

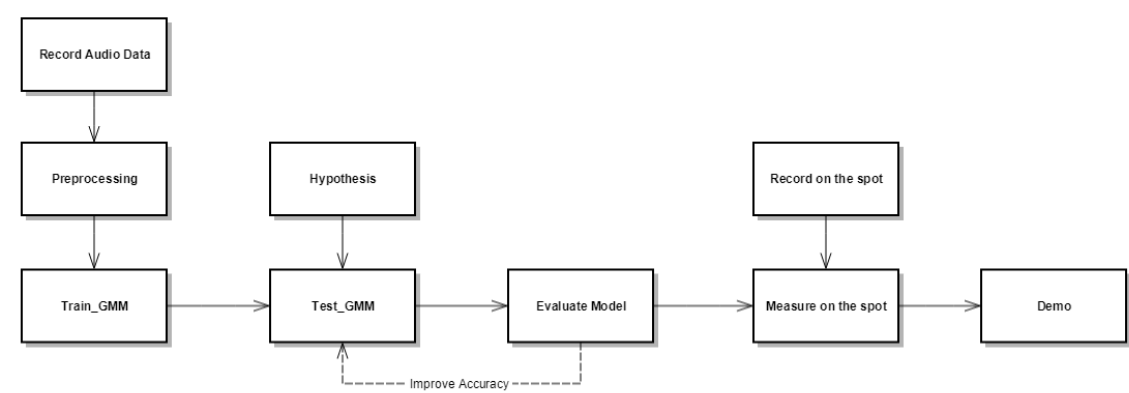
科研部_聲紋辨識

主要工作：說話人辨識

現場錄音 → 輸入欲驗證說話人的名字 → 說話人驗證 (1:1) → 顯示 預測結果、相似程度、照片
→ 說話人識別 (1:N) → 顯示 預測結果、照片

投影片連結:[蘇玫如_暑期實習驗收報告.pptx](#)

Flowchart



聲紋資料

* 準確率看看就好

姓名	是否已錄音	Train	Test	Test 內容	ver1 (1:N 識別) 76%→96%	原因	ver2 (1:1 驗證) 99.19%
任卓英	V	5	41/23	X 淵	100%		100 %
	V	5	27/25	比賽，法律條文	88%		90%
彭瑋玉	V	7	41/33	畢旅，保險，4種聲調	100%		100%
徐天鴻	V	5	48/38	比賽	92%		99%
戴珮君							
林上淳	V	5	46/27	房東	100%		100%
林淑芳	V	5	29/28	筑鈞，教授、畢照、選課	100%		100%
林頌修							
	V	5	14/14	比賽	41% → 100%	原本都是卓英干擾。已把 test 切乾淨、train 修正、重新訓練	100%
	V	5	17/17	比賽	68% → 100%	天鴻、卓英、嘉平。已把 test 切乾淨	100%
	V	5	16/16	X淵	82% → 100%	卓英又你。已把 test 切乾淨	100%

	V	5	25/25	房東	92%		100%
	V	5	23/23	寢室	83% → 100%	卓英全是你 = 。已把 test 切乾淨	100%
	V	5	28/28	減小強，生動	72% → 92%	不要再出現卓英了啦 QAQ。已把 test 切乾淨	100%
	V	6	25/25	PKU	52% → 100%	原本是天鴻跟我干擾。已把 test 切乾淨、train 修正、重新訓練	100%
	V	5	19/19	EAM	36% → 78%	已把 test 再重切乾淨	99%
	V	5	19/19	國財管	68% → 94%	全是卓英。已把 test 切乾淨	100%
	V	5	45/43	畢旅，保險	100%		100%
	V	5	16/16	PKU	62% → 93%	全是卓英。已把 test 切乾淨、train 修正、重新訓練	100%

Functions

```

import numpy as np
from scipy.io.wavfile import read
from python_speech_features import mfcc, delta
import os
import _pickle as pickle
from sklearn.mixture import GMM
from sklearn import preprocessing
import warnings
warnings.filterwarnings("ignore")
import pyaudio
import wave

import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import numpy as np
from PIL import Image
%matplotlib inline

def Train_MFCC_delta2(sr, audio):
    mfcc_feat = mfcc(audio,sr, 0.025, 0.01, 13,nfft=2048,appendEnergy =
False)
    d_mfcc_feat = delta(mfcc_feat,2)
    dd_mfcc_feat = delta(d_mfcc_feat,2)
    features = np.asarray(())
    features = np.hstack([mfcc_feat, d_mfcc_feat, dd_mfcc_feat])
    features = preprocessing.scale(features)
    return features

