

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

□□□□&□□□□

BÁO CÁO ĐỒ ÁN MÔN HỌC



MÁY HỌC
ĐỀ TÀI: NHẬN DIỆN THƯƠNG HIỆU XE DỰA TRÊN
LOGO

Giảng viên hướng dẫn:

ThS. Phạm Nguyễn Trường An

Sinh viên thực hiện:

Nguyễn Huỳnh Bảo Quốc 20520292

Đinh Vũ Minh Nhật 21522414

Đào Nam Thuận 21522649

TP. HỒ CHÍ MINH, THÁNG 07 NĂM 2023

MỤC LỤC

I. TÓM TẮT ĐỒ ÁN.....	1
II. NỘI DUNG ĐỒ ÁN.....	3
2.1 Mô tả bài toán và ngữ cảnh ứng dụng:	3
2.2 Mô tả dữ liệu	3
2.3 Quy trình xử lý dữ liệu:.....	4
2.3.1 Xử lý dữ liệu ảnh bị nhiễu, bias khi thu thập dữ liệu:	4
2.3.2 Import các thư viện cần thiết cho xử lý như os, numpy, tensorflow, keras, cv2, pandas, sklearn.....	5
2.3.3 Đọc dữ liệu và lưu trữ dữ liệu:.....	5
2.3.4 Tìm vị trí logo trong ảnh và cắt hình sát vào vị trí của logo:	5
2.3.5 Định dạng lại toàn bộ dữ liệu ảnh về cùng một kích thước.....	7
2.3.6 Tăng cường dữ liệu	7
2.3.7 Chia dữ liệu để huấn luyện mô hình:	8
2.4 Huấn luyện mô hình	8
2.4.1 Mô hình Support Vector Machine (SVM)	8
2.4.2 Mô hình VGG16	9
2.4.3 Mô hình VGG19	9
2.4.4 Mô hình Vision Transformer (ViT).....	9
2.5 Kết quả huấn luyện	10
III. SO SÁNH VỚI DỰ ÁN TƯƠNG TỰ	11
3.1 Đối tượng so sánh.....	11
3.2 Đã hoàn thành	11
3.3 Chưa hoàn thành	11
3.4 Phát triển thêm so với tác giả.....	11
3.5 Kết quả so sánh.....	11
IV.KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	13
4.1 Ưu điểm.....	13
4.2 Nhược điểm	13
4.3 Hướng phát triển	13
V. CẬP NHẬP SAU BÁO CÁO	15
TÀI LIỆU THAM KHẢO	15

I. TÓM TẮT ĐỒ ÁN

Vấn đề nghiên cứu:

Trong lĩnh vực giao thông và an ninh, việc nhận diện xe thông qua hình ảnh logo đóng vai trò quan trọng để đảm bảo an toàn giao thông, kiểm soát xe cộ ý vi phạm, và nâng cao hiệu suất quản lý đô thị. Ngoài ra, có rất nhiều loại xe trên thế giới chưa được nhiều người biết đến. Tuy nhiên, việc nhận diện logo trên xe gặp phải nhiều thách thức do sự biến đổi về kích thước, góc nhìn, ánh sáng và môi trường xung quanh. Vì vậy, vấn đề cần được nghiên cứu là ứng dụng Machine Learning và Deep Learning để nhận diện xe thông qua hình ảnh logo một cách chính xác và hiệu quả.

Hướng tiếp cận:

Để giải quyết vấn đề nghiên cứu, chúng ta có thể sử dụng các phương pháp và kỹ thuật của Máy học và Học sâu.

Đầu tiên, có thể xây dựng một mô hình học máy cơ bản sử dụng các thuật toán phân loại như Support Vector Machines (SVM) hoặc Random Forest để nhận diện logo trên xe.

Tiếp theo, có thể sử dụng kiến trúc của VGG16, VGG19 hoặc ViT để xây dựng mô hình nhận diện logo xe hơi. Các mô hình này được đào tạo trước trên ImageNet, một tập dữ liệu rất lớn và đa dạng các loại ảnh. Đồng thời, tận dụng khối convolutions và fully connected layers của mô hình đã được huấn luyện trước và thay thế lớp cuối cùng bằng một lớp mới phù hợp với bài toán nhận diện thương hiệu xe dựa trên logo.

