

12기 6팀

# Key-point detection을 활용한 K-POP 댄스가이드



박효정



김지우



이우림



문찬욱

# CONTENTS

---

1. 팀 구성 및 역할
2. EDA를 통한 비즈니스 인사이트
3. 데이터 설명
4. 모델과 학습 과정
5. 웹서비스 시연
6. 프로젝트 평가

# 1.팀 구성 및 역할

---

김지우 -> 팀장, 모델의 설계 및 학습, 막간의 프론트엔드

문찬욱 -> 데이터 베이스 구축, 웹 어플리케이션 구현, 배포

박효정 -> EDA를 통한 비즈니스 인사이트 도출

이우림 -> 데이터 전처리 및 모델의 학습

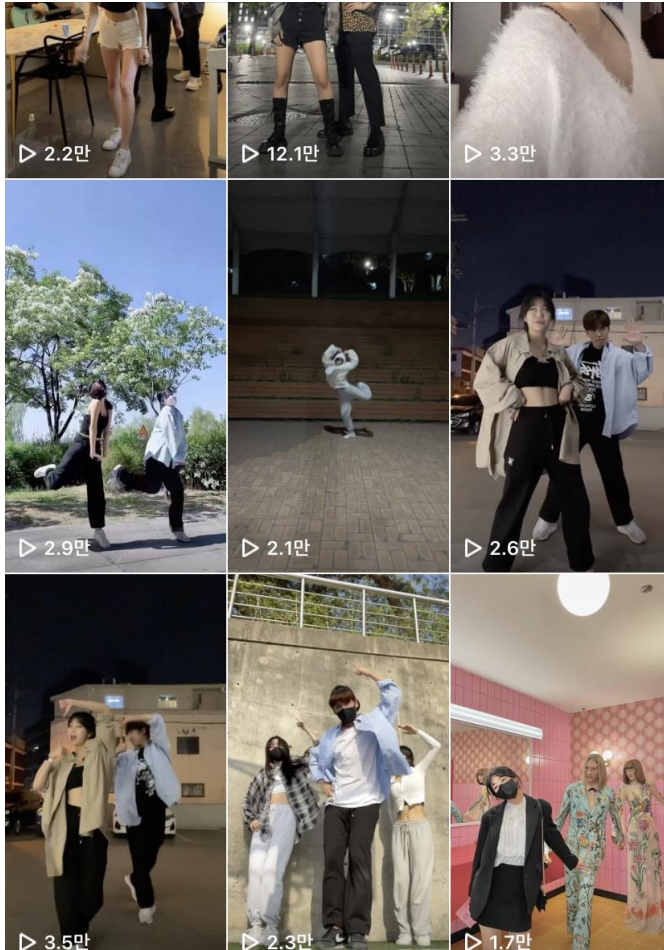
## 2.EDA를 통한 비즈니스 인사이트

---



2021 스트리트우먼파이터의 성공을 미루어 보았을 때  
대한민국 안에서 댄스 관련 콘텐츠에 대한 호감도가 매우 높다는 것을 파악할 수 있다.

## 2.EDA를 통한 비즈니스 인사이트



BUT 일상 콘텐츠와는 달리,  
일반인이 춤추는 콘텐츠는 아직 인플루언서 중심으로 생산된다.

그렇다면 일반인이 코인 노래방 가는 것처럼  
춤 추도록 자극하는 서비스가 있으면 어떨까?

춤에 대한 잠재적 수요를 자극하자!

<대학생 유튜버 darcie.j 의 인별릴스>

## 2.EDA를 통한 비즈니스 인사이트

---

우리 서비스의 존재 이유를 증명할 가설

우리 서비스를 이용할 **잠재적인 니즈**는 확실한가?

기업은 **우리 서비스에 투자할 이유**가 있는가?

## 2.EDA를 통한 비즈니스 인사이트

---



가설을 검증하기 위해

Spotify api에서 추출된 데이터를 바탕으로

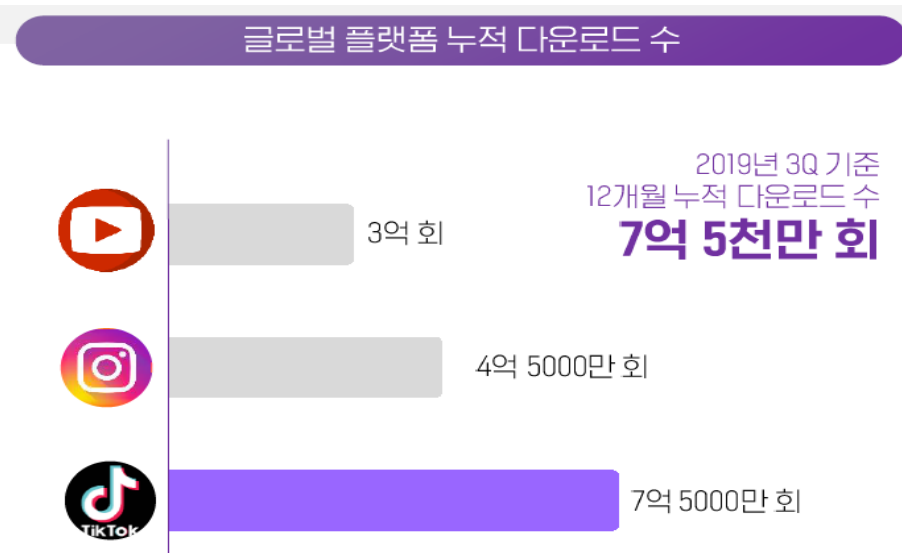
간단한 데이터 분석과 시각화를 진행했다!

