**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞

A blue and white logo with a book and a cap

AI-generated content may be incorrect.

**THỊ GIÁC MÁY TÍNH VÀ NHẬN DẠNG MẪU**

**BÁO CÁO NHÓM**

**PHÂN LOẠI PHƯƠNG TIỆN GIAO THÔNG ĐƯỜNG BỘ**

**TẠI VIỆT NAM**

|  |  |
| --- | --- |
| **Học viên thực hiện:**  1. Nguyễn Lê Quỳnh Như - 1001250016  2. Phan Trần Minh Tâm - 1001250019  3. Nguyễn Minh Trí - 1001250032  4. Cao Hoàng Khánh Băng - 1001250006 | **GVHD:** Trần Khải Thiện |

**TP. HỒ CHÍ MINH – 12/2025**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞

A blue and white logo with a book and a cap

AI-generated content may be incorrect.

**THỊ GIÁC MÁY TÍNH VÀ NHẬN DẠNG MẪU**

**BÁO CÁO NHÓM**

**PHÂN LOẠI PHƯƠNG TIỆN GIAO THÔNG ĐƯỜNG BỘ**

**TẠI VIỆT NAM**

|  |  |
| --- | --- |
| **Học viên thực hiện:**  1. Nguyễn Lê Quỳnh Như - 1001250016  2. Phan Trần Minh Tâm - 1001250019  3. Nguyễn Minh Trí - 1001250032  4. Cao Hoàng Khánh Băng - 1001250006 | **GVHD:** Trần Khải Thiện |

**TP. HỒ CHÍ MINH – 12/2025**

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH iv](#_Toc218871596)

[DANH MỤC BẢNG BIỂU v](#_Toc218871597)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc218871598)

[1.1. Lý do chọn đề tài 1](#_Toc218871599)

[1.2. Mục tiêu của đề tài 1](#_Toc218871600)

[1.3. Đối tượng và phạm vi nghiên cứu 2](#_Toc218871601)

[CHƯƠNG 2. MÔ TẢ DỮ LIỆU VÀ PHƯƠNG PHÁP TIỀN XỬ LÝ 3](#_Toc218871602)

[2.1. Mô tả bộ dữ liệu thu thập 3](#_Toc218871603)

[2.1.1. Tên bộ dữ liệu 3](#_Toc218871604)

[2.1.2. Mục đích và bối cảnh 3](#_Toc218871605)

[2.1.3. Nguồn thu thập dữ liệu 3](#_Toc218871606)

[2.1.4. Bảng thu thập dữ liệu 4](#_Toc218871607)

[2.1.5. Bảng thông số kỹ thuật 4](#_Toc218871608)

[2.1.6. Thống kê bộ dữ liệu 5](#_Toc218871609)

[2.2. Phương pháp tiền xử lý dữ liệu 6](#_Toc218871610)

[2.2.1. Chuẩn hóa kích thước ảnh 6](#_Toc218871611)

[2.2.2. Thuật toán INTER\_AREA là gì? 6](#_Toc218871612)

[2.2.3. So sánh với các thuật toán khác trong OpenCV 7](#_Toc218871613)

[2.2.4. Tại sao dùng INTER\_AREA 7](#_Toc218871614)

[2.2.5. Làm nét ảnh bằng Unsharp Masking 8](#_Toc218871615)

[2.2.6. Nguyên lý hoạt động của Unsharp Masking (USM) 8](#_Toc218871616)

[2.3. Chia tập dữ liệu 9](#_Toc218871617)

[2.3.1. Các quy tắc "vàng" khi chia dữ liệu 9](#_Toc218871618)

[2.3.2. Tại sao phải chia tập dữ liệu 9](#_Toc218871619)

[CHƯƠNG 3. PHƯƠNG PHÁP PHÂN LOẠI 12](#_Toc218871620)

[3.1. Mô hình ResNet 12](#_Toc218871621)

[3.1.1. Giới thiệu 12](#_Toc218871622)

[3.1.2. Cấu trúc và hoạt động ResNet 12](#_Toc218871623)

[3.1.3. Khối residual cơ bản 13](#_Toc218871624)

[3.1.4. Nguyên lý hoạt động ResNet 13](#_Toc218871625)

[3.2. Mô hình EfficientNet 14](#_Toc218871626)

[3.2.1. Giới thiệu 14](#_Toc218871627)

[3.2.2. Kiến trúc 15](#_Toc218871628)

[3.2.3. Compound Scaling 16](#_Toc218871629)

[3.2.4. Mobile Inverted Bottleneck Convolution (MBConv) 16](#_Toc218871630)

[3.2.5. Squeeze-and-Excitation (SE) Blocks 16](#_Toc218871631)

[3.2.6. Ứng dụng 16](#_Toc218871632)

[3.3. Ưu nhược điểm của ResNet và EfficientNet 17](#_Toc218871633)

[3.3.1. Ưu điểm ResNet 17](#_Toc218871634)

[3.3.2. Nhược điểm của ResNet 18](#_Toc218871635)

[3.3.3. Ưu điểm EfficientNet 18](#_Toc218871636)

[3.3.4. Nhược điểm EfficientNet 18](#_Toc218871637)

[CHƯƠNG 4. THỰC NGHIỆM 19](#_Toc218871638)

[4.1. Mô tả tập dữ liệu thực nghiệm 19](#_Toc218871639)

[4.2. Quy trình thực hiện 21](#_Toc218871640)

[4.3. So sánh kết quả giữa mô hình ResNet và EfficientNet 26](#_Toc218871641)

[CHƯƠNG 5. KẾT LUẬN 29](#_Toc218871642)

