

# 2021 年臺灣國際科學展覽會 優勝作品專輯

作品編號 190019  
參展科別 電腦科學與資訊工程  
作品名稱 基於觸控軌跡及裝置加速度資料提升年長者之  
觸控準確率

就讀學校 臺北市立第一女子高級中學

指導教師 陳彥仰、黃芳蘭

作者姓名 吳岱凌、范禎宸

關鍵詞 觸控、人機互動

## 作者簡介



我是北一女中的吳岱凌，加入數資班後，非常高興遇到了資訊專研的好夥伴們，也在專題研究中有幸獲得陳彥仰教授與黃芳蘭老師的悉心指導，得以在資訊這條路上成長許多。研究過程中不可避免的是無數次的障礙及修正，更遑論和專研夥伴熬過的每一夜，但是它們都無疑留下了深刻的痕跡。

我是北一女中三年級的范禎宸，我一直都很喜歡資訊領域、喜歡玩電腦和寫程式。很幸運在高中時找到陳彥仰教授指導我們的專題，我和岱凌也能夠針對人機互動的領域做研究。謝謝陳彥仰教授和黃芳蘭老師的指導，也謝謝實驗夥伴岱凌和我一起同甘共苦玩實驗、分析、和做科展！

## 摘要

本研究使用機器學習方法，改善年長者使用手機時觸控系統對於點按位置判斷之能力。首先設計實驗比較年長者使用手機時，點按位置及手勢判斷的準確率，接著收集年長使用者的觸控軌跡及裝置相關資料，並訓練模型以減少系統判斷的錯誤率和誤差幅度。再比較及分析不同機器學習模型對於本研究之資料的適用程度及經校準後點按位置準確率的提升，進而挑選出一個能夠最有效提升點按位置準確率的模型進行點按位置的預測。實驗過後選擇最有效提升準確率的 Random Forest Regressor 進行其他的校正實驗及分析。使用者點按位置的預測準確率能被有效提升，準確率能提高 32.3%。而最終，將訓練後的模型套回實驗用的手機程式，系統判斷受測者的點按位置能從原本的 63.7% 提升至 97.5%。

## Abstract

To improve the ability of touch screen systems to recognize the locations of taps correctly, we adopt different machine learning models to our data. We first design an experiment to compare the accuracy of the system's recognition on tap location and hand gestures when elderly people interact with touch screens. We then collect multiple tap data and touching tracks from random elderly people, and train different models to reduce the error rate and error distance of the system and compare the accuracy of each model to determine if they suit the features of our experiment, reaching a model which improves the accuracy of the tap data effectively and efficiently, that is, Random Forest Regressor. We use Random Forest Regressor to conduct our following experiments. Result shows a significant increase in tap accuracy, which is up to 32.3%. Lastly, we place the trained model into smartphones and reach an accuracy of 97.5%.

# 壹、前言

## 一、研究動機

隨著智慧型手機的普及，年長者開始接觸智慧型手機。相較於智障型手機的幾個簡單按鍵，智慧型手機有無數個點按位置可供點擊。然而並非所有年長者都能點擊到想點擊的位置，可能因為較遲緩的動作、手指不靈活，或眼睛看不清楚等原因而導致點歪或使系統誤判使用者的點按位置及手勢，進而造成使用者的不方便。因此，我們想要藉由機器學習的方法，提出一個更為精確的系統來校正干擾因素所造成的手機辨識錯誤和影響，藉此提升系統判斷點按位置及手勢的準確率，協助年長者更方便而準確的使用智慧型手機。

## 二、研究目的

- (一) 收集年長者使用觸控螢幕時系統判斷手勢及點按位置準確度。
- (二) 分析受測者手勢誤判及點按位置不準確的錯誤率。
- (三) 利用機器學習方法，使用收集到的點按資料訓練各種模型，針對手勢誤判或點按位置不準確進行校正。
- (四) 探討各種模型對於提升系統點按位置準確度的程度。

# 貳、研究方法

## 一、研究設備及器材

### (一) 硬體

1. MacBook Pro (處理器 2.3 GHz Intel Core i5)。
2. iPhone 6 Plus (iOS12)。
3. iPhone 8 Plus (iOS13)。
4. 筆電(處理器：Intel(R) Core(TM) i5-8250U CPU @1.60GHz  
1.80GHz、Windows)。

## (二) 軟體

1. Xcode - 用於開發收集資料的 App 的集成開發環境。
2. Swift - 用於編寫收集資料的 App 的程式語言。
3. Python - 用於編寫模型訓練及資料圖像化的程式語言。
4. Google Colaboratory - Google 的免費 Jupyter 筆記本環境。

## 二、相關文獻探討

找尋在此研究之前，針對使觸控螢幕變的更便利的論文進行閱讀及比較，主要分成點按位置優化與手勢優化。

### (一) 手勢優化

表 1：手勢優化探討

研究	實驗過程	實驗結果
為行動不使用者開發多模式自適應算法 <sup>[6]</sup>	分析手的握力影響人的觸控能力。	開發出一組線性方程式減少了點擊花費的時間。
通過共享用戶模型進行的可用觸控螢幕交流 <sup>[5]</sup>	針對 12 名患有多種視覺和運動障礙的成年人，改善介面的陳列方式以幫助他們更好使用觸控螢幕。	將螢幕分成寬 1:2:1 高 1:3:1 後，每個區域平均可以降低 10% 錯誤率。

### (二) 點按位置優化

在點按位置優化的這個項目中，許多研究會就情境障礙（如低溫、酒精等）進行該研究與改良。在此就最普遍的狀態（靜坐與行進）作探討。

表 2：點擊位置優化探討

研究	實驗過程	模型校正成效
Cluster Touch <sup>[1]</sup>	尋找 12 位測試者測試。透過多個用戶的資料、統計每個區塊點擊位置與目標位置的距離，製作一個模型範例，再透過用戶個人的測試進行調整，使其具有個人化。	無運動情境的使用者觸控準確率提高 14.65%，行進中的使用者提高 6.81%。
WalkType <sup>[2]</sup>	理論：走路時行走震動和注意力分散會加劇正確按鍵的困難 實驗過程：尋找 16 位測試者測試，利用加速度計的資料、紀錄點擊位置與在點擊期間手指移動距離，最後做成了一個複合模型，使用內置三軸加速度計以改善用戶的文本輸入準確性。	將用戶的打字速度從 28.3WPM 提升至 31.3WPM，將錯誤率從 10.5 降低至 5.8。

在這些文獻中，我們發現大部分人都忽視年紀對於觸控螢幕的互動造成的影響，因此我們就決定針對老年人和觸控螢幕互動的困難來進行校正。我們也將螢幕切割成更小的區塊，以期有更好的預測準確率。

### 三、研究架構

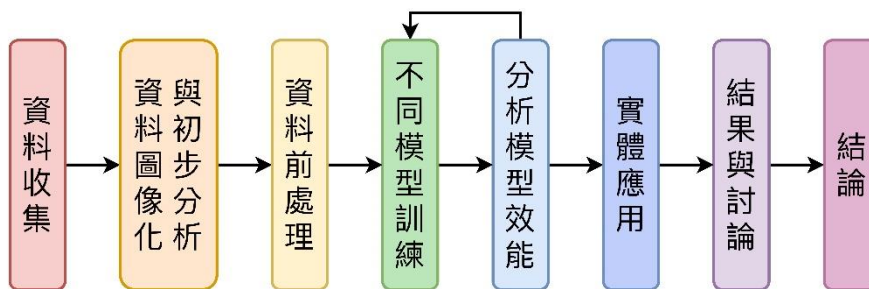


圖 1-1：研究架構圖 1

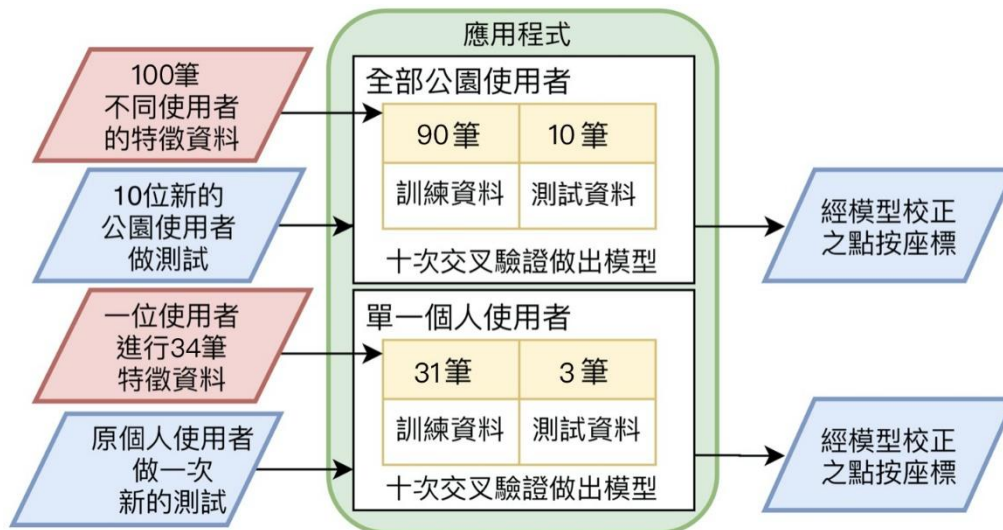


圖 1-2：研究架構圖 2

#### 四、研究流程

##### (一) 收集資料

##### 1. 備妥收集資料之工具

我們用程式語言 Swift，修改一個手機應用程式來收集人們使用手機時手勢及習慣的資料。這個應用程式的內容包含受試者基本資料的填寫處（使用者編號及出生年份）、問卷（包含健康狀況、行動裝置使用頻率及用途）以及我們的各個測試項目。