```

```

def Test_MFCC_delta2(sr, audio):
    mfcc_feat = mfcc(audio, sr, 0.025, 0.01, 13, nfft=2048, appendEnergy =
False)
    d_mfcc_feat = delta(mfcc_feat, 2)
    dd_mfcc_feat = delta(d_mfcc_feat, 2)
    features = np.asarray(())
    features = np.hstack([mfcc_feat, d_mfcc_feat, dd_mfcc_feat])
    #features = [np.concatenate((i, np.zeros((1208-i.shape[0], 39)))) for i
in features]
    feat = np.asarray(())

    for i in range(features.shape[0]):
        temp = features[i, :]
        if np.isnan(np.min(temp)):
            continue
        else:
            if feat.size == 0:
                feat = temp
            else:
                feat = np.vstack((feat, temp))

    features = feat

    # SD
    features = preprocessing.scale(features)
    return features

def Similarity(a, b, answer, predict):
    if predict == answer:
        score = 100

```

```

else:
    score = 100 + (abs(a-b)/b*1000)
return score

```

Model

```

path = "C:\\Users\\user\\Desktop\\_SR\\audio dataset\\Train\\"
folders = [os.path.join(path,f) for f in os.listdir(path)]
for i in range(len(folders)):
    folders[i] = folders[i]+'\\'

dest = "C:\\Users\\user\\Desktop\\_SR\\model\\"
for i in range(len(folders)):
    source = folders[i]
    name = source.split('\\')[-2]
    files = [os.path.join(source,f) for f in os.listdir(source) if
f.endswith('.wav')]
    features = np.asarray(())
    for f in files:
        sr, audio = read(f)
        vector = Train_MFCC_delta2(sr,audio)
        #print(vector.shape)
        if features.size == 0:
            features = vector
        else:
            features = np.vstack((features, vector))

    if(name==''):
        gmm = GMM(n_components = 10, n_iter = 200,
covariance_type='diag',n_init = 3)
        gmm.fit(features)
    else:
        gmm = GMM(n_components = 8, n_iter = 200,
covariance_type='diag',n_init = 3)
        gmm.fit(features)

    # save model with pickle
    picklefile = name+'.gmm'
    # model saved as .gmm
    pickle.dump(gmm,open(dest+picklefile,'wb'))
    print(i,': ',name,' finished')

```

建立好的模型會存在目的地資料夾內

Ver1 識別 (1:N)

Identification

```

# Record a WAV file
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 1024
RECORD_SECONDS = 10
WAVE_OUTPUT_FILENAME = "test.wav"

audio = pyaudio.PyAudio()

# start Recording
stream = audio.open(format=FORMAT, channels=CHANNELS, rate=RATE,
input=True, frames_per_buffer=CHUNK)
print ("recording...")
frames = []
for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
data = stream.read(CHUNK)
frames.append(data)
print ("finished recording")

# stop Recording
stream.stop_stream()
stream.close()
audio.terminate()

waveFile = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
waveFile.setnchannels(CHANNELS)
waveFile.setsampwidth(audio.get_sample_size(FORMAT))
waveFile.setframerate(RATE)
waveFile.writeframes(b''.join(frames))
waveFile.close()

# Voiceprint Identification
sourcepath = 'C:\\\\Users\\user\\Desktop\\_SR\\'
modelpath = "C:\\\\Users\\user\\Desktop\\_SR\\model\\"

gmm_files = [os.path.join(modelpath, fname) for fname in
os.listdir(modelpath) if fname.endswith('.gmm')]
files = [os.path.join(sourcepath, fname) for fname in os.listdir(sourcepath)
if fname.endswith('.wav')]
models = [pickle.load(open(fname, 'rb')) for fname in gmm_files]
people = [fname.split("\\")[-1].split(".gmm")[0] for fname in gmm_files]

for f in files:
    #print(f)
    sr, y = read(f)
    features= Test_MFCC_delta2(sr,y)
    scores= None
    log_likelihood= np.zeros(len(models))

```

```
for i in range(len(models)):
    gmm= models[i]          #checking with each model one by one
    scores= np.array(gmm.score(features))
    log_likelihood[i] = scores.sum()

winner = np.argmax(log_likelihood)

answer = people[winner]

print("\tdetected as - ", answer)
image1 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\'+answer+'.jpg', 'r')
```

```
plt.figure()
imshow(np.asarray(image1))
plt.show()
```

Example Output

detected as -

#牽涉肖像權，在這就不顯示照片囉~

Ver2 驗證 (1:1)