[5.1. Kết luận 29](#_Toc218871643)

[5.2. Hạn chế và hướng phát triển 29](#_Toc218871644)

[TÀI LIỆU THAM KHẢO 31](#_Toc218871645)

DANH MỤC HÌNH ẢNH

[Hình 2. 1. Ảnh sau khi được làm nét. 8](#_Toc218871647)

[Hình 4. 1. Kết quả hình ảnh sau khi thực hiện tiền xử lý 21](#_Toc218871648)

[Hình 4. 2. Đoạn code thực hiện load dataset 22](#_Toc218871649)

[Hình 4. 3. Đoạn code thực hiện resize ảnh 22](#_Toc218871650)

[Hình 4. 4. Đoạn code thực hiện làm nét ảnh 23](#_Toc218871651)

[Hình 4. 5. Đoạn code thực hiện chuẩn hóa dữ liệu và augmentation 23](#_Toc218871652)

[Hình 4. 6. Đoạn code thực hiện khởi tạo model 24](#_Toc218871653)

[Hình 4. 7. Đoạn code thực hiện huấn luyện mô hình 24](#_Toc218871654)

[Hình 4. 8. Đoạn code thực hiện đánh giá mô hình 25](#_Toc218871655)

[Hình 4. 9. Kết quả nhận được sau thực thi đánh giá mô hình 25](#_Toc218871656)

[Hình 4. 10. Biểu đồ các thông số sau training, accuracy giữa 2 mô hình 26](#_Toc218871657)

[Hình 4. 11. Kết sau thi thực hiện kiểm tra trên bộ dữ liệu test 27](#_Toc218871658)

[Hình 4. 12. Kết quả sau khi thực hiện upload bức ảnh ngẫu nhiên không nằm trong bộ test hay val 28](#_Toc218871659)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1. Bảng mô tả thu thập dữ liệu 4](#_Toc218661651)

[Bảng 2. 2. Bảng mô tả thông số kỹ thuật dữ liệu 4](#_Toc218661652)

[Bảng 2. 3. Bảng mô tả các lớp của bộ dữ liệu 5](#_Toc218661653)

[Bảng 2. 4. Bảng so sánh các thuật toán chuẩn hóa kích thước ảnh 7](#_Toc218661654)

[Bảng 2. 5. Bảng mô tả lý do cần phải phân chia tập dữ liệu 9](#_Toc218661655)

* 1. GIỚI THIỆU
     1. Lý do chọn đề tài

Giao thông đường bộ tại Việt Nam đang đối mặt với nhiều thách thức do tốc độ đô thị hóa nhanh, mật độ phương tiện cao và sự đa dạng về loại hình phương tiện, đặc biệt là tỷ lệ xe máy chiếm ưu thế. Việc quản lý và giám sát giao thông theo phương pháp thủ công hoặc bán tự động hiện nay còn gặp nhiều hạn chế về độ chính xác, chi phí và khả năng mở rộng.

Trong bối cảnh đó, thị giác máy tính (Computer Vision) đóng vai trò quan trọng trong việc xây dựng các hệ thống giao thông thông minh, cho phép máy tính tự động phân tích hình ảnh và video để nhận diện, phân loại và thống kê phương tiện giao thông. Ứng dụng Computer Vision giúp nâng cao hiệu quả giám sát, hỗ trợ điều tiết giao thông và cung cấp dữ liệu phục vụ phân tích, ra quyết định.

Tuy nhiên, phần lớn các bộ dữ liệu giao thông quốc tế hiện nay được xây dựng dựa trên điều kiện hạ tầng và đặc điểm phương tiện của các quốc gia phát triển, chưa phản ánh đầy đủ đặc thù giao thông tại Việt Nam, đặc biệt là sự phổ biến của xe máy và sự đa dạng về hình dáng phương tiện. Điều này làm giảm hiệu quả khi áp dụng trực tiếp các mô hình đã huấn luyện trên dữ liệu quốc tế. Xuất phát từ thực tế đó, đề tài lựa chọn bài toán phân loại ảnh phương tiện giao thông như một bước nền tảng trong các hệ thống giao thông thông minh.

Trong số các kiến trúc học sâu hiện nay, ResNet và EfficientNet là hai mô hình CNN tiêu biểu, đã chứng minh hiệu quả cao trong các bài toán phân loại ảnh. Việc lựa chọn và so sánh hai mô hình này giúp đánh giá khả năng ứng dụng của các kiến trúc hiện đại trên dữ liệu giao thông thực tế tại Việt Nam.

* + 1. Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng và đánh giá mô hình phân loại phương tiện giao thông đường bộ tại Việt Nam dựa trên dữ liệu ảnh.

Cụ thể, đề tài hướng tới việc xây dựng một bộ dữ liệu ảnh phương tiện giao thông phản ánh đặc điểm thực tế tại Việt Nam. Trên cơ sở đó, đề tài áp dụng và so sánh hai mô hình học sâu ResNet và EfficientNet nhằm đánh giá hiệu quả phân loại. Đồng thời, đề tài nghiên cứu ảnh hưởng của phương pháp tiền xử lý ảnh Unsharp Masking đến chất lượng dữ liệu đầu vào và hiệu suất của mô hình. Cuối cùng, đề tài phân tích ưu điểm và hạn chế của từng mô hình khi áp dụng trên dữ liệu thực tế.

* + 1. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là các ảnh tĩnh, trong đó mỗi ảnh chỉ chứa một phương tiện giao thông đường bộ. Các phương tiện được phân chia thành sáu lớp, bao gồm: xe máy, ô tô, xe buýt, xe tải, xe đạp và các phương tiện khác.