Cuối cùng, để nâng cao khả năng nhận diện, ta có thể áp dụng các mạng neural sâu như Convolutional Neural Networks (CNN) để xây dựng mô hình học sâu cho việc nhận diện xe thông qua hình ảnh logo. Qua việc huấn luyện mô hình trên tập dữ liệu lớn, mô hình sẽ tự động học và trích xuất các đặc trưng quan trọng từ hình ảnh logo để nhận diện xe một cách chính xác.

Đối tượng sử dụng:

- Các cơ quan phân loại xe.
- Các công ty nghiên cứu thị trường.

- Những doanh nghiệp kinh doanh xe hơi đã qua sử dụng.
- Những công ty cung cấp giải pháp quảng cáo, marketing.

Kết quả đạt được:

Kết quả của nghiên cứu này là một mô hình nhận diện xe thông qua hình ảnh logo đạt được độ chính xác và độ hiệu quả cao. Mô hình này có thể được triển khai trong các ứng dụng thực tế như hệ thống giám sát giao thông, kiểm soát xe cố ý vi phạm, trong nghiên cứu thị trường, các công ty kinh doanh oto hoặc xây dựng cơ sở dữ liệu liên quan đến xe và logo. Kết quả đạt được có thể cung cấp giá trị quan trọng cho các nghiệp vụ liên quan đến xe và có thể ứng dụng cho các loại hình giao thông thông minh trong tương lai.

II. NỘI DUNG ĐỒ ÁN

2.1 Mô tả bài toán và ngữ cảnh ứng dụng:

Bài toán cần giải quyết ở đề tài lần này là vận dụng các thuật toán học máy, học sâu để nhận diện thương hiệu oto dựa trên ảnh chụp logo của xe đó. Qua đó có thể giúp giải quyết được các vấn đề liên quan đến xe trong cuộc sống.

Một số ví dụ về ngữ cảnh sử dụng:

- Một công ty xe ABC muốn chỉ có xe của công ty mới được phép ra vào xưởng, hệ thống nhận diện sẽ quét logo xe và đồng ý nếu xe thuộc hãng ABC.
- Ở một buổi tiệc dành cho giới thượng lưu, hệ thống sẽ nhận diện các xe đi vào bãi gửi xe và thống kê xem giới thượng lưu thường ưa chuộng những xe gì, từ đó đưa ra các bài báo cáo, phân tích, đánh giá thị trường oto.
- Đối với các doanh nghiệp kinh doanh xe đã qua sử dụng, hệ thống sẽ giúp nhận biết, phân loại các hãng xe hơi để đưa lên các dây chuyền phù hợp (xe đắt tiền, xe tầm trung, xe phổ thông,...).
- Những công ty cung cấp giải pháp quảng cáo, marketing có thể cung cấp hệ thống này cho công ty xe ABC, sau đó tung ra thị trường để khách hàng có thể sử dụng hệ thống này, nhằm nhận biết xe đó có phải xe của công ty ABC hay không.

Input: Ảnh chụp logo của hãng xe oto.

Output: Tên hãng xe đó

2.2 Mô tả dữ liệu

2.2.1 Quá trình thu thập dữ liệu

Bộ dữ liệu được thu thập bằng thuật toán của các thư viện hỗ trợ bao gồm BeautifulSoup và Selenium trong đó:

- BeautifulSoup là một thư viện Python được sử dụng để phân tích và trích xuất dữ liệu từ các trang web. Nó cung cấp các công cụ để xử lý HTML và XML, cho phép dễ dàng tìm kiếm, điều hướng và trích xuất thông tin từ các phần tử trong tài liệu web. Thư viện BeautifulSoup giúp lấy URL của các ảnh từ Google image

- Selenium là một thư viện Python được sử dụng để tự động hóa các tác vụ trên trình duyệt web. Nó cung cấp một API cho phép tương tác với trình duyệt web và thực hiện các hoạt động như điều hướng trang, nhập liệu, click vào các phần tử, và trích xuất dữ liệu từ trang web. Thư viện Selenium giúp thực hiện thao tác cuộn chuột và click vào ảnh để tiến hành download.

