Hướng dẫn cài đặt hệ thống “**Phân lớp một số làn điệu chèo và dân ca quan họ**”

Môi trường chạy: Linux Bash Shell trên hệ điều hành Windows 10.

Mã nguồn được viết bởi ngôn ngữ lập trình Python.

Tất cả mã nguồn và các file config có thể tải xuống tại địa chỉ <https://github.com/LaiVanHai/sourse_code_gr>

## Tạo thư mục làm việc

Trong thư mục dùng để chạy chương trình lần lượt tạo thêm các thư mục sau: cfg, gmm, gmm\_ch, gmm\_qh, lbl, lst, lst\_ch, lst\_qh, ndx, ndx\_ch, ndx\_qh, prm, res, res\_ch, res\_qh, rpt, test, test\_ch, test\_qh, wav, mergeData

Trong đó:

* Thư mục cfg dùng để lưu các file config của hệ thống, có thể tải xuống tại địa chỉ bên trên.
* Thư mục mergeData dùng để lưu kết quả của mỗi lần đánh giá chéo và tổng hợp các kết quả này lại với nhau.
* Thư mục wav: dùng để lưu các file âm thanh dùng để thử nghiệm hệ thống, có thể tải xuống tại địa chỉ: <https://drive.google.com/drive/folders/1lCacJtw53LCQIlszmIuHePhE3dtvTQVC?usp=sharing>

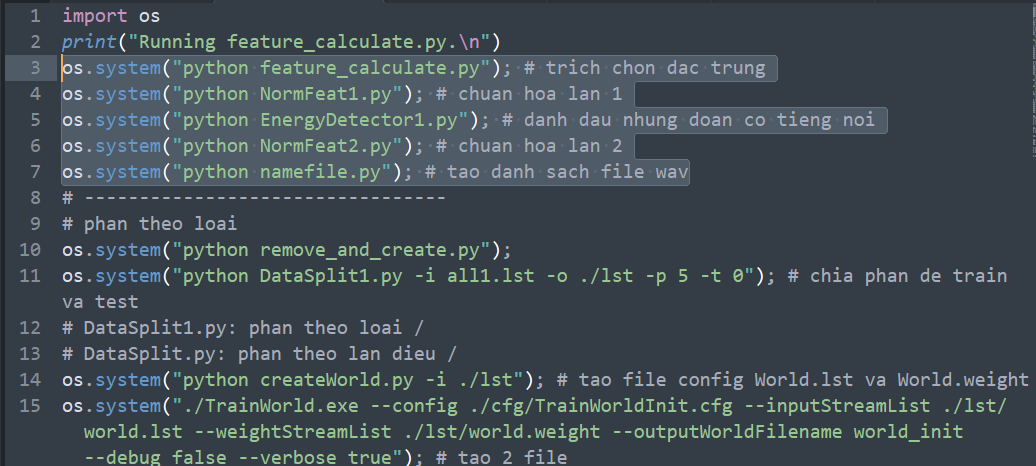
Bên ngoài thư mục chính copy các file trong thư mục **file\_exe** ra (bao gồm các file *TrainWord.exe, TrainTarget.exe, sfbcep.exe, EnergyDetector.exe, ComputeTest.exe, ComputeNorm.exe*)

## Giải thích ý nghĩa của các file chạy

Copy tất cả những file có đuôi **.py** trong thư mục **source\_code** ra thư mục làm việc chính.

### Thử nghiệm với dữ liệu thử nghiệm không nằm trong tập huấn luyện

* File chạy chính là file *runAlize.py*, dưới đây sẽ là phần giải thích ý nghĩa của từng chương trình con trong file *runAlize.py.*
* Trước khi chạy file này nên đọc phần dưới để config lại những file cần thiết

