SW Code

Openpose를 활용한 뮤지컬 동작 분석

2020.10.13.~2020.12.04.

In python-jupyternotebook, window10

1. 과제 개요

- 과제 선정 배경 및 필요성
- 과제 주요내용

오픈포즈를 통한 뮤지컬별 동작 차이 분석
- openpose skeleton tracking 기술을 접하고 이를 사용해보기 위해 해당 프로젝트를 진행하게 되었다.

뮤지컬에는 대극장, 소극장, 중극장 등 출연하는 인원 또는 무대장치의 크기에 따라서 사용하는 극장이 다르다.
주연이 1명인 1인극부터 2, 3인극까지 소극장에서 공연을 하고, 대극장의 경우 주조연, 앙상블까지 포함하면
40명이 훌쩍 넘는 경우도 많다. (대극장 : 지킬앤하이드, 엘리자벳 등, 소극장 : 어쩌면해피엔딩, 키다리아저씨 등)

따라서, 대극장과 소극장의 동작 차이를
뮤지컬 속 "춤 " 을 통해 분석해보고자 한다.

*극장 크기에 따른 배우별 동작의 크기 차이, 이동량의 차이 등

2. 수행 방법

- 도구적 방법
- 과제 수행 계획

사용되는 기술
OpenPose - https://github.com/CMU-Perceptual-Computing-Lab/openpose
: Caffe와 OpenCV를 기반으로 구성된 손, 얼굴 포함 몸의 움직임을 추적해주는 API
Python, JupyterNotebook 사용

1차 목표는 openpose를 사용한 대극장과 소극장의 동작 크기 확인
~~기간이 된다면 기계학습을 통한 대극장, 소극장 분류 모델 제작.~~

3. 진행상황

- 현재 상황
- 진행 계획

순번	추진내용	10월	11월	12월
1	오픈포즈 및 기술 연구	o		
2	뮤지컬 영상 데이터 수집	o		
3	영상별 오픈포즈 수행	o	o	
4	키포인트 검출값 정리		o	o
5	대극장 소극장 차이 분석			o
6	데이터 전처리			
7	머신 러닝 돌려보기			
8	모델 검증			

아래 파트는 진행상황에 따라 계획이 변동될 수 있음.
추가적으로 디텍트론도 사용해볼 수 있었으면 좋겠다.

Pip 업그레이드

```
c:₩Users₩김서영₩AppData₩Local₩Programs₩Python₩Python37>pip install --upgrade pip
Collecting pip
  Downloading pip-20.2.3-py2.py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 547 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.1
    Uninstalling pip-20.1:
      Successfully uninstalled pip-20.1
```

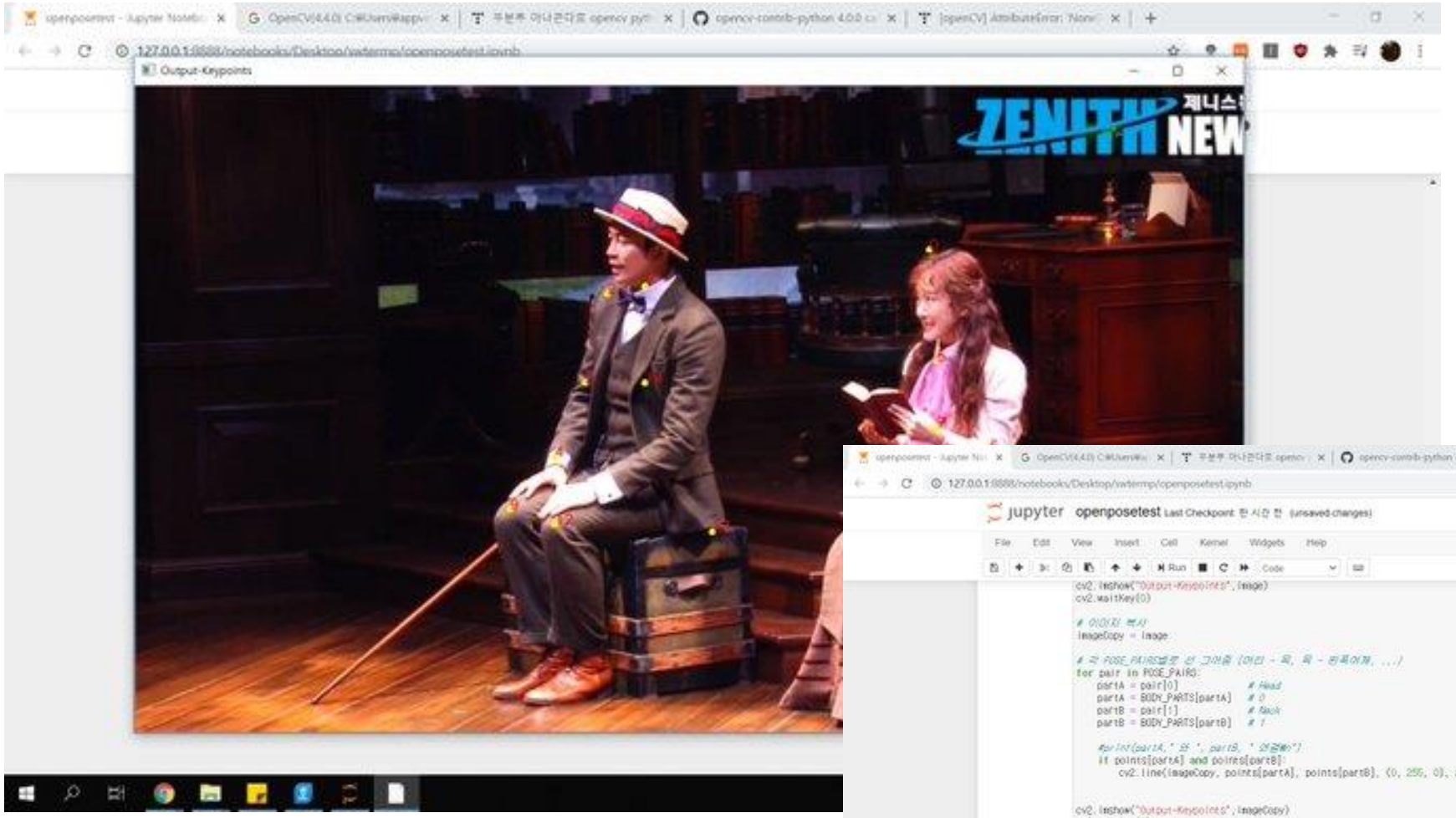
pip install opencv-python

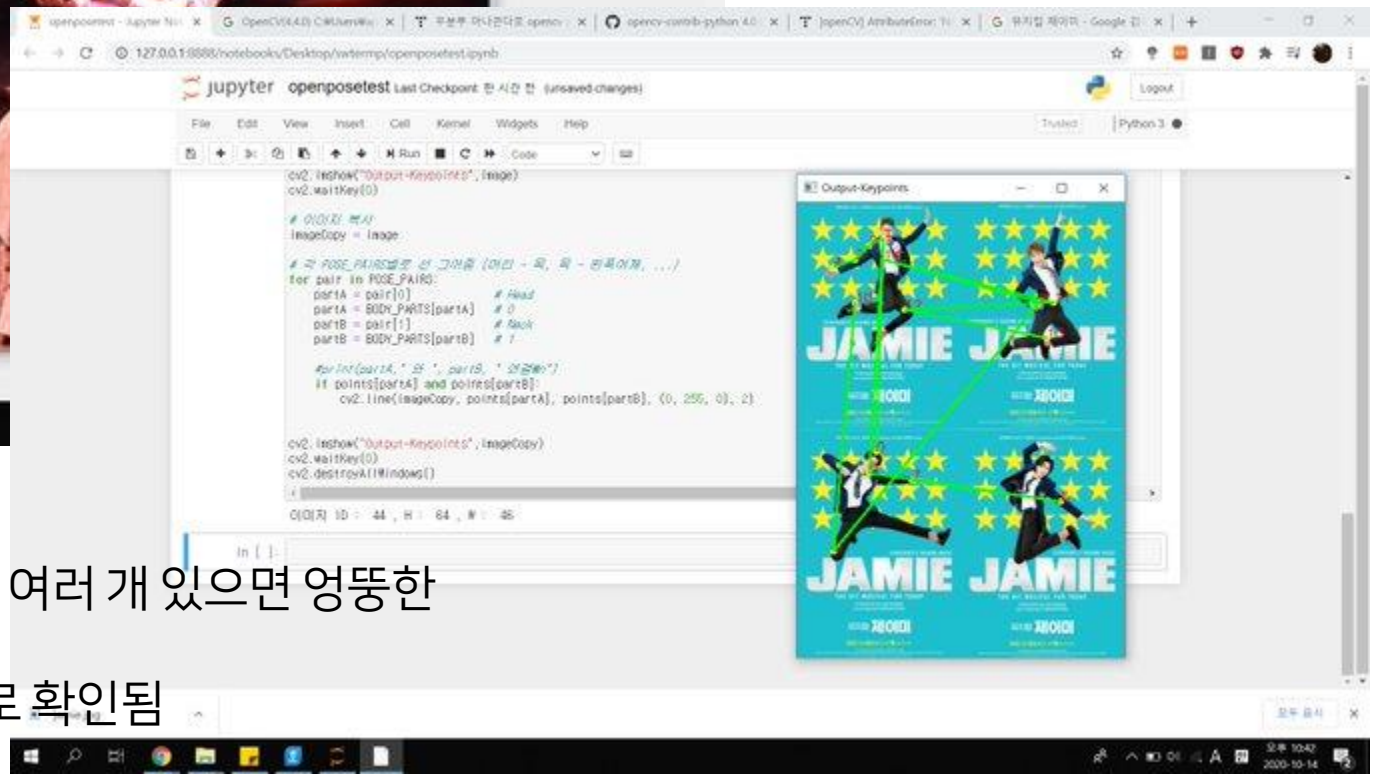
```
C:₩WINDOWS₩system32>pip install opencv-python
Requirement already satisfied: opencv-python in c:₩users₩김서영₩appdata₩local₩programs₩python₩python37₩lib₩site-packages
(4.4.0.44)
Requirement already satisfied: numpy>=1.14.5 in c:₩users₩김서영₩appdata₩local₩programs₩python₩python37₩lib₩site-packages
(from opencv-python) (1.19.2)
```

