基於GARMIN智慧手錶之個人化日程行為記憶助理

0816139 許綾恩、0816055 羅晴、0816091 荊姿芸

目錄

摘要

- 本專題利用 Garmin 智慧手錶收集資料做人類行為分析，提出能讓智慧手錶配戴者無論在何處都能分析並記錄其日程行為的方法。
- 透過我們在手錶上開發的 CIQ App，將資料上傳到 IoTtalk 平台，同時利用Dashboard和Dummy_Device記錄手錶配戴者的日程行為。
- 針對收集到的資料，利用六種演算法做行為識別並比較效果優劣。

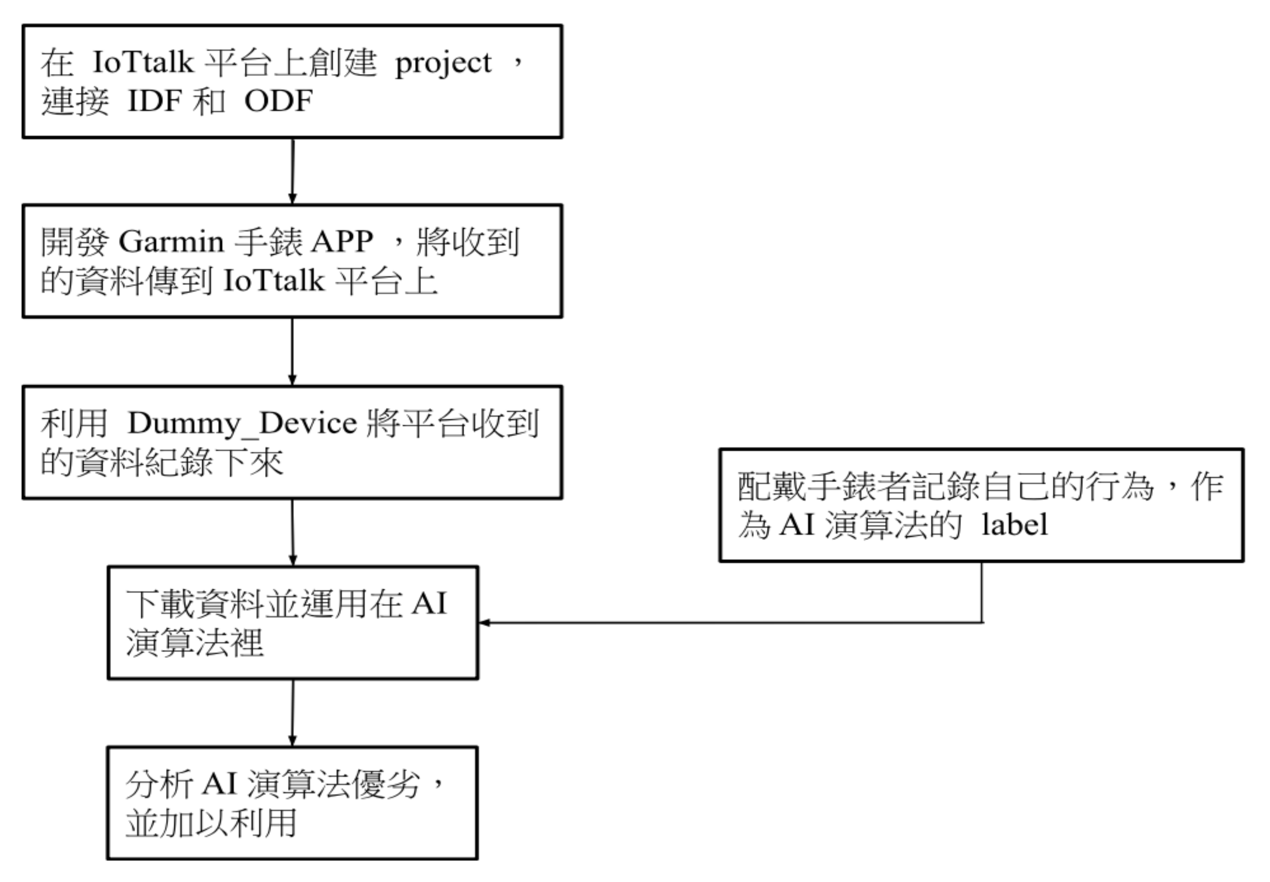
研究動機與目的

- 利用這幾年的所學應用到生活中，對目前的生活方式作出改善。
- 透過物聯網平台，最大程度的發揮智慧手錶的用途，改良智慧手錶在生活中的應用。

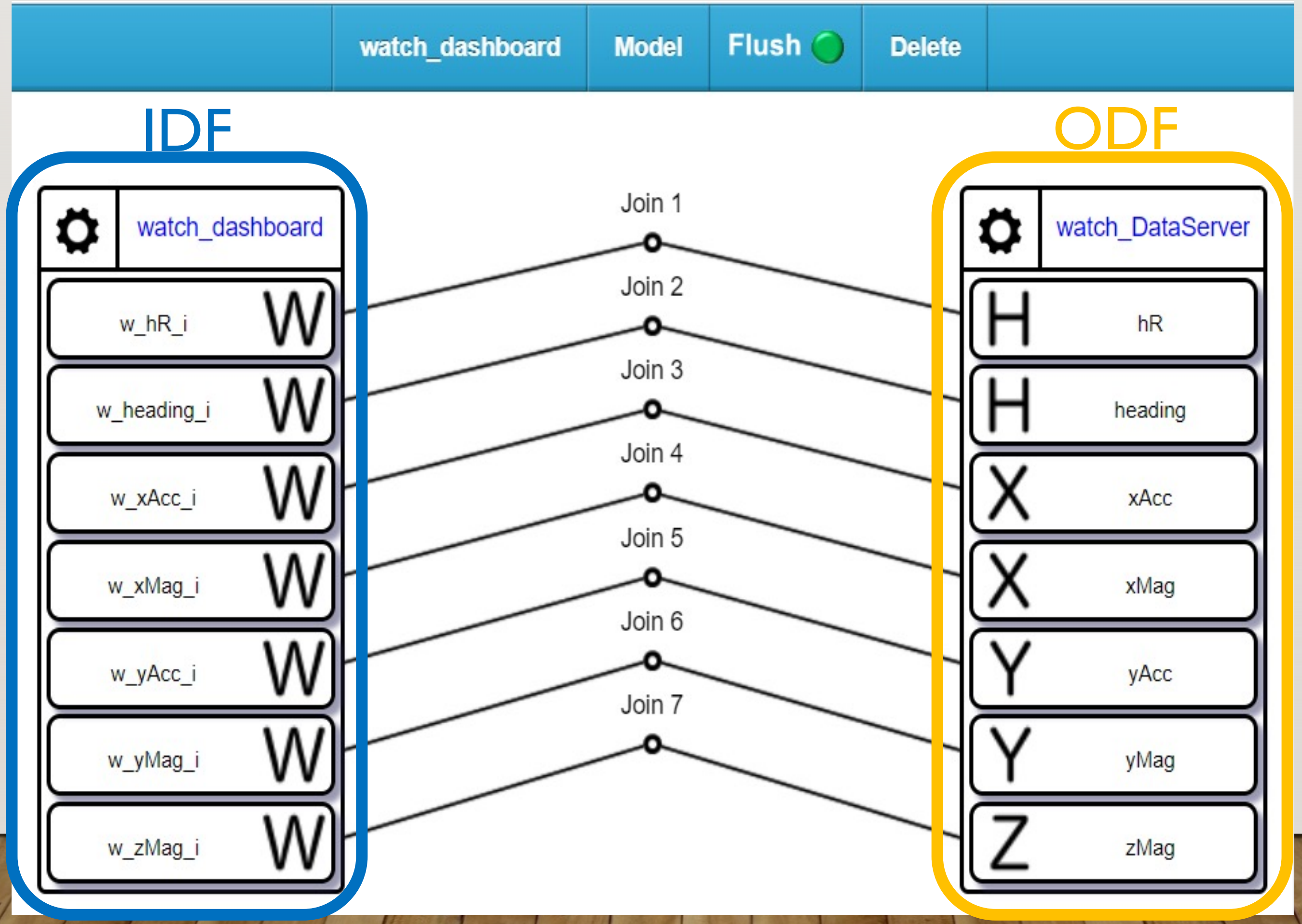
相關研究概況及比較

- 近年針對使用智慧手錶的行為識別研究相對著重在 CNN-LSTM 演算法上，透過使用卷積神經網路和長短期記憶網路進行深度學習與分析，著重在資料處理的成果和精準度。
- 研究中資料的收集都只針對特定地點，一但超出範圍便無法收集資料。

研究架構


在 IoTtalk 平台上創建 project，連接 IDF 和 ODF
開發 Garmin 手錶 APP ，將收到的資料傳到 IoTtalk 平台上
利用 Dummy_Device 將平台收到的資料紀錄下來
配戴手錶者記錄自己的行為，作為 AI 演算法的 label
下載資料並運用在 AI 演算法裡
分析 AI 演算法優劣，並加以利用

研究方法-IOTTALK



研究方法-CIQ APP

Register API function

```
function register() {
    var params = {
        "profile" =>
        {
            "d_name" => "watch_dashboard",
            "dm_name" => "watch_dashboard",
            "u_name" => "yb",
            "is_sim" => false,
            "df_list" => ["w_hR_i", "w_heading_i", "w_xAcc_i", "w_xMag_i", "w_yAcc_i", "w_yMag_i", "w_zMag_i"]
        }
    };
    var headers = {
        "Content-Type" => Communications.REQUEST_CONTENT_TYPE_JSON
    };
    var options = {
        :headers => headers,
        :method => Communications.HTTP_REQUEST_METHOD_POST,
        :responseType => Communications.HTTP_RESPONSE_CONTENT_TYPE_JSON
    };
    Communications.makeWebRequest("https://1.iottalk.tw/watch_dashboard", params, options, method(:onReceive));
}
```

