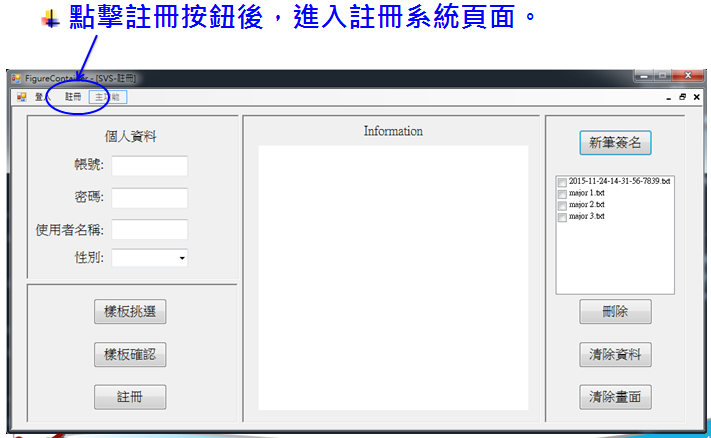
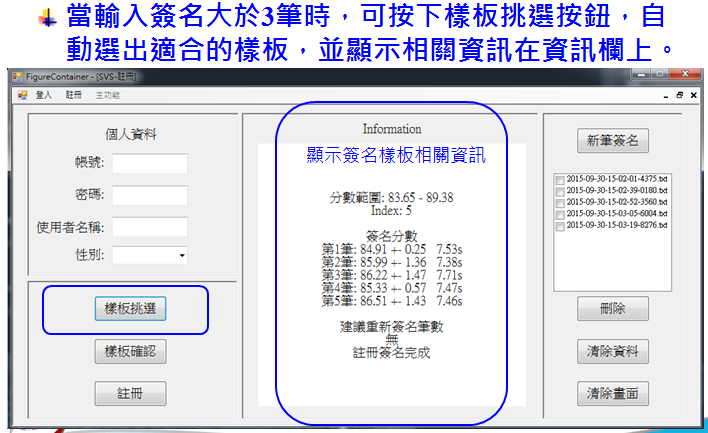
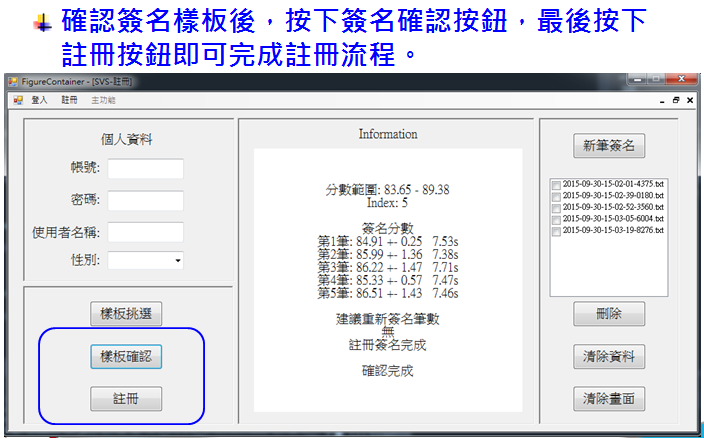
SVSystem專案程式說明

2016/5/19

* UI介面說明
* Library說明
* Database說明-使用前要開啟

## UI介面說明

## SVSystem library使用說明

介紹函數：

1. SVS\_Reset()
2. DataDecoder()

前處理步驟

1. SignatureDataProcessing()
2. CheckFeature()

樣板挑選

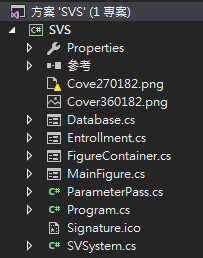
1. SignatureTemplateSelection()
2. Recognizer()

相似度比對及更改靈敏度設定

* 在SVS此專案中，基本上每一部分程式都有描述其功能，以下將針對專案的架構以及重要的簽名系統函數進行介紹。
* 在此專案中，演算法部分建立在SVSystem.cs中，每一筆簽名資料均須建立用SVSystem宣告的變數，例如：