2.2.2 Bảng dữ liệu

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	Car_logo	Image	Hình ảnh logo của các hãng xe oto
2	Car_name	String	Tên hãng xe

2.3 Quy trình xử lý dữ liệu:

2.3.1 Xử lý dữ liệu ảnh bị nhiễu, bias khi thu thập dữ liệu:

Dữ liệu khi crawl bao gồm tất cả các hình ảnh có logo của hãng xe bao gồm ảnh chụp xe, ảnh chụp logo xe, ảnh chụp áo của hãng xe,...

Dữ liệu khi thu thập được sẽ bị nhiễu vì bao gồm logo của các trang web chứa hình ảnh đó, ảnh của các biểu tượng quảng cáo trên trang web,...

Để xử lý ảnh nhiễu, sử dụng 2 thư viện bao gồm:

- Thư viện Shutil là một thư viện trong Python cung cấp các chức năng liên quan đến thao tác trên tệp và thư mục. Nó cung cấp các công cụ để sao chép, di chuyển, đổi tên và xóa các tệp và thư mục. Shutil cho phép thực hiện các tác vụ quản lý tệp và thư mục trên hệ thống tệp.
- Thư viện cv2 (OpenCV) là một thư viện mã nguồn mở cho xử lý ảnh và thị giác máy tính. Nó cung cấp các chức năng và công cụ để thực hiện các nhiệm vụ như đọc và ghi ảnh, xử lý ảnh, phân tích đối tượng, nhận dạng khuôn mặt, và nhiều nhiệm vụ khác liên quan đến xử lý ảnh và video.
- Kết hợp 2 thư viện trên để đọc kích thước ảnh và thực hiện xóa các ảnh nhiễu có kích thước 16*16 (vì toàn bộ hình ảnh logo của trang web đều có kích thước này).

Đối với các hãng xe có số lượng ảnh ít hơn 50 và có nhiều ảnh lỗi, xóa các hãng xe ấy đi. Và sử dụng smote over sampling để cân bằng số ảnh của mỗi xe ít nhất là 100 để tránh bị bias.

Cuối cùng ta thu được một bộ dataset bao gồm 70 hãng xe có số lượng ảnh mỗi xe ít nhất là 100 ảnh.

2.3.2 Import các thư viện cần thiết cho xử lý như os, numpy, tensorflow, keras, cv2, pandas, sklearn.

Công dụng của các thư viện trên trong đề tài:

- Os: đọc dữ liệu từ data path
- Numpy: sử dụng numpy stack
- Tensorflow: triển khai mô hình VGG16 và VGG 19
- Keras: hỗ trợ huấn luyện mô hình VGG16, VGG19, ViT
- Cv2: đọc và ghi dữ liệu ảnh
- Pandas: lấy dữ liệu ảnh đưa vào data frame
- Sklearn: hỗ trợ huấn luyện mô hình SVM, cung cấp các chỉ số đo hiệu suất mô hình

2.3.3 Đọc dữ liệu và lưu trữ dữ liệu:

Đọc dữ liệu từ đường link dataset của nhóm:

(<https://github.com/James2109/CS114.N21.KHCL/tree/main/dataset>), bộ dữ liệu bao gồm 70 hãng xe hơi với hơn 11.000 ảnh được chia sẵn thành các folder khác nhau cho từng hãng.

Trong khi đọc dữ liệu ảnh sẽ gán nhãn cho từng điểm dữ liệu thông qua thư mục cha trực tiếp của ảnh đó.

2.3.4 Tìm vị trí logo trong ảnh và cắt hình sát vào vị trí của logo:

Hiện nay có rất nhiều phương pháp để tìm ra vị trí của logo, tuy nhiên việc phát hiện khu vực chứa logo trong ảnh gặp nhiều khó khăn do có sự khác biệt về màu sắc, hình dạng của các logo. Đối với bài toán nhận diện logo đang chạy trên đường với góc nhìn từ camera chính diện vào phía trước của xe, người ta trích xuất khu vực chứa logo bằng cách dựa vào biển số xe, đèn pha, lưới tản nhiệt,... Sau đó, sử dụng các phương pháp như phát hiện cạnh, phân đoạn màu để định vị chính xác vị trí logo hoặc sử dụng mạng neural để đề xuất khu vực chứa logo. Việc sử dụng mạng neural để đề xuất khu vực chứa logo là phương pháp hiệu quả và được sử dụng nhiều hiện nay, tuy nhiên việc huấn luyện mô hình mạng neural dự đoán gặp nhiều khó khăn, và tốn nhiều chi phí. Do vấn đề của chúng tôi là phát hiện ảnh logo ở bất kì