Voiceprint Verification

```
# Record a WAV file
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 1024
RECORD_SECONDS = 5
WAVE_OUTPUT_FILENAME = "test.wav"

audio = pyaudio.PyAudio()

# start Recording
stream = audio.open(format=FORMAT, channels=CHANNELS, rate=RATE,
input=True, frames_per_buffer=CHUNK)
print ("recording...")
frames = []
for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    frames.append(data)
print ("finished recording")

# stop Recording
stream.stop_stream()
stream.close()
audio.terminate()

waveFile = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
waveFile.setnchannels(CHANNELS)
waveFile.setsampwidth(audio.get_sample_size(FORMAT))
waveFile.setframerate(RATE)
waveFile.writeframes(b''.join(frames))
waveFile.close()

sourcepath = "C:\\Users\\user\\Desktop\\_SR\\"
modelpath = "C:\\Users\\user\\Desktop\\_SR\\model\\"
files = [os.path.join(sourcepath, f) for f in os.listdir(sourcepath) if
f.endswith(".wav")]
```

```

gmm_files = [os.path.join(modelpath,fname) for fname in
os.listdir(modelpath) if fname.endswith('.gmm')]
models     = [pickle.load(open(fname,'rb')) for fname in gmm_files]
people     = [fname.split("\\")[1].split(".gmm")[0] for fname in gmm_files]

owner = '' #
for i in range(len(models)):
    if(people[i]==owner):
        owner_idx = i

for f in files:
    sr, y = read(f)
    features= Test_MFCC_delta2(sr,y)
    scores= None
    log_likelihood= np.zeros(len(models))

    for i in range(len(models)):
        gmm= models[i]          #checking with each model one by one
        scores= np.array(gmm.score(features))
        log_likelihood[i] = scores.sum()

    winner = np.argmax(log_likelihood)
    #print(f.split('\\')[1].split('.wav')[0])
    score = Similarity(log_likelihood[owner_idx], log_likelihood[winner],
people[owner_idx], people[winner])
    #print(log_likelihood[owner_idx], log_likelihood[winner])
    print('Detected as-', people[winner])
    print("Similarity: ", score ,'%')
    if score <= 80:
        image0 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\Error.jpg', 'r')
        plt.figure()
        imshow(np.asarray(image0))
        plt.show()
        print('Sorry, it belongs to '+owner+' instead of you. Please try
again QAQ')

        elif (score < 99 and score >=95):
            image1 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\'+owner+'.jpg', 'r')
            plt.figure()
            imshow(np.asarray(image1))
            plt.show()
            print('Almost Correct~ May I help you, '+owner+'?')

        elif (score < 95 and score >=90):
            image1 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\'+owner+'.jpg', 'r')
            plt.figure()
            imshow(np.asarray(image1))
            plt.show()
            print('Partial Correct~ May I help you, '+owner+'?\n')

```



```
elif (score < 90 and score >=80):
    image1 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\'+owner+'.jpg', 'r')
    plt.figure()
    imshow(np.asarray(image1))
    plt.show()
    print('You are maybe '+owner+' (?)\n')

else: # higher than 99
    image1 =
Image.open('C:\\Users\\user\\Desktop\\_SR\\photo\\'+owner+'.jpg', 'r')
    plt.figure()
```

```
imshow(np.asarray(image1))
plt.show()
print('Totally Correct~ May I help you, '+owner+'?\n')
```

這是上淳的建議，可以讓計算出來的相似程度分等級，來顯示驗證後的結果

相似程度	對應結果
99 以上	完全正確
95~99	大部分正確
90~95	部分正確
80~90	可能正確 (?)
80以下	錯誤

Example Output

#實際答案是彭瑋玉，但預測結果是蘇玫如。但由以上的等級對應，還是會顯示實際答案的照片

```
Detected as-
Similarity: 84.30089408216344 %
```

#牽涉肖像權，在這就不顯示照片囉~

```
You are maybe (?)
```

遇到的困難

- 訓練好的模型，有部分效果不彰：
非常嚴重：青宏、亭嘉、冠霖
有點嚴重：立全、建豪、威良、祐嘉
些微嚴重：玫如、昇頤
於是有重新訓練部分模型、Test data 重新切割等動作
- 測試資料空白音過多會影響模型判斷，所以盡量在10秒內多講點話，收集更多的聲紋特徵
- 測試資料盡量秒數在5秒以上，以免特徵收集不夠，而導致誤判
- 如果現場錄音跟當時幫大家錄音的聲音振幅差太多，也會導致誤判。故背景聲音、錄音者的狀況都要納入考量