Openpose 설치

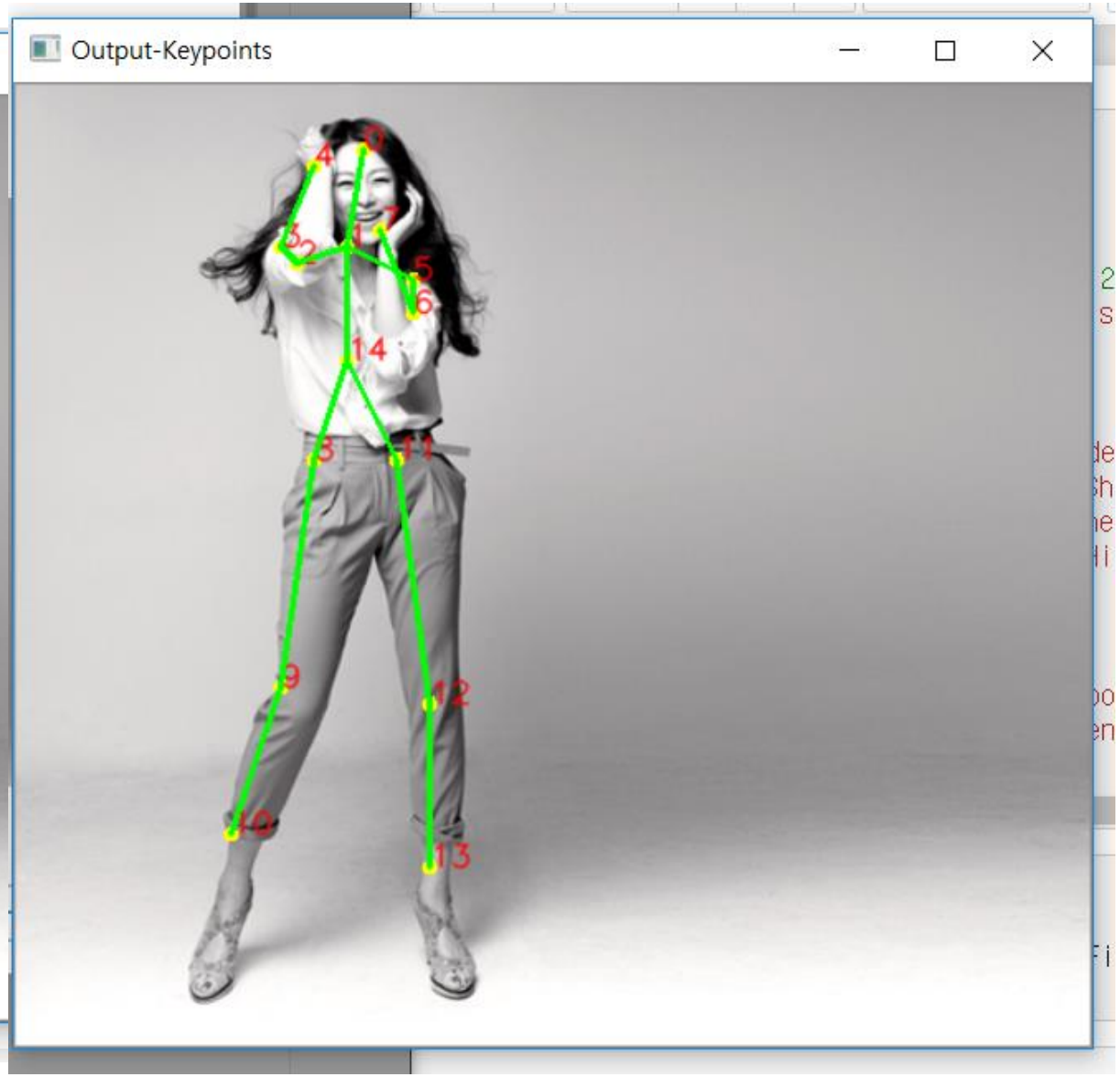
```
---------- BODY, FOOT, FACE, AND HAND MODELS ----------
----- Downloading body pose (COCO and MPI), face and hand models -----

---------- POSE (BODY+FOOT) MODELS ----------
Body (BODY_25)
--2020-10-13 21:49:08--  http://posefs1.perception.cs.cmu.edu/OpenPose/models/pose/body_25/pose_iter_584000.caffemodel
Resolving posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)... 128.2.176.37
Connecting to posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)|128.2.176.37|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 104715850 (100M) [text/plain]
Saving to: 'pose/body_25/pose_iter_584000.caffemodel'

pose_iter_584000.caffemodel                 100%[=============================================================================================================>]  99.86M  1.17MB/s    in 3m 56s

2020-10-13 21:53:06 (433 KB/s) - 'pose/body_25/pose_iter_584000.caffemodel' saved [104715850/104715850]

Body (COCO)
--2020-10-13 21:53:06--  http://posefs1.perception.cs.cmu.edu/OpenPose/models/pose/coco/pose_iter_440000.caffemodel
Resolving posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)... 128.2.176.37
Connecting to posefs1.perception.cs.cmu.edu (posefs1.perception.cs.cmu.edu)|128.2.176.37|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 209274056 (200M) [text/plain]
Saving to: 'pose/coco/pose_iter_440000.caffemodel'

pose_iter_440000.caffemodel                   5%[====>                                                                                                          ]  10.75M   265KB/s    eta 30m 31s
```

사실 오픈포즈라는 기술을 너무 써보고 싶었는데, 설렌다.



