

中華民國第 62 屆中小學科學展覽會

作品說明書

高級中等學校組 電腦與資訊科

編 號：

作品名稱：分秒必爭！智慧紅綠燈協助緊急車輛開道

關 鍵 詞：聲音分析、相似度比較演算法、智慧城市

## 摘要

在台灣多少都有看到新聞媒體報導因駕駛人未禮讓緊急車輛，而導致緊急車輛值勤車禍事件一再的發生，本研究為了保障緊急車輛與用路人的安全，提出三種演算法來判斷是否為緊急車輛，在此實驗過程中在我們利用 Arduino 開發版中結合動態演算法並應用了智慧城市理念，開發出一種能夠自動偵測是否有緊急車輛即將經過的紅綠燈，讓紅綠燈能夠在緊急車輛即將通過之前做合適的變換，藉以保障緊急車輛與用路人的安全。在實驗結果，我們初步得到不錯的車輛辨識率，未來可在改善演算法，提升辨識率，來全面達到禮讓緊急車輛行車的安全。

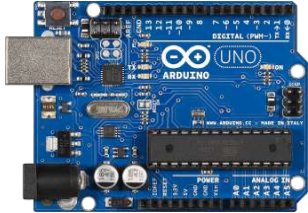



## 壹、研究動機

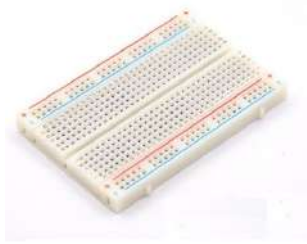

在台灣常常可以在大眾媒體上看到因有用路人未禮讓緊急車輛 (救護車、警車、消防車等)或因正值車流尖峰時段而耽誤了救援或執法的黃金時間，甚至引發車禍導致更多不可預測的後果，許多用路人未禮讓緊急車輛的情況都是因塞車或用路人應注意而未注意所導致，因此我們想出了在路口紅綠燈前架設感測器，利用聲音的感測、計算，讓緊急車輛接近時會使緊急車輛該方向的紅綠燈切換成綠燈，保障用路人與緊急車輛的用路安全，也縮減緊急車輛的行駛時間。

## 貳、研究目的

為了使緊急車輛在緊急狀況時能夠以最快的時間到達目的地以及維護緊急車輛在路途中的安全，因此我們希望能夠開發出一款智慧紅綠燈，這個紅綠燈將結合聲音與相似度演算法的分析，在緊急車輛快要到達紅綠燈之前讓緊急車輛行駛方向的紅綠燈轉換為綠燈，並且讓與緊急車輛同方向的車流疏散，也避免掉緊急車輛因闖紅燈而造成的風險，因此本研究希望能夠探討一些相似度演算法在聲音相似度判斷上的可行性，以及找出可行的方法對這些演算法做出改進，讓聲音相似度的判斷可以更加準確，期望未來有機會能用相同的方法實現在現實生活中。

## 參、研究設備及器材

實驗器材品項	實驗器材圖式	說明
Arduino UNO 板	 <p>▲ 圖一 Arduino UNO 板</p>	主要控制器
聲音感測模組  MAX - 4466	 <p>▲ 圖二 聲音感測模組</p>	接收聲音的裝置
Led 燈 (紅、25黃、綠)	 <p>▲ 圖三 Led 燈 (紅、黃、綠)</p>	紅綠燈的燈泡
杜邦線	 <p>▲ 圖四 杜邦線</p>	連接設備與通電

麵包版	 <p>▲ 圖五 麵包版</p>	用以放置杜邦線等相關設備
蜂鳴器	 <p>▲ 圖六 蜂鳴器</p>	發送聲音的裝置

## 肆、文獻探討

### 4.1 相似度分析演算法

#### 4.1.1 最長共同子序列 *Longest Common Subsequence*

在一個序列中，我們將該序列的數個字元或空字元刪除後，得到的序列都稱為該序列的共同子序列，而當有兩個  $X = [x_1, x_2, x_3, x_4, x_5]$ ,  $Y = [x_4, x_5]$ ，若  $Z$  同時為兩序列的子序列，則序列  $Z$  稱為兩序列的共同子序列(*Subsequence*)，並將所有共同子序列中最長的那一個稱為最長共同子序列(*LCS, Longest Common Subsequence*)。

#### 4.1.2 漢明距離 *Hamming distance*

漢明距離是由美國數學家理察·衛斯里·漢明所提出的一種字串相似度的指標，在資訊理論當中，兩個等長字串之間的漢明距離是兩個字串對應位置的不同字元個數，換句話說，漢明距離是指其中一個字串要變成另一個字串所需要替換的字元數量。

#### 4.1.3 餘弦相似度 *Cosine Similarity*

餘弦相似度是透過測量兩向量夾角的餘弦值來度量其之間的相似性。

兩個向量間的餘弦值可以通過使用歐幾里得點積公式求出：

$$a \cdot b = |a| |b| \cos\theta$$

給定兩個屬性向量  $A$  和  $B$ ，其餘弦相似性 $\theta$ 由點積和向量長度給出，如下：

$$similarity = \cos(\theta) = \frac{A \cdot B}{|A| |B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

這裡的  $A_i$  與  $B_i$  分別表示向量  $A$  與向量  $B$  的各分量。

## 4.2 控制系統實作

### 4.2.1 控制器

本研究各控制器的節點均使用 Arduino Uno 開發板，Arduino Uno 開發版是一個相當受歡迎的開發版，也是最適合初學者使用的開發版，Arduino Uno 是基於Microchip ATmega328P 微控制器的開源微控制器板，由Arduino.cc開發。該開發板具有14個數位I/O引腳（其中6個可用於PWM輸出），6個類比輸入引腳，可以連接到各種擴展板和其他電路，並且可以通過B型USB線(和列表機相同的方口USB線)與Arduino IDE進行程式編寫。



