報告題名：

即時語音辨識系統

Real-time Speech Recognition System

作者：文偉君

系級：電子工程學系 四甲

學號：D0182686

開課老師：陳冠宏老師

課程名稱：專題研究

開課系所：電子工程學系

開課學年： 104 學年度 第 一 學期

中文摘要

隨者科技業的蓬勃發展，語音辨識一直是眾人關注的議題，其現今的應用涵蓋甚廣，例如：Apple 公司將它拿來製成 Siri；Google 公司將它拿來發展成各國翻譯程式；以及各家科技業者也都應用語音辨識來執行不同的指令。

本系統是由測試者即時錄製一段數字語音並且進行辨識，錄製完後由 Matlab 以「過零率」與「音量大小」偵測一段話裡面的數個音節(syllable)端點後，將音節切割，並交由 HTK(Hidden Markov Model Toolkit)系統將音節轉換為音素(Phone)以並且抽取特徵值。其中，HTK 系統以梅爾倒頻參數法對每個音素截取 39 維(包含差量及差差量)之特徵向量。當 HTK 完成測試者的音訊節取特徵後，再與我們事先交由 HTK 訓練好的特徵隱藏式碼可夫模型範本進行音素(Phone)辨識。辨識音素之後是採用最大似然率決策法，會從音素分群中選擇最接近的音節作為歸類，辨識完成後會完整顯示測試者的數字語音內容。

而研究結果顯示本系統對特定語者辨識比較精準，但對於非特定語者辨識還需要加強辨識率，而本研究對提升非特定語者的辨識率上提出兩項建議：第一項為「分群語者」：將語者進行分群(男、女；長、幼…)後再以不同的分群範本來辨識；第二項為「回饋資料庫」：將辨識錯誤之範本由測試者透過介面改正後傳回建構資料庫。未來之研究面向將以提升辨識率的方面進行。

關鍵字：ATK、Matlab、即時語音辨識、MFCC、資料庫

Abstract

With those booming technology industries, speech recognition has been the subject of attention, which now covers a wide range of applications. Such as: Apple brings it into Siri; Google develops it into translation program with different countries; and various technology companies also apply speech recognition to perform different commands.

In the 4G generations, “Internet of Things” is well known. Through the internet of things, we can save the consumption of human resources. Moreover, it can bring great convenience to our life. As we know, it has a close relationship between networking and speech recognition. This study hopes to learn the speech recognition principle better. So that I can have a deeper understanding about speech recognition technology. Next, I tried different algorithms to understand which the best speech recognition method is. So that users can input digital audio files for real time, and print out the results after identification. Hopefully, it can be used together with internet of things by converting the identification result into operating instructions. The challenging tasks to learn include Matlab programming, understanding its instructions, trying different audio sequence capturing techniques, and identification methods.

Hidden Markov Model Toolkit (HTK) is a portable toolkit to build and manipulate hidden Markov model, which provides tools consist of a set of library modules and the C source codes. The tool is an advanced facility which provides speech analysis, HMM training, testing and results analysis. Both continuous density mixture of Gaussian and discrete distributions can be used to build complex HMM systems software support for HMM. The HTK release contains extensive documentation and examples. HTK is mainly used for speech recognition as well as many other applications, including research speech synthesis, character recognition, and DNA sequencing. HTK is commonly used worldwide.

The system is composed of users’ instant record digital audio clips and show identification results. After Recording, Matlab starts to use both "Zero Crossing Rate" and "Volume" to detect the number of syllables endpoint which inside passage and cut them. Then, HTK system extracts these syllables feature values. After that, with the features we can train the phonemes HMM models for identifying.

I use maximum likelihood decision method to realize the identification of the phonemes in which I select the closest syllable from the phoneme group. Finally, the system prints out the results of the identification of numbers what the users just say.

The study shows that the system performs more accurate for identifying particular speakers. But, for recognizing general speakers needs to strengthen the recognition rate. As a result, this study presents two proposals to enhance the general

speaker recognition rate. The first one is "grouping speakers" that grouped speakers in terms of male, female, elder, young and so on to identify with the different model. The second one is "feedback library" that transmits the identification error of the model by the user through the interface and then the database can be corrected. I expect this will enhance the recognition performance.

Keywords: ATK, Matlab, Real-time, Speech recognition, MFCC, Databases

目　次

一、緒論

(一) 前言

隨者科技業的蓬發展，語音辨識一直是眾人關注的議題，其現今的應用涵蓋甚廣，例如：Apple 公司將它拿來製成 Siri；Google 公司將它拿來發展成各國翻譯程式；以及各家科技業者也都應用語音辨識來執行不同的指令。

本研究希望藉著了解語音辨識原理，對於語音辨識有更深層的概念，並且嘗試不同的運算法來了解何謂最佳語音辨識方法

(二) 研究動機

在 4G 世代中，物連網是一個家喻戶曉的名詞，透過物聯網，我們可以透過計算機節省人力資源的耗用，而物聯網更是與語音辨識間為密不可分的關係，本篇希望藉由了解即時的數字語音辨識技術，令使用者能及時輸入數字音訊檔，並且辨識後能將結果轉換成運作指令，並且未來能搭配物連網來使用。

未來希望能將語音直接轉換為能夠直接操作的數據，有望讓物聯網開發者取得更多資料、為每一個用戶量身打造專屬需求，並進一步拓展人工智慧領域。

語音辨識指電腦系統對於人類語言理解能力的技術，讓人機互動以最自然的對話方式進行。企業藉由語音辨識可為客戶提供最佳及最有效率的服務品質，同時也能降低成本，提升競爭力。而家庭可以依靠語音辨識技術讓不同模組的電源開關依據不同的指令而運作，打造快捷且舒適的居住空間。

本研究選擇以語音辨識的方式是因為語音辨識技術相對於其他辨識技術而言成本相對較低，而且對於使用者也較容易上手，因此以此著手。

(三) 研究方法

本研究所使用的工具與技術：

1. 使用技術：
   - A. 過零率與音量偵測端點
   - B. 梅爾倒頻譜參數(MFCC)
   - C. 高斯混合模型(Gaussian Mixture Model)
   - D. 最大似然率決策法(Maximum Likelihood Decision)
   - E. 隱藏式馬可夫模型(Hidden Markov Model)
2. 使用軟體
   - A. Matlab
   - B. HTK(Hidden Markov Model Toolkit)

(四) 本系統概述

本系統為測試者現場進行一段錄音之後，以端點偵測並且針對端點分割錄音檔之音節，接著用 HTK 檔提取 MFCC 特徵參數後，與先前訓練的資料庫 MFCC 模型進行個別比對，用最大似然率決策法決定最後的輸出值，最後用數字輸出辨識結果