-> 음악의 인기도나, 음악적 특징을  
수치화했기 때문에 분석에 적절했다고 판단

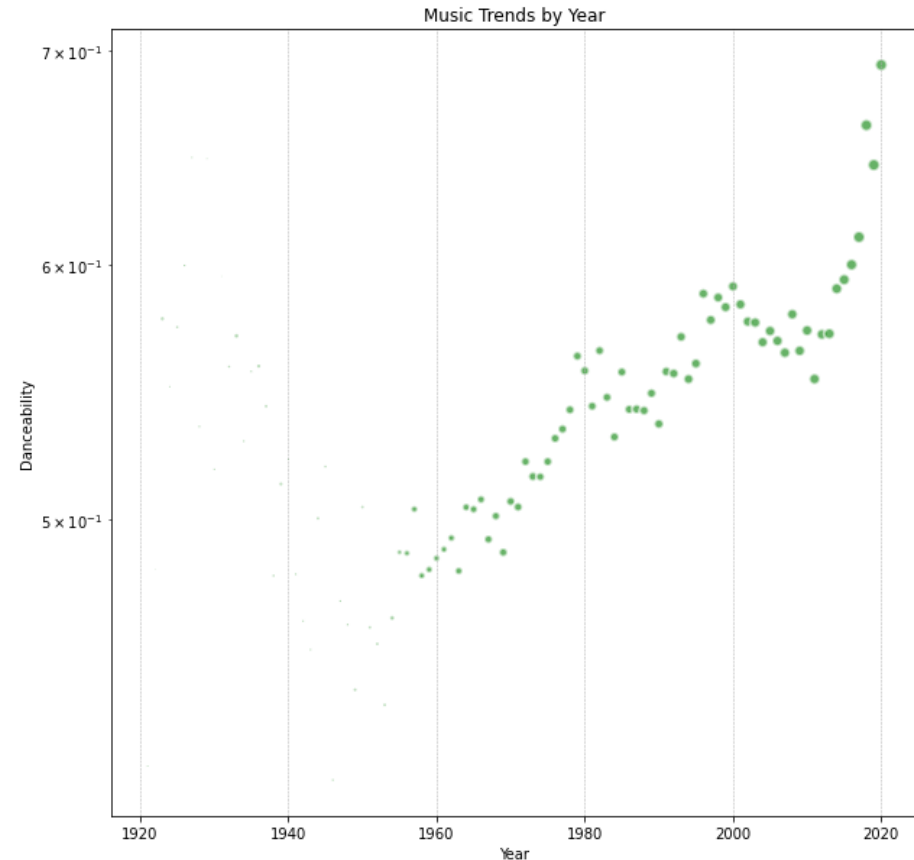
+ 전세계인의 보편적 취향을 점검하기도 용이

## 2.EDA를 통한 비즈니스 인사이트

### <분석 1>



\* 출처 : 센서타워



[영상으로 표현하는 MZ세대]

최근 틱톡, 숏츠, 릴스 등 평균 15~60초, 최대 10분을 넘기지 않는 동영상 콘텐츠 '숏폼' 성행  
분석 결과, 해가 지남에 따라 댄스음악적 성향의 높아짐->인기도 역시 상승 경향



## 2.EDA를 통한 비즈니스 인사이트

OLS Regression Results

Dep. Variable:	popularity	R-squared:	0.499
Model:	OLS	Adj. R-squared:	0.435
Method:	Least Squares	F-statistic:	7.755
Date:	Tue, 05 Jul 2022	Prob (F-statistic):	8.06e-08
Time:	17:35:15	Log-Likelihood:	-289.66
No. Observations:	80	AIC:	599.3
Df Residuals:	70	BIC:	623.1
Df Model:	9		

Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	95.2975	15.521	6.140	0.000	64.343	126.252
danceability	23.6888	11.858	1.998	0.050	0.038	47.339
valence	-28.1986	7.715	-3.655	0.000	-43.585	-12.812
acousticness	4.8597	7.463	0.651	0.517	-10.025	19.744
duration_ms	-0.0002	4.58e-05	-3.375	0.001	-0.000	-6.32e-05
liveness	9.3378	9.441	0.989	0.326	-9.491	28.166
loudness	0.5673	0.684	0.830	0.409	-0.796	1.931
speechiness	11.3803	19.246	0.591	0.556	-27.004	49.764
tempo	0.0050	0.030	0.169	0.867	-0.054	0.064
instrumentalness	-78.7611	21.058	-3.740	0.000	-120.759	-36.763
tempo	0.0050	0.030	0.169	0.867	-0.054	0.064

