

당플리 

# 당신의 얼굴이 말하는 플레이리스트.

실시간 얼굴 감정 인식 기반 음악 추천 서비스



팔색조

변하영 / 송경준 / 최진수





# CONTENTS

01

## 팀 팔색조 소개

02

## 기획 의도

- 2020 음악 이용 실태
- 기존 음악 개인화 서비스 분석
- 감정 분석 시장 확대

03

## 프로젝트 설명

- 기술 스택
- 프로젝트 모델
- 당플리 기능 상세

04

## 시연 영상

05

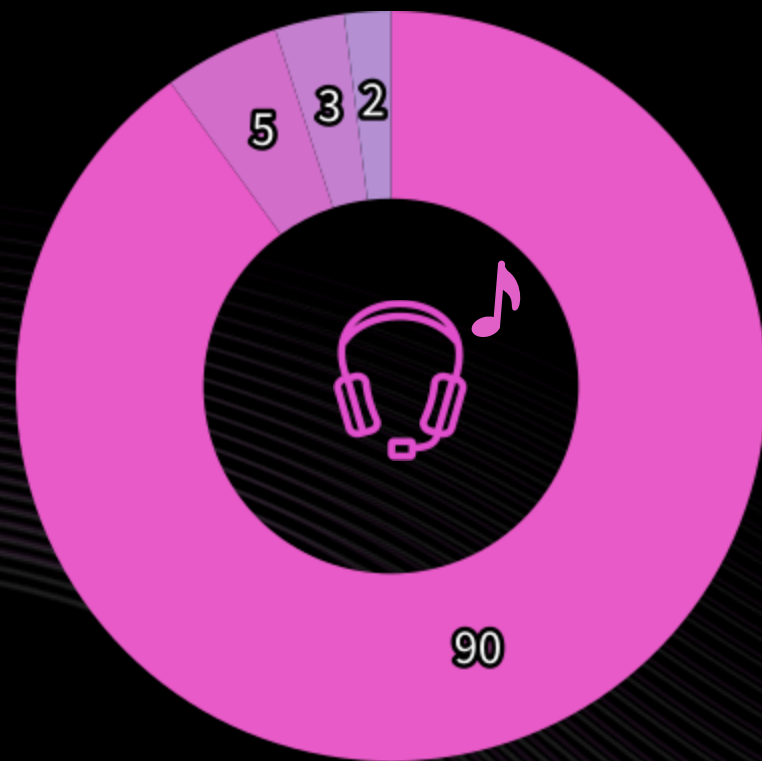
## 느낀점

01

# 팀 팔색조 소개



# Our Team Members



■ 매우만족 ■ 많이만족  
■ 엄청만족 ■ 베리만족

"코드짜기가  
제일 쉬웠어요"



송경준(조장)  
코드머신,  
햄버거 덕후,  
키보드

"이거 좀 더  
공부해볼까요?"



최진수  
논리킹 분석가,  
네비게이션,  
플룻

"새로운 아이디어는  
언제나 환영이야"



변하영  
프로 기획러,  
맛잘알,  
드럼

02

# 기획 의도

지금 당신의 표정이 말하는 음악을 들려드립니다.

