ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN: MÁY HỌC**

**ĐỀ TÀI: DỰ ĐOÁN ĐIỂM TỪ DỮ LIỆU WECODE,**

**NHẬN DIỆN HÃNG XE QUA ẢNH CHỤP**

**Giảng viên hướng dẫn: PGS. TS. Lê Đình Duy**

**THS. Phạm Nguyễn Trường An**

**Lớp: CS114.P11**

**Sinh viên thực hiện:**

|  |  |
| --- | --- |
| **MSSV** | **Họ và tên** |
| 22520550 | Lương Anh Huy |
| 22520967 | Hồng Khải Nguyên |

**Tháng 1/2025**

\* Link github bài làm của nhóm: <https://github.com/anhhuyluong/CS114.P11>

\* Cấu trúc của thư mục: Gồm 2 folder chính là **CarBrandDetection** và **ScorePrediction**

**- Đối với CarBrandDetection**

**+** Gồm 2 file là “CarBrandDetection\_Final.ipynb”, “CarBrandDetection\_PreparingData.ipynb”, **chạy file PreparingData trước**

+ Đối với file PreparingData, thay thế đường dẫn **dataset\_path** thành đường dẫn tới dữ liệu các thư mục hãng xe trong máy, giả sử lưu folder của các hãng xe ở ổ D, thư mục cha là CS114 thì đường dẫn là "D:/CS114"

+ Đối với file Final, chỉ chạy các cell chứa phần import thư viện và các cell bên trong 2 phần Markdown là “**Chuẩn bị dữ liệu**” và “**Third (Final Model)”**

**- Đối với ScorePrediction**

**+** Gồm một file là ScorePrediction.ipynb

+ Thay thế đường dẫn ở các dòng pd.read\_csv ở cell thứ hai từ trên xuống thành đường dẫn tới các file csv về các cột điểm, sau đó chạy tất cả các cell

+ Link drive dùng để tải 5 file csv: <https://drive.google.com/drive/folders/14-rK2QsUjbp-5kLV7IFMvsZ3JBbgQlQW?usp=sharing>

**A screen shot of a computer code

Description automatically generated**

Đường dẫn cần thay thế tùy thuộc vào nơi lưu trữ dữ liệu

**LỜI CẢM ƠN**

Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Duy và thầy An vì sự hỗ trợ, chỉ bảo tận tình trong suốt quá trình thực hiện dự án. Nhờ sự hướng dẫn cẩn thận và những góp ý quý báu từ các thầy, nhóm chúng em không chỉ cải thiện được bài làm của mình mà còn học hỏi được rất nhiều điều

Mặc dù kết quả đạt được của nhóm chúng em chưa thực sự nổi bật hoặc xuất sắc như mong đợi, nhưng chúng em tự hào vì đã nỗ lực hết mình trong từng giai đoạn của đồ án. Nhóm đã cố gắng lắng nghe, ghi nhận và tiếp thu mọi ý kiến đóng góp từ các thầy để cải thiện sản phẩm của mình.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành đến thầy Duy và thầy An, và hy vọng sẽ tiếp tục nhận được sự đồng hành, chỉ dẫn từ các thầy trong những chặng đường học tập tiếp theo.

Mục lục

[1 DỰ ĐOÁN ĐIỂM TỪ DỮ LIỆU WECODE 5](#_Toc188304331)

[1.1 CẬP NHẬT SAU VẤN ĐÁP 5](#_Toc188304332)

[1.2 GIỚI THIỆU BÀI TOÁN 5](#_Toc188304333)

[1.3 CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN 5](#_Toc188304334)

[1.3.1. Feature Engineering 5](#_Toc188304335)

[1.3.2. Merged dataset và chia tập train, test 7](#_Toc188304336)

[1.3.3. Khởi tạo model và feature mapping cho từng điểm 7](#_Toc188304337)

[1.3.4. Dự đoán các điểm số và kết quả R2 + correlation matrix 9](#_Toc188304338)

[1.4 KẾT QUẢ ĐẠT ĐƯỢC VÀ NHẬN XÉT 13](#_Toc188304339)

[1.5 PHÂN TÍCH KẾT QUẢ 14](#_Toc188304340)

[1.6 TÀI LIỆU THAM KHẢO 14](#_Toc188304341)

[2 NHẬN DIỆN HÃNG XE QUA ẢNH CHỤP 15](#_Toc188304342)

[2.1 CẬP NHẬT SAU VẤN ĐÁP 15](#_Toc188304343)

[2.2 GIỚI THIỆU BÀI TOÁN 15](#_Toc188304344)

[2.3 CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN 15](#_Toc188304345)

[2.3.1. THỐNG KÊ DỮ LIỆU 15](#_Toc188304346)

[2.3.2. PHÂN CHIA DỮ LIỆU 16](#_Toc188304347)