Omnibus: 49.797 Durbin-Watson: 1.955  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 227.801  
Skew: -1.856 Prob(JB): 3.42e-50  
Kurtosis: 10.386 Cond. No. 1.30e+19

### <분석2>

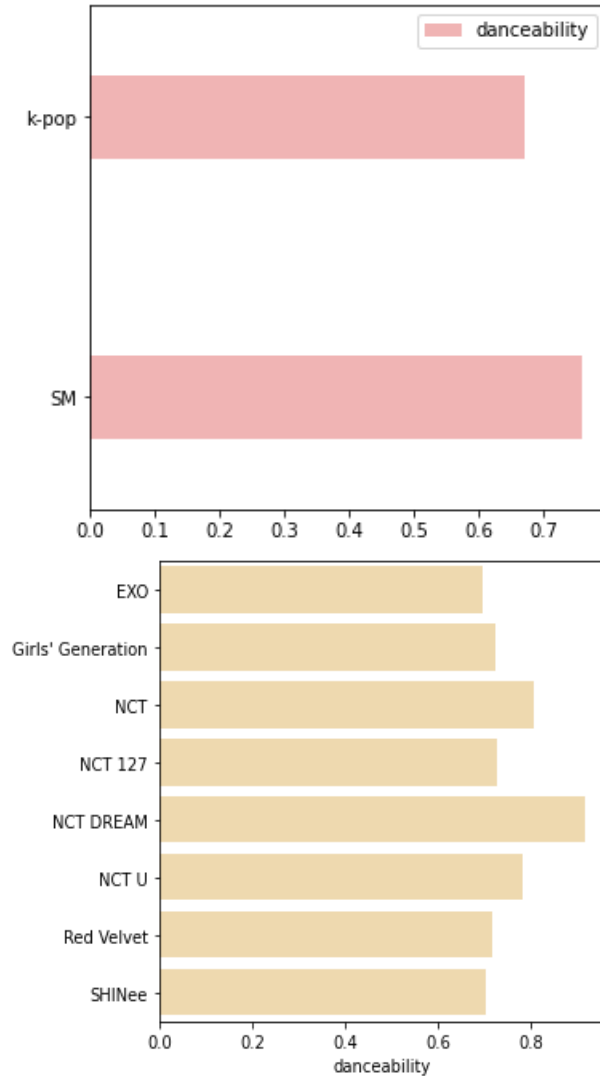
Danceability의 특징이 인기도에 얼마나 영향을 미칠지  
Popularity를 종속변수로 한 다중 선형 회귀 분석으로 확인!

Prob(F-statistic)이 8e-08로  
자체적으로 설정한 유의수준 0.01에 미치지 않아  
통계적으로 유의하다고 볼 수 있다.

Danceability의 회귀계수는 23.6888로  
양의 값을 갖는 변수 중 가장 큰 특징이다.

-> 추출 수 있는 음악인지는 대중의 니즈에 중요한 요소!

## 2.EDA를 통한 비즈니스 인사이트



### <분석3>

2010년대 K-POP을 이끌었던 가수들과 SM 소속 가수들과 비교해봤을 때,

SM 소속 가수들의 음악은 전반적으로 danceability 0.6 이상, SM은 댄스 음악이 주력인 기업

## 2.EDA를 통한 비즈니스 인사이트



### <결론>

- 댄스 형식의 숏폼 영상은 챌린지 형식으로 많이 소비됨
- 댄스 음악에 대한 대중의 니즈는 존재하고 -> 인기로 이어짐
- SM과 같이 댄스음악이 주력인 회사는 댄스 가이드 서비스로 대중을 자극하고, 더 많은 참여를 독려해 자사 콘텐츠를 홍보 할 수 있다.

따라서 우리 서비스 필요한 서비스라는 겁니다.

## 2.EDA를 통한 비즈니스 인사이트



그래서 하려는 게 뭐라고???

