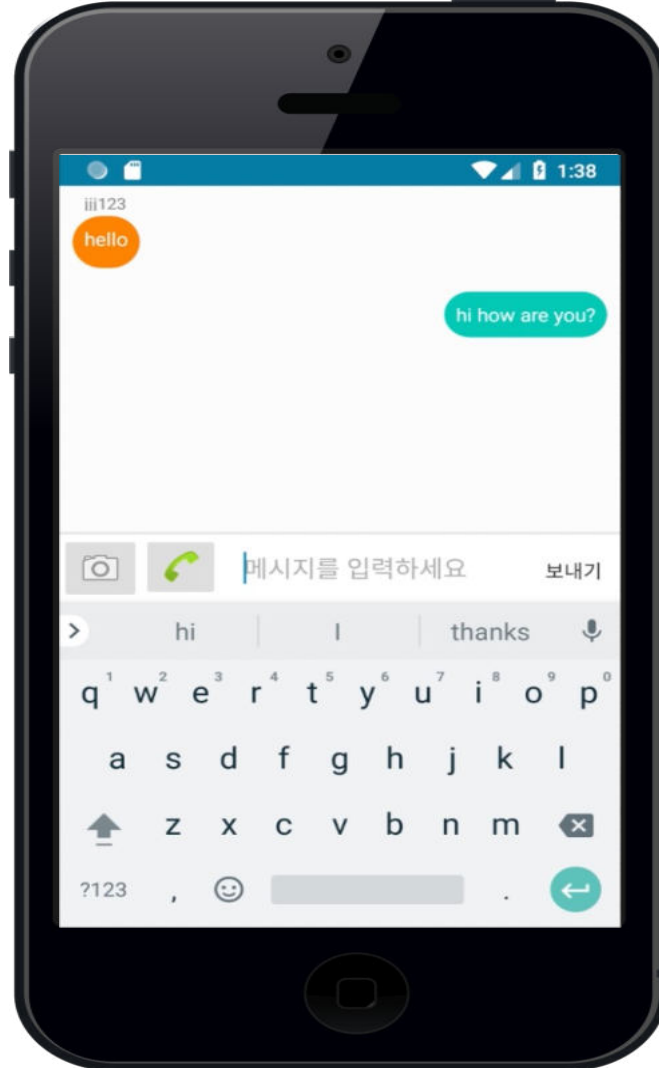
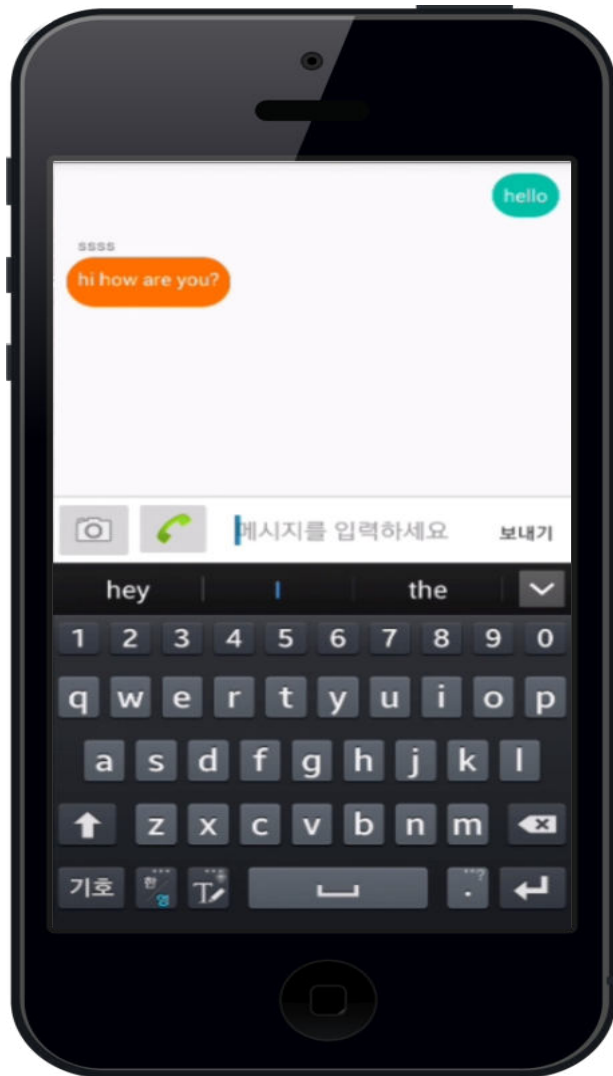
기능설명

저장되거나 촬영된 이미지를 수정하거나 바로 업로드 할 수 있습니다.