[2.3.3. TRỰC QUAN HÓA DỮ LIỆU 19](#_Toc188304348)

[2.3.4. PHÂN CỤM DỮ LIỆU 19](#_Toc188304349)

[2.3.5. HUẤN LUYỆN MÔ HÌNH 20](#_Toc188304350)

[2.4 KẾT QUẢ ĐẠT ĐƯỢC VÀ NHẬN XÉT 22](#_Toc188304351)

[2.5 PHÂN TÍCH CHUYÊN SÂU 23](#_Toc188304352)

[2.5.1. Phân cụm dữ liệu 23](#_Toc188304353)

[2.5.2. Xây dựng và huấn luyện mô hình 26](#_Toc188304354)

[2.6 TÀI LIỆU THAM KHẢO 28](#_Toc188304355)

# DỰ ĐOÁN ĐIỂM TỪ DỮ LIỆU WECODE

## CẬP NHẬT SAU VẤN ĐÁP

- Thay đổi việc lựa chọn đặc trưng cho từng cột điểm (Phần **Các bước nhóm đã thực hiện)**

- Thêm một vài đặc trưng mới cho mô hình (Phần **Các bước nhóm đã thực hiện – Feature Engineering)**

- Sử dụng thêm một mô hình so với lần vấn đáp (Phần **Các bước nhóm đã thực hiện)**

- Điểm đánh giá trên Wecode có phần cải thiện (Phần **Kết quả đạt được và nhận xét)**

## GIỚI THIỆU BÀI TOÁN

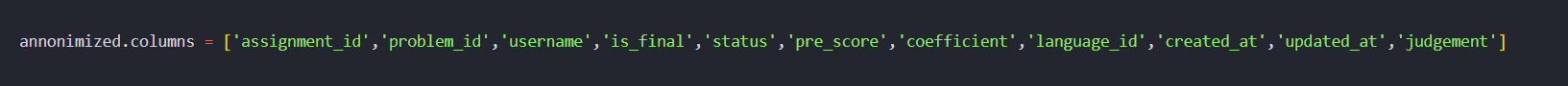
Dự đoán điểm từ dữ liệu wecode là bài toán thuộc lĩnh vực Máy học, sử dụng các mô hình hồi quy để dự đoán các cột điểm cho từng sinh viên dựa trên dữ liệu làm bài và nộp bài trên hệ thống chấm bài tự động Wecode. Bài toán này sẽ tập trung vào việc lựa chọn đặc trưng phù hợp cho từng cột điểm và huấn luyện các mô hình hồi quy để dự đoán sao cho kết quả R2-Score được tốt nhất

## CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN

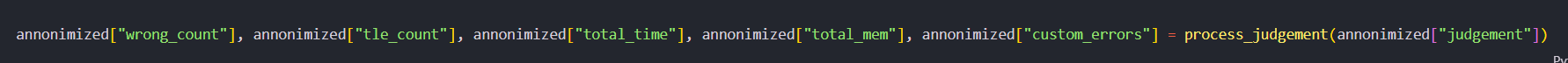
- Load dữ liệu từ các file csv gồm “annonimized.csv”, “qt-public.csv”, “th-public.csv”, “ck-public.csv” và “tbtl-public.csv”, sau đó kiểm tra và loại bỏ các dòng dữ liệu bị thiếu trong các file csv ở các cột điểm như “qt”, “th”, “ck” và “tbtl”

### Feature Engineering

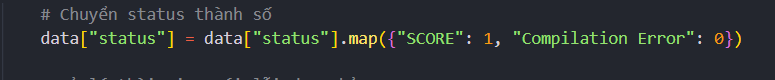
-Đầu tiên, nhóm sẽ đặt tên lại các cột trong file annonimized.csv để tiện cho sau này sử dụng



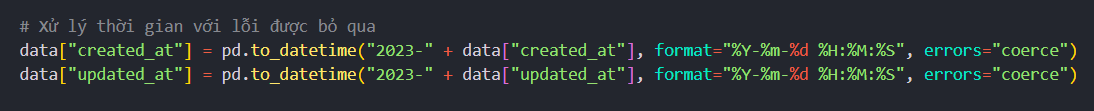
Tên các cột trong file annonimized.csv

-Sau đó, nhóm sẽ xử lý phần cột ‘judgement’ trong file annonimized.csv, tách cột này thành 5 cột dữ liệu lần lượt là: *‘wrong\_count’, ‘tle\_count’, ‘total\_time’, ‘total\_mem’, ‘count\_errors’* qua một hàm nhóm tạo ra để xử lý việc tách này  


-Tiếp đến, nhóm bắt đầu feature engineer các feature bao gồm:  
 + Đầu tiên là mã hóa số cột status như sau:



+ Giả định rằng dataset này đang trong năm 2023 và định dạng lại thời gian:



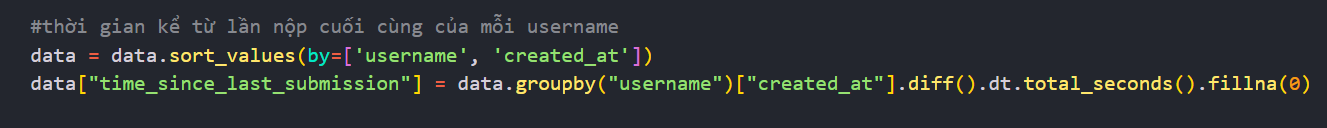
+ Tạo ra các feature liên quan đến thời gian và error đã trích suất được từ phần xử   
lý cột ‘judgement’:  
A screen shot of a computer code

Description automatically generated

+ Các feature mà nhóm bổ sung sau buổi vấn đáp:  
Feature đầu tiên tính tổng thời gian của mỗi username dành cho mỗi bài toán



Feature tiếp theo tính khoảng thời gian kể từ lần nộp cuối cùng của mỗi username



Feature sau tính tỉ lệ thành công cuối cùng của mỗi username và trung bình thời gian và lỗi theo từng bài toán

A screen shot of a computer code

Description automatically generated

Hai feature cuối cùng tính thời gian có trọng số và tính số lần nộp mỗi ngày của username đó:

A computer code with green and yellow text

Description automatically generated

### Merged dataset và chia tập train, test

- Phần merged các dataset lại thì nhóm sẽ merged trên username:

A computer code with colorful text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

- Về việc chia train test, thì do đây là bài dự đoán các điểm bị mất của username nên tập train sẽ chứa tất cả dữ liệu của user nào có đầy đủ 4 cột điểm, còn tập test sẽ là phần còn lại:

A computer screen shot of a computer code

Description automatically generated

### Khởi tạo model và feature mapping cho từng điểm

- Trước hết, nhóm sẽ khởi tạo scaler để trước khi chạy model dùng scaler cho chuẩn hóa dữ liệu. đây là hàm scaler của nhóm:

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated- Feature mapping là phần mà nhóm chọn tay các feature được sử dụng trong quá trình train model và predict, đây là feature mapping mà nhóm dùng để có được điểm số trên wecode:

A screen shot of a computer program

Description automatically generated

Hàm train\_and\_predict dùng cho huấn luyện và dự đoán điểm

### Dự đoán các điểm số và kết quả R2 + correlation matrix

- Nhóm sử dụng mô hình RandomForestRegressor cho dự đoán điểm quá trình và mô hình XGBoost cho 3 cột điểm còn lại

A screen shot of a computer code

Description automatically generated

R2-Score cho dự đoán điểm quá trình

A screenshot of a computer screen

Description automatically generated

Correlation Matrix của cột điểm quá trình

A computer screen shot of a program

Description automatically generated

R2-Score cho dự đoán điểm thực hành

A screenshot of a computer

Description automatically generated

Correlation matrix của cột điểm thực hành

A computer screen shot of a program code

Description automatically generated

R2-Score cho dự đoán điểm cuối kỳ

A screenshot of a computer

Description automatically generated

Correlation Matrix cho cột điểm cuối kỳ

A screen shot of a computer

Description automatically generated

R2-Score cho dự đoán điểm trung bình tích lũy

A screenshot of a computer

Description automatically generated

Correlation Matrix cho cột điểm trung bình tích lũy

## KẾT QUẢ ĐẠT ĐƯỢC VÀ NHẬN XÉT

- Kết quả **nhóm** đạt được **trên wecode** là: TBTL là 7/100, Thực hành là 18/100, Quá trình là 6/100, Cuối kì là 18/100

- Nhận xét: Trong Correlation Matrix ở các từng cột điểm, có một số feature bị khuyết thiếu. Qua tìm hiểu, nhóm xác định một số nguyên nhân chính như sau:

+ Không có giá trị tương quan đáng kể hoặc giá trị tương quan gần bằng 0

+ Dữ liệu bị thiếu hoặc mất mát

=> Những vấn đề này có thể ảnh hưởng đến R2-Score, vì một số đặc trưng tuy có mặt nhưng không mang lại hiệu quả khi sử dụng