### KEY-POINT DETECTION

인체의 관절이나 중요부위의 좌표를 레이블로 삼아

사람의 포즈나, 관절 위치를 탐지하는 TASK

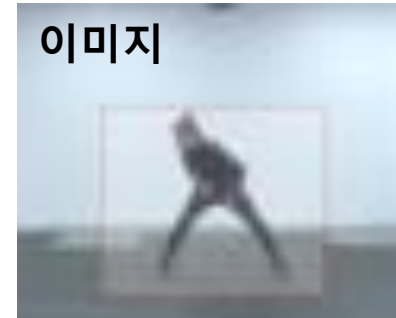
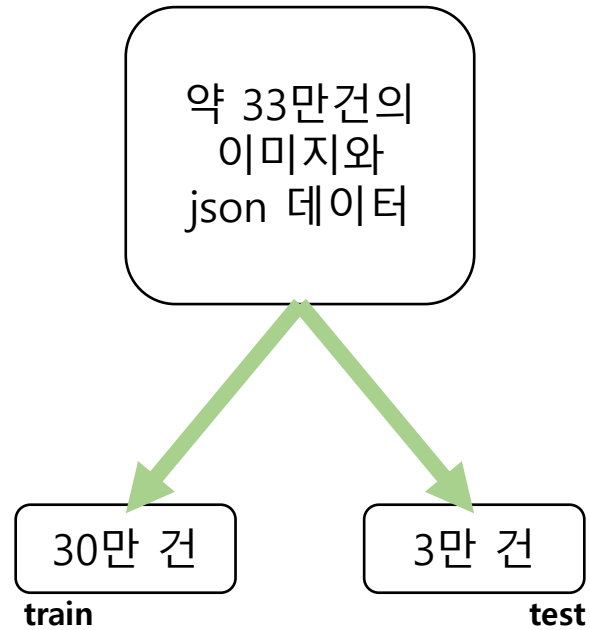
해당 TASK를 바탕으로 사람의 춤선을 시각화해,

춤에 대한 이해도를 높이려는 서비스

-> 댄스 가이드의 춤을 시각화, 내 춤을 시각화

### 3.데이터 설명

#### 1. AI Hub의 K-pop 안무 영상 데이터



```
"keypoints": [  
  368, 61, 1,  
  369, 52, 2,  
  369, 55, 1,  
  382, 48, 2,  
  350, 80, 1,  
  368, 84, 2,  
  435, 81, 2,  
  362, 125, 2,  
  .....  
  .....  
  397, 167, 1,  
  439, 166, 1,  
  369, 193, 2,  
  461, 234, 2,  
  361, 246, 2,  
  474, 287, 2,  
  371, 266, 2,  
  484, 277, 2  
],
```

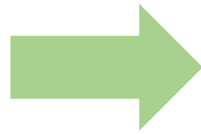
json

# 3.데이터 설명

## 2. json 형식의 데이터를 csv로 변환

```
"keypoints": [  
  368, 61, 1,  
  369, 52, 2,  
  369, 55, 1,  
  382, 48, 2,  
  350, 80, 1,  
  368, 84, 2,  
  435, 81, 2,  
  362, 125, 2,  
  .....  
  .....  
  397, 167, 1,  
  439, 166, 1,  
  369, 193, 2,  
  461, 234, 2,  
  361, 246, 2,  
  474, 287, 2,  
  371, 266, 2,  
  484, 277, 2  
],
```

json



CSV

```
1 ,image,center_hip_x,center_hip_y,left_hip_x,left_hip_y,left_knee_x,left_knee_y,left_ankle_x,left_ankle_y,left_bigtoe_X,left_bigtoe_Y  
2 0,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000000.jpg,1065,530,1108,532,1103,694,1108,834,1102,88  
3 1,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000002.jpg,1062,528,1107,528,1103,694,1106,845,1102,88  
4 2,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000004.jpg,1068,532,1105,531,1103,694,1106,848,1101,88  
5 3,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000006.jpg,1068,532,1109,532,1103,694,1106,842,1101,88  
6 4,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000008.jpg,1068,532,1105,532,1103,694,1103,839,1103,88  
7 5,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000010.jpg,1068,532,1107,532,1103,694,1106,838,1099,88  
8 6,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000012.jpg,1068,532,1102,532,1103,694,1106,843,1099,88  
9 7,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000014.jpg,1069,532,1110,533,1103,694,1105,839,1100,88  
10 8,./K-pop 안무 영상/Training/[원천]100 (슈퍼엠)/083_I00_001_F/083_I00_001_F_0000016.jpg,1069,532,1112,535,1103,694,1106,841,1103,87
```

이미지의 경로와 총 58개의 관절이 column으로 존재

➤ 총 59개의 column

## 3.데이터 설명

### 3. Column 소개 (1)



image	이미지 경로
"center_hip_x", "center_hip_y"	가운데 힙
"left_hip_x", "left_hip_y"	왼쪽 힙
"left_knee_x", "left_knee_y"	왼쪽 무릎
"left_ankle_x", "left_ankle_y"	왼쪽 발목
"left_bigtoe_X", "left_bigtoe_y"	왼쪽 엄지발가락
"left_littletoe_x", "left_littletoe_y"	왼쪽 새끼발가락
"right_hip_x", "right_hip_y"	오른쪽 힙
"right_knee_x", "right_knee_y"	오른쪽 무릎
"right_ankle_x", "right_ankle_y"	오른쪽 손목
"right_bigtoe_x", "right_bigtoe_y",	오른쪽 엄지발가락
"right_littletoe_x", "right_littletoe_y"	오른쪽 새끼발가락