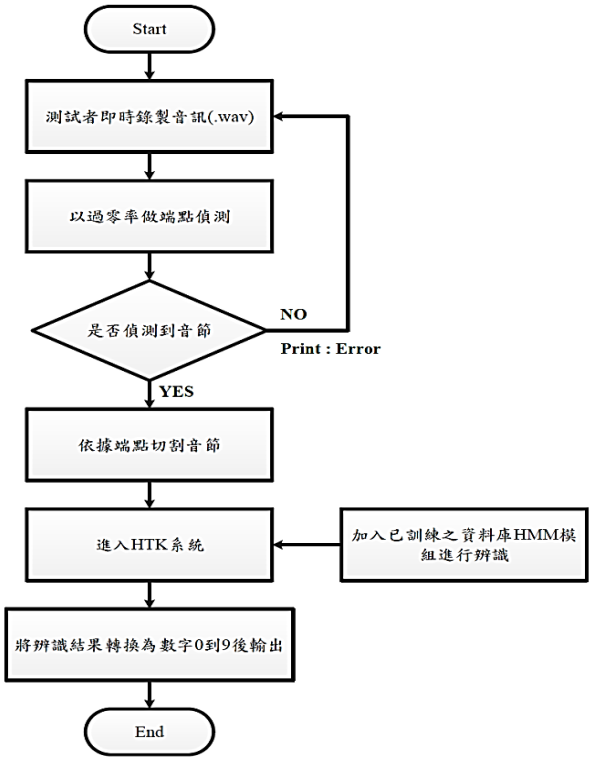


圖 1 系統總流程圖

(五) 文獻探討

HTK(Hidden Markov Model Toolkit)為英國劍橋大學機器智能實驗室所開發的開放原始碼(open source)免費軟體，主要用在語音辨識的研究上，可以在 Linux/Unix、Windows 等平台上運作。 HTK 最主要的核心為隱藏式馬可夫模型(HMM) ，是由 C 語言所撰寫，應用於語音辨識等領 域，可使用連續混合高斯與離散分布，建置複雜的 HMM 模型。 HTK 工具提供語音辨識中包含了資料準備工具、模型訓練工具、辨識工具以及分析 工具。下圖為 HTK 的處理流程。

(六) HTK 處理階段及流程

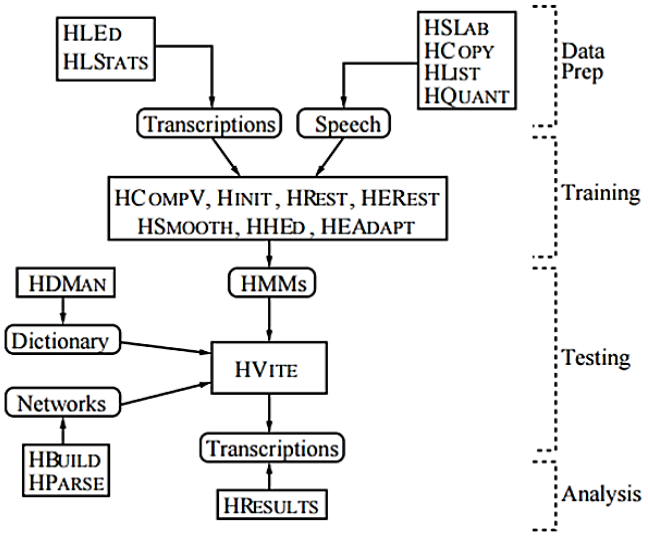


圖 1 HTK 系統總流程(源自：HTK 官方手冊)

二、研究技術介紹

(一) 過零率(Zero Crossing Rate)

在每個音框中，音訊通過零點的次數，具有下列特性：

1. 一般而言，雜訊及氣音的過零率均大於有聲音（具有清晰可辨之音高，例如母音）。
2. 而雜訊和氣音兩者較難從過零率來分辨，會依照錄音情況及環境雜訊而互有高低。但通常氣音的音量會大於雜訊。
3. 通常用在端點偵測，特別是用在估測氣音的啟始位置及結束位置。

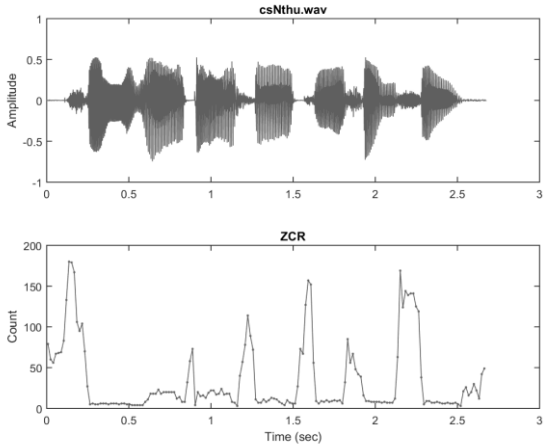


圖 2 過零率示意圖(源自：張志星語音辨識教學網站)

(二) 特徵擷取參數

由於語音訊號的資料量非常龐大，因此必須要從語音的特性、特徵，求取適當的特徵參數，以進行比對辨識。語音訊號的特徵會跟著時域作急遽的改變，但是在頻域中頻譜並不會隨著時間的改變而有急遽的變化，故頻譜具有短時距穩定的特性。利用此性質，我們可以將語音訊號分割成一串連續的音框(frame)，並對每一個音框求取特徵參數。

本專題的 HTK 系統選用梅爾倒頻譜係數作為提取特徵參數的方式。

1. 預強調 (Pre-emphasis)

將語音訊號 s(n) 通過一個高通濾波器：

$$H(z) = 1 - a * z - 1$$

其中 a 介於 0.9 和 1.0 之間。若以時域的運算式來表示，預強調後的訊號 $s_2(n)$ 為：

$$s_2(n) = s(n) - a * s(n-1)$$

這個目的就是為了消除發聲過程中聲帶和嘴唇的效應，來補償語音信號受到發音系統所壓抑的高頻部分。（另一種說法則是要突顯在高頻的共振峰。）

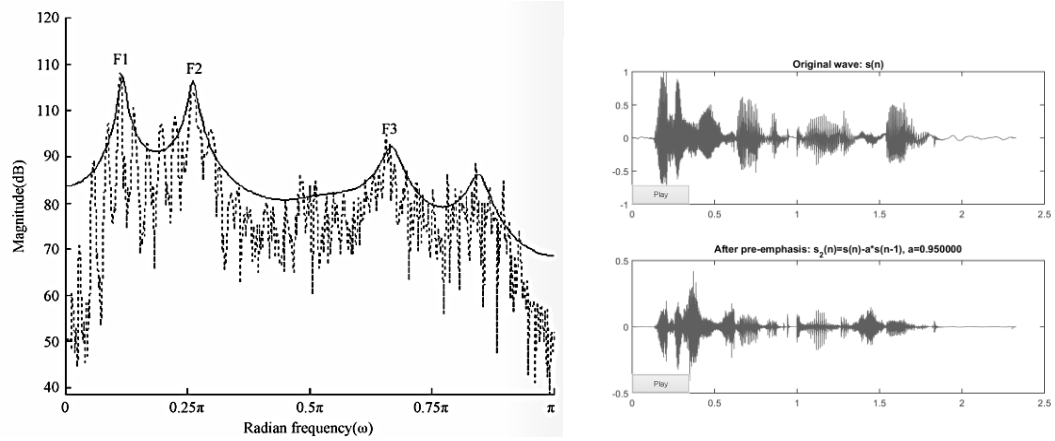


圖 3 預強調示意圖(源自：張志星語音辨識教學網站)

