

CNN_LSTM_AF_Detection

Development Guide Documentation

July, 2022

Contents

1. 시스템 구조

2. CNN-LSTM Deep Learning Model

2.1 CNN-LSTM 모델 구조

2.2 학습 및 테스트 데이터

2.3 CNN-LSTM 모델 학습 및 검증 방법

2.4 상세설계 및 소스코드 블록도 설명

3. Tizen-Android Application

3.1 시스템 구조

3.2 블루투스 통신

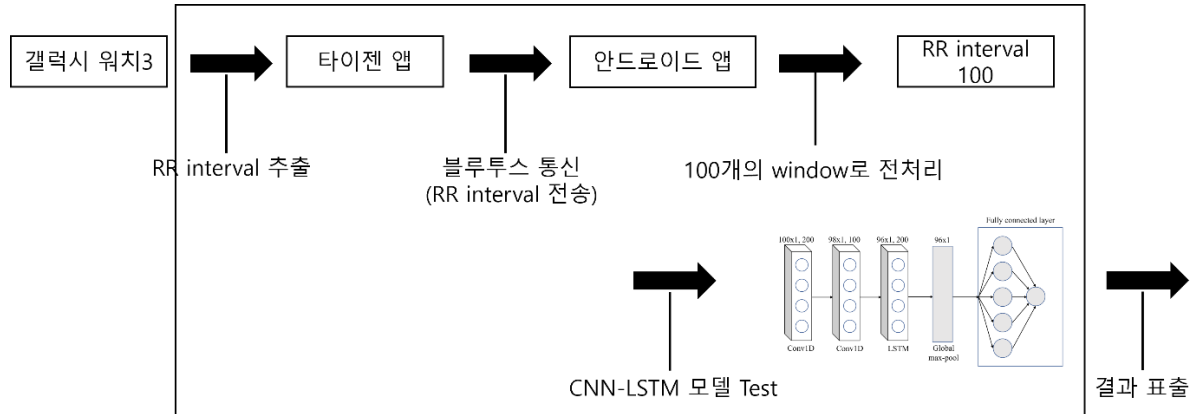
3.3 Tizen 설치 및 기기 연결

3.4 기술적 추가설명 - Tizen

3.5 기술적 추가설명 - Android

4. Contacts

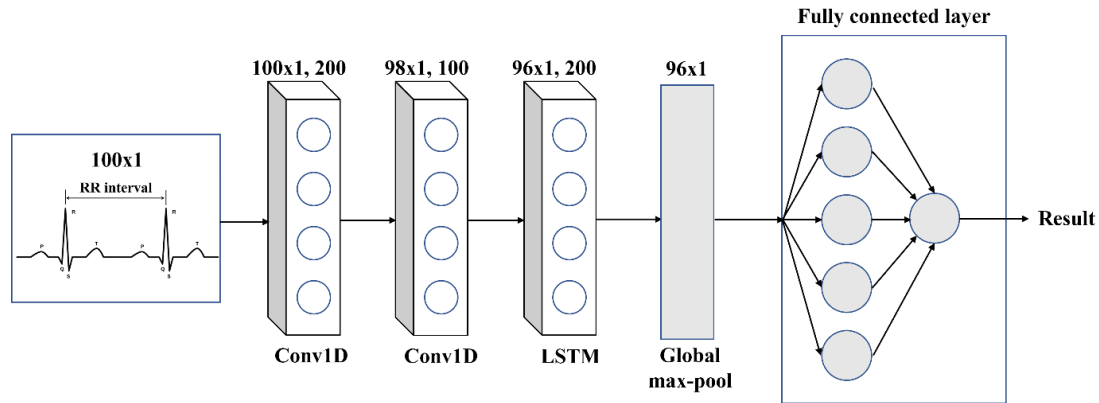
1. 전체 시스템 구조



전체 시스템의 구조는 위와 같이 구성해 설계해 보았다. 갤럭시 워치 3에서 타이젠 앱을 통해 RR interval을 추출 후 안드로이드 앱과의 블루투스 통신을 통해 추출된 RR interval을 전송한다. 전송된 RR interval의 실시간 데이터들은 자바 안드로이드 앱에서 100개의 segment로 하나의 윈도우를 이루도록 전처리되고 미리 학습된 CNN-LSTM 모델을 통해 테스트 되어서 심방세동인지 정상인지 결과를 앱 화면에 출력되도록 한다. 하지만 본 캡스톤 연구에선 갤럭시 워치3에서 RR interval 데이터를 자바 안드로이드 앱으로 전송하는 것까지 구현했고 제안된 딥러닝 모델로 테스트하여 결과를 표출하는 과정은 구현되지 않은 상태이다.