đâu, không nhất thiết phải là logo trực tiếp trên xe nên việc sử dụng phương pháp phát hiện cạnh Canny là hợp lý đối với bài toán của chúng tôi. Ý tưởng sẽ là tạo ra một bản đồ cạnh từ ảnh đầu vào, sau đó dựa vào bản đồ cạnh và cắt hình xung quanh khu vực có diện tích lớn nhất và nhiều cạnh hơn so với khu vực xung quanh (bởi vì logo của các hãng xe hiện nay thường sẽ nhiều cạnh hơn khu vực xung quanh của chúng). Tuy nhiên, sử dụng phương pháp này sẽ vẫn có sai sót trong việc cắt hình tùy thuộc vào hình ảnh cụ thể. Trong hình ảnh, thường tồn tại các thành phần như: vùng trơn, góc/cạnh và nhiễu. Cạnh trong ảnh mang đặc trưng quan trọng, thường là thuộc đối tượng trong ảnh (object). Do đó, để phát hiện cạnh trong ảnh, giải thuật Canny là một trong những giải thuật phổ biến/nổi tiếng nhất trong Xử lý ảnh.

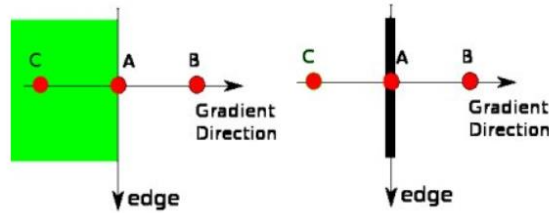
- Nói qua về phương pháp phát hiện cạnh Canny:

- Phát hiện cạnh Canny là một thuật toán phát hiện cạnh phổ biến, được phát triển bởi John F. Canny trong xử lý: giảm nhiễu hình ảnh (Noise Reduction) và tìm cường độ dốc của hình ảnh (Finding Intensity Gradient of the Image).
- Giải thuật phát hiện cạnh Canny gồm 4 bước chính:
 - **Giảm nhiễu:** Làm mờ ảnh, giảm nhiễu dùng bộ lọc Gaussian kích thước 5x5.
 - **Tính Gradient và hướng gradient:** ta dùng bộ lọc Sobel X và Sobel Y (3x3) để tính được ảnh đạo hàm G_x và G_y . Sau đó, ta tiếp tục tính ảnh Gradient và góc của Gradient theo công thức. Ảnh đạo hàm G_x và G_y là ma trận (ví dụ: 640x640), thì kết quả tính ảnh đạo hàm Edge Gradient cũng là một ma trận (640x640), mỗi pixel trên ma trận này thể hiện độ lớn của biến đổi mức sáng ở vị trí tương ứng trên ảnh gốc. Tương tự, ma trận Angle cũng có cùng kích thước (640x640), mỗi pixel trên Angle thể hiện góc, hay nói cách khác là hướng của cạnh.

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2}$$

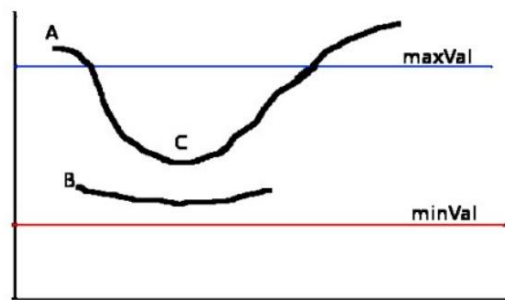
$$Angle(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

- **Non-maximum Suppression (viết tắt NMS):** loại bỏ các pixel ở vị trí không phải cực đại toàn cục. Ở bước này, ta dùng một filter 3x3 lần lượt chạy qua các pixel trên ảnh gradient. Trong quá trình lọc, ta xem xét xem độ lớn gradient của pixel trung tâm có phải là cực đại (lớn nhất trong cục bộ - local maximum) so với các gradient ở các pixel xung quanh. Nếu là cực đại, ta sẽ ghi nhận sẽ giữ pixel đó lại. Còn nếu pixel tại đó không phải là cực đại lân cận, ta sẽ set độ lớn gradient của nó về zero.