##### 2. 問卷題目

為了更加了解受測者的個人背景，我們向受測者收集了以下這些個人資料：




表 3：問卷項目表

出生年	生理性別	慣用手
運動障礙	症狀	是否難以專心
是否焦慮緊張	是否難以看清楚	聽力是否有困難
溝通理解是否有困難	提重物是否有困難	撿小物件是否有困難



走路爬樓梯是否困難	疲憊頻率/時間/程度	醫療情況
服用藥物情況	是否有行動裝置	有哪些行動裝置
有哪些智慧型手機	有哪些平板電腦	最常使用的行動裝置
使用裝置用圖	使用裝置頻率	裝置使用的困難程度
用過何裝置輔助工具	裝置互動方式	

### 3. 測試項目

表 4：App 測試項目表

	<p>名稱：<b>點按能力測驗</b></p> <p>項目圖案：50*50 pt 的靶心（iPhone 圖示大小）</p> <p>出現方式：將手機螢幕分成 7 行 5 列，隨機在其中一個位置出現，每次出現位置不重複</p> <p>測驗方式：點按各靶心，共 35 個測驗點</p>
	<p>名稱：<b>長按能力測驗</b></p> <p>項目圖案：50*50 pt 的靶心</p> <p>出現方式：將手機螢幕分成 7 行 5 列，隨機在其中一個位置出現，每次出現位置不重複</p> <p>測驗方式：點按各靶心一秒，共 35 個測驗點</p>
	<p>名稱：<b>掃動能力測驗</b></p> <p>項目圖案：綠色箭頭</p> <p>出現方式：上下左右的箭頭隨機出現</p> <p>測驗方式：在螢幕上按照箭頭方向進行掃動，共 8 個測試項目</p>



	<p>名稱：<b>垂直滾動能力測驗</b></p> <p>項目圖案：一個綠色十字、一個黑色十字</p> <p>出現方式：兩個十字呈垂直排列，距離為隨機</p> <p>測驗方式：從綠色十字滾動到黑色十字，共 8 個測試項目</p>
	<p>名稱：<b>水平滾動能力測驗</b></p> <p>項目圖案：一個綠色十字、一個黑色十字</p> <p>出現方式：兩個十字呈水平排列，距離為隨機</p> <p>測驗方式：從綠色十字滾動到黑色十字，共 8 個測試項目</p>

#### 4. 找尋受試者

由於我們的受試者為年長者，因此我們將受試對象訂為 65 歲以上之長者。我們先請自己的祖父母協助我們受試，再請同學的祖父母協助受試。接著到公園請在公園內休息的長輩協助受試，並給予書籤及零食作為回饋。公園為年長者常聚集之地，能收集到較多且個別差異較大的年長者受試資料。最後收集到 100 位老年人的測試資料。相較其他的論文使用者的數量（文獻探討中有提到），使用者數量明顯大於平均值。1 位受測者測試時間約 10~15 分鐘（包含問卷調查及測試項目）。受測者的姿勢被規定為坐著，並以一隻手拿手機另一隻手觸控螢幕。

根據問卷資料，我們的受測者男女比例約相等，而年齡大多為 65~75 歲，右撇子佔大多數；較少人有特殊運動障礙，少數人有醫療情形及服用藥物。

## (二) 資料的圖像化與分析

將所收集到的資料以 json 檔格式儲存後，依年齡與狀態整理過後，用 Python 撰寫程式繪製出各種的比較圖。在收集到 20 筆資料後我們進行第一階段的分析。

### 1. 手勢判斷準確率分析

討論在點按測驗及滑動測試（包含掃動及滾動）時，手機所判斷的手勢是否與目標測驗項目相符。圖 2 中上方的圓為點按測驗時的手勢判斷圓餅圖，其中紅色色塊代表手機將受測者的手勢判斷為點按，綠色色塊則是判斷為「點按+滑動」；而下方的圓為滑動測試的手勢判斷圓餅圖，其中橘色代表手機判斷為滑動。

根據圖 2 及表 5，我們可以看到在點按測試和非點按測試中手機的判斷都會有一些零星的錯誤。手機主要會將點按或滑動判斷成「點按+滑動」，但錯誤率都不到一成，並沒有特別顯著。

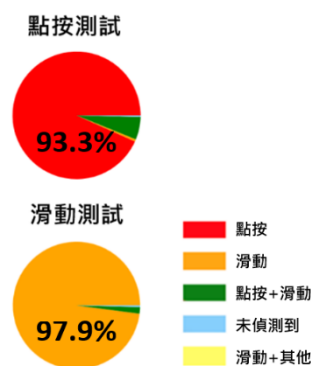


圖 2：手勢判斷圓餅圖

表 5：手勢判斷準確率表

	正確率
點按測驗	93.3%
滑動測驗 (掃動、滾動)	97.9%

### 2. 個人手勢判斷準確率分析

上述圖 2 為所有受測者的資料一起做分析比較，而下方圖 3 為每一個受測者的手勢判斷準確率圓餅圖(一個圓代表一個受測者)。

可以看見只有少數人的滑動會被判斷為滑動+其他手勢，錯誤率並不高（每個人最多錯八次測驗中的兩次）。且經過比對受測者資料後發現，錯誤率較高的受測者擁有較少的智慧型裝置使用經驗，應為經驗不足而不熟悉手機操作方式而致的錯誤。

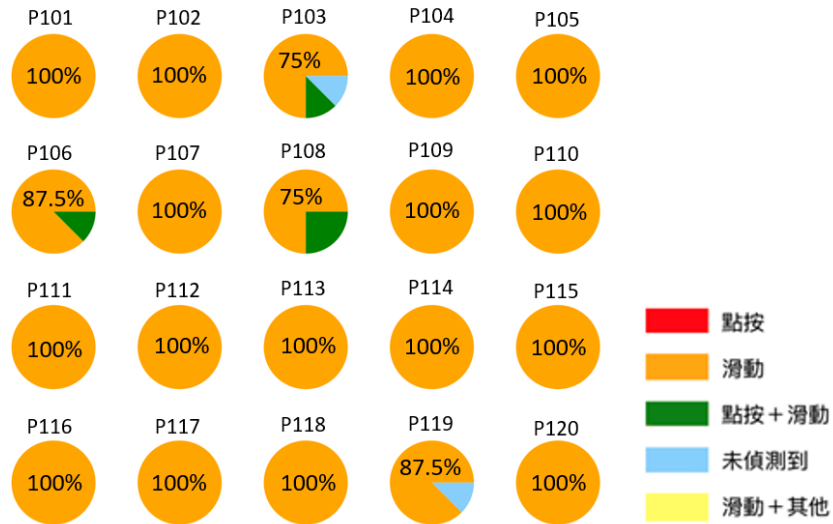


圖 3：個人手勢判斷圓餅圖

### 3. 個人單項測驗點按位置偏差圖

圖 4 為在點按測試中，每個人在各狀態下的點按位置偏差圖。圖中包含使用者編號及測驗時的狀態。每一個不同顏色（不同顏色沒有特殊意義）的圖形包含一個點和以它為起點的線段。該點為一個目標位置，而該線段為目標位置及使用者實際點按位置的偏差，包含距離及方向，亦即線段越長偏差越多。

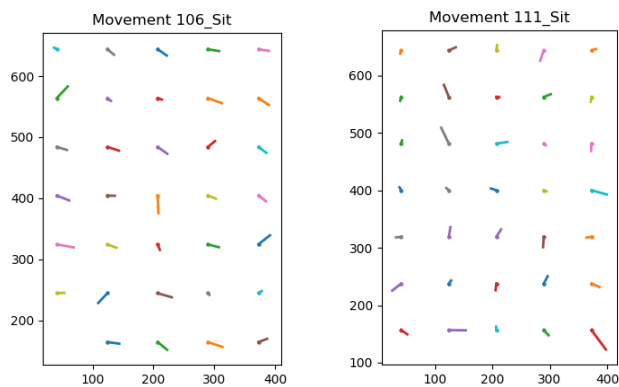


圖 4：個人點按位置偏差示意圖

從圖中可見，由於我們將點（目標位置）設定在 50pt\*50pt 靶心測試點的正中央，而使用者常會點按在靶心圖案上，因此多少會出現一點小條直線。但仍會有一些區域有較大的偏差，表示該測試點沒有做到精確的點按。通常同一區域的偏差會偏向同一方向，如圖 5 的左上角及右上角，他們都往中心偏差，可能因為距離中心較遠而較難點按準確。偶爾會出現空格如圖 6，此為系統未偵測到點按資料，在後續的實驗我們將刪除這些明顯的人為錯誤不納入實驗比較。