2. 音框化 (Frame blocking)

先將 N 個取樣點集合成一個觀測單位，稱為音框（Frame），通常 N 的值是 256 或 512，涵蓋的時間約為 20~30ms 左右。為了避免相鄰兩音框的變化過大，所以我們會讓兩相鄰因框之間有一段重疊區域，此重疊區域包含了 M 個取樣點，通常 M 的值約是 N 的一半或 1/3。通常語音辨識所用的音訊的取樣頻率為 8 KHz 或 16 KHz，以 8 KHz 來說，若音框長度為 256 個取樣點，則對應的時間長度是 256/8000*1000 = 32 ms。

3. 漢明窗 (Hamming windows)

將每一個音框乘上漢明窗，以增加音框左端和右端的連續性（請見下一個步驟的說明）。假設音框化的訊號為 S(n), n = 0,…N-1。那麼乘上漢明窗後為 $S'(n) = S(n) * W(n)$，此 W(n) 形式如下：

$$W(n, a) = (1 - a) - a\cos(\frac{2pn}{N-1})$$

$$0 \leqq n \leqq N-1$$

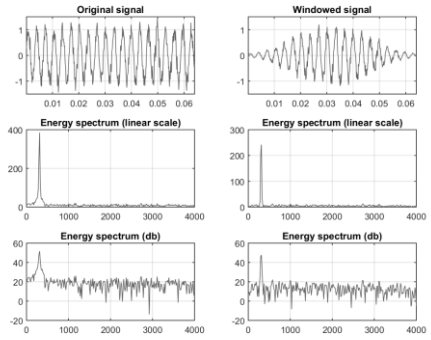
不同的 a 值會產生不同的漢明窗，一般我們都取 a =0.46。

4. 快速傅立葉轉換 (Fast Fourier Transform, FFT)

由於訊號在時域（Time domain）上的變化通常很難看出訊號的特性，所以通常將它轉換成頻域（Frequency domain）上的能量分佈來觀察，不同的能量分佈，就能代表不同語音的特性。所以在乘上漢明窗後，每個音框還必需再經過 FFT 以得到在頻譜上的能量分佈。

乘上漢明窗的主要目的，是要加強音框左端和右端的連續性，這是因為在進行 FFT 時，都是假設一個音框內的訊號是代表一個週期性訊號，如果這個週期性不存在，FFT 會為了要符合左右端不連續的變化，而產生一些不存在原訊號的能量分佈，造成分析上的誤差。

FFT 的運算原理為利用音訊的對稱性以及週期性，降低對離散型傅立葉轉換的複雜度



$$Y(k)=\begin{cases}\sum_{n=n_1}^{n_2} y(n)e^{-j\frac{2\pi}{n_2}kn}, n_1 \le k \le n_2 \\ 0, otherwise\end{cases}$$

圖 4 快速傅立葉轉換示意圖

5. 三角帶通濾波器 (Triangular bandpass filter)

將能量頻譜能量乘以一組 20 個三角帶通濾波器，求得每一個濾波器輸出的對數能量（Log Energy）。必須注意的是：這 20 個三角帶通濾波器在「梅爾頻率」(Mel Frequency)上是平均分佈的，而梅爾頻率和一般頻率 f 的關係式如下：

$$mel(f) = 2595 * log10(1 + \frac{f}{700})$$

梅爾頻率代表一般人耳對於頻率的感受度，由此也可以看出人耳對於頻率 f 的感受是呈對數變化的：

a. 在低頻部分，人耳感受是比較敏銳

b. 在高頻部分，人耳的感受就會越來越粗糙

三角帶通濾波器有兩個主要目的：

a. 對頻譜進行平滑化，並消除諧波的作用，突顯原先語音的共振峰。

b. 降低資料量

因此一段語音的音調或音高，是不會呈現在 MFCC 參數內，換句話說，以 MFCC 為特徵的語音辨識系統，並不會受到輸入語音的音調不同而有所影響。

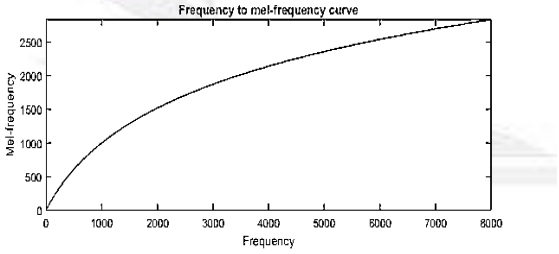


圖 5 梅爾頻率參照圖(源自：張志星 語音辨識教學網站)

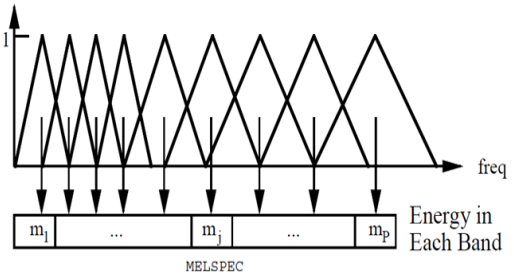


圖 6 梅爾頻率與能量對照圖(源自：HTK 官方手冊)

6. 離散餘弦轉換（Discrete cosine transform, or DCT）

將上述的 20 個對數能量 m_j帶入離散餘弦轉換，求出 N 階的 Mel- scale Cepstrum 參數，這裡 N 通常取 12。離散餘弦轉換公式如下：

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos\left(\frac{\pi i}{N}(j - 0.5)\right)$$

其中 Ek 是由前一個步驟所算出來的三角濾波器和頻譜能量的內積值，N 是三角濾波器的個數。由於之前作了 FFT，所以採用 DCT 轉換是期望能轉回類似 Time Domain 的情況來看，又稱 Quefrency Domain，其實也就是 Cepstrum(倒頻譜)。又因為之前採用 Mel- Frequency 來轉換至梅爾頻率，所以才稱之 Mel-scale Cepstrum。

目的：為了避免頻率越高的率波器寬度越大，造成高頻帶的能量被放大，因此以 DCT 運算式讓能量隨濾波器的寬度增加而能量減小，使其正規化。

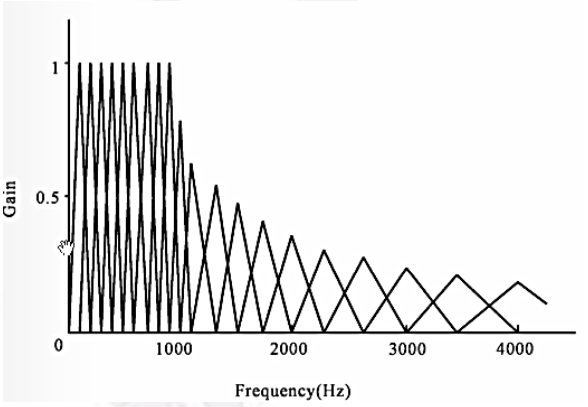


圖 7 DCT 轉換後增益圖(源自：張志星 語音辨識教學網站)