▲ 圖七 Arduino UNO 板

### 4.2.2 聲音感測模組 MAX - 4466

本研究採集聲音的音高是使用聲音感測模組 MAX-4466，這個模組的前方有一個麥克風接收器，並且會依照外部接收到的聲音依照高低輸出一個數字表示收到的聲音的音高，不過此數字並不代表聲音的分貝，只是一個表示震幅的數字，這個數字的範圍介於 0 ~ 1023 之間。



## 伍、研究過程及方法

### 5.1 相似度比較演算法

#### 5.1.1 最長共同子序列與相似度的運算

我們將當前接收到的聲音與我們預先建立好的緊急車輛的聲音求最長共同子序列的長度以計算其相似度。

最長共同子序列長度我們採用動態規劃法 (*Dynamic Programming*, *DP*) 來做計算。

假設現在有兩個序列  $A, B$  並且定義  $DP[i, j]$  表示為序列  $A$  的前  $i$  個字元與序列  $B$  的前  $j$  個字元採用相同的方式做最長共同子序列後計算出來的最長共同子序列長度。 $LCS$  使用動態規劃做計算的方式如下：

$$DP[i, j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ \max(DP[i-1][j], DP[i][j-1]) & \text{if } A_i \neq B_j \\ DP[i-1][j-1] + 1 & \text{if } A_i = B_j \end{cases}$$

在運算的過程中，先將  $A$  序列的第 1 個元素依序與序列  $B$  的第 1 個元素、第 2 個元素，依序至序列  $B$  的最後一個元素做上方式子的判斷與計算，再由  $A$  序列的第 2 個元素與序列  $B$  的第 1 個元素、第 2 個元素，依序至序列  $B$  的最後一個元素做上方式子的判斷與運算。

依此方式，計算至  $A$  序列的最後一個元素與  $B$  序列的最後一個元素，即可算出  $A, B$  兩序列的最長共同子序列長度。

#### 5.1.2 最長共同子序列判斷相似度的方式

為了避免誤差，因此我們定義只要兩個聲音的音高差不超過誤差範圍都視為同樣的聲音，並將這個方法套用到最長共同子序列的比對上，當麥克風每接收到 320 筆資料時與我們預先建立好的 1000 筆消防車資料做一次最長共同子序列長度的計算，並用此結果分析數據的相似程度。

#### 5.1.3 漢明距離的計算

漢明距離的計算方式為，取兩個相同長度的字串，並計算這兩個字串之間共有多少個位置的字元為不相同的，假設其數量為  $K$ ，則我們稱這兩個字串之間的漢明距離為  $K$ 。

#### 5.1.4 漢明距離計算相似度的方式

為了避免誤差，我們定義兩個聲音的音高差不超過誤差範圍都視為同樣的聲音，並將其套用到漢明距離的比對上。

當麥克風接收到 320 筆資料時，在我們預先建立好的 1000 筆消防車資料上做漢明距離的計算，由於漢明距離計算時長度必須相同，因此我們依序枚舉這 1000 筆資料的起點，定義其起點為  $L$ ，那終點就會是  $L + 319$ ，換句話說，我們依序將這 320 筆資料對消防車 1000 筆資料中所有  $[L, L + 320]$  且  $L + 320 \leq 1000$  的閉區間取漢明距離，假設所有符合條件的區間的最短漢明距離為  $Y$ ，那麼我們定義其 320 筆資料與消防車鳴笛的相似度為  $\frac{320 - Y}{320}$ 。

#### 5.1.5 餘弦相似度的計算

給定兩個相同長度的序列，序列中的每一個數字都為其分量，並將分量帶入餘弦相似度公式計算出這兩個相同長度序列的餘弦相似度，計算餘弦相似度的公式如下：

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{|A| |B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

#### 5.1.6 餘弦相似度判斷相似度的方式

當麥克風接收到 320 筆資料時，在我們預先建立好的 1000 筆消防車資料上做餘弦相似度的計算，由於餘弦相似度計算時兩序列長度必須相同（分量的數量才會相同），因此我們依序枚舉這 1000 筆資料的起點，定義其起點為  $L$ ，那終點就會是  $L + 319$ ，換句話說，我們依序將這 320 筆資料對消防車 1000 筆資料中所有  $[L, L + 320]$  且  $L + 320 \leq 1000$  的閉區間。假設所有符合條件的區間的最大餘弦相似度為  $C$ ，則我們定義這 320 筆資料與消防車鳴笛的相似度為  $C$ 。

### 5.2 二分法與演算法的搭配

#### 5.2.1 為什麼要使用二分法

由於數據的誤差值可能會造成判斷上出來的結果有所誤差，因此我們利用二分法將聲音分成高音與低音，利用高音與低音來判斷是否為消防車的聲音。

### 5.2.2 二分法與最長共同子序列的搭配

我們將序列的高音 ( 大於等於 750 單位)轉為字元  $H$ , 低音 ( 小於 750 單位 )轉為字元  $L$ , 重新組合後, 求出兩字元序列的最長共同子序列長度, 並計算其與消防車的相似程度。

