

# 第四屆旺宏科學獎 成果報告書

參賽編號：SA4-100

作品名稱：鳥的身分證-鳥類聲譜資料庫的建立

姓名：陳冠儒

關鍵字：聲紋圖、時頻圖、傅立葉轉換

## 一、研究動機：

常常看到在家門口大樹上及隔壁鄰居屋頂上，到處有小鳥飛來飛去，但是如果對鳥類不是很熟悉的人，乍看之下都無法確認牠們是什麼鳥？屬於那一科？因為捕捉鳥不容易，於是想到我們可以利用鳥的叫聲去辨識牠的種類。

## 二、研究目的：

- (一) 利用 SFS 軟體分析各種鳥類聲音的頻率，並產生時間對頻率圖(時頻圖)及聲紋圖，建立一個鳥類聲譜資料庫，方便以後比對。
- (二) 用已知鳥叫聲的時頻圖及聲紋圖來建立資料庫，取未知的鳥叫聲比對資料庫，來判斷生物的種類。

## 三、研究過程：

- (一) 利用孫清松先生所提供台灣常見的 91 種鳥類的叫聲，共 225 個聲音檔，分析並輸入聲譜資料庫建檔。利用 SFS 軟體來分析所錄製的鳥類叫聲強度及頻譜，並產生時頻圖及聲紋圖，將所產生的圖形數位化且建立成資料庫，再用 C++Builder 6.0 寫 C++程式讀取資料庫，產生 normalize(數位化)的圖形時頻圖檔，比對了解未知鳥類的種類。
- (二) 一般鳥類的聲音分為鳴唱(song)和叫聲(call)，鳴唱通常比較長、複雜且較多變化，通常是在繁殖期的雄鳥會出現這種聲音，而叫聲則較為短促及簡單，而包括雄鳥和雌鳥，整年度都會發出這種聲音，此次研究偏重於鳴唱，叫聲也有做一些，一般我們所聽到鳥類清脆悅耳的歌聲，多屬鳴唱。

## 四、討論及應用：

- (一)
  - 1. 取得 225 個鳥鳴聲的 wave 檔案，依序編號。
  - 2. 利用 SFS 軟體分析鳥叫聲之 wave 檔，以產生聲紋圖，並利用其中傅立葉轉換公式，轉換成時頻圖，如圖 1 及圖 2。

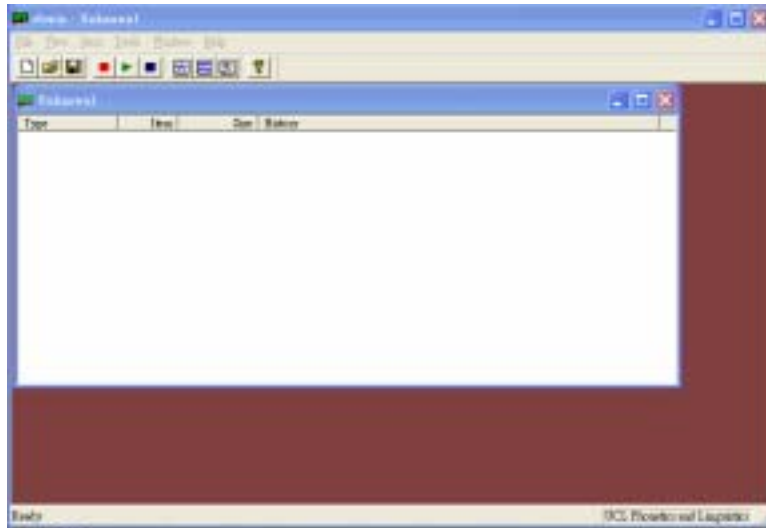


圖 1 SFS 軟體的圖形使用者介面



圖 2 利用 SFS 所產生的聲紋圖(上)及時頻圖(下)

3. 將 SFS 所產生的時間對頻率圖數位化：依時間對頻率圖中鳴叫聲的時間點(X 軸)及頻率(Y 軸)，取出每一個柱狀圖的左上座標(X1, Y1)以及右下座標(X2, Y2)輸入至鳥鳴資料庫，時間單位為秒、頻率單位為 Hz，Y 軸單位以 KHz 來表示。
4. 利用數位化後的資料建立鳥鳴資料庫

1	7	0.4	18	0.6	3.4	0.8	18	1	3.4
2	7	1.9	19.4	2.1	3.6	2.2	19.4	2.4	3.6
3	6	1.9	12.4	2.1	1.4	2.4	28.4	2.6	2
4	6	0.7	11	1.4	1	3	11	3.3	1
5	4	1.3	13	1.6	1	4.5	13	4.8	1
6	6	0.35	13	0.5	1	0.6	18	0.7	1
7	6	0.55	15	0.65	1	0.8	12.5	0.9	1
8	7	3	17	3.5	1.6	4	17	4.5	1.6
9	9	0.4	19.5	0.5	2	0.7	19.5	0.78	2
10	11	0.5	15	0.6	1.5	0.8	15	1	1.5
11	10	1.2	18	1.2	2	1.38	9	1.43	2
12	7	0.7	15.6	0.76	1.8	2.47	15.6	2.53	1.8
13	7	0.4	28	0.5	1.4	1.2	28	1.25	1.4
14	7	0.48	13.4	0.55	1.3	0.84	13.4	0.96	1.3
15	11	0.88	12.7	0.12	3	0.16	12.7	0.21	3
16	11	0.85	15	0.14	3.5	0.19	14.8	0.37	3.5
17	4	1.32	15.9	1.53	1.4	2.91	15.9	3.11	1.3
18	8	0.17	18	0.3	1.54	0.54	18	0.63	1.54
19	11	1.21	3.6	1.4	1	1.45	4.1	1.52	1.4
20	8	1.88	7.5	1.27	2.9	1.81	11	1.6	3.1
21	9	1.71	4.1	2.86	2.9	2.89	14.5	2.73	2.2
22	11	0.68	4.3	1.86	3.1	1.11	7.3	1.43	2.4
23	18	1.98	28	2.12	3.5	2.16	28.6	2.46	3.5
24	9	1.8	19.7	2.2	1.7	2.42	19.7	2.74	1.7
25	8	0.5	16	0.64	0.8	0.7	16	0.77	0.8
26	11	0.97	13.3	1.11	0.6	1.14	13.3	1.29	1.4
27	8	0.56	3.1	0.69	0.7	0.79	2.7	0.87	0.7
28	5	0.29	8.3	1.12	0.8	4	8.3	4.75	0.8
29	12	2.83	16.5	3.05	0.8	3.14	18	3.4	1
30	8	1.18	9	1.34	1.4	1.52	18	2.53	1.8

圖 3 建立完成的資料庫，依序為：鳥鳴編號、柱狀圖數目、柱狀圖左上角(X1, Y1)、柱狀圖右下角(X2, Y2)、下一個柱狀圖左上角(X1, Y1)、柱狀圖右下角(X2, Y2)依此類推。

5. 用 C++Builder 編寫 C++ 程式 BirdSoundSearching\_v3.exe 將資料庫中的數據以 normalize(數位化)過後的長條圖形方式呈現，程式演算過程如下：

- (1) 取得第一個欄位的鳥鳴編號，並顯示出鳥的名稱。
- (2) 讀取第二個欄位，取得將彙製的總柱狀圖數量，用來配置記憶體大小。
- (3) 接著後面每 4 個欄位為一個單位，分別為(X1, Y1)、(X2, Y2)。