7. 對數能量（Log energy）

一個音框的音量（即能量），也是語音的重要特徵，而且非常容易計算。因此我們通常再加上一個音框的對數能量（定義為一個音框內訊號的平方和，再取以 10 為底的對數值，再乘以 10），使得每一個音框基本的語音特徵就有 13 維，包含了 1 個對數能量和 12 個倒頻譜參數。

8. 差量倒頻譜參數（Delta cepstrum）

雖然已經求出 13 個特徵參數，然而在實際應用於語音辨識時，我們通常會再加上差量倒頻譜參數，以顯示倒頻譜參數對時間的變化。它的意義為倒頻譜參數相對於時間的斜率，也就是代表倒頻譜參數在時間上的動態變化，公式如下：

$$\Delta\, Cm(t) = \sum_{\tau=-M}^{M} \frac{C(t+\tau)\tau}{\tau^2}$$

這裡 M 的值一般是取 2 或 3。因此，如果加上差量運算，就會產生 26 維的特徵向量；如果再加上差差量運算，就會產生 39 維的特徵向量。本研究使用 39 維的特徵向量。

(三) 馬可夫模型(Zero Crossing Rate)

當一個隨機過程在給定現在狀態及所有過去狀態情況下，其未來狀態的條件機率分布僅依賴於當前狀態；換句話說，在給定現在狀態時，它與過去狀態（即該過程的歷史路徑）是條件獨立的，那麼此隨機過程即具有馬可夫性質。具有馬可夫性質的過程通常稱之為馬可夫過程。

1. 隱藏式馬可夫模型 Hidden Markov Model (HMM)

隱馬爾可夫模型是統計模型，它用來描述一個含有隱含未知參數的馬爾可夫過程。其難點是從可觀察的參數中確定該過程的隱含參數。然後利用這些參數來作進一步的分析，例如模式識別。

在正常的馬爾可夫模型中，狀態對於觀察者來說是直接可見的。這樣狀態的轉換機率便是全部的參數。而在隱馬爾可夫模型中,狀態並不是直接可見的，但受狀態影響的某些變量則是可見的。每一個狀態在可能輸出的符號上都有一機率分布。因此輸出符號的序列能夠透露出狀態序列的一些信息。

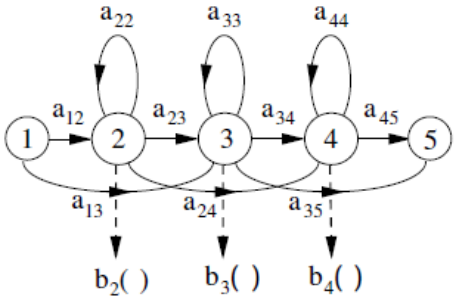


圖 8 馬可夫模型概念圖(源自：wiki)

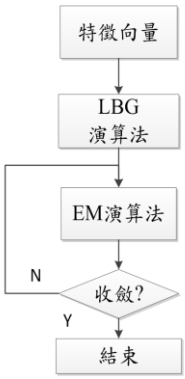


圖 9 GMM 流程圖

2. 高斯混合模型 Gaussian Mixture Model (GMM)

高斯混合模型是語音信號處理中的一種常用的統計模型，該模型的一個基本理論前提是只要高斯混合的數目足夠多，一個任意的分布就可以在任意的精度下用這些高斯混合的加權平均來逼近。

一個包含 M 個分量的高斯混合分布的機率密度函數是 M 個高斯機率密度分布函數的加權組合，定義為

$$P(x|\lambda) = \sum_{i}^{M} \omega_i p_i(x)$$

其中的X是 D 維隨機矢量，$p_i(x), i = 1,2,3 \dots M$為 M 個機率密度函數分量，$\omega_i, i = 1,2, \dots, M$為各個機率密度函數分量的權重。

