

Feeling Manager

감정인식을 활용한 실시간 노래추천 서비스



헬스케어 팀프로젝트 김지훈 조 최종발표

역할분담

이름	역할
김지훈	팀장, 백엔드 서버 관리, 개발
김성환	백엔드, CICD
송준혁	웹 프론트엔드 개발
안성준	안드로이드 어플리케이션 개발
이웅규	DB관리, 데이터수집
한상범	ML 관리자
장유란	백엔드 서버 관리, 개발





목차

part1 소프트웨어 개발

part2 감정인식 AI 개발



Part 1, 소프트웨어 개발



프로젝트 개요

음악은 13가지 감정을 일으킨다



곽노필 기자

+구독

f t talk link ★ 📁 가+

미국인-중국인 대상 2천여곡 반응 조사 결과
기쁨, 평안, 공포, 활력 등 13가지 범주 나뉘져
같은 음악엔 같은 감정 반응...보편성도 확인
인터랙티브 지도 공개...개인 목록 활용 가능

좋아하는 음악 들으면 스트레스와 피로도 감소에 큰 도움

좋아하는 음악 30분 감상하면 코르티솔 농도 감소...비선호...

등록 2017-07-24 오전 9:26:40
수정 2017-07-24 오전 9:32:51

가

가

스트레스 감소를 위한 음악치료 연구 메타분석

박현경, 이주영 - 한국음악치료학회지, 2014 - dbpia.co.kr

이 연구는 스트레스 감소를 목적으로 실시된 음악치료 연구의 효과를 알아보기 위해 메타분석을 실시했다. 2004년부터 2013년까지 실시된 실험연구 논문 43편에서 132개의 효과크기를 산출...

☆ 저장 99 인용 8회 인용 관련 학술자료 전체 3개의 버전



프로젝트 개요

"스트레스, 약간만 받아도 기억력 떨어져"

이해나 헬스조선 기자 | 김명주 헬스조선 인턴기자

캐나다 브리티시컬럼비아대 연구팀은 성인 남녀 140명을 대상으로 스트레스가 작업 수행 능력에 미치는 영향을 연구했다. 연구팀은 대상자들에게 컴퓨터 기반 인지 시험을 보게 했고, 대상자의 양옆에 그들을 감시하는 사람을 뒤 약간의 스트레스를 받게 했다. 그 결과, 약간의 스트레스가 주어질 때 대상자 대부분은 집중력·기억력·문제 해결력·자기 통제력이 떨어졌다. 약간의 스트레스가 주어진 후 작업 수행 능력이 개선된 대상자는 소수에 불과했으며, 개선의 정도는 크지 않은 것으로 나타났다.



AI를 이용한 실시간 음악 자동 추천 서비스



사용자의 표정 Detection으로 감정 Classification



현재 사용자의 감정에 맞는 음악 추천



스트레스관리와 업무 효율 증대



개발 전략

개발 프로세스



AGILE - SCRUM Process

현재 서로의 개발 능력파악이 힘들고, 본인도 어디까지 할 수 있는지 모름

- 에자일 스크럼의 짧은 스프린트와 지속적인 피드백을 통해 프로젝트 수행능력을 토대로 빠른 의견수용, 프로젝트의 변화 대처 가능

팀 프로젝트의 특성상 서로 무엇을 하고 있는지 몰라,업무 치중의 오해 발생

- 매일 열리는 스탠드업 미팅으로 팀 전체가 무엇을 하는지 알 수 있어 오해를 줄일 수 있음



개발 전략

형상관리와 협업 툴



커밋 규칙, 브랜치 및 PR 관리, 변수표기법 통일 등, 프로젝트 규칙에 맞춰 작성

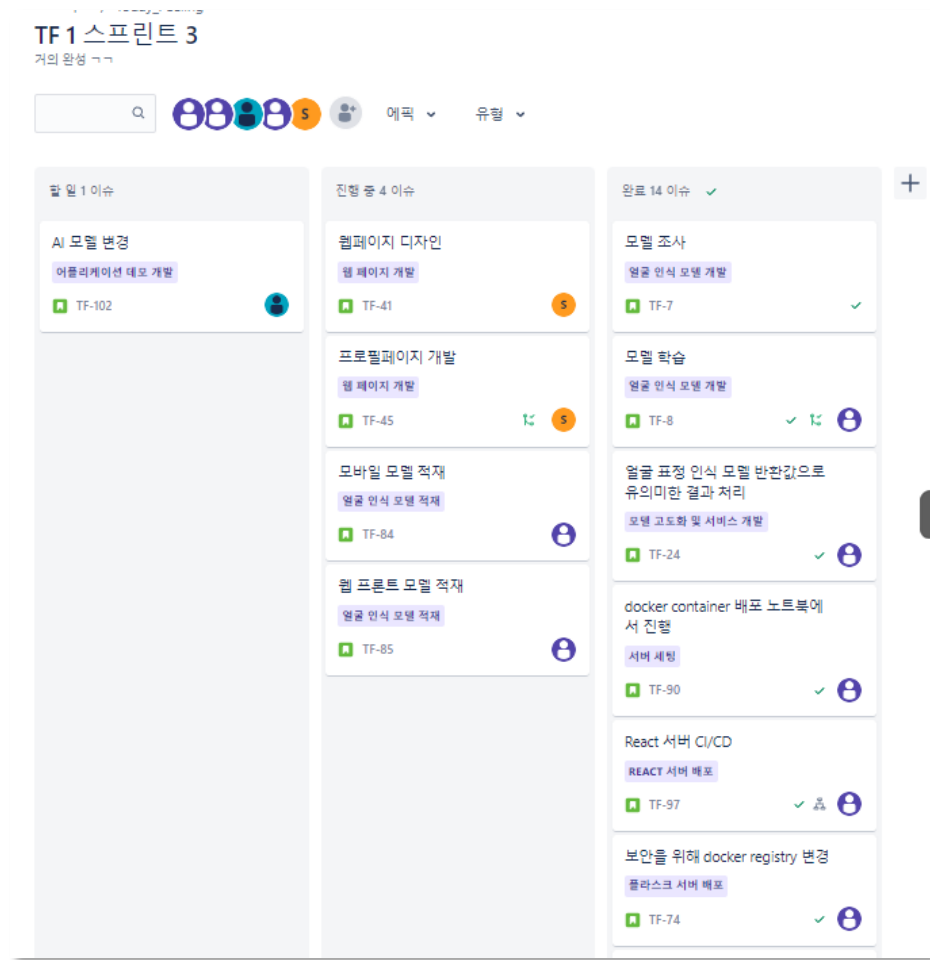
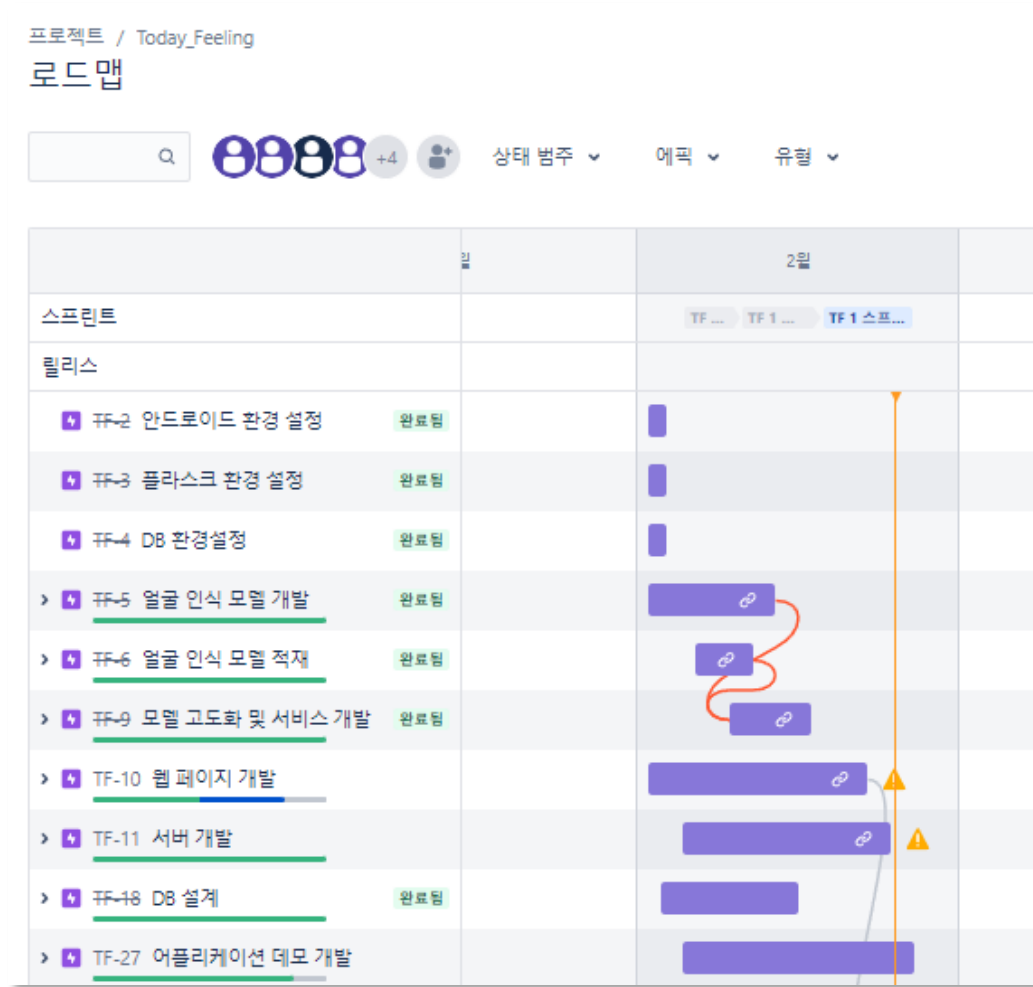


