Maker’s Day

척추수술 2300만원

CONTENT

Arduino

AI

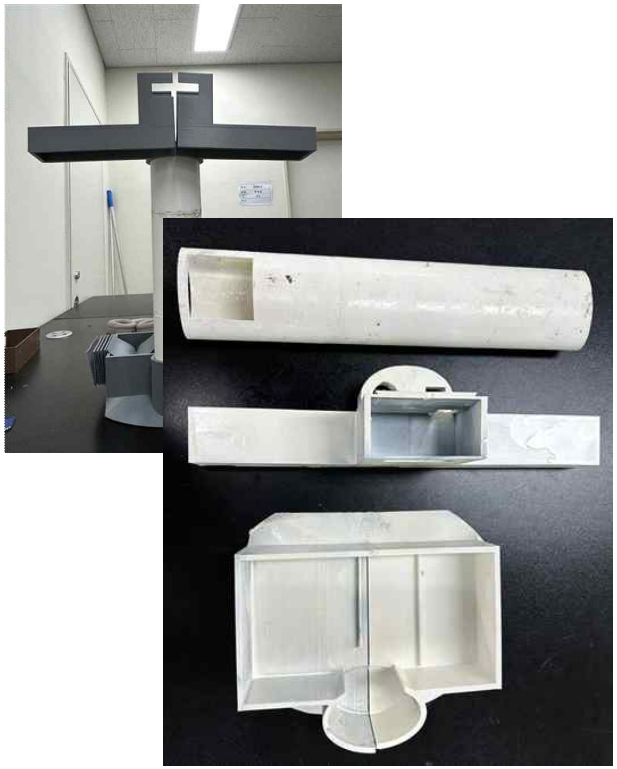
Frontend

Arduino

감압 방석

- 방석 구현에 필요한 선정리 내설설, 바느질 완료
- 사람 마다 다른 무게 영점 조절을 위한 스위치, on, off를 위한 슬라이드 스위치 제작 및 코드 작성 완료
- 방석에 있는 센서들의 빵판과 보드를 보관하기 위한 (case) 제작 완료
- 출전 모듈과 배터리 장착 완료(case에 보관)