2. CNN-LSTM Deep Learning Model

2.1 CNN-LSTM 모델 구조





본 프로젝트에서 구현된 모델은 2개의 1D convolution neural network(CNN) 층들과 1개의 long short-term memory(LSTM) 층의 조합으로 이루어져 있다. 2개의 1D CNN 층의 각각의 filter의 개수는 200과 100으로 설정하였고 kernel 크기는 동일하게 3으로 설정하였다. 단일 LSTM 층의 셀 수는 입력 시퀀스 길이의 2배로 설정되었으며 최종 분류를 수행하는 상위 모델로 사용되었다. 1차원의 Global max pooling은 단방향 LSTM 층과 완전 연결 층들 사이에 사용되어 2개의 convolution layer 층과 단방향 LSTM 층에 의해 생성된 출력 시퀀스들의 특징들을 압축했다. CNN-LSTM 층은 효과적으로 input HR(Heart rate) 데이터 시퀀스의 특징들을 학습하고 추출한 후, 이러한 특징들을 완전 연결 계층에 전달하여 심방세동 증상이 있는지 여부를 분류한다.

2.2 학습 및 테스트 데이터

Database Open Access

MIT-BIH Atrial Fibrillation Database

George Moody , Roger Mark 

Published: Nov. 4, 2000. Version: 1.0.0

When using this resource, please cite the original publication:

Moody GB, Mark RG. A new method for detecting atrial fibrillation using R-R intervals. Computers in Cardiology. 10:227-230 (1983).

Please include the standard citation for PhysioNet: (show more options)

Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215-e220.

실험은 MIT-BIH Atrial Fibrillation Database의 데이터를 기반으로 수행되었다. 이 데이터 베이스는 서로 다른 피험자의 23개의 장기 심전도 신호 기록이다. 각 피험자의 데이터 세

트는 심방세동 주석을 가지고 있는 250 Hz로 샘플링 된 10시간 동안의 두 개의 심전도 신호이다. 이 기록에는 전문 임상이가 직접 표시한 심장 박동 주석과 리듬 주석도 포함된다. 또한, R 피크는 라벨로 표시되며 RR 간격 시퀀스는 이러한 라벨에 기초하여 추출되었다. RR 간격 시퀀스는 100개의 윈도우로 나누어져 있고 99개씩 겹치도록 분할되어 각 피험자의 데이터 세트를 이룬다. 심방세동의 징후를 보이는 것으로 분류된 박동이 하나 이상 포함된 경우 RR 간격 시퀀스는 심방세동으로 라벨링 되었으며, 다른 모든 시퀀스는 정상으로 라벨링 되었다. 실험에 사용된 정상과 심방세동 심전도 신호의 RR 간격 데이터 세트의 정규분포도는 위 그림에 나와있다. 20명의 환자의 데이터는 모델의 훈련을 위해 사용되었고, 나머지 3명의 환자의 데이터는 모델의 훈련 및 검증 이후 블라인드 검증을 위해 보류되었다. 블라인드 검증 단계는 제안된 실험이 임의의 데이터 뿐만이 아니라 임의의 환자들에게도 일반적으로 적용되는 것을 보장한다.