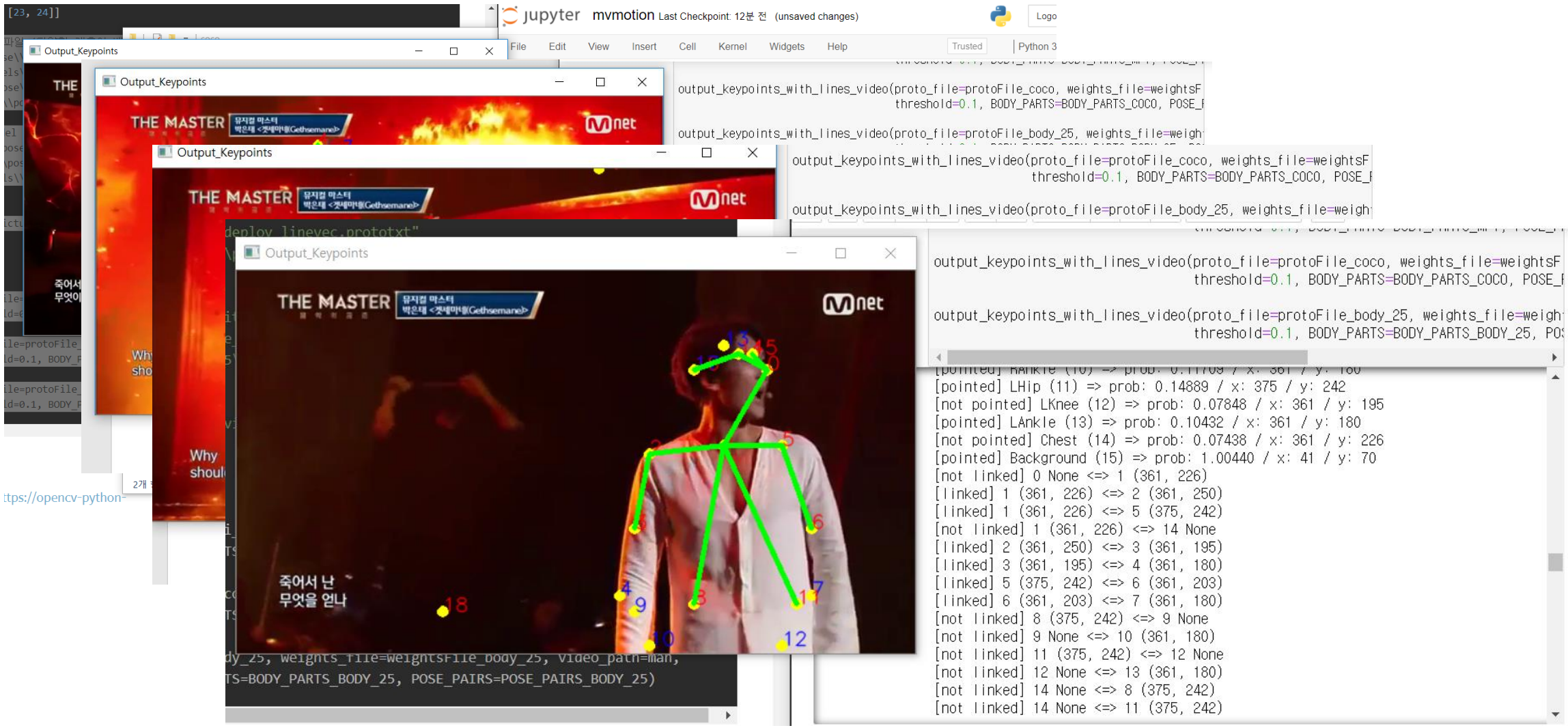


고화질일수록 결과가 좋지 않고 오브젝트가 여러 개 있으면 엉뚱한 점이 이어진다.
=>이는 멀티 인식이 아니어서 그랬던 것으로 확인됨



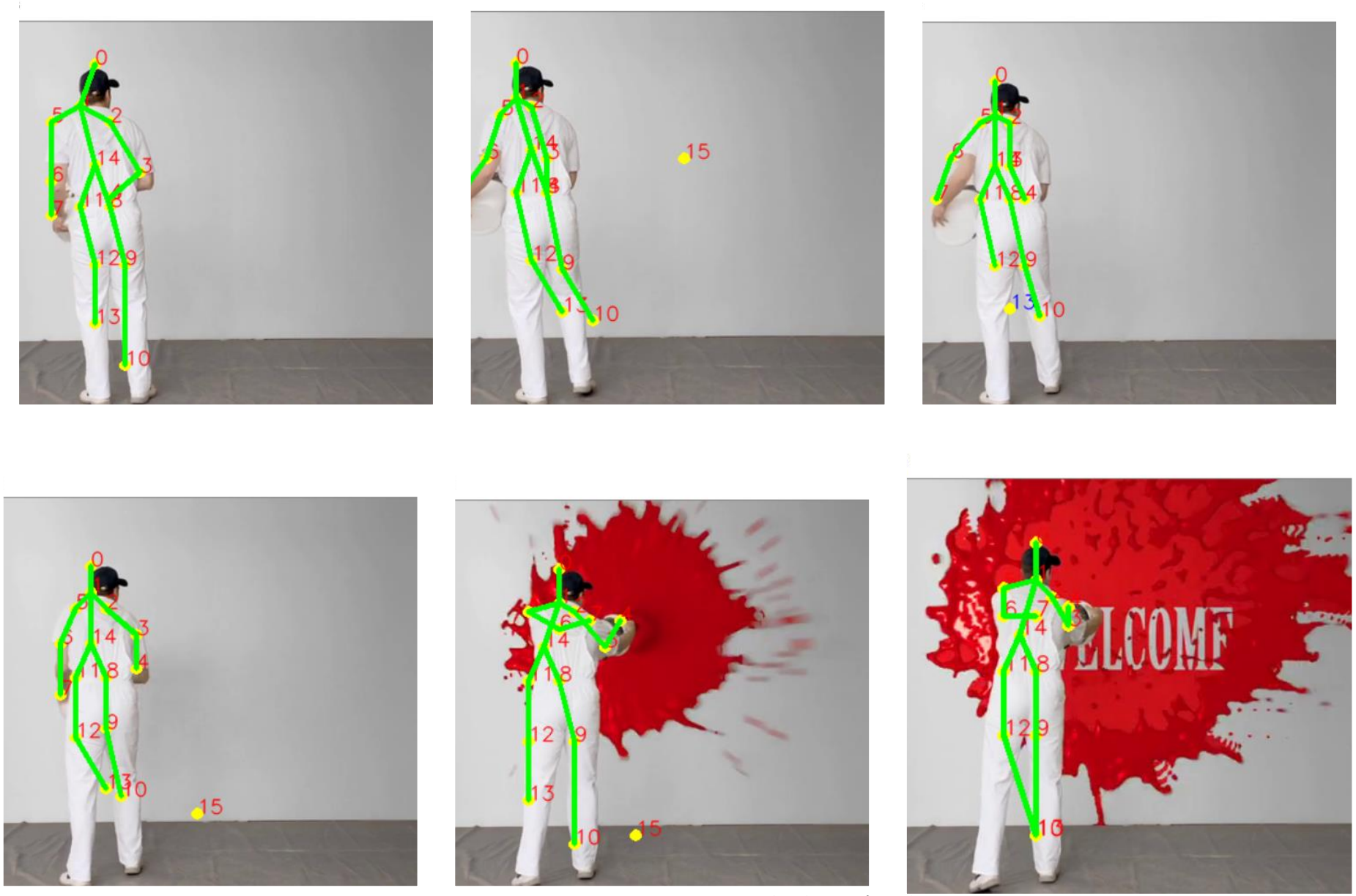
단일 인물의 경우 간단하게 성공!

영상 결과 : 뮤지컬 지저스 크라이스트 수퍼스타 – 박은태, 겟세마네



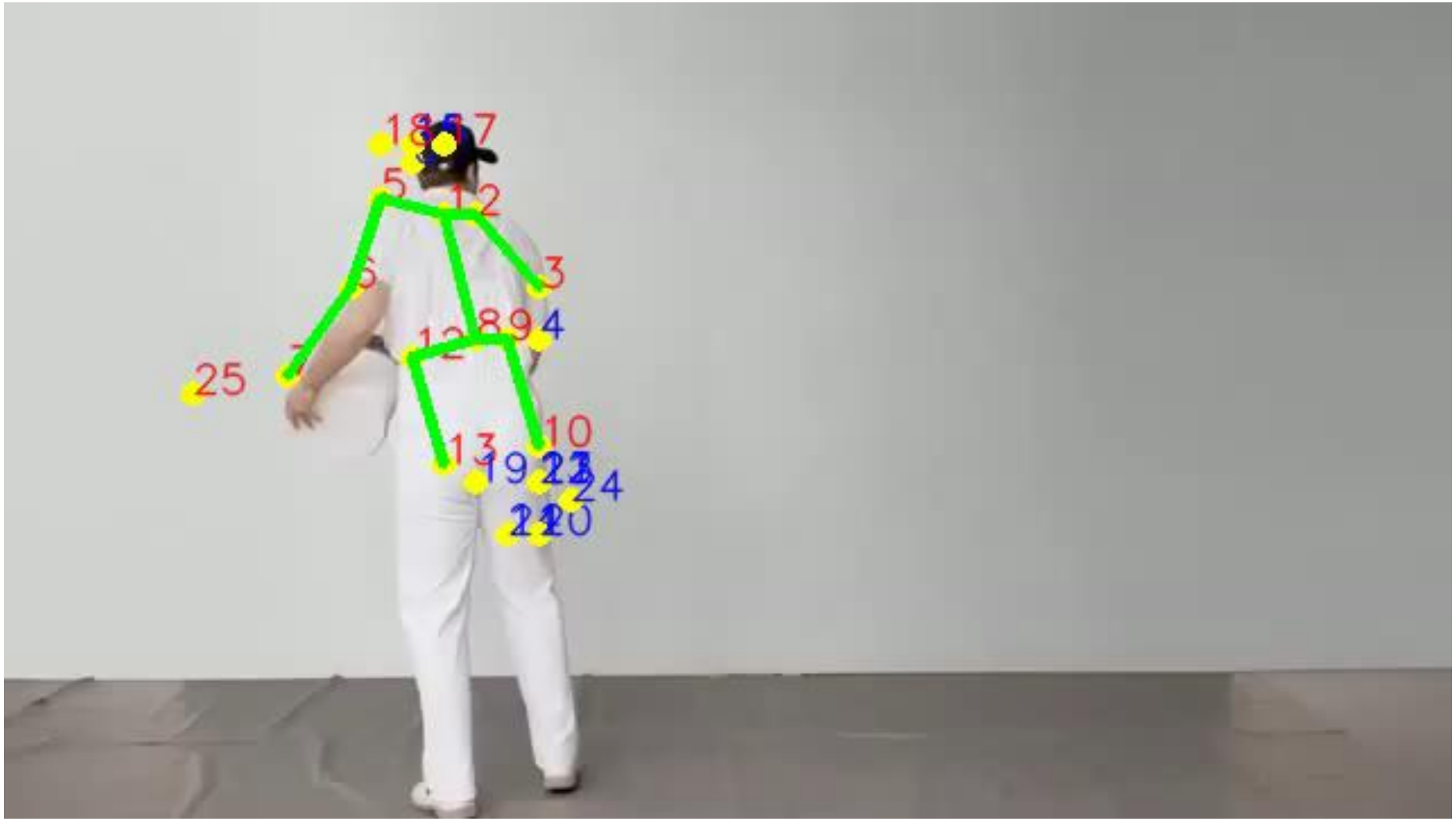
영상 테스트 결과. 왼쪽은 영상 프레임 별 추정결과, 오른쪽은 포인트 위치 출력