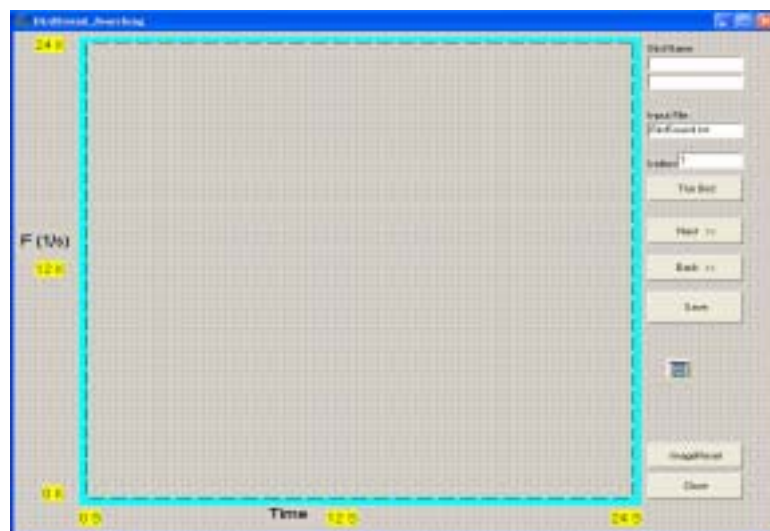


圖 4 bcb 程式的使用圖形介面

6. 將 bcb 程式所產生的圖形建檔

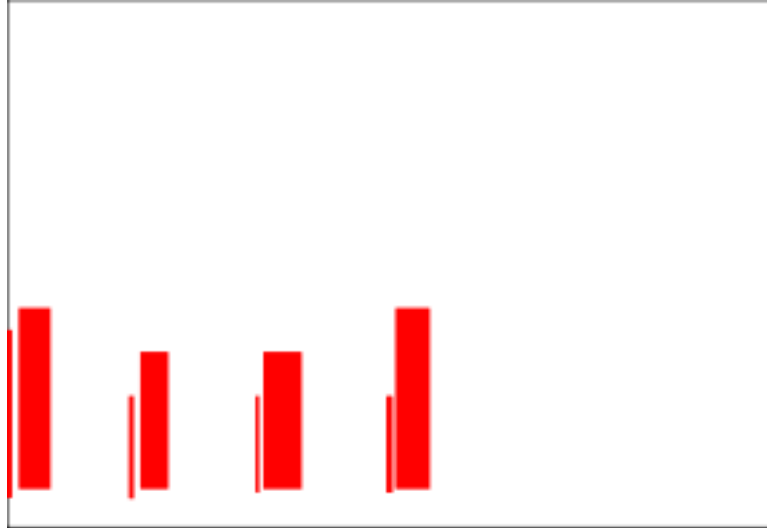


圖 5    bcb 程式所產生的山紅頭鳥鳴聲紋圖數位化的結果，橫座標為時間(24s)，縱座標為頻率(24kHz)

7. 取得未知聲音處理流程

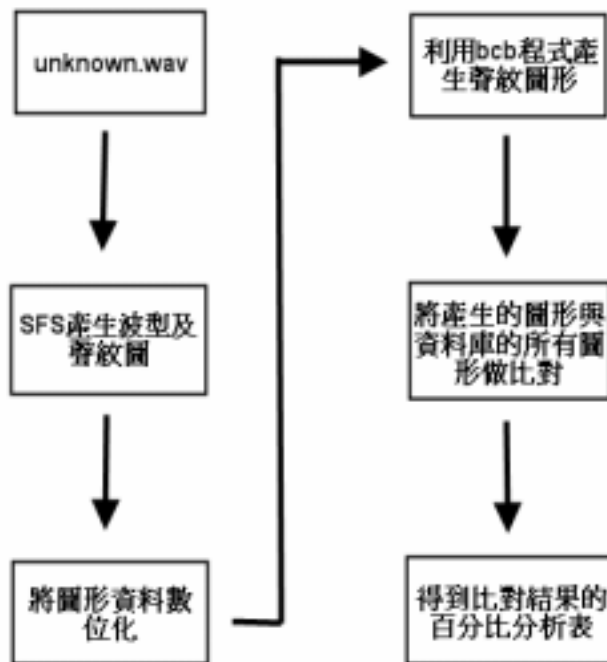


圖 6    未知鳥鳴聲處理流程示意圖

8. 將未知的聲音檔用前面的方法數位化並用 bcb 程式畫成圖形，方便與資料庫中的圖形作比較。

## 9. 比較的方法：

- (1) 因為圖形大部分為白色，為了防止過大的誤差，所以我們只比對紅色的部份，也就是比較兩者之間的含有顏色的位置之 correlation(相似度)，correlation 高代表相似度高。
- (2) 將未知圖形的紅色部分的座標何資料庫中圖形該座標的顏色作比對，計算出同樣為紅色的座標個數與未知圖形紅色座標各數的比值，此即為相似的百分比，又因為怕發生被比資料有整個包含的現象，所以需要相反程序再做一次，就是要再算一個紅色的座標個數與資料庫圖形紅色座標各數的比值。
- (3) 先以資料庫中的第一張圖當作未知圖片，以來測試程式正確無誤。

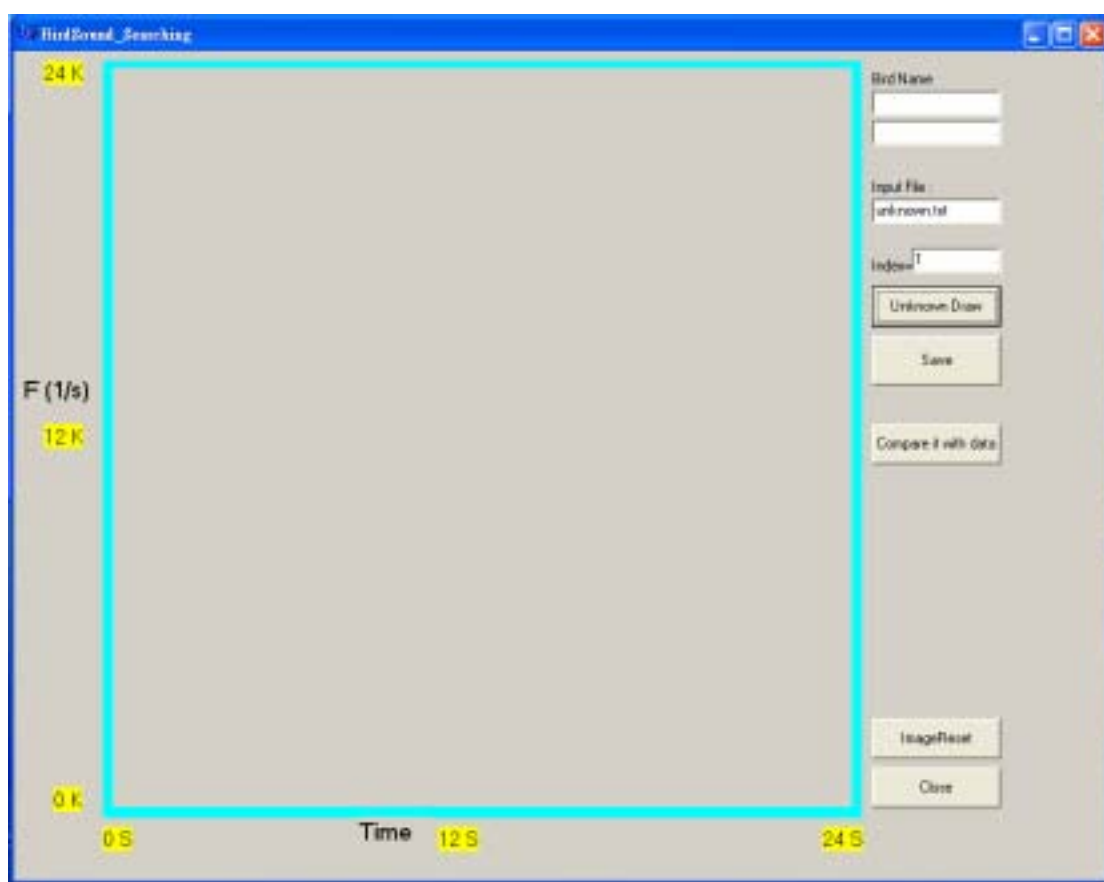


圖 7 圖形比對程式的使用圖形介面

1	1.000	1.000
2	0.706	0.609
3	0.130	0.164
4	0.149	0.098
5	0.078	0.133
6	0.121	0.370
7	0.161	0.512
8	0.532	0.256
9	0.133	0.445
10	0.217	0.283
11	0.158	0.526
12	0.040	0.176
13	0.119	0.477
14	0.114	0.289
15	0.132	0.524
16	0.417	0.557
17	0.183	0.395
18	0.127	0.409
19	0.049	0.272
20	0.135	0.265

圖 8 此為測試的一部份結果，依序為編號、相似度明顯看出除了第一張圖相似度為 1 以外(100%)，其他的相似度都非常低，代表程式正確無誤，原則上這兩部分的比值均應該大於臨界質，0.8 就是擁有相當不錯的相似度。