- d_name: device name
- dm_name: device model name
- u_name: yb
- is_sim: false
- df_list: 一個 list 裡面列出 device 擁有的 feature name 。

研究方法-CIQ APP

```
var headers = {
    "Content-Type" => Communications.REQUEST_CONTENT_TYPE_JSON
};
var options = {
    :headers => headers,
    :method => Communications.HTTP_REQUEST_METHOD_PUT,
    :responseType => Communications.HTTP_RESPONSE_CONTENT_TYPE_TEXT_PLAIN
};
var sensorInfo = Sensor.getInfo();
var xAccel = 0;
if (sensorInfo has :accel && sensorInfo.accel != null) {
    xAccel = sensorInfo.accel[0];
}
else {
    xAccel = 0;
}
Communications.makeWebRequest("https://1.iottalk.tw/watch_dashboard/w_xAcc_i", {"data" => [xAccel.toNumber()]}, options, method(:onReceive1));
```

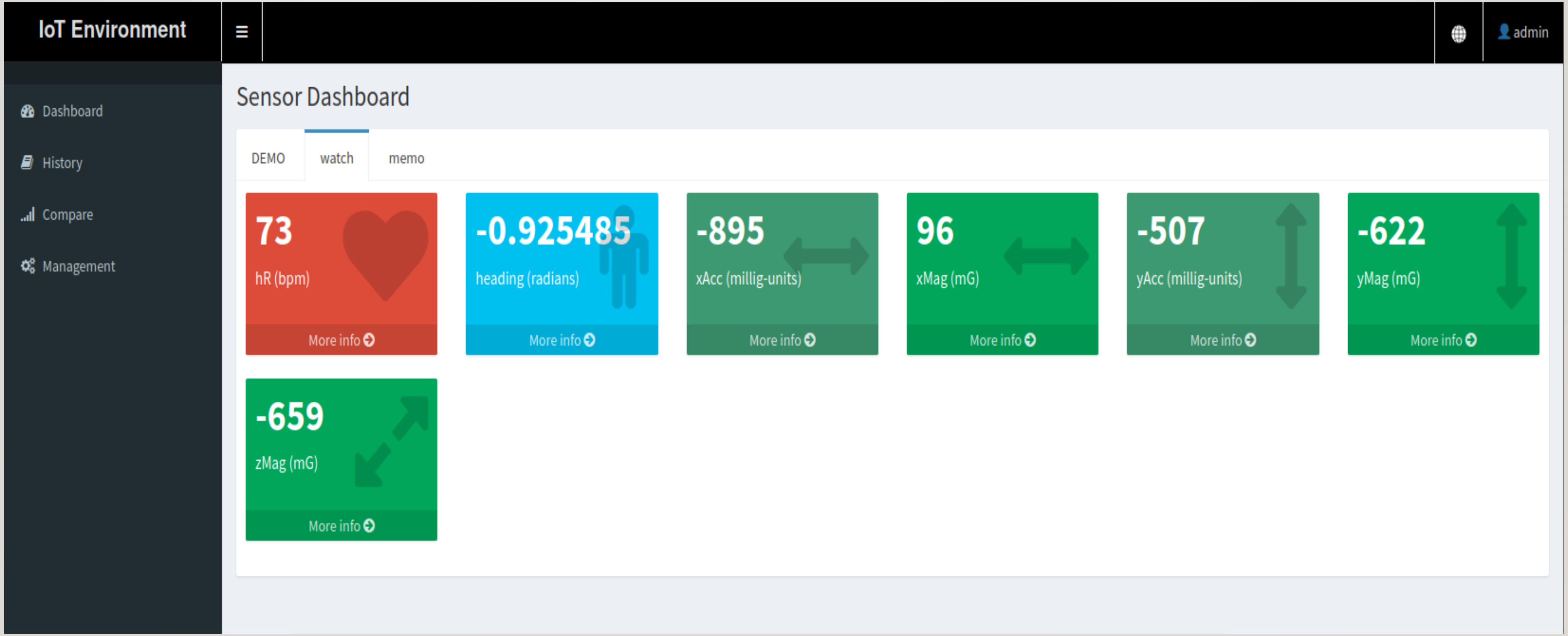
Push API function

研究方法-CIQ APP

```
function onReceive2(responseCode, data) { Receive function
    // System.println("2 requestCallback " + responseCode + ", data = " + data);
    receive = receive + 1;
    // System.println(receive);
    if(receive == 4) {
        receive = 0;
        sensor1();
    }
}
```