GMM 的參數估計方法有多種方法，其中應用最廣泛的是基於最大似然準則(Maximum Likelihood Estimation, MLE)的方法。

最大似然估計的主要思想就是要找到使得 GMM 模型對於訓練語料的似然度最大的模型參數λ

三、研究處理流程

(一) HTK 應用於本系統建構資料庫之流程

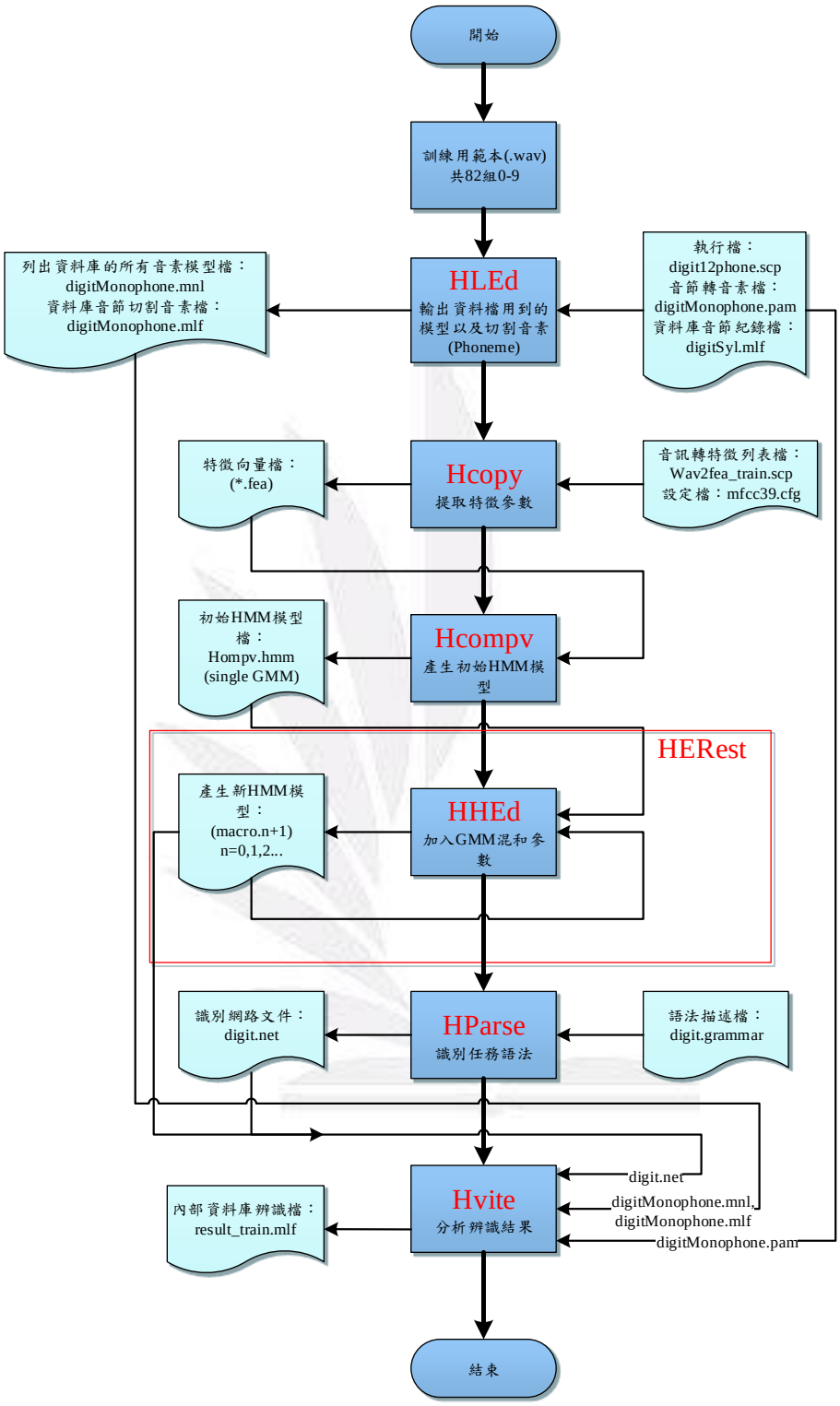


圖 10 資料庫建構流程

(二) HTK 應用於本系統識別測試音訊流程

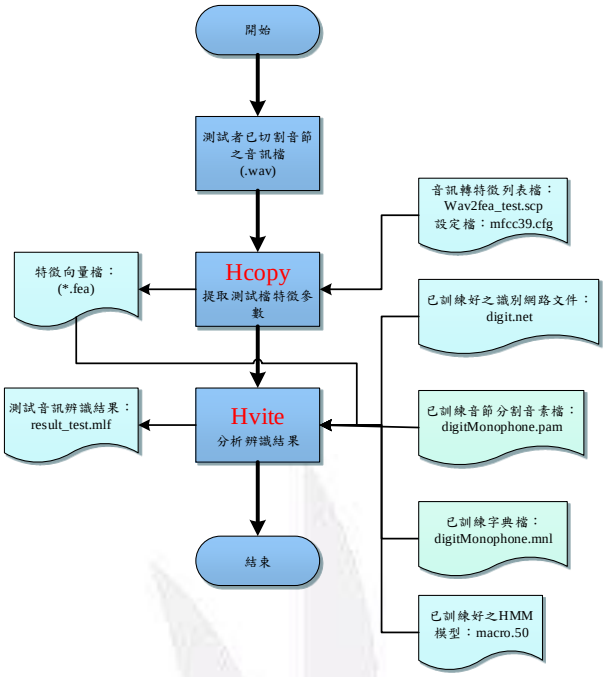


圖 11 音訊辨識流程

1. 即時錄音

程式碼

```
fs=96000;   % 取樣頻率
nBits=24;   % 取樣點解析度，必須是 8 或 16 或 24
nChannel=1; % 聲道個數，必須是 1（單聲道） 或 2（雙聲道或立體音）
            % duration錄音時間（秒）

recObj=audiorecorder(fs, nBits, nChannel);
fprintf('\t按任意鍵後開始錄音：（本系統僅能測試數字0~9)\n'); pause; tic;
record(recObj);
fprintf('\t錄音中，按任意鍵後結束錄音\n'); pause;
stop(recObj);
fprintf('\t錄音結束共計%f秒\n',toc);
myRecording = getaudiodata(recObj, 'double');   % get data as a double array
Coordinate=plot((1:length(myRecording))/fs, myRecording); xlabel('Time (sec)'); ylabel('Amplitude');
audioFile='myRecording.wav';    % 欲儲存的 wav 檔案
fprintf('Saving to %s...\n', audioFile);
audiowrite(audioFile, myRecording, fs);
```

圖 12 即時錄音程式碼

命令欄呈現

```
>> main_test
    按任意鍵後開始錄音：（本系統僅能測試數字0~9)
    錄音中，按任意鍵後結束錄音
    錄音結束共計4.216830秒
Saving to myRecording.wav...
```

圖 13 命令欄呈現

2. 端點偵測及音節輸出

程式碼

```
%端點偵測
waveFile='myRecording.wav';
au=myAudioRead(waveFile);
opt=endPointDetect('defaultOpt');
opt.method='vol';
showPlot = 1;
[epInSampleIndex, epInFrameIndex, soundSegment] = endPointDetect(au, opt, showPlot);

%切割錄音檔
for i=1:length(soundSegment)
    audioFile=['output\soundscut\myRecordingcut',int2str(i-1),'.wav'];  % 欲儲存的 wav 檔案
    fprintf('Saving to %s...\n', audioFile);
    [y,fs]=audioread(waveFile, [soundSegment(i).beginSample-5000 soundSegment(i).endSample+8000]);
    audiowrite(audioFile, y, fs);
```

圖 14 端點偵測及輸出程式碼

端點偵測示圖

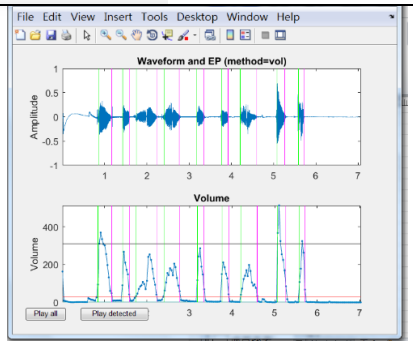


圖 156

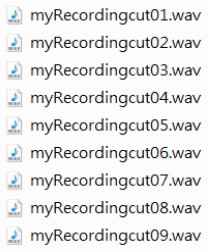
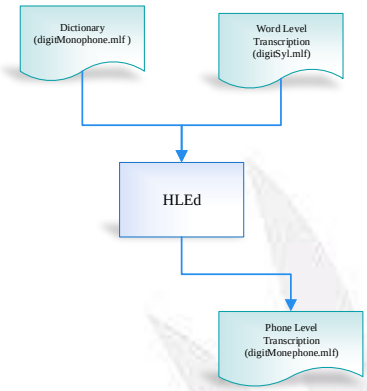
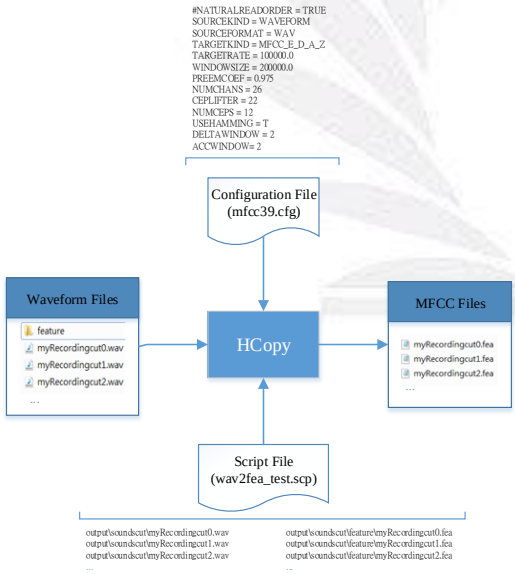