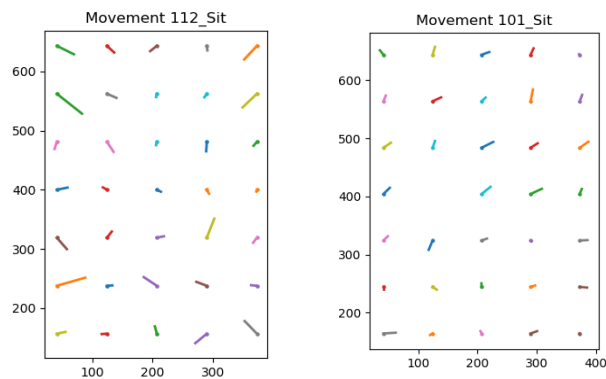


圖 5：個人點按位置

圖 6：個人點按位置

#### 4. 點按位置偏差熱力圖

圖 7 為在點按測試中，在各狀態下每個人的點按位置平均偏差圖。圖中每一個方框代表一個測試點，顏色對應到右方標示，為 pt 偏差量，顏色越深代表偏差越多。

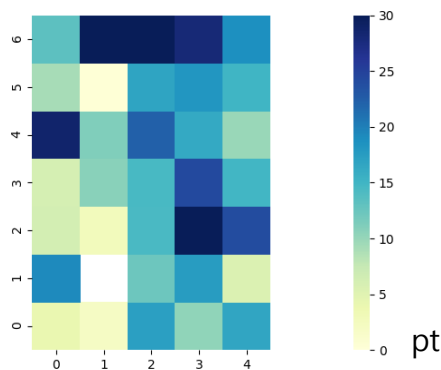


圖 7：點按位置偏差熱力圖

觀察上圖，可以看見圖的上方及右邊及左上右上右下角部分較深，代表點按位置與目標位置有較大的偏差。搭配個人點按位置偏

差圖可以得知這些角落常被點按到較接近中心的位置，推測因為手機螢幕較大，因為手指必須移動較大的距離，才造成越邊角的地方越難以準確點按。

## 5. 準確度比較

我們將每個點按位置的目標座標設為每個目標方格的中心點，分別向上下左右展開 50/2 pt 形成一個 50\*50 pt 的正方形區域。若受測者之點按座標位於這個區域中，即判斷點按成功。

在未將資料作處理前的各狀態準確率如下表，其中 RMSE (Root Mean Square Error) 為方均根誤差，就是將所有的點按的誤差開平方後取算數平均數再開根號。

表 6：原點擊 RMSE 及準確率

RMSE	原點按準確率
204.3	64.0%

### (三) 選擇校正項目

根據我們實驗過後的分析，只有一些使用者的測驗中有出現手勢判斷錯誤的情況，且那些使用者通常為較少使用智慧型手機等電子產品的人；又整體而言錯誤率並不顯著，因此我們選擇準確率較低的點按位置校準來進行我們的校正實驗。

### (四) 資料前處理

為了將收集到的資料轉化成有利於機器學習的特徵，進行資料前處理。參考相關研究的特徵選取並加入自己的想法，因觀察得出使用者在點按時的變數有進行點按的手的點按位置、點擊時間、拿著手機的手的晃動，我們針對了每筆點按資料共提取了 33 個特徵項目。

表 7：特徵項目表

特徵項目	個數
點按起點 X、Y 座標（手指一觸碰到螢幕的座標）	2
點按終點 X、Y 座標（手指離開螢幕前的座標）	2
起終點距離	1
點按時間長	1
手機加速度資料 X、Y、Z 軸各取三筆	9
手機陀螺儀資料 X、Y、Z 軸各取三筆	9
加速度資料經 2 次 cumtrapz 積分得手機位移資料(X、Y、Z 軸)	9

註：cumtrapz 積分為通過梯形法按單位間距計算 Y 的近似累積積分。其本質是將積分區間畫分為一定數量的等間距的子區間後將每段子區間上的函數曲線用一段直線近似，該直線經過由小區間端點對應的原曲線上的兩個端點。附圖如下：

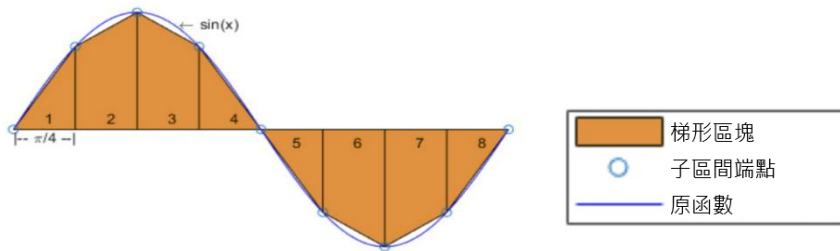


圖 8：cumtrapz 積分示意圖

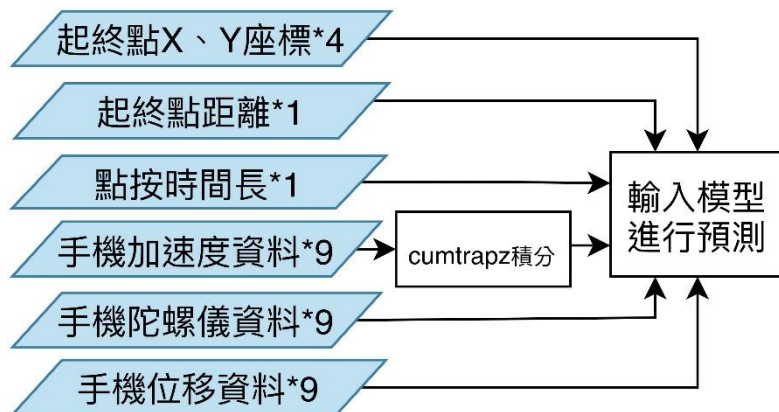


圖 9：特徵輸入示意圖

當測驗結束後，每筆測驗會先取得上述 33 筆資料。之後將這些資料自動上傳至 Amazon Web Services(AWS)雲端服務，放入雲端資料夾。利用 Google Colaboratory 連結雲端資料夾載入多筆測試資料。再將資料共 33 筆在 Google Colaboratory 中輸入模型進行訓練。最後將測試資料套入訓練過後的模型進行預測。

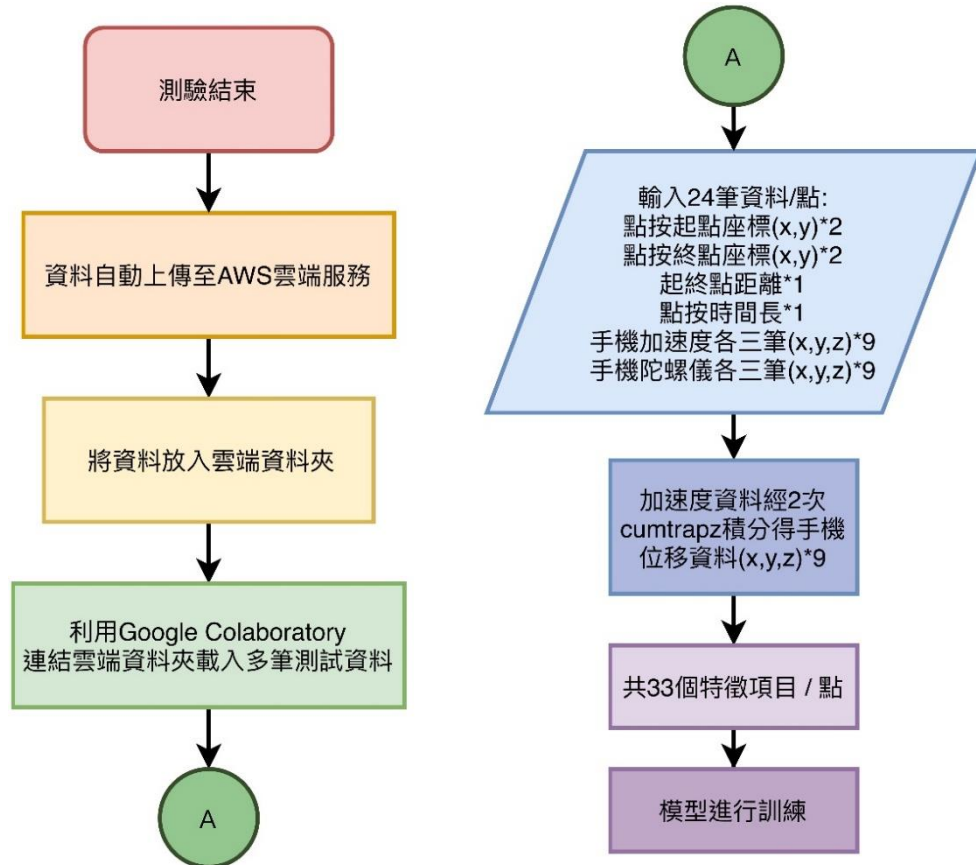


圖 10：資料前處理流程圖

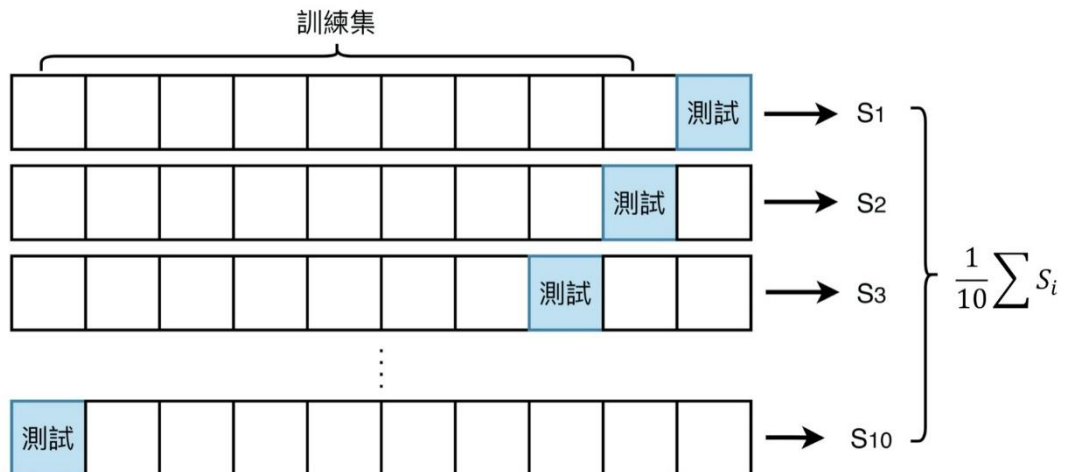


圖 11：交叉驗證示意圖

## (五) 套入模型並分析效能

根據 sklearn 官方提出的模型選擇方法（附錄一），我們不是做種類預測而是做位置座標（數字）的預測，因此我們選擇了 Linear Regression、Random Forest Regressor、Ridge Regression、Lasso、Elastic Net 和 RANSAC 六種模型進行實驗。我們將十分之九的資料拿去訓練模型，其餘十分之一的資料拿來測試。