2020년 음악  
이용자 실태

기존 음악 추천  
서비스 분석

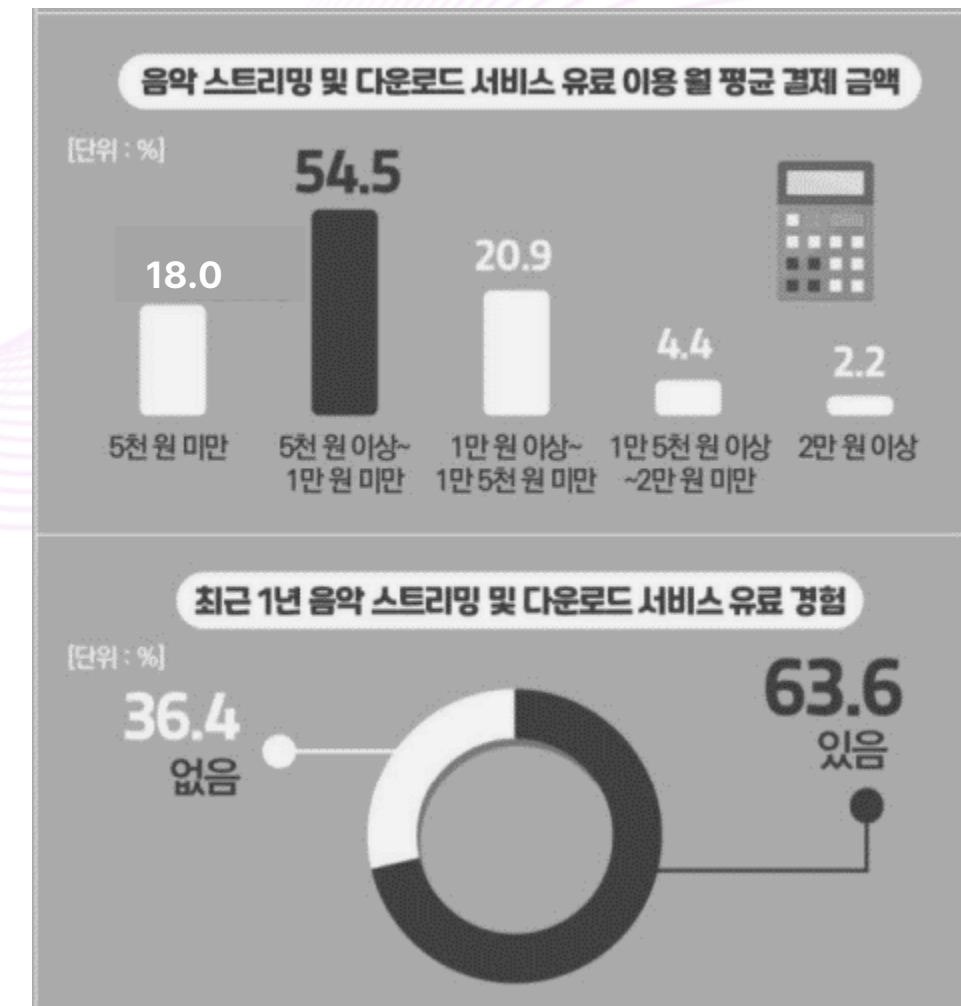
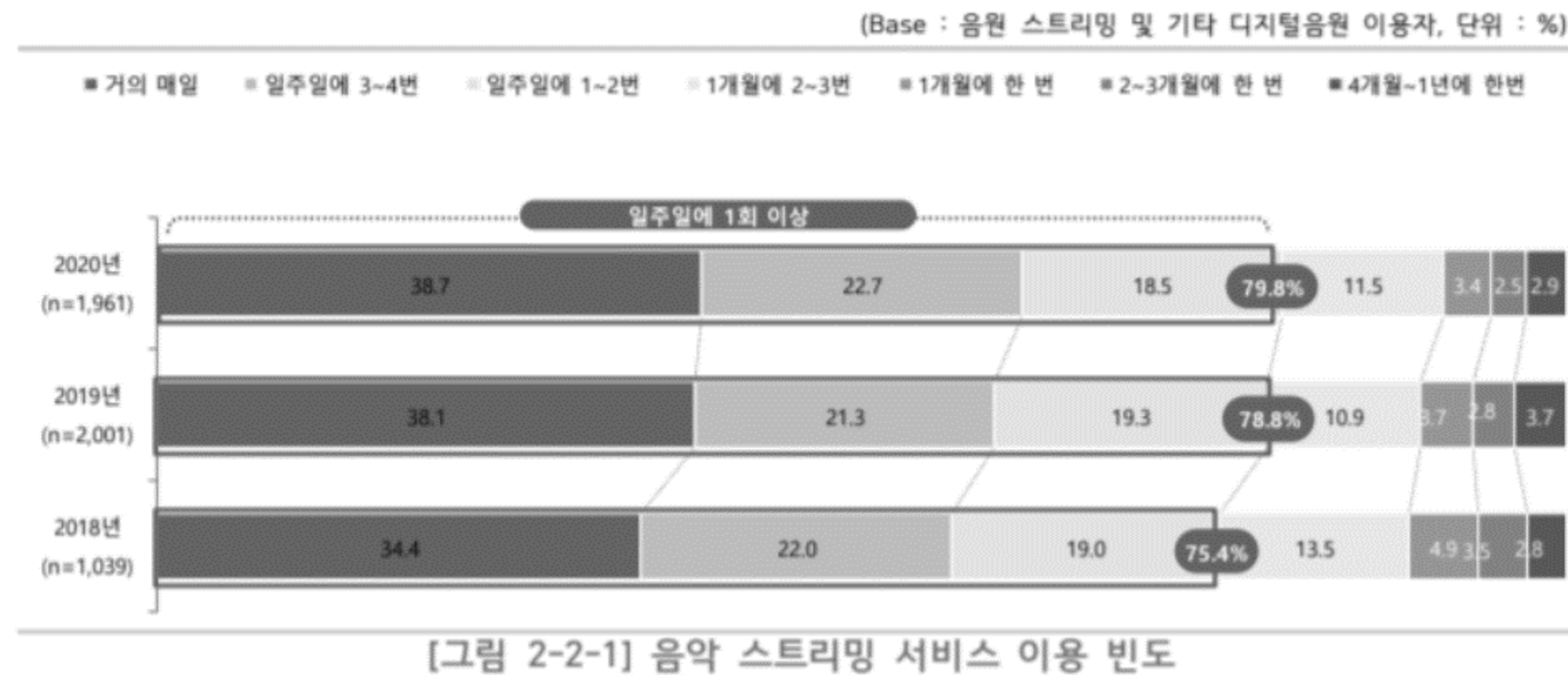
글로벌 감정 분석  
시장의 확대