영상 결과 : 배경이 깔끔한 경우



비디오 출처 https://pixabay.com/videos/id-25553/

영상 결과 : 배경이 깔끔한 경우(동영상)



비디오 출처 https://pixabay.com/videos/id-25553/

문제점
비슷한 동작의 영상 찾기 어려움(데이터 부족),
영상 속의 모든 모션을 확인할 수 없음, 눈으로 보고 데이터를 만들어야 해서 직접 본 뮤지컬이 아니면 찾기 어려움
공식 영상의 부족 문제, 각도나 방향에 따라 같은 동작이라도 보이는 게 달라짐.
동작이 큰 경우 보통 화 내는 씬에서 많이 발견되는데, 이는 비교하기에 어려움이 있음.

⇒ 춤 영상을 분석하기로 결정

더 라스트 키스 – 왈츠
팬레터 – 거울
줄리 앤 폴 – 사랑한다면 지금처럼

멀티 개체 인식하도록 코드 변경

In [48]:
```
image1 = cv2.imread("C:₩₩softtermp₩₩S1.jpg")
# Fix the input Height and get the width according to the Aspect Ratio
imageHeight, imageWidth, _ = image1.shape
inHeight = 368
inWidth = int((inHeight/imageHeight)*imageWidth)

inpBlob = cv2.dnn.blobFromImage(image1, 1.0 / 255, (inWidth, inHeight),(0, 0, 0), swapRB=False, crop=False)

net.setInput(inpBlob)
output = net.forward()

i = 0
probMap = output[0, i, :, :]
probMap = cv2.resize(probMap, (imageWidth, imageHeight))

plt.imshow(cv2.cvtColor(image1, cv2.COLOR_BGR2RGB))
plt.imshow(probMap, alpha=0.6)
```

Out[48]:
```
<matplotlib.image.AxesImage at 0x22969d8fcc8>
```

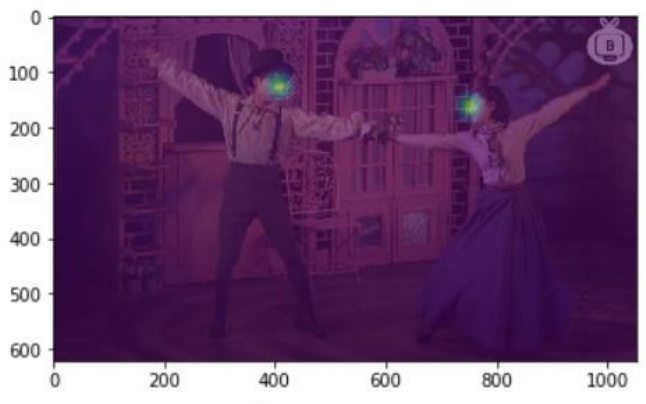
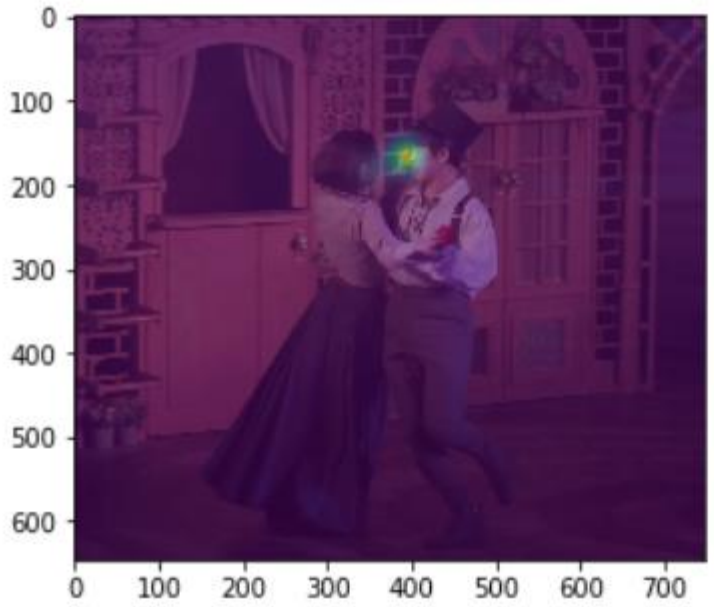
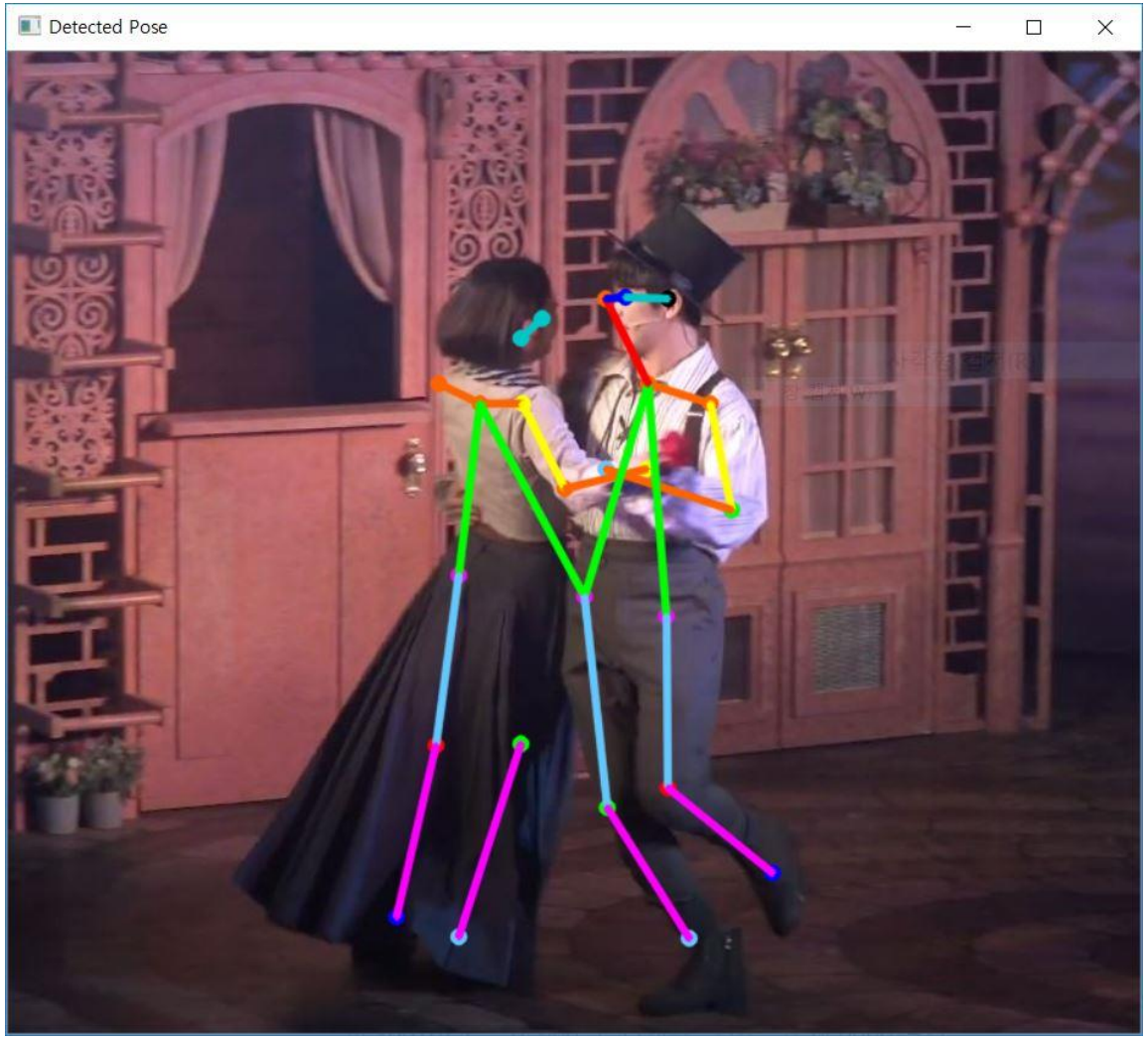