* 專案下目錄：



其中，Database.cs為登入頁面、

Entrollment.cs為註冊頁面、

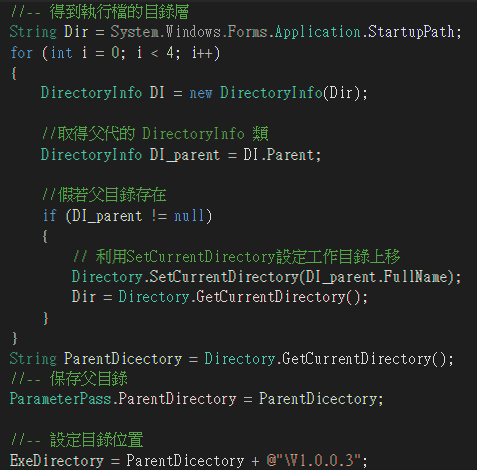
MainFigure.cs為簽名識別頁面、

SVSystem.cs為演算法函數庫。

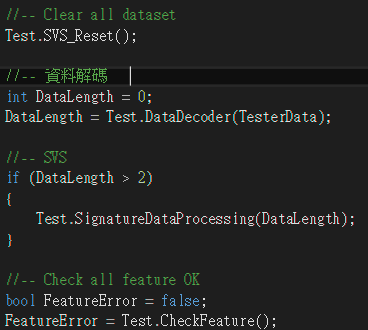
* 預設簽名訊號擷取執行檔資料夾位置為VS專案同一層目錄。

(如要更改位置，請改Entrollment.cs和MainFigure.cs建構子函數中的設定。)

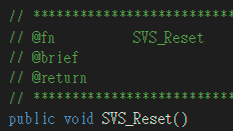




前處理步驟：



* 清除所有變數函數：SVS\_Reset()



範例使用說明：



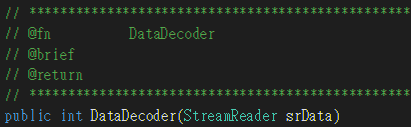
* 訊號解碼函數：DataDecoder()

從txt檔案擷取出XYPT資料，在Entrollment.cs和MainFigure.cs中皆有使用到。

輸入 srData：txt檔資料

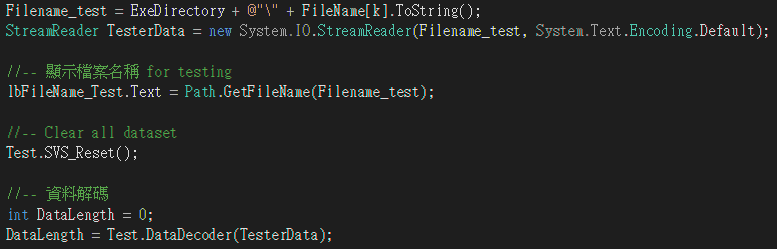
輸出：訊號長度

解碼函數：



使用範例：

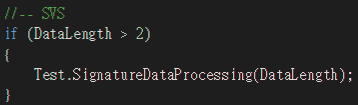
利用StreamReader讀取txt檔案資料，在代入DataDecoder函數中。



* 訊號前處理函數：SignatureDataProcessing()

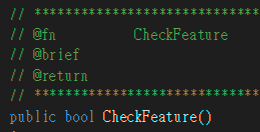
輸入 DataLength：解碼完成後的資料長度

解碼函數完成後，代入訊號前處理函數。

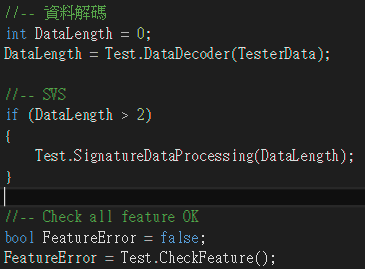


* 確認資料是否解碼及前處理成功：CheckFeature()