Phạm vi nghiên cứu của đề tài được giới hạn ở bài toán phân loại ảnh (image classification) trong lĩnh vực thị giác máy tính. Đề tài không xem xét các bài toán liên quan như phát hiện đối tượng, theo dõi phương tiện hay nhận dạng biển số. Các thí nghiệm được thực hiện trong môi trường thử nghiệm nhằm đánh giá hiệu quả của mô hình và phương pháp đề xuất.

* 1. CÁC NGHIÊN CỨU LIÊN QUAN
     1. Các kiến trúc mạng CNN trong nhận diện phương tiện

Trong nghiên cứu về phân loại phương tiện, các bộ khung (backbone) đóng vai trò quyết định trong việc trích xuất các đặc trưng hình ảnh từ cơ bản đến phức tạp. Nghiên cứu của Sanjaya và cộng sự [1] đã thực hiện một khảo sát thực nghiệm quan trọng trên các kiến trúc CNN hiện đại để xác định mô hình tối ưu cho nhận diện xe.

* + 1. ResNet và cơ chế Residual Learning

ResNet (Residual Network) được Sanjaya [1] sử dụng như một nền tảng nhờ vào đột phá trong việc giải quyết hiện tượng suy giảm đạo hàm (vanishing gradient) khi mạng quá sâu. Bằng cách sử dụng các "kết nối tắt" (skip connections), ResNet cho phép các dòng thông tin truyền trực tiếp qua các lớp, giúp mô hình có thể xây dựng cấu trúc cực sâu (như ResNet-50, ResNet-101). Trong nhận diện phương tiện, điều này giúp mạng học được các đặc trưng ngữ cảnh phức tạp mà không làm mất đi các chi tiết nhỏ ở các lớp đầu tiên.

* + 1. EfficientNet và tối ưu hóa đa chiều

Khác với cách tiếp cận tăng độ sâu thuần túy của ResNet, EfficientNet được Sanjaya [1] đánh giá cao nhờ cơ chế *Compound Scaling*. Cơ chế này điều chỉnh đồng thời ba yếu tố: độ sâu (depth), độ rộng (width) và độ phân giải (resolution) của hình ảnh đầu vào theo một hệ số cố định. Kết quả thực nghiệm trong bài báo cho thấy EfficientNet mang lại độ chính xác vượt trội trong khi sử dụng ít tài nguyên tính toán và bộ nhớ hơn so với các dòng mạng truyền thống, đặc biệt hiệu quả khi triển khai trên các hệ thống có cấu hình phần cứng hạn chế.

* + 1. Phân loại xe chi tiết (Fine-grained Classification)

Thử thách lớn nhất trong nhận diện phương tiện là sự khác biệt rất nhỏ giữa các dòng xe (ví dụ: phân biệt hai đời xe của cùng một hãng). Sử dụng bộ dữ liệu Stanford Cars, Sanjaya [1] chỉ ra rằng các mô hình CNN cần phải có khả năng tập trung vào các chi tiết cực nhỏ như lưới tản nhiệt, đèn pha hoặc đường cong cửa sổ. Bài báo nhấn mạnh rằng việc chọn backbone mạnh (như EfficientNet) là yếu tố tiên quyết để tách biệt các lớp dữ liệu có độ tương đồng cao này.

* + 1. Phát triển hệ thống phát hiện và theo dõi thời gian thực

Chuyển từ bài toán phân loại tĩnh sang giám sát động, Neupane và cộng sự [2] tập trung vào việc tích hợp khả năng nhận diện vào luồng video thực tế.

* + 1. Sự tiến hóa của dòng thuật toán YOLO

Nghiên cứu của Neupane [2] tập trung vào YOLOv5 – một bước tiến lớn so với YOLOv3. Mặc dù YOLO là mạng phát hiện vật thể (Object Detection), nhưng nó kế thừa các khối chức năng từ ResNet (như các lớp Residual) để tăng cường khả năng trích xuất đặc trưng. YOLOv5 được tối ưu hóa về tốc độ thông qua các lớp *Focus* và cấu trúc *CSPNet*, giúp hệ thống đạt được tốc độ xử lý nhanh hơn nhiều lần so với các phương pháp trích xuất đặc trưng truyền thống mà vẫn đảm bảo độ chính xác trong điều kiện giao thông phức tạp.

* + 1. Kỹ thuật Theo dõi đa đối tượng (Multi-Object Tracking - MOT)

Phát hiện vật thể chỉ là bước đầu; để giám sát giao thông, cần phải "giữ dấu" phương tiện qua từng khung hình. Neupane [2] thảo luận về việc kết hợp mô hình phát hiện với các thuật toán theo dõi để gán một ID duy nhất cho mỗi phương tiện. Quá trình này giúp tính toán lưu lượng giao thông và vận tốc, đồng thời giải quyết bài toán vật thể bị che khuất tạm thời khi di chuyển qua các điểm mù của camera.

* + 1. Phương pháp Học chuyển đổi (Transfer Learning)

Cả hai bài báo đều thống nhất rằng Học chuyển đổi là chiến lược tối ưu để huấn luyện mô hình hiệu quả.

* + 1. Tầm quan trọng của Pre-trained Models

Thay vì bắt đầu từ con số không, cả Sanjaya [1] và Neupane [2] đều tận dụng các mô hình đã được huấn luyện sẵn trên tập **ImageNet**. Việc này giúp mô hình sở hữu sẵn "khả năng thị giác" cơ bản (nhận diện đường kẻ, hình khối, màu sắc), từ đó chỉ cần tập trung học các đặc điểm riêng biệt của phương tiện giao thông.