107	0.050	0.140
108	0.127	0.119
109	0.122	0.247
110	0.191	0.258
111	0.117	0.191
112	0.106	0.151
113	0.122	0.259
114	0.167	0.268
115	0.245	0.245
116	0.822	0.119
117	0.800	0.083
118	0.762	0.088
119	0.163	0.061
120	0.026	0.044
121	0.047	0.055
122	0.243	0.198
123	0.416	0.167
124	0.070	0.163
125	0.061	0.043
126	0.087	0.063

圖 9 紅色圈圈部分明顯出現包含的現象。

(二) 表 1 : 91 種鳥類分類及其聲音率頻資料庫範圍


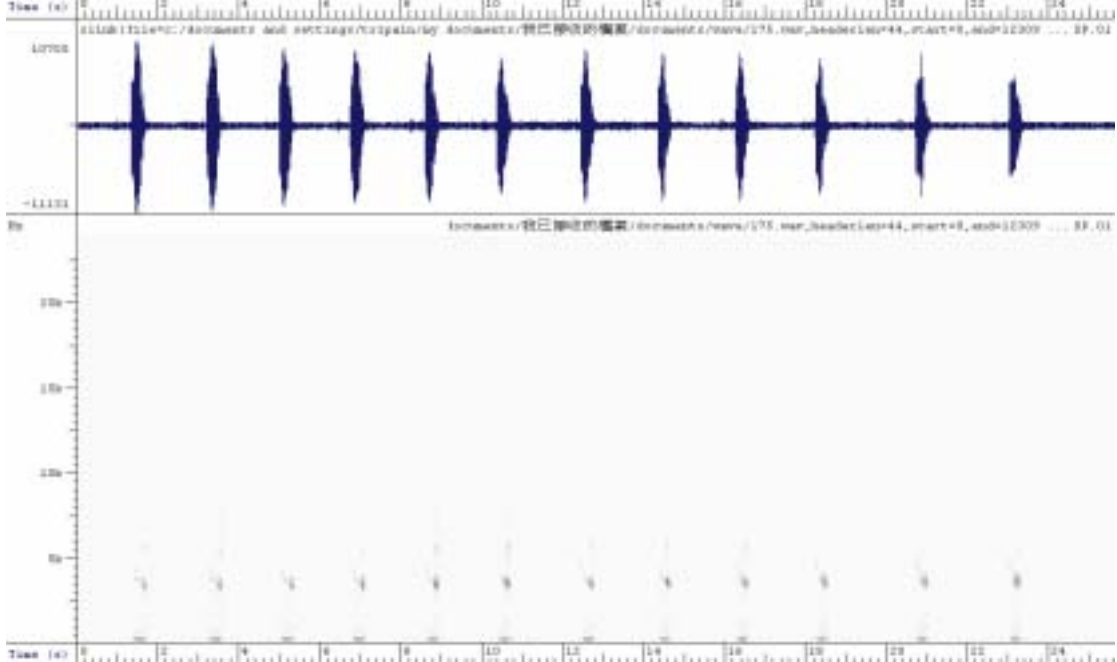
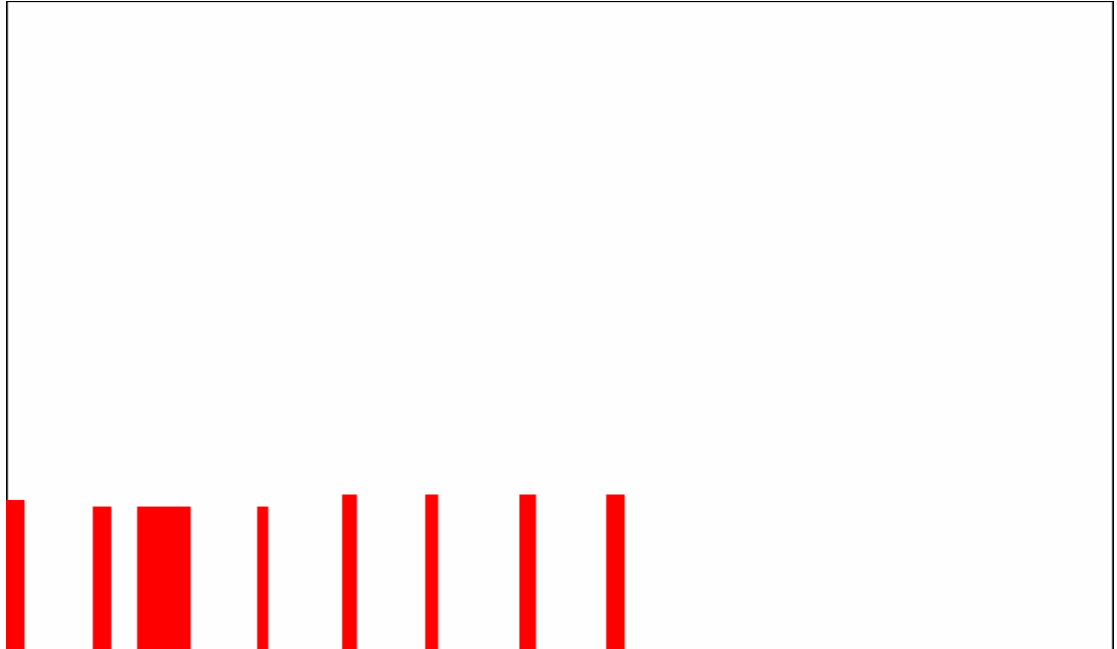
目	科	俗名	編號	頻率範圍由小到大 (kHz)
雀形目	繡眼科	綠繡眼	B1	2.5 , 19
雀形目	鴉科	大卷尾	B2	1 , 22.5
燕雀目	鶇科／畫眉亞科	大彎嘴畫眉	B3	1 , 14
燕雀目	鴉科	小卷尾	B4	1 , 17.5
啄木鳥目	啄木鳥科	小啄木	B5	1 , 17
燕雀目	伯靈科	小雲雀	B6	2.5 , 17.5
鶴形目	珣科	小環頸珣	B7	1 , 16
燕雀目	鶇亞科	小翼鶇	B8	1 , 15
燕雀目	畫眉亞科	小彎嘴畫眉	B9	0.5 , 19
燕雀目	畫眉亞科	山紅頭	B10	2 , 10.5
鵲形目	杜鵑科	中杜鵑	B11	0.5 , 4.5
啄木鳥目	五色鳥科	五色鳥	B12	0.5 , 10
燕雀目	鶯亞科	台灣小鶯	B13	1.5 , 5
雞形目	雉科	台灣山鷓鴣	B14	0.5 , 7
燕雀目	畫眉科	台灣畫眉	B15	1.5 , 20
燕雀目	鶇科	台灣紫嘯鶇	B16	1.5 , 13
燕雀目	鴉科	台灣藍鶇	B17	1 , 15
燕雀目	畫眉亞科	白耳畫眉	B18	1.5 , 18.5
燕雀目	鶇亞科	白尾鶇	B19	2.5 , 23.5
燕雀目	鶇鶇科	白面白鶇鶇	B20	1.5 , 24
鶴形目	秧雞科	白腹秧雞	B21	0.5 , 10.5
燕雀目	鶇科／鶇亞科	白腹鶇	B22	1 , 17.5
燕雀目	鶇科	白頭翁	B23	1 , 24
燕雀目	鶇科	白環鸚嘴鶇	B24	1.5 , 15
燕雀目	雀科	朱雀	B25	5 , 19.5
燕雀目	鶯亞科	灰頭鷓鶯	B26	2 , 17
燕雀目	鶇鶇科	灰鶇鶇	B27	2.5 , 14
燕雀目	雀科	灰鶯	B28	1.5 , 15
鶇鶇目	雉科	竹雞	B29	0.5 , 21.5
雀形目	燕科	赤腰燕	B30	1 , 19
夜鷹目	夜鷹科	夜鷹	B31	2 , 15
燕雀目	岩鶇科	岩鶇	B32	2 , 17