사진 : 뮤지컬 줄리 앤 폴 – 사랑한다면 지금처럼

결과물

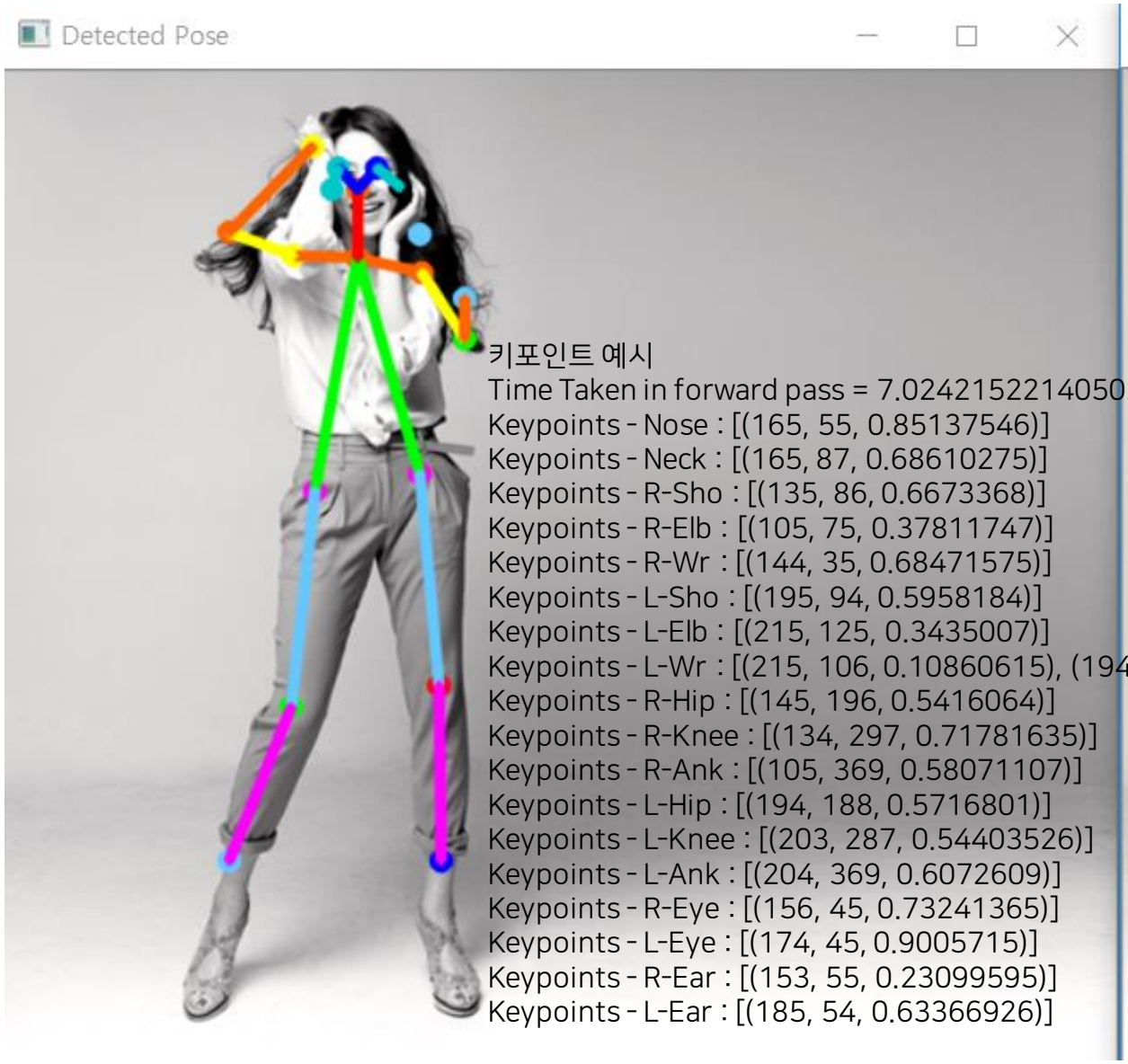
Out[5]: <matplotlib.image.AxesImage at 0x251ac5dbac8>





가까이 붙어있는 사람도 구분이 가능했다.

결과가 잘 나오지 않던 사진도 잘 확인되는것을 볼 수 있다.
사진 : 뮤지컬 키다리아저씨


Detected Pose
키포인트 예시
Time Taken in forward pass = 7.024215221405029
Keypoints - Nose : [(165, 55, 0.85137546)]
Keypoints - Neck : [(165, 87, 0.68610275)]
Keypoints - R-Sho : [(135, 86, 0.6673368)]
Keypoints - R-Elb : [(105, 75, 0.37811747)]
Keypoints - R-Wr : [(144, 35, 0.68471575)]
Keypoints - L-Sho : [(195, 94, 0.5958184)]
Keypoints - L-Elb : [(215, 125, 0.3435007)]
Keypoints - L-Wr : [(215, 106, 0.10860615), (194, 76, 0.16259715)]
Keypoints - R-Hip : [(145, 196, 0.5416064)]
Keypoints - R-Knee : [(134, 297, 0.71781635)]
Keypoints - R-Ank : [(105, 369, 0.58071107)]
Keypoints - L-Hip : [(194, 188, 0.5716801)]
Keypoints - L-Knee : [(203, 287, 0.54403526)]
Keypoints - L-Ank : [(204, 369, 0.6072609)]
Keypoints - R-Eye : [(156, 45, 0.73241365)]
Keypoints - L-Eye : [(174, 45, 0.9005715)]
Keypoints - R-Ear : [(153, 55, 0.23099595)]
Keypoints - L-Ear : [(185, 54, 0.63366926)]