### 5.2.3 二分法與漢明距離的搭配

我們將序列的高音轉為字元  $H$ , 低音轉為字元  $L$ , 重新組合後, 求出兩字元序列之間的漢明距離, 並計算其與消防車的相似程度。

## 5.3 消防車數據的建立 ( 基準組 )

我們利用蜂鳴器與預先建立好的頻譜模擬出消防車的聲音, 並將發出來的聲音利用聲音感測模組接收, 並輸出每 20 毫秒的音高值記錄起來, 用以當作消防車的基準值。

## 5.4 實驗 ( 待比對 ) 數據的建立

### 5.4.1 一般組

與消防車數據採用同方式建立, 由於兩組數據在頻譜上的起點不會是同一個位置, 故設立一個一般組來檢查各個比對演算法的相似度比較結果。

### 5.4.2 干擾組

與消防車數據採用同方式建立, 不過在建立時故意添加一些環境噪音 ( 如: 拍手、大聲講話等) 來干擾消防車聲音的數據。環境噪音用以仿真現實路況的吵雜聲, 並藉由比對後的結果探討在有噪音干擾的情況下各個相似度演算法的相似度比較結果, 並加以做各演算法的優劣分析。

### 5.4.3 錯誤組

錯誤組即為隨便收集一段聲音的數據, 並用此數據檢查相似度演算法的相似度判定。

本研究為了研究在現實中的可行性, 因此將錯誤組以以下三種方式生成

1. 在一開始製造噪音
2. 偶爾製造噪音
3. 持續製造噪音

## 5.5 起點的特判

### 5.5.1 為什麼需要特判起點

如果一般組為消防車的聲音, 那麼我們如果畫成圖來看的話會發現, 我們將一般組的圖形稍微平移一下就會幾乎疊合於基準組某段區間的圖形, 會導



致這樣的結果的原因是我們接收到消防車聲音的音高與我們預先建立好的消防車的起始聲音會不同，因此我們必須先將起點做匹配的動作，這個動作我們預期可以提高消防車的辨識度與降低判斷錯誤的機率。

#### 5.5.2 需要特判起點的演算法

由於最長共同子序列與漢明距離和餘弦相似度需要逐一比較每一個字元，故需要特判起點。

#### 5.5.3 特判起點的方式

我們判斷起點的方式為，逐一取得基準組連續 5 個單位時間 ( 1 秒 ) 的數據，這 5 筆數據必須要與待比對的數據的前 5 筆數據一樣 (數據一樣的定義為兩個相同位置的數據的音高差不超過誤差範圍) 才會判定從這連續 5 個單位時間開始是起點，判定為起點之後才會去做最長共同子序列、漢明距離、餘弦相似度的計算。

### 5.6 誤差範圍的設定

#### 5.6.1 為什麼需要設定誤差範圍

由於麥克風所接收到聲音後所輸出的音高可能會受到外在環境的影響而導致數值有所浮動，因此我們必須設置合理的誤差範圍來避免環境影響到判斷。

#### 5.6.2 誤差範圍設定的基準

在我們實際測試之後，發現在一個無人的教室裡，偵測到的聲音數據都介於 480 到 550 之間。起初我們利用直接測定出的數值做分析，並設定不同的誤差值比較，多次實驗後找出最準確合適的誤差值為 80。

### 5.7 相似度演算法優劣的探討方式

#### 5.7.1 分組

我們將實驗分成以下幾組以探討每個變因與準確度的關係

1. 無二分數據、無判斷起點組
2. 無二分數據、有判斷起點組
3. 有二分數據、無判斷起點組
4. 有二分數據、有判斷起點組

### 5.8 與緊急車輛鳴笛聲音相似度的判定

#### 5.8.1 最長共同子序列

最長共同子序列長度	判定
200 以上	非常相似
150 ~ 199	部分相似
0 ~ 149	非常不相似

▲ 表一 用最長共同子序列判定緊急車輛的鳴笛聲音相似度

### 5.8.2 漢明距離

漢明距離轉換後的相似度	判定
70 % 以上	非常相似
50 % ~ 69 %	部分相似
未達 50 %	非常不相似

▲ 表二 漢明距離判定緊急車輛的鳴笛聲音相似度

### 5.8.3 餘弦相似度

餘弦相似度	判定
70 % 以上	非常相似
50 % ~ 69 %	部分相似
未達 50 %	非常不相似

▲ 表三 餘弦相似度判定緊急車輛的鳴笛聲音相似度

## 5.9 模型實際模擬與測試

### 5.9.1 模擬緊急車輛的聲音

實驗前，我們先利用預先建立好的消防車頻譜輸入進 Arduino UNO 板，並透過蜂鳴器來發出消防車的聲音，藉以模擬真實情況。

### 5.9.2 紅綠燈

模型的紅綠燈我們使用 LED 燈泡製作，藉以模擬道路真實的情況。



▲ 圖九 模擬道路

#### 5.9.3 聲音感測模組

我們將聲音感測模組放置在紅綠燈前，藉以收音判斷是否即將有緊急車輛經過。



▲ 圖十 聲音感測模組(放置在紅綠燈前)

#### 5.9.4 判定的依據

為了避免掉許多誤判，因此我們此實驗只有在判定結果為非常相似時才會認定該車為緊急車輛。

#### 5.9.5 實際使用模型模擬的方式

我們利用裝有蜂鳴器且蜂鳴器會發出緊急車輛音效的手工車模擬為緊急車輛，並且我們預先將緊急車輛方向的紅綠燈設定為紅燈且永遠不會轉為綠燈，只有在緊急車輛接近時該方向的紅綠燈才會轉為綠燈，藉以觀察是否能夠對緊急車輛進行判定。