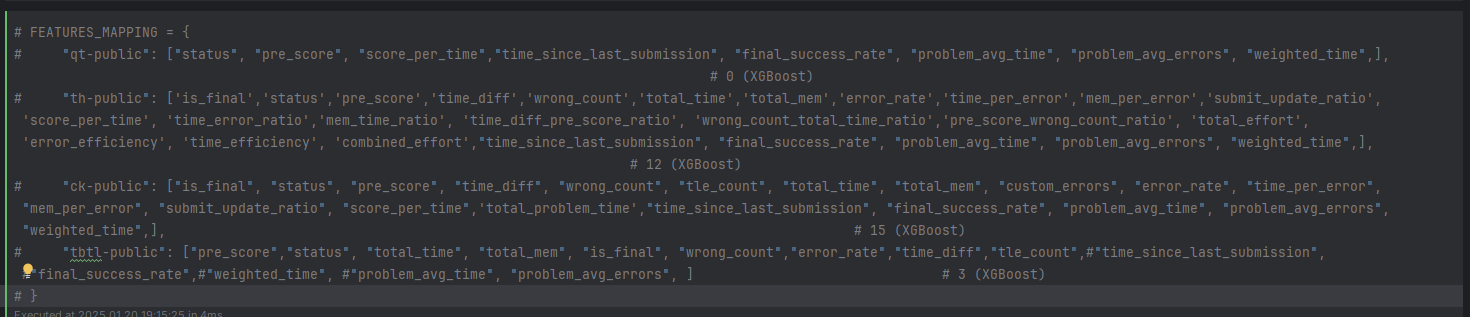
- Ngoài ra, R2-score của TBTL cao hơn gấp đôi so với QT nhưng khi đưa ra submit trên wecode thì kết quả của chúng chỉ chênh nhau 1 điểm

## PHÂN TÍCH KẾT QUẢ

**-** StandardScaler được **nhóm** dùng ở đây chỉ để chuẩn hóa các số trong các feature lại cho việc chạy training và predict không có bị lỗi hay bị chênh lệch quá nhiều khiến cho kết quả bị ảnh hưởng.  
**-** Phần feature mapping thì lý do **nhóm** chọn như vậy tại vì có **thử** qua nhiều feature mapping khác trên model khác thì kết quả sinh ra không khá tốt

A computer screen with text

Description automatically generated  
 Feature Mapping cho mô hình RandomForestRegressor



*Feature Mapping cho mô hình XGBoost*

- Kết quả sau các dấu thăng là điểm số nhóm nhận được khi submit các predict theo các feature mapping lên wecode và ghi nhận thì nhóm có thấy rằng mặc dù cùng 1 feature mapping của QT nhưng thay đổi model thì score của nó cho ra khác nhau hoàn toàn vì thế đó là lý do mà nhóm chọn RandomForest cho điểm QT còn về các điểm còn lại thì XGBoost lại làm cho điểm chúng tăng thêm kha khá.

## TÀI LIỆU THAM KHẢO

* CHATGPT 4o

# NHẬN DIỆN HÃNG XE QUA ẢNH CHỤP

## CẬP NHẬT SAU VẤN ĐÁP

- Có số liệu cụ thể về số chiều trước và sau khi áp dụng PCA lên vector đặc trưng rút trích được từ mô hình MobileNet (Phần **Phân cụm dữ liệu** trong phần **Các bước nhóm đã thực hiện)**

- Giải thích được vì sao vừa sử dụng KMeans vừa sử dụng độ đo Cosine để tìm kiếm ảnh trùng (Phần **Phân tích chuyên sâu)**

**-** Thêm các kĩ thuật để tối ưu mô hình (Phần **Phân tích chuyên sâu)**

**-** Độ chính xác của mô hình đã được cải thiện trên cả hai tập huấn luyện và kiểm tra (Phần **Kết quả nhận được và Nhận xét)**

## GIỚI THIỆU BÀI TOÁN

Nhận diện hãng xe qua ảnh chụp là bài toán thuộc lĩnh vực của Máy học, với mục tiêu xác định hãng xe từ những bức ảnh chụp đầu vào. Kết quả đầu ra là tên hãng xe tương ứng với chiếc xe trong ảnh. Bài toán được chia thành hai thành phần chính là chuẩn bị, tiền xử lý dữ liệu và huấn luyện mô hình máy học để nhận diện hãng xe với dữ liệu được thu thập từ các sinh viên trong lớp

## CÁC BƯỚC NHÓM ĐÃ THỰC HIỆN

### THỐNG KÊ DỮ LIỆU

- Mục tiêu: Xử lý tập dữ liệu ảnh của các hãng xe, trích xuất thông tin từ file và thống kê đóng góp của các sinh viên dựa trên tên file ảnh

- Duyệt qua các thư mục và file ảnh, mỗi thư mục tương ứng với một hãng xe, file ảnh trong mỗi thư mục mang thông tin về danh sách mã số sinh viên đóng góp và hãng xe trong tên ảnh

- Tách thông tin cần thiết từ tên file bao gồm danh sách các mã số sinh viên, tên hãng xe, sau đó tính trọng số đóng góp của từng mã số sinh viên (trọng số sẽ chia đều nếu một tấm ảnh có nhiều mã số sinh viên)