輸出：確認正確後回傳 true，否則false。



使用範例說明：



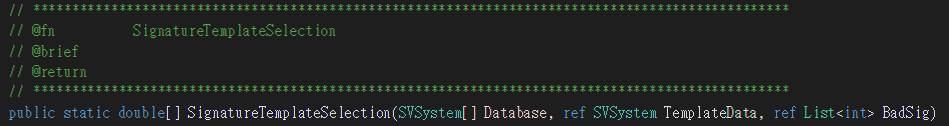
* Entrollment.cs：樣板挑選函數

在註冊頁面主要講解樣板挑選函數的使用。

在此函數前半部是擷取所有訓練樣板訊號，並且完成解碼部分，儲存於。

後半部分則將訓練資料代入樣板挑選演算法的函數

SignatureTemplateSelection()



其中輸入為， Database：訓練資料、

TemplateData：訓練後的樣板相關訊息、

BadSig：需要建議重簽的簽名筆數。

TemplateData和BadSig為演算法執行結束後，將相關訊息寫入該陣列中。

TemplateData陣列說明：

|  |  |
| --- | --- |
| [0] | 最佳樣板結果 |
| [1] | 最佳樣板分數(平均值-標準差) |
| [2] | 最佳樣板分數(平均值) |
| [3] | 最佳樣板分數(標準差) |
| [4] | 最大建議分數 |
| [5] | 最低建議分數(預設樣板建議分數) |
| [6] | 總簽名時間 |
| [7~9] | 0 |
| [10] | 第一筆訓練資料分數平均值 |
| [11] | 第一筆訓練資料分數標準差 |
| [12] | 第二筆訓練資料分數平均值 |
| [13] | 第二筆訓練資料分數標準差 |
| [14] | 以此接續，直至最後一筆訓練資料 |
| [15] | … |
| … | … |

* MainFigure.cs：訊號比對函數

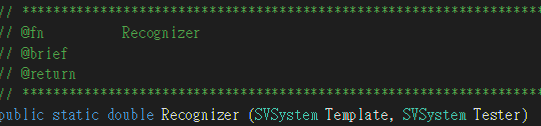
在簽名識別頁面中，主要講解選擇好測試的簽名後，與樣板簽名進行相似度比對的程式使用。

訊號比對函數： Recognizer()

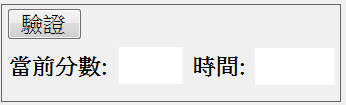
輸入 Template: 樣板簽名

Tester：輸入簽名

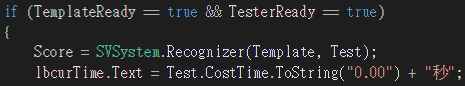
回傳：相似度分數



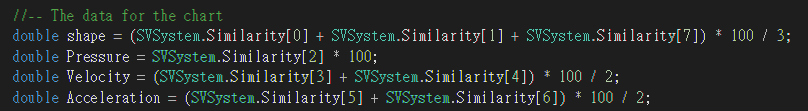
按下則將輸入的簽名進行解碼和訊號前處理。

後按下則將該簽名與樣板簽名做相似度比對，並顯示在中。

範例說明：



當按下驗證後，除了顯示相似度分數以及書寫時間，而外分析四種特徵相似度分數供參考，分別為外型、筆壓、速度及加速度部分。

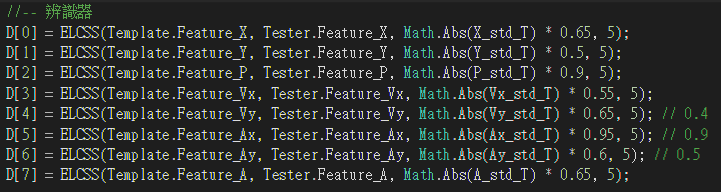


|  |  |
| --- | --- |
| [0] | X座標特徵分數 |
| [1] | Y座標特徵分數 |
| [2] | P壓力特徵分數 |
| [3] | X速度特徵分數 |
| [4] | Y速度特徵分數 |
| [5] | X加速度特徵分數 |
| [6] | Y加速度特徵分數 |
| [7] | A筆跡角度特徵分數 |