애자일에 최적화된 JIRA를 통해 협업 관리



개발 전략

JIRA와 AGILE SCRUM









개발 전략


JIRA와 AGILE SCRUM

연결된 풀리퀘스트
지난 30일 동안 팀의 연결된 풀리퀘스트입니다. 자세히 알아보기

풀리퀘스트 검색 리포지토리 작성자 검토자 상태

요약 상태 :

 Tf 94 cors → main	#88 • 리포지토리: Gocha64/Today Feeling	병합됨
 TF-46-Camera → main	#87 • 리포지토리: Gocha64/Today Feeling	병합됨
 TF-46-Camera → main	#86 • 리포지토리: Gocha64/Today Feeling	병합됨
 TF-94 json으로 userid를 보내도 인지가능 → main	#83 • 리포지토리: Gocha64/Today Feeling	병합됨
 TF-94 CORS 정책 수정 → main	#82 • 리포지토리: Gocha64/Today Feeling	병합됨
 TF-45-Profile → main	#81 • 리포지토리: Gocha64/Today Feeling	병합됨

▼ 자료실		업데이트 2월 10, 2023
> 설계 폴더		업데이트 2월 10, 2023
> 발표 폴더		만들 2월 10, 2023
▼ 회의록 폴더		만들 2월 10, 2023
23.02.06 회의록		만들 2월 14, 2023
23.02.07 회의록		
23.02.08 회의록		
23.02.09 회의록		
23.02.10 회의록		

회의내용

1. 진행상황

프론트

안성준 : 로그인, 회원가입, 프로필화면, 회원정보수정, 얼굴탐지, 전처리 및 모델타입, 유튜브 재생, 메인페이지, 통계페이지 -막대그래프, 표정분석을때 -> 추천했던 노래와 시간 출력, 기존모델로 교체

송준혁 : 로그인, 회원가입, 프로필페이지, 회원정보수정

백엔드

김지훈, 장유란 : API서버 모든 작업 완료, 기술 지원, cors 정책

김성환 : 플라스크 CI/CD 완료, 서버를 노트북으로 배포(터널링 사용)

ML

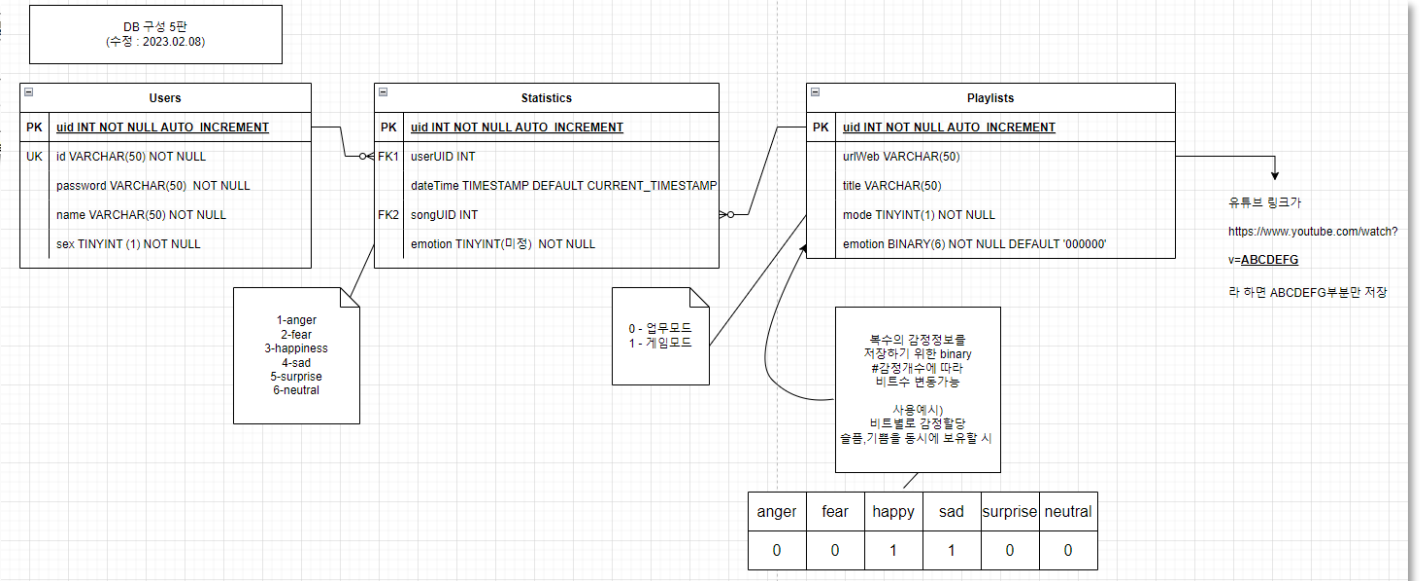
한상범 : 모델조사, 고도화 방법 연구, 수차례 모델학습 완료, 얼굴인식 모델 구축

