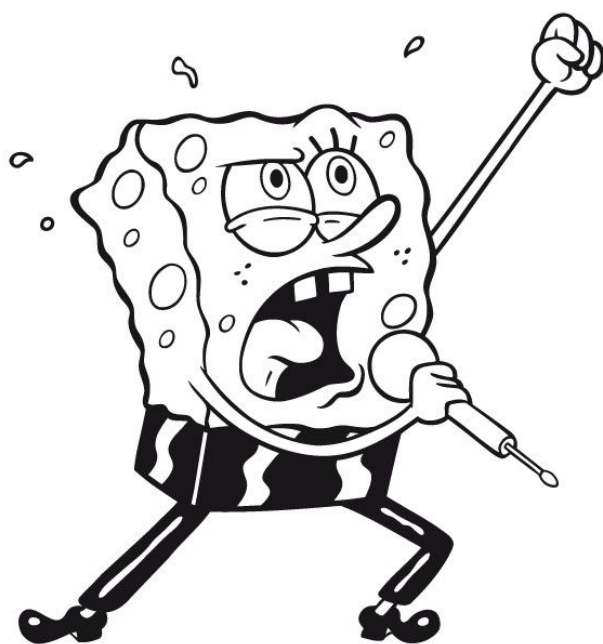


聽歌辨識歌手 SRS

第六組 李政憲 張哲郡 游登翔 劉彥麟 張友澤

June 11, 2019



1 introduction

1.1 Purpose

做出一個只要輸入進去一首歌的音檔 就能判別出唱的歌手是誰的project

1.2 Intended Audience and Reading Suggestions

intended Audience : 想要查詢自己聽到的歌是誰唱的的user

閱讀建議:對使用者而言要多注意”External interface”的部分
並且需要具備一定程式知識

1.3 Project Scope

這是個獨立的project, 其目的為使user能辨識出聽到的歌是來自於
哪位歌手

2 Overall Description

2.1 Product Perspective

這個project利用很多首歌訓練出了各種歌手的特徵值 再利用這些特徵值辨別出我們輸入進去的是哪位歌手

2.2 Product Functions

輸入進去歌曲的音檔 就能讓它判斷是我們給的歌手們(分類)中的哪一位

2.3 User Classes and Characteristics

聽線上電台的人, 實況的觀眾等等想知道當下聽到的歌是誰的歌的user

2.4 Operating Environment

Operating system: Windows 10

platform:pycharm python3

音訊處理:ffmpeg

2.5 Design and Implementation Constraints

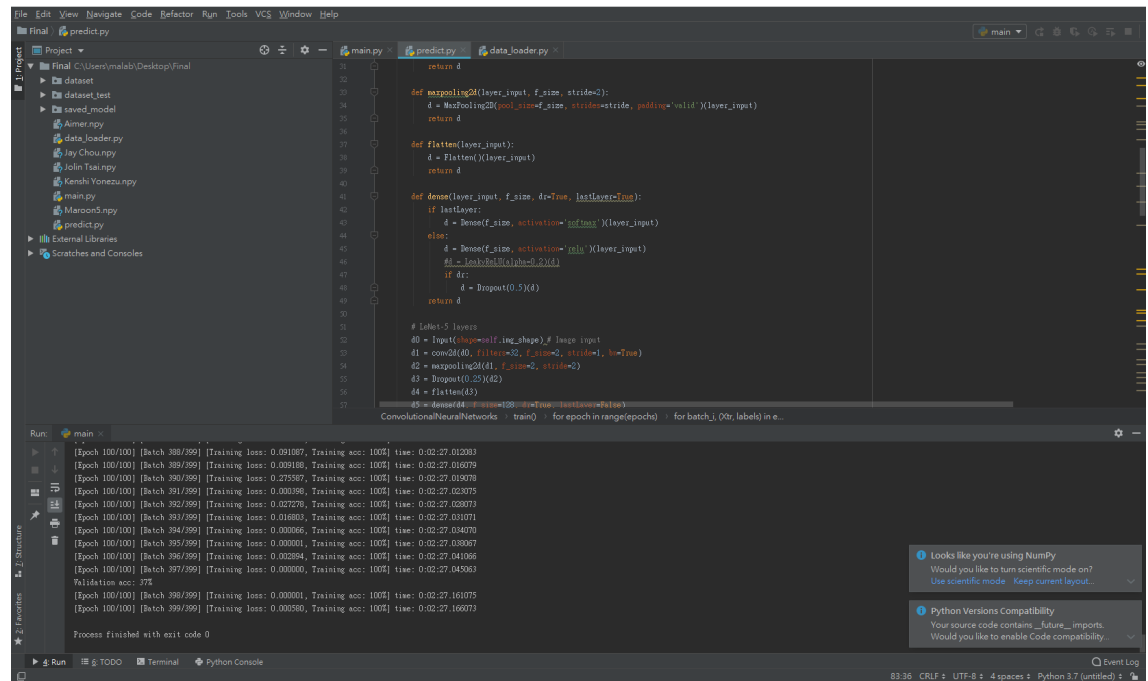
因爲大部分的歌都會有伴奏
有些甚至快大過歌手本身的歌聲，
所以實際再辨別時是有難度的
音檔轉換爲頻譜圖可能會失真