Hình 2.1 Non-maximum Suppression

- **Lọc ngưỡng:** ta sẽ xét các pixel dương trên mặt nạ nhị phân kết quả của bước trước. Nếu giá trị gradient vượt ngưỡng max_val thì pixel đó chắc chắn là cạnh. Các pixel có độ lớn gradient nhỏ hơn ngưỡng min_val sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liên kề với những pixel được cho là "chắc chắn là cạnh" hay không.



Hình 2.2 Lọc ngưỡng gradient

2.3.5 Định dạng lại toàn bộ dữ liệu ảnh về cùng một kích thước

Đưa tất cả ảnh về cùng một kích thước (224x224). Các mô hình CNN có một số lượng lớp cố định và yêu cầu đầu vào có kích thước cố định. Việc chuyển đổi ảnh về cùng một kích thước giúp đảm bảo rằng đầu vào của mô hình phù hợp với kiến trúc và kích thước dự kiến của các lớp trong mạng.

2.3.6 Tăng cường dữ liệu

Áp dụng các thay đổi ngẫu nhiên để làm biến đổi trong bộ dữ liệu như xoay, thu/phóng, lật ảnh để làm tăng tính đa dạng của dữ liệu huấn luyện.

- Xoay hình ảnh (Rotate images): Xoay các hình ảnh ngẫu nhiên theo một góc ngẫu nhiên. Bằng cách này, hình ảnh tăng cường có thể gần giống với ảnh chụp được thực hiện dưới một góc nhỏ hơn.
- Lật hình ảnh (Flip images): lật theo chiều dọc, ngang miễn sao ý nghĩa của ảnh (label) được giữ nguyên hoặc suy ra được.

- Thu/phóng hình ảnh (Zoom images): giúp mô hình học sâu không chỉ học được từ những mẫu dữ liệu gốc mà còn từ các biến thể tỉ lệ của chúng, cải thiện khả năng tổng quát hóa của mô hình.

2.3.7 Chia dữ liệu để huấn luyện mô hình:

Tập dữ liệu được chia thành 2 tập bao gồm tập train và tập test với tỉ lệ 8:2 và `random_state = 42` để cố định sự ngẫu nhiên trong quá trình chia dữ liệu.

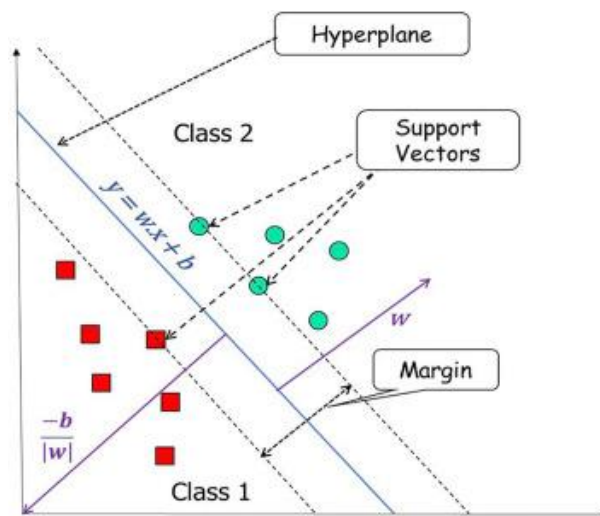
2.4 Huấn luyện mô hình

Dựa trên tập dữ liệu đã xử lý, tiến hành huấn luyện các mô hình khác nhau để xem hiệu quả của các mô hình, từ đó chọn mô hình có độ chính xác tốt nhất.

2.4.1 Mô hình Support Vector Machine (SVM)

Mô hình SVM (Support Vector Machine) là một thuật toán học máy được sử dụng trong bài toán phân loại và hồi quy. Nó được phát triển dựa trên lý thuyết học tập véc-tơ hỗ trợ trong lĩnh vực lý thuyết tối ưu.

Ý tưởng cơ bản của SVM là tìm một siêu phẳng trong không gian đa chiều để phân chia các điểm dữ liệu thuộc vào hai lớp khác nhau. Siêu phẳng này được chọn sao cho cách xa nhất với các điểm dữ liệu gần nhất của các lớp, gọi là các vector hỗ trợ. Điều này giúp tăng tính tổng quát hóa và độ chính xác của mô hình.