주요기능

- 이미지 크롭
- 이미지 필터 적용
- 이미지 업로드

1대1 채팅



기능설명

다른 사용자와 개인적인 메시지를 주고 받을 수 있습니다.

주요기능

- 각각 다른 사용자와 1대1 채팅
- 메시지 전송(이미지 포함)
- 메시지 수신(이미지 포함)
- 메시지 저장

Portfolio IV PHP 홈페이지(JInside)

← → ↻

localhost:8888/webpage/total.php


NAVER Google YouTube 쿠팡! 일반도서 - 리디북스 YES24 Amazon Boards | Trello 령킹닷컴 Stack Overflow ravh-cal/Convnet-... Gerald Edelman ~...

JInside 메인 자유게시판 스포츠게시판 패션/미용게시판

01X

NIKE

나이키 에어맥스 97



구매하기

전체게시판 목록

자유게시판

글번호	제목	작성자	날짜	조회수	추천
414	loren ipsum is the test texture	111	2020-10-28	5	1
413	Image upload	111	2020-10-22	3	0
206	I feel happy now	test98	2018-04-21	11	0
144	image upload	test98	2018-04-21	2	0
143	test	test98	2018-04-21	2	0

더보기

스포츠게시판

글번호	제목	작성자	날짜	조회수	추천
415	sports	jinigood	2020-10-28	2	0
205	test1user	test1	2018-04-21	0	0
204	test1user	test1	2018-04-21	2	0
203	test1user	test1	2018-04-21	0	0
202	test1user	test1	2018-04-21	0	0

111 님 로그인하셨습니다.