鸛形目	珩科	東方環頸珩	B33	1 , 15
燕雀目	鴉科	松鴉	B34	0.5 , 18
鳩形目	鳩鴿科	金背鳩	B35	0.1 , 9
燕雀目	鶺鴒科／畫眉亞科	金翼白眉	B36	1.5 , 12
燕雀目	畫眉亞科	冠羽畫眉	B37	1.5 , 15
燕雀目	鴉科	星鴉	B38	0.5 , 16
燕雀目	伯勞科	紅尾伯勞	B39	0.5 , 24
燕雀目	啄花鳥科	紅胸啄花	B40	3.5 , 22.5
鳩形目	鳩鴿科	紅鳩	B41	0.3 , 15
燕雀目	鶺鴒科	紅嘴黑鶺	B42	1.5 , 16.5
燕雀目	長尾山雀科	紅頭山雀	B43	1.5 , 24
雀形目	燕科	家燕	B44	1 , 16.5
燕雀目	鶺鴒科／鶺鴒亞科	背林鶺鴒	B45	2 , 16.5
鸚鵡目	八哥科	泰國八哥	B46	1.5 , 17.5
雀形目	鴉科	烏鴉	B47	0.5 , 8.5
鳩形目	鳩鴿科	珠頸斑鳩	B48	0.5 , 5.5
雀形目	鸚嘴亞科	粉紅鸚嘴	B49	1.5 , 17
鸛形目	珩科	高蹺珩	B50	1 , 19.5
燕雀目	鶯亞科	深山鶯	B51	2 , 14.5
燕雀目	鶺鴒亞科	野鶺	B52	2 , 15
佛法僧目	翡翠科	魚狗	B53	4.5 , 24
雀形目	文鳥科	麻雀	B54	1 , 19.5
燕雀目	鶯亞科	斑紋鷓鴣	B55	2.5 , 24
鳩形目	鳩鴿科	斑頸鳩	B56	0.3 , 6
燕雀目	伯勞科	棕背伯勞	B57	1 , 21
燕雀目	鶯亞科	棕面鶯	B58	4.5 , 15
燕雀目	鶯科	棕扇尾鶯	B59	2.5 , 24
燕雀目	畫眉科	畫眉	B60	1.5 , 16.5
鴉形目	杜鵑科	番鵲	B61	0.5 , 15
燕雀目	鶺鴒亞科	紫嘯鶺	B62	1.5 , 17
燕雀目	山雀科	黃山雀	B63	1 , 14
燕雀目	鶺鴒亞科	黃尾鶺	B64	5 , 10.5
燕雀目	王鶺科	黃胸青鶺	B65	1 , 23.5
燕雀目	王鶺科	黃腹青鶺	B66	3 , 15.5
鴉形目	鴉科	黃嘴角鴉	B67	1.3 , 4
燕雀目	鶯亞科	黃頭扇尾鶯	B68	1 , 20
燕雀目	王鶺科	黑枕藍鶺	B69	2 , 20.5

鸛形目	鷺科	黑冠麻鷺	B70	0 , 7.5
燕雀目	山雀科	煤山雀	B71	1.5 , 10.5
燕雀目	山雀科	綠背山雀	B72	1 , 20
鷲形目	啄木鳥科	綠啄木	B73	1 , 16
燕雀目	啄花鳥科	綠啄花	B74	2 , 22.5
燕雀目	畫眉科	綠畫眉	B75	1.5 , 21.5
鳩形目	鳩鴿科	翠翼鳩	B76	0.4 , 0.6
鷓鴣目	鷓鴣科	領角鴉	B77	0.5 , 11.5
鷓形目	鷓鴣科	領角鴉	B78	0.5 , 18
燕雀目	雀科	鳴	B79	2 , 24
燕雀目	鷺亞科	褐色叢樹鷺	B80	2 , 20.5
燕雀目	畫眉亞科	褐頭花翼	B81	2.5 , 22.5
燕雀目	鷺亞科	褐頭鷓鷺	B82	1 , 21.5
燕雀目	雀科	褐鷺	B83	1.5 , 20
燕雀目	鴉科	樹鴉	B84	1 , 17
燕雀目	畫眉亞科	頭烏線	B85	1.5 , 17
鷓形目	鷓鴣科	鳩鴿	B86	1 , 12
燕雀目	畫眉亞科	繡眼畫眉	B87	2 , 20
燕雀目	畫眉科	藪鳥	B88	1 , 16
燕雀目	畫眉科	鱗胸鷓鷺	B89	4 , 19.5
燕雀目	鷓鷺科	鷓鷺	B90	3 , 19
燕雀目	鷓科	赤腹鷓	B91	6 , 24

(三) 225 筆鳥鳴聲紋圖資料庫分析圖 10-14(因資料眾多，這裡僅列出 5 筆較為清楚的資料，其他的將收錄在附錄中)。

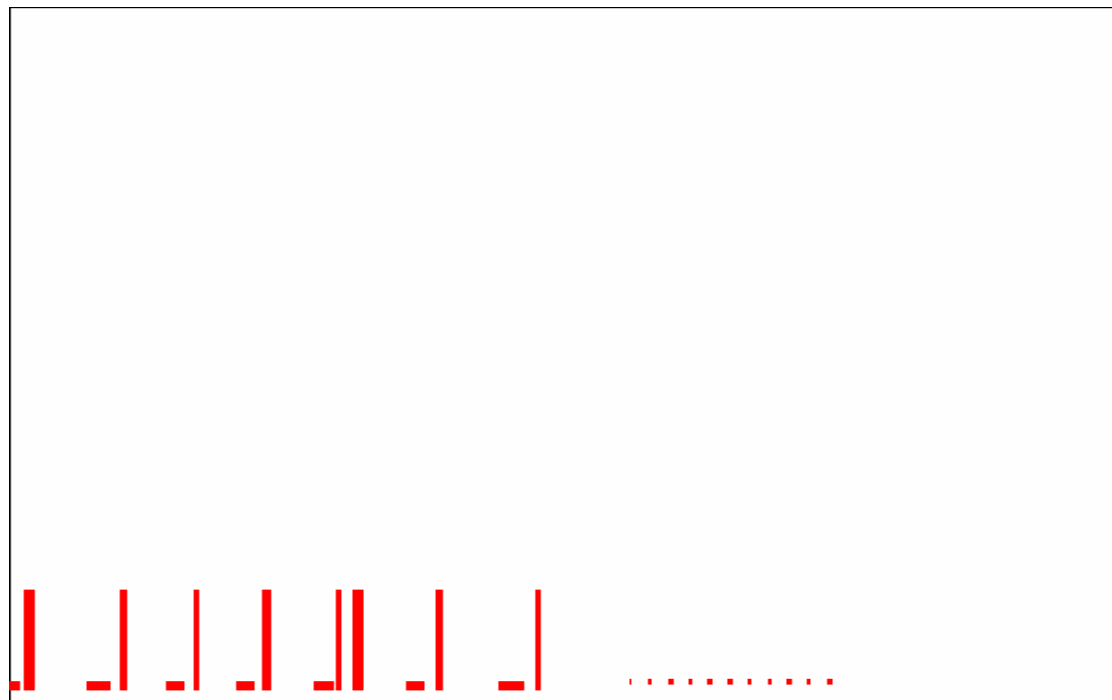
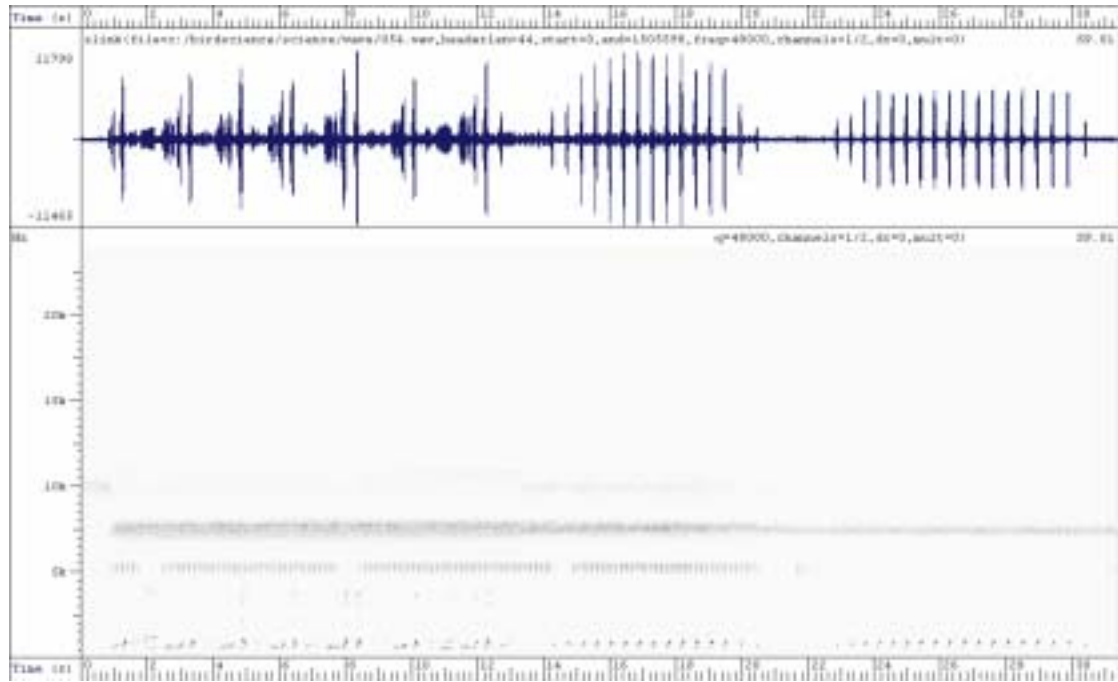
<p>5</p> <p>黑冠麻鷲</p>	
<p>學名：Gorsakius melanolophus</p>	
<p>特徵：額、頭頂及冠羽為黑色，常單獨活動於陰濕的樹林下或溪澗水邊覓食，近距離其叫似為「誤等於誤等於誤等於...」，遠則只聽「誤 誤 誤...」</p>	
	