## 3.데이터 설명

### 3. Column 소개 (2)



"navel_x", "navel_y"	배
"chest_x", "chest_y"	가슴
"neck_x", "neck_y"	목
"left_shoulder_x", "left_shoulder_y"	왼쪽 어깨
"left_elbow_x", "left_elbow_y"	왼쪽 팔꿈치
"left_wrist_x", "left_wrist_y"	왼쪽 손목
"left_palm_thumb_x", "left_palm_thumb_y"	왼쪽 엄지손가락
"left_palm_pinky_x", "left_palm_pinky_y",	왼쪽 새끼손가락
"right_shoulder_x", "right_shoulder_y"	오른쪽 어깨



## 3.데이터 설명

### 3. Column 소개 (3)

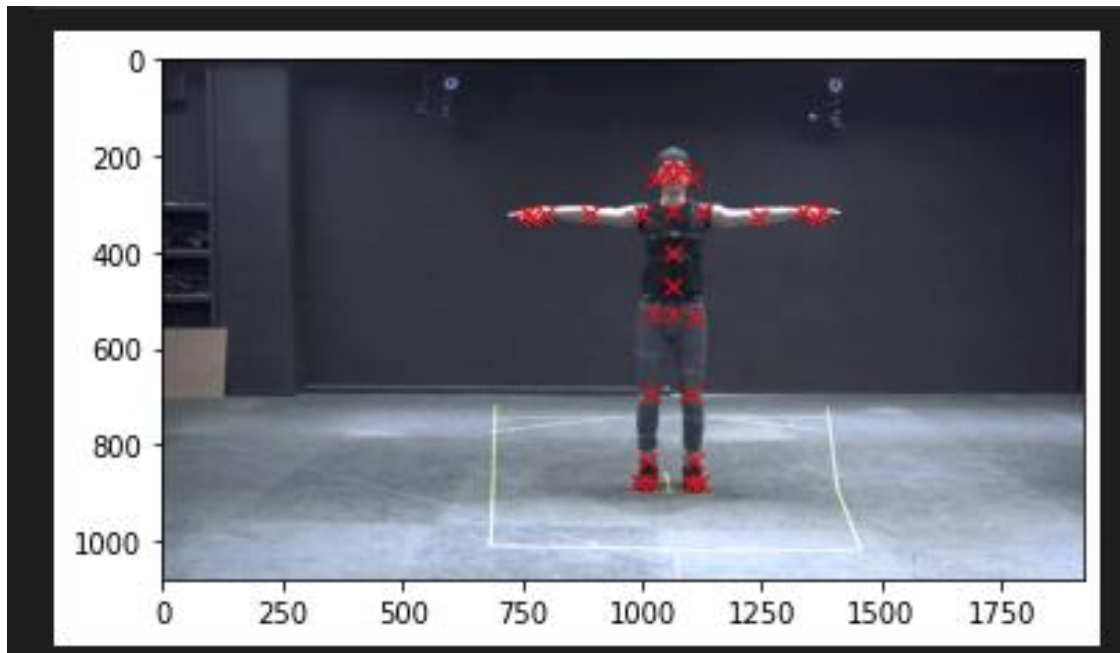


"right_elbow_x", "right_elbow_y"	오른쪽 팔꿈치
"right_wrist_x", "right_wrist_y"	오른쪽 손목
"right_palm_thumb_x", "right_palm_thumb_y"	오른쪽 엄지손가락
"right_palm_pinky_x", "right_palm_pinky_y"	오른쪽 새끼손가락
"nose_x", "nose_y",	코
"left_eye_x", "left_eye_y"	왼쪽 눈
"right_eye_x", "right_eye_y"	오른쪽 눈
"left_ear_x", "left_ear_y"	왼쪽 귀
"right_ear_x", "right_ear_y"	오른쪽 귀

### 3.데이터 설명

---

#### 4. 데이터 테스트 및 학습을 위한 전처리



(1) 학습을 위해서 180\*320으로 resize  
➤ **1080\*1920 -> 180\*320**



**좌표도 해상도 비율에 맞춰서 1/6**

(2) 이미지 정규화  
**모든 픽셀을 정규화 하기 위해 255로 나눴**

## 4. 모델과 학습과정

---

### 리소스 문제

1. 구글 드라이브에 올라가지 않는 방대한 데이터 양( 압축 해제 시 약 80기가)
2. GPU없이 작업할 때 발생하는 OOM
3. 오류가 나지 않지만 진행되지 않는 학습 속도 -> resnet152를 사용한 모델은 아예 진행x
4. CPU로 되다가도 GPU 설정만 하면 발생하는 호환성 오류

=> Hour Glass + U-Net , transformer를 접목시킨 SOTA모델들은 언감생심이었다.

