걸음걸이 인식 기술을 활용한 보안 시스템

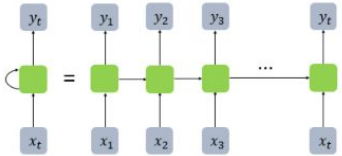
돌개바람
윤성규 박은서 권재민 전수환 이현빈

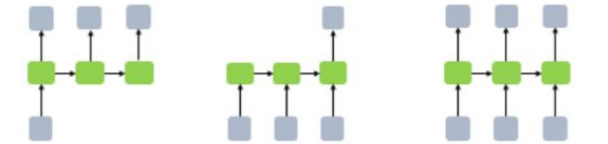
1. 사전 학습

머신러닝, kinect센서, openpose, RNN에 대해 팀원들이 각자 세미나를 준비해 진행하였다.

8. RNN의 기본 개념

RNN는 활성화 함수로 부터 나온 결과 값을
출력층과 다음 은닉층 노드로 보냄

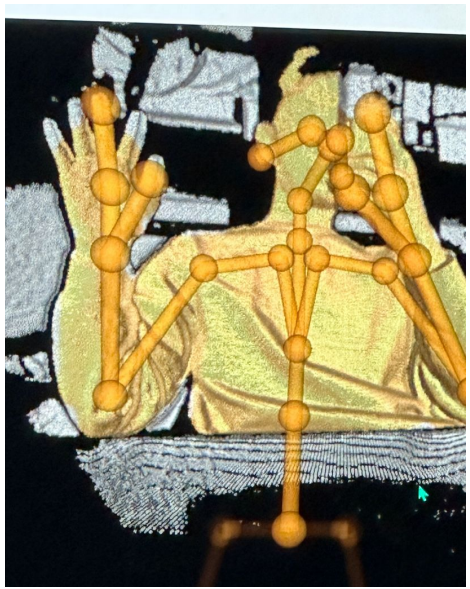




일 대 다(one-to-many) 다 대 일(many-to-one) 다 대 다(many-to-many)

2. 데이터 수집

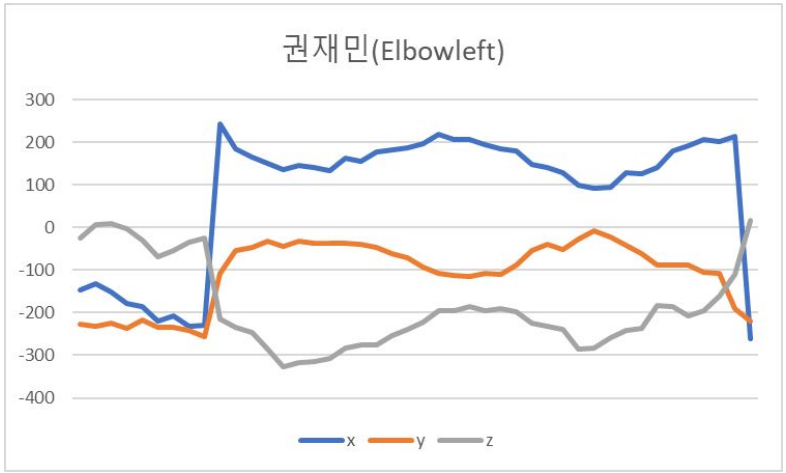
Kinect센서를 이용해 팀원들 개개인의 관절 데이터 값을 5개를 검출하였다.