# 우리가 어떤 민족입니까?

국민의 79.8%는 일주일에 1번 이상 음악 서비스를 사용하고,  
63.6%는 음악 서비스 유료 구매 경험이 있는 **흥의 민족**

한국콘텐츠진흥원, <2020 음악 이용자 실태조사>



# 음악 스트리밍 서비스의 메가트렌드, "초개인화 마케팅"

추천 알고리즘을 통해 실시간으로 **개인에게 정확히 맞춘 음악**을 제공

소비자신문 노수은 기자, <음악 스트리밍의 새로운 경쟁, "취향 따라 추천">(2021.11.01)>



## #글로벌 1위 #개인화 원조

- 선호 아티스트들의 믹스 음악 추천
- 전 세계 이용자와 청취 습관 비교, 분석
- 서비스 이용 시간대, 음악 청취 순서 고려



## #유사장르 음악 무한 추천

- 끝까지 들은 음악과 비슷한 장르 추천
  - '첫 14일의 추천' 서비스
- '비슷한 음악으로 jump' 기능 제공



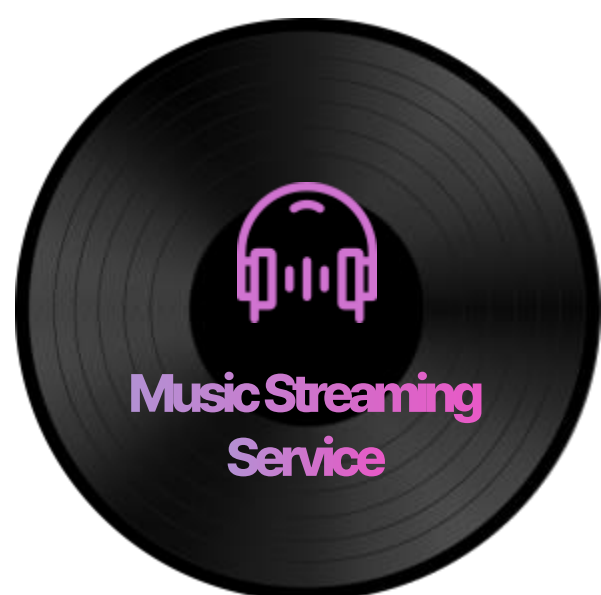
## #국내 1위 #빅데이터 보유

- 약 4,000만 곡, 800만 명 이상의 이용자 보유
- 감상 이력, 선호도 등 이용자 활동 내역 기반 추천
- '유사곡 추천' 기능 제공

# 기존의 음악 추천에 얼굴 감정 인식을 더하다!

2023년 45억 9,330만 달러로 전망되는  
글로벌 감정 분석 시장의 핵심인 **<얼굴 분석>** 시장 선점

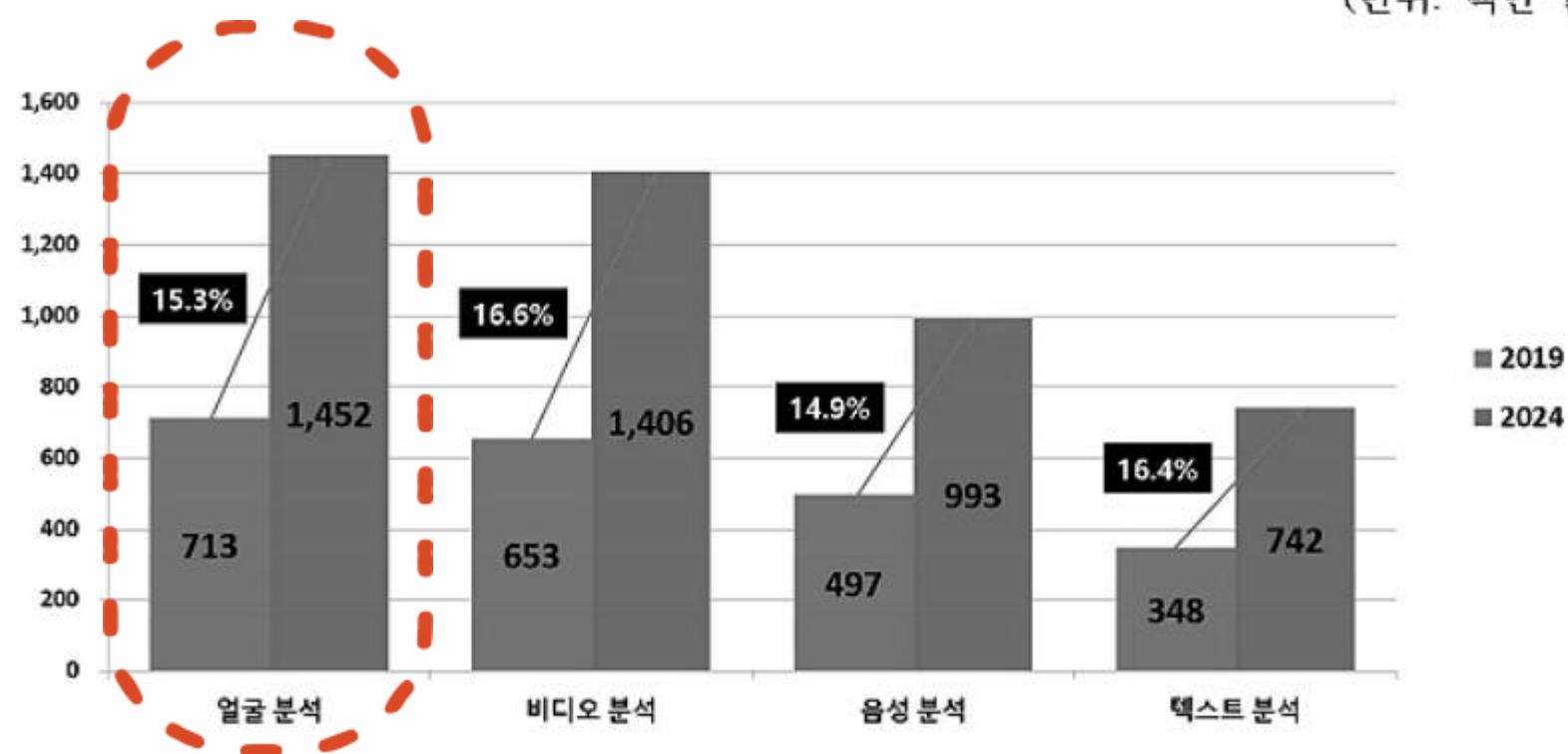
연구개발특구진흥재단, <글로벌 시장동향보고서 '감정 분석 시장(2021.04)'>



+

[그림 2-3] 글로벌 감정 분석 시장의 종류별 시장 규모 및 전망

(단위: 백만 달러)