* + 1. Tinh chỉnh mô hình (Fine-tuning)

Quá trình tinh chỉnh bao gồm việc giữ lại phần lớn các lớp tiền huấn luyện và thay thế lớp phân loại cuối cùng (Softmax) để phù hợp với số lượng loại xe cụ thể (như 196 loại xe trong Stanford Cars [1] hoặc các lớp xe cơ bản như Bus, Truck, Car trong camera giao thông [2]). Kỹ thuật này giúp giảm thiểu đáng kể thời gian huấn luyện và rủi ro quá tải (overfitting) khi tập dữ liệu mục tiêu không quá lớn.

* + 1. Các kỹ thuật tiền xử lý và tăng cường dữ liệu (Data Augmentation)

Dựa trên thực nghiệm của Sanjaya [1], việc tăng cường dữ liệu là bắt buộc để mô hình đạt tính bền vững. Các kỹ thuật như:

* **Xoay và cắt ảnh (Rotation/Cropping):** Giúp mô hình nhận diện xe ở nhiều góc độ và khoảng cách khác nhau.
* **Biến đổi màu sắc (Color Jittering):** Mô phỏng các điều kiện ánh sáng khác nhau (nắng, mưa, sương mờ). Điều này đảm bảo rằng mô hình không bị đánh lừa bởi các yếu tố ngoại cảnh khi triển khai thực tế.
  + 1. Đánh giá hiệu suất và các tiêu chí đo lường

Việc so sánh giữa hai nghiên cứu cho thấy sự khác biệt trong mục tiêu đánh giá:

* **Độ chính xác (Accuracy) & mAP:** Sanjaya [1] tập trung vào Accuracy để đánh giá khả năng phân loại đúng các mẫu xe chi tiết. Trong khi đó, Neupane [2] sử dụng mAP (mean Average Precision) – tiêu chuẩn vàng để đánh giá cả khả năng định vị và phân loại trong video.
* **Tốc độ xử lý (FPS):** Là ưu tiên hàng đầu trong nghiên cứu của Neupane [2] để đảm bảo hệ thống chạy được thời gian thực trên camera giám sát.
* **Kích thước mô hình (Model Size/Parameters):** Cả hai bài báo đều nhấn mạnh rằng một mô hình tốt không chỉ là mô hình chính xác nhất, mà còn phải đủ nhẹ để triển khai trên các thiết bị nhúng hoặc máy tính biên (Edge devices).
  1. MÔ TẢ DỮ LIỆU VÀ PHƯƠNG PHÁP TIỀN XỬ LÝ
     1. Mô tả bộ dữ liệu thu thập
        1. Tên bộ dữ liệu

Bộ dữ liệu phương tiện giao thông đường bộ tại Việt Nam.

* + - 1. Mục đích và bối cảnh

Bộ dữ liệu này góp phần hỗ trợ xây dựng và đánh giá các mô hình học máy và thị giác máy tính cho bài toán nhận dạng và phân loại phương tiện giao thông, đặc biệt hữu ích trong bối cảnh xây dựng hệ thống giám sát giao thông thông minh, đô thị thông minh, và hỗ trợ an toàn giao thông.

Ngoài ra bộ dữ liệu còn được tạo ra với mục tiêu hỗ trợ các nghiên cứu về nhận dạng phương tiện trong môi trường giao thông thực tế tại Việt Nam, nơi có mật độ xe cao và sự đa dạng về chủng loại phương tiện.

Dataset này có thể được ứng dụng trong các hệ thống như:

- Giám sát và quản lý giao thông thông minh

- Đếm và phân loại phương tiện tự động,

- Mô hình dự báo lưu lượng và hành vi phương tiện.

* + - 1. Nguồn thu thập dữ liệu

Bộ dữ liệu bao gồm khoảng hơn 2.000 hình ảnh tĩnh, thu thập bằng cách chụp trực tiếp trên các thiết bị di động khi xe lưu thông trên đường, các bãi giữ xe, ở trường học,...

Các thiết bị sử dụng trong quá trình thu thập dữ liệu là: iPhone 12 Pro Max, Samsung Galaxy Z Flip 4, iphone 7. Các bức ảnh ở định dạng JPG và PNG với kích thước lớn hơn 244x244 pixels. Mỗi ảnh chứa 1 phương tiện, được gán nhãn phân loại theo 6 lớp.

Video và hình ảnh được chụp và trích xuất trong nhiều điều kiện khác nhau về:

- Thời gian (ban ngày, ban đêm)

- Thời tiết (trời nắng, mưa nhẹ, nhiều mây)

- Mức độ đông đúc và góc nhìn camera (chính diện, chéo, trên cao)

Việc thu thập dữ liệu được tiến hành trong tháng 10 và tháng 11 năm 2025. Hình ảnh các phương tiện giao thông đường bộ được chụp vào những ngày khác nhau và vào các thời điểm khác nhau trong ngày ở nhiều địa điểm.