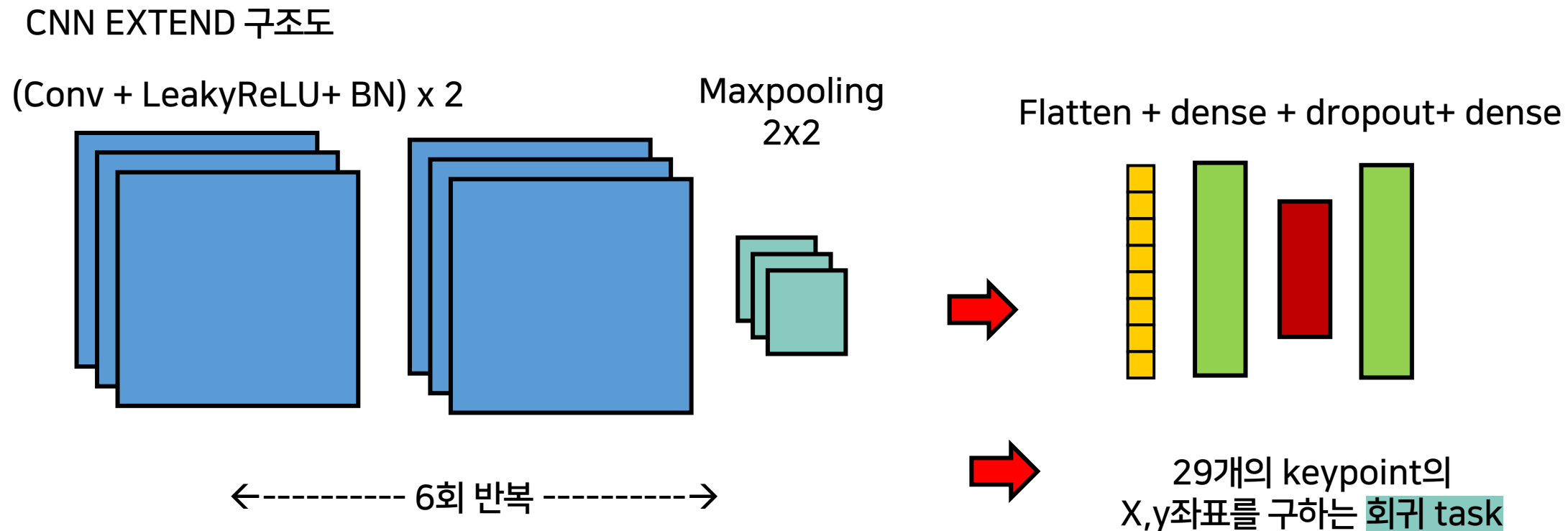
## 4. 모델과 학습과정

---

CNN EXTEND

리소스 문제에도 불구하고 결과를 내기 위해서는  
기본적인 CNN 모델을 최대한 응용해야겠다고 판단함.

## 4. 모델과 학습과정



1. SOTA 모델들에 비해 이미지 특성추출이 빈약
2. 1을 보완하기 위해 층이 깊어져서 생기는 문제 -> 기울기 소실, 증폭, 과적합

## 4. 모델과 학습과정

### CNN EXTEND code

```
model = Sequential()

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False, input_shape=(180,320,3)))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Convolution2D(96, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(96, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
```

### <이미지 특성 추출을 보완할 방안>

#### 1. Conv + Conv + Pool

Pooling layer 전에 conv를 반복하는 것이 이미지 특성 추출에 효과적이다.

출처: <An Introduction to Convolutional Neural Networks >

## 4. 모델과 학습과정

### CNN EXTEND code

```
model = Sequential()

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False, input_shape=(180,320,3)))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Convolution2D(96, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(96, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
```

<이미지 특성 추출을 보완할 방안>

### 2. MaxPooling2D

회귀 기반 keypoint detection에서  
Average Pooling을 사용하나,

이미지 특성 추출을 보완하기 위해  
MaxPooling사용

## 4. 모델과 학습과정

### CNN EXTEND code

```
model = Sequential()

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False, input_shape=(180,320,3)))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(32, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())

model.add(Convolution2D(64, (3,3), padding='same', use_bias=False))
model.add(LeakyReLU(alpha = 0.1))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
```

```
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(56))
```