※ 출처 : MarketsandMarkets, Emotion Analytics Market, 2019



03

# 프로젝트 설명

지금 당신의 표정이 말하는 음악을 들려드립니다.

기술 스택

프로젝트 모델

프로젝트 구조도

# 당플리 주요 기술 스택

- AI model



Pandas



Opencv



Tensorflow



Keras

- Backend



Django



SQLite

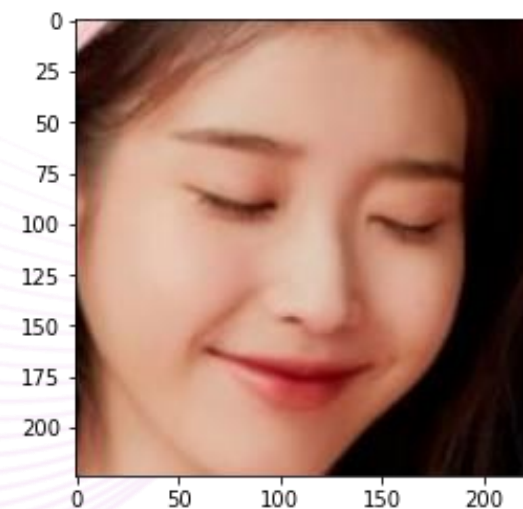


## 01 MTCNN 모델로 얼굴 인식



정확한 감정 검출을 위한 얼굴인식 진행

## 02 Custom CNN 모델로 감정 검출



angry: 0.060  
disgust: 0.006  
fear: 0.052  
**happy: 92.134**  
sad: 0.438  
surprise: 0.002  
neutral: 7.305

dominant\_emotion : **happy**

인식해낸 얼굴 이미지에서 가장 대표적인 감정을 사용

## 03 감정별 음악 플레이리스트 재생

항 목	happy
1번	아이유 - 라일락
2번	조이 - 안녕
3번	Pharrell Williams - Happy
4번	백예린 - november song
5번	Cheeze - madeleine Love

감지된 감정에 맞게 라벨링 해 둔  
음악 플레이리스트를 재생

# 01

## 조금 느려도 정확한 얼굴을 찾아내는 MTCNN 모델을 통한 얼굴 인식

Kaipeng Zhang외, <Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks(2016.04)>