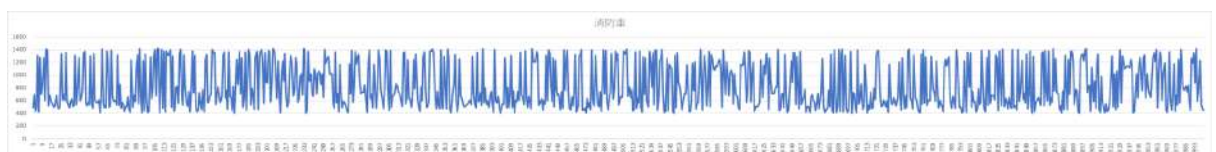


▲ 圖十一 實際模型模擬緊急車輛通過紅綠燈

## 陸、研究結果

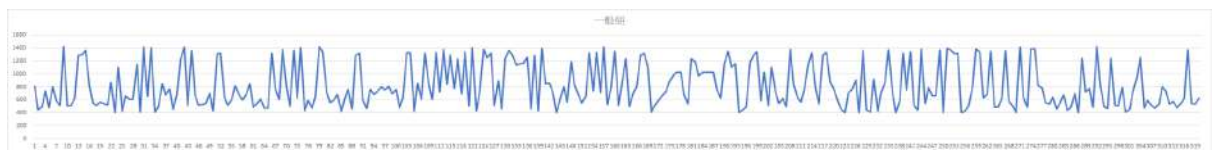
### 6.1 實驗數據樣本

#### 6.1.1 基準組



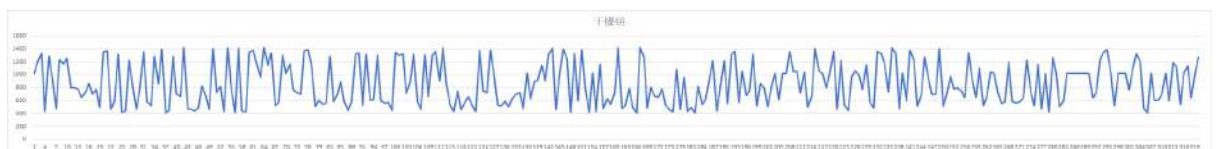
▲ 圖十二 基準組的音高曲線圖

#### 6.1.2 一般組



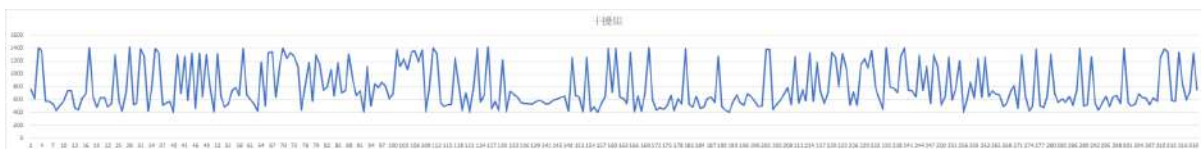
▲ 圖十三 一般組的音高曲線圖

#### 6.1.3 干擾組 - 1



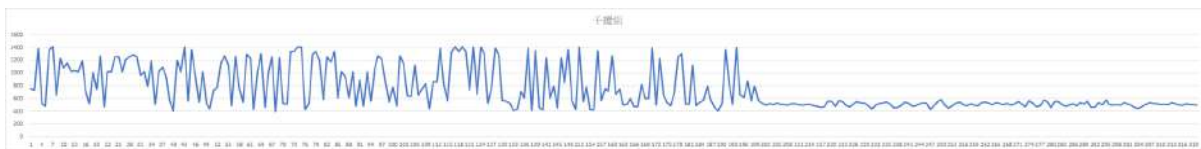
▲ 圖十四 干擾組 - 1的音高曲線圖

### 6.1.2 干擾組 - 2



▲ 圖十五 干擾組 - 2的音高曲線圖

### 6.1.3 干擾組 - 3



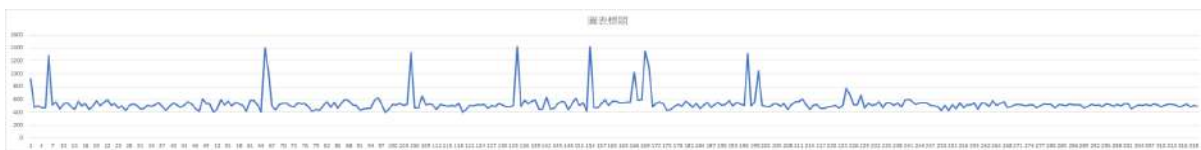
▲ 圖十六 干擾組 - 3的音高曲線圖

### 6.1.4 錯誤組 - 1 (一開始時給予雜音)



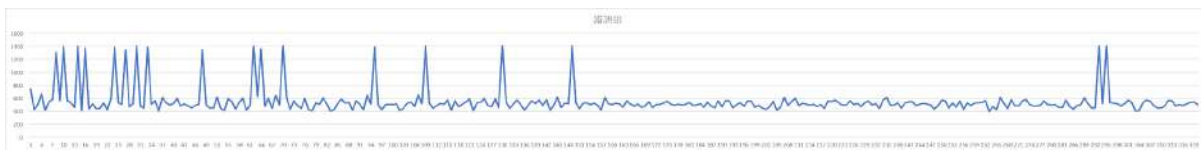
▲ 圖十七 錯誤組 - 1的音高曲線圖

### 6.1.5 錯誤組 - 2 (偶爾給予雜音)



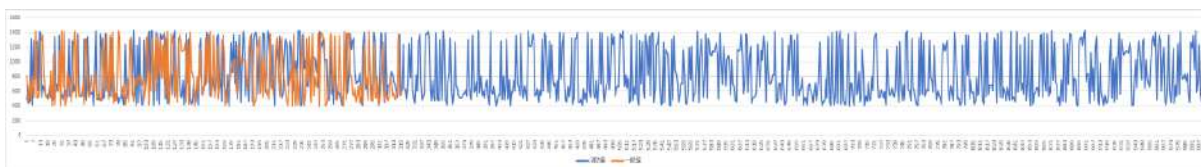
▲ 圖十八 錯誤組 - 2的音高曲線圖

### 6.1.6 錯誤組 - 3 (持續給予雜音)



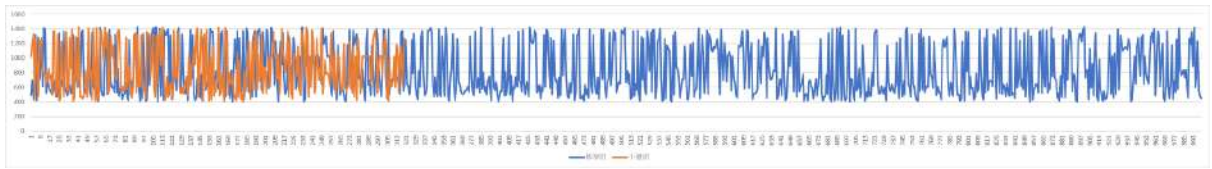
▲ 圖十九 錯誤組 - 3的音高曲線圖