### 1. 線性回歸（Linear Regression）

#### (1) 原理簡介

線性迴歸是在找多個自變數和依變數之間的關係建出來的模型。異於只有一個自變數的 Simple Linear Regression (SLM), 本研究中所使用的模型為適用於多變數的 Multiple Linear Regression (MLR)。

多元線性回歸方程式為：

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon$$

其中有  $k=1\sim p$  個自變數

$i=1\sim n$  (資料筆數)。

$y_i$  表示依變數

$x_{ik}$  表示自變數

$\beta_0$  表示迴歸線之  $y$  截距

$\beta_k$  表示個自變數之回歸係數、 $\epsilon$  表示誤差值

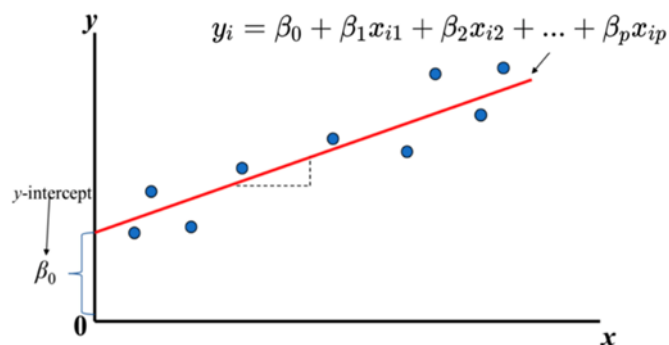


圖 12：回歸方程式示意圖



回歸線的分析就是由最小平方方法(Least Square)來找所有的回歸係數，而我們會很希望所有訓練樣本的誤差平方和(Sum Square error, SSE)接近 0。

(2) 符合本研究原因

- I. 此模型之輸出為數值，符合我們應有的輸出(X、Y 的座標位置)
- II. 建模速度快，不需要很複雜的計算，在資料量大的情況下依然執行速度很快。

2. 隨機森林 (Random Forest Regressor)

(1) 原理簡介

原始隨機森林演算法中分類器為 CART (Classification and Regression Tree) 樹，透過 Bagging 演算法進行組合學習，並在 CART 樹生長時隨機選取變數進行分裂。此方法基本概念為使用二元分割規則來歸納與分析大量複雜變數的資料集。CART 演算法在過程中將資料進行分類，分類過程與樹狀結構類似，含有根 (root), 點 (node) 與樹葉 (leaf)。當分枝點進行分裂時，就是一次白努力實驗 (Bernoulli experiment)，表示由重複出現獨立但是相同分布的白努力試驗組成，如拋硬幣數次，結果呈現二項式分布等。

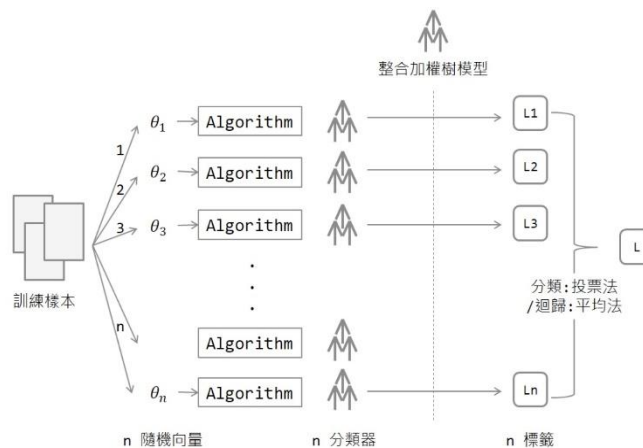


圖 13：Random Forest Bagging 示意圖

使用 Bagging 演算法將資料生成訓練資料：假設有  $N$  的樣本，每個樣本有  $M$  個變數(特徵)，以隨機抽取但會放回的方式取得數個樣本而組成訓練資料，共生成  $n$  資料集。對每個訓練資料集，生成不同的隨機向量  $\theta_i$ ，隨機選擇  $m$  個變數 (且  $m \leq M$ )，對其中每個變數都嘗試分割，以選擇達到最小的 Gini 係數的分割方進行分裂，生成 CART 樹。讓每顆樹生長，不進行剪枝。對這  $n$  顆樹的結果進行組合：若為分類資料(Classifier)，則用簡單多數投票法，若為迴歸法(Regressor)，則用平均法。本研究即用迴歸法。

(2) 符合本研究原因

- I. 隨機森林的基礎演算法是基於 CART 演算法，故可以處理類別資料與連續資料。
- II. 接受高維度特徵資料，可以處理大量的輸入變數。
- III. 處理非平衡誤差資料時，能平衡誤差；分類資料時亦能算出相似度。
- IV. 它計算各例中的親近度，對於數據挖掘、偵測偏離者(outlier)和將資料視覺化非常有用。
- V. 對大多資料而言隨機森林演算法的擬和結果準確率高。

3. Regularized Regression

(1) 原理簡介

在線性迴歸模型(Linear Regression)中，為了最佳化目標函式(最小化誤差平方和, SSE)，資料需符合許多假設，才能得到不偏迴歸係數，使得模型變異量最低。

$$\text{minimize} \left\{ SSE = \sum_{i=1}^n (y_i - \hat{y})^2 \right\}$$

註：  $i = 1 \sim n$  (資料筆數)， $y_i$  為每一筆  $x$  資料的對應資料， $\hat{y}$  則是所有  $y_i$  的平均數。

但現實中數據非可能有多個特徵變數，這會產生三個主要問題，分別是：共線性、解決方案不充分、可解釋性，使模型假設不成立而產生過度擬合問題，這時則需透過正規化法 (Regularization) 控制回歸係數，藉此降低模型變異以及樣本外誤差。

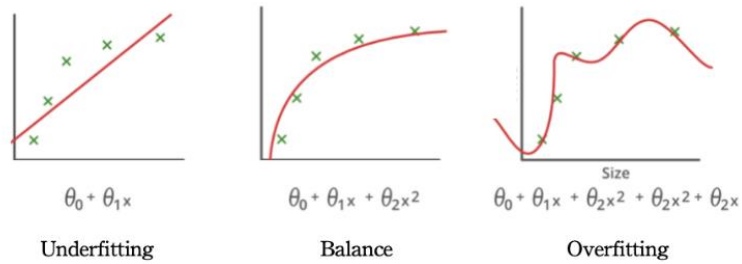


圖 14：回歸方程式擬合程度示意圖

由於過度擬合的迴歸方程式的迴歸係數會特別大，因此，Regularized Regression 將透過將懲罰參數(Penalty parameter, P) 加入目標函式中。

$$\text{minimize}\{SSE + P\}$$

以下三種 Regularized Regression 模型便是使用不同的懲罰參數來避免過度擬合的情況。

#### I. Ridge Regression

$$\text{minimize}\{SSE + \lambda \sum_{j=1}^p \beta_j^2\}$$

註：其中有  $p$  個自變數， $\beta$  為各自變數之係數

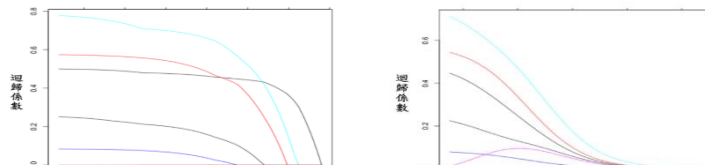
當  $\lambda \rightarrow \infty$  時，懲罰效果最大，迫使所有係數都趨近於 0。如下圖（數字）所示， $\log \lambda$  越大，迴歸係數越接近零。其中有  $p$  個自變數， $\beta$  為各自變數之係數。

## II. Lasso

$$\text{minimize}\{SSE + \lambda \sum_{j=1}^p |\beta_j|\}$$

註：其中有 $p$ 個自變數， $\beta$ 為各自變數之係數

不像 Ridge 模型只會將係數  $\lambda \sum_{j=1}^p |\beta_j|$  逼近到接近零（但不會真的是 0），Lasso 模型則真的會將係數推進成 0（如下圖）。因此，Lasso 模型不僅能使用正規化 (regularization) 來優化模型，亦可以自動執行變數篩選 (Feature selection)。



註：其中有 $p$ 個自變數， $\beta$ 為各自變數之係數

圖 15：Lasso(左)及 Ridge(右)之  $\log \lambda$  與迴歸係數關係圖

## III. Elastic Net

$$\text{minimize}\{SSE + \lambda_1 \sum_{j=1}^p \beta_j^2 + \lambda_2 \sum_{j=1}^p |\beta_j|\}$$

Elastic Net 涵蓋了上面兩個模型 (Ridge Regression & Lasso)。於是，Elastic Net 模型的優勢就在於，它綜合了 Ridge Penalty 達到有效正規化優勢以及 Lasso Penalty 能夠進行變數挑選優勢。