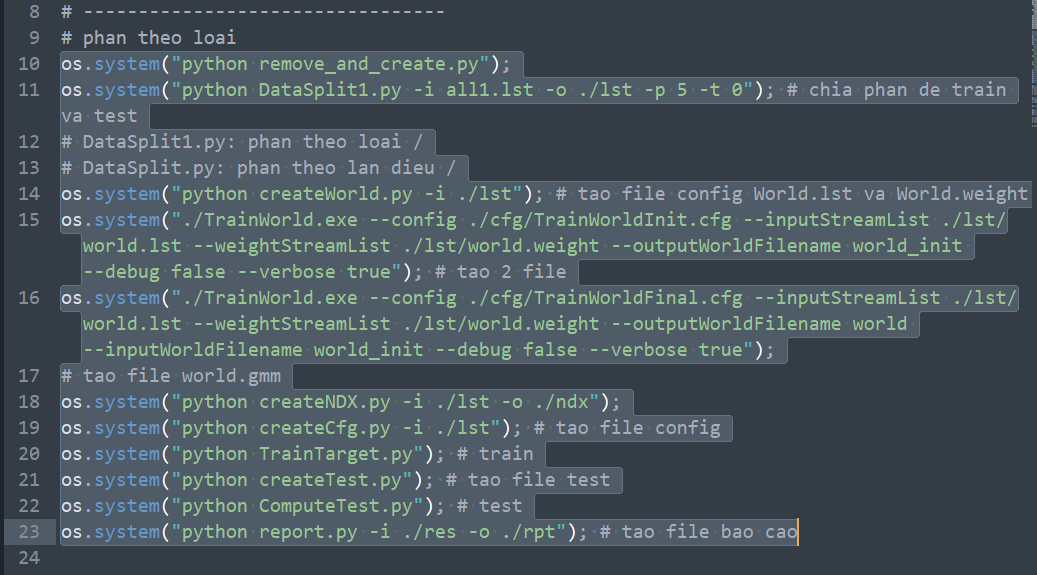


Từ dòng code thứ 3 đến dòng thứ 7 là lệnh chạy các chương trình liên quan đến việc tạo danh sách các file wav, chuẩn hóa và trích chọn đặc trưng của những file wav.

Đoạn code này có thể chỉ cần chạy một lần đầu tiên, còn các lần chạy sau có thể comment dòng code này lại để hạn giảm thiểu thời gian chạy.

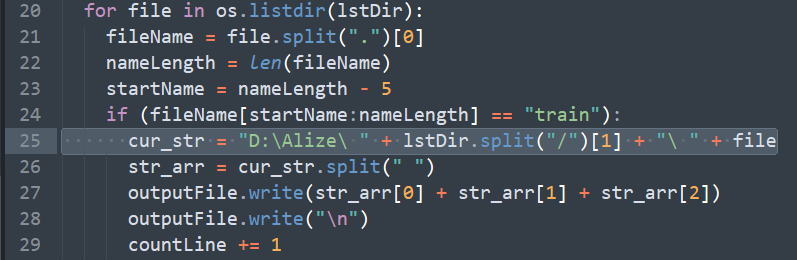
**Kết quả:**

* Sinh ra các file *all.lst* và *all1.lst* trong thư mục **lst**.
* Sinh ra các file có đuôi *.prm, .norm.prm, .enr.prm* trong thư mục **prm**
* Sinh ra các file có đuôi *.lbl* trong thư mục **lbl**



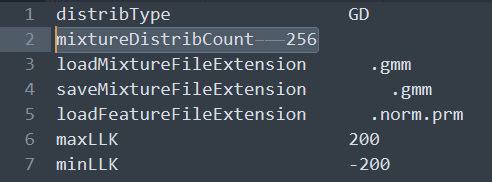
Từ dòng code thứ 10 đến dòng code thứ 23 làm nhiệm vụ giải quyết bài toán “**Phân lớp sơ bộ**” - xác định một làn điệu là chèo hay dân ca quan họ.

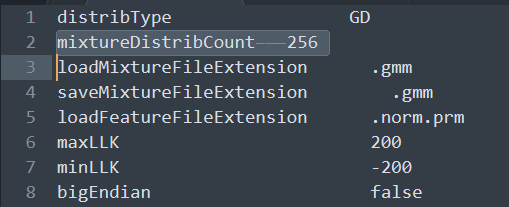
* Dòng code “**python remove\_and\_create.py**” để xóa đi những file của phiên làm việc trước, bởi đây là phương pháp đánh giá chéo phải chạy 5 lần mới có thể tổng hợp lại được kết quả cuối cùng. Sau khi chạy lần đánh giá đầu tiên có thể comment 5 dòng code từ dòng thứ 3 -> 7, các lần chạy sau thì chỉ cần chạy từ dòng code thứ 10 này.
* Dòng code "**python DataSplit1.py -i all1.lst -o ./lst -p 5 -t 1**" để chia dữ liệu thành 5 phần và lấy lần 1 để làm test, còn ví dụ nếu là “-t 2” thì hệ thống sẽ lấy phần thứ 2 làm test. Trên hình để là “-t 0” có nghĩa là chỉ cần tạo file train mà không cần file test.
* Dòng code “**python createWorld.py -i ./lst**” chỉ để tạo tự động 2 file World.lst và World.weight trong thư mục lst. Tuy nhiên, ở đây có một chút điều cần chú ý. Mở file createWorld.py, ta chú ý vào dòng 25