DB

이웅규 : 서버구축, 스키마 설계, 크롤링프로그램 제작, 데이터 DB저장 자동화 프로그램 제작, 데이터 DB저장 자동화 프로그램에 log로 저장

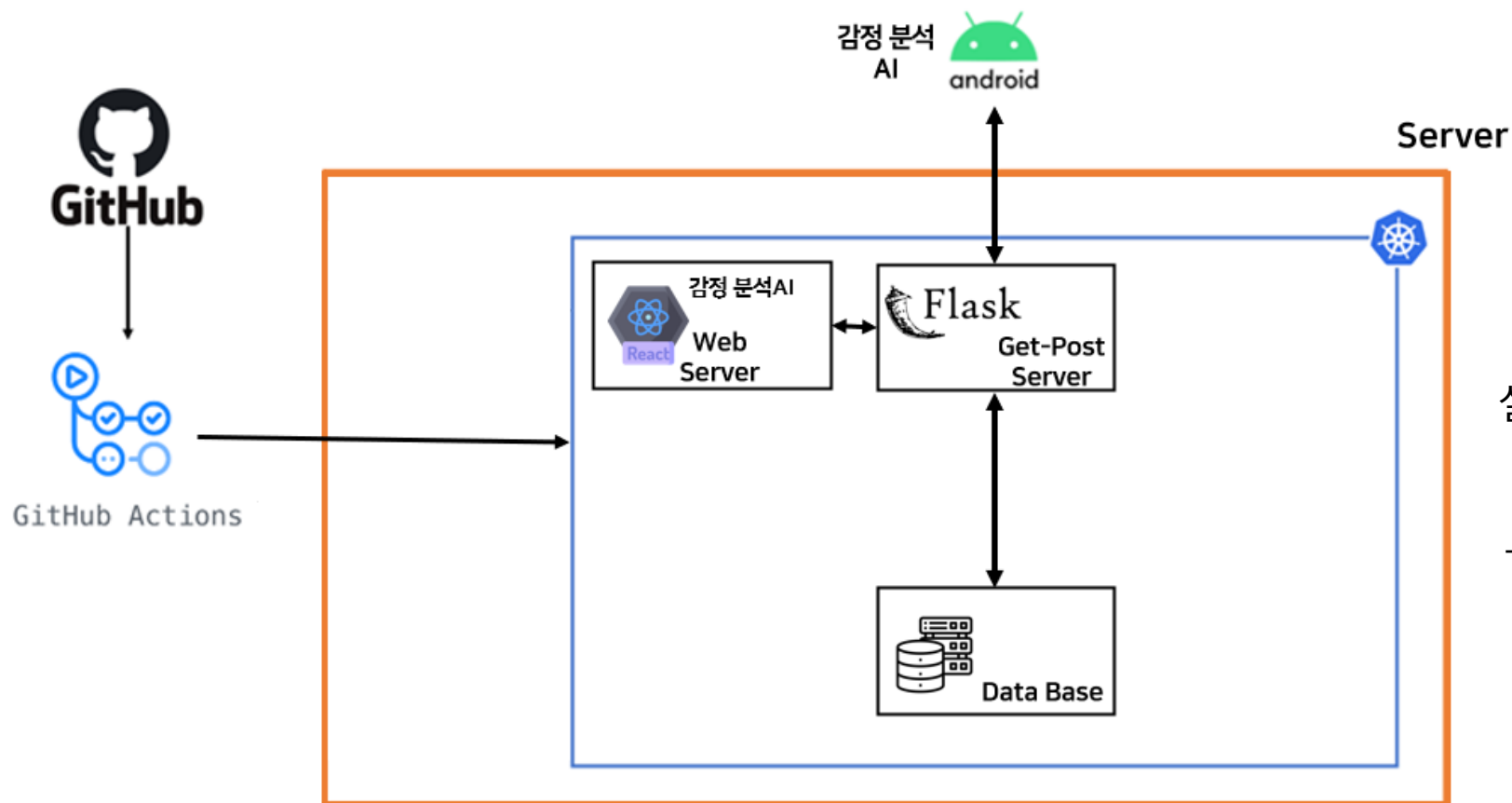
설계

화면	시작 화면	기능	기능 설명
splash(일반 화면)	로그인 화면		
	상단 바 - 로그인, 회원가입, 마이페이지 버튼, 로그인 버튼		
	로그인 - Info Page		
	회원 - ID/비밀번호 찾기 버튼		
회원가입 페이지 (일반)			
로그인 페이지 (일반)	로그인 화면	아이디, 비밀번호 입력 & 회원가입, 비밀번호 찾기	DB 구성 5만 (수정 : 2023.02.08)
회원가입 페이지	회원가입 화면	아이디, 비밀번호, 비밀번호 확인, 생년월일, 이름, 이메일, 성별	<div> <div>Users</div> <div>PK uid INT NOT NULL AUTO INCREMENT</div> <div>UK (varchar(50)) NOT NULL</div> </div>
프로필 페이지	로그아웃	개인정보 삭제, 회원정보 변경, 내 정보	