### 6.1.7 基準組與一般組、干擾組、錯誤組的圖形疊合

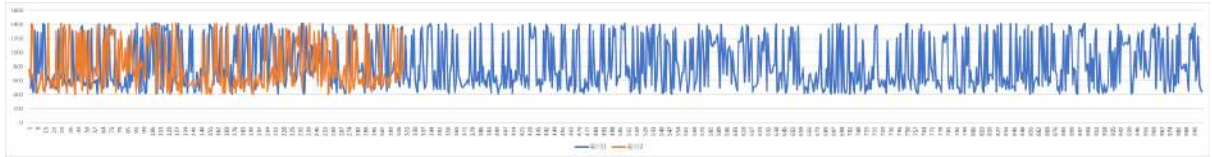


▲ 圖二十 基準組與一般組的音高曲線疊合圖

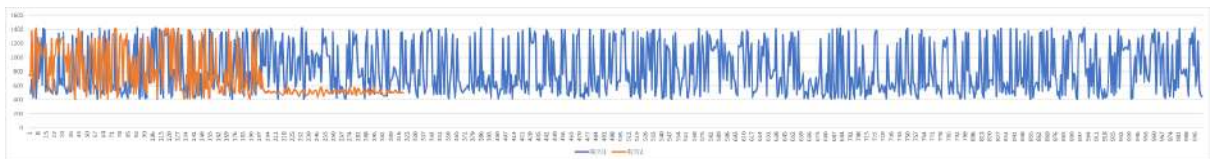




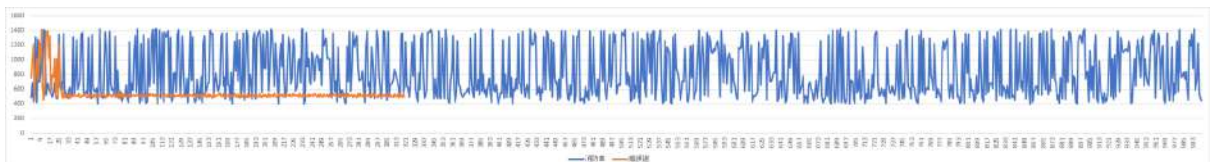
▲ 圖二十一 基準組與干擾組 - 1的音高曲線疊合圖



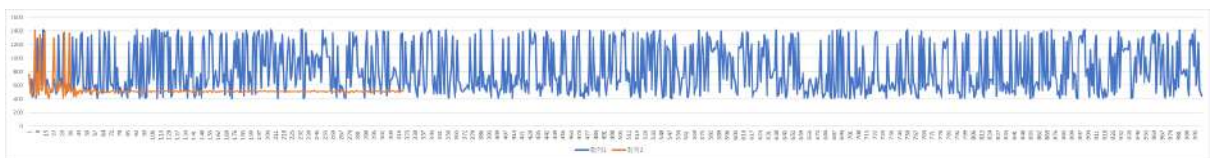
▲ 圖二十二 基準組與干擾組 - 2的音高曲線疊合圖



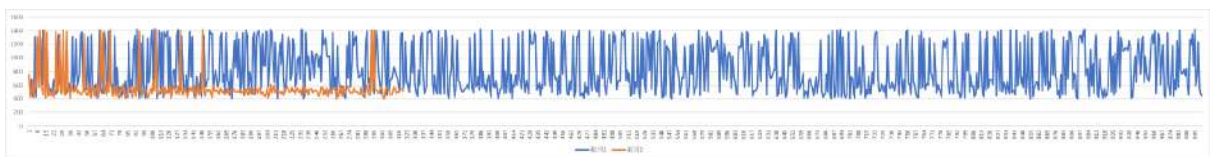
▲ 圖二十三 基準組與干擾組 - 3的音高曲線疊合圖



▲ 圖二十四 基準組與錯誤組 - 1的音高曲線疊合圖



▲ 圖二十五 基準組與錯誤組 - 2的音高曲線疊合圖



▲ 圖二十六 基準組與錯誤組 - 3的音高曲線疊合圖

## 6.2 各個演算法在各實驗數據的表現 (無二分數據、無特判起點)

### 6.2.1 最長共同子序列 (無二分數據、無特判起點)

實驗組別	與基準組的最長共同子序列長度
一般組	244
干擾組 1	228

干擾組 2	258
干擾組 3	250
錯誤組 1 (一開始時給予雜音)	201
錯誤組 2 (偶爾給予雜音)	211
錯誤組 3 (持續給予雜音)	227

▲ 表四 實驗組別與基準組的最長共同子序列長度

#### 6.2.2 漢明距離 (無二分數據、無特判起點)

實驗組別	漢明距離轉換後的相似度
一般組	45.31 %
干擾組 1	41.87 %
干擾組 2	56.56 %
干擾組 3	58.43 %
錯誤組 1 (一開始時給予雜音)	55.93 %
錯誤組 2 (偶爾給予雜音)	55.93 %
錯誤組 3 (持續給予雜音)	48.75 %