로그아웃

마이페이지

조회수 많은 글TOP5

제목	조회수
image test 4	137
test	58
another test	55
test	48
test	41

추천수 많은 글 TOP5

제목	추천수
image test 4	15

프로젝트 소개

관심있는 주제에 대해 자유롭게 게시물을 올릴 수 있는 커뮤니티

참여인원 1명

주요기능

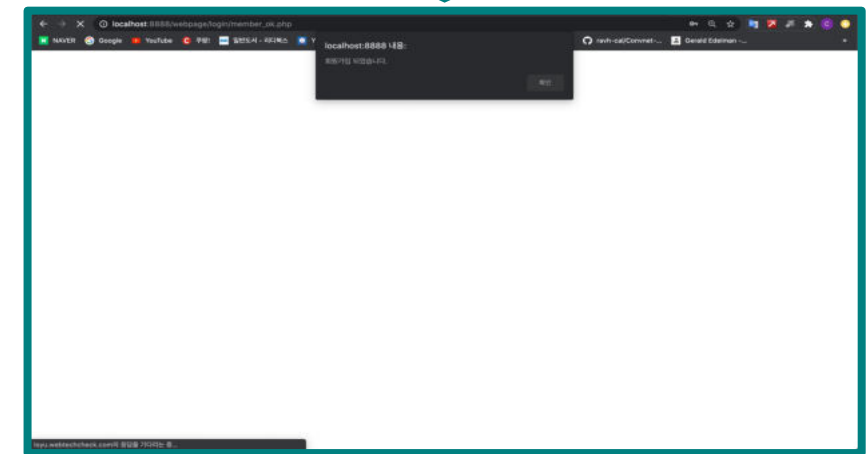
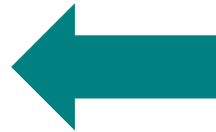
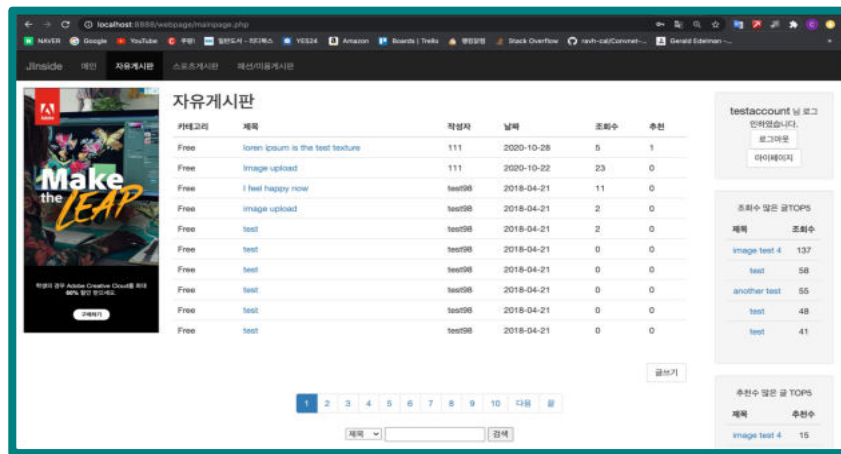
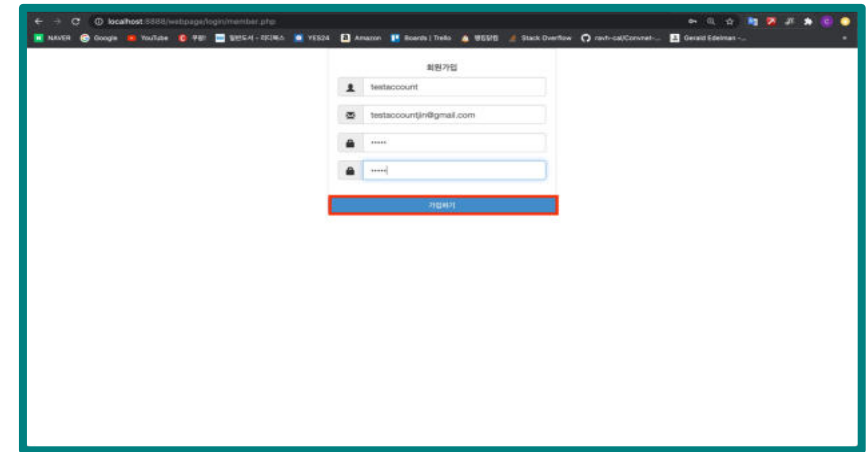
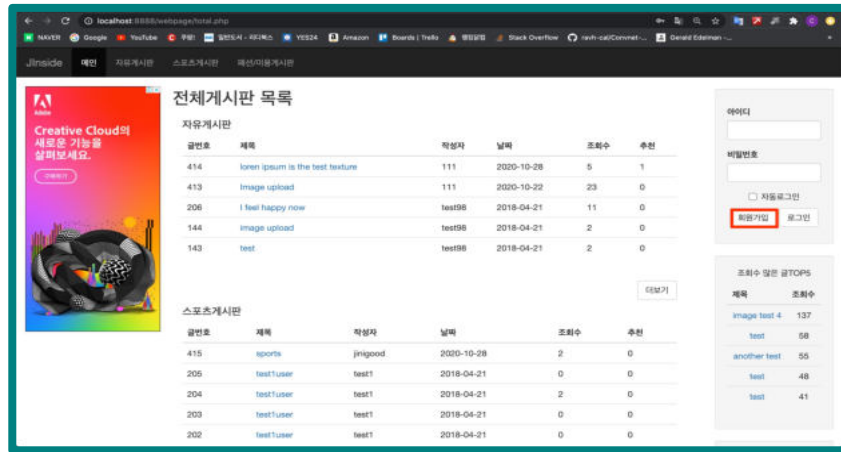
- 회원가입
- 로그인/로그아웃
- 게시물 업로드/수정/삭제
- 댓글작성/확인