기존 opencv에서 제공하는 CascadeClassifier 모델은 속도는 빠르지만 얼굴을 정확히 인식하는데 한계 확인



작은 사이즈의 얼굴까지 인식이 가능하고, SSD에 비해 속도는 느리더라도 더 정확한 결과를 보장하는 MTCNN 모델을 사용

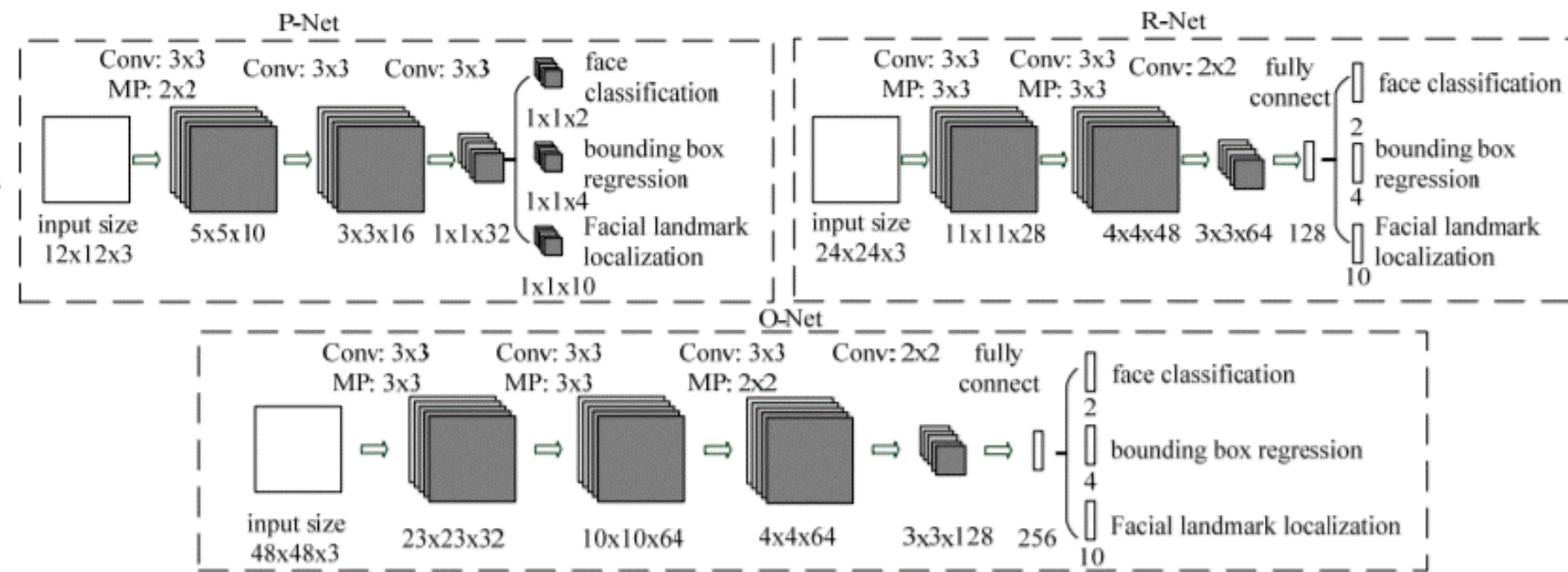


## 01

조금 느려도 정확한 얼굴을 찾아내는 **MTCNN 모델**을 통한 얼굴 인식

Kaipeng Zhang외, &lt;Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks(2016.04)&gt;

**P-net**  
이미지에서 얼굴을  
찾아내는데 중점을 두는 network



**R-net**  
P-net에서 찾아낸 후보 영역을  
추려내는데 중점을 두는 network

**O-net**  
얼굴 유무와 얼굴 좌표인  
face landmark를 찾아내는데  
중점을 두는 network

# 02 7개의 감정을 검출해낼 수 있는 Custom CNN 모델을 사용

Sefik Ilkin Serengil, <Kaggle Face Facial Expression Recognition Challenge 2013 data set >

## • Custom CNN 모델

3개의 CNN과 1개의 fully connected network로 이루어진 모델

```
model = Sequential()

#1st convolution layer
model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))

#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

model.add(Flatten())

#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(num_classes, activation='softmax'))
```

## • 정확도 테스트 결과 57%의 성능

열 : 예측값, 행 : 실제 감정 데이터

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	214	9	53	30	67	8	86
Disgust	10	24	9	2	6	0	5
Fear	45	2	208	29	89	45	78
Happy	24	0	40	696	37	18	80
Sad	65	3	83	56	285	10	151
Surprise	7	1	42	27	9	303	26
Neutral	45	2	68	65	88	8	331