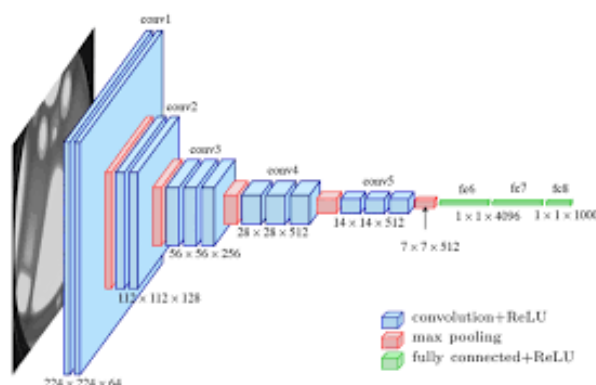


Hình 2.3 Support Vector Machine

2.4.2 Mô hình VGG16

Mô hình VGG16 là một kiến trúc mạng neural convolutional (CNN) phổ biến trong lĩnh vực học sâu (deep learning). Nó được đề xuất bởi nhóm nghiên cứu của VGG (Visual Geometry Group) thuộc Đại học Oxford.

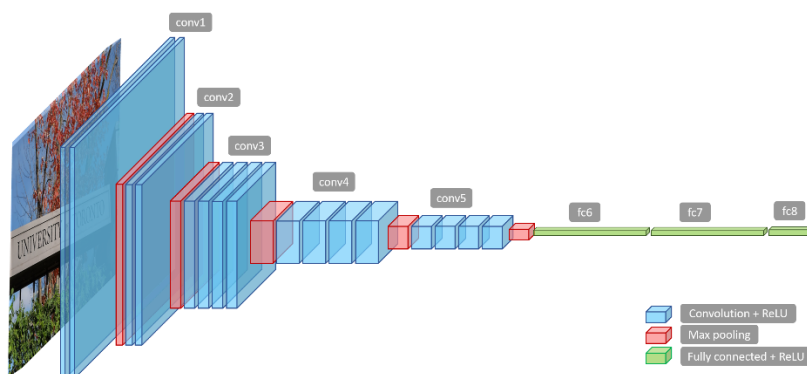
Được xây dựng dựa trên nguyên tắc chính là sử dụng các lớp convolutional có kích thước nhỏ 3x3 liên tiếp nhau nhiều lần. Số "16" trong tên gọi chỉ số lượng các lớp trong mạng, bao gồm 13 lớp convolutional và 3 lớp fully connected.



Hình 2.4 Mô hình VGG16

2.4.3 Mô hình VGG19

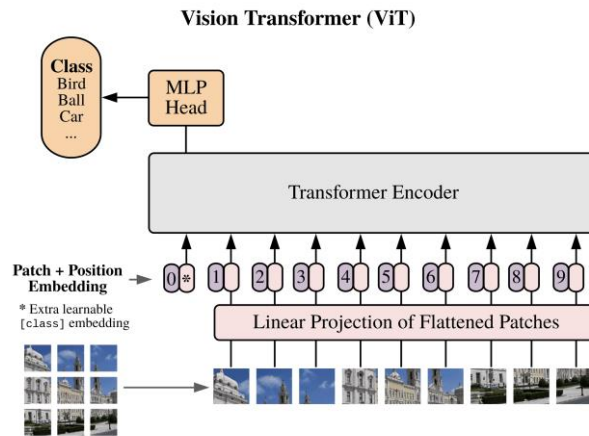
Có cấu trúc tương tự và là phiên bản nâng cấp của VGG16. Bao gồm 16 lớp convolutional và 3 lớp fully connected. So với VGG16, VGG19 có thêm bốn lớp convolutional để tăng độ sâu của mạng.



Hình 2.5 Mô hình VGG16

2.4.4 Mô hình Vision Transformer (ViT)

Mô hình ViT (Vision Transformer) là một kiến trúc mạng neural sử dụng Transformer trong lĩnh vực thị giác máy tính. Nó đã đạt được sự chú ý lớn khi đưa ra phương pháp mới cho việc xử lý ảnh bằng cách áp dụng cơ chế tự chú ý trong mạng Transformer.