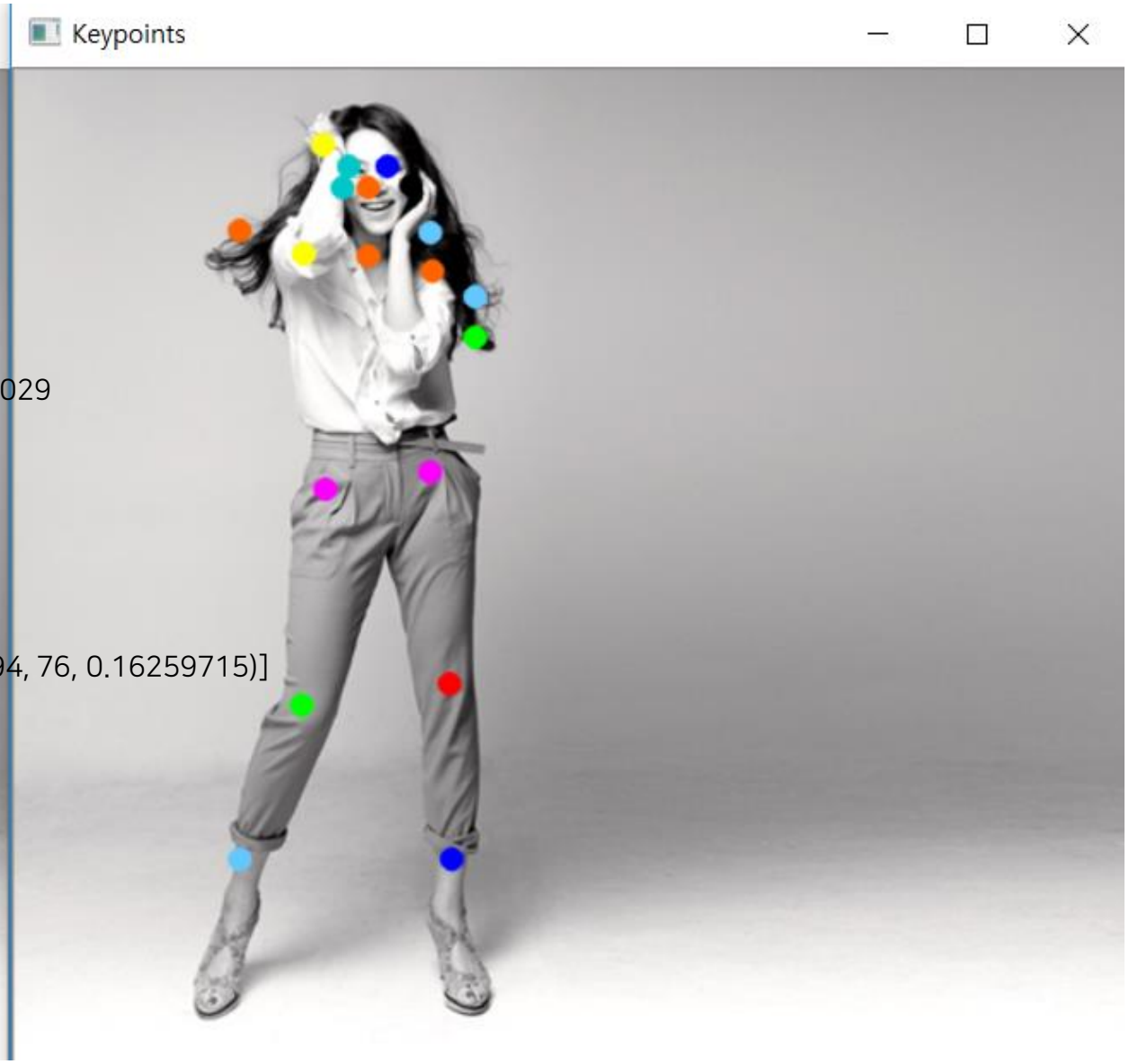

Keypoints

사진 별 동작의 차이를 구하기 위해
세 점 사이의 각도를 구하는 함수 생성
각도에 들어갈 값은 목-어깨-손목 또는 어깨-팔꿈치-손목으로 함.

In [1]:
```python
# for Angle calculation
import math

# 세점 사이 각도 구하기
def getAngle(a, b, c):
    Angle1 = c[0]-b[0]
    Angle2 = a[0]-b[0]

    if Angle1 == 0 :
        Angle1 = 1
    if Angle2 == 0 :
        Angle2 = 1

    ang = math.degrees(math.atan2(c[1]-b[1], Angle1) - math.atan2(a[1]-b[1], Angle2))
    if ang < 0 :
        ang = ang + 360
        if ang > 180 :
            ang = 180 - (ang%180)
            return int(ang)
        else :
            return int(ang)
    else :
        if ang > 180 :
            ang = 180 - (ang%180)
            return int(ang)
        else :
            return int(ang)
```

예시

```python
PartsXY[1][0] = 790
PartsXY[1][1] = 209
PartsXY[2][0] = 790
PartsXY[2][1] = 209

PartsXY[4][0] = 749
PartsXY[4][1] = 222

PartsXY[6][0] = 614
PartsXY[6][1] = 210


PartsXY[3][0] = 830
PartsXY[3][1] = 208

PartsXY[5][0] = 546
PartsXY[5][1] = 196

if (PartsXY[3][0]-PartsXY[5][0]) != 0 and (PartsXY[7][0]-PartsXY[5][0]) :
    angleL = getAngle(PartsXY[3], PartsXY[5], PartsXY[7])
    print("AngleL : " + str(angleL))

if (PartsXY[2][0]-PartsXY[4][0]) != 0 and (PartsXY[6][0]-PartsXY[4][0]) :
    angleR = getAngle(PartsXY[2], PartsXY[4], PartsXY[6])
    print("AngleR : " + str(angleR))
```

```
AngleL : 162
AngleR : 157
```

multi person dt - Jupyter Noteb

127.0.0.1:8888/notebooks/Desktop/swtermp/multi%20person%20dt.ipynb

jupyter multi person dt Last Checkpoint: 2시간 전 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [ ]:

```
import cv2
import time
import numpy as np
from random import randint

# 이미지 path
image1 = cv2.imread("C:₩₩softtermp₩₩s1.jpg")

# 각 파일 path
protoFile = "C:₩₩softtermp₩₩openpose-master₩₩openpose-master₩₩models₩₩pose₩₩coco₩₩pose_deploy_linevec.prototxt"
weightsFile = "C:₩₩softtermp₩₩openpose-master₩₩openpose-master₩₩models₩₩pose₩₩coco₩₩pose_iter_440000.caffemodel"

nPoints = 18
# COCO Output Format
keypointsMapping = ['Nose', 'Neck', 'R-Sho', 'R-Elb', 'R-Wr', 'L-Sho', 'L-Elb', 'L-Wr', 'R-Hip', 'R-Knee', 'R-Ank', 'L-Hip', 'L-Knee', 'L-A

POSE_PAIRS = [[1,2], [1,5], [2,3], [3,4], [5,6], [6,7],
              [1,8], [8,9], [9,10], [1,11], [11,12], [12,13],
              [1,0], [0,14], [14,16], [0,15], [15,17],
              [2,17], [5,16] ]

BODY_PARTS = { "Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
               "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
               "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13, "Chest": 14,
               "Background": 15 }


# index of pafs corresponding to the POSE_PAIRS
# e.g for POSE_PAIR(1,2), the PAFs are located at indices (31,32) of output, Similarly, (1,5)
mapIdx = [[31,32], [39,40], [33,34], [35,36], [41,42], [43,44],
          [19,20], [21,22], [23,24], [25,26], [27,28], [29,30],
          [47,48], [49,50], [53,54], [51,52], [55,56],
          [37,38], [45,46]]

colors = [ [0,100,255], [0,100,255], [0,255,255], [0,100,255], [0,255,255], [0,100,255],
           [0,255,0], [255,200,100], [255,0,255], [0,255,0], [255,200,100], [255,0,255],
           [0,0,255], [255,0,0], [200,200,0], [255,0,0], [200,200,0], [0,0,0]]


def getKeypoints(probMap, threshold=0.1):

    mapSmooth = cv2.GaussianBlur(probMap,(3,3),0,0)

    mapMask = np.uint8(mapSmooth>threshold)
    keypoints = []
```

사용한 영상들


[1열중앙석] 뮤지컬 '줄리 앤 폴' 사랑한다면 그들처럼 - 이지수, 송유택
소극장 예
1:24 / 3:33


[뮤지컬 팬레터-글자 그대로+별이 반짝이는 시간] 완벽한 호흡과 가창력 !
중소극장 예
NCTv
뉴스컬처
5:58 / 8:29


더 라스트 키스 프레스콜_왈츠~알 수 없는 그 곳으로_전동석×김소향/전동석×민경아
DAS MUSICAL
더 라스트키스
대극장 예
NAVER 공연 live
CyberLink
by PowerDirector
2:50 / 13:47