	

6

白腹秧雞

學名：Amaurornis phoenicurus

特徵：臉面、胸腹白色，體背、頭後翼黑灰色，鳴聲像是「苦啊，苦啊，……」，故在鄉野有“苦雞母”的俗名。

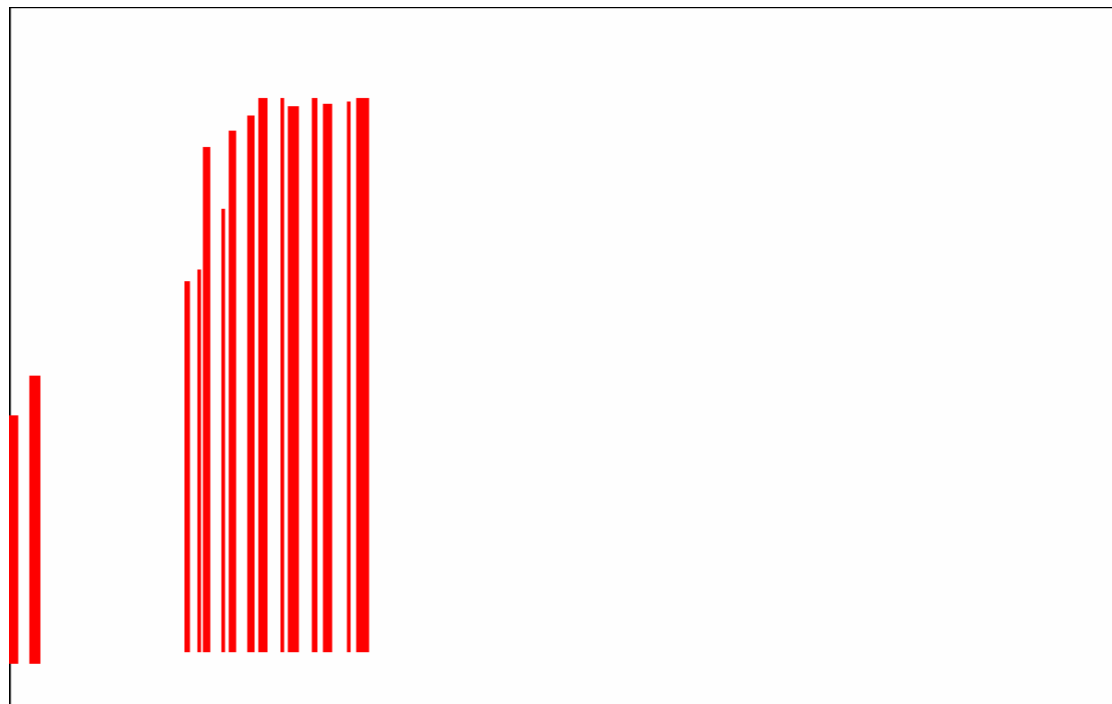
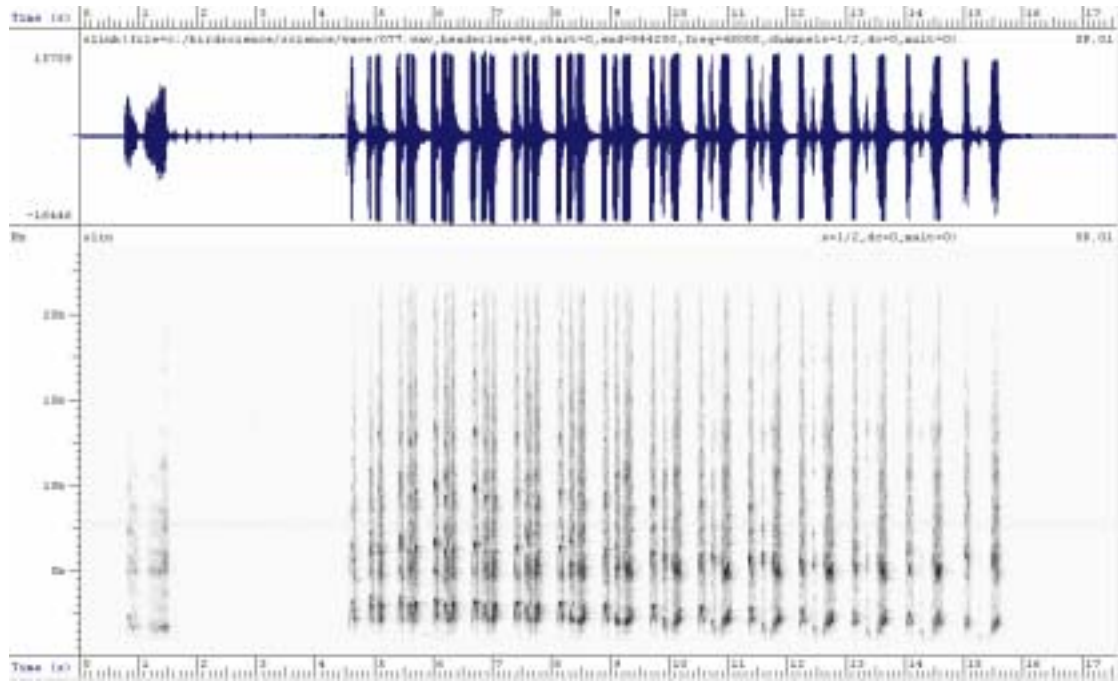