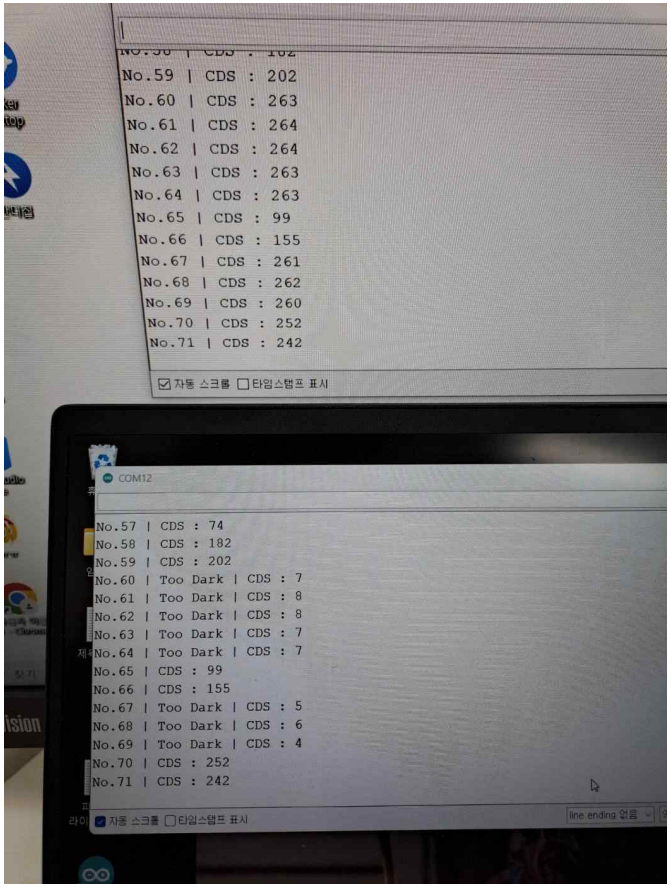
Arduino



스탠드

- 조도센서 구멍, 네오픽셀 전선 구멍 구현
- 스탠드 모델링, 프린팅 완료
- 각 부품 접착, 사포질 완료
- 도색 작업 진행 중

Arduino



통신

- 블루투스 ➡□ 와이파이 ➡□ 블루투스
- 블루투스 모듈 오류 원인 파악 및 부품교체
- 아두이노 - PC 간 블루투스 통신
- 아두이노 - 아두이노 간 블루투스 통신
- 데이터 송/수신을 위해 코드 업데이트 예정

AI

```
#VISUALIZE DEGREE
cap =cv2.VideoCapture(0) #setting video capture device(number은 웹캠을 대표하는 숫자)

# Curl counter variables
counter = 0

# Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence=0.5,min_tracking_confidence=0.5) as pose:

    while cap.isOpened(): #실시간 영상을 가져올 수 있도록 함.
        ret, frame = cap.read() #frame은 현재 프레임 이미지가 담긴 것.

        #Detect stuff and render
        #Recolor image to RGB
        image =cv2.cvtColor(frame,cv2.COLOR_BGR2RGB) #웹캠으로 읽어온 frame을 BGR에서 RGB로 변환(Mediapipe는 RGB 형식임.)
        image.flags.writeable =False #이미지를 불변으로 설정하여 처리 속도를 향상 시킴.

        #Make detection -> 자세 detection을 results라는 변수에
        results= pose.process(image)

        # Recolor back to BGR
        image.flags.writeable=True #image 위에 그릴 수 있도록.
        image =cv2.cvtColor(image,cv2.COLOR_RGB2BGR) #Mediapipe 처리 결과를 BGR로 변환

         #Extract landmarks
        try:
            row = np.array([[res.x,res.y,res.z,res.visibility] for res in results.pose_landmarks.landmark]).flatten()
            X = pd.DataFrame([row],columns=landmarks[1:])
            body_language_class = model.predict(X)[0]
            body_language_prob = model.predict_proba(X)[0]

            if body_language_class == 0 and body_language_prob[body_language_prob.argmax()] >= .7:
                current_stage = 'Good'
            elif current_stage == 'Good' and body_language_class ==1 and body_language_prob[body_language_prob.argmax()] >= .7:
                current_stage = 'Shoulder'
                counter +=1
```

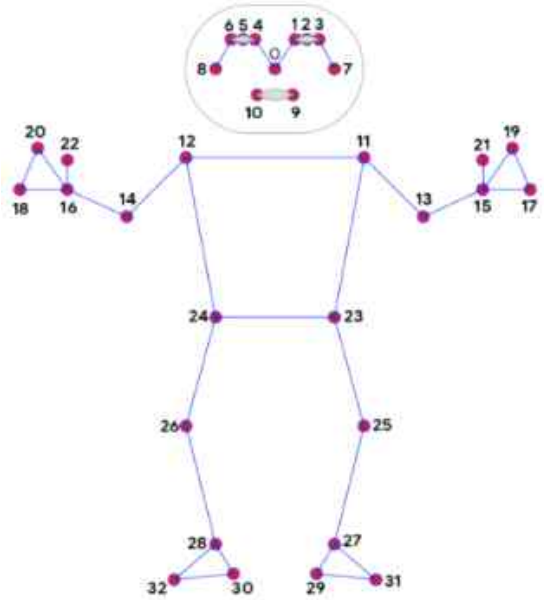
1330	1	#######	#######	-1.667116	#######	0.512183	#######	-1.586819	#######	#######	#######
1331	0	#######	#######	-1.431041	#######	#######	#######	-1.330829	#######	#######	#######
1332	1	#######	#######	-1.433286	#######	#######	#######	-1.350468	#######	#######	#######
1333	0	#######	#######	-1.254452	#######	#######	#######	-1.161227	#######	#######	#######
1334	0	#######	#######	-1.356707	#######	#######	#######	-1.258789	0.9992	#######	#######
1335	0	#######	#######	-1.251369	#######	#######	#######	-1.163955	#######	0.684631	#######
1336	1	#######	#######	-1.02261	0.99991	#######	#######	-0.929411	#######	#######	#######
1337	0	#######	#######	-1.228032	#######	#######	#######	-1.151494	#######	#######	#######
1338	1	#######	#######	-1.483268	#######	#######	#######	-1.413634	#######	#######	#######
1339	0	#######	#######	-1.561058	#######	#######	0.444309	-1.46307	#######	#######	#######
1340	0	#######	#######	-1.137405	0.999954	#######	0.455125	-1.055987	#######	#######	#######
1341	0	#######	0.549234	-0.935775	#######	#######	#######	-0.870641	#######	#######	#######
1342	0	#######	#######	-0.621805	#######	#######	#######	-0.549567	#######	#######	#######
1343	1	#######	#######	-0.785391	#######	#######	#######	-0.715706	#######	#######	0.534487
1344	0	#######	#######	-0.855964	#######	0.655196	0.464889	-0.792099	#######	#######	#######
1345	0	#######	#######	-1.094838	#######	#######	#######	-0.996565	#######	#######	#######