각각의 각도
AngleL : 162
AngleR : 157

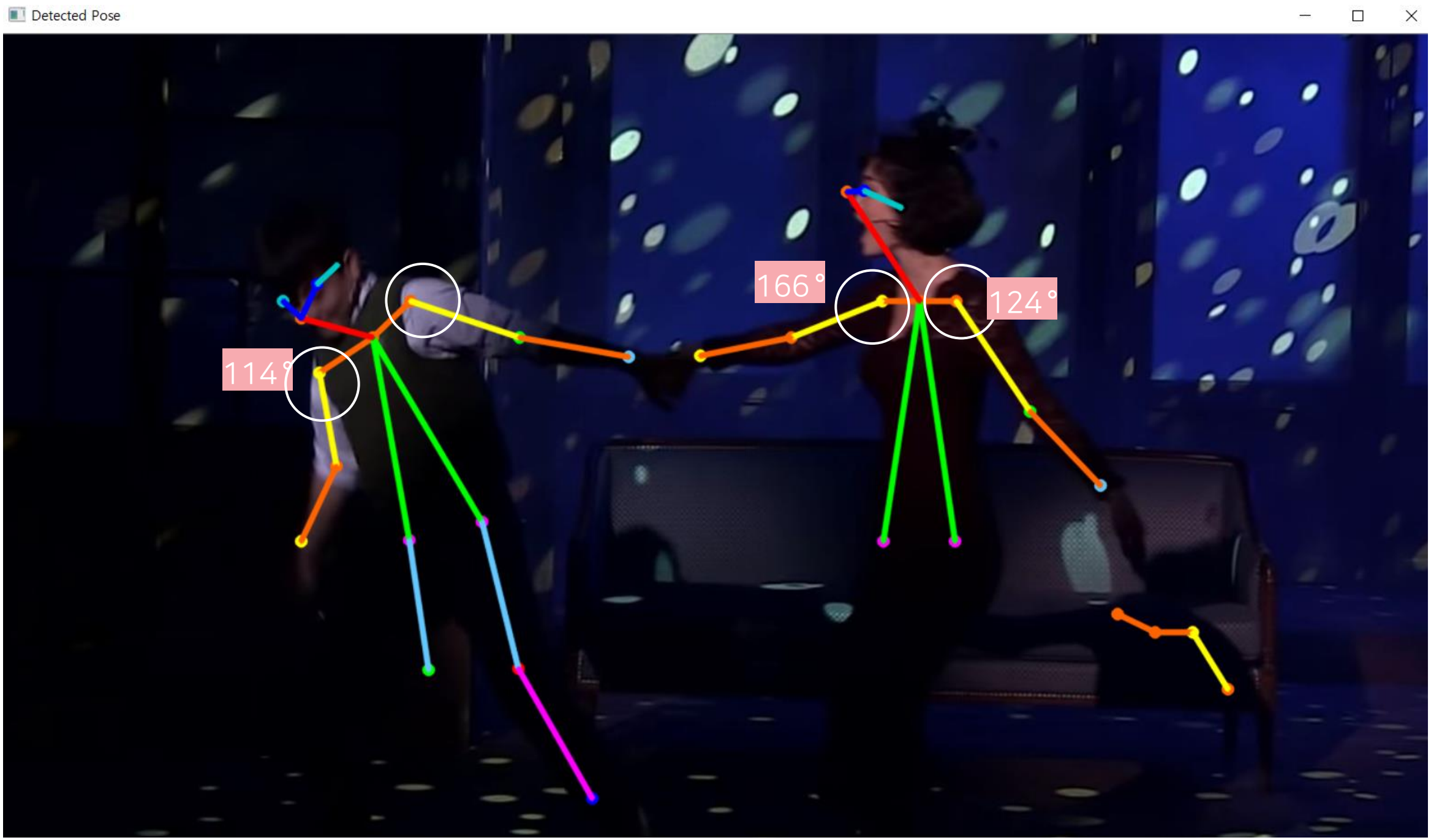


사진 : 뮤지컬 팬레터 – 별이 반짝이는 시간


Detected Pose

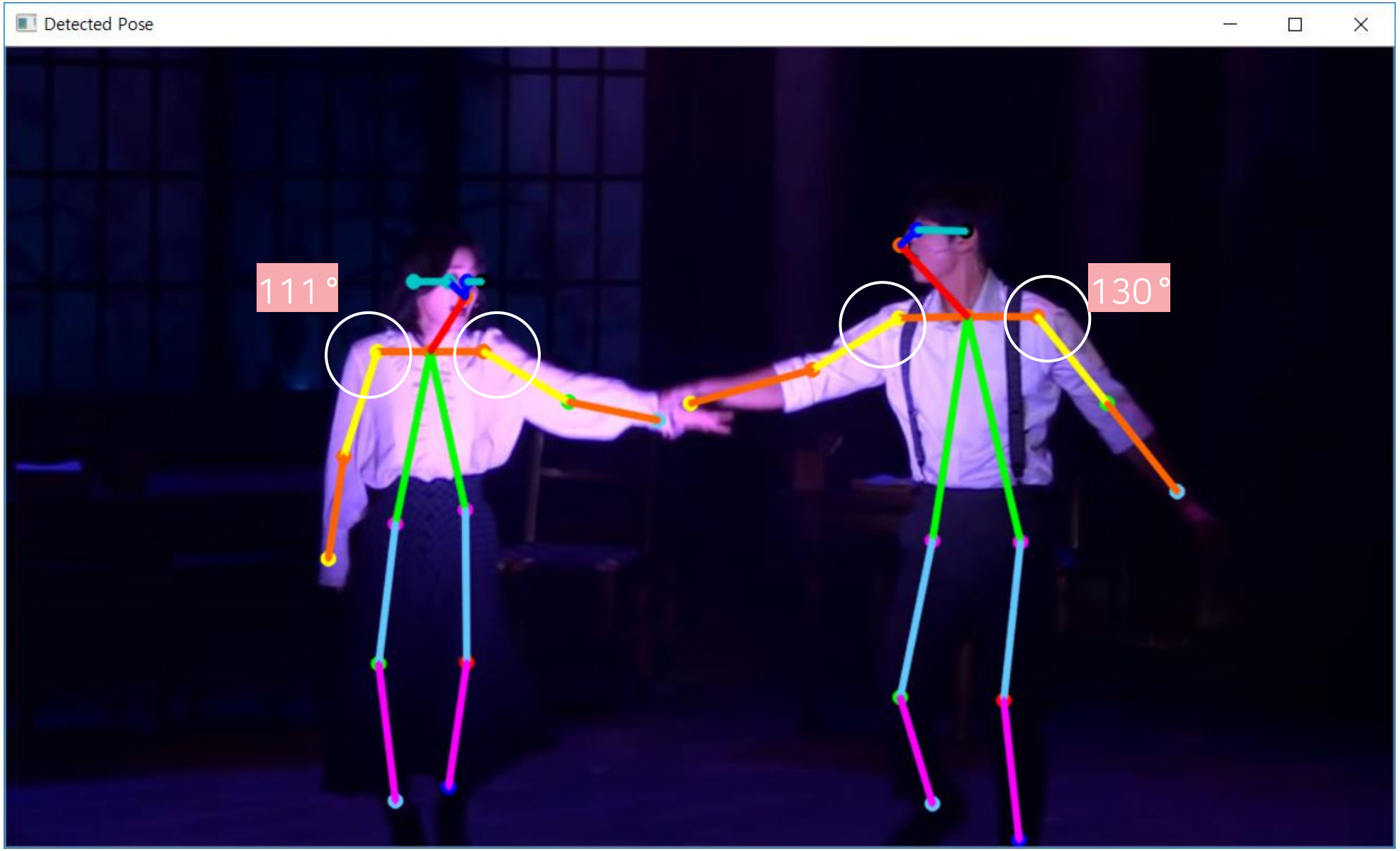
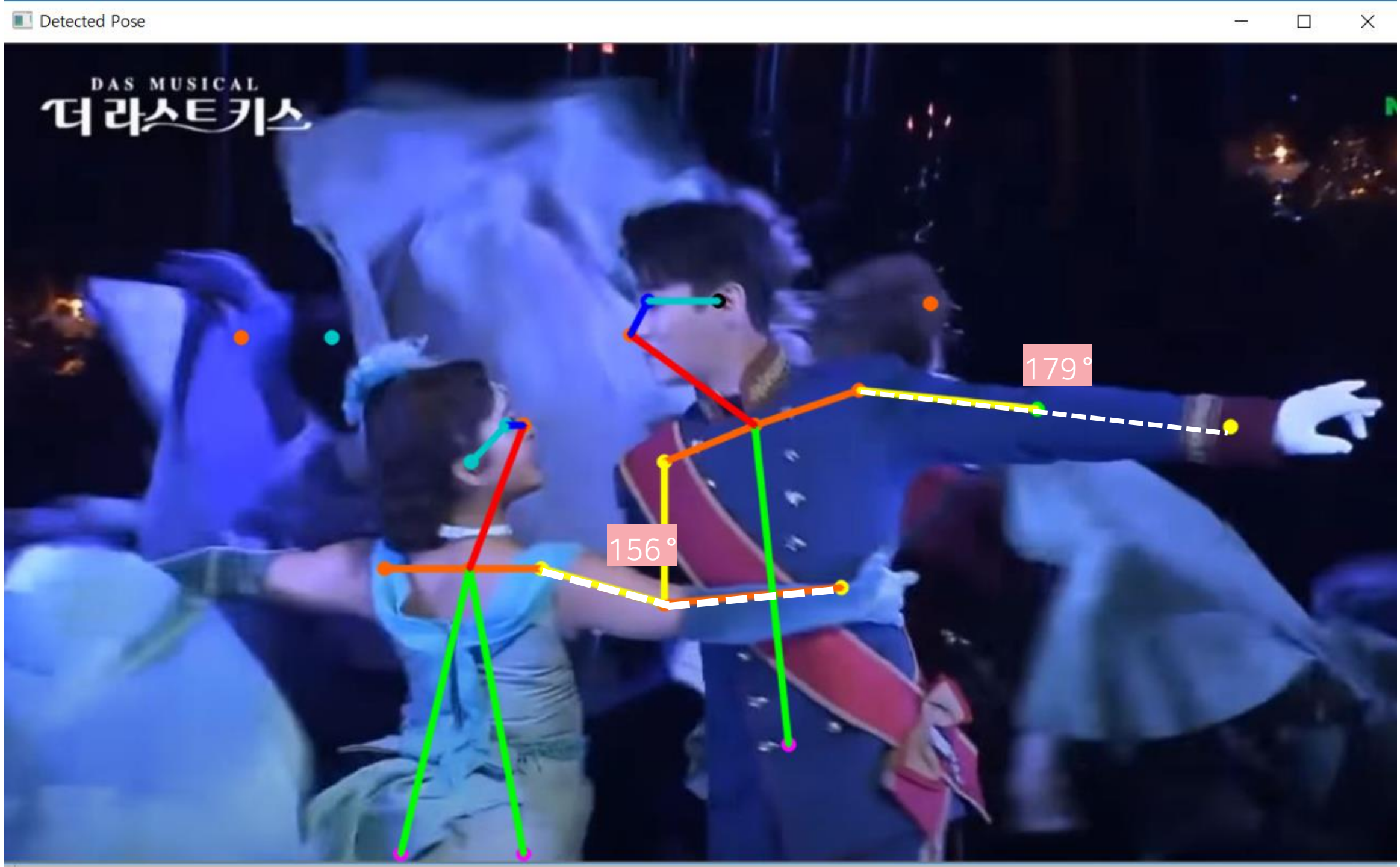