7-1

竹雞

學名：Bambusicola thoracica

特徵：全身大致灰褐色，頭頂有赤褐色斑點，喉部有明顯赤栗色，常三五成群於林緣或林道旁覓食地面昆蟲。鳴聲類似「雞狗乖、雞狗乖」。

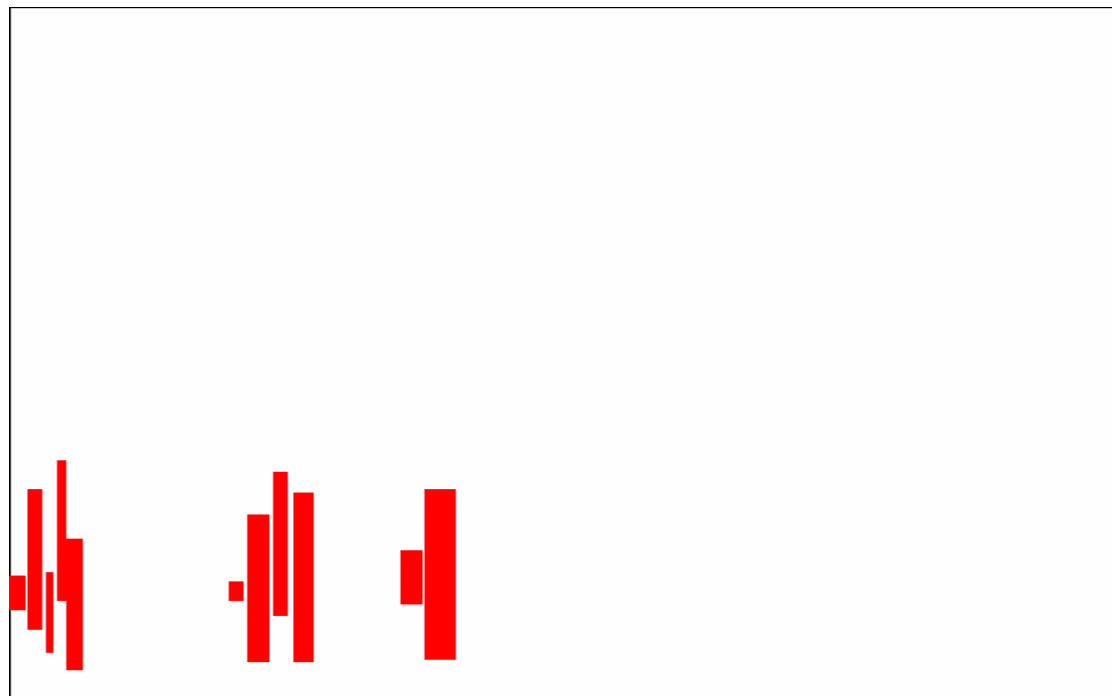
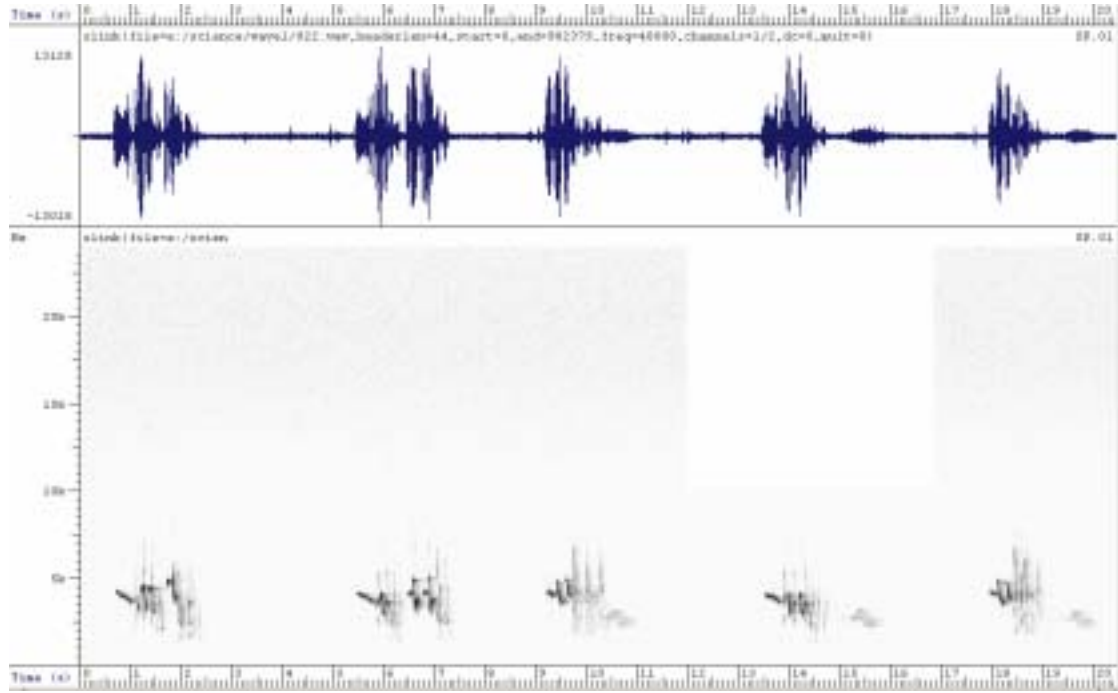


23-3

小翼鶇

學名：Brachypteryx montana

特徵：為台灣特有亞種。眉上有明顯的眉斑，喜棲息於濃密的樹林底層，藏匿性極強，很不容易見其露臉。鳴聲響亮，在近距離可聞其歌鳴。

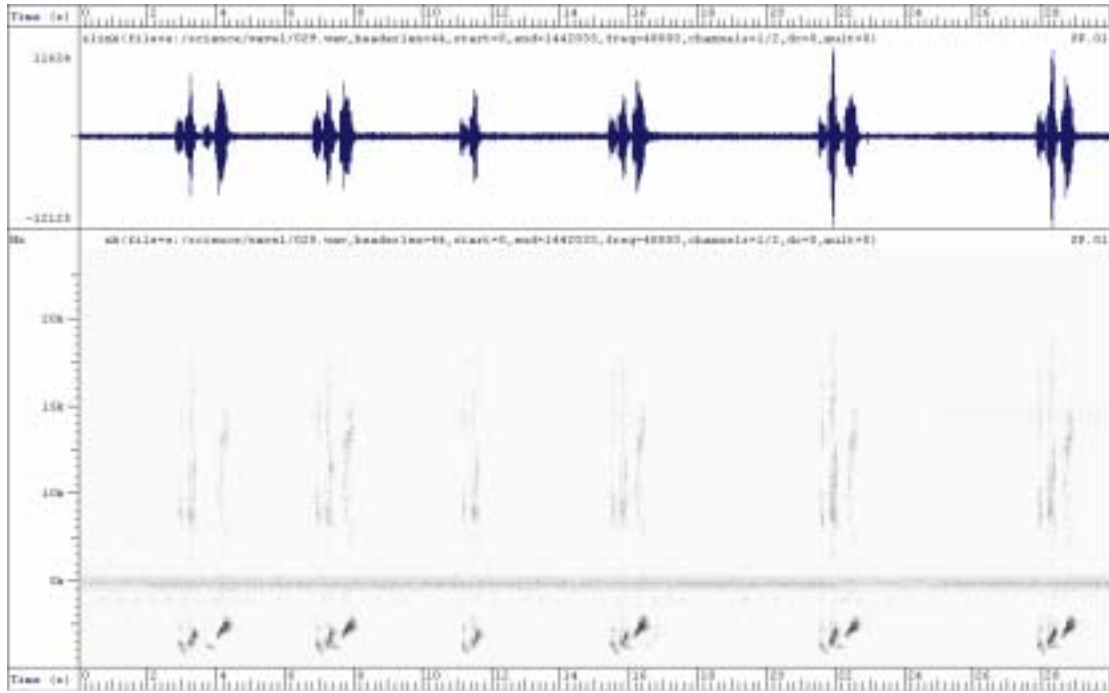


24-6

小彎嘴畫眉

學名：Pomatorhinus ruficollis

特徵：頭上為暗褐色，眉斑白色粗且長，出現在中、低海拔闊葉林中、上層，成小群活動，善鳴叫，叫聲為低沈的「都、都、都」或「嘎歸—嘎歸」。



## 五、結論；

由已知聲紋圖以及時頻圖來建立鳥類聲譜資料庫來比較分析未知的鳥類的方法是可行的，只要聲音清晰而且資料庫完整，一般來說是可以分辨未知鳥類聲音，但若聲音錄製訊號太弱，或是有背景雜音，則需再重錄製更清楚的聲音才能加以判斷。展望未來多加補充錄製各式鳥類資料庫，才可使辨識能力更加提高。

## 六、參考資料：

1. 孫清松，野鳥鳴唱圖鑑
2. C.K.CATCHPOLE, P.J.B.SLATER , *BIRD SONG-Biological themes and variations*, Cambridge University Press, 1995
3. Crawford H. Greenewalt, *BIRD SONG:ACOUSTICS AND PHYSIOLOGY*, Smithsonian Institution Press, 1968
4. C.K. CATCHPOLE and P.J.B. SLATER, *BIRD SONG: Biological themes and variations*, 1995
5. Rosemary Jellis, *Bird Sounds and Their Meaning*, Cornell University Press, 1984
6. [http://ccms.ntu.edu.tw/~ding/aves/chcklst\\_e.htm](http://ccms.ntu.edu.tw/~ding/aves/chcklst_e.htm)
7. [http://archive.zo.ntu.edu.tw/bird/r\\_bird\\_index.asp?bird\\_id=B0370](http://archive.zo.ntu.edu.tw/bird/r_bird_index.asp?bird_id=B0370)