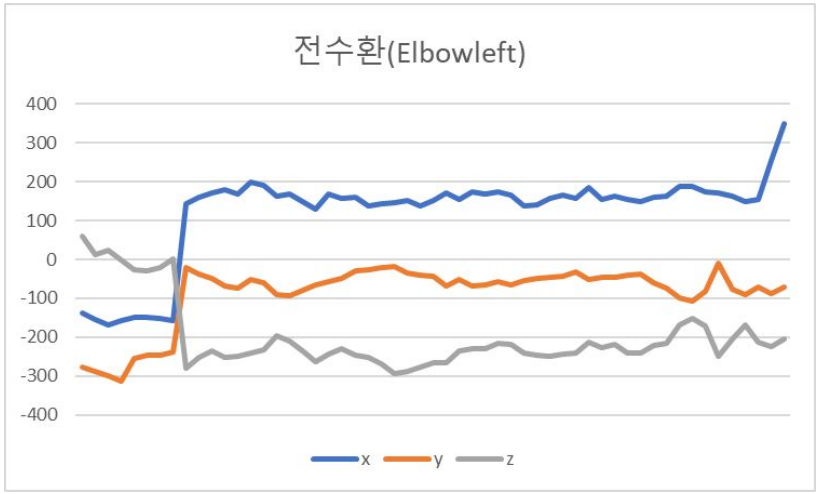


Count	Kind of Jc	X	Y	Z
1	Pelvis	0	0	0
1	Knee_Righ	194.7091	242.704	248.3975
1	Knee_Left	-105.0068	281.3219	249.5308
1	Ankle_Rig	302.9077	469.9652	512.7732
1	Ankle_Left	-140.2446	590.7492	432.5676
1	Hip_Left	-72.58289	-28.95898	36.12036
1	Hip_Right	99.48596	81.65881	-19.72095
1	SpineNave	2.249878	-133.5869	-102.5806
1	SpineChes	5.691895	-241.9082	-182.5095
1	Neck	43.39685	-409.7667	-293.9529
1	ShoulderR	122.6718	-267.4031	-332.9087
1	ShoulderL	-51.76831	-432.6051	-146.6333
1	ElbowRigh	103.0247	-10.16931	-268.7231
1	ElbowLeft	-118.4734	-254.5313	32.01465
1	WristRight	108.5829	135.0784	-101.2458
1	WristLeft	-132.6139	-80.38269	164.0259
1	Head	66.91455	-475.0901	-327.5408
1	FootRight	380.5919	515.1501	648.4451
1	FootLeft	-140.6422	667.84	591.762
2	Pelvis	0	0	0
2	Knee_Righ	155.1682	282.6002	265.4419
2	Knee_Left	-84.26563	283.5808	279.7021
2	Ankle_Rig	221.749	545.9737	539.5225
2	Ankle_Left	-97.02478	599.5557	493.7852
2	Hip_Left	-78.62659	-31.17725	33.60034
2	Hip_Right	94.17444	120.5037	-19.76807
2	SpineNave	10.62537	-142.0135	-106.9075
2	SpineChes	16.78247	-253.1687	-195.6379
2	Neck	64.02356	-429.1259	-312.9675

2. 데이터 수집

관절 데이터 값을 그래프로 나타내 사람마다 고유한 걸음걸이가 있음을 확인하였다.



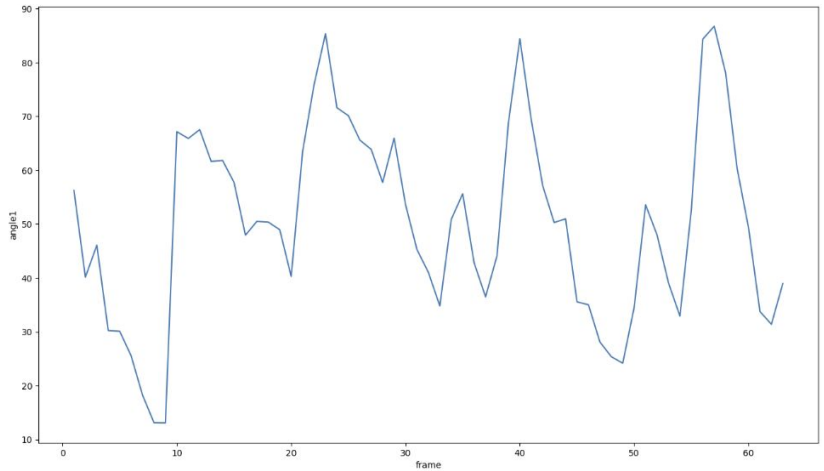


3. 데이터 전처리

RNN전에 동적, 정적데이터에 대해 전처리를 하였다.

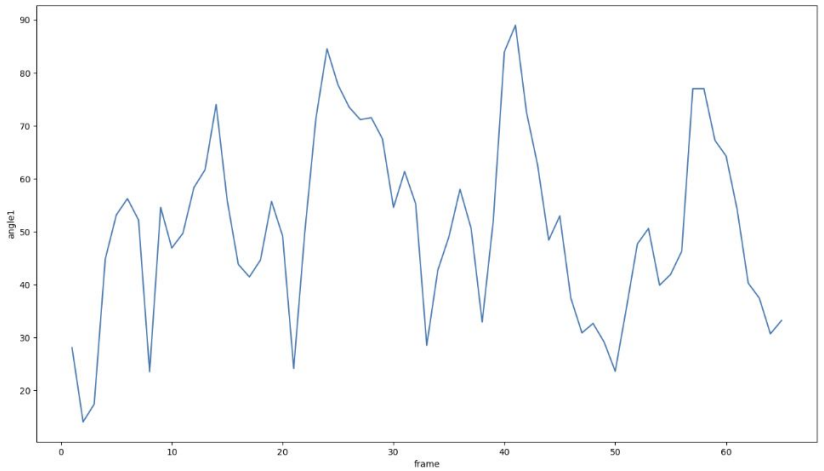
```
plt.figure(figsize=(16, 9))
x = [frame for frame, _ in ES1angleLFAK]  # x 값으로 frame을 추출
y = [angle for _, angle in ES1angleLFAK]  # y 값으로 angle1을 추출
sns.lineplot(x=x, y=y)
plt.xlabel('frame')
plt.ylabel('angle1')
```

```
Text(0, 0.5, 'angle1')
```




```
plt.figure(figsize=(16, 9))
x = [frame for frame, _ in ES2angleLFAK]  # x 값으로 frame을 추출
y = [angle for _, angle in ES2angleLFAK]  # y 값으로 angle1을 추출
sns.lineplot(x=x, y=y)
plt.xlabel('frame')
plt.ylabel('angle1')
```

```
Text(0, 0.5, 'angle1')
```