- 자세교정을 위한 거북목, 어깨 비대칭 모델 제작 완료
- 자세가 바르면 ‘good’을 표시하고,
그렇지 않으면 각각 ‘Textneck’, ‘Shoulder’ 표시
- 자세가 바르지 못할 경우에는, 그 횟수를 세서
글자 옆에 count 횟수를 나타냄
- AI를 학습시킬수록 정확도가 올라감

AI

```
#열 중에서 v가 들어 있는 것들을 찾고, 평균 찾아내기
df.filter(regex='v').mean()
```

```
v1     0.998986
v2     0.998505
v3     0.998386
v4     0.998170
v5     0.998642
v6     0.998681
v7     0.998741
v8     0.997861
v9     0.998656
v10    0.996005
v11    0.995936
v12    0.929396
v14    0.152548
v15    0.259727
v16    0.035997
v17    0.064872
```



EDA 및 전처리

● Visibility 분포 확인
-> Visibility는 해당 관절을 얼마나 잘 추적하고 있는지에 대한 지표로, 이 **수치가 낮**으면 학습을 진행하는데에 **방해**가 되는 요소로 판단.
->관절 13까지 0.9 이상으로 잘 관측된다.
(상체에서 어깨 라인까지 데이터만 남기고 제거.)

AI

```
[81] #상관계수가 낮은 것들을 확인해본다.
     for item in X.columns:
       if stats.pearsonr(X_train[item],y_train)[0]<=0.3 and stats.pearsonr(X_train[item],y_train)[0]>=-0.3:
           print(item)
           print('Correlation:{:.2f}'.format(stats.pearsonr(X_train[item],y_train)[0]))
```

```
x5
Correlation:0.17
x6
Correlation:-0.11
x7
Correlation:-0.26
x11
Correlation:0.09
y12
Correlation:0.15
```

```
7] #상관계수 확인 후, visibility가 소용없는 정보라는 것을 알게 되었음. 제거 ㄱㄱ
   df = df.filter(regex='^x[0-9]$|^y[0-9]$|^z[0-9]$|^x1[0-2]|^y1[0-2]|^z1[0-2]|^c')
   df
```

