

우리도 차겠지

오윤재 박현서 송지영 이수정 정지현 사드

목차



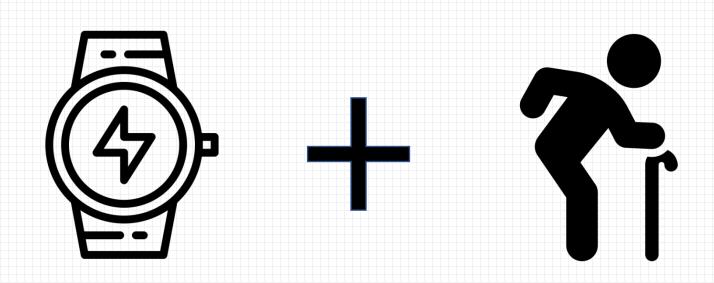
○**1. 프로젝트 목표**

○ **2.** 수행 내용 및 중간 결과

○ 3. 향후 추진 계획

프로젝트 목표

주제 선정 배경



스마트워치를 이용한 종합 노인 케어 시스템

SILVER WATCH 기능



수행 내용 및 중간 결과

Wear 앱

1 데이터 수집을 위한 앱을 제작하여 팀원의 워치에 모두 설치 후 실제 수집 진행.



데이터 수집 앱

	X	у	z
0	-0.347922	1.412436	9.371981
1	0.330484	1.817755	9.883181
2	0.121596	2.068804	10.851442
3	-0.942006	0.498314	10.119856
4	-1.779952	0.655459	9.350901
5	0.318985	2.505264	6.790112
6	-3.047650	1.026282	7.275439
7	-2.260489	1.442141	8.356768
8	-3.355712	1.254334	6.930488
9	-7.808467	-3.641589	3.516421
10	-12.247329	-2.758608	4.710338

수집된 csv파일

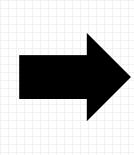
Wear 앱

- 2 Wear 앱에서 사용자를 구별하기 위한 방법으로 UUID생성.
- 3 필요한 기능을 테스트 할 수 있는 데모 앱 개발 완료.
- 4 센서 데이터 및 배터리 잔량을 전송하는 백그라운드 서비스 개발 중.

머신러닝

1 시계열 순서로 들어오는 데이터들을 일정 크기(series)로 잘라 데이터셋을 생성.

분류	데이터 개수	
낙상	543개	
추 걷기	713개	



분류	데이터 개수	
낙상	613개	
수 걷기	5756개	

머신러닝

2 시계열 데이터를 처리하기 위해 RNN의 일종인 LSTM이용해 분류기 모델 구현.

```
answer = 0
for i in range(len(submit['activity'])):
    if submit['activity'][i] == y_tst['activity'][i]:
        answer = answer + 1
print("Test Accuracy : ", answer / len(submit['activity']) *100, "%")
```

Test Accuracy: 96.45502645502646 %

테스트 데이터에 대한 정확도

머신러닝

학습된 LSTM Classifier를 Python Flask를 사용하여 API로 제공.

```
>>> r = requests.post('http://0.0.0.0:8000/', data=json.dumps({'id':'20171666','data': 'x,y,z\n-1.9514096,2.6419582,8.344559\n-1.5393872,2.18011,4.9477706\n-3.5444
026, -0.5033051, 6.283969 \\ n-6.0567803, 1.8634275, 3.97329 \\ n-10.898043, -0.8640642, -2.09
87687\n-21.095116,-5.0135117,-3.8100986\n-8.608825,-4.291711,-1.9870939\n-15.69743
2,1.678959,4.308383\n-9.351412,-1.8180516,-1.653336\n-10.835162,0.65084004,-1.9179
02958,3.9508402\n-12.677405,3.422084,1.7062722\n-15.747534,3.2238204,-2.5545316\n5
 .072305, 2.4042962, 9.258686 \\ n-0.10917139, 12.026088, -5.3779664 \\ n-0.73918164, 14.20732
      ,-4.1368904\n-5.6948605,12.650257,-2.8675735\n-2.71848,8.99585,-1.0327718\n-0.883
42106,9.895357,-0.8419029\n-7.0600076,5.9299917,-5.3249993\n-6.132478,11.796041,-8
 .054408\n-9.321817,4.9464083,1.8211924\n-12.845568,7.4266877,4.0130553\n-7.5874915
 ,3.973365,3.041449\\n-4.4992404,5.164876,-1.6776435\\n-2.0002773,7.8075686,-2.718718
 8\n-2.5325527,8.975125,-2.51654\n-3.1438787,8.647423,-2.4326987\n'}))
 >>> r = requests.post('http://0.0.0.8000/', data=json.dumps({'id':'20171666','da
ta': 'x,y,z\n-0.3479218,1.4124365,9.371981\n0.33048382,1.8177551,9.883181\n0.1215
9622,2.0688035,10.851442\n-0.9420058,0.4983138,10.119856\n-1.7799518,0.6554586,9.3
50901 \cdot n0.3189854, 2.5052638, 6.7901115 \cdot n-3.0476503, 1.026282, 7.2754393 \cdot n-2.260489, 1.4
421408,8.356768\n-3.3557117,1.2543336,6.9304876\n-7.808467,-3.6415894,3.5164208\n-
12.247329, -2.758608, 4.710338 \cdot n-11.148752, -2.500852, 2.520372 \cdot n-11.883692, -2.021752,
1.4491043 \ln -10.381234, -2.5736754, 2.7886682 \ln -9.430221, -2.4495883, 2.0019855 \ln -14.4491043 \ln -10.381234, -2.5736754, 2.7886682 \ln -9.430221, -2.4495883, 2.0019855 \ln -14.4491043 \ln -10.381234, -2.5736754, 2.7886682 \ln -9.430221, -2.4495883, 2.0019855 \ln -14.4491043 \ln -10.381234, -2.5736754, 2.7886682 \ln -9.430221, -2.4495883, 2.0019855 \ln -14.4491043 \ln -10.381234, -2.5736754, 2.7886682 \ln -9.430221, -2.4495883, 2.0019855 \ln -14.449104, -2.5736754, -2.4495883, -2.0019855 \ln -14.449104, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5736754, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, -2.5756744, 
9272,-2.4031157,-0.09311859\n-10.343386,0.18689878,-0.82997435\n-11.249842,-3.2305
214,-1.2616435\n-9.506876,0.09970259,-0.9665179\n-7.2244444,-1.3337647,-0.9085468\
n-5.585922,0.0867669,-0.15013151 \\ n-9.719597,-0.122599766,-1.570184 \\ n-15.944544,-3.
0738559, -1.671274 -12.899384, 1.3932726, 0.3093254 -1.8673964, -1.9187458, 1.141043
n-8.75421, -3.645422, -0.3417715 \\ \\ n-14.400884, -4.2720847, -1.5572482 \\ \\ n-18.420053, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, -2.72120847, 
74666, -4.204755 \cdot n-15.752904, 0.2170821, -3.3557897 \cdot n-10.113417, 0.46573505, -1.7939236
```

```
* Serving Flask app "__main__" (lazy loading)

* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

* Debug mode: off

* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
127.0.0.1 - - [26/May/2020 17:47:07] "POST / HTTP/1.1" 200 -

Predicting on test dataset
[0]

127.0.0.1 - - [26/May/2020 17:47:11] "POST / HTTP/1.1" 200 -

Predicting on test dataset
[1]
```