### (2) 符合本研究原因

#### I. Ridge

- i. 會將具有相關性的變數推向彼此，並避免使得其中一個有極大正係數另一個有極大負係數的情況。
- ii. 許多不相干的變數係數會被逼近為 0。

#### II. Lasso

- i. 與 Ridge 模型一樣，Lasso 模型亦會將具有相關性的變數推向彼此
- ii. Lasso 會將不具影響力的變數係數變成 0。

### III. Elastic Net

- i. 它綜合了 Ridge Penalty 的有效的正規化過程以及 Lasso Penalty 的變數篩選功能。
- ii. 能夠控制變數共線性的問題。

## 4. 隨機抽樣一致 (RANSAC)

### (1) 原理簡介

RANSAC 是透過觀察到的資料，隨機取數筆數據來估計資料模型的迭代方法，並以此模型將資料做分類。

使用 RANSAC 的步驟如下：

- I. 假定一模型(如一直線)，隨機選擇一些點做為內群，對模型進行擬合。

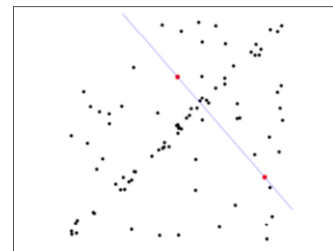


圖 16：RANSAC 步驟 1

- II. 設定容錯範圍，選定其他尚未選過的點與原模型比較，計算是否在容錯範圍內，若是，則視為一內群。

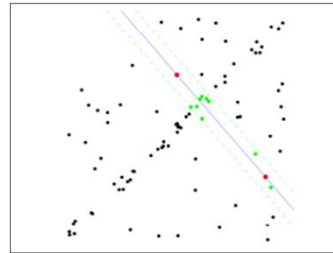


圖 17：RANSAC 步驟 2

- III. 重新隨機選取未被選過的點，重複第一步~第二步的操作，直到結束迭代。

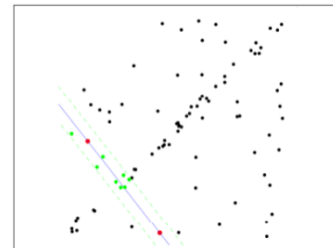


圖 18：RANSAC 步驟 3

- IV. 每一次擬合後，容差範圍內都有對應的數據點數，找出數據點個數最多的情況，就是最終的擬合結果。

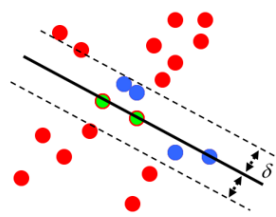
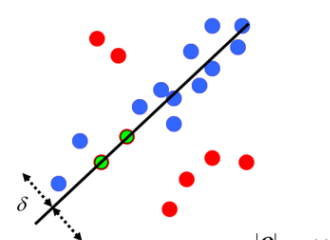


圖 19：RANSAC 步驟 4  $|O| = 6$



$|O| = 14$

- (2) 符合本研究原因
  - I. 能從包含大量局外點的數據集中估計出高精度的參數
  - II. 不用煩惱群集數量
  - III. 同時提供邊緣點的位置和方向信息

#### (六) 將訓練後之模型套入 App

經過 20 筆測試資料的分析後，我們選擇點按位置預測進行研究。我們持續收集到了 100 筆資料，並將這 100 筆資料放進上述 6 種模型進行訓練。分析比較資料後，我們發現 Random Forest Regressor 的成效最高。因此我們選擇將 Random Forest Regressor 訓練後的整體模型放入應用程式中測試（以下稱此 app 為點按 app）。

將模型套回點按 app 中有以下幾個步驟：

##### 1. 轉成 mlmodel 檔

由於訓練模型時是以 python 語言寫的，因此為了將它套入點按 app 中，得在訓練完模型後先把它轉成 mlmodel 檔，存至 Google 雲端中。其中因為 Random forest Regressor 的 mlmodel 檔無法做預測點按的 X、Y 軸座標兩項輸出，因此我們將原來的模型拆成兩個，分別預判 X、Y 軸座標。

##### 2. 將 mlmodel 檔放進和點按 app 的 workspace 同一資料夾中

為了方便在點按 app 運作的時候的呼叫和引用，將 mlmodel 檔的模型從雲端中下載下來放進和點按 app 的 workspace 同一個資料夾中。

##### 3. 在點按 app 中引用模型

在點按 app 中將 mlmodel 檔轉成 Swift 語言，並將之後使用到它的變數都定義好，方便呼叫及使用。

##### 4. 呼叫模型

在即將使用到模型的程式檔案中事先呼叫模型。

## 5. 輸入 33 個特徵項目

在測試中的使用者每點按一次，就對模型輸入每個點產生的 33 個特徵項目（詳細資訊在資料前處理的地方）。

## 6. 模型進行預測+判斷點按成功與否

將模型針對這 33 筆資料作出的預測（兩個座標數值）輸入手機讓它判斷是否超出點按目標的範圍，即判斷是否成功點按。隨著每一點的預測及判斷慢慢計算整次測驗的點按準確率。

## 7. 結束測試+顯示此測試準確度

在結束測試後，最後顯示此次測試的點按準確度。

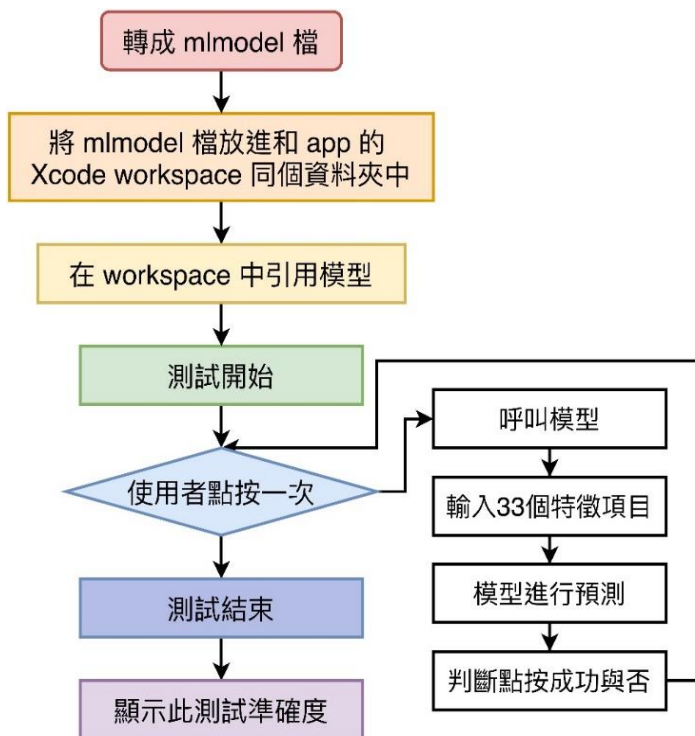


圖 20：將模型套入應用程式流程圖

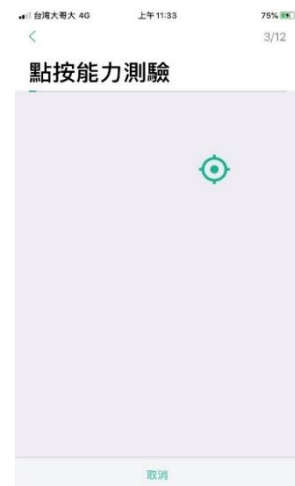


圖 21：點按 app

## 參、研究結果與討論

### 一、各種模型之效能比較

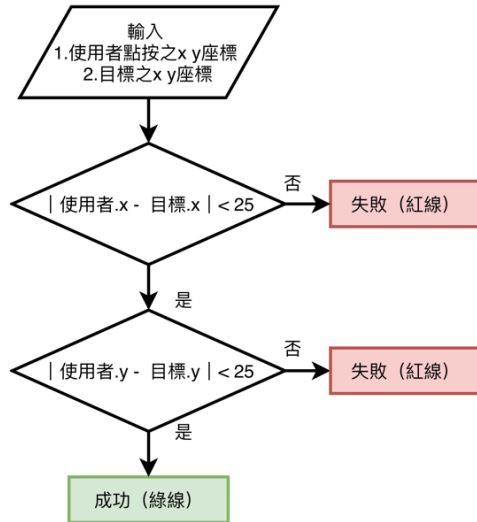


圖 22：判斷成功點按流程圖

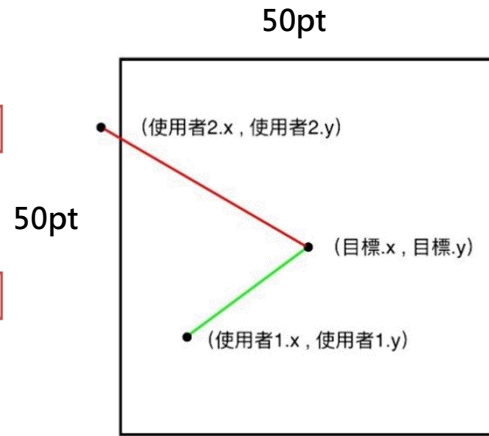


圖 23：判斷成功點按示意圖

前處理過後的資料分為訓練資料與測試資料，將訓練套入進各種模型進行訓練。再將測試資料放進模型中預測目標點按位置，繪圖、並以準確率比較之。我們將每一個點按位置的目標座標設為每一個目標方格的中心點，分別向上下左右展開 50/2 pt 形成一個 50\*50 pt 的正方形區域。若受測者之點按座標位於這個區域中，即判斷點按成功。

附錄三的圖中，顯現了每一個模型中每一筆資料與目標座標的偏差。紅色線為被判斷為錯誤的測資，綠色線為被判斷正確的測資。各線段的交點為我們的 35 個目標座標。Original 項目為受測者原本點按的位置與目標位置的偏差，而其他的項目為訓練出的模型預測的點與目標位置的偏差。

(圖以 Original 及 Random Forest Regressor 為範例)