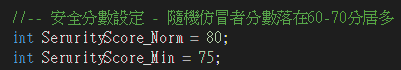
Similarity陣列說明：

CostTime：書寫時間

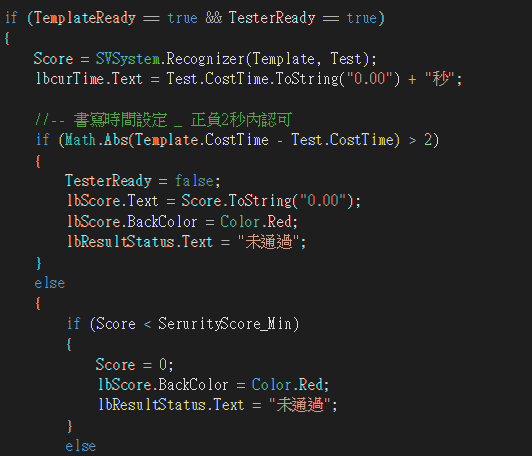
在Recognizer()下更改比對靈敏度設定，更改在X\_std\_T、Y\_std\_T等等乘的係數即可更改靈敏度，越高代表越容易讓仿冒者仿冒，而越低代表本人越難通過，建議無大量數據測試的話，不要更改目前參數。



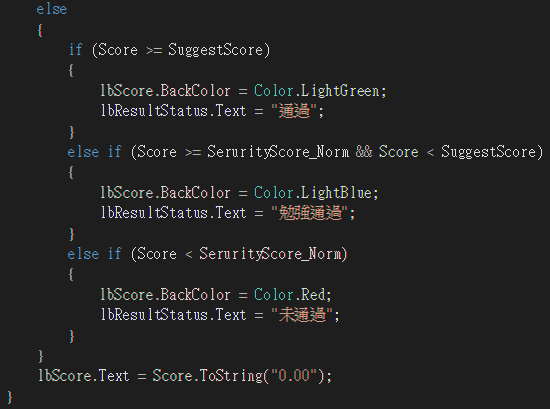
* 真偽判別標準：



1. 當前書寫時間在樣板簽名時間正負2秒內通過，否則未通過。
2. 相似度分數直接小於最低通過標準，預設75分，則未通過。



1. 相似度分數大於建議分數，則通過。
2. 相似度分數介於最低通過標準和建議分數之間，則勉強通過。



## Database說明

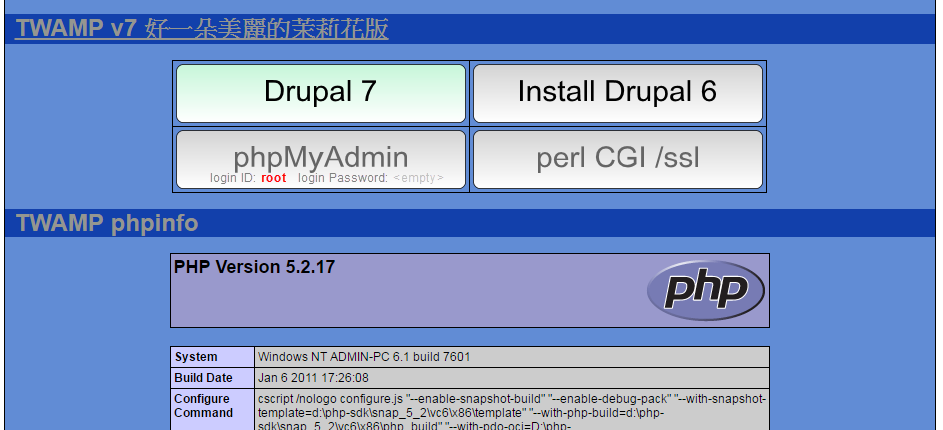
資料庫使用MySQL，利用第三方phpMyAdmin網頁設定來完成。

**注意：該資料庫資料夾要放置C槽下，否則無法正常開啟。**

步驟一：開啟本機伺服器，執行apmxe\_zh-TW.exe。