Test loss: 2.27945706329  
Test accuracy: 57.4254667071

Train loss: 0.223031098232  
Train accuracy: 92.0512731201

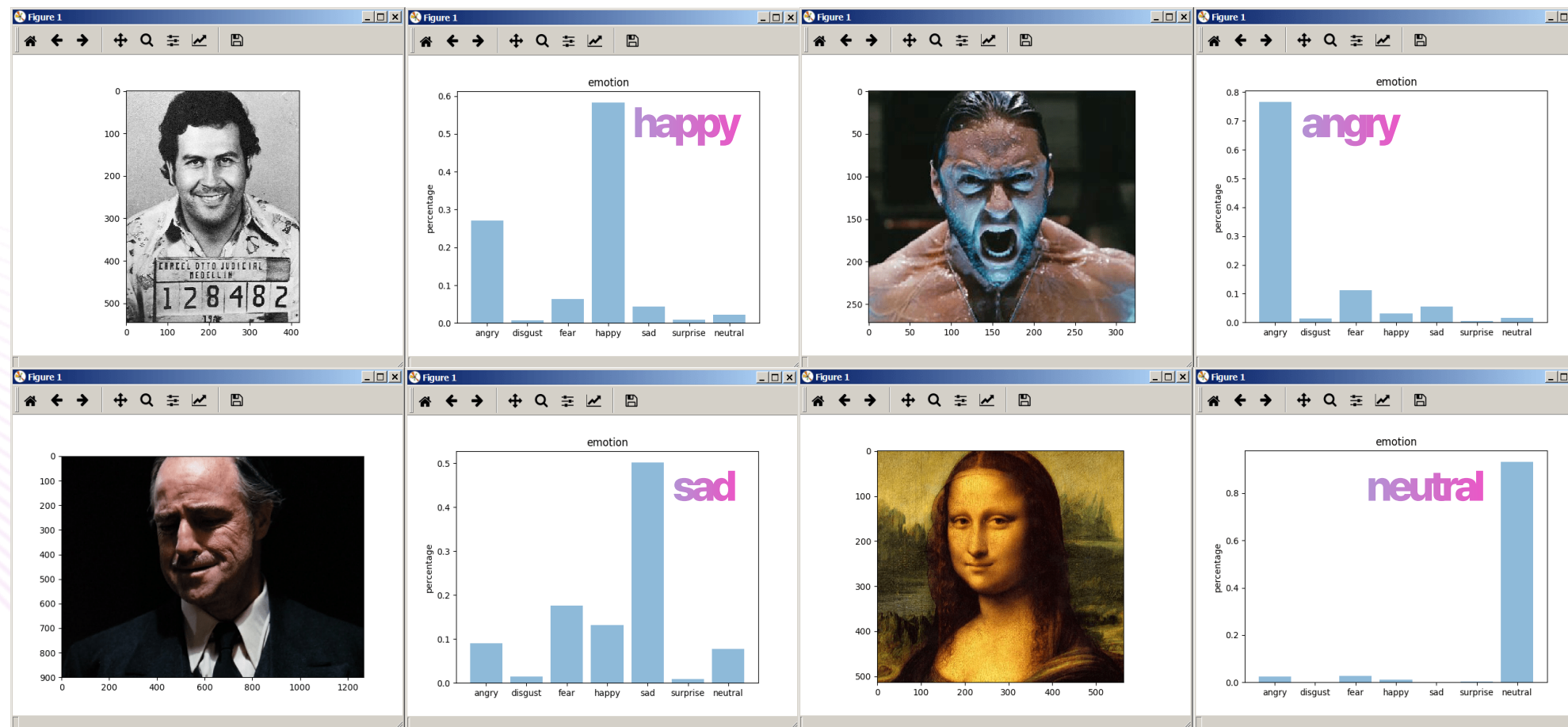


# 02 7개의 감정을 검출해낼 수 있는 Custom CNN 모델을 사용

Kaggle, <Kaggle Face Facial Expression Recognition Challenge 2013 data set>

kaggle 데이터를 활용한 감정 검출 예시 이미지

총 7개의 감정 검출(angry, disgust, fear, happy, sad, surprise, neutral)