Hình 2.6 Mô hình ViT

2.5 Kết quả huấn luyện

Kết quả trên dataset của nhóm xây dựng (70 classes, 11027 images):

Model	Accuracy train	Accuracy test	Parameters
SVM		0.3848	
VGG16	0.6302	0.7353	46,276,230
VGG19	0.5520	0.6505	46,389,638
ViT	0.7107	0.7026	87,260,230

Kết quả trên dataset có sẵn từ kaggle (8 classes, ~5k images):

Model	Accuracy train	Accuracy test	Parameters
VGG19 (của người ta)		X	
VGG16	0.7923	0.7225	46,276,230
VGG19	0.737	0.6725	46,389,638
ViT	0.8556	0.76	87,260,230

**Kết quả của mô hình VGG16, VGG19, ViT (3 dòng cuối) sử dụng model đã train trên dataset của nhóm đánh giá trực tiếp trên dataset 8 lớp này (không train lại).

III. SO SÁNH VỚI DỰ ÁN TƯƠNG TỰ

3.1 Đối tượng so sánh

Tiến hành so sánh kết quả huấn luyện từ 3 mô hình VGG19, VGG16 và ViT của nhóm với mô hình VGG19 của tác giả Volkan Özdemir trên link

<https://www.kaggle.com/datasets/volkandl/car-brand-logos>

3.2 Đã hoàn thành

- Thu thập và xử lý dữ liệu bị nhiễu, kết quả cho ra được dữ liệu của 70 hãng xe hơi phổ biến trên thế giới (11027 ảnh).
- Tiền xử lý dữ liệu với các thao tác: Flip(lật hình), Rotate(xoay hình), Resize.
- Huấn luyện thành công mô hình VGG19 giống như tác giả đã làm.

3.3 Chưa hoàn thành

- Hiện tại những gì tác giả làm được đều đã được nhóm ứng dụng trong việc xây dựng lại mô hình khác với hiệu quả tốt hơn.

3.4 Phát triển thêm so với tác giả

- Huấn luyện thêm 3 mô hình khác là SVM, VGG16, ViT để so sánh với mô hình VGG19.
- Có bước crop ảnh để dò tìm vị trí chính xác của các logo có trong bức ảnh.
- Thử các chỉ số training model khác nhau để tăng độ chính xác.

3.5 Kết quả so sánh

Tất cả các mô hình đều được huấn luyện từ bộ dataset có sẵn của tác giả, 8 classes, ~5k images.

Kết quả huấn luyện model của tác giả:

```
Epoch 9/12
315/315 [=====] - 374s 1s/step - loss: 1.1236 - accuracy: 0.6077 - val_loss: 0.8470 - val_accu
acy: 0.7325
Epoch 10/12
315/315 [=====] - 374s 1s/step - loss: 1.0672 - accuracy: 0.6364 - val_loss: 0.8500 - val_accu
acy: 0.7100
Epoch 11/12
315/315 [=====] - 374s 1s/step - loss: 1.0269 - accuracy: 0.6530 - val_loss: 0.8904 - val_accu
acy: 0.7375
Epoch 12/12
315/315 [=====] - 374s 1s/step - loss: 0.9666 - accuracy: 0.6685 - val_loss: 0.8379 - val_accu
acy: 0.7425
```

Kết quả huấn luyện model của nhóm:

Model	Accuracy train	Accuracy test	Parameters
VGG16	0.7923	0.7225	46,276,230
VGG19	0.737	0.6725	46,389,638
ViT	0.8556	0.76	87,260,230

Kết luận: Cả 3 mô hình của nhóm đều có độ chính xác tốt hơn trên cả tập train và tập test so với mô hình VGG19 của tác giả.

IV.KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Ưu điểm

- Xử lý dữ liệu phức tạp: machine learning và deep learning có khả năng xử lý các dữ liệu ảnh phức tạp, bao gồm các biến thể của logo như kích thước, góc nhìn, màu sắc và độ sáng khác nhau. Điều này giúp hệ thống có khả năng nhận diện logo xe dù trong điều kiện ánh sáng thay đổi hay góc nhìn khác nhau.
- Có thể dễ dàng mở rộng và nâng cấp nếu xây dựng thành công: hệ thống nhận diện dựa trên machine learning và deep learning có khả năng nhận diện thương hiệu xe dựa trên logo, điều này có ích khi có thêm các thương hiệu mới trên thị trường. không cần chỉnh sửa mã nguồn hoặc cấu trúc hệ thống, chỉ cần cung cấp dữ liệu mới và huấn luyện lại.