개발 전략

구조도



실시간 영상분류를 백엔드에서 처리 시,
서버 과부하, 트래픽 과부하 발생

- 안드로이드, 프론트 AI 실시간 분류로
변경



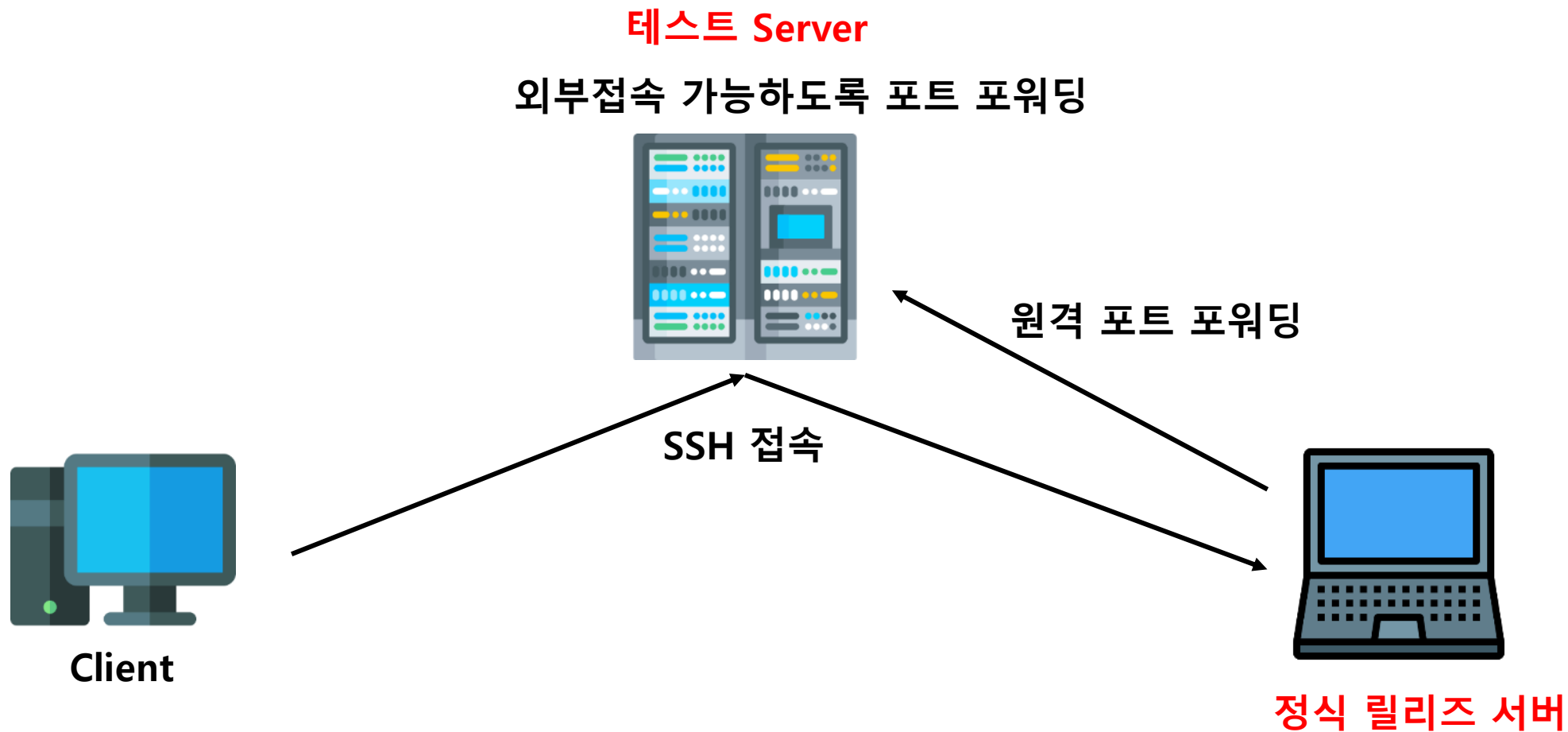
CI/CD

정식 테스트서버와 정식 릴리즈 서버



CI/CD

정식 테스트서버와 정식 릴리즈 서버



CI/CD

정식 테스트서버와 정식 릴리즈 서버

외부접속 가능하도록 포트 포워딩

2	순번	프로그램	중요도	빈도	품질특성	설명	대응	처리여부
3	TC1	안드로이드 앱	상	상	신뢰성	얼굴 인식 시 카메라에 너무 가깝거나 얼굴이 겹쳐 있을 경우 프로그램이 비정상종료됨	음수 길이에 따른 오류로 이에 대한 예외처리	완료
4	TC2	안드로이드 앱	하	상	사용성	감정별 선호 노래 UI에서 따옴표가 비정상적으로 표기됨	마지막 문자 뒤에는 따옴표가 붙지 않도록 처리	완료
5	TC3	안드로이드 앱	상	상	사용성	로그인 실패 시 이에 대한 피드백 없음		처리중
6	TC4	웹 크롤러	상	중	신뢰성	일부 플레이리스트 파싱 중 오류발생	lazyload되는 객체로 인한 문제이며 스크롤 내리기를 추가해 모든 객체가 로드된 후 파싱하도록 수정	완료
7	TC5	웹 크롤러	상	하	신뢰성	사용자 수정 가능한 플레이리스트에서 오류발생	사용자 수정 가능한 플레이리스트는 파싱하지 않도록 설정	완료
8	TC6	데이터 주입 툴	상	중	신뢰성	일부 제목을 INSERT시 오류 발생	"" %등의 특수문자가 들어가 있을 경우 발생하며 이스케이프 시퀀스로 변환	완료
9	TC7	데이터 주입 툴	상	하	신뢰성	너무 긴 제목을 INSERT시 오류 발생	노래의 길이가 너무 길 경우 DB의 제한 길이만큼으로 잘라냄	완료
10	TC8	데이터베이스	상	하	보안성	DB의 port가 3306임	보안을 위하여 well known포트가 아닌 포트를 사용	완료
11	TC9	데이터베이스	상	상	기능적합성	DB의 시간이 실제 시간과 다름	DB의 시간 설정 문제로 UTC+9를 이용하도록 변경	완료
12	TC10	웹 크롤러	상	상	이식성	일부 시스템에서 경로 문제 발생	파이썬의 실행 경로 문제로 실행 시 현재 파일 위치에 따라 경로를 세팅하도록 함	완료
13	TC11	웹 크롤러	상	하	신뢰성	플레이리스트에 재생 불가능한 노래가 포함되었을 시 오류발생	재생 불가능한 노래는 생략하도록 예외처리함	완료

Client

Main Server



BackEnd

FLASK



왜 Flask 인가?

매우 가벼움 -> 간단한 API 작성에 유리
백엔드 인원의 겹치는 기술 스택



BackEnd

API 설계

일별 감정 정보 조회

✓ 일별 감정 정보를 분석 할 수 있는 API

METHOD: GET

URL: /emotion/stat_day

PARAMETER: date : timestamp(default: timestamp.now())

REQUEST HEADER

```
1 cookie {'session' : session_id}
```

REQUEST BODY

```
1 None
```

RESPONSE HEADER

```
1 set-cookie {'session' : session_id}
```

RESPONSE

```
1 {"result" : statData}
2 or
3 {"result_f" : 실패이유}
4
5 statData :
6 {
7   "count": int,
8   "angerCount" : int,
9   "fearCount" : int,
10  "happinessCount" : int,
11  "sadnessCount" : int,
12  "surpriseCount" : int,
13 }
14
15 실패이유 : "authentication failed", "undefined error"
```

SERVICE

- DB에서 해당하는 유저와 기간의 유저 감정 데이터 전달
- 해당 기간 내의 데이터 개수, 감정 별 데이터 개수를 보내줌

API 명세 폴더	이름	업데이트 일자
회원 가입 API	03	업데이트 2월 14, 2023
감정 정보 분석 API	0	업데이트 2월 17, 2023
감정 정보 조회 API	033	업데이트 2월 16, 2023
회원 감정 별 장르 데이터 수정 API	0	업데이트 2월 14, 2023
회원 정보 수정 API	03	업데이트 2월 14, 2023
회원 정보 조회 API	0	업데이트 2월 14, 2023
로그인 API	03	업데이트 2월 14, 2023
노래추천/감정정보 기록 API	03	업데이트 2월 16, 2023

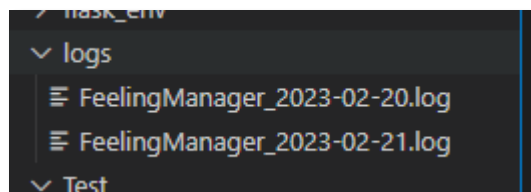
JIRA의 Confluence 활용

API명세 변경 시 팀원 즉시 확인 가능



BackEnd

LOG



```
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:03:56] "GET /favicon.ico HTTP/1.1" 200 -
INFO:root:connected from 192.168.55.96, userId: regiTest
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:03:56] "GET /emotion/info HTTP/1.1" 200 -
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:03:56] "GET /favicon.ico HTTP/1.1" 200 -
INFO:root:connected from 192.168.55.96, userId: regiTest
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:03] "GET /member/search HTTP/1.1" 200 -
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:03] "GET /favicon.ico HTTP/1.1" 200 -
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:11] "GET /logout HTTP/1.1" 200 -
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:11] "GET /favicon.ico HTTP/1.1" 200 -
INFO:root:connected from 192.168.55.96, userId: regiTest
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:19] "[32mGET /member/logout HTTP/1.1[esc[0m" 302 -
INFO:root:connected from 192.168.55.96
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:19] "GET / HTTP/1.1" 200 -
INFO:werkzeug:192.168.55.96 - - [21/Feb/2023 14:04:19] "GET /favicon.ico HTTP/1.1" 200 -
```

로그를 통한 테스트서버 오류 확인 및 저장



Database

유튜브 뮤직 자동 크롤러

```
PS C:\bootcamp\Today_Feeling\DB\tools> & C:/Users/lee/App
Data/Local/Programs/Python/Python39/python.exe c:/bootcam
p/Today_Feeling/DB/tools/crawler.py
open data.xlsx
open data.xlsx OK
connecting Chrome
connecting Chrome OK

playlist detector started
go to playlist

playlist detected : 연인과의 달달한 드라이브 가요
songs detected : 96
parsing start
processing 96/96
parsing OK
genre of current playlist(quit : -1) :
```

유튜브 뮤직 자동 크롤러

```
2023-02-21 21:01:55,822 - INFO - INSERT OK sql : insert into Playlists(urlWeb,title,genre) values ("iusSuTrMRno","Drive Away","1")
2023-02-21 21:01:55,830 - INFO - INSERT OK sql : insert into Playlists(urlWeb,title,genre) values ("TC5Bf1XkIcs","Connect","1")
2023-02-21 21:01:55,838 - INFO - INSERT OK sql : insert into Playlists(urlWeb,title,genre) values ("0iB47JaNFxk","ice cream","1")
2023-02-21 21:01:55,845 - INFO - INSERT OK sql : insert into Playlists(urlWeb,title,genre) values ("niJV_FUeqA8","사랑느낌","1")
2023-02-21 21:01:55,854 - INFO - total elements : 96
2023-02-21 21:01:55,854 - INFO - insert OK : 96
2023-02-21 21:01:55,855 - INFO - insert KO : 0
```