0: 낙상 1: 걷기

안드로이드 앱

1 레이아웃 디자인 수정.

- 메인 메뉴 화면을 linear layout으로 변경.

- 메뉴 버튼을 설명하는 text view를 더욱 크게 설정.



안드로이드 앱

2 QR코드 스캔 기능 추가.

- QR코드 스캔 버튼 및 사용자 정보 입력 버튼 추가.
- 워치 등록 메뉴에서 사용자 정보를 전달 받는다.
- 사용자 정보를 aws 서버에 전달.
- 서버로 제대로 전달하지 못하는 경우를 대비하여 Exception Handling 구현.

워치 등록 메뉴

01-

전화번호(000-0000-0000)

워치 아이디

QR 코드 스캔 (워치 아이디 가져오기)

새로운 사용자 등록하기

메인 메뉴로 돌아가기

안드로이드 앱

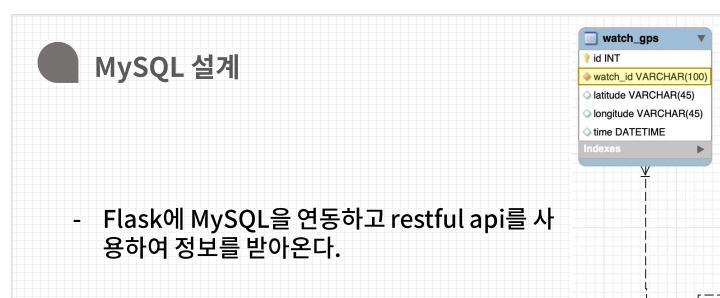
3 접근 권한 부여 알림 설정

앱을 작동시키기 위한 권한들을 사용자로부터 받기 위하여 권한 부여 신청 알림을 만들어 놓았다.

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

접근 권한: 문자 보내기, 인터넷 사용, 카메라 사용, 연락처 접근

데이터베이스



- 기능별로 table을 제작.

watch_battery id INT watch_id VARCHAR(100) watch_battery VARCHAR(45) time DATETIME watch_wear watch_id VARCHAR(100) wear TINYINT(1) watch_user watch_id VARCHAR(100) name VARCHAR(45) phone_number VARCHAR(45)

서버

- 1 서버는 GET과 POST 을 사용하여 데이터를 처리.
- Firebase를 사용해 안드로이드 앱 푸시알림 제작.



향후 추진 계획

Wear 앱 & 머신러닝

Wear 앱

백그라운드 서비스 개발을 완료한다. 제작한 데모 앱을 바탕으로 서비스를 백그라운드에서 작동할 수 있 게 변경하고, 모든 기능을 통합한다.

머신러닝

API 제공을 발전시킨다.

- 서버 API에 연동해 낙상여부에 따라 처리가 가능하도록 만들 예정이다.
- 모델의 추가 학습을 위해 다른 저장소에 업로드 하는 기능 구현.

안드로이드

- 1 사용자 등록과 동시에 작성 정보를 서버에 보내고 db 에 저장하는 기능 구현.
- 사용자 관리에서 등록된 기기의 사용자 인적사항을 list 형식으로 확인하는 기능 구현.
- google map api를 통해 현재 워치 사용자의 위치를 관리자 앱에서 확인하는 기능 구현.
- 4 기기의 배터리가 충전이 필요하면 관리자 어플에 푸시 알림이 오는 기능 구현.
- recyclable view를 활용하여 어떤 단말기에서도 레이아웃들이 동일하게 보이는 기능 구현.

서버 & 데이터베이스

- 1 스마트 워치의 사용자의 행동 반경을 MySQL에 저장할 수 있도록 구현.
- 현재 개발 중인 안드로이드 개발 앱에 푸시알림 기능을 추가하여 완성할 예정.