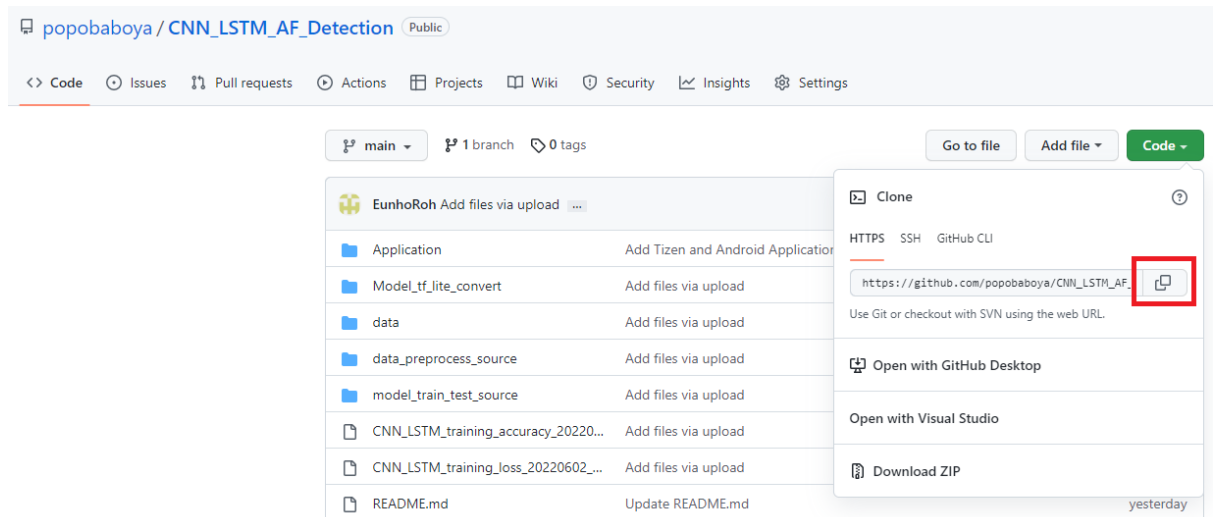


학습 및 훈련데이터인 csv 파일에서 맨 왼쪽은 심방세동 라벨링 데이터이며, 각각의 값은 해당 행의 심방세동 주석의 RR interval 데이터를 개수를 의미하고, 1열을 제외한 나머지 100개의 열은 순차적인 RR 간격 데이터이다. 따라서 심방세동 라벨링이 0일 경우는 해당 행의 100개의 RR 간격 데이터 중 심방세동을 포함한 데이터가 없다는 것을 의미하고, 심방세동 라벨링이 10일 경우는 해당 행에 심방세동을 포함한 RR 간격이 10개가 있다는 것을 의미한다.

2.3 CNN-LSTM 모델 학습 및 검증 방법

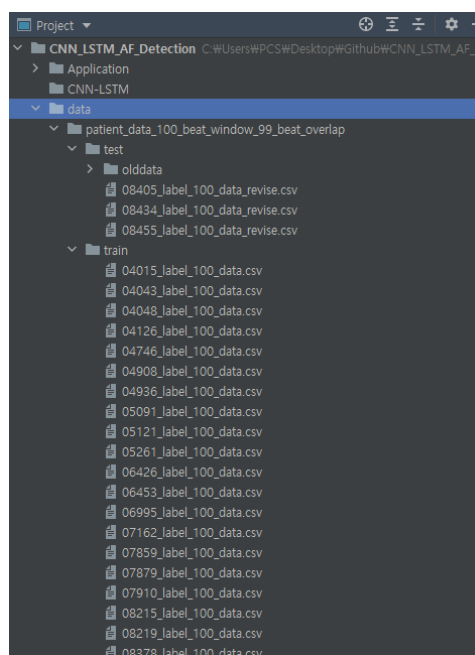
a) Github에서 Repository clone 또는 ZIP 파일 다운로드

Github 주소: https://github.com/popobaboya/CNN_LSTM_AF_Detection



b) 훈련 및 검증 데이터 확인

Git clone 을 올바르게 했다면 CNN_LSTM_AF_Detection/CNN-LSTM/data/ 경로에서 test 데이터와 train 데이터를 확인할 수 있다.



c) tmux 생성하기

터미널에서 `tmux new -s test1` 입력

```
heartsavior@bcilab_server1:~/blstm_model_v2$ tmux new -s test1
```

d) test1 으로 이동 후 가상환경 활성화하기

터미널에서 `tmux attach -t test1` 입력

터미널에서 `source myenv/bin/activate` 입력

```
heartsavior@bcilab_server1:~/blstm_model_v2$ tmux attach -t test1
```

```
heartsavior@bcilab_server1:~/blstm_model_v2$ source myenv/bin/activate
```

e) 모델 훈련시키기

터미널에서 `python3 CNN_LSTM_Model_Evaluate.py` 입력

```
(myenv) heartsavior@bcilab_server1:~/blstm_model_v2$ python3 CNN_LSTM_Model_Evaluate.py
```

f) 모델 훈련 화면 확인

```
Epoch 00031: val_loss improved from 0.04185 to 0.04055, saving model to ./model/CNN_LSTM/initial_runs_20220426_2028/CNN_LSTM_af_lstm_weights.31-0.04.hdf5
Epoch 32/100
355/355 [=====] - 106s 298ms/step - loss: 0.0456 - acc: 0.9859 - val_loss: 0.0403 - val_acc: 0.9873
Epoch 00032: val_loss improved from 0.04055 to 0.04030, saving model to ./model/CNN_LSTM/initial_runs_20220426_2028/CNN_LSTM_af_lstm_weights.32-0.04.hdf5
Epoch 33/100
355/355 [=====] - 104s 293ms/step - loss: 0.0831 - acc: 0.9719 - val_loss: 0.1052 - val_acc: 0.9669
```