동일한 사람의 걸음걸이 그래프

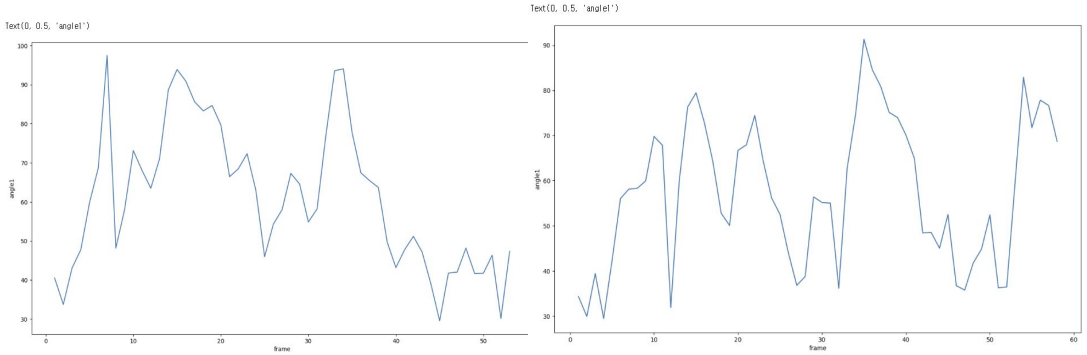
4. RNN

```
input_data = [
    [ES1avg_distance1, ES1avg_distance2, ES1avg_distance3, ES1angleLFAK, ES1angleRFAK,
    ES1angleLAKH, ES1angleRAKH, ES1angleLKHP, ES1angleRKHP, ES1anglePSN],
    [JM1avg_distance1, JM1avg_distance2, JM1avg_distance3, JM1angleLFAK, JM1angleRFAK,
    JM1angleLAKH, JM1angleRAKH, JM1angleLKHP, JM1angleRKHP, JM1anglePSN],
    [SH1avg_distance1, SH1avg_distance2, SH1avg_distance3, SH1angleLFAK, SH1angleRFAK,
    SH1angleLAKH, SH1angleRAKH, SH1angleLKHP, SH1angleRKHP, SH1anglePSN],
    [SG1avg_distance1, SG1avg_distance2, SG1avg_distance3, SG1angleLFAK, SG1angleRFAK,
    SG1angleLAKH, SG1angleRAKH, SG1angleLKHP, SG1angleRKHP, SG1anglePSN],
    [HB1avg_distance1, HB1avg_distance2, HB1avg_distance3, HB1angleLFAK, HB1angleRFAK,
    HB1angleLAKH, HB1angleRAKH, HB1angleLKHP, HB1angleRKHP, HB1anglePSN]
]

input_data_normalized = scaler.fit_transform(np.array(input_data))
```

```
# Min-Max 정규화
scaler = MinMaxScaler()
input_data_normalized = scaler.fit_transform(input_data.reshape(-1, 1)).reshape(-1, 10)

# LSTM 모델 구성
model = Sequential()
model.add(LSTM(units=64, input_shape=(10,)))  # 입력 데이터의 크기에 맞게 조정
model.add(Dense(units=1))

# 모델 컴파일
model.compile(optimizer='adam', loss='mse')

# 입력 데이터 형태 변경
input_data_reshaped = input_data_normalized.reshape(-1, 10, 1)  # 입력 데이터의 크기에 맞게 조정

# 모델 학습
model.fit(input_data_reshaped, target_data, epochs=10)  # target_data는 목표 변수에 해당하는 값

# 모델 예측
prediction_normalized = model.predict(input_data_reshaped)

# 예측 결과 역정규화
prediction = scaler.inverse_transform(prediction_normalized)

# 예측 결과 출력
print(prediction)
```

전처리한 데이터를 정규화시켜 **RNN** 알고리즘에 학습시킨 이후 유사도를 판정하였다.

5. 결과




```
유사도 판정 : 34.7% ... Person1은 박은서가 아닙니다.
유사도 판정 : 48.6% ... Person1은 이현빈이 아닙니다.
유사도 판정 : 60.3% ... Person1은 윤성규이 아닙니다.
유사도 판정 : 97.2% ... Person1은 전수환이 맞습니다.
유사도 판정 : 49.4% ... Person1은 권재민이 아닙니다.
```

- 걸음걸이를 **kinect** 센서로 찍어 관절의 좌표값을 얻어냈다.
- 머신러닝을 통해 개개인의 걸음걸이 특징을 구별해 냈다.
- 새로운 보행자의 걸음 걸이를 인식하여 기존에 학습 시켜놓은 데이터들과 유사도를 판정하는 프로그램을 구현하였다.

감사합니다