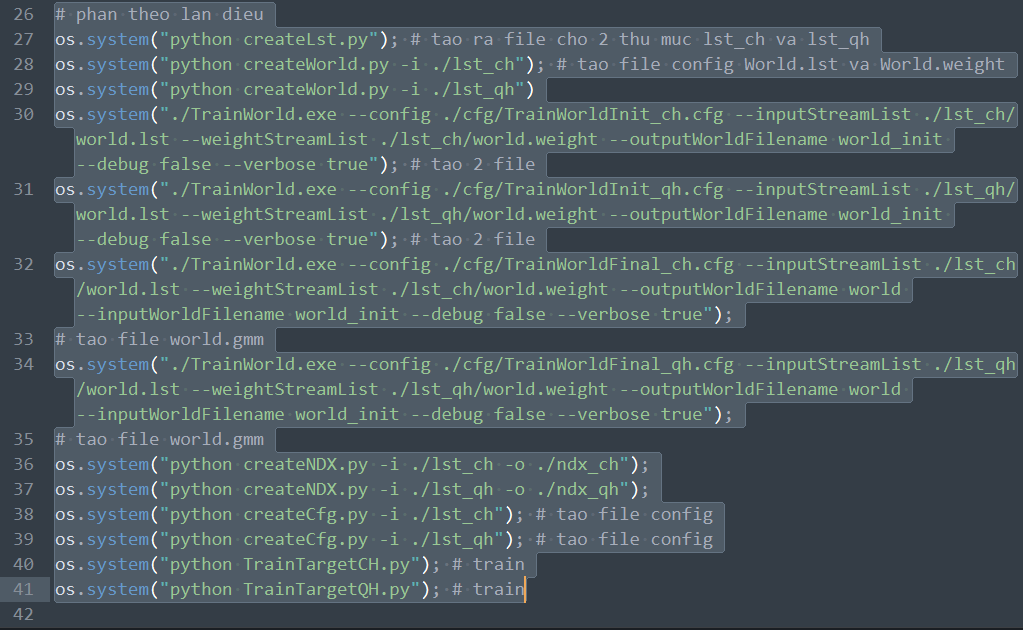
Ở đây thư mục làm việc chính được đặt trong ổ D và có tên là Alize. Với các các tên thư mục và vị trí đặt khác nhau thì cần phải sửa tại đây.

* Từ dòng code thứ 15 đến 20 trong hình dùng để tạo các file config và huấn luyện GMM, muốn tùy biến các giá trị M thì có thể sửa trong file *./cfg/TrainWorldInit.cfg* và *./cfg/TrainWorldFinal.cfg*. (dòng 2)