2.4 상세설계 및 소스코드 블록도 설명

Script	Description
CNN_LSTM_Model_Evalute.py	CNN-LSTM 모델을 생성하고 지정된 데이터를 통해 학습을 진행하는 스크립트
CNN_LSTM_Blindfold_evaluate.py	학습이 끝난 모델의 가중치 파일을 사용하여 모델의 성능을 검증하는 스크립트
af_blind_fold_data_exploration.py	3명의 심방세동 환자의 RR interval csv 파일을 통해 test 용 npz 파일을 생성하는 스크립트
af_data_exploration.py	20명의 심방세동 환자의 RR interval csv 파일을 통해 training and validation npz 파일을 생성하는 스크립트
tensorflowlite convert_CNN_LSTM.ipynb	학습이 끝난 모델을 tensorflow lite 파일로 변환하기 위한 스크립트
AnnotationExtract_100windowSplit.ipynb	학습에 사용되는 csv 파일을 생성하는 스크립트. RR 간격의 sequence 데이터를 100개 단위로 자르고 각 단위마다 심방세동 포함 개수를 표시한다.
tizenApp.js	Tizen 앱의 동작 function 들을 포함하는 스크립트. 블루투스 통신, PPG 데이터의 측정 코드를 포함한다.
ConsumerActivity.java	Android 앱의 function 들을 포함하는 스크립트. 블루투스 통신, 버튼과 관련된 코드를 포함한다.

2.5 기술적 추가설명

```
n_epochs = 80
batch_size = 1024
```

```
opt = Adam()
```

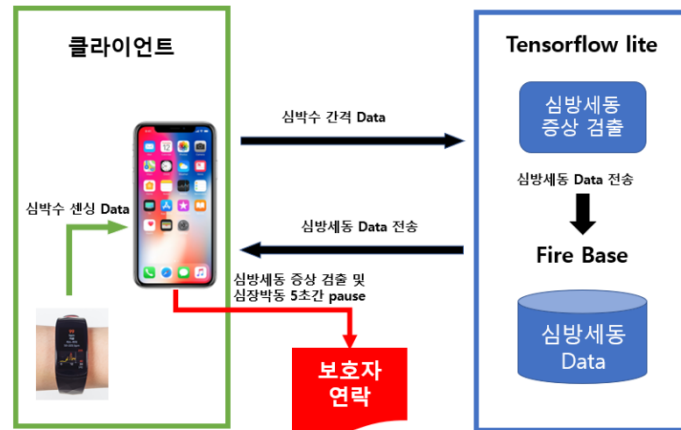
훈련 과정에서 epoch 수는 80으로 batch size는 1024로 설정하였다. 이는 GPU 메모리와 훈련 속도 간에 좋은 균형을 이룰 수 있기 위해 설정한 값이다. 그리고 모델의 가중치를 업데이트 하기 위해 Adam optimizer를 이용한 gradient descent 알고리즘을 사용하였고, 딥러닝 신경망의 손실값을 평가하기 위해 binary cross-entropy 함수를 사용하였다.


```
model = Sequential()
model.add(layers.Conv1D(200, 3, activation='relu',
                        input_shape=inp))
model.add(layers.Conv1D(100, 3, activation='relu'))
model.add(layers.LSTM(200, return_sequences=True, dropout=0.1, recurrent_dropout=0.1))
model.add(GlobalMaxPooling1D())
model.add(Dense(50, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1, activation='sigmoid'))
```

LSTM 셀들의 input들과 hidden states 둘 다에 recurrent dropout을 0.1로 적용하였고, standard dropout도 0.1로 모델 과적합을 줄이고 모델 일반화를 개선하기 위해 완전 연결 층들 사이에 적용하였다. 완전 연결 계층은 50층과 1층으로 은닉층들을 구성하였고 relu 함수로 뉴런을 활성화하고 sigmoid 함수로 데이터 시퀀스를 분류하였다. 블라인드 검증을 통해 모델 성능을 평가하고 모델 구조와 초매개변수들을 조정하였다.