避免 BLE_QUEUE_FULL:
CIQ 的 web requset 等待佇列只能放四個請求

研究方法-DASHBOARD&DUMMY_DEVICE



研究方法-DATA TRAINING

```
allFiles = glob.glob("../watch"+"*.csv")
frame = pd.DataFrame()
list_ = []
for file_ in allFiles:
    list_.append(file_)
list_ = sorted(list_)

#有幾天的資料
csv_num = int(len(list_)/7)

for i in range(csv_num):
    if i==0:
        hR = pd.read_csv(list_[i])
        heading = pd.read_csv(list_[i+csv_num])
        xAcc = pd.read_csv(list_[i+2*csv_num])
        xMag = pd.read_csv(list_[i+3*csv_num])
        yAcc = pd.read_csv(list_[i+4*csv_num])
        yMag = pd.read_csv(list_[i+5*csv_num])
        zMag = pd.read_csv(list_[i+6*csv_num])

    else:
        hR = pd.concat([hR,pd.read_csv(list_[i])], join='inner').reset_index(drop=True)
        heading = pd.concat([heading, pd.read_csv(list_[i+csv_num])], join='inner').reset_index(drop=True)
        xAcc = pd.concat([xAcc,pd.read_csv(list_[i+2*csv_num])], join='inner').reset_index(drop=True)
        xMag = pd.concat([xMag,pd.read_csv(list_[i+3*csv_num])], join='inner').reset_index(drop=True)
        yAcc = pd.concat([yAcc,pd.read_csv(list_[i+4*csv_num])], join='inner').reset_index(drop=True)
        yMag = pd.concat([yMag,pd.read_csv(list_[i+5*csv_num])], join='inner').reset_index(drop=True)
        zMag = pd.concat([zMag,pd.read_csv(list_[i+6*csv_num])], join='inner').reset_index(drop=True)
```

研究方法-DATA TRAINING

```python
excepts=['other','standing','riding','bathroom']
for index in range(min_length):
    data_count = data_count + 1
#     print(index)
#     print(hR['datetime'][int(index)][2:])
    dt = datetime.strptime(hR['datetime'][int(index)][2:], '%y-%m-%d %H:%M:%S')


    val.append(hR['value'][index])
    for s in sensors[1:]:
        val.append(s['value'][index])

    if(dt > dt_act_next and act_index < len(raw_act)-2):
        dt_act = dt_act_next
        act_index = act_index + 1
        dt_act_next = datetime.strptime(raw_act['datetime'][act_index+1][2:], '%y/%m/%d %H:%M')

    if(raw_act['activity'][act_index] != cur_act):
        cur_act = raw_act['activity'][act_index]
        val = []
        data_count = 0

    if(data_count == data_len):
        #not in val
        if(cur_act not in excepts ):
            acts.append([cur_act])
            values.append(val)
```

研究方法-DATA TRAINING

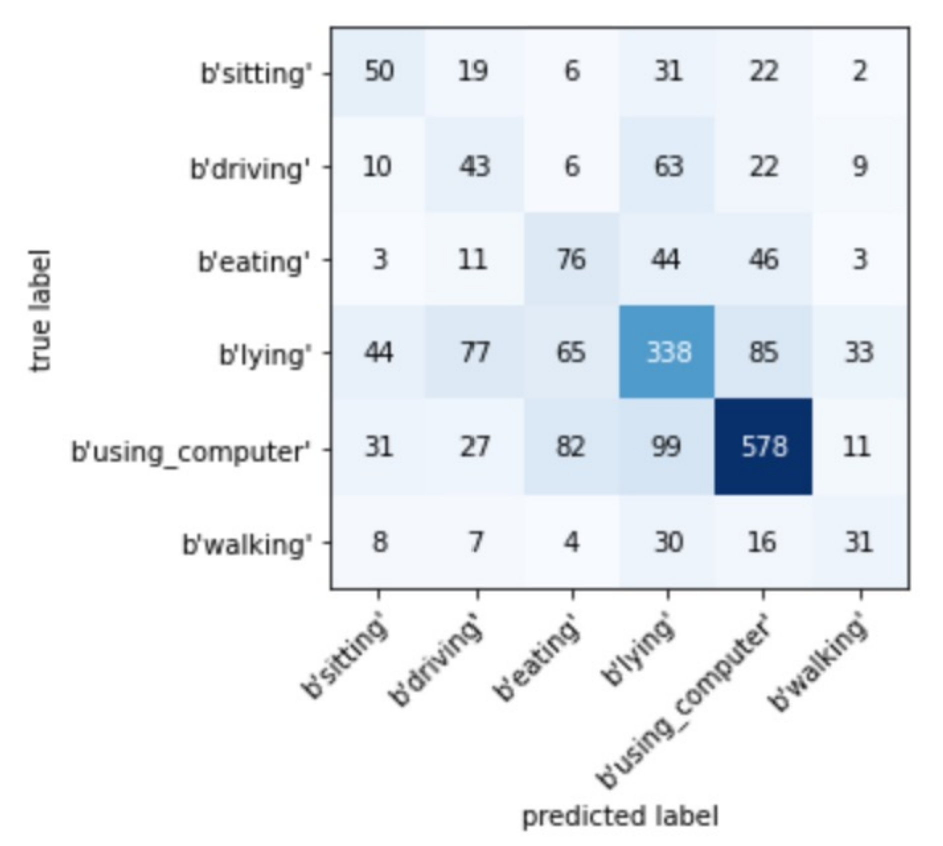
```
dt = DecisionTreeClassifier()

fin_pred = cross_val_predict(dt, x, y, cv=5)
print("Desicion Tree")
print(classification_report(y, fin_pred))

cm = confusion_matrix(y_target=y,
                      y_predicted=fin_pred,
                      binary=False)
fig,ax=plot_confusion_matrix(conf_mat=cm
                      ,show_absolute=True
                      ,show_normed=False
                      ,colorbar=False
                      ,class_names=feature_names
                      #,cmap='PuBu'
                      ,fontcolor_threshold=0.6
                      )
plt.show()
```