資料庫比對程式碼(1)-(3)

```
(1)//-----BirdSoubd.cpp-----  
#include <vcl.h>  
#pragma hdrstop  
#include "BirdSound.h"  
#include <stdio.h>  
#include "part_function_v1.cpp"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm1::btnUnknownClick(TObject *Sender)
```



```

{
//-----load Bird Name from file-----
    AnsiString str_Line, str_temp, str_check, result_List, str_bird_name;
    TStrings *In_File_Bird_Name;
    In_File_Bird_Name = new TStringList;
    In_File_Bird_Name->LoadFromFile(edtFileName->Text);
    TStrings * list_bird;
    list_bird = new TStringList;

    Index_bird = edtIndex->Text.ToInt(); //取得哪隻鳥

    str_Line = In_File_Bird_Name->Strings[Index_bird-1];
    part_func(str_Line, "\t", list_bird);
    str_bird_name = list_bird->Strings[1];
    edtBirdName->Text = str_bird_name;
    edtSerial->Text = list_bird->Strings[0];
//-----load data from file-----
    TStrings *In_File_List;
    In_File_List = new TStringList;
    TStrings *Out_FileList;
    Out_FileList = new TStringList;
    TStrings * list_items;
    list_items = new TStringList;
    list_items->Clear();
    In_File_List->LoadFromFile("D:\\PHD15\\bcb\\compare_v1\\BirdSound.txt");
    long list_length, list_width ,counter, temp_boxes;

    list_length = In_File_List->Count;

    if(Index_bird > list_length | Index_bird < 1)
    {
        ShowMessage("Error!!! There is no this Index in the file.");
        return;
    }
    else
        edtIndex->Text = Index_bird;

    str_Line = In_File_List->Strings[Index_bird-1];

```

```

part_func(str_Line,"\t",list_items);
list_width = list_items->Count;

float* pt = new float[list_length];
for( long j = 0; j < list_length ; j++ )
    pt[j] = 0 ;

long* lpt = new long[list_length];
for( long j = 0; j < list_length ; j++ )
    lpt[j] = 0;

char buffer[128];
counter =0;
str_temp = "";
//----save this line of data-----
long temp_index;
for( long i = 0; i < list_width ;i++ )
{
    str_check = list_items->Strings[i].TrimRight();
    pt[i]=StrToFloat(str_check);
    temp_index = pt[0];

    if( i == 0 )
    {
        sprintf(buffer,"%-6d ",temp_index);
    }
    if( i == 1 )
    {
        Boxes = pt[1];
        sprintf(buffer,"%-6d ", Boxes);
    }
    else
        sprintf(buffer,"%-7.3f ",pt[i]);
    str_temp += buffer;
}
Out_FileList->Add(str_temp);
list_items->Clear();
Out_FileList->SaveToFile("test_unkown.txt");

```

```

//-----check the list_width-----
if( list_width < 3)
{
    ShowMessage("Error!!! There is no data for this Index.");
    return;
}
else if( list_width != (Boxes *4 + 2) )
{
    ShowMessage("Box number is Error!!! .");
    return;
}
//-----scaling begin from x=0-----
long test;
float begin;
begin = pt[2];
for( long i = 0; i < (list_width - 2)/2 ; i++ )
{
    pt[2*i+2]= pt[2*i+2] - begin;
    test = pt[2*i+2];
}
//-----scaling (x,y)=to image index(25x, 25y)-----
for( long i = 0; i < list_width ; i++ )
{
    if(i < 2)
        lpt[i] = pt[i];
    else
        lpt[i]= long(pt[i]*25);
    test = lpt[i];
}
//-----Draw the image of time-frequence boxes with index -----
Graphics::TBitmap *Bitmap_new = new Graphics::TBitmap;
Bitmap_new->Width = imgTimeFrequence->Width;
Bitmap_new->Height = imgTimeFrequence->Height;
if( Index_bird <= list_length )
    SetImgColor( Bitmap_new, Index_bird, lpt, Boxes);
else
{
    ShowMessage("Index Error!!!");
}

```

```

        return;
    }
    imgTimeFrequency->Canvas->Draw(0, 0, Bitmap_new);
//-----
    delete [] pt;
    delete [] lpt;

    delete In_File_List;
    delete Out_FileList;
    delete list_items;
    ShowMessage("Unknown picture is drawn.");
}
//-----
void __fastcall TForm1::btnCloseClick(TObject *Sender)
{
    Close();
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Index_bird = edtIndex->Text.ToInt();
}
//-----
void __fastcall TForm1::edtIndexChange(TObject *Sender)
{
    Index_bird = edtIndex->Text.ToInt();
}
//-----
void __fastcall TForm1::btnSaveImageClick(TObject *Sender)
{
    SaveDialog1->Title = "不用打副檔名";
    bool Success = SaveDialog1->Execute();
    if(Success == false)
        return;

    AnsiString FileName = SaveDialog1->FileName;

    imgTimeFrequency->Picture->SaveToFile("unknown.bmp");
}

```

```

}
//-----
void __fastcall TForm1::btnImageResetClick(TObject *Sender)
{
//-----Draw the image of time-frequency boxes with index -----
    Graphics::TBitmap *Bitmap_new = new Graphics::TBitmap;
    Bitmap_new->Width = imgTimeFrequency->Width;
    Bitmap_new->Height = imgTimeFrequency->Height;
    TColor ColorBaground = clWhite ;
    TColor ColorTemp = clRed ;
    int width = Bitmap_new->Width;
    int height = Bitmap_new->Height;

    for(int RealX = 0; RealX < width; RealX++)
        for(int RealY = 0; RealY < height; RealY++)
            Bitmap_new->Canvas->Pixels[RealX][RealY] = ColorBaground;
    imgTimeFrequency->Canvas->Draw(0, 0, Bitmap_new);
}
//-----
void __fastcall TForm1::btnCompareClick(TObject *Sender)
{
    TStrings *Out_FileList;
    Out_FileList = new TStringList;
//-----load Bird Name from file-----
    AnsiString str_Line, str_temp, str_check, result_List, str_bird_name;
    char buffer[128];

    str_temp = "";
    float ratio1, ratio2;
//-----get unknown_image-----
    Graphics::TBitmap *Bitmap_unknown = new Graphics::TBitmap;

    Bitmap_unknown->LoadFromFile("D:\\PHD15\\bcb\\compare_v1\\unknown.bmp");
//-----Load Bitmap from files -----
    AnsiString str_file;
    for(int i=1; i<= 225; i++)
    {
        if( i != 32)

```

```

    {
        str_file = "D:\\PHD15\\bcb\\compare_v1\\t\\t";
        str_file += i;
        str_file += ".bmp";
        Graphics::TBitmap *Bitmap_load = new Graphics::TBitmap;
        Bitmap_load->LoadFromFile(str_file);

        ratio1 = fCompare(Bitmap_unknown,Bitmap_load);
        ratio2 = fCompare(Bitmap_load,Bitmap_unknown);
        sprintf(buffer,"%-4d %-6.3f %-6.3f",i, ratio1, ratio2);
        str_temp = buffer;
        Out_FileList->Add(str_temp);
        delete Bitmap_load;
    }
}
Out_FileList->SaveToFile("compare.txt");
//-----
delete Out_FileList;

    ShowMessage("Comparison unknown_picture with data_picture is done.");
}
//-----

```

```

(2)//----BirdSound.h-----
#ifndef BirdSoundH
#define BirdSoundH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <Dialogs.hpp>
//-----
class TForm1 : public TForm
{
    __published: // IDE-managed Components

```

```

        TImage *imgTimeFrequency;
TButton *btnUnknown;
        TEdit *edtFileName;
        TEdit *edtIndex;
        TButton *btnSaveImage;
        TLabel *Label1;
        TButton *btnClose;
TSaveDialog *SaveDialog1;
TButton *btnImageReset;
TLabel *Label2;
TLabel *Label3;
TLabel *Label4;
TLabel *Label5;
TLabel *Label6;
TEdit *edtBirdName;
TLabel *Label7;
TEdit *edtSerial;
TLabel *Label8;
TLabel *Label9;
        TLabel *Label10;
        TLabel *Label11;
        TLabel *Label12;
        TLabel *Label13;
        TLabel *Label14;
        TLabel *Label15;
TButton *btnCompare;
        void __fastcall btnUnknownClick(TObject *Sender);
        void __fastcall btnCloseClick(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall edtIndexChange(TObject *Sender);
void __fastcall btnSaveImageClick(TObject *Sender);
void __fastcall btnImageResetClick(TObject *Sender);
        void __fastcall btnCompareClick(TObject *Sender);
private: // User declarations
public: // User declarations
        long Index_bird;
        long Boxes;
        long Line_length;