2.6 Assumptions and Dependencies

這份專案是先假設即使沒有去除掉背景音
電腦仍然能透過train來辨別出該歌手的特徵
並順利辨識

3 External Interface Requirements

3.1 User Interfaces



The screenshot displays an IDE with a project named 'final' containing files like 'dataset', 'dataset_test', 'saved_model', 'Aimer.npy', 'data_loader.py', 'Jay Chou.npy', 'Jolin Tsai.npy', 'Kenshi Yonezuka.npy', 'main.py', 'Maroon3.npy', and 'predict.py'. The 'predict.py' file is open, showing a neural network architecture with layers: MaxPooling2D, Flatten, Dense, and Dropout. The execution output in the Run console shows training progress over 100 epochs, with training loss decreasing from 0.091067 to 0.000000 and training accuracy increasing from 100% to 100%. The validation accuracy is 37%.

```
def maxpooling2d(layer_input, f_size, strides):  
    d = MaxPooling2D(pool_size=f_size, strides=strides, padding='valid')(layer_input)  
    return d  
  
def flatten(layer_input):  
    d = Flatten()(layer_input)  
    return d  
  
def dense(layer_input, f_size, dr=True, last_layer=True):  
    if last_layer:  
        d = Dense(f_size, activation='softmax')(layer_input)  
    else:  
        d = Dense(f_size, activation='relu')(layer_input)  
        d = LeakyReLU(alpha=0.2)(d)  
        if dr:  
            d = Dropout(0.5)(d)  
    return d  
  
# latent 5 layers  
d0 = Input(shape=(mlf_img_shape), # image input  
d1 = conv2d(d0, filters=32, kernel_size=3, strides=1, border_mode='same')  
d2 = maxpooling2d(d1, f_size=2, strides=2)  
d3 = Dropout(0.25)(d2)  
d4 = Flatten(d3)  
d5 = dense(d4, f_size=100, dr=True, last_layer=False)  
ConvolutionalNeuralNetworks train() for epoch in range(epochs) for batch_i, (0or, labels) in e...
```

Run: main
(Epoch 100/100) [Batch 388/399] [Training loss: 0.091067, Training acc: 100%] time: 0:02:27.013093
(Epoch 100/100) [Batch 389/399] [Training loss: 0.089180, Training acc: 100%] time: 0:02:27.016079
(Epoch 100/100) [Batch 390/399] [Training loss: 0.273367, Training acc: 100%] time: 0:02:27.019070
(Epoch 100/100) [Batch 391/399] [Training loss: 0.000396, Training acc: 100%] time: 0:02:27.023075
(Epoch 100/100) [Batch 392/399] [Training loss: 0.027278, Training acc: 100%] time: 0:02:27.026073
(Epoch 100/100) [Batch 393/399] [Training loss: 0.016803, Training acc: 100%] time: 0:02:27.031071
(Epoch 100/100) [Batch 394/399] [Training loss: 0.000366, Training acc: 100%] time: 0:02:27.034070
(Epoch 100/100) [Batch 395/399] [Training loss: 0.000001, Training acc: 100%] time: 0:02:27.038067
(Epoch 100/100) [Batch 396/399] [Training loss: 0.002884, Training acc: 100%] time: 0:02:27.041066
(Epoch 100/100) [Batch 397/399] [Training loss: 0.000000, Training acc: 100%] time: 0:02:27.045063
Validation acc: 37%
(Epoch 100/100) [Batch 398/399] [Training loss: 0.000001, Training acc: 100%] time: 0:02:27.161075
(Epoch 100/100) [Batch 399/399] [Training loss: 0.000000, Training acc: 100%] time: 0:02:27.166073
Process finished with ctrl code 0

3.2 Hardware Interfaces

windows 10//

3.3 Software Interfaces

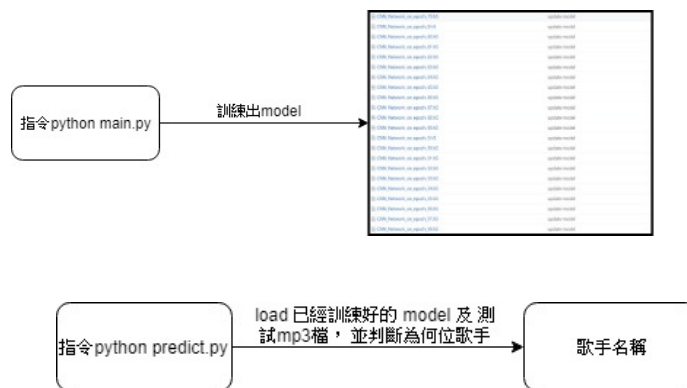
operating system:windows pycharm
ffmpeg

4 System Features

4.1 Description and Priority

將數據分類好放入 “/dataset”
將所有歌手的mp3檔進行資料前處理壓縮成.npy檔
使用.npy檔進行model的學習
透過學習成果分辨其他mp3檔的歌手

4.2 Stimulus/Response Sequences



4.3 Functional Requirements

REQ-1:新增歌手進入數據集

5 Other Nonfunctional Requirements

5.1 Performance Requirements

訓練歌手的mp3檔需200筆

Thank you for watching