개발기간 1일

사용기술

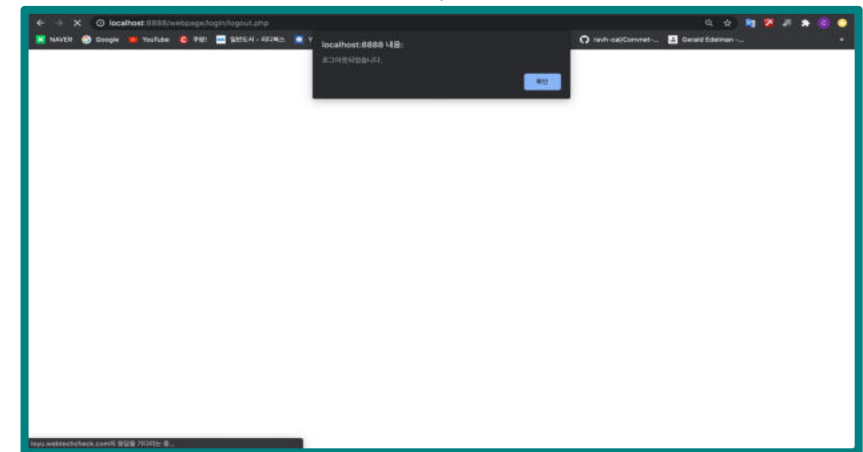
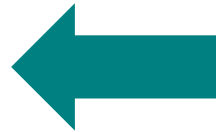
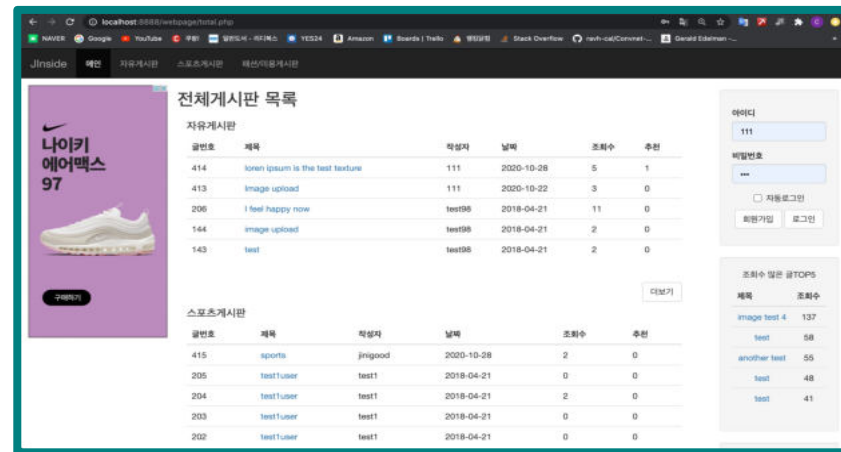
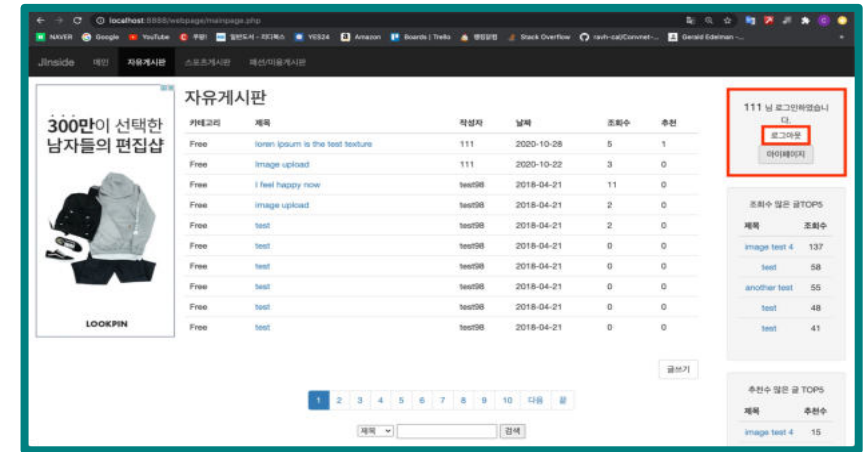
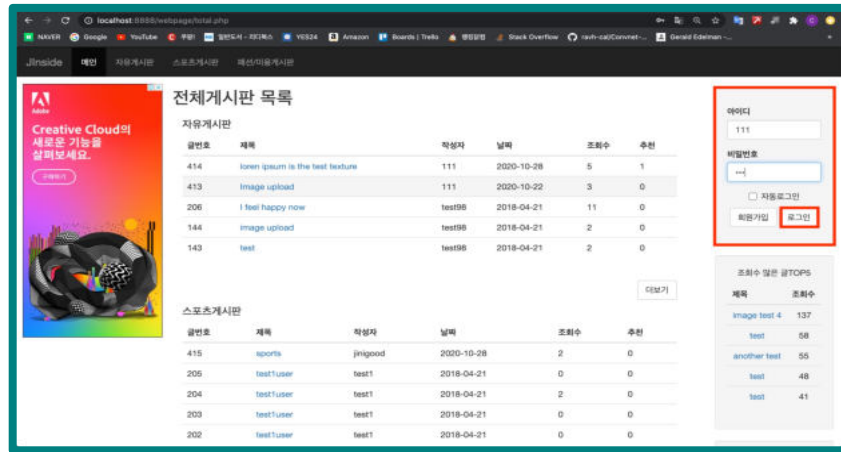
- 프로토콜 HTTP
- IDE Atom
- 언어 Java Script, PHP, HTML
- 웹서버 Apache
- Database MySQL