* + - 1. Bảng thu thập dữ liệu

Bảng 2. 1. Bảng mô tả thu thập dữ liệu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phương tiện** | **Thời tiết** | **Ngày** | **Thời gian** | **Thiết bị** | **Địa điểm** |
| 6 loại | Nắng | Tháng 10 và 11 | Trưa | iPhone 12 Pro Max, Samsung Galaxy Z Flip 4, iphone 7 | TP. Hồ Chí Minh |
| 6 loại | Nhiều mây | Tháng 10 và 11 | Tối | iPhone 12 Pro Max, Samsung Galaxy Z Flip 4, iphone 7 | TP. Hồ Chí Minh |

* + - 1. Bảng thông số kỹ thuật

Bảng 2. 2. Bảng mô tả thông số kỹ thuật dữ liệu

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| **Chủ đề** | Khoa học máy tính, Hệ thống Giao thông Thông minh (ITS) |
| **Lĩnh vực cụ thể** | Phân loại Ảnh (Image Classification), Thị giác máy tính, Học máy |
| **Loại dữ liệu** | Hình ảnh .PNG .JPG (Mỗi ảnh chứa 1 phương tiện) |
| **Thu thập dữ liệu** | Hình ảnh được chụp bằng camera Iphone 14. Mỗi mẫu phương tiện được tách/cắt ra khỏi bối cảnh gốc, đảm bảo chỉ có một phương tiện duy nhất trong mỗi ảnh. |
| **Vị trí/Nguồn dữ liệu** | Các tuyến đường chính tại TP. Hồ Chí Minh và Tây Ninh, Việt Nam. |

* + - 1. Thống kê bộ dữ liệu

Bộ dữ liệu 2000 ảnh được phân chia thành 6 lớp:

Bảng 2. 3. Bảng mô tả các lớp của bộ dữ liệu

| **Tên lớp (Classes Name)** | **Số lượng ảnh gốc (Original Images)** | **Mô tả phạm vi lớp** | **Ảnh minh họa** |
| --- | --- | --- | --- |
| **Xe ô tô (Car)** | 375 | Sedan, Hatchback, SUV/CUV thông thường Xe 4 bánh, xe bán tải. |  |
| **Xe tải (Truck)** | 325 | Xe tải thùng, xe container, xe ben. |  |
| **Xe khách (Bus/Coach)** | 350 | Xe 16 chỗ trở lên, xe buýt thành phố. |  |
| **Xe máy (Motorbike)** | 345 | Xe máy, xe tay ga, xe đạp điện có hình dáng xe máy. |  |
| **Xe đạp (Bicycle/E-bike)** | 350 | Xe đạp, xe đạp điện/trợ lực có hình dáng xe đạp. |  |
| **Xe khác** | 255 | Các xe không thuộc 5 lớp trên |  |

* + 1. Phương pháp tiền xử lý dữ liệu
       1. Chuẩn hóa kích thước ảnh

Sử dụng thuật toán nội suy (interpolation) để thay đổi kích thước hình ảnh

* + - 1. Thuật toán INTER\_AREA là gì?

INTER\_AREA (Resampling using pixel area relation) hoạt động dựa trên mối quan hệ giữa các vùng pixel.

- Khi thu nhỏ (Downsampling): Đây là thuật toán tối ưu nhất trong OpenCV. Thay vì chỉ lấy mẫu một vài pixel điểm, nó sẽ lấy trung bình cộng (hoặc tổng trọng số) của tất cả các pixel trong vùng bị thu hẹp để tạo ra pixel mới. Điều này giúp tránh hiện tượng moiré (vân sọc) và giữ được độ sắc nét tốt hơn so với các phương pháp khác.

- Khi phóng to (Upsampling): Nó hoạt động tương tự như thuật toán nội suy lân cận gần nhất (Nearest Neighbor).

* + - 1. So sánh với các thuật toán khác trong OpenCV

Bảng 2. 4. Bảng so sánh các thuật toán chuẩn hóa kích thước ảnh

| **Thuật toán** | **Đặc điểm** | **Trường hợp sử dụng** |
| --- | --- | --- |
| INTER\_NEAREST | Lấy pixel gần nhất. | Nhanh nhất nhưng chất lượng kém, dễ bị răng cưa. |
| INTER\_LINEAR | Nội suy tuyến tính (mặc định của OpenCV). | Tốc độ và chất lượng cân bằng, tốt cho việc phóng to. |
| INTER\_CUBIC | Nội suy bậc ba (trên vùng 4x4 pixel). | Chậm hơn nhưng ảnh mượt hơn, rất tốt cho việc phóng to. |
| INTER\_AREA | Dựa trên diện tích vùng pixel. | Tốt nhất cho việc thu nhỏ ảnh (như size 224x224 nhóm đang chọn). |

* + - 1. Tại sao dùng INTER\_AREA

Trong các bài toán Deep Learning (như sử dụng MobileNet, ResNet với đầu vào thường là 224x224), việc thu nhỏ ảnh từ độ phân giải cao (ví dụ Full HD) xuống 224x224 là rất phổ biến.

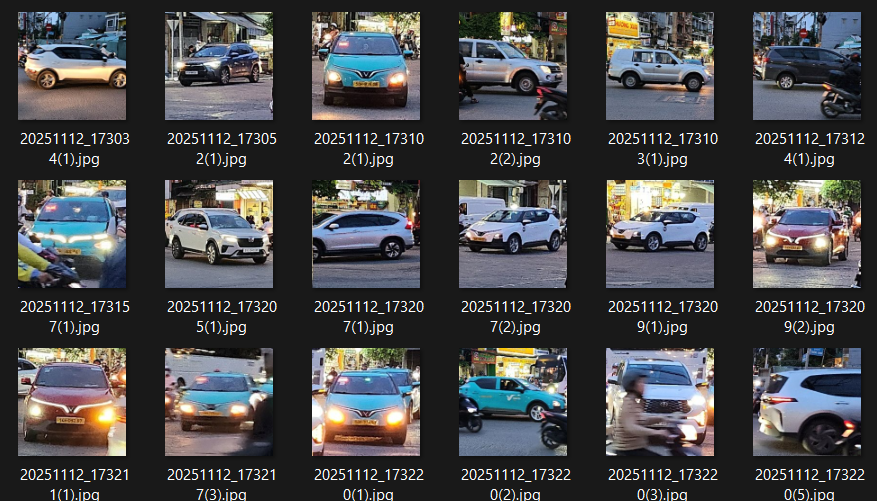
Việc dùng INTER\_AREA giúp:

- Tránh mất mát thông tin: Giảm thiểu việc bỏ sót các chi tiết nhỏ khi nén ảnh.