class	x1	y1	z1	x2	y2	z2	x3	y3	z3	...
1.0	0.574625	0.645155	-2.716616	0.613432	0.541128	-2.666490	0.639249	0.540640	-2.666276	...
0.0	0.553470	0.509528	-0.966766	0.578971	0.449093	-0.893713	0.593508	0.450831	-0.893681	...
0.0	0.558947	0.515051	-0.976771	0.586876	0.458080	-0.891556	0.602352	0.459791	-0.891732	...
0.0	0.514702	0.311331	-1.469513	0.542640	0.250292	-1.378310	0.562116	0.251408	-1.378602	...
0.0	0.549252	0.420117	-1.490885	0.575709	0.350068	-1.412757	0.590431	0.351519	-1.413012	...
...	...	...	...	...	...	...	...	...	...	...
1.0	0.581935	0.793034	-3.754220	0.656170	0.674340	-3.753478	0.694430	0.672747	-3.752829	...
0.0	0.567253	0.800274	-4.041223	0.650178	0.681071	-4.038610	0.687658	0.678491	-4.038176	...
1.0	0.610389	0.679030	-2.702558	0.665414	0.567028	-2.717772	0.703793	0.568688	-2.716740	...

EDA 및 전처리

● 상관계수 확인
-> 13 * 4개의 상관계수를 모두 그래프로 표현하기 어려울 것으로 판단.
-> 각각의 관절 정보에 대한 상관계수를 뽑아내는 stats 모듈의 pearsonr 함수를 만들어 낮은 **상관계수들을 판별**함.
=> 1-12까지의 관절 정보 중에서 **visibility와 거북목 class 사이의 연관성이 없는** 것으로 나와 전처리 진행

.

AI

```python
def angle3(a,b,c): #세 점 사이의 각도를 구한다.
    a_x= np.array(df['x{}'.format(a)]) #First
    a_y = np.array(df['y{}'.format(a)])

    b_x = np.array(df['x{}'.format(b)]) #Mid
    b_y = np.array(df['y{}'.format(b)])

    c_x = np.array(df['x{}'.format(c)])
    c_y = np.array(df['y{}'.format(c)]) #End

    #y from endpoint - y form midpoint, x form end - x from mind
    radians=np.arctan2(c_y-b_y,c_x-b_x) - np.arctan2(a_y-b_y,a_x-b_x)
    angle=np.abs(radians*180.0/np.pi)

    for i in range(0,angle.shape[0]):
      if angle[i]>180.0:
            angle[i] = 360-angle[i]

    return pd.Series(angle)
```

```python
# left right angle 상관계수 추가 확인 -> 충분히 강력한 상관계수를 가진다.
for item in ['left_angle','right_angle']:
  print(item)
  print('Correlation:{:.2f}'.format(stats.pearsonr(X_train[item],y_train)[0]))
```

```
left_angle
Correlation:0.73
right_angle
Correlation:0.84
```

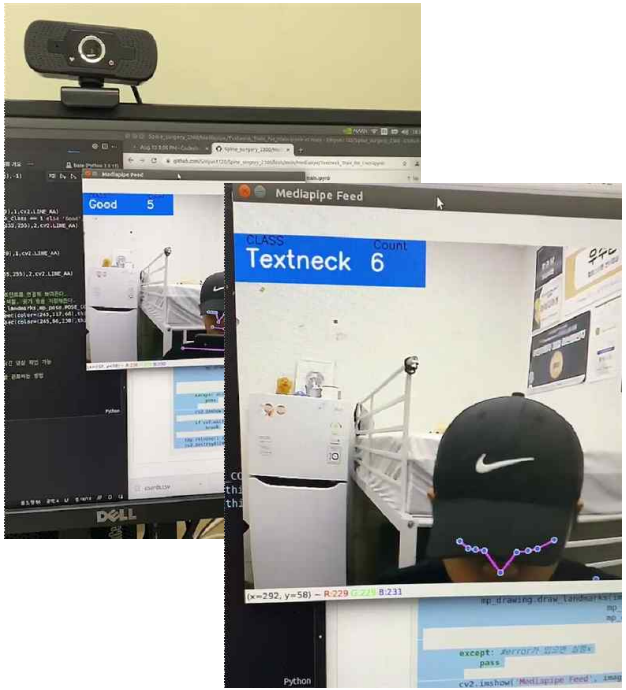
EDA 및 전처리

- 각도에 대한 정보 csv에 추가

->원래는 각 관절의 x,y,z좌표만 나타나 있었음. 하지만, 이전의 실험에서 확인해보았을 때 얼굴에 나타나는 **각도가 거북목을 판단**하는 데 중요한 지표가 될 것이라고 판단을 내림

-> 실제로 함수를 만들어 각도 column을 추가해주었고, **80% 내외의 강력한 상관관계**를 가지고 있음을 확인할 수 있었음.