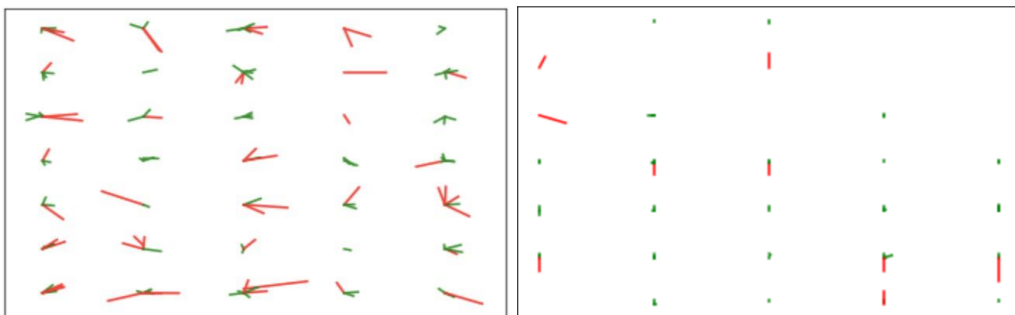


圖 24-1：目標位置偏差圖-Original

圖 24-2：預測位置偏差圖- Random Forest



如上圖 24-1 及圖 24-2，不論紅色或綠色的線條皆變短了，紅色線也變少，代表經過模型預測後的位置更接近原始目標位置、準確率變更高。

測試資料的原準確率為 58.3%，而套入各種模型後的 RMSE（方均根）和校正準確率皆有所提升。因此接下來的實驗我們皆選用 Random Forest Regressor 這個模型進行實驗。

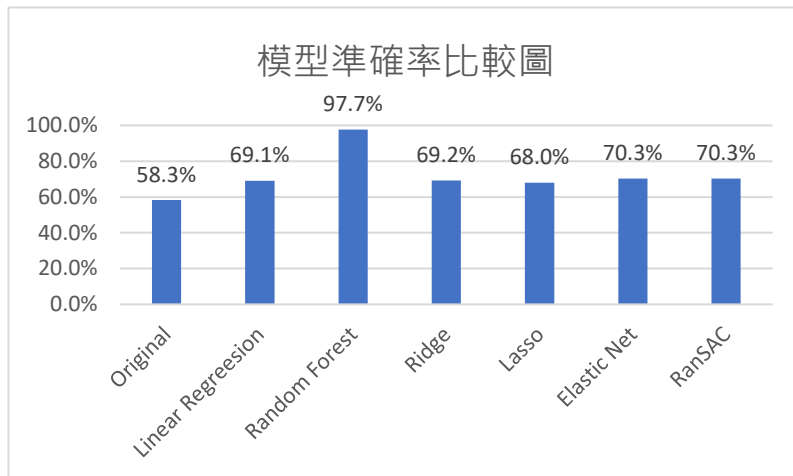


圖 25：模型準確率比較圖

模型	RMSE	準確率
Original	197.3	58.3%
Linear Regression	12.7	69.1%
Random Forest	5.2	97.7%
Ridge Regression	12.0	69.2%
Lasso	13.2	68.0%
ElasticNet	12.4	70.3%
RANSAC	12.7	70.3%

我們利用 Bartlett test 去驗證 Original Data 和 Random Forest Regressor 訓練過的 Data 兩個群體的 error distance（每個點按位置和目標位置之間的偏差值）之變異數是否有差異（資料散布情況），我們在 Random Forest Regressor 的模型上得到  $p\text{-value}<0.001$  的結果，說明校正後的變異數存在顯著的差異。

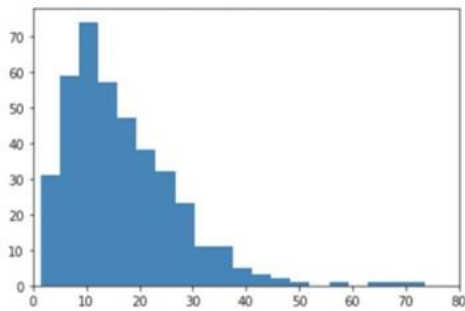


圖 26-1：Bartlett test-Original

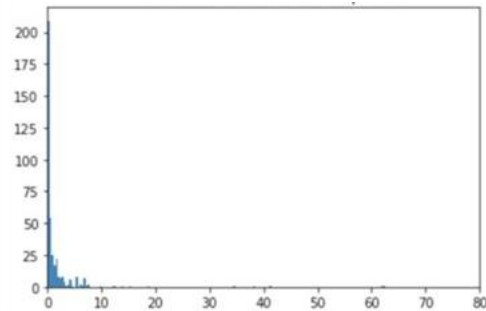


圖 26-2：Bartlett test-Random Forest

上圖 26-1 和圖 26-2 分別是 Original Data 和 Random Forest Regressor 訓練過的 Data 的偏差值的散布圖。橫軸為偏差值之高低（以 pt 為單位），縱軸為次數統計。我們根據散布圖的結果，我們可以推論出 Random Forest Regressor 的模型可以有效的緩解錯誤率。這一項檢測一般採用的是 F test，但是 F test 需要 Data 呈現常態分布，我們的 Data 並沒有呈現常態分布，我們便使用 Bartlett test。

## 二、校正成果分析

表 9：套模型校正後 RMSE 及準確率表

原 RMSE	校正後 RMSE	校正幅度	原點按準確率	校正後點按準確率	準確率提升幅度
197.3	5.3	97.3%	58.3%	97.7%	67.6%

註：RMSE 為方均根誤差，為所有的點擊的誤差開平方後取算數平均數再開根號。  
校正幅度為  $(\text{原 RMSE} - \text{校正後 RMSE}) / \text{原 RMSE}$ 。  
準確率提升幅度  $(\text{原點擊準確率} - \text{校正後點擊準確率}) / \text{原點擊準確率}$ 。

上表是我們用 Random Forest Regressor 對老年人坐著時點按測試進行校正的結果。可以看見，經過模型校正後，不論是在 RMSE 還是準確率都有顯著的進步，RMSE 校正量為 97.3%，點按準確率提升了原來的 39.4%。

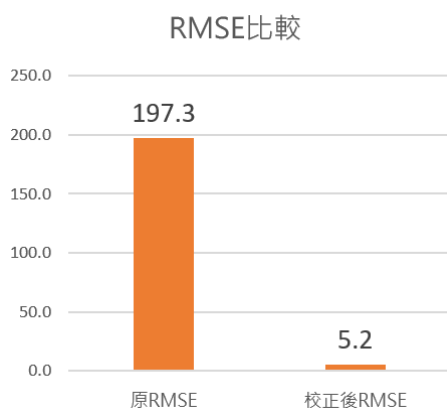


圖 27：模型準確率比較圖

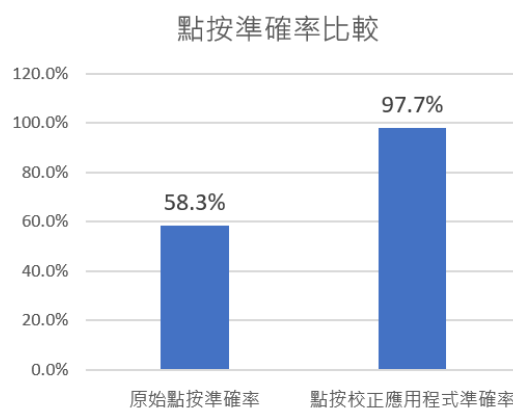


圖 27：模型準確率比較圖

### 三、個人模型校正成效

下表為我們尋找一位使用者收集 34 筆資料所訓練的個人校正模型。

表 10：個人模型成效表

模型	個人模型準確率	全部使用者的模型準確率
Original	59.0%	58.0%
Linear Regression	74.6%	69.1%
Random Forest	98.5%	97.7%
Ridge Regression	75.9%	69.2%
Lasso	73.7%	68.0%
ElasticNet	76.7%	70.3%
RANSAC	76.2%	70.3%

個人測試資料的原始準確率為 59.0%，經由每一種訓練出的模型皆可校正至少 8 個百分比。與全部使用者所訓練出的模型相比，個人模型的每一種模型預測出的準確率皆可提升較多的百分比，其中以 Random Forest Regressor 預測出的準確率來到 98.5%。推測因為個人的資料有比較相似的點按方式，資料變化不大，這讓模型能夠利用各種特徵更準確的預測個人點按位置。

### 四、將模型代入測試 app

將點按校正應用程式做好後，分別尋找新的公園年長者受測，及原本個人模型的使用者受測。在針對所有使用者資料進行訓練的模型中，手機能及時將原本準確率 60.4%的點按準確率提升至 94.2%，提升了 33.8%。而個人模型中手機能將原本準確率 64.0%提升至 97.5%，提升了 33.5%。

表 11：模型套入手機程式成效表

模型	原始點按準確率	點按校正應用程式準確率
整體	60.4%	94.2%
個人	64.0%	98.5%

## 五、以不同特徵項目訓練模型

在進行特徵項目的篩選時，一開始我們將資料分成了觸控軌跡資料和觸控軌跡資料+裝置將速度、陀螺儀、位移資料，總共 33 項特徵項目進行訓練。而後來考慮到裝置位移資料外的特徵項目對於觸控螢幕的影響可能沒有那麼大，我們實驗只用位移資料當作特徵項目進行訓練。

下表為兩種特徵項目的訓練成效，顯示了裝置相關資料作為特徵項目比起只有位移資料的特徵項目提升了更多的校正準確率。證實了使用者點按時，影響準確率的因素不只有使用者點按的觸控軌跡，還有裝置本身的晃動。

表 12：不同特徵項目成效比較表

模型	觸控軌跡資料	觸控軌跡資料+其它裝置相關資料
Original	58.3%	58.3%
Linear Regression	63.9%	69.1%
Random Forest	96.6%	97.7%
Ridge Regression	62.8%	69.2%
Lasso	61.6%	68.0%
ElasticNet	64.4%	70.3%
RANSAC	63.7%	70.3%