회원가입



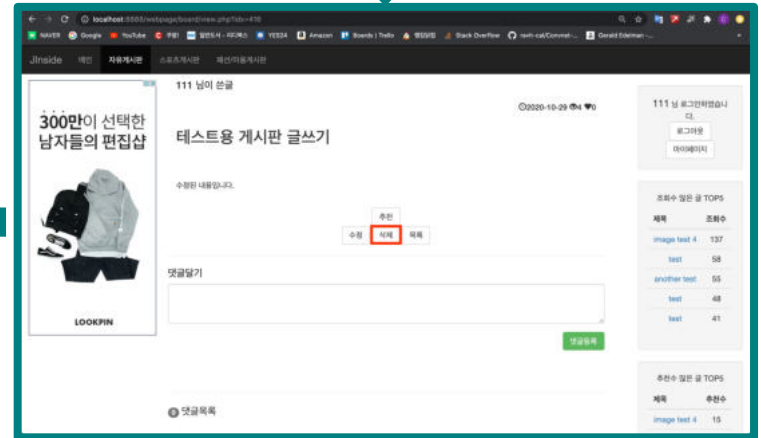
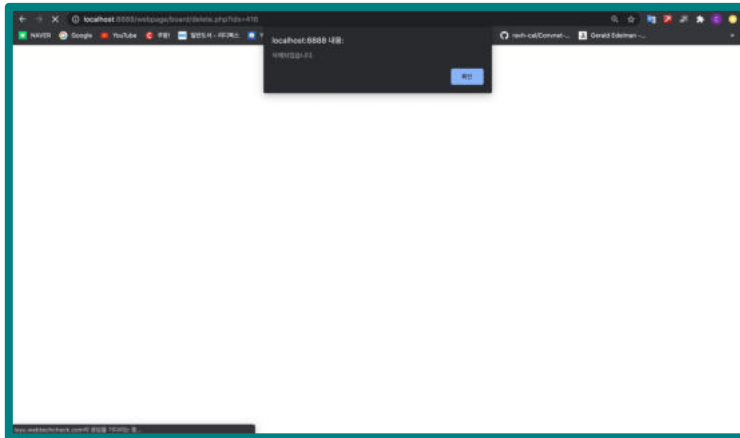
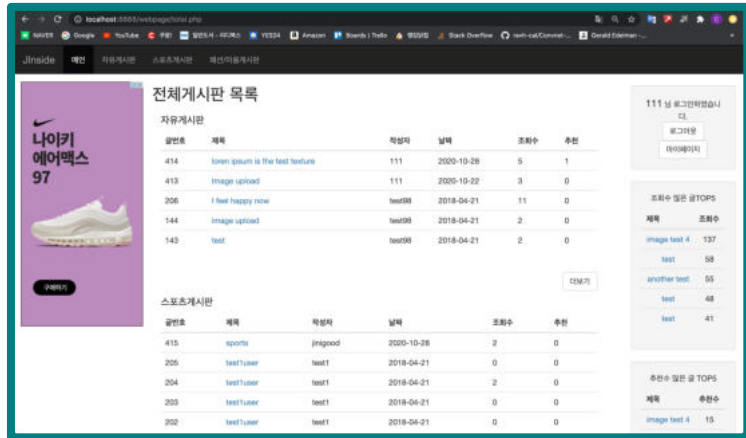
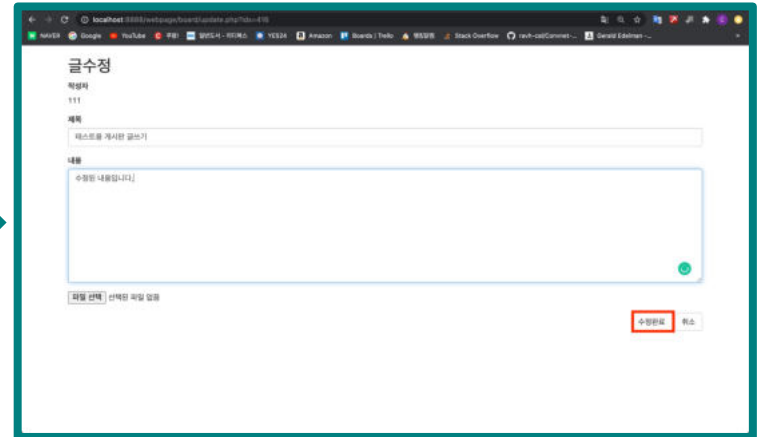
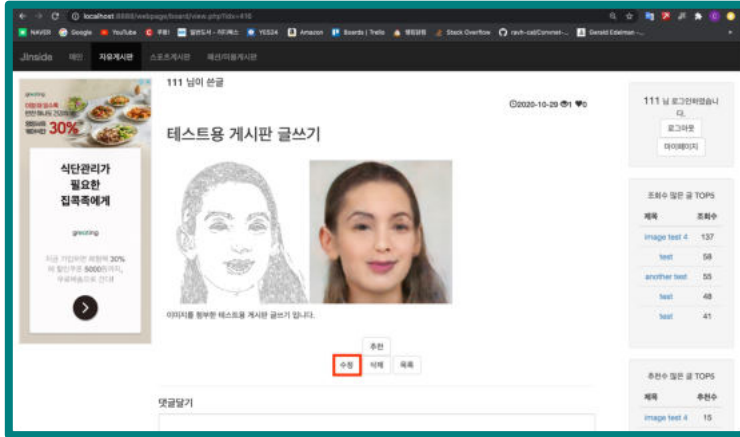
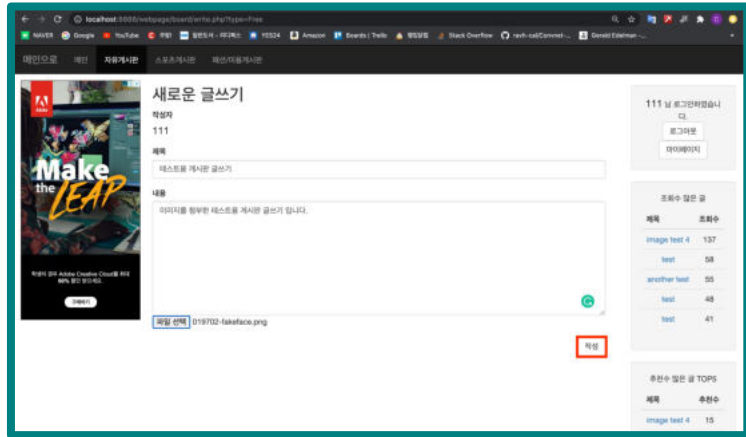
회원이 아닐 경우 신규회원 가입을 할 수 있습니다.