4.2 Nhược điểm

- Độ chính xác thấp: tuy đã ứng dụng CNN để cải thiện độ chính xác trong quá trình dự đoán, tuy nhiên những mô hình như thế này cần độ chính xác rất cao để đảm bảo nhận diện phù hợp
- Dữ liệu huấn luyện hạn chế: Để huấn luyện một hệ thống nhận diện logo xe đáng tin cậy, cần có một tập dữ liệu lớn, đa dạng và đại diện cho các thương hiệu xe khác nhau. Tuy nhiên, việc thu thập và gán nhãn dữ liệu có thể là một công việc tốn kém và tốn thời gian.
- Khả năng nhận diện bị hạn chế: Đôi khi các logo xe có các biến thể, biểu đồ phức tạp, hoặc số lượng logo là quá lớn trong một bức ảnh khiến cho việc nhận diện trở nên khó khăn. Điều này có thể gây ra nhầm lẫn hoặc sai sót trong quá trình nhận diện logo.
- Hiện tại, tính ứng dụng vẫn còn chưa cao: các mô hình giao thông thông minh chưa được phát triển nhiều ở Việt Nam, do đó phần mềm nhận diện này cần phải đặt trong các ngữ cảnh phù hợp để sử dụng.

4.3 Hướng phát triển

- Sử dụng Recurrent Neural Networks (RNN) để cải thiện độ chính xác và hiệu suất của hệ thống nhận diện logo xe.

- Phát triển hệ thống có khả năng nhận diện logo xe trực tiếp từ video hoặc dữ liệu thời gian thực, điều này hữu ích trong các ứng dụng yêu cầu độ trễ thấp và tương tác nhanh.
- Sử dụng hệ thống nhận diện logo xe để phân tích dữ liệu thương hiệu, tìm hiểu thói quen người dùng và phản hồi thị trường, hoặc áp dụng trong lĩnh vực marketing và quảng cáo.
- Gom dữ liệu từ nhiều nguồn khác nhau và xử lý nhiều để cải thiện khả năng nhận diện logo xe trong các điều kiện khác nhau.
- Tạo cộng đồng về oto để cung cấp thông tin, hình ảnh về các hãng xe, nếu có lượng dữ liệu đủ lớn có thể đưa ra output là toàn bộ những thông tin cần thiết liên quan đến xe đó.
- Phát triển các phương pháp để đánh giá độ tin cậy của hệ thống nhận diện logo xe, như đo lường độ chính xác, độ nhạy.

V. CẬP NHẬP SAU BÁO CÁO

STT	Nội dung	Liên kết
1	Chỉnh sửa lại đối tượng sử dụng phần mềm nhận diện này	Trang 1
2	Thêm ví dụ ngữ cảnh ứng dụng của đề tài	Trang 3
3	Chỉnh sửa lại quy trình xử lý dữ liệu (trước đó chỉ ghi các bước, bây giờ nêu cụ thể bước đó làm những công việc gì)	Trang 4
4	Chỉnh sửa lại phần huấn luyện mô hình (chia ra từng mô hình phù hợp)	Trang 8
5	Huấn luyện thêm mô hình ViT	Trang 9
6	Thêm kết quả huấn luyện cho các mô hình SVM, VGG19, ViT	Trang 10
7	Thêm kết quả so sánh với mô hình VGG19 của tác giả	Trang 11
8	Chỉnh sửa lại phần kết luận và hướng phát triển để phù hợp với kết quả huấn luyện cuối cùng	Trang 13

TÀI LIỆU THAM KHẢO

[1] Car Logos Detection Computer Vision Project.

Link: <https://universe.roboflow.com/new-whnec/car-logos-detection>

[2] Unique Andys.Vehicle Logos Recognition.

Link: <https://github.com/UniqueAndys/vehicle-logo-recognition>

[3] VOLKAN ÖZDEMİR.Car Brand Logos.

Link: <https://www.kaggle.com/datasets/volkandl/car-brand-logos>

[4] Canny Edges Detection – OpenCV

Link: https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

[5] Hugging Face. Vision Transformer (ViT).

Link: https://huggingface.co/docs/transformers/v4.27.0/model_doc/vit