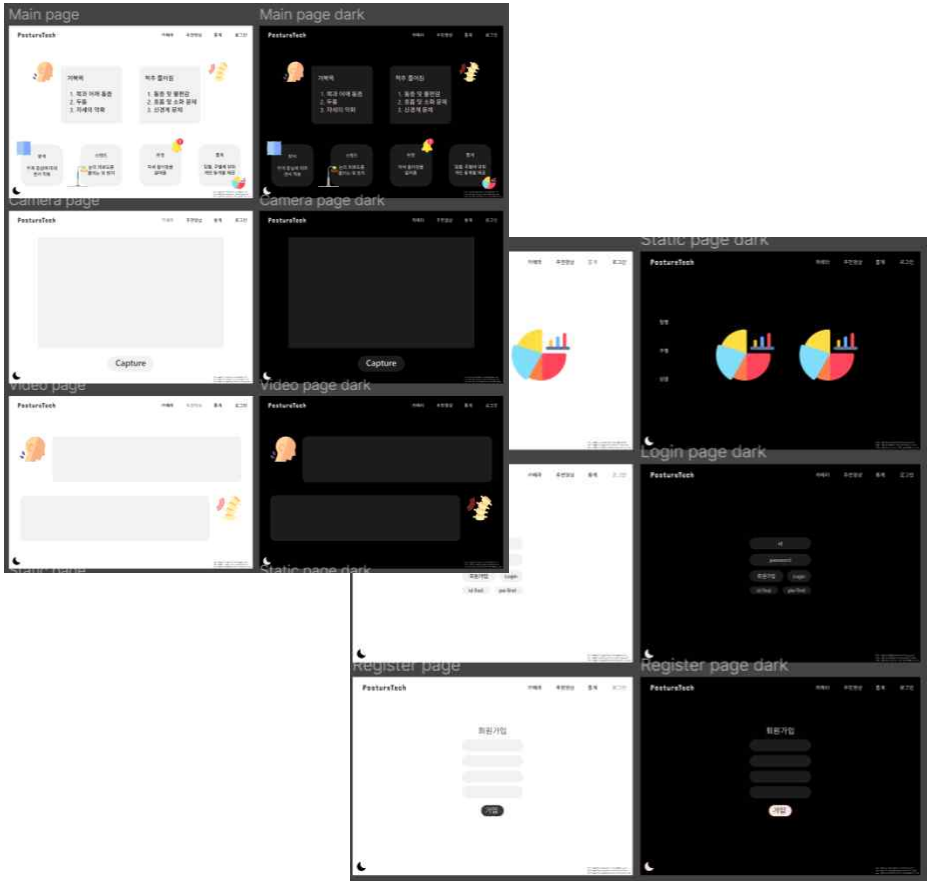
AI



- 젯슨나노에서 실행시키기 완료
- CUDA 설치를 통해 실시간 처리 할 예정

FRONT-END

- FIGMA 웹 디자인
- FIGMA DEVMODE기반 코딩
- 페이지 이동 버튼 구현



BACK-END

- Node.js 서버 구축
- MySQL DB 생성
- Express, Node.js, MySQL 연동
  회원가입, 로그인 구현

Plan

8/16(수)	CSV파일 수집, 통신, CUDA 세팅 완료
8/17(목)	MODEL 배포
8/18(금)	TEST
8/19(토)	검증
8/20(일)	모니터링 및 유지보수, PPT제작 시작
8/21(월)	PPT제작 완료
8/22(화)	발표 연습 및 마무리
8/23(수)	최종발표

Thank you