로그인/로그아웃



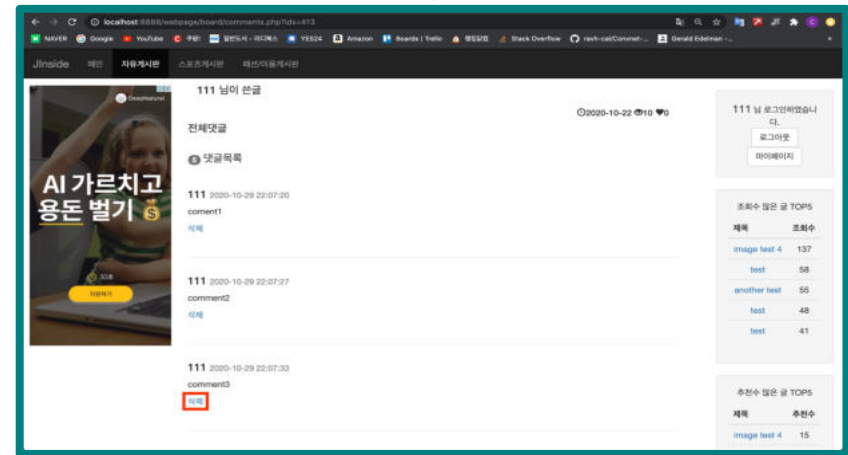
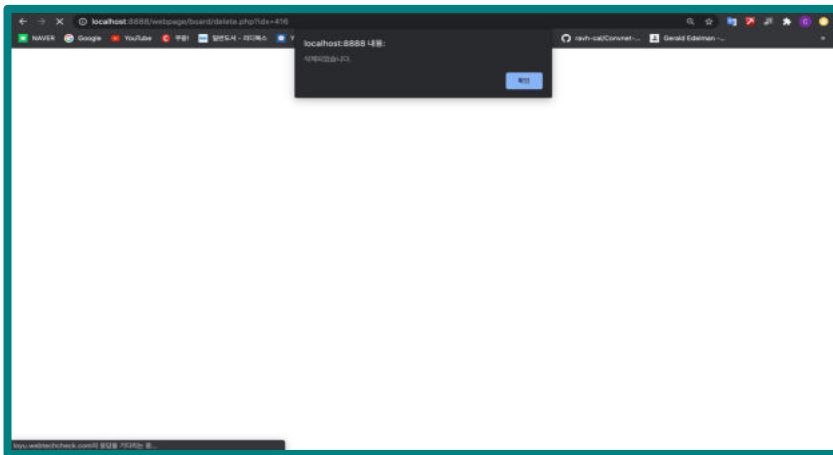
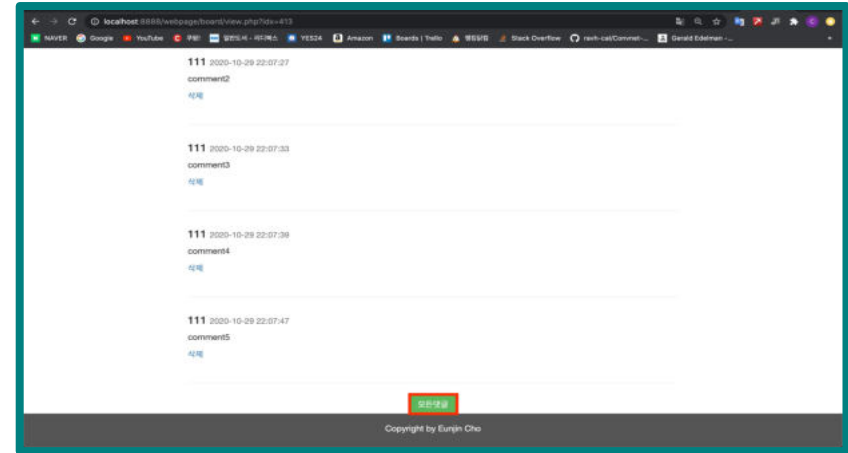
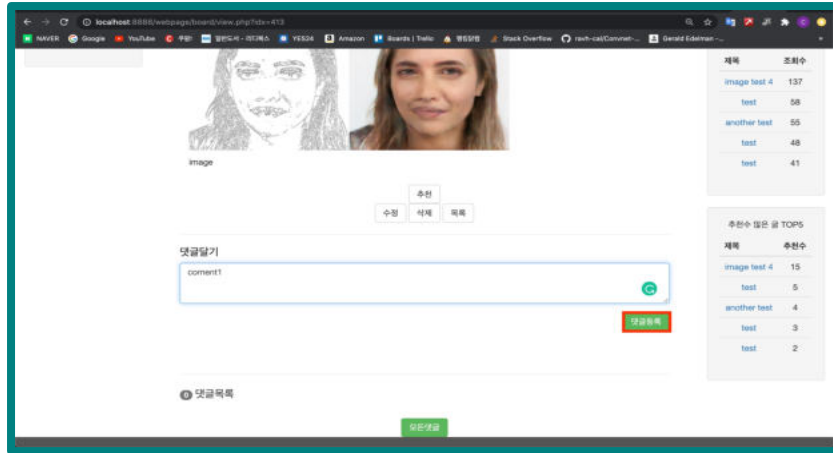
등록된 아이디로 로그인 로그아웃이 가능합니다.