자동 DB Insert



Android

핵심 기술

MLKit을 이용한 실시간
얼굴탐지, CROP

데이터 전처리

표정 인식 모델



Android

시연 영상



소프트웨어 파트

향후 계획

WEB 개발 완료

노래 추천 알고리즘 개발

플레이스토어 런칭



Part 2, 감정인식 AI 개발

2.1 사전 조사

2.2 분석 순서도

2.3 분석



데이터 수집

사용 데이터
**Aihub : 한국인 감정
인식을 위한 복합 영상**

anger : 7853개
fear : 6952개
happiness : 7624개
neutral : 7191개
sadness : 7499개
surprise : 6988개



데이터 전처리

Face Detection
통해 원본 이미지의 얼굴만 검출
Haar Cascades 활용



모델

Face Detection을 거친
얼굴 이미지로부터
감정 인식

**Facial-expression
recognition-using-cnn
활용**



“neutral”

중요도가 높은 Face Landmarks
선정 후 훈련

모델평가 -> 정확도/Inference

데이터 수집

aihub : 한국인 감정 인식을 위한 복합 영상

anger, fear, happiness, neutral, sadness, surprise 등
7개 카테고리의 감정 데이터

사용 데이터

- anger(7853개)
- fear(6952개)
- happiness(7624개)
- neutral(7191개)
- sadness(7499개)
- surprise(6988개)

[anger]



[neutral]



[sadness]



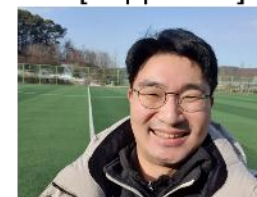
[fear]



[surprise]



[happiness]



데이터 전처리 - Face detection

Opencv Haar Cascades를 이용한 데이터 전처리



원본 이미지



이미지 crop 결과 이미지 (256 by 256)

데이터 전처리

Face_cascade.detectMultiScale 인자 비교

```
faces = face_cascade.detectMultiScale(image, scalefactor = 1.3, minNeighbors = 5)
```

image : 입력 영상

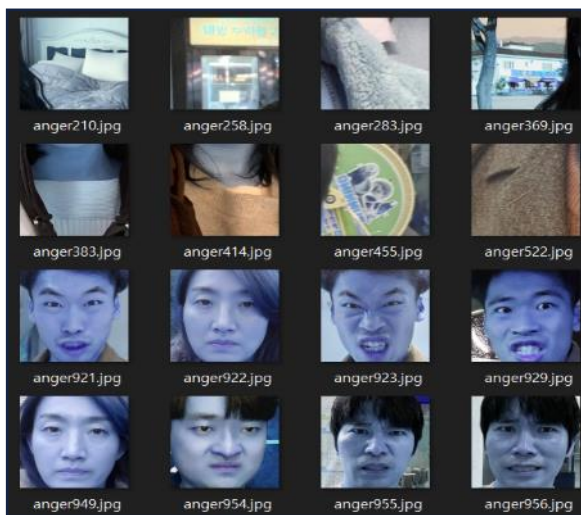
scalefactor : crop 결과 이미지 크기가 축소되는 정도

minNeighbors : 검출 영역으로 선택되기 위한 최소 검출 횟수(default : 5)

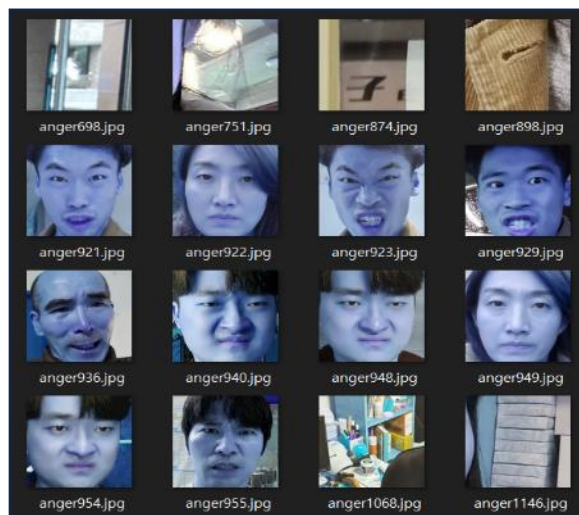
minSize : 가능한 최소 개체 크기 그보다 작은 개체는 무시

maxSize : 가능한 최대 개체 크기 그보다 큰 객체는 무시

default



minSize [30, 30]

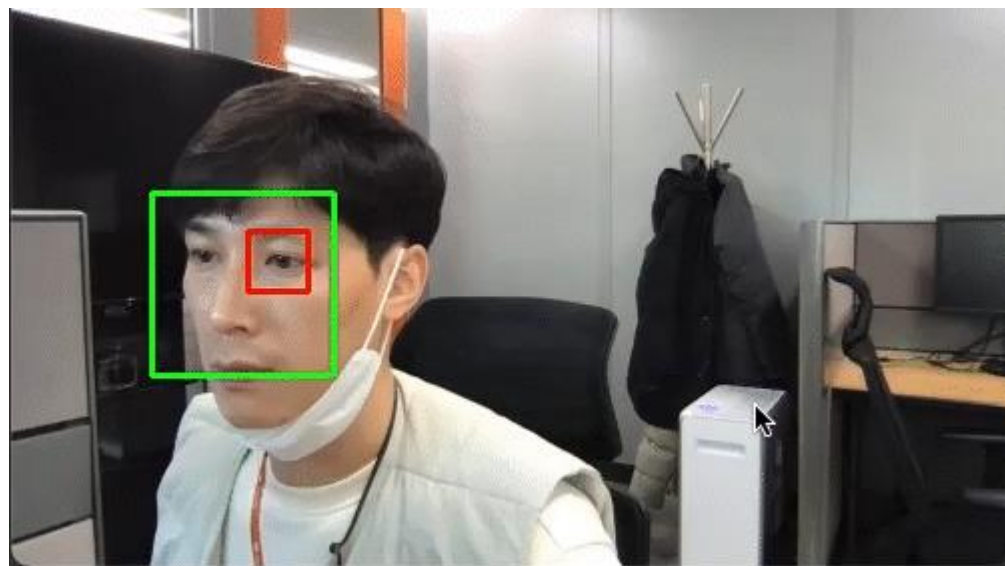


maxSize [500, 500]