圖 16 端點偵測後的切割檔案

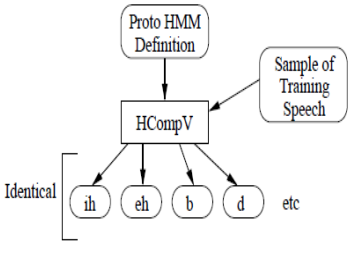
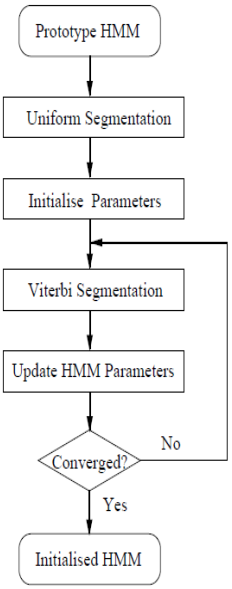
3. HELd

流程圖	語法
 圖 18	HLEd -n output\digitMonophone.mnl -d digitMonophone.pam -l * -i output\digitMonophone.mlf output\sy36phone.scp digitSyl.mlf

4. HCopy

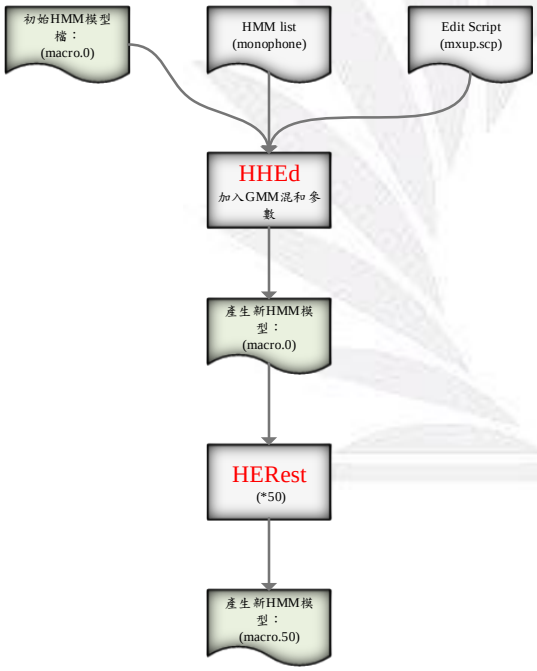
流程圖	語法
 圖 19	vHCopy -C mfcc39.cfg -S output\wav2fea_train.scp HCopy -C mfcc39.cfg -S output\wav2fea_test.scp

5. HCompV

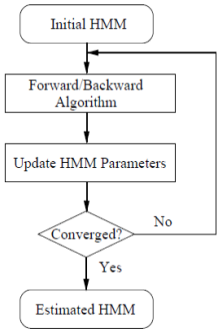
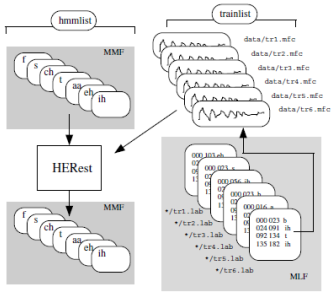
流程圖	語法
 圖 20  圖 21	HCompV -m -o hcompv.hmm -M output -I output\digitMonophone.mlf -S output\trainFea.scp output\template.hmm

6. HHEd

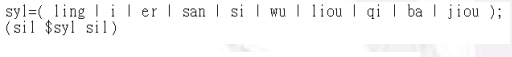
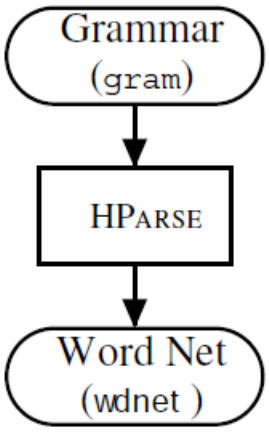
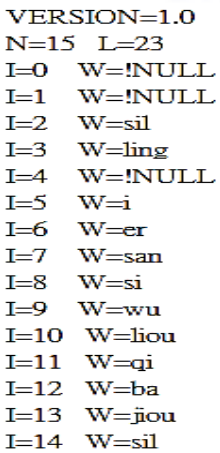
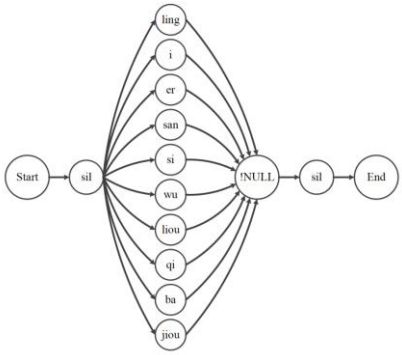
流程圖	語法

 圖 22	fprintf(' \tCreate more Gaussian components to have macro.0\n');
	fid=fopen('output\mxup.scp', 'w'); fprintf(fid, 'MU 3 {*.state[2-4].mix}'); fclose(fid);
	copyfile('output/macro.init', 'output/hmm/macro.0');
	cmd='HHEd -H output\hmm\macro.0 output\mxup.scp output\digitMonophone.mnl';
	dos(cmd);

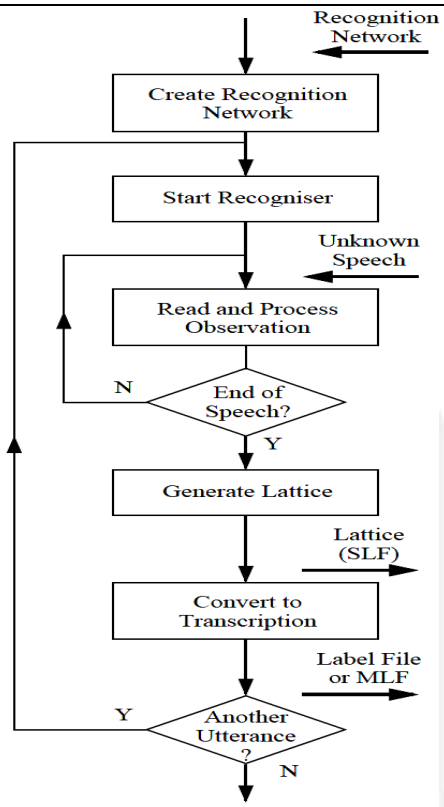
7. HRest

流程圖	語法
 圖 24  圖 23	

8. HParse

流程圖	語法
   圖 25	Hparse digit.grammar output\digit.net  圖 17 HParse 辨識步驟

9. HVite

流程圖	語法
 圖 27	HVite -H %s -l * -i output\\result_train.mlf -w output\\digit.net -S output\\trainFea.scp digitMonophone.pam output\\digitMonophone.mnl', targetMacro