게시물 업로드/수정/삭제



이미지를 첨부한 게시글 작성 및 수정 그리고 삭제를 할 수 있습니다.

댓글작성/삭제



게시글에 대한 댓글을 작성할 수 있고 댓글 확인 및 내가 작성한 댓글에 대해 삭제를 할 수 있습니다.

Portfolio V 모바일 카드인식 솔루션(AOS)

소개

실시간 은행카드 스캐너

참여인원 2

역할 연구원, 메인 개발자(70% 기여도) : 설계, 안드로이드 개발, C++ 네이티브 개발, 모델 변환 및 최적화

메인기능

- 은행카드 번호/날짜 탐지
- 탐지한 번호인식,
- 카드번호 유효성 체크

데모 비디오 [\[링크\]](#)

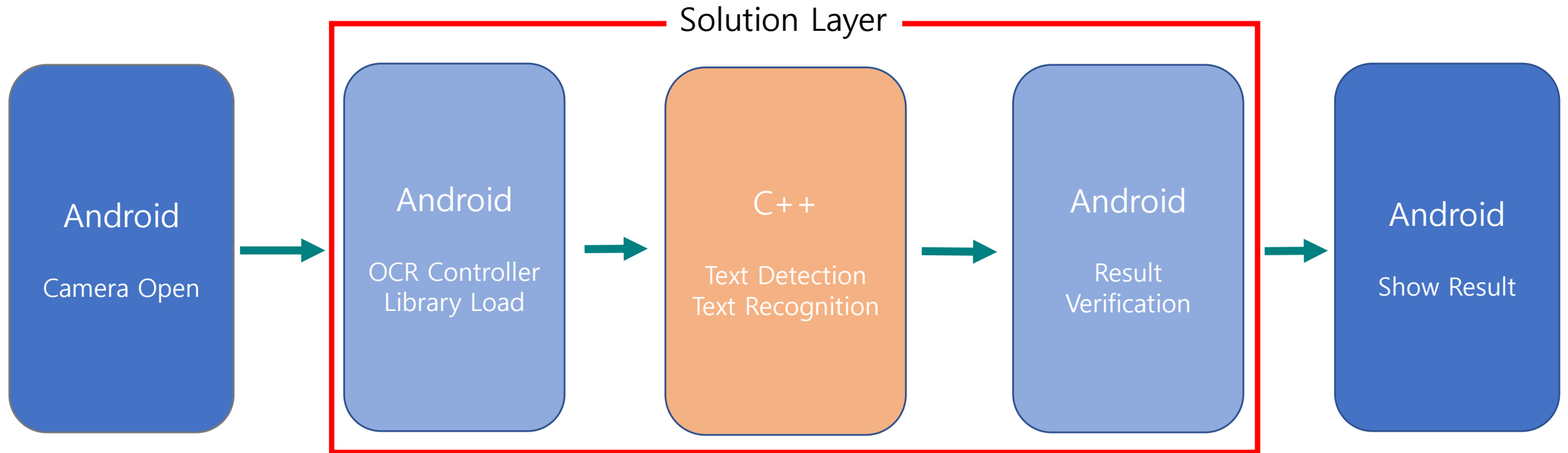
총 개발기간 6주

기술목록

- 언어 Java, C++, Python
- IDE Android Studio, Pycharm
- 사용 알고리즘 EAST-MobileNetV3, CRNN-MobileNetV3
- API/Library OpenCV, PaddleLite, Numpy, Shapely,



Solution Architecture



Portfolio VI 모바일 카드인식 솔루션(iOS)



소개

실시간 은행카드 스캐너

참여인원 1

역할 연구원, 메인 개발자 : 설계, iOS 개발, 모델 변환 및 최적화