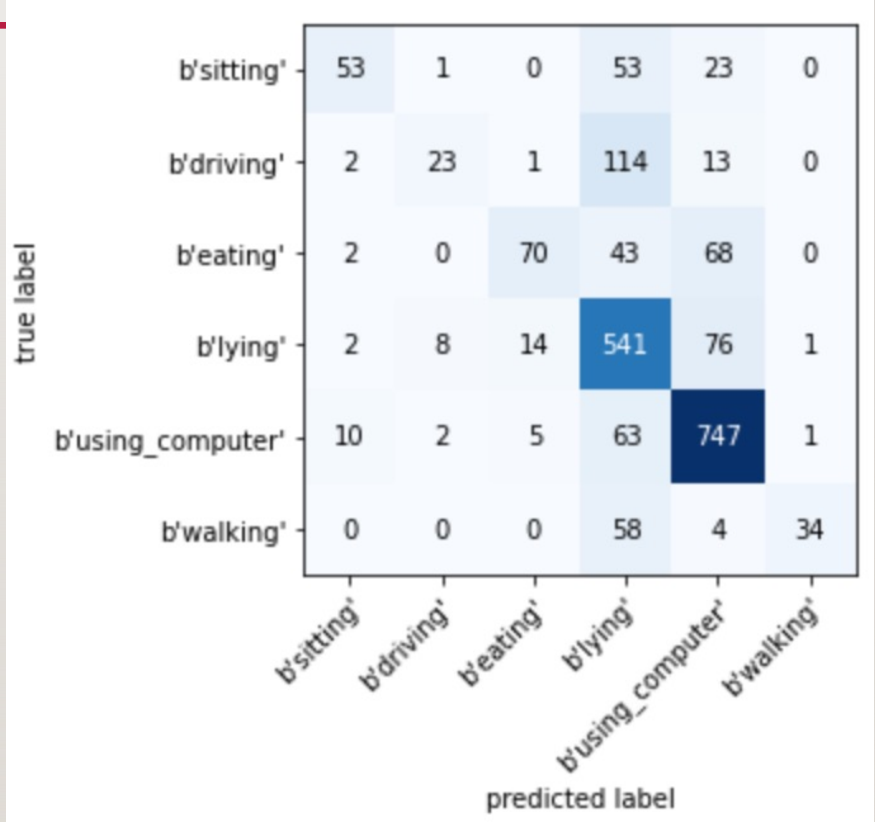
研究成果-DECISION TREE

	precision	recall	f1-score	support
b'driving'	0.35	0.29	0.32	130
b'eating'	0.24	0.27	0.26	153
b'lying'	0.38	0.41	0.40	183
b'sitting'	0.56	0.59	0.57	642
b'using_computer'	0.75	0.72	0.73	828
b'walking'	0.37	0.32	0.34	96
accuracy			0.57	2032
macro avg	0.44	0.43	0.44	2032
weighted avg	0.57	0.57	0.57	2032