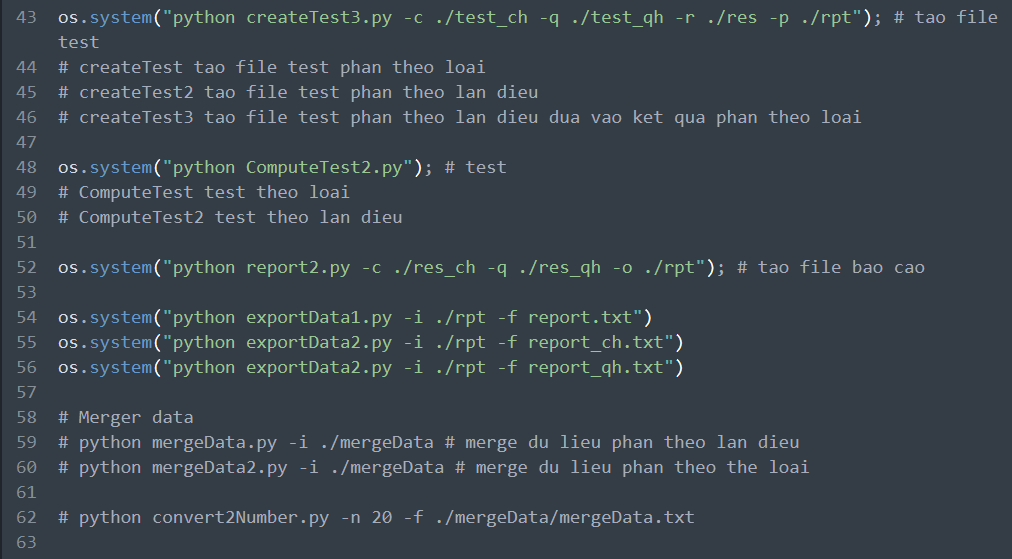




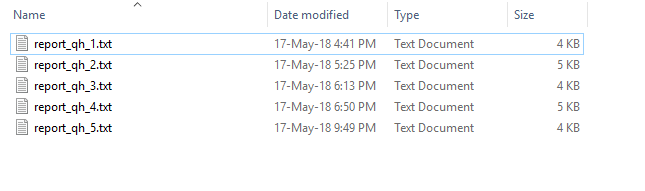
* Từ dòng code 21 và 22 dùng để tạo file thử nghiệm và tiến hành thử nghiệm
* Dòng code thứ 23 dùng để export các kết quả trong thư mục **res** ra thành file tổng kết kết quả đo độ chính xác theo phần trăm trong thư mục **rpt**, sau khi chạy xong dòng code này thì chúng ta sẽ thu được file **report.txt**



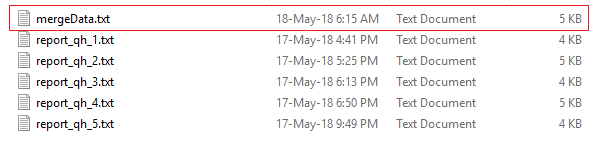
* Dòng code 27 dùng để chia tập dữ liệu ban đầu thành 2 phần chèo và dân ca quan họ để phục vụ cho việc giải quyết bài toán “**Phân lớp cụ thể**” – Xác định cụ thể tên của một làn điệu chèo hoặc quan họ.
* Từ dòng code thứ 28 đến dòng code thứ 39, ý nghĩa vẫn giống như phần trên. Tuy nhiên nếu khi thay đổi số thành phần Gauss thì phải sửa trong 4 file *./cfg/TrainWorldInit\_ch.cfg, ./cfg/TrainWorldInit\_qh.cfg, ./cfg/TrainWorldFinal\_ch.cfg, ./cfg/TrainWorldFinal\_qh.cfg*. Vị trí cần sửa giống như khi sửa file ./cfg/TrainWorldInit.cfg và ./cfg/TrainWorldFinal.cfg.
* Dòng code thứ 40 và 41 là tiến hành huấn luyện GMM cho 2 lớp chèo và dân ca quan họ



* Dòng code thứ 43 dùng để tạo file test từ kết quả thu được từ bài toán phân lớp sơ bộ và lưu vào thư mục **test\_ch** và **test\_qh**
* Dỏng code 48 dùng để tiến hành thử nghiệm
* Dòng code 52 dùng để tổng hợp kết quả thu được theo phần trăm và lưu vào thư mục rpt, sau khi chạy xong dòng code này sẽ thu được 2 file **report\_ch.txt** và **report\_qh.text**.
* Sau khi chạy đến đây thì trong thư mục rpt chúng ta có 3 file **report.txt** – lưu lại kết quả phân lớp sơ bộ, file **report\_ch.txt** – lưu lại kết quả phân lớp cụ thể làn điệu chèo, file **report\_qh.txt** – lưu lại kết quả phân lớp cụ thể làn điệu dân ca quan họ.
* Dòng code từ 54 đến 56 thực hiện tính trung bình độ chính xác của kết quả thu được, độ chính xác trung bình là bao nhiêu %, làn điệu nào nhận đúng hoàn toàn, làn điệu nào nhận sai một phần, làn điệu nào nhận sai 1 phần.
* Sau khi thực hiện xong các dòng code từ 54 đến 56 thì nên đổi tên các file *report.txt, report\_ch.txt* và *report\_qh.txt* thành *report\_SoPhanLayRaDeTest.txt, report\_ch\_SoPhanLayRaDeTest.txt, report\_qh\_SoPhanLayRaDeTest.txt*. Ví dụ, lấy phần 1 ra để test thì đổi tên các file thành report\_1.txt, report\_ch\_1.txt, report\_qh\_1.txt. Sau đó lưu lại để sau này dùng để tổng hợp kết quả.
* Dòng code 59 và 60 bị comment lại bởi nó chỉ nên thể chạy sau khi đã chạy 5 lần đánh giá chéo khi đảm bảo tất cả các file đều đã được đưa vào huấn luyện và thử nghiệm. Sau 5 lần thử nghiệm và đổi tên file theo quy tắc nêu ra ở trên thì ta sẽ thu được các file report như sau: report\_1.txt, report\_2.txt,.., report\_5.txt, report\_ch\_1.txt, report\_ch\_2.txt, report\_ch\_5.txt, report\_qh\_1.txt,…report\_qh\_5.txt.
  + Dòng code thứ 59 dùng để tổng hợp kết quả của bài toán phân lớp cụ thể, trước khi chạy dòng code này ta copy các file report phân lớp cụ thể theo từng làn điệu vào thư mục mergeData, sau khi copy xong thư mục sẽ như hình