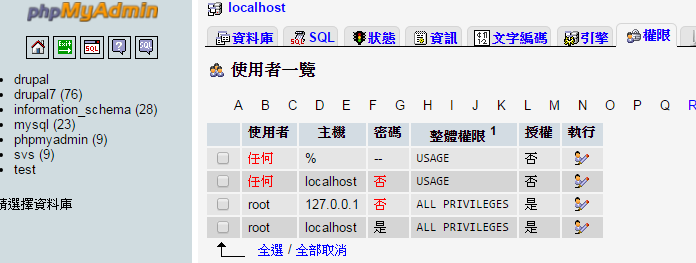
步驟二：開啟網頁，輸入127.0.0.1進入下方畫面，點選phpMyAdmin。



步驟三：帳號為root，無密碼，直接點選執行，進入設定。



若要新增使用者，點選權限。



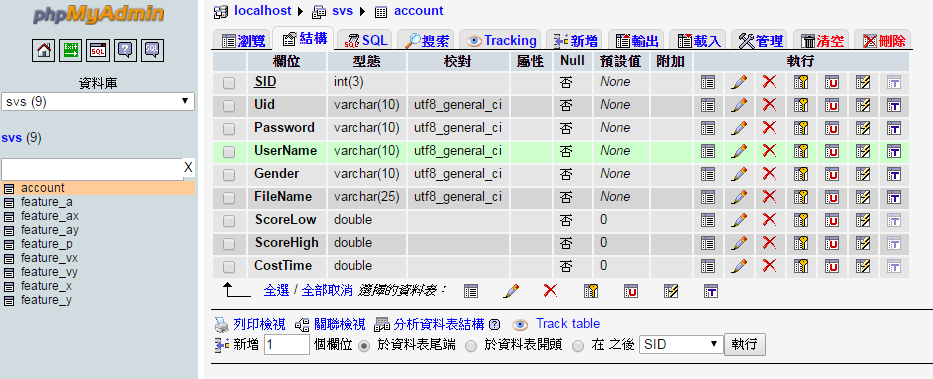
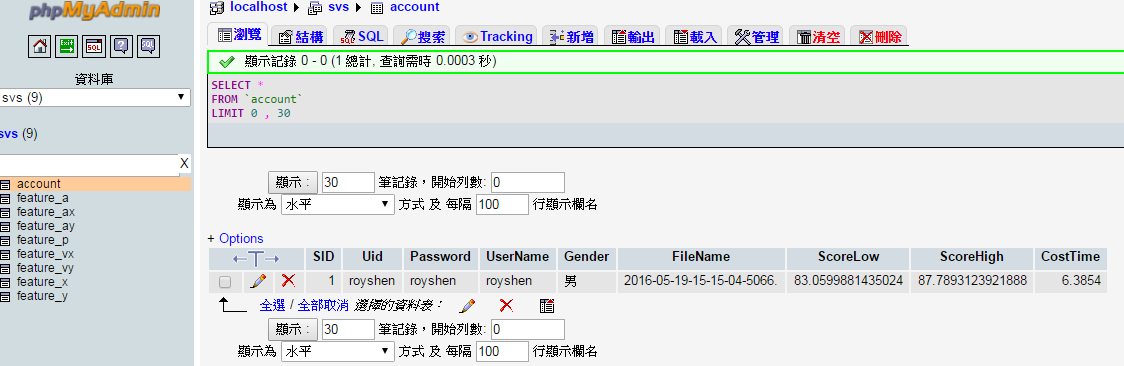
步驟四：進入以下畫面，點選svs資料庫。



步驟五：資料庫中已建立9個資料表，分別為帳號及共8個特徵表。

帳號密碼儲存格式為：

|  |  |
| --- | --- |
| Sid | 流水序號，新增每一人次逐增加1 |
| Uid | 帳號 |
| Password | 密碼 |
| UserName | 使用者名稱 |
| Gender | 性別 |
| FileName | 檔案名稱 |
| ScoreLow | 最低建議分數 |
| ScoreHigh | 最高建議分數 |
| CostTime | 簽名總時間 |

每個特徵表中均建立128筆資料點，按下新增即可增加新欄位。