48x48의 이미지 사이즈  
총 28,709개의 훈련데이터  
총 3,589개의 테스트데이터

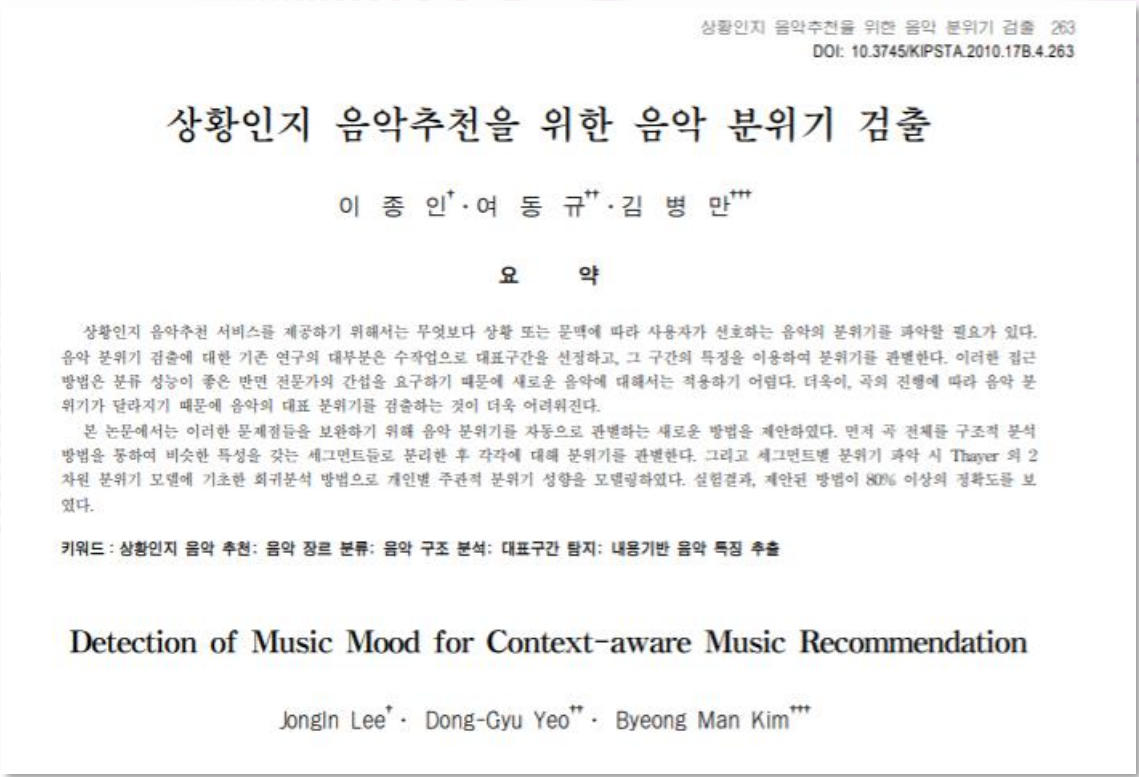
03

검출된 감정에 맞는 **음악 플레이리스트** 재생

이종인 외, <2010 "상황인지 음악추천을 위한 음악 분위기 검출">

• 감정에 맞게 임의로 설정한 음악 플레이리스트 재생

항 목	angry 	happy 	sad 	fear 
1번	힐링되는 유령송	아이유 - 라일락	백예린 그건 아마 우리의 잘못은 아닐꺼야	andre gagnon prologue
2번	반야심경 리믹스	조이 – 안녕	새소년 난춘	센과 치히로의 행방불명 생명의 이름
3번	mick gordon at doom's gate	Pharrell Williams - Happy	chet baker I fall in love too easily	마녀배달부키키 따스함에 둘러싸인다면 피아노
4번	죄임스 좇같은 인생	백예린 – november song	frank sinatra that's life	네가 되는 꿈 감정소묘
5번	넌 1분만 닥쳐줄래요	Cheeze – madeleine Love	daniel caesar japanese denim	all of me john smith piano



추후 단순 라벨링이 아니라  
음원 특징을 추출해 감정별로  
음악을 분류하는 논문 등을 참고,  
**음원에 따른 감정 예측으로 개선예정**



```
class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        (self.grabbed, self.frame) = self.video.read()
        threading.Thread(target=self.update, args=()).start()

        self.mtcnn_detector = MTCNN()
```

웹캠을 사용한 얼굴 촬영

```
def __del__(self):
    self.video.release()

def get_frame(self):
    img = self.frame

    detections = self.mtcnn_detector.detect_faces(img)
    # 얼굴 인식 -> detections 를 얻어내고
    # Song_list = pd.read_excel('./Song_list.xlsx')

    model = model_from_json(open("./model/facial_expression_model_structure.json", "r").read())
    model.load_weights('./model/facial_expression_model_weights.h5') # load weights
```

MTCNN을 활용한  
얼굴 인식

```
emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')

# 인식된 얼굴 잘라서 감정 추출 -> cropped_image, emotions
emotion_list = []
```

인식된 얼굴을 크롭 후  
Custom CNN 모델을  
통해 7개의 감정 감지

```
emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')  
  
emotion_list = []
```