Ở đây tiến hành tổng hợp kết quả phân lớp cụ thể làn điệu dân ca quan họ. Sauk hi chạy xong ta sẽ thu được file mergeData.txt là tổng hợp kết quả của 5 file trên.

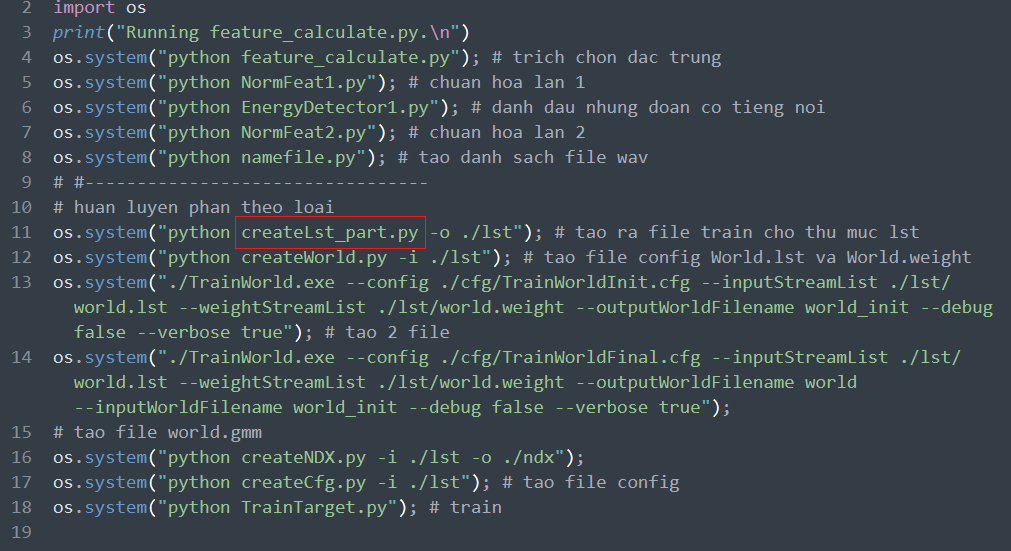


Sau khi thu được file tổng hợp mà muốn chạy tiếp những lần khác thì chương trình chỉ chạy đúng khi số lượng file trong thư mục là 5 (5 kết quả từ 5 lần đánh gia).