사진 : 뮤지컬 팬레터 – 거울

사진 : 뮤지컬 더 라스트 키스 – 왈츠


Detected Pose
DAS MUSICAL
더 라스트 키스
179°
156°


Detected Pose
DAS MUSICAL
더 라스트 키스
123°
158°

최종 코드 일부(파일 및 링크 첨부)

```
import cv2
import time
import numpy as np
from random import randint

# 이미지 path
image1 = cv2.imread("C:₩₩softtermp₩₩b4.jpg")

# 각 파일 path
protoFile = "C:₩₩softtermp₩₩openpose-master₩₩openpose-master₩₩models₩₩pose₩₩coco₩₩pose_deploy_linevec.prototxt"
weightsFile = "C:₩₩softtermp₩₩openpose-master₩₩openpose-master₩₩models₩₩pose₩₩coco₩₩pose_iter_440000.caffemodel
nPoints = 18
# COCO Output Format
keypointsMapping = ['Nose', 'Neck', 'R-Sho', 'R-Elb', 'R-Wr', 'L-Sho', 'L-Elb', 'L-Wr', 'R-Hip', 'R-Knee', 'R-A

POSE_PAIRS = [[1,2], [1,5], [2,3], [3,4], [5,6], [6,7],
              [1,8], [8,9], [9,10], [1,11], [11,12], [12,13],
              [1,0], [0,14], [14,16], [0,15], [15,17],
              [2,17], [5,16] ]

BODY_PARTS = { "Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
               "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
               "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13, "Chest": 14,
               "Background": 15 }


# index of pafs correspoding to the POSE_PAIRS
# e.g for POSE_PAIR(1,2), the PAFs are located at indices (31,32) of output, Similarly, (1,5) -> (39,40) and so
mapIdx = [[31,32], [39,40], [33,34], [35,36], [41,42], [43,44],
          [19,20], [21,22], [23,24], [25,26], [27,28], [29,30],
          [47,48], [49,50], [53,54], [51,52], [55,56],
          [37,38], [45,46]]
```

```
            partBs = valid_pairs[k][:,1]
            indexA, indexB = np.array(POSE_PAIRS[k])

            for i in range(len(valid_pairs[k])):
                found = 0
                person_idx = -1
                for j in range(len(personwiseKeypoints)):
                    if personwiseKeypoints[j][indexA] == partAs[i]:
                        person_idx = j
                        found = 1
                        break

                if found:
                    personwiseKeypoints[person_idx][indexB] = partBs[i]
                    personwiseKeypoints[person_idx][-1] += keypoints_list[partBs[i].astype(int), 2] + valid_pairs[k][i][2]

                # if find no partA in the subset, create a new subset
                elif not found and k < 17:
                    row = -1 * np.ones(19)
                    row[indexA] = partAs[i]
                    row[indexB] = partBs[i]
                    # add the keypoint_scores for the two keypoints and the paf_score
                    row[-1] = sum(keypoints_list[valid_pairs[k][i,:2].astype(int), 2]) + valid_pairs[k][i][2]
                    personwiseKeypoints = np.vstack([personwiseKeypoints, row])
    return personwiseKeypoints


frameWidth = image1.shape[1]
frameHeight = image1.shape[0]

t = time.time()
net = cv2.dnn.readNetFromCaffe(protoFile, weightsFile)

# Fix the input Height and get the width according to the Aspect Ratio
inHeight = 368
inWidth = int((inHeight/frameHeight)*frameWidth)

inpBlob = cv2.dnn.blobFromImage(image1, 1.0 / 255, (inWidth, inHeight),
                          (0, 0, 0), swapRB=False, crop=False)

net.setInput(inpBlob)
output = net.forward()
print("Time Taken in forward pass = {}".format(time.time() - t))

detected_keypoints = []
```

https://www.notion.so/Openpose-7371ec2791084cd2bd7ecb06e98488d6

최종 결론

1. 춤 영상에서의 동작 크기(어깨-팔꿈치-손목의 각도) 차이는 미비했다.
2. 팔을 들어올린 정도(목-어깨-팔)의 차이는 있었는데, 이는 대극장에서는 가로 공간을 더 사용하는 것으로 볼 수 있었다.
3. 대극장의 경우 팔을 거의 굽히지 않고(춤을 추는 내내 150도 이상으로 벌리고 있다.), 항상 신체와 90도 정도의 각도로 들고 있었다.
4. 중소극장의 경우 대극장에 비해 조금더 동작이 자연스럽고 자유로웠다.

(이는 대극장 영상에 나온 전동석 배우의 뻣뻣하기로 유명한 실력 탓도 있는 것 같다.)

추후 계획 / 아쉬운 점

1. 영상에서의 멀티 개체 인식 및 각도 계산을 진행하려 했으나 디텍팅 방법에 따라 구현할 수 있는 함수가 달라 어려웠다.
2. 카메라의 시점이 고정된 영상이 많지 않아 정확한 사람의 이동량을 구할 수 없어 아쉬웠다.
3. Openpose 관련 내용이 적어 어려웠던 부분들(설치부터 예제까지)을 정리해서 포스팅해 볼 예정이다.
4. 리눅스와 Google colab 역시 제대로 활용을 못해 아쉽다.
5. 기회가 된다면 영상을 통해 openpose로 뼈대를 찾아 학습한 후, 각각의 영상이 어떤 춤을 추는지 분류할 수 있는 기술을 연구해보고 싶다. (현재는 함수를 통해 인체의 각도나 척추 비례 좌표값 등을 통해 각각의 동작들을 분류하고 있다.)

openpose를 사용하기 위해 힘들었던 만큼 많은 공부가 되었다. 아쉬운 점도 많지만 새로운 기술을 사용해볼 수 있는 좋은 기회였고 본 프로젝트를 통해서 뮤지컬 속의 춤과 연출을 여러 각도에서 분석해 볼 수 있어서 즐거웠다. 앞으로도 새로운 기능을 사용해볼 때 겁 없이 시작할 수 있을 듯 하다.

참고문헌

장재호, 지준환, 김두환, 최민기, 윤태진. (2020). 인체 자세 인식 딥러닝을 이용한 운동 자세 훈련 시스템 개발. 한국컴퓨터정보학회 학술발표논문집 , 28(2), 289-290.

Yoo, H.-R., & Lee, B.-H. (2019). 감시 영상을 활용한 OpenPose 기반 아동 학대 판단시스템. 한국정보통신학회논문지, *23*(3), 282-290.

Kim, Minjoon, Lee, Zucheul, & Kim, Wonha. (2016). 영상 특성과 스켈레톤 분석을 이용한 실시간 인간 객체 추출. 방송공학회논문지, *21*(5), 782-791.

황일산, 서형규, 윤명훈, 서강현, 이동엽,김진일.(2017).OpenCV의 Tracking 기능을 이용한 외부인 출입감지 시스템의 설계 및 구현. 한국정보통신학회 2017년도 춘계학술대회, pp.797 - 799

참고링크

오픈포즈 관련

https://hiseon.me/data-analytics/introduction-openpose/

http://daddynkidsmakers.blogspot.com/2020/07/openpose.html

https://blog.naver.com/PostView.nhn?blogId=rhrkdfus&logNo=221531159811&categoryNo=0&parentCategoryNo=0

https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/

https://www.learnopencv.com/multi-person-pose-estimation-in-opencv-using-openpose/

카페투-디텍트론 기사
https://zdnet.co.kr/view/?no=20180125091145

Thank You

감사합니다