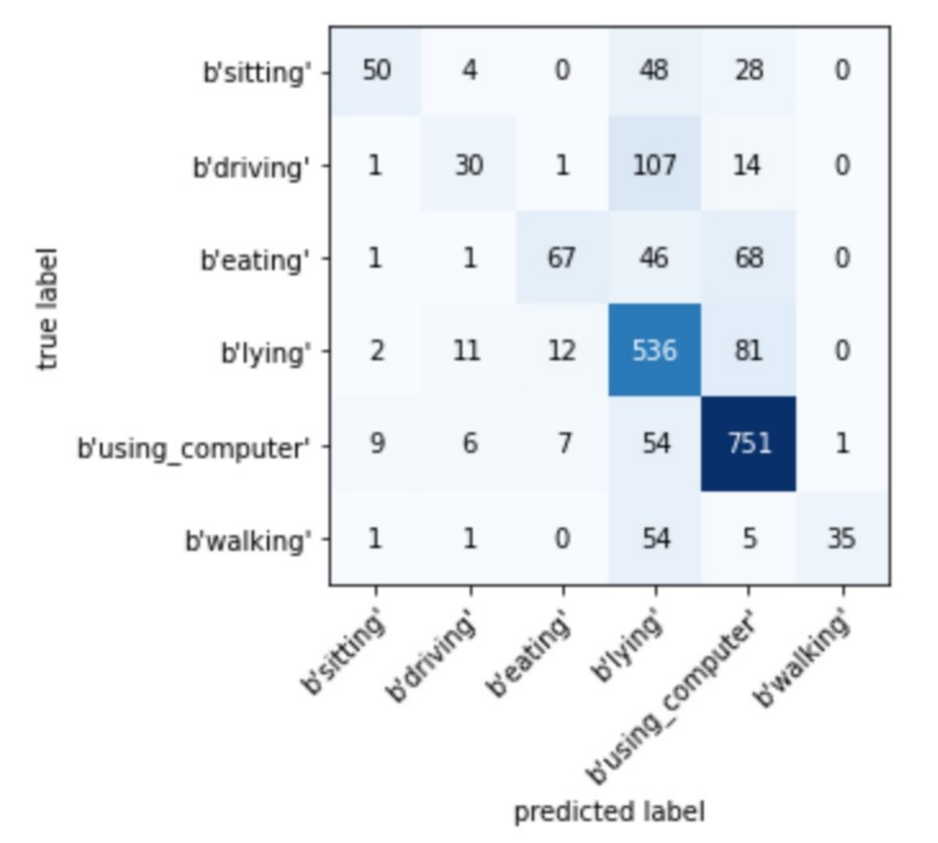
研究成果-RANDOM FOREST

	precision	recall	f1-score	support
b'driving'	0.80	0.39	0.53	130
b'eating'	0.63	0.14	0.23	153
b'lying'	0.80	0.40	0.54	183
b'sitting'	0.61	0.83	0.70	642
b'using_computer'	0.80	0.90	0.85	828
b'walking'	0.97	0.41	0.57	96
accuracy			0.72	2032
macro avg	0.77	0.51	0.57	2032
weighted avg	0.74	0.72	0.69	2032



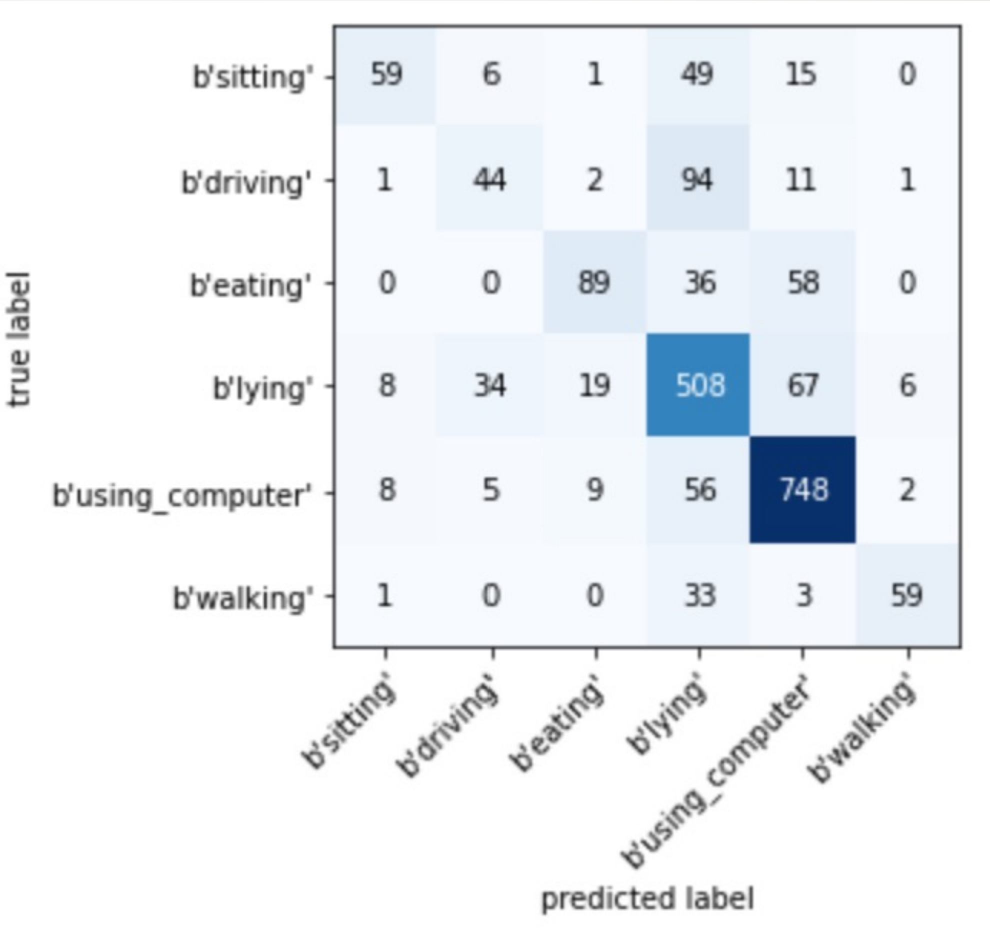
研究成果-ADABOOST

	precision	recall	f1-score	support
b'driving'	0.78	0.36	0.49	130
b'eating'	0.59	0.15	0.24	153
b'lying'	0.84	0.39	0.54	183
b'sitting'	0.61	0.82	0.70	642
b'using_computer'	0.79	0.91	0.85	828
b'walking'	0.95	0.36	0.53	96
accuracy			0.72	2032
macro avg	0.76	0.50	0.56	2032
weighted avg	0.73	0.72	0.69	2032