## 六、使用者的身心狀況與點按準確率和校正成效之關係

根據使用者所填寫的問卷，有 18%的使用者平時覺得撿取小物件是困難的，有 10%的使用者有在服用藥物，有 58%的使用者有時會焦慮，有 3%的使用者常常會焦慮，有 74%的使用者視力不佳。這些使用者他們的原始準確率通常會比整體的原始準確率低，可以得知這些項目的確會影響使用者的點按準確率。然而經過模型校正後，他們的準確率也可以顯著的提升。可以得知整體的模型能夠有效提升所有使用者的準確率，無論使用者是否有特殊的身心狀況。

表 13：特殊身心狀況準確率比較表

項目	原始準確率	模型校正後的準確率
有撿取小物件的困難	45.3%	92.6%
有服用藥物	47.8%	93.9%
有時會焦慮	56.4%	97.5%
常常會焦慮	31.4%	84.3%
視力不佳	54.1%	97.4%

## 肆、 結論與應用

### 一、 問題與討論

(一) 模型預測及應用程式反應的時間是否會造成延遲而造成使用者不便？

根據計算，在實際將模型套入我們的應用程式使用時，當接收到使用者觸控螢幕的點按資料時，便會透過模型預測一個點按位置，並由手機做判斷，期間約 0.0175 秒。這個模型判斷的時間並不至於影響到使用者者的觸控螢幕體驗。

另外我們也針對點按不同次數所需花的平均時間進行實驗。發現不論是進行 10 次點按，或是進行 100 次點按，所需花費的平均時間都落在 0.01~0.02 秒之間，因此得知點按次數的多寡也不會影響程式的運行時間。

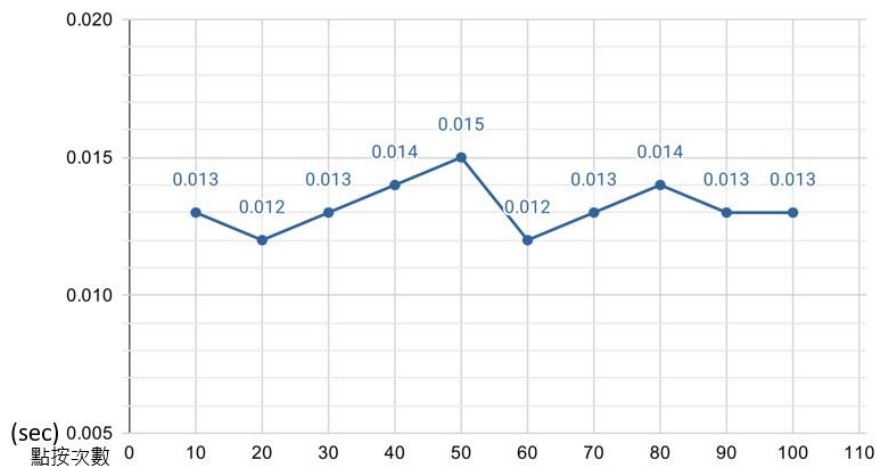


圖 28：點按次數對反應時間折線圖

(二) 訓練後的模型是否會造成使用者點不到想點按的位置的問題？

目前我們將螢幕分割成 7\*5 個區塊來做實驗，而在進行模型的校正時，我們已將所有可能的偏差考慮在內，包含手指過大導致的偏差，以及裝置傾斜所導致的偏差等等。我們以此實驗驗證這種模型校

正法的可能性，未來也能將它運用至不只本實驗的區塊上進行模型的訓練及準確率的校正。

(三) 本研究中 Random Forest Regressor 方法最為突出的原因？

本研究以 Random Forest Regressor 這個模型進行校準時的校正準確率提升最高。推測原因為 Random Forest Regressor 採用了集成算法，本身精度比大多數單個算法要好，所以準確性會較高。Random Forest 有兩種演算法，分別為回歸線及分類器；而在本研究中，我們採用的演算法是回歸線。依據下面 Random Forest Regressor 的目標位置偏差圖可以看見，它所預測的結果大部分都出現在 35 個目標座標附近，這表示 Random Forest Regressor 的回歸線在校正時聚焦於那 35 個座標，因此它能準確的校正這些位置。然而並非所有的校正結果都會準確的落在那 35 個點上，證明我們採用的 Random Forest Regressor 模型是回歸演算法，其結果不會造成在其它的應用程式上就只會判斷點按位置為這 35 個點的問題。

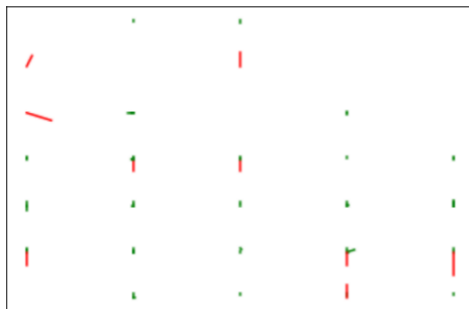


圖 29-1：目標位置偏差圖  
-Random Forest

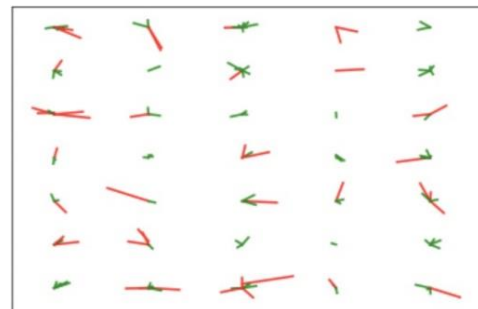


圖 29-1：目標位置偏差圖  
-Linear Regression

## 二、總結與應用

本研究收集及分析年長者使用觸控螢幕時，系統判斷手勢以及點按位置的準確度。相較其它研究未針對年齡進行探討，本研究主要針對年長者



對於按不準目標位置進行研究及校準，我們也藉由實驗發現年長者確實有校正的需求，期望年長者能因為此研究而能更為順暢的使用手機。

本研究收集了廣泛而齊全的使用者背景資料。用多項特徵項目進行前處理，以六種模型進行校正。在分析及比較各模型之優劣及校正成效後，本研究以加速度、陀螺儀、時間、和點按起終點之座標作等等特徵，配合校正成效最明顯的 Random Forest Regressor 進行校正。本研究之校正方法通用於多元化的資料及較有規律的資料，在針對相似年齡的年長使用者之點按位置進行校正時，效果顯著。並能將模型實際運用於手機程式，使年長者使用手機時能即時且顯著的提升點按準確率，證明了以此方法校正點按準確率的可行性。

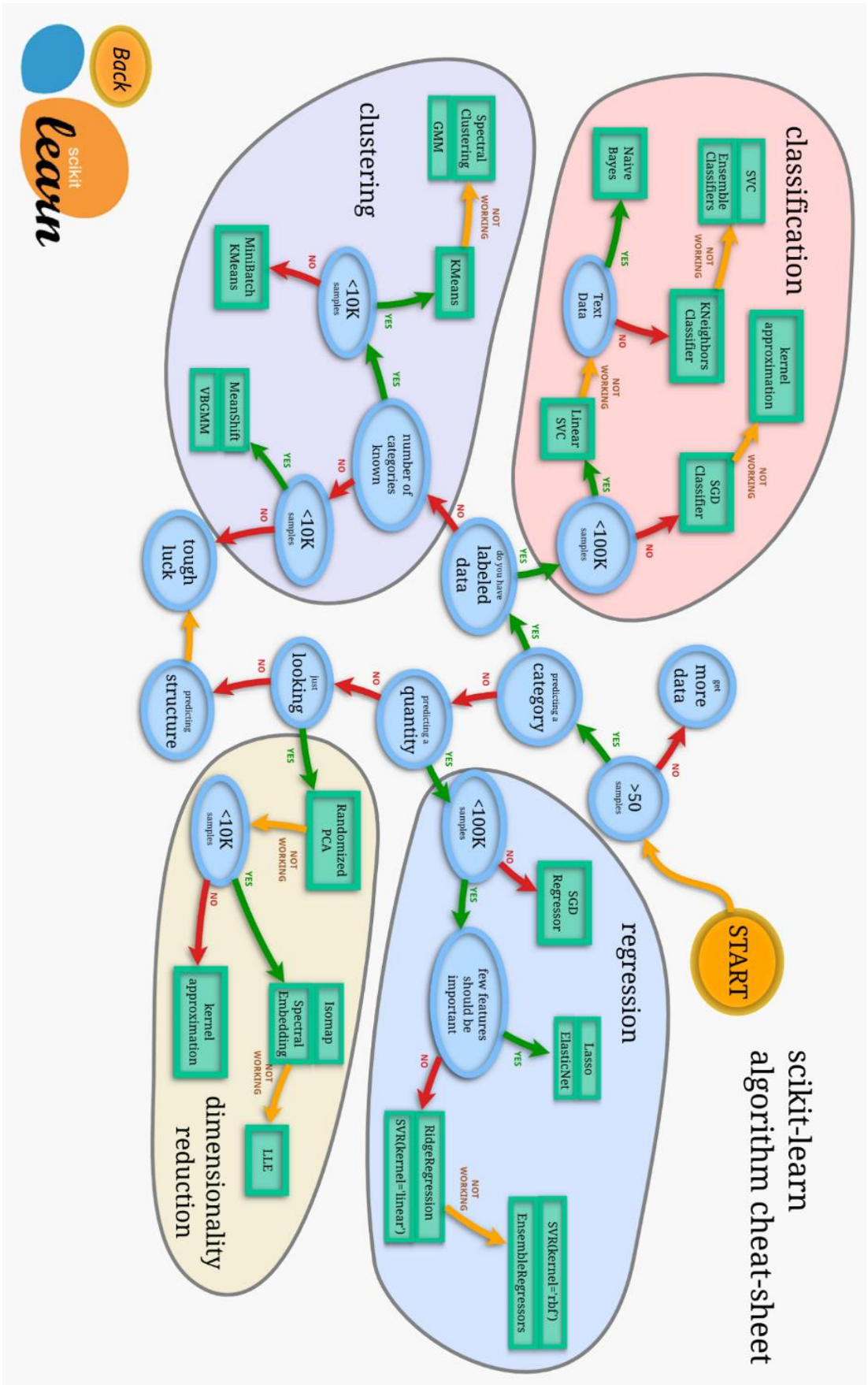
### 三、未來與展望

- (一) 將此模型運用於其它手機應用程式之中，讓更多年長者能在使用各種手機應用程式時及時並有效的提升觸控螢幕的點按準確率。
- (二) 蒐集手機螢幕使用熱區（即螢幕上最常被點按的位置，如返回鍵），加強針對熱區的模型訓練。