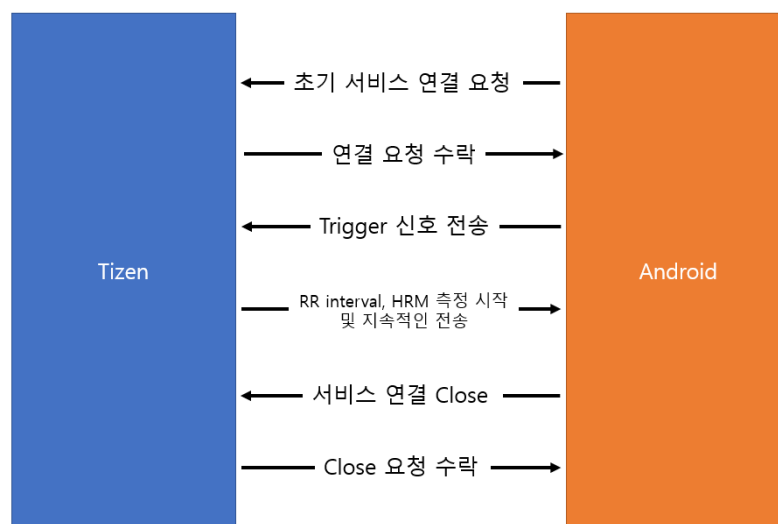
3. Tizen-Android Application

3.1 시스템 구조



설계된 CNN-LSTM 딥러닝 모델을 사용하여 심방세동을 검출하는 서비스의 시스템을 설계하였다. 웨어러블 기기에서 RR interval 과 HRM 데이터를 안드로이드 기기로 전송하면 안드로이드 앱에서는 해당 데이터를 CNN-LSTM 모델을 사용하여 심방세동 유무를 판별한다. 안드로이드 앱에서는 검출 결과를 출력하며, 심방세동 발생시 또는 심박이 일정 시간 이상 멈출 시 보호자에게 연락을 취한다.

3.2 블루투스 통신



Tizen 과 안드로이드 앱의 통신은 블루투스를 통해 이루어지며, 안드로이드 앱에서 연결 요청을 보낼 시 Tizen 쪽에서 연결을 수락한다. 연결이 이루어지면 안드로이드에서 특정 신호를 전송하고 Tizen 에서는 해당 신호를 수신한 시점부터 RR interval 과 HRM

데이터의 측정을 시작한다. HRM 데이터의 경우 중복되지 않는 값 10개가 측정되면 전송을 하며, RR interval 의 경우 중복되지 않는 값 100개가 측정되면 전송을 한다. 이는 HRM 데이터를 비교적 자주 받아 사용자에게 심박수 정보를 짧은 주기마다 업데이트 해주기 위함이다. RR interval 의 경우, 딥러닝 모델에서 사용되는 RR interval 의 단위가 100개 이므로 100개 마다 전송을 해준다.

3.3 Tizen 설치 및 기기 연결

Tizen Studio 설치 및 기기 연결은 다른 플랫폼과 비교했을 때 비교적 까다로울 수 있다. 따라서 초기 Tizen 개발 환경 세팅을 진행할 때 아래 링크의 참고를 추천한다. 주의할 점은 바로 아래 링크에서 사용하는 예제는 native 앱 용이지만 해당 프로젝트에서 사용한 예제는 Web 앱 용이다. 아래에 Web 앱 용 예제의 링크도 추가적으로 첨부하였다.

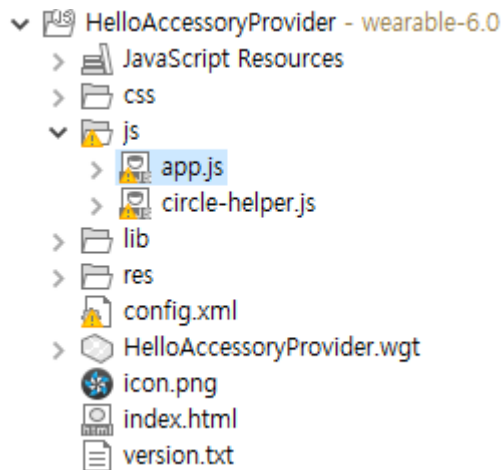
삼성 Gear에서 Android 스마트폰으로 센서 데이터 보내기 – 환경설정편

<https://jaehoonx2.tistory.com/52?category=840350>

Samsung Developers : Hello Accessory – Web

<https://developer.samsung.com/galaxy-watch-tizen/samples/companion/hello-web.html>