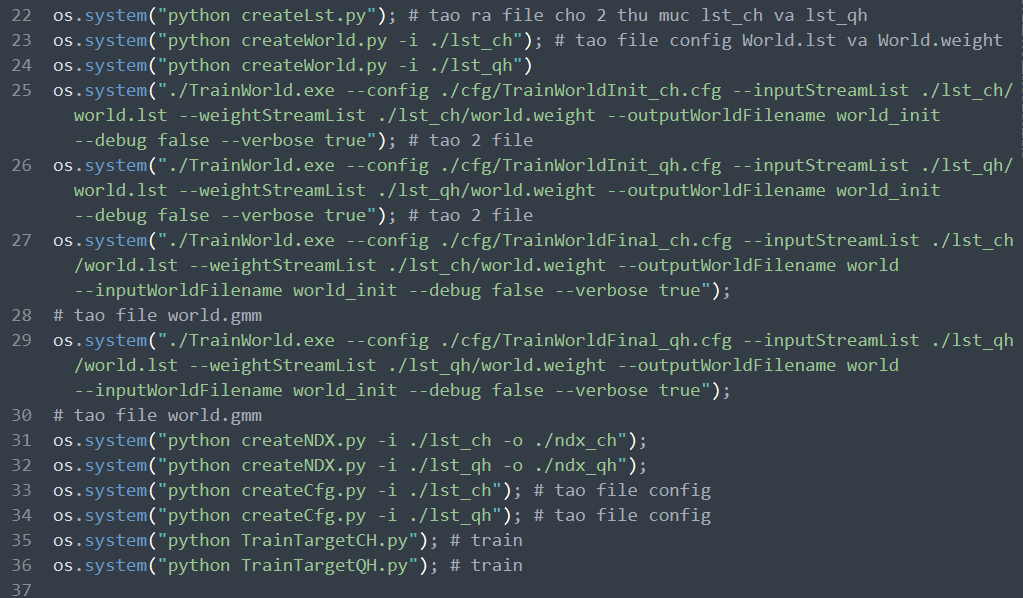
* + Dòng code thứ 60 dùng để tổng hợp kết quả của bài toán phân lớp sơ bộ, cách thức thực hiện như tổng hợp kết quả của bài toán phân lớp cụ thể.
* Vì kết quả tổng hợp biểu thị ở dạng % nên muốn đổi sang định dạng số lượng file để vẽ ma trận nhầm lần thì chúng ta sử dụng dòng code thứ 62 “**python convert2Number.py -n 20 -f ./mergeData/mergeData.txt**”, với “-n 20” là biểu thị có 20 file test, “-f ./mergeData/mergeData.txt” là dẫn đường dẫn tới file ./mergeData/mergeData.txt, sau khi chạy xong thì kết quả cũng được lưu tại thư mục của file muốn chuyển đổi, cụ thể ở đây kết quả thu được lưu tại thư mục mergaData.

### Thử nghiệm với dữ liệu cắt ngắn từ tập huấn luyện

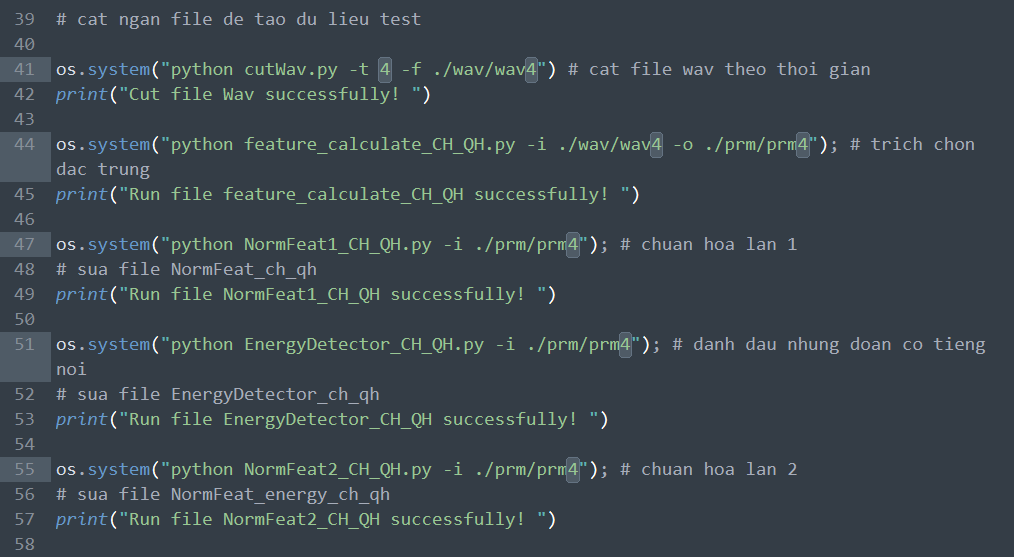
* File chạy chính là file runAlize2.py
* Trước khi chạy file này nên đọc phần dưới để config lại những file cần thiết



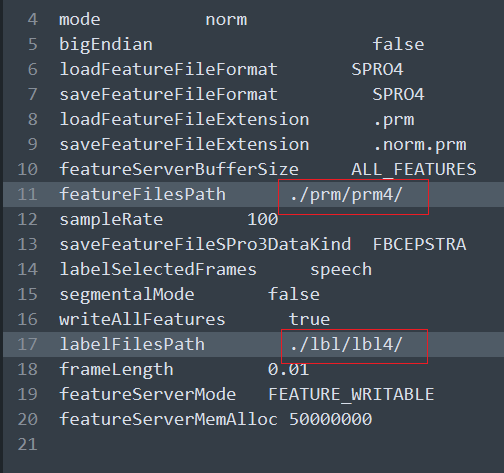
* Từ dòng code thứ 4 đến dòng code 18, ý nghĩa các lệnh vẫn giống như trong phần 2.1, trong dòng code thứ 11 có sự thay đổi file chạy bởi với việc thử nghiệm với dữ liệu cắt ngắn từ tập huấn luyện thì sẽ không có giai đoạn chia dữ liệu mẫu thành 2 phần huấn luyện và thử nghiệm, dó đó sau khi chạy dòng code thứ 11 thì tất cả các file từ tập dữ liệu mẫu sẽ được phân vào danh sách những file dùng để huấn luyện tương ứng với 2 thể loại chèo và quan họ.