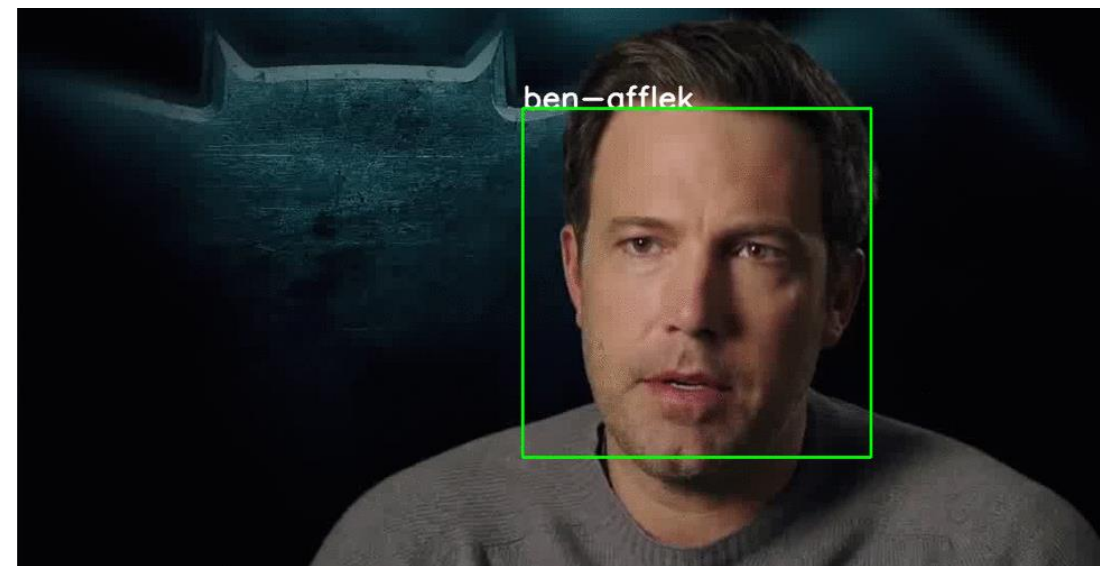
데이터 전처리 - Lite Face Detection 비교

Face_cascade.detectMultiScale



실시간 얼굴 탐지 불가능

ML kit / tensorflow js

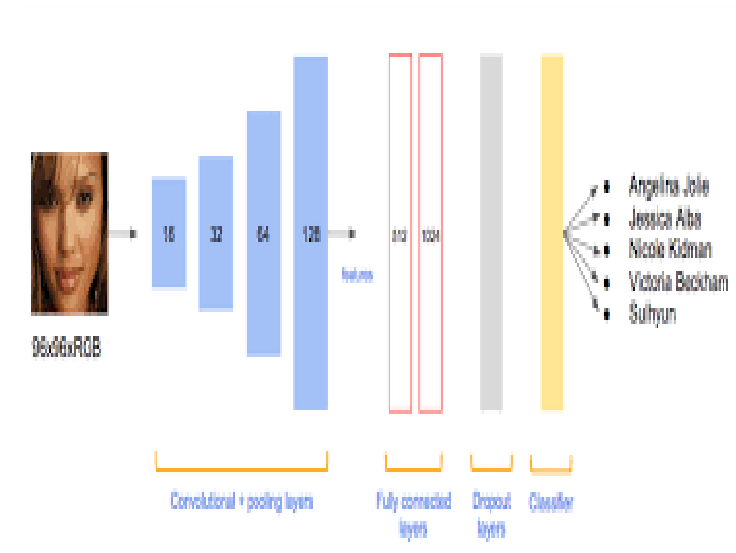


실시간 얼굴 탐지 가능

감정 인식 모델

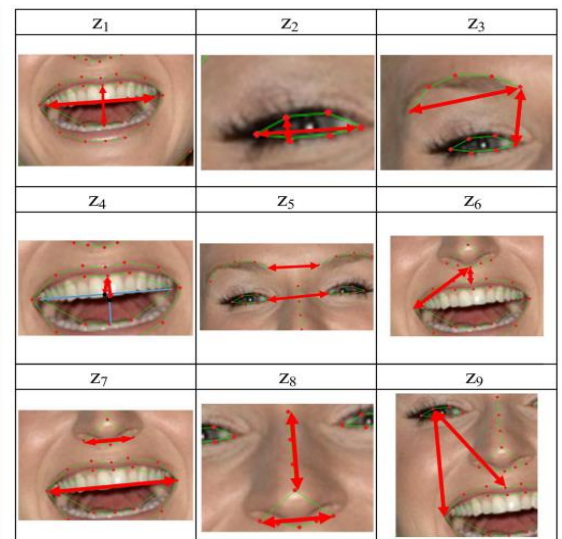
Original CNN X

부적합: 감정이란 내면의 주관에 영향을 받으므로 픽셀에 따른 분류는 부적합



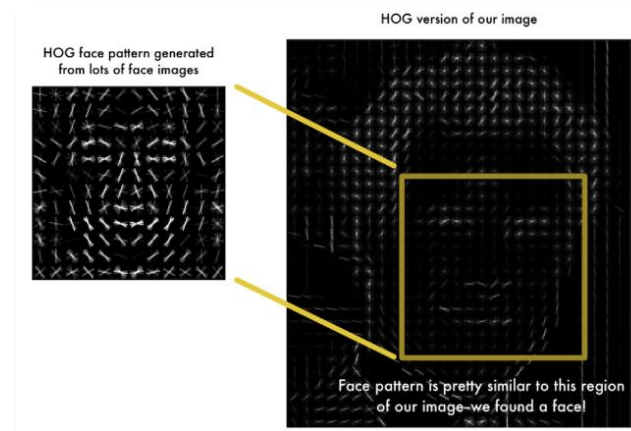
Facial Expression using Distance Features X

부적합: 모델 정확도 86%로 분류 모델보다 높은 편, 그러나 데이터의 각 column을 직접 제작해야 하므로 신빙성이 떨어짐



Histogram of Oriented Gradient for human detecting X

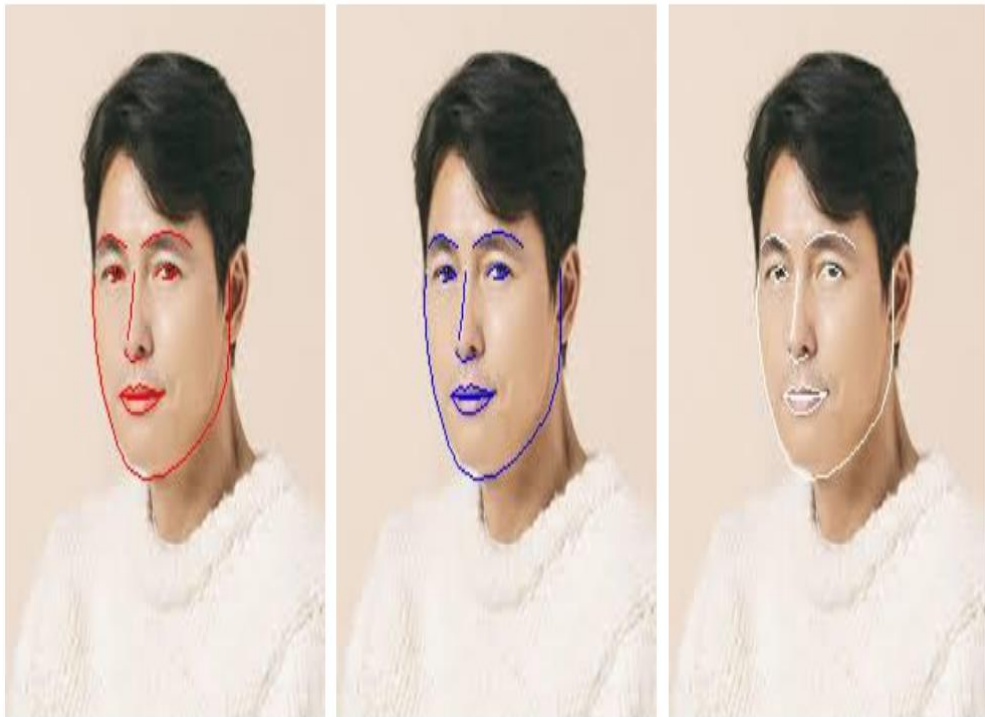
부적합: 과거 딥러닝이 발전하기 전에 사용하던 방식으로 sota 모델 정확도가 67%로 낮음



감정 인식 모델

Draw lines and classify Landmarks X

부적합: 임의로 시도해본 결과 모델 정확도 25%



Facial-expression-recognition-using-cnn

적합: 딥러닝 sota 모델로 모델 정확도가 70%
=> 하지만 코드를 수집한 데이터에 맞게 수정할 수 있음
모델 변경도 가능하여 개선 가능성 有