▲ 表五 實驗組別與基準組用漢明距離轉換後的相似度

#### 6.2.3 餘弦相似度 (無二分數據、無特判起點)

實驗組別	餘弦相似度
------	-------

一般組	88.26 %
干擾組 1	88.95 %
干擾組 2	87.27 %
干擾組 3	86.75 %
錯誤組 1 (一開始時給予雜音)	91.36 %
錯誤組 2 (偶爾給予雜音)	90.81 %
錯誤組 3 (持續給予雜音)	87.91 %

▲ 表六 實驗組別與基準組的餘弦相似度

### 6.3 各個演算法在各實驗數據的表現 (無二分數據、有特判起點)

#### 6.3.1 最長共同子序列 (無二分數據、有特判起點)

實驗組別	最長共同子序列長度
一般組	240
干擾組 1	224
干擾組 2	258
干擾組 3	250
錯誤組 1 (一開始時給予雜音)	0 (找不到起點)
錯誤組 2 (偶爾給予雜音)	203
錯誤組 3 (持續給予雜音)	225

▲ 表七 實驗組別與基準組的最長共同子序列長度

#### 6.3.2 漢明距離 (無二分數據、有特判起點)

實驗組別	漢明距離轉換後的相似度
一般組	42.81 %
干擾組 1	36.56 %
干擾組 2	48.12 %
干擾組 3	53.75 %



錯誤組 1 (一開始時給予雜音)	0.00 % (找不到起點)
錯誤組 2 (偶爾給予雜音)	49.37 %
錯誤組 3 (持續給予雜音)	45.76 %

▲ 表八 實驗組別與基準組用漢明距離轉換後的相似度

### 6.3.3 餘弦相似度 (無二分數據、有特判起點)

實驗組別	餘弦相似度
一般組	86.62 %
干擾組 1	87.81 %
干擾組 2	84.05 %
干擾組 3	85.32 %
錯誤組 1 (一開始時給予雜音)	0.00 % (找不到起點)
錯誤組 2 (偶爾給予雜音)	89.31 %
錯誤組 3 (持續給予雜音)	86.92 %

▲ 表九 實驗組別與基準組的餘弦相似度

## 6.4 各個演算法在各實驗數據的表現 (二分數據、無特判起點)

### 6.4.1 最長共同子序列 (二分數據、無特判起點)

實驗組別	與基準組的最長共同子序列長度
一般組	204
干擾組 1	173
干擾組 2	211
干擾組 3	213
錯誤組 1 (一開始時給予雜音)	17
錯誤組 2 (偶爾給予雜音)	13
錯誤組 3 (持續給予雜音)	113

▲ 表十 實驗組別與基準組的最長共同子序列長度

6.4.2 漢明距離 ( 二分數據、無特判起點 )

實驗組別	漢明距離轉換後的相似度
一般組	50.62 %
干擾組 1	53.75 %
干擾組 2	55.91 %
干擾組 3	58.82 %
錯誤組 1 ( 一開始時給予雜音 )	0.05 %
錯誤組 2 ( 偶爾給予雜音 )	0.04 %
錯誤組 3 ( 持續給予雜音 )	36.05 %

▲ 表十一 實驗組別與基準組用漢明距離轉換後的相似度

6.4.3 餘弦相似度 ( 二分數據、無特判起點 )

實驗組別	餘弦相似度
一般組	71.54 %
干擾組 1	73.52 %
干擾組 2	74.13 %
干擾組 3	76.92 %
錯誤組 1 ( 一開始時給予雜音 )	23.04 %
錯誤組 2 ( 偶爾給予雜音 )	20.15 %
錯誤組 3 ( 持續給予雜音 )	20.64 %

▲ 表十二 實驗組別與基準組的餘弦相似度

6.5 各個演算法在各實驗數據的表現 ( 二分數據、特判起點 )

6.5.1 最長共同子序列 ( 二分數據、特判起點 )

實驗組別	與基準組的最長共同子序列長度
一般組	204
干擾組 1	173

干擾組 2	211
干擾組 3	213
錯誤組 1 (一開始時給予雜音)	0 (找不到起點)
錯誤組 2 (偶爾給予雜音)	13
錯誤組 3 (持續給予雜音)	113

▲ 表十三 實驗組別與基準組的最長共同子序列長度

#### 6.5.2 漢明距離 (二分數據、特判起點)

實驗組別	漢明距離轉換後的相似度
一般組	50.62 %
干擾組 1	53.75 %
干擾組 2	55.91 %
干擾組 3	58.82 %
錯誤組 1 (一開始時給予雜音)	0.00 % (找不到起點)
錯誤組 2 (偶爾給予雜音)	0.04 %
錯誤組 3 (持續給予雜音)	36.05 %

▲ 表十四 實驗組別與基準組用漢明距離轉換後的相似度

#### 6.5.3 餘弦相似度 (二分數據、特判起點)

實驗組別	餘弦相似度
一般組	71.54 %
干擾組 1	73.52 %
干擾組 2	74.13 %
干擾組 3	76.92 %
錯誤組 1 (一開始時給予雜音)	0.00 % (找不到起點)
錯誤組 2 (偶爾給予雜音)	20.15 %
錯誤組 3 (持續給予雜音)	20.64 %