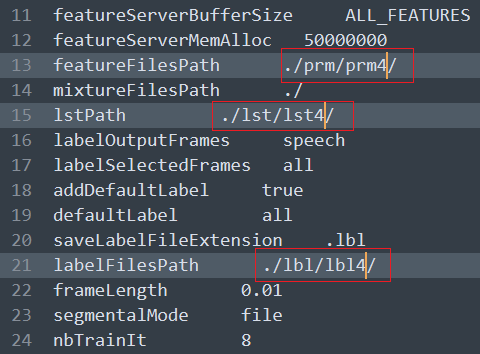
* Từ dòng code thứ 22 đến dòng code 36 là giai đoạn huấn luyện phục vụ cho bài toán phân lớp cụ thể, ý nghĩa của các dòng lệnh, giống như trong phần 2.1.



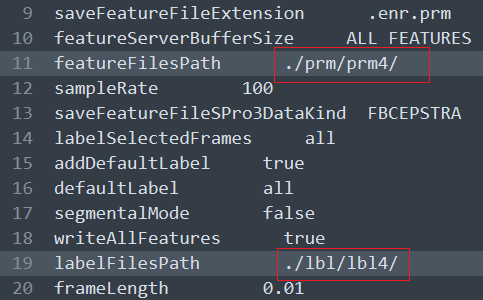
* Từ dòng code thứ 41 đến dòng 57, là quá trình cắt ngắn file từ tập huấn luyện để tạo dữ liệu thử nghiệm.
  + **python cutWav.py -t 4 -f ./wav/wav4** : ở đây số 4 đại diện cho độ dài file thử nghiệm là 4s, sau khi chạy dòng lệnh này thì trong thư mục wav sẽ tạo ra thư mục wav4 chứa những file âm thanh được cắt ngắn từ tập dữ liệu mẫu có độ dài là 4s, tương tự đối với các độ dài khác ta chỉ cần thay “-t 4” bằng độ dài mong muốn thì sẽ thực hiện cắt được, còn “-f wav/wav4” là chỉ định thư mục sẽ lưu những file sau khi cắt.
  + Trên hình là thử nghiệm với độ dài file là 4s, nếu muốn thay bằng 6s thì ta tiến hành thay tất cả số 4 bên trên bằng số 6, tương tự với các độ dài khác. Tuy nhiên, ở đây khi thay đổi độ dài file thử nghiệm chúng ta cũng phải thay đổi file config, những file config cần thay đổi được comment tại dòng code thứ 48, 52 và 56 của hình trên.
* Với file **./cfg/NormFeat\_ch\_qh.cfg** thì dòng cần sửa sẽ là dòng 11 và 17, thay số 4 bằng độ dài tương ứng của file thử nghiệm

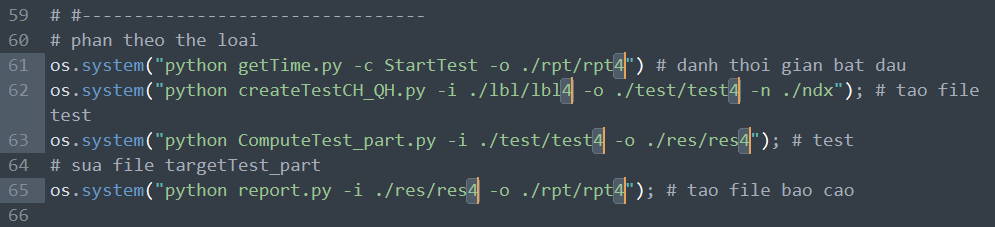


* Với file **./cfg/ EnergyDetector\_ch\_qh.cfg**, nội dung cần sửa sẽ là dòng 13, 15 và 21