메인기능

- 은행카드 번호/날짜 탐지
- 탐지한 번호인식,
- 카드번호 유효성 체크

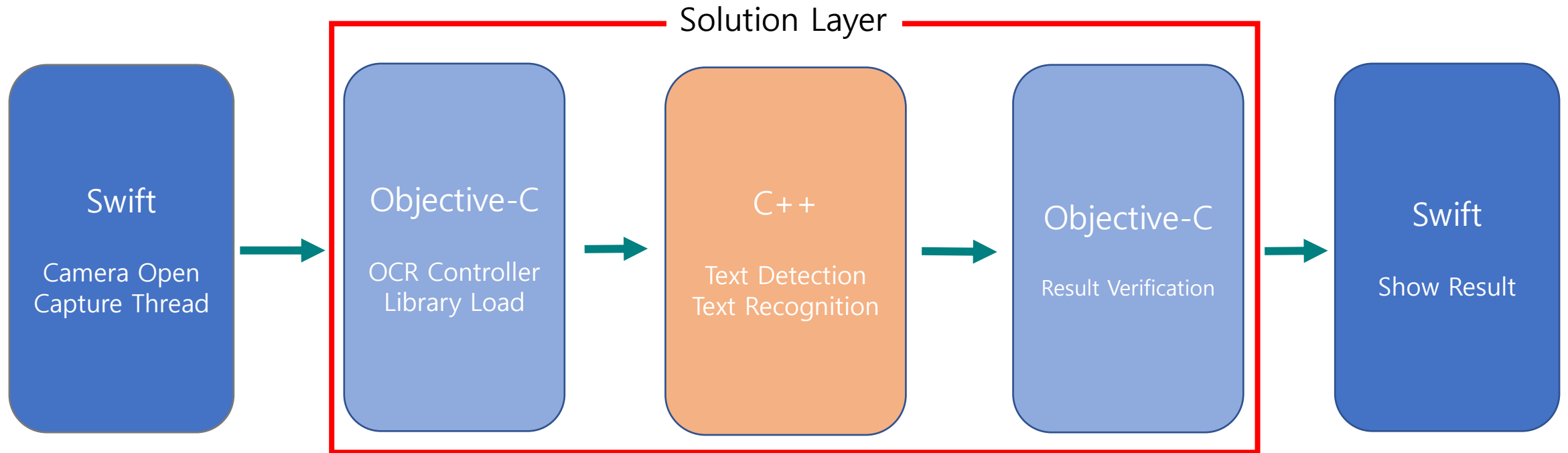
데모 비디오 [\[링크\]](#)

총 개발기간 4주

기술목록

- 언어 Swift, Objective-C, C++, Python
- IDE XCode, Pycharm
- 사용 알고리즘 EAST-MobileNetV3, CRNN-MobileNetV3
- API/Library OpenCV, PaddleLite, Numpy, Shapely

Solution Architecture



Portfolio VI 모바일 신분증 인식 솔루션(iOS)

소개

실시간 주민등록증, 운전면허증 스캐너

참여인원 1

역할 연구원, 메인 개발자 : 설계, iOS 개발, 모델 변환 및 최적화

메인기능

- 신분증 정보 탐지
- 탐지한 정보 인식
- 필수정보 마스킹
- 얼굴영역 크롭

데모 비디오 [\[링크\]](#)

총 개발기간 3주

기술목록

- 언어 Swift, Objective-C
- IDE XCode
- 사용 알고리즘 YoloV5, TesseractOCR
- API/Library OpenCV, PaddleLite, Tesseract



Solution Architecture