- Tính tổng trọng số đóng góp của mỗi mã số sinh viên cho toàn bộ các ảnh và tính đóng góp cụ thể của từng mã số sinh viên cho từng hãng xe và lưu lần lượt vào hai file CarDataset-1.csv và CarDataset-2.csv theo yêu cầu

A screenshot of a computer program

Description automatically generated

*Dữ liệu từ file CarDataset-1.csv*

*A screenshot of a computer

Description automatically generated*

*Dữ liệu từ file CarDataset-2.csv*

### PHÂN CHIA DỮ LIỆU

- Mục tiêu: Sử dụng K-Fold Cross-Validation để chia dữ liệu thành các tập huấn luyện và kiểm tra, đảm bảo sự đồng đều và đầy đủ dữ liệu trong mỗi tập

- Duyệt qua các thư mục với tên đại diện cho các hãng xe, đọc danh sách file ảnh trong từng thư mục và gán nhãn (categoryID) tương ứng cho từng hãng xe dựa trên ánh xạ *tag\_mapping*, sau đó ghi toàn bộ thông tin về ảnh gồm có đường dẫn và nhãn vào file CarDataset.csv

A screen shot of a computer program

Description automatically generated

*tag\_mapping dùng cho ánh xạ giữa tên hiệu xe và categoryID tương ứng*

A screenshot of a computer

Description automatically generated

*Dữ liệu từ file CarDataset.csv*

- Sử dụng kĩ thuật k-fold cross validation để đảm bảo tính ngẫu nhiên và đồng đều khi chia tập dữ liệu, và giá trị k được sử dụng bằng 5, tức là dữ liệu được chia thành 5 phần, mỗi lần một phần được sử dụng cho kiểm tra và các phần còn lại cho huấn luyện

*A screenshot of a computer

Description automatically generated*

*Các file huấn luyện và kiểm tra được tạo ra*

*A screenshot of a computer

Description automatically generated*

*Dữ liệu được phân chia ngẫu nhiên trong file CarDataset-Splits-1-Test.csv*

- Hiển thị danh sách các tệp ảnh trong từng tâp huấn luyện và kiểm tra, bao gồm cả nhãn và thống kê số lượng ảnh của từng nhãn trong mỗi tập huấn luyện và kiểm tra, đảm bảo tính đồng đều và đầy đủ

A screenshot of a computer program

Description automatically generated

*Các file ảnh và nhãn trong file huấn luyện của Split-1*

*A screenshot of a computer

Description automatically generated*

*Thống kê số lượng ảnh của từng nhãn trong tập huấn luyện và kiểm tra*

### TRỰC QUAN HÓA DỮ LIỆU

- Mục tiêu: Trích xuất đường dẫn ảnh từ tập dữ liệu và gán nhãn tương ứng dựa trên thư mục chứa các file ảnh. Sau đó, danh sách được lưu vào file csv và hiển thị một số hình ảnh minh họa theo mẫu đã cho trước

- Duyệt qua các thư mục, duyệt qua các file trong các thư mục, giữ lại các file có đuôi “.jpg”, gán nhãn tương ứng theo tên thư mục, lưu đường dẫn và nhãn tương ứng vào file csv

- Sử dụng file csv vừa tạo được, tạo mapping ngược từ categoryID (số) sang tên hãng xe để hiển thị, lọc và giữ lại các nhãn có ít nhất một hình ảnh (không hiển thị nếu như nhãn đó không có bất kì hình ảnh nào), với từng nhãn, hiển thị tên nhãn và một số hình ảnh ngẫu nhiên, kích thước ảnh hiển thị tuân theo tham số yêu cầu trước

A screenshot of a computer screen

Description automatically generated

*Hình ảnh các ô tô theo từng nhãn*

### PHÂN CỤM DỮ LIỆU

- Mục tiêu: Trích xuất các đặc trưng hình ảnh từ tập dữ liệu, phân cụm hình ảnh có đặc trưng tương tự nhau bằng KMeans, sau đó với từng cụm, tìm kiếm và lọc các hình ảnh bị trùng lặp trong cụm, xuất kết quả ra các file csv để lưu trữ tập dữ liệu đã phân cụm và danh sách hình ảnh bị trùng lặp

- Trích xuất đặc trưng hình ảnh: Duyệt qua các đường dẫn tới hình ảnh từ file CarDataset.csv, resize về kích thước (224, 224), chuẩn hóa dữ liệu bằng *preprocess\_input* của MobileNet, sau đó sử dụng mô hình MobileNet để trích xuất vector đặc trưng cho từng hình ảnh, kết hợp PCA để giảm số chiều của vector đặc trưng về 50 chiều, giúp tăng tốc độ phân cụm cho bước tiếp theo và giảm bộ nhớ sử dụng vì một hình ảnh có kích thước (224, 224, 3), sau khi dùng MobileNet để trích xuất đặc trưng thì số chiều của vector đặc trưng thường là 1024