(三) 辨識結果

1. 以數字取代字串辨識結果的程式碼：

```
temp = strrep(c','qi','7');
temp = strrep(temp,'liou','6');
temp = strrep(temp,'jiou','9');
temp = strrep(temp,'ling','0');
temp = strrep(temp,'si','4');
temp = strrep(temp,'i','1');
temp = strrep(temp,'er','2');
temp = strrep(temp,'san','3');
temp = strrep(temp,'wu','5');
temp = strrep(temp,'ba','8');
disp(temp);
```

圖 18 字串轉數字之程式碼

2. 實際測試情況

a. 測試數字：

1 、2 、3 、4 、5 、6 、7 、8 、9 、0

b. 測試結果：

```
output\hmm\macro.50辨識率測試(Outside test) 產生 result_test.mlf    辨識結果為
'1'    '4'    '1'    '4'    '0'    '0'    '8'    '1'    '4'    '2'
```

圖 19

（四） 辨識率測試(我)

次數	辨識結果
1.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '8' '4' '2' '5' '5' '0' '0' '8' '8' '0'
2.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '8' '8' '0' '8' '5' '0' '2' '8' '8' '0'
3.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '0' '8' '2' '4' '4' '2' '0' '6' '8' '4'
4.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '0' '2' '2' '2' '7' '0' '2' '7' '8' '0'
5.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '0' '5' '2' '4' '7' '5' '0' '1' '2' '0'
6.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '4' '4' '2' '2' '4' '6' '1' '4' '8' '0'
7.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '1' '5' '2' '5' '4' '6' '0' '8' '8' '2'
8.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '8' '8' '0' '4' '8' '2' '2' '7' '8' '0'
9.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '0' '6' '0' '8' '4' '3' '4' '8' '8' '0'
10.	output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為 '7' '3' '8' '4' '4' '5' '0' '7' '8' '4'

圖 30

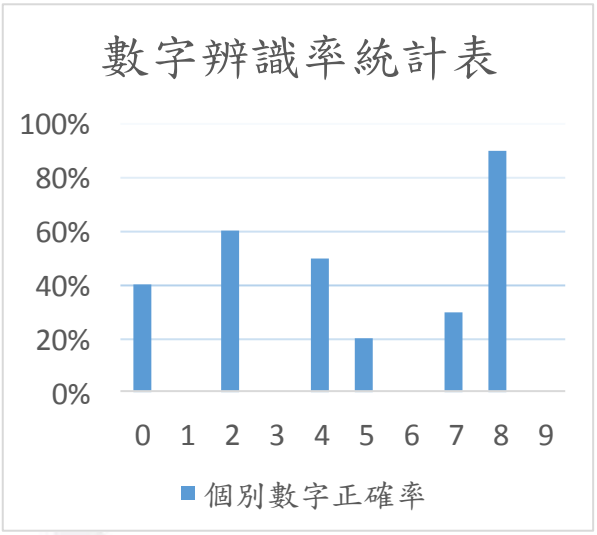


圖 31

四、研究結果

(一) 識率的額外測試(針對特定語者)

output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為
'0' '1' '2' '3' '4' '5' '6' '7' '8' '9'

output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為
'6' '1' '2' '3' '4' '5' '6' '1' '8' '9'

output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為
'0' '3' '2' '3' '4' '5' '6' '7' '8' '9'

output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為
'0' '3' '2' '3' '4' '5' '6' '7' '8' '9'

output\hnn\macro.50辨識率測試(Outside test) 產生 result_test.mlf 辨識結果為
'0' '7' '2' '3' '4' '5' '6' '1' '8' '9'

圖 32

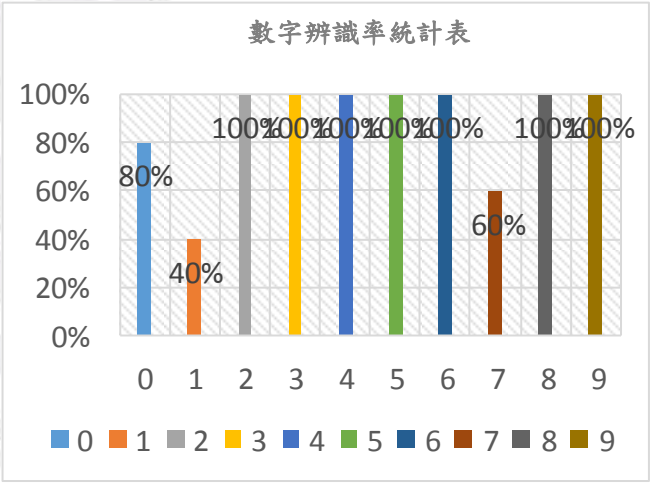


圖 33

(二) 研究結論

若改用特定語者資料做特定語者辨識(也就是訓練資料只用測試者的語音資料做測試者的語音辨識)，發現辨識率可高達 80%以上！！此現象可表示為先前非特定者的資料檔案模組不夠多無法算出更精確的數字特徵植；因此系統可以改為往特定語者即時語音辨識做研究等。

五、遭遇困難及解決方法

1. 端點偵測當音節出現抖音時，可能視為兩個以上音節輸出
   →未來可研究更進階之端點偵測技巧以及加入濾波器改善
2. 音節辨識度不高→轉為音素
3. 超過十個音節會因為檔案放置問題而輸出錯誤順序(檔案 1 與檔案 10 會排在一起)→加上 a=sprintf('%02d',i);使其數字前面補零
4. 針對廣泛語者：
   A. 數字 1 有極大機率辨識成 8
   B. 數字 3 有極大機率辨識成 4 跟 8
   C. 數字 6 有極大機率辨識成 2
   D. 數字 9 有極大機率辨識成 0
5. 針對特定語者
   A. 數字 0 有可能辨識成 6
   B. 數字 1 有可能辨識成 7
   C. 數字 7 有可能辨識成 1

六、心得

因為第一次接觸語音辨識以及學習 Matlab 軟體的使用，因此很多不懂的地方是在網路查詢各大論壇及網站，一個個慢慢查才把所有程式碼湊完成，過程歷經千辛，但也從中學到很多，並且也因為有所付出而在成果上獲得成就感。

從各種面向學習語音辨識技術，再從中挑選適合本研究的方法，雖然在寫程式碼的過程中常常會出現程式錯誤，在網路上尋找的論文及程式碼也會有看不懂的地方，但是還好在指導教授與學長熱心的指點之下能順利完成本專題內容。