<층이 깊어졌을 때의 문제를 보완>

1. LeakyReLU ->  
기울기 소실 문제를 보완
2. Dropout ->  
과적합 규제



## 4. 모델과 학습과정

---

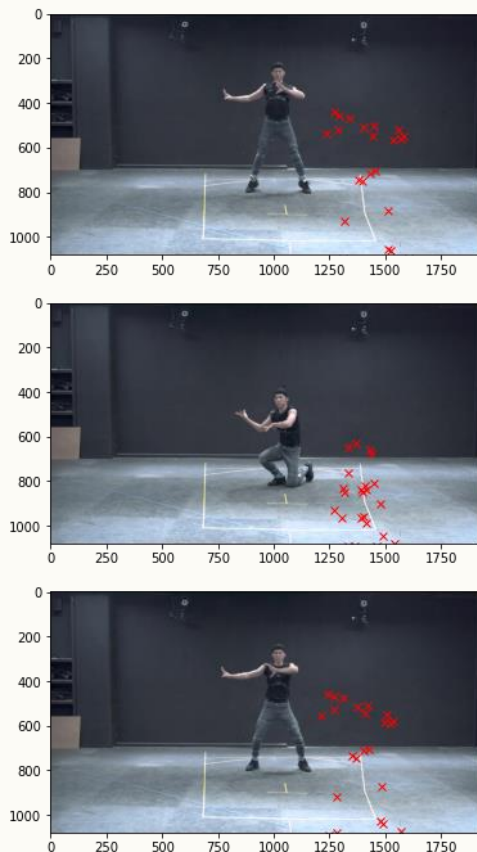
### 학습 과정

1. 이미지 데이터 2000개, epochs = 5 => mae : 66.8235 수행시간: 35분 (비고: target scalingx)
2. 이미지 데이터 8000개, epochs = 10 => mae : 13.8769 수행시간 : 5시간
3. 이미지 데이터 16000개, epochs = 15 => mae : 8.6318 수행시간 :14 시간
4. 이미지 데이터 32000개 , epochs = 20 => mae : 8.2247 수행시간 : 35 시간

Epochs은 5씩, 이미지 데이터는 1 -> 2를 제외하고 2배수 씩 증가  
-> 학습량 증가에 따라 수행시간은 증가하고, mae는 감소한다.

## 4. 모델과 학습과정

### 학습 과정의 유의사항



2000장의 이미지와 5epochs으로 돌린 결과

2가지 특이 사항 발견

1. Keypoint의 탐지 결과가 주로 **우하단에 편향됨**

-> input 이미지 resize 과정에서 발생한 문제

2. 탐지된 keypoint 들이 **사람의 모형과 비교해 있을 때 일그러짐**

-> 모델의 기본 이미지 추출 성능과 학습 데이터 양의 문제

# CONTENTS

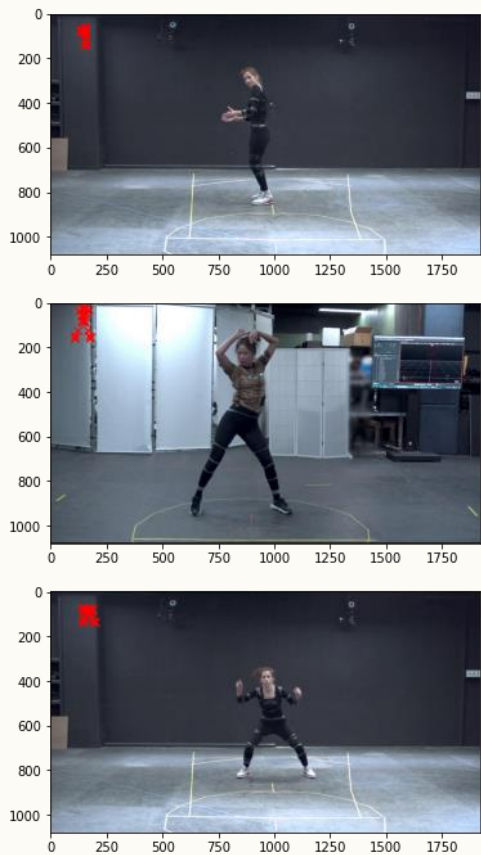
---

1번 문제를 풀어서 설명하자면?

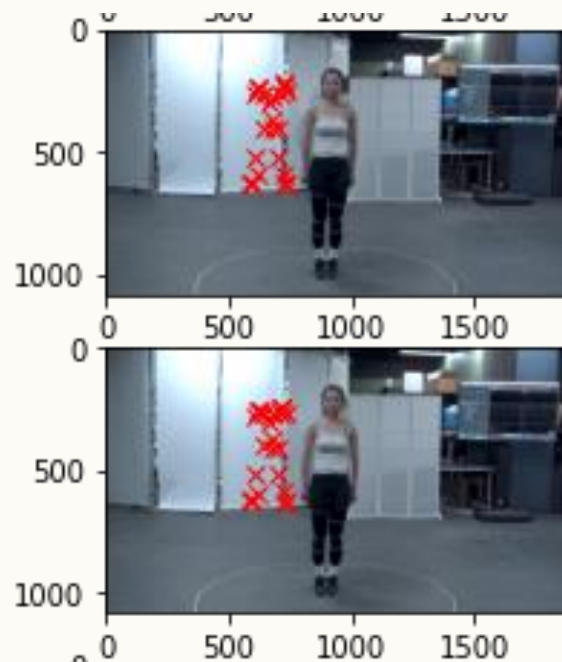
연산량 감소를 위해 1920x1080 픽셀의 이미지를  
320x180사이즈로 바꿨지만 keypoint는 변하지 않아 생기는 문제  
이에 학습 전 keypoint들을 모두 1/6하고  
학습 후 보정을 거치기로 한다.