## 柒、討論

### 7.1 採用的相似度演算法的優點與缺點

#### 7.1.1 最長共同子序列的優點

最長共同子序列的優點為非常適合用以判斷聲音的特徵值，而且不受於每一個聲音位置上的限制，只要一段聲音與聲音特徵值在相對順序下匹配的越多，最長共同子序列的長度也會越多。

#### 7.1.2 漢明距離的優點

從實驗數據中我們可以發現在使用漢明距離的情況下所算出來的相似度數值都是偏低的，不過這也驗證了漢明距離在判斷相似度上的嚴謹度，由於漢明距離計算時會對區間內每一個位置做嚴格的比對，不像最長共同子序列是以相對順序做判斷，因此匹配較為嚴謹與嚴格，能夠減少判斷錯誤的情況發生。

#### 7.1.3 餘弦相似度的優點

餘弦相似度是透過兩個向量夾角的餘弦值來計算出兩向量之間的相似度，換句話說，餘弦相似度是以兩個向量之間角度的餘弦值確定兩個向量是否大致為同一個方向，因此相較於漢明距離之下，餘弦相似度在判斷下不會像漢明距離這麼嚴格，因此也能避免接收的聲音有所誤差而影響到相似度的判斷。從無二分數據的實驗就可以發現，餘弦相似度所計算出來的相似度比漢明距離高出了許多。

#### 7.1.4 最長共同子序列的缺點

從無二分的數據可以發現，一般組、干擾組、錯誤組的最長共同子序列長度大部分都非常接近，難以正確判斷出是否真的為緊急車輛，最長共同子序列雖然對特徵值的判斷特別突出，但是最長共同子序列是針對相對位置做判斷，因此在判斷相似度上較不穩定。

#### 7.1.5 漢明距離的缺點

透過實驗數據可以發現在使用漢明距離判斷的情況下，在一般組、干擾組、錯誤組之間的相似度數值是很接近的，而且甚至錯誤組的相似度比一般組和干擾組還要高，一般組的相似度也偏低，可以合理的推判是因為一般組的數據無法在一般組找到一個最適合平移的點，即使平移了，整體重和度也偏低，會有這

樣的情況也跟實際收音的誤差有關，因此這也重新驗證了漢明距離在判斷相似度的嚴謹性，但是過於嚴謹也成為了漢明距離的缺點。

#### 7.1.6 餘弦相似度的缺點

從沒有二分數據的實驗結果可以發現使用餘弦相似度不管是哪一組餘弦相似度都非常的高，由於數據在沒有二分的情況下浮動偏低，因此使用餘弦相似度會因為轉換成向量後發現向量的整體方向都偏相似，因此算出來的餘弦相似度也會偏高。

### 7.2 相似度演算法的改進

#### 7.2.1 改進的方式

為了改進我們所使用的相似度演算法的缺點，因此我們使用特判起點與二分數據來嘗試解決那些缺點。

#### 7.2.2 相似度演算法在特判起點後的狀況

從實驗數據中可以發現在特判起點後數據沒有太大的變化，除了錯誤組有一組發生找不到起點的情況，其餘找到的起點的數據都沒有變化太大，甚至跟原本沒有判斷起點的一樣，因此使用特判起點這個方法對改進演算法對相似度的判定的效果沒有非常顯卓，還需要再改進或是搭配其他的改進方式讓整體判斷的結果能夠越準確。

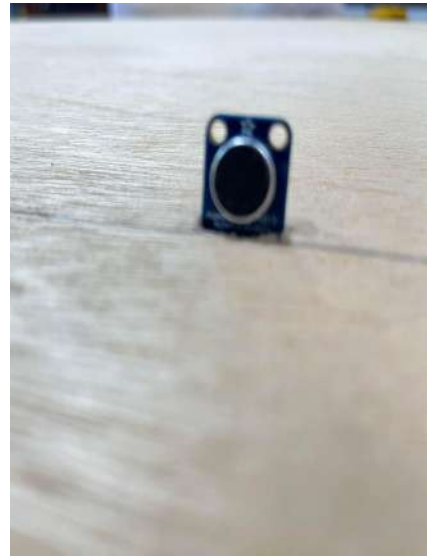
#### 7.2.3 相似度演算法在二分數據後的狀況

從二分後的實驗數據可以發現在數據經過二分之後，所有我們使用的相似度演算法在判斷相似度上變得非常有鑑別度，就不會像數據沒有二分時在判斷上沒有什麼鑑別度甚至是有判斷錯誤的情況，特別是餘弦相似度在二分之後變得非常有鑑別度，一般組與干擾組相似度都有超過 70%，三個錯誤組相似度都只有 20% 左右，漢明距離雖然算出來的相似度還是偏低，但是已經比沒有二分數據的結果準確了不少，最長共同子序列的鑑別度也比無二分數據的鑑別度還要好，因此二分數據是一個效果非常顯卓的改進方法。

### 7.3 實際使用模型實驗的結果



▲ 圖二十七  
紅綠燈原本為紅色



▲ 圖二十八  
路邊有感測器持續偵測聲音



▲ 圖二十九  
經過感測器時判斷為緊急車輛的聲音  
紅綠燈轉為綠色



▲ 圖三十  
順利讓緊急車輛安全通過

## 捌、結論

### 8.1 在不改進演算法的情況下

從實驗數據可以發現如果我們完全不改進演算法的話所判斷出來相似度跟實際情況的誤差是很大的，因此如果我們完全不對演算法作改進的話是非常難以應用在現實中的。

### 8.2 對演算法做出改進之後

當我們對演算法做出改進 (特判起點、二分數據) 之後，從實驗的數據可以發現在我們對演算法做出改進之後判斷相似度上比無對演算法改進準確了非常多，因此如果要使用相同方式對相似度做判斷的話必須要對數據做一些處理與對演算法做出改進，才會使整套系統在判斷緊急車輛上更加準確。