감정 인식 모델 - Facial-expression-recognition-using-cnn

Step 1

랜드마크 추출

얼굴에서 Landmarks를 뽑아
Feature Extraction 진행
기존 51개 -> 34개 랜드마크 축소

```
new_landmarks=[ 'x17', 'x18', 'x19', 'x21', 'x22', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35',
'x36', 'x37', 'x38', 'x39', 'x40', 'x41', 'x42', 'x43', 'x44', 'x45', 'x46', 'x47', 'x48', 'x49', 'x50', 'x51', 'x52', 'x53', 'x54',
'x55', 'x56', 'x57', 'x58', 'x59', 'x60', 'x61', 'x62', 'x63', 'x64', 'x65', 'x66', 'x67',
'y17', 'y18', 'y19', 'y21', 'y22', 'y24', 'y25', 'y26', 'y27', 'y28', 'y29', 'y30', 'y31', 'y32', 'y33', 'y34', 'y35',
'y36', 'y37', 'y38', 'y39', 'y40', 'y41', 'y42', 'y43', 'y44', 'y45', 'y46', 'y47', 'y48', 'y49', 'y50', 'y51', 'y52', 'y53', 'y54',
'y55', 'y56', 'y57', 'y58', 'y59', 'y60', 'y61', 'y62', 'y63', 'y64', 'y65', 'y66', 'y67',
'd_cent19', 'd_cent19', 'd_cent19', 'd_cent21', 'd_cent22', 'd_cent23', 'd_cent24', 'd_cent25', 'd_cent26',
'd_cent27', 'd_cent28', 'd_cent29', 'd_cent30', 'd_cent31', 'd_cent32', 'd_cent33', 'd_cent34', 'd_cent35',
'd_cent36', 'd_cent37', 'd_cent38', 'd_cent39', 'd_cent40', 'd_cent41', 'd_cent42', 'd_cent43', 'd_cent44', 'd_cent45',
'd_cent46', 'd_cent47', 'd_cent48', 'd_cent49', 'd_cent50', 'd_cent51', 'd_cent52', 'd_cent53', 'd_cent54',
'd_cent55', 'd_cent56', 'd_cent57', 'd_cent58', 'd_cent59', 'd_cent60', 'd_cent61', 'd_cent62', 'd_cent63', 'd_cent64',
'd_cent65', 'd_cent66', 'd_cent67',
'a_cent17', 'a_cent18', 'a_cent19', 'a_cent20', 'a_cent21', 'a_cent22', 'a_cent23', 'a_cent24', 'a_cent25', 'a_cent26',
'a_cent27', 'a_cent28', 'a_cent29', 'a_cent30', 'a_cent31', 'a_cent32', 'a_cent33', 'a_cent34', 'a_cent35',
'a_cent36', 'a_cent37', 'a_cent38', 'a_cent39', 'a_cent40', 'a_cent41', 'a_cent42', 'a_cent43', 'a_cent44', 'a_cent45',
'a_cent46', 'a_cent47', 'a_cent48', 'a_cent49', 'a_cent50', 'a_cent51', 'a_cent52', 'a_cent53', 'a_cent54',
'a_cent55', 'a_cent56', 'a_cent57', 'a_cent58', 'a_cent59', 'a_cent60', 'a_cent61', 'a_cent62', 'a_cent63', 'a_cent64',
'a_cent65', 'a_cent66', 'a_cent67 ]
```

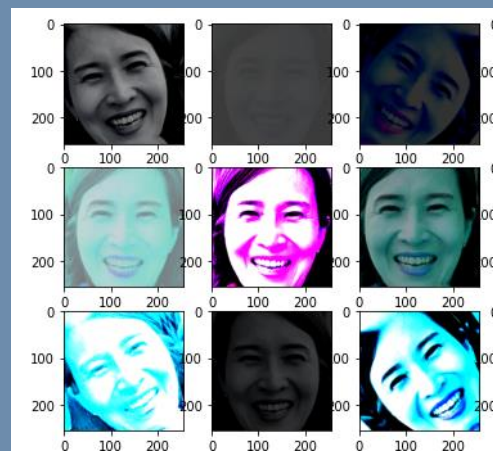


```
new_landmarks=[ 'x18', 'x19', 'x20', 'x21', 'x22', 'x23', 'x24', 'x25', 'x26', 'x27', 'x28', 'x29', 'x30', 'x31', 'x32', 'x33', 'x34',
'x35', 'x36', 'x37', 'x38', 'x39',
'y18', 'y19', 'y20', 'y21', 'y22', 'y23', 'y24', 'y25', 'y26', 'y27', 'y28', 'y29', 'y30', 'y31', 'y32', 'y33', 'y34',
'y35', 'y36', 'y37', 'y38', 'y39',
'd_cent20', 'd_cent21', 'd_cent22', 'd_cent23', 'd_cent24', 'd_cent25', 'd_cent26', 'd_cent27', 'd_cent28', 'd_cent29',
'd_cent30', 'd_cent31', 'd_cent32', 'd_cent33', 'd_cent34', 'd_cent35',
'a_cent20', 'a_cent21', 'a_cent22', 'a_cent23', 'a_cent24', 'a_cent25', 'a_cent26', 'a_cent27', 'a_cent28', 'a_cent29',
'a_cent30', 'a_cent31', 'a_cent32', 'a_cent33', 'a_cent34', 'a_cent35',
'a_cent36', 'a_cent37', 'a_cent38', 'a_cent39', 'a_cent40', 'a_cent41', 'a_cent42', 'a_cent43', 'a_cent44', 'a_cent45',
'a_cent46', 'a_cent47', 'a_cent48', 'a_cent49', 'a_cent50', 'a_cent51', 'a_cent52', 'a_cent53', 'a_cent54',
'a_cent55', 'a_cent56', 'a_cent57', 'a_cent58', 'a_cent59' ]
```

Step 2

Data Augmentation

각도, 밝기, 그림자 위주 증강



Step 3

Base model 변경

Original CNN을 Efficientnet으로 변경

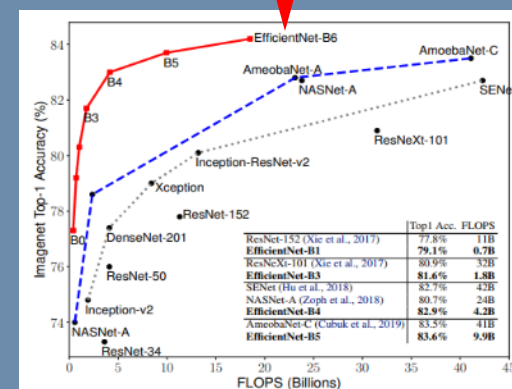


Figure 5. FLOPs vs. ImageNet Accuracy – Similar to Figure 1 except it compares FLOPs rather than model size.