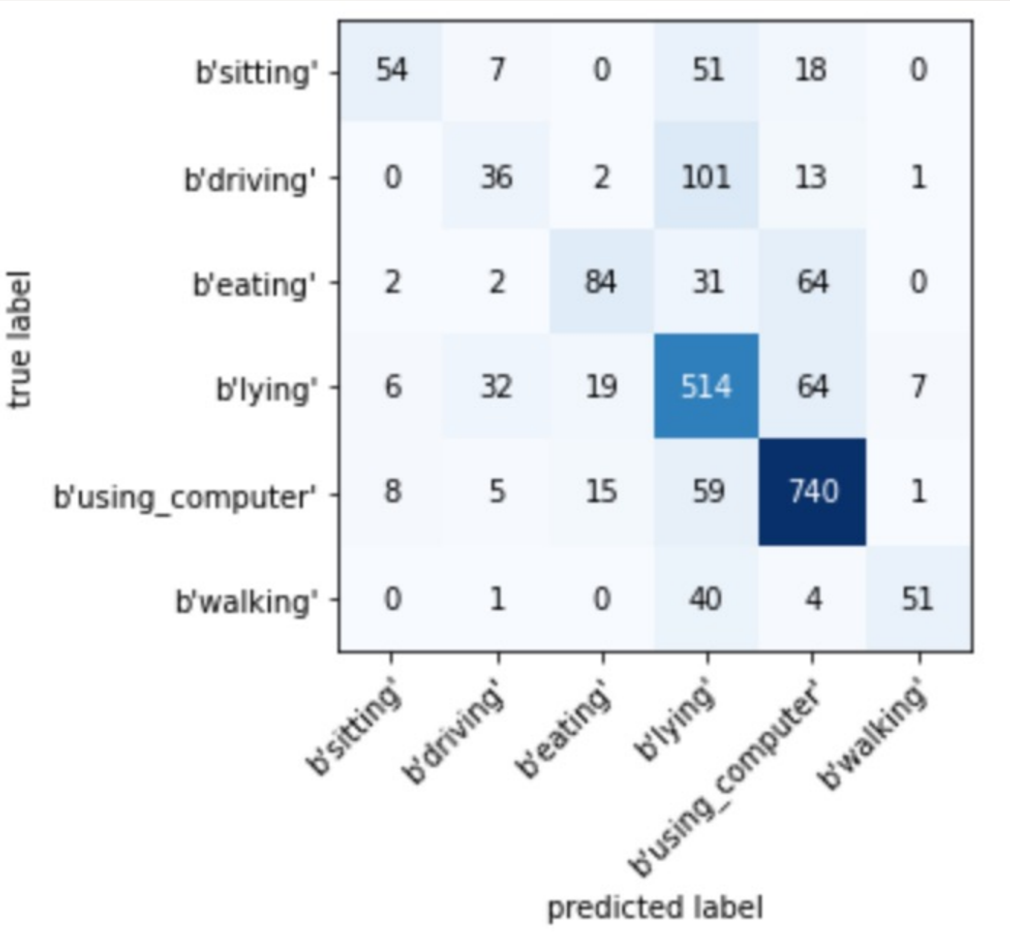
研究成果-HISTOGRAM-BASED GRADIENTBOOST

	precision	recall	f1-score	support
b'driving'	0.77	0.45	0.57	130
b'eating'	0.49	0.29	0.36	153
b'lying'	0.74	0.49	0.59	183
b'sitting'	0.65	0.79	0.72	642
b'using_computer'	0.83	0.90	0.86	828
b'walking'	0.87	0.61	0.72	96
accuracy			0.74	2032
macro avg	0.73	0.59	0.64	2032
weighted avg	0.74	0.74	0.73	2032