希望未來可以結合本專題所學，加以改良後除了提高辨識率之外，更能結合其他語音的辨識，並且搭配指令使用，讓本語音辨識系統能更純熟，更能有效的運用在人類的生活需求上。

七、未來發展面向

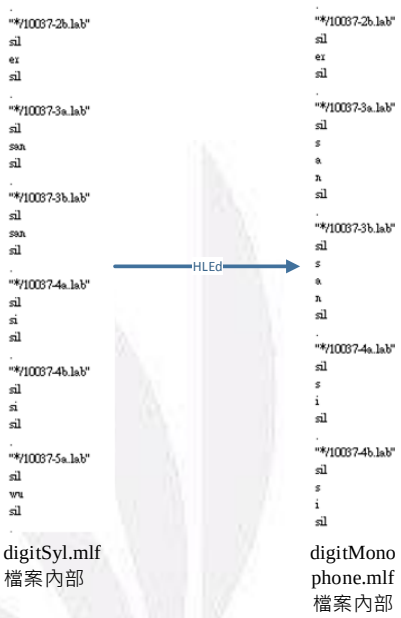
1. 更新與分群資料庫：本研究為用內建而無法更新的資料庫對即時未分群測試資料做比對，因此沒有很高的辨識率。若是能使用分群資料庫(例如男、女)方式讓測試者選擇指定資料庫，且能夠去建立自動更新資料庫，讓測試資料可以經過測試者改正音節之方式更新訓練資料庫，預期將提高辨識程度。
2. 圖形使用者介面 (Graphics User Interface, GUI) 可將 MATLAB 軟體中的程式轉換成 GUI 的介面，而 GUI 是一種以圖形化為基礎的使用者介面，利用統一的圖形與操作方式，如可移動的視窗、選項與滑鼠游標，作為使用者與作業系統之間的對話介面。 設計得當的圖形畫面得以幫助使用者快速了解與尋找功能，且透過統一的操作方式，讓使用者在學習使用一次後，即可順

利使用本研究程式。

3. 與其他系統(例如:物聯網)指令做結合：隨著物聯網興起，本研究可讓使用者設定數字 0-9 之模式內容，並與家用產品做搭配，又或者可增加語言辨識能力來做更多實用性的結合，使得人類能生活在更方便的科技圈。

八、 附錄

一、HELd 轉換流程圖

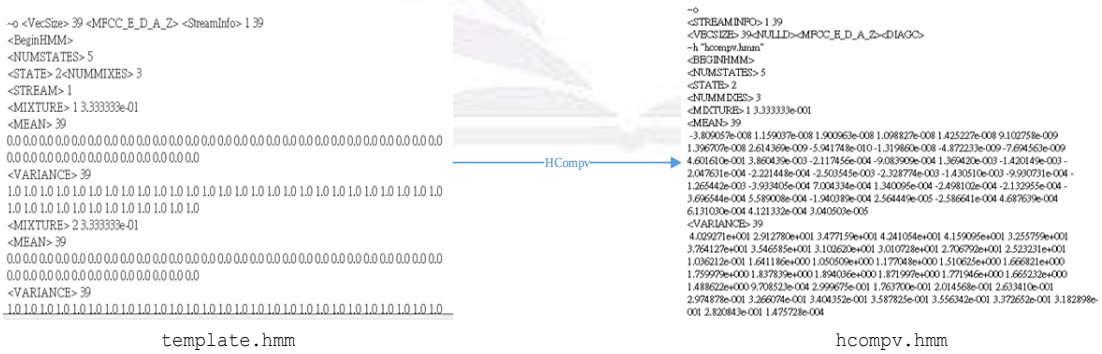


digitSyl.mlf 檔案內部

digitMonophone.mlf 檔案內部

圖 34

二、 HELd 轉換流程圖



template.hmm

hcompv.hmm

圖 35

三、HH Ed 訓練內容(macro.01)

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A_Z><DIAGC>
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<NUMMIXES> 3
<MIXTURE> 1 3.333333e-001
<MEAN> 39
 -3.809057e-008 1.159037e-008 1.900963e-008 1.098827e-008 1.425227e-008 9.102758e-009
1.396707e-008 2.614369e-009 -5.941748e-010 -1.319860e-008 -4.872233e-009 -7.694563e-009
4.601610e-001 3.860439e-003 -2.117456e-004 -9.083909e-004 1.369420e-003 -1.420149e-003 -
2.047631e-004 -2.221448e-004 -2.503545e-003 -2.328774e-003 -1.430510e-003 -9.930731e-004 -
1.265442e-003 -3.933405e-004 7.004334e-004 1.340095e-004 -2.498102e-004 -2.132955e-004 -
3.696544e-004 5.589008e-004 -1.940389e-004 2.564449e-005 -2.586641e-004 4.687639e-004
6.131030e-004 4.121332e-004 3.040503e-005
<VARIANCE> 39
 4.029271e+001 2.912780e+001 3.477159e+001 4.241054e+001 4.159095e+001 3.255759e+001
```

圖 36

四、HRest 訓練 50 次之 macro.50 檔案內容

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A_Z><DIAGC>
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<NUMMIXES> 3
<MIXTURE> 1 3.333339e-001
<MEAN> 39
 9.569440e+000 -2.750445e+000 -2.476514e+000 -4.489776e+000 -3.829736e+000 -2.686209e+
000 -4.166689e+000 -5.205840e+000 -3.830750e+000 -7.347855e+000 -7.850455e+000 -
5.650592e+000 7.970228e-001 -2.862872e-001 1.326642e+000 7.823126e-001 1.196349e+000 -
1.111420e-002 -1.698502e-001 -3.463871e-002 -4.426633e-001 -1.241272e-001 1.127809e-001
6.654242e-002 2.558294e-001 -2.081805e-002 -1.211085e-001 2.090633e-002 8.676448e-002
1.116498e-001 1.164631e-001 1.597860e-001 8.931107e-002 -8.015735e-002 -3.114138e-002
3.951019e-002 4.436139e-002 2.662078e-002 -2.565573e-003
<VARIANCE> 39
 3.255151e+001 7.176430e+001 3.509531e+001 3.649323e+001 1.049558e+002 5.151472e+001
5.900343e+001 8.813634e+001 4.181998e+001 4.570561e+001 2.560419e+001 2.620058e+001
```

圖 37

九、參考資料

(一) 辨識用數字語音資料庫部分來源：

1. 張智星，2005，「 Audio Signal Processing and Recognition (音訊處理與辨識)」,網站範例下載 ,http://mirlab.org/jang/books/audiosignalprocessing/
2. 王小川，2009，「語音訊號處理」，光碟檔案，全華科技圖書股份有限公司

(二) 文獻參考：

1. Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw,Gareth Moore, Julian Odell,Dave Ollason, Dan Povey, ValtchoValtchev, Phil Woodland, "The HTK Book(for HTK version 3.4)", Cambridge University Engineering Department, 2006.
2. Berlin Chen," Introduction to HTK Toolkit", Department of Computer Science & Information Engineering National Taiwan Normal University,2006
3. 王小川，2009，「語音訊號處理」，全華科技圖書股份有限公司
4. 張智星，2005，「 Audio Signal Processing and Recognition (音訊處理與辨識)」
5. 凌偉益、朱耀志，「MATLAB 軟體應用於數字語音辨識」，國立屏東科技大學專題論文
6. 黃志賢、張家翔，2013，「自動文稿產生與執行系統-以 HTK 工具為例」，崑山科技大學數位生活科技研究所論文
7. 維基百科