### 伍、參考文獻

1. Martez E. Mott, & Jacob O. Wobbrock. ( 2019, May ) . *Cluster Touch: Improving Touch Accuracy on Smartphones for People with Motor and Situational Impairments*. Paper proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland UK.
2. Leah Findlater, Mayank Goel, Jacob Wobbrock. ( 2012, May ) . *WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry*. Paper proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, Texas, USA.

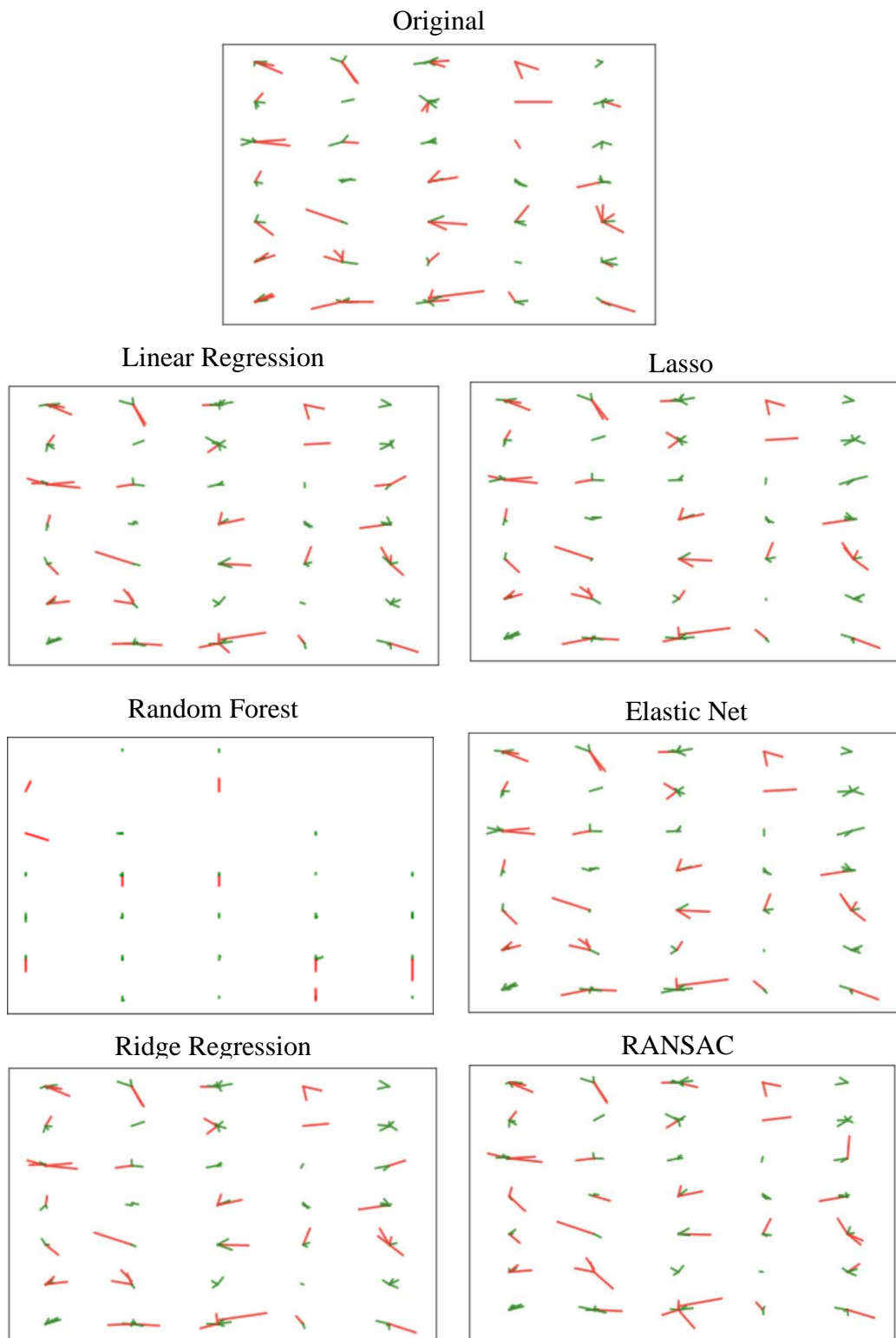
3. Niels van Berkel, Jorge Goncalves, Vassilis Kostakos, Chu Luo, Zhanna Sarsenbayeva. ( 2017, November ) . *Challenges of situational impairments during interaction with mobile devices*. Paper proceedings of the 29th Australian Conference on Computer-Human Interaction, Brisbane, Queensland, Australia.
4. Stephen A. Brewster, Alexander Ng, John H. Williamson. ( 2014, September ) . *Comparing evaluation methods for encumbrance and walking on interaction with touchscreen mobile devices*. Paper proceedings of the 16th international conference on Human-computer interaction with mobile devices & services, Toronto, ON, Canada.
5. Kyle Montague, Vicki L. Hanson, Andy Cogley. ( 2012, Oct ) . *Designing for individuals: usable touch-screen interaction through shared user models*. 14th International ACM SIGACCESS Conference on Computers and Accessibility - Millenium Harvest House Hotel, Boulder, Colorado, United States.
6. Pradipta Biswas, Pat Langdon. ( 2012 September ) . *Developing Multimodal Adaptation Algorithm for Mobility Impaired Users by Evaluating Their Hand Strength*. International Journal of Human-Computer Interaction.



附錄二、各模型所使用的係數

<p>Linear Regression</p>	<p>[[ 2.41798892e-01 -5.18151204e-02 7.51755071e-01 4.70149356e-02 -3.98619422e-02 -2.07155715e+01 6.38261209e+00 2.26007561e+01 1.75565353e+01 -1.40214544e+01 -5.13643226e+01 -4.56816813e+01 7.63872783e+00 2.87622974e+01 2.81254697e+01] [ 1.76421823e-01 3.03032457e-01 -1.76413409e-01 7.06554640e-01 2.12179276e-01 -1.91027063e+01 -4.19591415e+01 -1.09109160e+01 -4.72260148e+01 8.66488779e+01 2.23198973e+01 1.04789100e+02 -4.46895839e+01 -1.14083379e+01 -5.75581843e+01]]</p>
<p>Ridge Regression</p>	<p>[[ 0.23422133 -0.04926419 0.75924647 0.04442937 -0.04533287 -17.97871935 0.23855894 0.59121483 -4.72754276 -1.53673222 -6.11525161 0.88818052 1.29812269 5.5230274 3.84013832] [ 0.16927805 0.30304257 -0.16931558 0.70659289 0.20822054 -17.08714276 -1.50314194 0.3278344 3.03491176 5.5508724 -0.41344712 2.75312877 -4.04747434 0.08624638 -5.78341201]]</p>
<p>Lasso</p>	<p>[[ 5.51610855e-01 -5.52213214e-04 4.41312682e-01 -4.85834569e-03 -6.23540318e-02 -0.00000000e+00 7.98781276e-05 0.00000000e+00 2.01117533e-03 0.00000000e+00 2.85105140e-04 6.19761413e-04 0.00000000e+00 1.49719396e-04 3.72106952e-04] [ 6.86011343e-02 7.92273860e-01 -6.84801326e-02 2.17104222e-01 3.31341879e-02 -0.00000000e+00 -3.34189993e-04 2.58042363e-04 5.37571759e-04 -5.45212185e-05 1.29667740e-04 2.28399532e-06 -1.76094740e-04 2.57607360e-05 0.00000000e+00]]</p>
<p>Elastic Net</p>	<p>[[ 5.42699551e-01 -4.20721319e-03 4.50167704e-01 -1.16438554e-03 -4.25053979e-02 -0.00000000e+00 7.18019090e-05 0.00000000e+00 1.98085942e-03 0.00000000e+00 2.80045500e-04 6.32209450e-04 0.00000000e+00 1.51824740e-04 3.85546295e-04] [ 2.78333728e-02 7.81560040e-01 -2.76516031e-02 2.27815615e-01 1.45311267e-02 -0.00000000e+00 -3.15727469e-04 2.46983471e-04 5.37250769e-04 -4.88657724e-05 1.30294156e-04 1.35022851e-05 -1.70478328e-04 2.86589875e-05 0.00000000e+00]]</p>

附錄三、各模型的位置偏差圖



## 【評語】 190019

本研究主題清楚且聚焦，建議要有更嚴謹的實驗設計，以用科學方法檢驗研究成果。另建議作品用以解決的問題要有其必要性(年長者觸控問題是否存在?)。