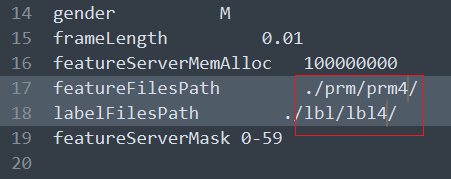


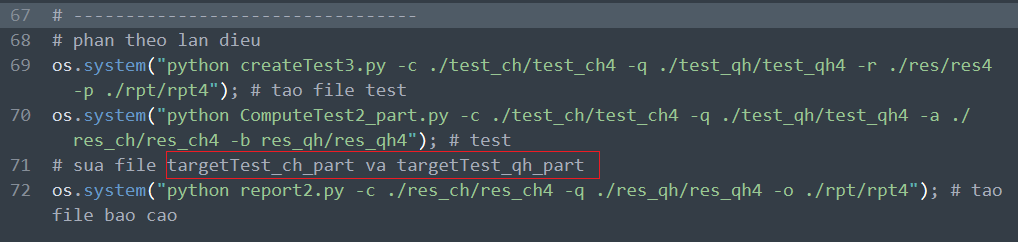
* Với file **./cfg/ NormFeat\_energy\_ch\_qh.cfg** thì nội dung cần sửa sẽ là dòng 11 và 19



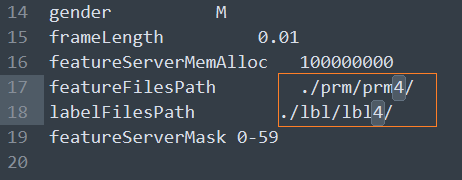


* Dòng code 62 là chương trình dùng để lấy thời gian của hệ thống, *“****python getTime.py -c StartTest -o ./rpt/rpt4***”, thì chương trình sẽ tạo một file là time.txt trong thư mục ./rpt/rpt4, có nội dung “*StartTest Thời\_gian\_bắt\_đầu\_chạy\_chương\_trình\_này*” .
* Từ dòng code 62 đến dòng 65 là quá trình tạo file thử nghiệm, tiến hành thử nghiệm và tổng hợp kết quả ở bài toán **phân lớp sơ bộ**, trước khi chạy chúng ta cũng phải sửa những số 4 trong hình sang độ dài của file thử nghiệm thực hiện cắt ở bên trên, ví dụ trong hình đang thử nghiệm với độ dài file là 4s, còn nếu là 6s thì ta thay tất cả những số 4 trong hình bằng số 6. Ngoài ra, cũng cần sửa file config **./cfg/ targetTest\_part.cfg**, nội dung cần sửa như hình bên dưới, với số 4 là độ dài theo thời gian của file thử nghiệm.

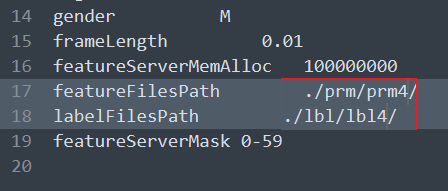


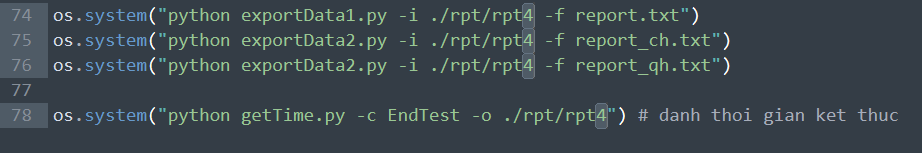


* Từ dòng code thứ 69 đến dòng code 72 là quá trình tạo file thử nghiệm từ kết quả thu được của bài toán phân lớp sơ bộ để chuẩn bị cho việc thử nghiệm hệ thống phân lớp cụ thể theo làn điệu.
* Trước khi chạy đoạn code này chúng ta cũng cần sửa 2 file config để đảm bảo việc thử nghiệm được chính xác, đó là file **./cfg/targetTest\_ch\_part.cfg** và **./cfg/targetTest\_qh\_part.cfg**
  + Với file **./cfg/targetTest\_ch\_part.cfg** ta sửa dòng 17 và 18



* + Với file **./cfg/targetTest\_qh\_part.cfg** ta sửa dòng 17 và 18





* Từ dòng code thứ 74 -> dòng code 76 là thực hiện việc tổng hợp kết quả trung bình từ các file report.
* Dòng code thứ 78, sửa vào nội dung của file time.txt bên trên để ghi thời gian kết thúc việc thử nghiệm.