A screen shot of a computer program

Description automatically generated

*Sử dụng ảnh giả lập với kích thước như ảnh dùng cho huấn luyện để in ra số chiều ban đầu*

- Sử dụng thuật toán KMeans với k = 5 để phân thành 5 cụm, hình ảnh ở mỗi cụm sẽ có đặc trưng tượng tự nhau và được gán nhãn của cụm (ClusterID) cho từng hình ảnh

- Xác định hình ảnh trùng lặp trong cùng một cụm dựa trên độ đo tương đồng giữa các vector đặc trưng, nhóm sử dụng ma trận độ tương đồng cosine giữa các vector với giá trị ngưỡng là lớn hơn 0,95, nếu độ đo giữa hai hình ảnh vượt mức 0,95 thì được xem là trùng nhau

- Lưu trữ kết quả phân cụm và danh sách hình ảnh trùng lặp:

+ *clustering\_results.csv*: Dùng để lưu trữ những hình ảnh không bị trùng lặp sau khi phân cụm và đánh gía

+ *refined\_potential\_duplicates.csv*: Lưu trữ các cặp hình ảnh được phát hiện trùng lặp cùng với độ đo cosine tương ứng

+ *Hậu Kiểm.csv*: Danh sách hình ảnh bị trùng lặp sau khi lọc

### HUẤN LUYỆN MÔ HÌNH

- Mục tiêu: Huấn luyện mô hình nhận diện hãng xe qua ảnh chụp. Các bước chính thực hiện là tiền xử lý dữ liệu, chuẩn bị tập huấn luyện, tập kiểm tra và xây dựng mô hình

- Từ file “clustering\_results.csv” ở phần phân cụm hình ảnh, nhóm trích xuất tên hãng xe từ đường dẫn của hình ảnh và gán nhãn dưới dạng số (CategoryID) dựa trên mapping của các hãng xe trong tag\_mapping bằng cách sử dụng *extract\_brand\_from\_path*

- Ở phần chuẩn bị dữ liệu cho tập huấn luyện và kiểm tra, đầu tiên sẽ loại bỏ các đường dẫn hình ảnh không tồn tại, sau đó tách thành hai danh sách gồm đường dẫn hình ảnh và nhãn tương ứng, chia dữ liệu với 80% dùng cho huấn luyện và 20% dùng cho kiểm tra với kĩ thuật stratified sampling để đảm bảo tỷ lệ nhãn đồng đều, cuối cùng load hình ảnh và resize về kích thước nhóm quy định là (224, 224)

- Xây dựng và huấn luyện mô hình:

A screenshot of a phone

Description automatically generated

*Số lượng ảnh cho từng loại xe trong file clustering\_result.csv*

+ Do số lượng hình ảnh của từng hãng xe trong tập dữ liệu có sự chênh lệch đáng kể, nhóm đã sử dụng kỹ thuật Class Weighting để giải quyết vấn đề mất cân bằng dữ liệu. Phương pháp này gán trọng số (class weight) cho từng nhãn, nhằm giảm thiểu hiện tượng thiên vị (bias) trong quá trình huấn luyện và giúp mô hình học tốt hơn đối với các nhãn có ít dữ liệu.

A screenshot of a computer program

Description automatically generated

Trọng số của từng nhãn sau khi áp dụng kỹ thuật Class Weighting

+ Sử dụng mô hình MobileNetV2 đã được huấn luyện trước (pre-trained) trên “ImageNet” để làm mô hình gốc (base-model) và xây dựng các lớp “Fully Connected” (FC) để học các đặc trưng tốt hơn từ base-model và tự tùy chỉnh để tăng khả nhận diện các hãng xe trong bài toán cụ thể

+ Sử dụng hàm chi phí categorical\_crossentropy và optimizer Adam với learning rate = 0,00001

+ Sử dụng và triển khai callback với ModelCheckpoint dùng để lưu trữ mô hình cho kết quả tốt nhất trong quá trình huấn luyện và EarlyStopping để ngừng việc huấn luyện nếu độ chính xác trên tập kiểm tra không còn cải thiện trong 7 epoch

+ Mô hình được huấn luyện với tập dữ liệu huấn luyện và tập kiểm tra trong 40 epoch và batch\_size là 32

## KẾT QUẢ ĐẠT ĐƯỢC VÀ NHẬN XÉT

- Trong quá trình huấn luyện với 40 epoch, độ chính xác cao nhất cho huấn luyện và kiểm tra lần lượt là 0,94 và 0,64 (đã có cải thiện so với lần vấn đáp, ở lần vấn đáp, độ chính xác lần lượt là 0,75 và 0,5)



Độ chính xác của mô hình ở lần vấn đáp



Độ chính xác sau khi được cập nhật

- Classification report của nhóm:

A screenshot of a computer

Description automatically generated

Classification report của mô hình ở lần vấn đáp

A screenshot of a computer screen

Description automatically generated

Classification report của mô hình sau cập nhật

- Nhận xét:

+ Sau khi áp dụng kĩ thuật Class Weighting và đóng băng (freeze) 100 lớp đầu tiên của mô hình MobileNet (sẽ nói cụ thể ở phần phân tích kết quả) thì độ chính xác mô hình có sự cải thiện khá tốt, đặc biệt đối với các nhãn ít dữ liệu. Tuy nhiên độ chính xác 64% trên tập kiểm tra là một kết quả không quá tốt và mô hình có xu hướng overfitting, có thể do mô hình vẫn còn gặp khó khăn khi nhận diện một số nhãn có dữ liệu còn chưa được đầy đủ và đặc trưng khó phân biệt

+ Tuy vậy, mô hình cho kết quả nhận diện khá tốt đối với nhóm xe như Suzuki, Vinfast và Mazda. Ngoài ra các độ đo của các hãng xe hầu hết có giá trị lớn hơn 0,5, cho thấy mô hình có hiệu suất tổng thể khá ổn định

## PHÂN TÍCH CHUYÊN SÂU

### Phân cụm dữ liệu

- Như đã trình bày ở phần **Các bước nhóm đã thực hiện**, nhóm sử dụng mô hình MobileNet để trích xuất đặc trưng từ các hình ảnh trong tập dữ liệu ban đầu, tiếp đến sử dụng kĩ thuật PCA để giảm số chiều của vector đặc trưng, sau đó sử dụng thuật toán KMeans với giá trị k = 5 để gom cụm những hình ảnh có vector đặc trưng tương tự nhau, với mỗi cụm thì sử dụng độ đo Cosine với ngưỡng do nhóm quy định để xác định các cặp ảnh có khả năng trùng nhau

- Những đặc trưng hình ảnh trích xuất được từ mô hình MobileNet sẽ bao gồm hình dáng tổng quát như khung xe, đường viền trên xe và hình dạng đặc trưng của các hãng xe, chi tiết nổi bật như logo hãng, các chi tiết đặc thù cho từng hãng, đèn pha,.. và màu sắc cũng như bố cục các thành phần trong ảnh. MobileNet được sử dụng như một công cụ trích xuất đặc trưng, tầng đầu tiên là các đặc trưng thấp và cơ bản như cạnh, góc và màu sắc, những tầng sâu hơn học các đặc trưng phức tạp và tốt hơn như cấu trúc tổng thể xe, logo, các chi tiết đặc biệt

- Việc sử dụng KMeans là để phân cụm các hình ảnh có đặc trưng tương tự nhau dựa trên khoảng cách Euclidean giữa các vector đặc trưng và các tâm cụm, điều này sẽ giúp giảm phạm vi xử lý bằng cách giới hạn việc kiểm tra sự trùng lặp giữa các ảnh trên từng cụm thay vì phải kiểm tra tên toàn bộ tập dữ liệu lớn. Tuy nhiên, nếu chỉ sử dụng mỗi KMeans thì sẽ không hiệu quả tốt vì KMeans chỉ xem xét khoảng cách, không trực tiếp đo lường góc giữa các vector. Hai vector có thể gần nhau về khoảng cách nhưng có thể không cùng hướng dẫn đến hai hình ảnh không tương tự về nghĩa, có thể là hai hình ảnh khác nhau.

- Việc sử dụng độ đo Cosine dùng để đánh giá mức độ tương tự giữa các vector đặc trưng trong một cụm sau khi sử dụng KMeans, độ đo Cosine sẽ tính thêm góc giữa hai vector, nhóm sử dụng ngưỡng là 0,95, nếu lớn hơn thì được xem là trùng nhau (tức là góc giữa hai vector có giá trị tiến dần về 0 đồng nghĩa với việc hai vector có cùng hướng, thể hiện mức độ giống nhau cao)

- Sau đây là một vài hình ảnh sau khi phân cụm và vài hình ảnh về các cặp ảnh mà máy cho là trùng sau khi dùng độ đo Cosine

A screenshot of a car

Description automatically generated

Các hình ảnh sau khi sử dụng KMeans

- Đối với nhóm có clusterID là 0, các hình ảnh ô tô sẽ là hình ảnh chụp từ phía trước xe, còn đối với nhóm clusterID là 3 thì các ô tô phần lớn sẽ hướng về bên trái

- Đối với các cặp hình ảnh mà máy tính cho là trùng, sẽ có những cặp hình ảnh chính xác là trùng nhau (giống nhau như đúc), và có những cặp hình ảnh về cùng một xe nhưng khoảng cách chụp là xa và gần, cũng có những cặp hình ảnh hai xe có kiểu dáng trùng nhau và khác màu,..