- Giảm nhiễu: Loại bỏ hiện tượng nhiễu răng cưa thường gặp khi thu nhỏ ảnh bằng phương pháp nội suy tuyến tính thông thường.

* + - 1. Làm nét ảnh bằng Unsharp Masking

Đây là một kỹ thuật kinh điển trong xử lý ảnh giúp tăng cường độ tương phản tại các đường biên (edges), làm cho mắt người cảm thấy ảnh rõ nét hơn.



Hình 2. 1. Ảnh sau khi được làm nét.

* + - 1. Nguyên lý hoạt động của Unsharp Masking (USM)

Thuật toán này hoạt động dựa trên logic: Ảnh sắc nét = Ảnh gốc + (Ảnh gốc - Ảnh mờ).

Các bước được thực hiện như sau:

1. Làm mờ (Blurring): blurred = cv2.GaussianBlur(img, (0, 0), sigmaX=2.0) Tạo ra một bản sao bị mờ của ảnh gốc bằng bộ lọc Gaussian. Bản sao này đóng vai trò xác định các vùng "nền" và loại bỏ các chi tiết tần số cao.

2. Trừ ảnh (Masking): Khi lấy ảnh gốc trừ đi ảnh mờ, bạn sẽ thu được một "mặt nạ" chứa các chi tiết góc cạnh và đường biên (tần số cao).

3. Tăng cường (Sharpening): sharpened = cv2.addWeighted(img, 1.5, blurred, -0.5, 0) Dòng lệnh này thực thi công thức trộn ảnh:



Kết quả là các đường biên được đẩy độ tương phản lên cao, khiến ảnh trông sắc nét hơn đáng kể.

Các tham số quan trọng trong code:

- sigmaX=2.0: Quyết định mức độ lan tỏa của bộ lọc mờ. Giá trị càng lớn, các đường biên được xác định sẽ càng rộng.

- Trọng số 1.5 và -0.5: Đây là hệ số khuếch đại (Amount).

- Nếu bạn tăng 1.5 lên (ví dụ 2.0) và giảm -0.5 xuống (ví dụ -1.0), ảnh sẽ cực kỳ sắc nét nhưng có thể bị nhiễu hạt (noise).

* + 1. Chia tập dữ liệu
       1. Các quy tắc "vàng" khi chia dữ liệu

Tính ngẫu nhiên (Randomness): Phải xáo trộn dữ liệu trước khi chia để tránh trường hợp tập Train toàn ảnh ban ngày, còn tập Test toàn ảnh ban đêm.

Tính cân bằng (Stratification): Đảm bảo tỷ lệ các lớp (ví dụ: số lượng ảnh xe máy, ô tô, xe đạp) trong các tập phải tương đương nhau.

Không rò rỉ (Data Leakage): Tuyệt đối không để một ảnh xuất hiện ở cả hai tập. Nếu ảnh "đã biết" lọt vào tập Test, kết quả sẽ bị ảo.

* + - 1. Tại sao phải chia tập dữ liệu

Bảng 2. 5. Bảng mô tả lý do cần phải phân chia tập dữ liệu

|  |  |
| --- | --- |
| Lý do | Giải thích |
| Tránh học vẹt (Overfitting) | Nếu dùng toàn bộ dữ liệu để học, mô hình sẽ ghi nhớ thay vì hiểu quy luật. Khi gặp ảnh thực tế bên ngoài, nó sẽ nhận diện sai. |
| Đánh giá khách quan | Không thể tự chấm điểm bài làm của mình bằng chính những câu hỏi bạn đã biết đáp án. Cần một bộ đề mới (Test set). |
| Lựa chọn mô hình tốt nhất | Có thể thử nghiệm nhiều kiến trúc (ResNet, YOLO, MobileNet) và dùng tập Validation để chọn ra cái tốt nhất. |

**- Tập huấn luyện (Training set):** chiếm ~70-80%, đây là tập dữ liệu lớn nhất, dùng để "dạy" mô hình:

+ Mục đích: Mô hình sẽ nhìn vào các ảnh này và nhãn tương ứng (ví dụ: ảnh con mèo - nhãn "mèo") để học các đặc trưng như hình dáng, màu sắc, đường nét.

+ Hoạt động: Trong quá trình này, các tham số bên trong mạng thần kinh sẽ được điều chỉnh liên tục để giảm thiểu sai số.

- **Tập kiểm định(Validation set):** chiếm ~10-15%, đây là tập dữ liệu "thi thử" trong quá trình huấn luyện:

+ Mục đích: Dùng để tinh chỉnh các siêu tham số (hyperparameters) như tốc độ học (learning rate), số lớp (layers), hoặc kích thước bộ lọc.

+ Quan trọng: Nó giúp phát hiện hiện tượng Overfitting (Học vẹt) — khi mô hình học thuộc lòng tập train nhưng không nhận diện được ảnh mới.

- **Tập kiểm tra (Test set**): chiếm ~ 10-15%, đây là tập dữ liệu hoàn toàn mới, mô hình chưa từng được thấy trong suốt quá trình học và chỉnh sửa. Mục đích: Đánh giá khách quan sức mạnh thực tế của mô hình. Kết quả từ tập này chính là "điểm số thực" mà mô hình đạt được khi mang ra sử dụng thực tế.