### 8.3 哪一個改進方法效果最顯卓

從實驗數據可以發現，特判起點這一個方法雖然有讓錯誤組的其中一組因找不到起點而無數據結果，但是對於其他的錯誤組來說效果沒有非常顯卓，效果最為顯卓的為將數據以二分法二分後再利用相似度演算法做相似度的判斷，從實驗的數據就可以發現在數據被二分後再做相似度判定的鑑別度比無二分數據的鑑別度高出了非常多，且正確性也比無二分數據的正確性高出許多，因此將數據以二分法二分後再利用相似度演算法做相似度判定這個改進方法的效果是最顯卓的。

### 8.4 二分法為什麼效果會這麼顯卓

二分法可以避免掉麥克風在接收聲音時的誤差問題，在數據沒有被二分之前我們在使用漢明距離與最長共同子序列計算相似度時都必須使用我們所設置的誤差範圍來為兩個音判斷是否相同，但實際收音時有時候可能同一個音但因為外在的環境噪音干擾而造成誤差非常大而使程式判定兩個音並非同一個音，除此之外，由於將數據二分之後可以凸顯出每一時間聲音的特徵值(高或低)，因此利於相似度演算法做相似度的判定，總和以上，二分法是我們所發現的最好的改進演算法的方式。

### 8.5 如果要在現實中實現此想法

從實驗數據與結果來看，將數據二分後再使用相似度演算法做相似度的判斷會是最好的，而以演算法來看的話，在數據被二分之後使用餘弦相似度是最好的，因此如果要套用在我們現實中做使用的話，使用將數據以二分法二分之後再使用餘弦相似度做判斷會是最好的方法。

## 玖、使用限制與未來展望

### 9.1 使用限制

如果要套用此方式在現實中的話必須再額外考量到緊急車輛行駛速度的問題，如果車輛行駛的速度過快可能就會影響到判斷的結果，因此，未來實驗時速度會是一個需要克服的點。演算法的改進及提升辨識率也是我們未來可以持續研究的方向。

### 9.2 未來展望

未來如果想要實現類似的想法怎麼樣會更加完善呢？都卜勒效應(英語:Doppler effect)是波源和觀察者有相對運動時，觀察者接受到波的頻率與波源發出的頻率並不相同的現象。我們可以考量都卜勒效應並加入濾波技術來改善環境噪音對判斷造成的影響，除此之外我們也可以將此研究的想法結合到其他資訊相關領域上來讓整套系統更加完善，舉例來說，我們可以套用到 AI 相關領域上，利用 AI 智慧辨識來辨識緊急車輛或是利用 AI 音頻分離技術將各個車子的音頻分離甚至是結合 IoT 物聯網與車聯網(V2X)等等，如此一來，這樣的想法不僅能幫緊急車輛爭取到更多的時間同時也維護所有用路人與緊急車輛的行駛安全，期許有一天這樣的想法能夠實現在現實中。

## 壹拾、參考文獻資料及其他

- 1.Anushka Yadav, Kalyani Daharwal, Ashwini Kale, Priti Tayade, Asst. Prof. Akbar Nagani.Density & Sound Based Vehicular Traffic Controller.International Journal of Scientific Research & Engineering Trends Volume 6, Issue 4, July-Aug-2020.
- 2.Arif Shaik, Natalie Bowen†, Jennifer Bole, Gary Kunzi, Daniel Bruce, Ahmed Abdelgawad, Kumar Yelamarthi,  
Smart Car: An IoT Based Accident Detection System,2018 IEEE Global Conference on Internet of Things (GCIoT)
- 3.Bhandari Prachi, Dalvi Kasturi, Chopade Priyanka,  
Intelligent Accident-Detection And Ambulance- Rescue System,INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSUE 6, JUNE 2014



4. B.Janani Saradha, G.Vijayshri, T.Subha.Intelligent Traffic Signal Control System For Ambulance Using RFID And CLOUD. Second International Conference On Computing and Communications Technologies(ICCCT'17),p90-96.

5.Dipro Pramanick,Haaris Ansar,Hemant Kumar, Pranav S, Richa Tengshe and Binish Fatimah,Deep learning based urban sound classification and ambulance siren detector using spectrogram,2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)

6.Parveen Kaur, Animesh Das, Manash P. Borah, Smriti Dey.Smart Vehicle System using Arduino.2019. ADBU Journal of Electrical and Electronics Engineering (AJEEE) | Volume 3, Issue 1,p20-25

7.Pradip Togrikar

Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle ,Imperial Journal of Interdisciplinary Research (IJIR) Vol-2, Issue-4, 2016

8.Takuya Miyazakia,\* , Yuhki Kitazonoa, Manabu Shimakawab.Ambulance Siren Detector using FFT on dsPIC. 2013 . Proceedings of the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing 2013 ,p266-269.

9.Vasuki Shankar\*, RuthvikGautham, Vedaprakashvarma

Automated Traffic Signal for Hassle Free Movement of Ambulance,2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)

10.王文聖。

應用最長共同子序列與模糊方法辨識救護車鳴笛聲 ,民101,朝陽科大 資訊與通訊系碩士論文

。