```

```

        __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

(3)//-----pParrrt_function.cpp-----
#include <vcl.h>
#include <math.h>
void part_func(AnsiString in_sss, AnsiString deli, TStrings * ret)
{
    int deli_len, len, pos;
    AnsiString part;
    AnsiString sss;
    sss = in_sss.Trim();
    ret->Clear();

    deli_len = deli.Length();

    while( 0 < (pos=sss.Pos( deli )) )
    {
        part = sss.SubString(1,pos-1);
        ret->Add( part.Trim() );

        len = sss.Length();
        sss = sss.SubString(pos+deli_len, len-(pos-1)-deli_len );
        sss = sss.Trim(); //少了這項 空白會當成數字 會出錯
    }
    sss = sss.Trim();
    len = sss.Length();
    if( len > 0 )
        ret->Add( sss );
}
void part_func_fix(AnsiString in_sss, AnsiString deli, TStrings * ret)
{
    int deli_len, len, pos, check_has_deli;

```



```

AnsiString part;
AnsiString sss;
sss = in_sss;
ret->Clear();
check_has_deli = 0;

deli_len = deli.Length();

while( 0 < (pos=sss.Pos( deli )) )
{
    check_has_deli = 1;
    part = sss.SubString(1,pos-1);
    ret->Add( part.Trim() );

    len = sss.Length();
    sss = sss.SubString(pos+deli_len, len-(pos-1)-deli_len );
    sss = sss.Trim(); //少了這項 空白會當成數字 會出錯
}
if(!check_has_deli) //如果 DATA 突然沒有 deli 只能換成 " " 判斷
{
    while( 0 < (pos=sss.Pos(" ")) )
    {
        part = sss.SubString(1,pos-1);

        ret->Add( part.Trim() );

        len = sss.Length();
        sss = sss.SubString(pos+deli_len, len-(pos-1)-deli_len );
        sss = sss.Trim(); //少了這項 空白會當成數字 會出錯
    }
}
sss = sss.Trim();
len = sss.Length();
if( len > 0 )
ret->Add( sss );
/*
int deli_len,len,pos;
AnsiString part[10];

```

```

AnsiString sss;
sss = in_sss;
ret->Clear();

part[0] = sss.SubString(1,6);
part[1] = sss.SubString(9,5);
part[2] = sss.SubString(17,4);
part[3] = sss.SubString(25,2);
part[4] = sss.SubString(33,2);
part[5] = sss.SubString(41,2);
part[6] = sss.SubString(49,2);
part[7] = sss.SubString(57,5);
part[8] = sss.SubString(65,4);
part[9] = sss.SubString(73,4);
// 121.99 23.83 1991 1 1 3 0 51.04 11.8 3.3
for( int i = 0; i < 10 ; i++)
    ret->Add( part[i].Trim() );
*/
}
//-----
/* the example about how to use the part_function

void __fastcall TForm1::btnUsingPartFunctionClick(TObject *Sender)
{
    TStringList * LL;

    LL = new TStringList;
    part_func(Edit1->Text, Edit2->Text, LL);

    Memo1->Lines->Text = LL->Text;
    delete LL;
}
*/
//-----
AnsiString tran_int_2_4space( int k)
{
    AnsiString ret, strtemp;
    ret = AnsiString(k);

```

```

int len;
len = ret.Length();
if(len < 4)
{
    for(int i=1;i<= (4 -len) ;i++)    // 由 1 開始 非 0
    {
        strtemp += ' ';
    }
    strtemp += ret;
}
else
{
    strtemp = ret;
}
return strtemp;
}

//-----
double calculate_ln_slope(double *pX, double *pY, int length)
{
    double sigma_xy = 0;
    double sigma_x = 0;
    double sigma_y = 0;
    double sigma_x2 = 0;
    double n = length;

    for(int i = 0; i < length; i++)
    {
        if(pX[i] != 0 && pY[i] != 0)
        {
            sigma_xy = sigma_xy + log10(pX[i]) * log10(pY[i]);
            sigma_x = sigma_x + log10(pX[i]);
            sigma_y = sigma_y + log10(pY[i]);
            sigma_x2 = sigma_x2 + log10(pX[i]) * log10(pX[i]);
        }
        else
            n = n - 1;
    }
}

```

```

    double slope = 0;
    slope = (n * sigma_xy - sigma_x * sigma_y) / (n * sigma_x2 - sigma_x *
sigma_x);
    return slope;
}
void SetImgColor(Graphics::TBitmap * ptImg, long index, long* lpt, long boxes)
{
    int width = ptImg->Width;
    int height = ptImg->Height;
    bool has_color = false;
    bool x_check, y_check;
    x_check = false;
    y_check = false;
    TColor ColorBaground = clWhite ;
    TColor ColorTemp = clRed ;

    for(int RealX = 0; RealX < width; RealX++)
        for(int RealY = 0; RealY < height; RealY++)
            ptImg->Canvas->Pixels[RealX][RealY] = ColorBaground;
    int counter =0;
//-----畫框-----
    for(int RealX = 0; RealX < width; RealX++)
        for(int RealY = 0; RealY < height; RealY++)
        {
            if(RealY==0)
                ptImg->Canvas->Pixels[RealX][height - RealY - 1] = clBlack;
            if(RealY== (height - 1))
                ptImg->Canvas->Pixels[RealX][height - RealY - 1] = clBlack;
            if(RealX==0)
                ptImg->Canvas->Pixels[RealX][height - RealY - 1] = clBlack;
            if(RealX==(width - 1))
                ptImg->Canvas->Pixels[RealX][height - RealY - 1] = clBlack;
        }
//-----畫 Boxes-----
    for(int i=0; i < boxes; i++)
    {
        for(int RealX = lpt[i*4+2]; RealX < lpt[i*4+4]; RealX++)
            for(int RealY = lpt[i*4+5]; RealY < lpt[i*4+3]; RealY++)

```

```

        ptImg->Canvas->Pixels[RealX][height - RealY -1] = ColorTemp;
    }
}
//-----
float fCompare(Graphics::TBitmap * pt_unknown_Img, Graphics::TBitmap *
pt_data_Img)
{
    int
unknown_Img_height,unknown_Img_width,data_Img_height,data_Img_width;
    int match_counter, mother_box_area;
    float ratio;
    unknown_Img_height = pt_unknown_Img->Height;
    unknown_Img_width = pt_unknown_Img->Width;
    data_Img_height = pt_data_Img->Height;
    data_Img_width = pt_data_Img->Width;
    TColor check_color;
    match_counter = 0;
    mother_box_area =0;
    if((unknown_Img_height != data_Img_height)||
data_Img_width !=
data_Img_width))
        return -1.0;
    else
    {
        for(int y=0; y < unknown_Img_height; y++)
            for(int x=0; x < unknown_Img_width; x++)
                {
                    if(pt_unknown_Img->Canvas->Pixels[x][y]==clRed)
                    {
                        check_color = pt_data_Img->Canvas->Pixels[x][y];
                        if(check_color == clRed)
                            match_counter++;

                            mother_box_area++;
                    }
                }
            ratio = float(match_counter)/float(mother_box_area) ;
    }
    return ratio;
}

```

```

}
/*
//   in this function, x means log(index number); y means log(data)
double calculate_ln_slope(double *data, int length)
{
    double sigma_xy = 0;
    double sigma_x = 0;
    double sigma_y = 0;
    double sigma_x2 = 0;
    double n = length;

    for(int i = 0; i < length; i++)
    {
        if(i != 0 && data[i] != 0)
        {
            sigma_xy = sigma_xy + log(i) * log(data[i]);
            sigma_x = sigma_x + log(i);
            sigma_y = sigma_y + log(data[i]);
            sigma_x2 = sigma_x2 + log(i * i);
        }
        else
            n = n - 1;
    }
    double slope = 0;
    slope = (n * sigma_xy - sigma_x * sigma_y) / (n * sigma_x2 - sigma_x *
sigma_x);
    return slope;
}
//-----
*/

```