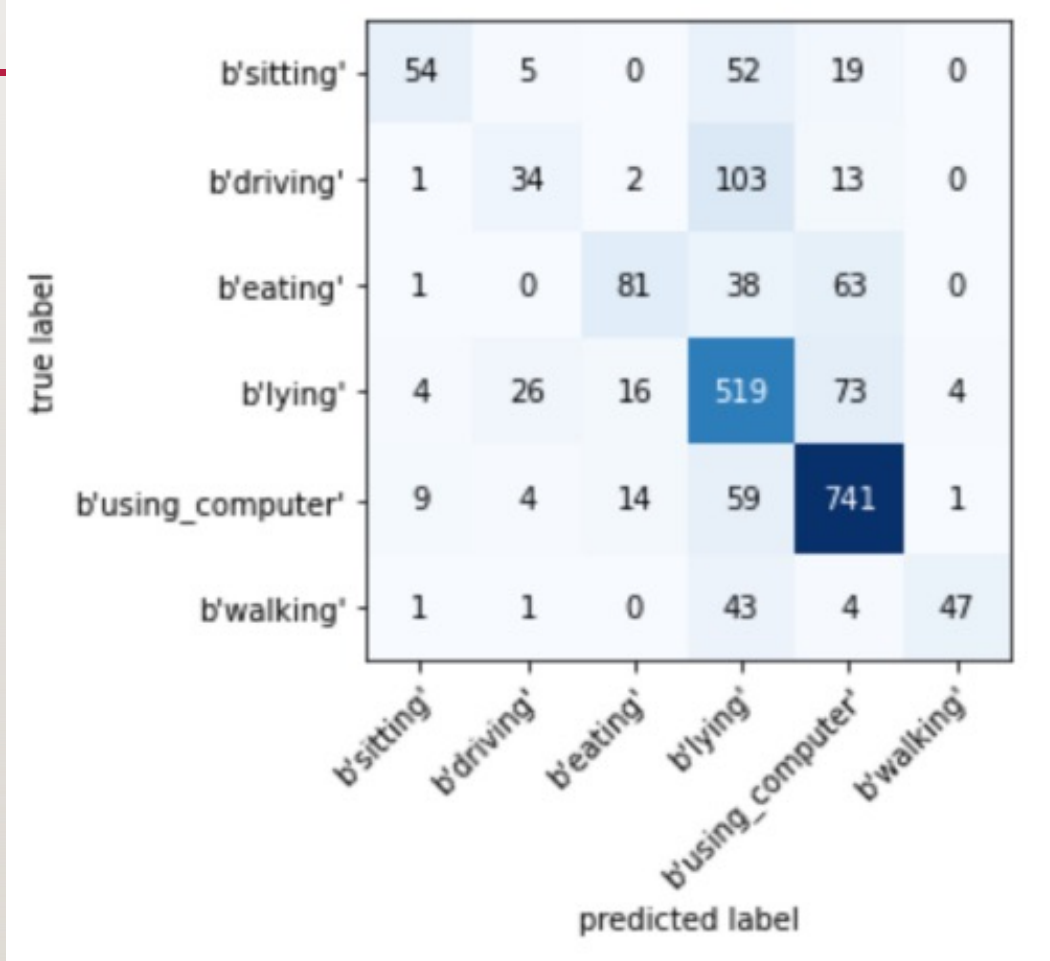
研究成果-XGBOOST

	precision	recall	f1-score	support
b'driving'	0.77	0.42	0.55	130
b'eating'	0.49	0.22	0.31	153
b'lying'	0.75	0.48	0.58	183
b'sitting'	0.63	0.79	0.70	642
b'using_computer'	0.82	0.89	0.85	828
b'walking'	0.82	0.52	0.64	96
accuracy			0.73	2032
macro avg	0.71	0.56	0.61	2032
weighted avg	0.73	0.73	0.71	2032



研究成果-ENSEMBLE

	precision	recall	f1-score	support
b'driving'	0.81	0.48	0.61	130
b'eating'	0.55	0.25	0.35	153
b'lying'	0.74	0.43	0.54	183
b'sitting'	0.65	0.81	0.72	642
b'using_computer'	0.81	0.90	0.85	828
b'walking'	0.91	0.51	0.65	96
accuracy			0.74	2032
macro avg	0.74	0.57	0.62	2032
weighted avg	0.74	0.74	0.72	2032



結論

- Model Training 的準確率大幅提升。從一開始的0.5，到現在除了Decision Tree以外都達到0.7以上，最高達到0.74。
- 因為CIQ request queue的限制，我們大約一至兩秒鐘才會有一筆資料。比起文獻中10秒200筆資料，sample rate太低。導致與現有研究相比，準確率仍然較低。
- 資料收集不受地域限制，只要能接上網路，就能更新資料。

參考文獻

- [1] Garmin developer API module :
- https://developer.garmin.com/connect-iq/api-docs/
- [2]S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based Human Activity Recognition Using Hybrid LSTM Network," 2020 IEEE SENSORS, 2020
- [3] S. Al-Janabi and A. H. Salman, "Sensitive integration of multilevel optimization model in human activity recognition for smartphone and smartwatch applications," in Big Data Mining and Analytics, vol. 4, no. 2
- [4] ATTAL, Ferhat, et al. Physical human activity recognition using wearable sensors. Sensors, 2015
- [5] MEKRUKSAVANICH, Sakorn, et al. Enhanced hand-oriented activity recognition based on smartwatch sensor data using lstms. Symmetry, 2020
- [6] S. Mekruksavanich and A. Jitpattanakul, "Sensor-based Complex Human Activity Recognition from Smartwatch Data using Hybrid Deep Learning Network," 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2021
- [7] S. Mekruksavanich and A. Jitpattanakul, "A Multichannel CNN-LSTM Network for Daily Activity Recognition using Smartwatch Sensor Data," 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, 2021
- [8] Luwe, Yee Jia, Chin Poo Lee, and Kian Ming Lim. "Wearable Sensor-Based Human Activity Recognition with Hybrid Deep Learning Model." Informatics. Vol. 9. No. 3. Multidisciplinary Digital Publishing Institute, 2022.

謝謝觀賞