A collage of images of a car

Description automatically generated

Các cặp hình ảnh trùng lặp chính xác

A couple of cars on a road

Description automatically generated

Cặp hình ảnh chụp cùng một xe nhưng khoảng cách chụp là khác nhau

A group of cars with text and numbers

Description automatically generated

Cặp hình ảnh các xe giống nhau về kiểu dáng nhưng khác màu sắc

A white car with a red tail light

Description automatically generated

Cặp hình ảnh xe giống nhau nhưng khác góc chụp

- Nhận xét: Để các hình ảnh có thể chính xác hơn về việc bị trùng lắp thì nhóm cần tăng ngưỡng của độ đo Similarity hiện tại, có thể tăng lên giá trị khoảng 0,98; 0,99

### Xây dựng và huấn luyện mô hình

- Những cải tiến của mô hình cuối cùng so với mô hình ở buổi vấn đáp: Sử dụng stratified sampling trong quá trình chuẩn bị dữ liệu huấn luyện và kiểm tra, sử dụng thêm kỹ thuật Class Weighting, đóng băng n lớp đầu tiên của mô hình MobileNet, giảm learning rate đối với optimizer Adam từ 0,0001 thành 0,00001

- Đối với stratified sampling: Giúp đảm bảo rằng các nhãn trong tập huấn luyện và kiểm tra có tỷ lệ đồng đều để mô hình học được đầy đủ tất cả các nhóm nhãn, đặc biệt là đối với các nhãn có số lượng mẫu (sample) ít

A screenshot of a computer

Description automatically generated

Sử dụng stratified sampling

- Đối với kĩ thuật Class Weighting: Tăng trọng số cho các nhãn ít mẫu nhằm giảm thiểu thiên vị (bias) trong quá trình huấn luyện. Ví dụ như đối với nhãn “Vinfast” có số lượng dữ liệu là ít nhất, nếu không sử dụng Class Weighting thì khả năng cao mô hình sẽ bỏ qua nhãn. Kĩ thuật này sẽ cho các nhãn có số mẫu càng ít thì trọng số càng lớn



Các độ đo cho nhãn vinfast của mô hình ban đầu



Các độ đo cho nhãn vinfast của mô hình sau khi cập nhật

A screenshot of a computer program

Description automatically generated

Trọng số cho các lớp khi áp dụng Class Weighting

- Đóng băng (Freeze) n lớp đầu tiên của mô hình: Giữ lại các đặc trưng thấp đã học từ ImageNet của mô hình MobileNet vì các đặc trưng cơ bản như đường viền, kết cấu chung giữa các đối tượng được sử dụng cho các bài toán nhận diện đối tượng trong hình ảnh

- Giảm Learning Rate: Đảm bảo mô hình học chậm hơn, tăng độ ổn định trong quá trình tối ưu hóa, việc giảm learning rate từ 0,0001 xuống 0,00001 là vì khi learning rate lớn, nếu có sự thay đổi nhỏ trong hàm chi phí (loss function) giữa các batch dữ liệu thì mô hình sẽ dễ bị nhạy cảm dẫn đến không hội tụ hoặc dao động xung quanh giá trị tối ưu thay vì đạt tới giá trị đó. Ngoài ra việc giảm learning rate xuống 0,00001 giúp cho mô hình cập nhật trọng số chậm hơn nhưng cẩn thận hơn, nó có thể giúp tăng khả năng hội tụ một cách ổn định, chính xác hơn

- Ở phần thêm các lớp Fully Connected (FC):

+ Dense (256 units) + BatchNormalization + Dropout(0.3) giúp học các đặc trưng phức tạp, giảm overfitting

+ Dense (128 units) + BatchNormalization + Dropout(0.2) tiếp tục tối đa hóa mô hình nhưng vẫn bảo toàn tổng quát hóa, không lược bỏ đi những đặc trưng cơ bản

+ Dense (9 units: số lượng nhãn) với activation là softmax dùng để phân loại

A computer code with text

Description automatically generated with medium confidence

Thêm các lớp Fully Connected

+ Flatten dùng để chuyển đổi dữ liệu từ dạng 2D sang vector, chuẩn bị cho các lớp FC

+ BatchNormalization giúp tăng tốc độ huấn luyện nhưng vẫn đảm bảo mô hình ổn định và giảm thiểu overfitting

+ Dropout giúp giảm thiểu overfitting bằng việc loại bỏ ngẫu nhiên một tỷ lệ kết nối giữa lớp trên và lớp dưới

## TÀI LIỆU THAM KHẢO

[1] CHATGPT 4o

[2] <https://www.geeksforgeeks.org/adam-optimizer/> - Sử dụng cho tối ưu mô hình và thay đổi learning rate

[3] <https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2/> - Tìm hiểu kiến trúc với các lớp cho mô hình MobileNetV2