감정 인식 모델 - Facial-expression-recognition-using-cnn

※ 약 5만개의 데이터로 6개의 Multi Classification 모델 훈련



원본 이미지



Face Detection
crop 이미지



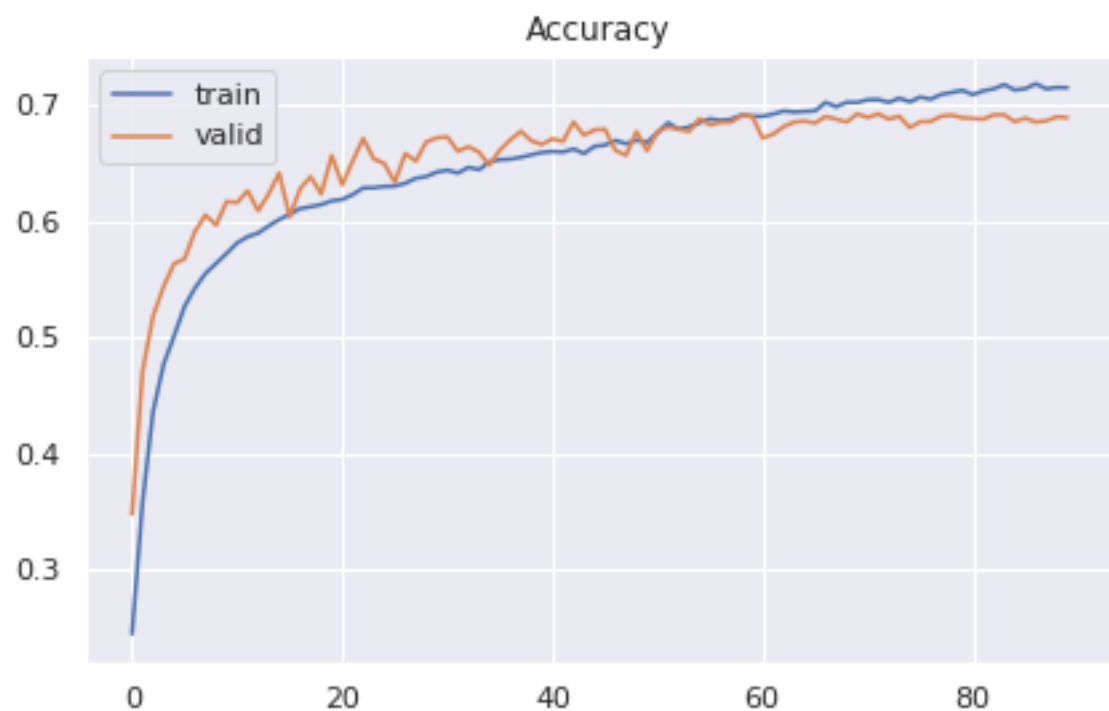
Face Detection
+ Landmarks



“neutral”

전처리 이미지
Efficientnet에 넣기

감정 인식 모델 - 모델 평가 : Accuracy

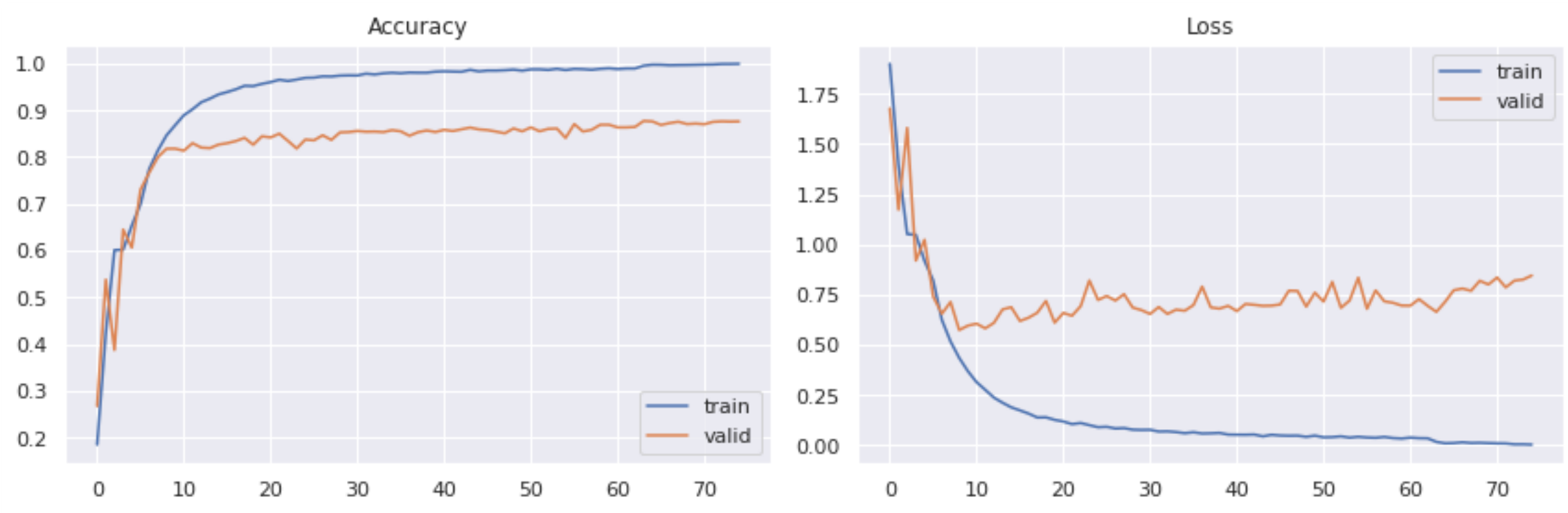


약 70%의 정확도



약 0.9의 오차

감정 인식 모델 - 모델 평가 : Accuracy



약 87%의 정확도(17% 향상)

약 0.7의 오차(0.2 개선)

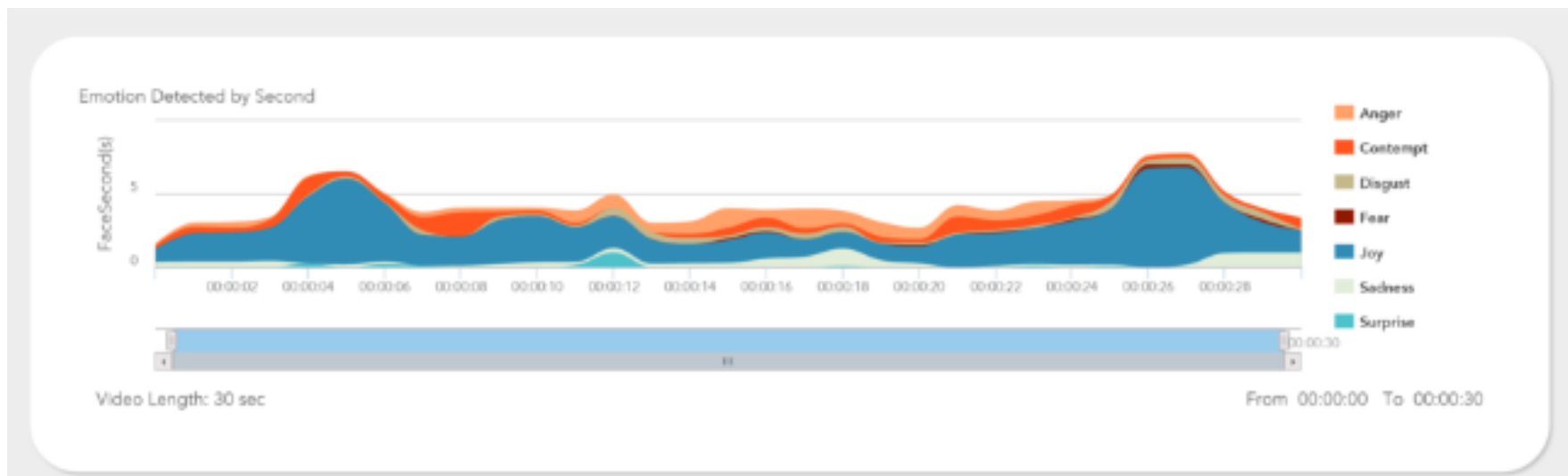
감정 인식 모델 - 향후 계획

한계점

- 얼굴 표정만으로 내면의 감정까지 정확하게 맵핑하는 어려움
- 촬영 시점에 따라 변화하는 감정을 반영하지 못함

추후 고도화

- 턱을 괴거나 머리를 쥐어짜는 등의 행동을 적용하여 감정 분류에 기여
- 시간에 따른 감정 변화를 구할 수 있다면 의미 있는 결과 도출 가능



감사합니다