## 4. 모델과 학습과정

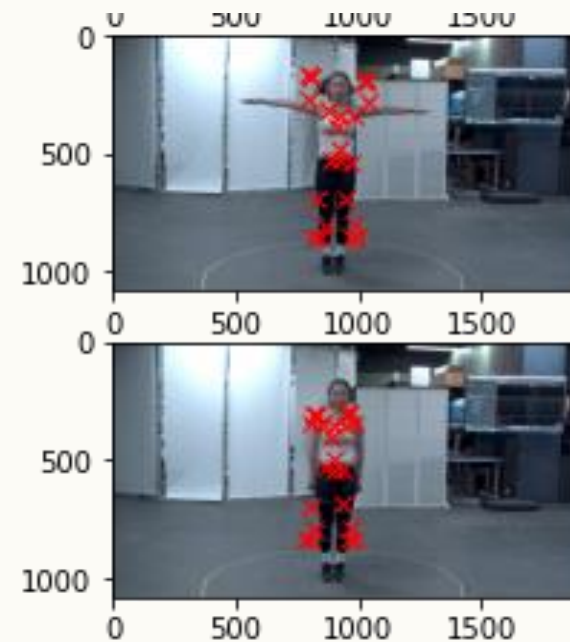
문제 해결을 위한 후보정 작업



후 보정의 배급이 높아질 수록 점들의 군집이 **우하향**,  
너무 적은 편차로 **왜곡된 결과를 퍼주는 효과**



<각 6배 곱 보정>

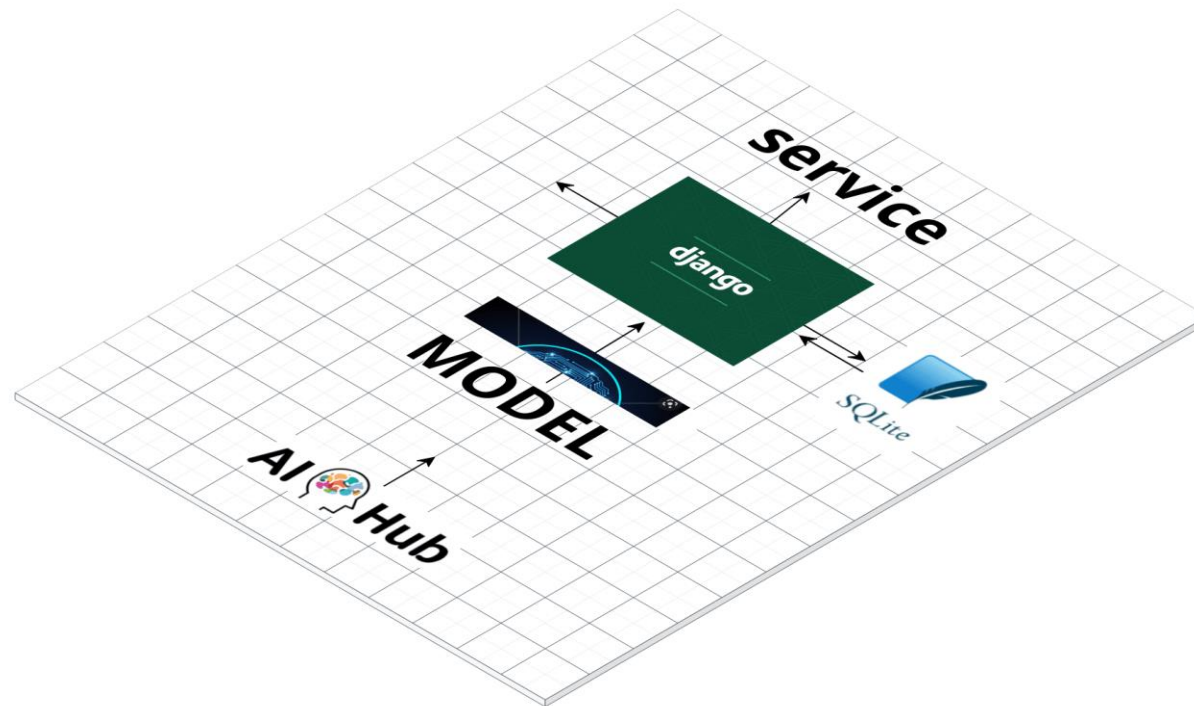


<각 8배 곱 보정>

## 5. 웹서비스 시연

---

Data - Pipeline



## 6. 프로젝트 평가

---

1. 당장 서비스 할 수 있는 수준은 아님 -> 정확도 문제, 탈골현상
2. 리소스 부족을 해결하면 모델의 교체도 가능하다.
  - GPU로 작업할 수 있게 되면, 전이학습모델, Hour glass, transformer 를 추가해 더 좋은 모델을 만들 수 있다.
3. 영상에 실시간으로 탐지 결과를 보여주는 방향으로 UX증가 기대
4. Spotify서비스의 특징을 고려해 보았을때, 해외 진출도 가능

---

*Thank you*