* 1. PHƯƠNG PHÁP PHÂN LOẠI
     1. Mô hình ResNet
        1. Giới thiệu

Trong lĩnh vực trí tuệ nhân tạo (AI) và học sâu (Deep Learning), việc phát triển các mô hình mạng nơ-ron với độ sâu lớn thường đối mặt với nhiều thách thức. Một trong những thách thức lớn nhất là degradation problem – vấn đề suy giảm hiệu suất khi mô hình trở nên quá sâu. Để giải quyết vấn đề này, mô hình ResNet (Residual Network) đã được giới thiệu và nhanh chóng trở thành một công cụ mạnh mẽ trong việc nhận diện hình ảnh và nhiều ứng dụng khác.

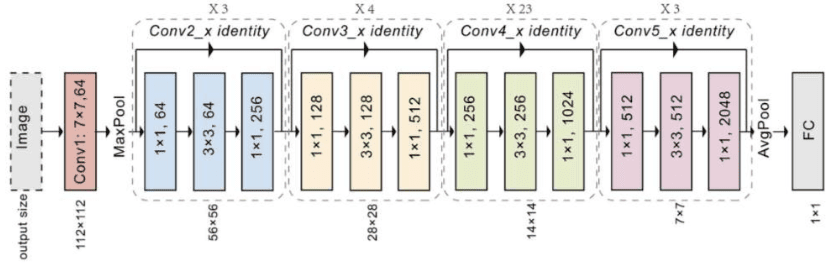
Mô hình Resnet được giới thiệu bởi Kaiming He và các cộng sự trong bài báo nổi tiếng “Deep Residual Learning for Image Recognition” vào năm 2015. Ý tưởng đột phá của ResNet là sử dụng skip connections hoặc residual connections giữa các lớp để giải quyết vấn đề suy giảm độ chính xác khi mô hình trở nên sâu hơn. Điều này cho phép thông tin được truyền qua các lớp của mạng mà không bị suy giảm, giúp mô hình có thể học được các đặc trưng (features) phức tạp từ dữ liệu.

A diagram of a cat's body

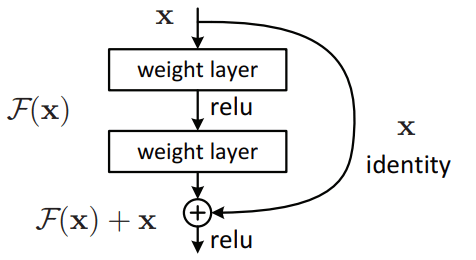
AI-generated content may be incorrect.

* + - 1. Cấu trúc và hoạt động ResNet

Cấu trúc của ResNet được xây dựng dựa trên việc chia nhỏ mạng thành các khối residual (Residual Blocks), trong đó mỗi khối bao gồm một số lớp tích chập (convolutional layers) cùng với một kết nối tắt trực tiếp từ đầu vào đến đầu ra của khối. Ý tưởng chính là việc học một hàm residual F(x) thay vì học trực tiếp một hàm mục tiêu H(x). Điều này giúp duy trì tính ổn định và độ chính xác khi mạng có nhiều tầng.



* + - 1. Khối residual cơ bản



Một khối residual cơ bản trong ResNet có cấu trúc như sau:

A black and white text

AI-generated content may be incorrect.

trong đó:

x: là đầu vào của khối.

y: là đầu ra của khối.

A group of black letters

AI-generated content may be incorrect.: là hàm residual, thường được biểu diễn bằng một chuỗi các lớp tích chập với trọng số Wi

* + - 1. Nguyên lý hoạt động ResNet

Nguyên lý hoạt động của ResNet dựa trên việc thay vì học trực tiếp hàm H(x), mô hình sẽ học một hàm residual  Sau đó, đầu ra của mô hình sẽ là: A black and white math equation

AI-generated content may be incorrect.

Cách tiếp cận này dựa trên giả định rằng việc học một hàm sai khác F(x) sẽ dễ dàng hơn so với việc học trực tiếp hàm H(x). Khi mô hình học được F(x) , nó thực chất đang học cách “điều chỉnh” đầu vào x để đạt được đầu ra mong muốn H(x) . Nhờ vào các kết nối tắt (skip connections), các tín hiệu thông tin có thể dễ dàng được truyền qua mạng mà không bị suy giảm hoặc mất mát, đặc biệt khi mạng trở nên rất sâu.

Các kết nối tắt trong ResNet mang lại nhiều lợi ích, bao gồm:

**Giảm thiểu vanishing gradient:** Khi mạng trở nên rất sâu, các giá trị gradient có thể bị giảm mạnh (vanishing) hoặc tăng mạnh (exploding), gây khó khăn trong quá trình huấn luyện. Kết nối tắt giúp giữ lại các giá trị gradient, giảm thiểu hiện tượng này.

**Dễ dàng học các đặc trưng**: Việc học một hàm residual F(x) thường đơn giản hơn so với việc học một hàm ánh xạ hoàn toàn mới H(x), do đó mô hình có thể nhanh chóng đạt được độ chính xác cao hơn.

**Tăng độ sâu mà không giảm hiệu suất**: Với các kết nối tắt, ResNet có thể được mở rộng đến hàng trăm hoặc hàng ngàn tầng mà không gặp phải vấn đề suy giảm hiệu suất.

* + 1. Mô hình EfficientNet
       1. Giới thiệu

Trong những năm gần đây, trí tuệ nhân tạo (AI) và học sâu (deep learning) đã có những bước tiến vượt bậc, với nhiều ứng dụng đột phá trong nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và nhiều lĩnh vực khác.