3.4 기술적 추가설명 – Tizen



Tizen 에서 핵심이 되는 파일은 app.js 이며, 아래에 나오는 설명들 모두 해당 파일 내부에 있는 코드들이다.

```
tizen.power.request('SCREEN', 'SCREEN_NORMAL');  
//tizen.power.request("CPU", "CPU_AWAKE");
```

갤럭시 워치3 의 경우 해당 앱의 실행 중 화면이 꺼질 경우 통신이 원활하게 이루어지지 않는 문제가 존재했다. 이를 방지하기 위하여 위에 있는 화면이 꺼지지 않는 코드를 추가하였다.

```
SASocket.setSocketStatusListener(onConnectionLost);  
  
dataOnReceive = function dataOnReceive (channelId, data) {  
  
    tizen.humanactivitymonitor.start('HRM', onChangedCB);  
    HRMrawsensor.start(onsuccessCB1);
```

Android 로부터 처음 데이터를 받게 되면 HRM 데이터와 RR interval 의 측정을 시작하게 된다.

```

if (hrm_arr.length == 10){
    var result = hrm_arr.reduce(function add(sum, currValue) {
        return sum + currValue;
    }, 0);

    var average = result / hrm_arr.length;

    hrm_arr.push(average)
    SASocket.sendData(SAAgent.channelIds[0], hrm_arr);
    console.log("HRM has been sent");
    hrm_arr = [];
}

if (selected_rr.length == 100){
    SASocket.sendData(SAAgent.channelIds[0], selected_rr);
    selected_rr = [];
    console.log("100 RR intervals have been sent");
}

```

HRM 데이터의 경우 10개 단위로 전송되며, RR interval 의 경우 100개씩 전송된다. HRM 은 사용자에게 비교적 자주 제공돼야 하고 RR interval 은 100개가 딥러닝 모델 input 의 최소단위이기 때문에 이렇게 설정하였다.

3.5 기술적 추가설명 – Android

```

public void mOnClick(View v) {
    switch (v.getId()) {
        case R.id.buttonConnect: {
            if (mIsBound == true && mConsumerService != null) {
                mConsumerService.findPeers();
                sendingData();
            }
            break;
        }
    }
}

```

```

public void sendingData(){
    connectThread = new Thread((Runnable) () -> {
        try {
            if(count == 0){
                Thread.sleep( time: 1500);
            }
            else Thread.sleep( time: 1000);

            if (mConsumerService.sendData("Hello Accessory!")) {
            } else {
                Toast.makeText(getApplicationContext(), "Service Connection has already been disconnected!!!", Toast.LENGTH_LONG).show();
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    });
    connectThread.start();
}

```

Android 에서는 app/java/consumerActivity 파일과 app/java/consumerService 파일이 중요하다. 위 코드는 consumerActivity 파일의 일부이다. 특정 버튼을 클릭하게 되면 sendingData() 가 실행되고, sendingData() 는 tizen 앱에 데이터를 전송한다. 이 때 안드로이드에서 보내는 데이터의 내용은 크게 중요하지 않다. 안드로이드에서 처음 전송하는 내용은 단순히 Tizen 에서 센서 측정 시작을 위한 Trigger 역할을 할 뿐이기 때문이다. 추가적으로 sendingData() 안에는 Thread 를 사용하는데, Thread 를 사용하지 않을 경우에는 앱이 정상적으로 동작하지 않는다.

main CNN_LSTM_AF_Detection / CNN-LSTM / Model_tf_lite_convert / Go to file Add file ...

popobaboya Modify folders c19274d 2 hours ago History		
..		
CNN_LSTM	Modify folders	2 hours ago
pb	Modify folders	2 hours ago
tf	Modify folders	2 hours ago
CNN_LSTM_af_lstm_weights.66-0.00.hdf5	Modify folders	2 hours ago
tensorflowlite convert.ipynb	Modify folders	2 hours ago
tensorflowlite convert_CNN_LSTM.ipynb	Modify folders	2 hours ago
test.py	Modify folders	2 hours ago

Android 앱에서 CNN-LSTM 모델을 사용하기 위해서는 tensorflow lite 를 사용해야한다. 기존의 tensorflow 는 파이썬 코드로 동작하지만 tensorflow lite 는 파이썬에서 만들어진 모델을 자바 언어에서 돌릴 수 있도록 변환된 파일을 사용한다. 변환된 파일은 해당 프로젝트의 Github 폴더의 CNN-LSTM/Model_tf_lite_convert 안에 있다.

4. Contacts

박천성 popobaboya@gmail.com

노은호 deu04189@gmail.com