검출된 감정은  
emotion\_list에서 처리

```
for i in range(len(predictions[0])):  
    emotion = "%s %s%s" % (emotions[i], round(predictions[0][i] * 100, 2), '%')  
    j = np.argmax(predictions[0])  
    k = emotions[j]  
    emotion_list.append(k)
```

예측한 감정은 순차적으로  
emotion\_list에 담기고  
20 frame 동안  
가장 많이 검출된 감정을 찾아냄

```
else:  
    b_emotion = pd.value_counts(emotion_list).head(1).index[0]  
    Song_list = UserSong.objects.filter(emotion=f'{b_emotion}')
```

가장 많이 검출된 감정을  
Songlist에 넣어  
해당 감정에 맞는  
플레이리스트를 재생



```
if emotion_list == []:  
    return render(request, 'Music_streamer/testcopy.html')
```

얼굴이 감지되지 않을 경우  
감지 실패 창으로 연결

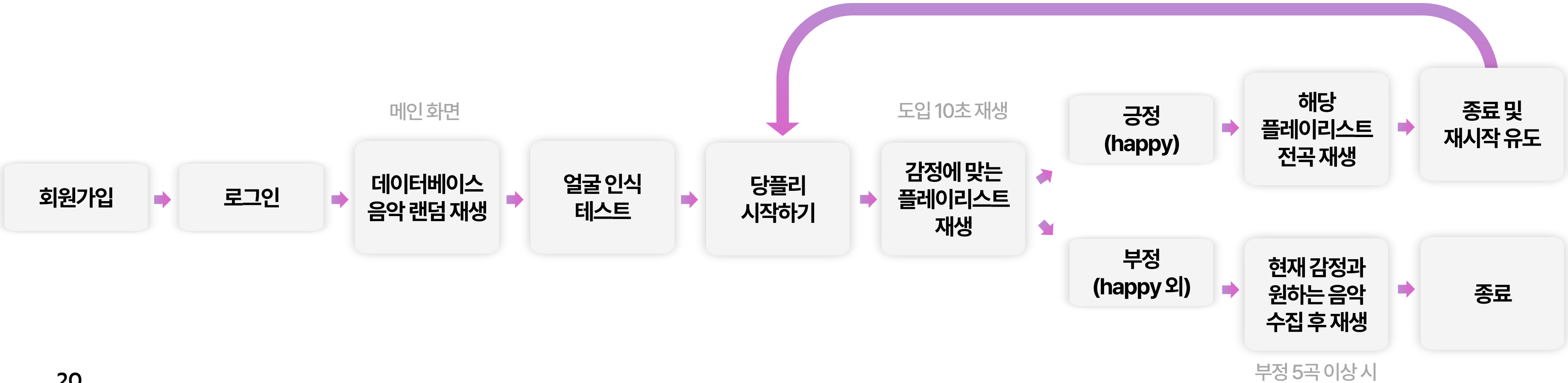
```
k = pd.value_counts(emotion_list).head(1).index[0]  
if k == 'happy':  
    # youtubeUrl_2 = driver.current_url[32:]  
    return render(request, 'Music_streamer/test.html')
```

음악 10초 재생 동안  
가장 많이 검출된 감정이 Happy면  
플레이리스트 재생 지속

```
elif k != 'happy':  
    player.pause()  
elif i == count:  
    player.pause()  
return render(request, 'Music_streamer/create.html')
```

해당 감정 플레이리스트 랜덤 재생 5곡에  
Happy 감정이 나오지 않으면  
음악 재생을 멈추고 지금 듣고 싶은 곡과  
감정 상태를 입력하는 창으로 연결

# 프로젝트 구조도





04

# 시연 영상



05

# 느낌점



# 느낀점

## 소감

현재 프로젝트를 수행하기까지 다양한 프로젝트를 진행하였습니다.

- (1) 음원의 멜스펙트로그램 데이터를 이미지로 학습 후 유사 음원 추천
- (2) 유튜브 썸네일로 영상 조회수 예측 및 제목 추천
- (3) 실시간 얼굴 감정 인식 기반 음악 추천 서비스(당플리)

이 중 실시간 얼굴 감정 인식 기반 음악 추천 서비스가 가장 매력적으로 느껴져 Final project로 선정하였고 머신러닝에 대한 깊은 경험을 할 수 있었습니다.

## 개선방향

1. 당플리 프로젝트는 현재 웹캠으로 제작되어 있어 유저의 사용환경을 고려해보았을 때 모바일 앱 구현이 필요하다.
2. 감정 인식 모델의 성능을 더욱 개선하여 정확도를 높인다.
3. 음원 특징을 추출해 감정별로 음악을 분류하는 논문 등을 참고, 음원에 따른 감정 예측으로 개선예정



당신의 얼굴이 말하는 플레이리스트



당플리