Trong đó, các mô hình mạng neural convolutional (CNNs) đã trở thành nền tảng cho nhiều hệ thống AI hiện đại. Một trong những kiến trúc CNN nổi bật nhất trong thời gian gần đây là EfficientNet, được giới thiệu bởi nhóm nghiên cứu của Google Brain vào năm 2019. EfficientNet không chỉ đem lại hiệu suất cao mà còn tối ưu hóa về mặt tài nguyên, giúp mô hình hoạt động tốt trên cả các thiết bị hạn chế về phần cứng.

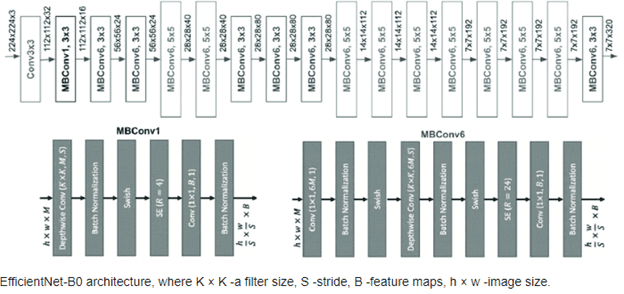
Các kiến trúc CNN cổ điển như AlexNet, VGGNet, và ResNet mang lại những bước nhảy vọt trong hiệu suất của các hệ thống nhận diện hình ảnh. Tuy nhiên, một trong những vấn đề lớn với các mô hình này là sự phức tạp và yêu cầu tài nguyên phần cứng rất cao.

**EfficientNet** được thiết kế để giải quyết những hạn chế này bằng cách đề xuất một phương pháp mở rộng cân bằng (compound scaling). Thay vì tăng một trong các yếu tố như depth, width, hoặc resolution, EfficientNet mở rộng tất cả các yếu tố này theo một tỷ lệ nhất định, dựa trên việc tối ưu hóa hiệu suất và tài nguyên phần cứng.



* + - 1. Kiến trúc

Kiến trúc của EfficientNet dựa trên mô hình nền tảng là MobileNetV2 và MnasNet, hai mô hình được tối ưu hóa cho các thiết bị di động. EfficientNet áp dụng phương pháp compound scaling để mở rộng mô hình từ một phiên bản nhỏ (EfficientNet-B0) lên các phiên bản lớn hơn (EfficientNet-B1 đến B7), trong đó B7 là phiên bản lớn nhất và chính xác nhất.



* + - 1. Compound Scaling

Compound Scaling trong EfficientNet là một phương pháp mở rộng mô hình theo ba chiều chính: chiều sâu (depth), chiều rộng (width), và độ phân giải ảnh (resolution). Thay vì chỉ tăng một chiều như các phương pháp truyền thống, EfficientNet đồng thời mở rộng cả ba chiều theo một tỷ lệ cân bằng đã được tối ưu hóa. Cách tiếp cận này giúp cải thiện hiệu suất mô hình mà không làm tăng quá mức số lượng tham số hoặc yêu cầu tài nguyên tính toán, mang lại độ chính xác cao hơn với hiệu quả tính toán tối ưu.

* + - 1. Mobile Inverted Bottleneck Convolution (MBConv)

Một trong những điểm đặc biệt của EfficientNet là sử dụng các khối MBConv, tương tự như trong MobileNetV2. Khối MBConv kết hợp giữa các convolution điểm 1×1 (pointwise convolution) và depthwise convolution, giúp giảm số lượng tham số nhưng vẫn giữ được độ chính xác cao.

* + - 1. Squeeze-and-Excitation (SE) Blocks

EfficientNet cũng sử dụng các khối Squeeze-and-Excitation (SE) để tự động điều chỉnh trọng số của các kênh, giúp mô hình tập trung vào các kênh quan trọng hơn. SE block hoạt động bằng cách học các hệ số điều chỉnh (attention) cho từng kênh, dựa trên giá trị tổng hợp của toàn bộ không gian đầu vào.

* + - 1. Ứng dụng

EfficientNet đã được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ nhận diện hình ảnh, phân loại hình ảnh, đến các bài toán phức tạp hơn như phát hiện đối tượng (object detection) và phân đoạn ảnh (image segmentation).

**Nhận diện và phân loại hình ảnh:** EfficientNet đã chứng minh được khả năng vượt trội trong các bài toán nhận diện và phân loại hình ảnh, với hiệu suất tốt trên nhiều tập dữ liệu chuẩn như ImageNet. Đặc biệt, EfficientNet đạt được độ chính xác hàng đầu trong khi yêu cầu ít tài nguyên hơn so với các mô hình khác.

**Phát hiện đối tượng và phân đoạn ảnh**: EfficientNet cũng được tích hợp vào các hệ thống phát hiện đối tượng như YOLO và SSD, cũng như các mô hình phân đoạn ảnh như U-Net. Với khả năng mở rộng linh hoạt, EfficientNet giúp các hệ thống này hoạt động nhanh hơn và chính xác hơn.

**Ứng dụng trong công nghiệp:** EfficientNet đã được sử dụng trong nhiều ứng dụng thực tế, từ kiểm tra chất lượng sản phẩm trong dây chuyền sản xuất, nhận diện khuôn mặt trong an ninh, đến phân tích hình ảnh y tế. Nhờ khả năng hoạt động hiệu quả trên cả các thiết bị phần cứng giới hạn, EfficientNet trở thành lựa chọn hàng đầu trong nhiều ứng dụng công nghiệp.

* + 1. Ưu nhược điểm của ResNet và EfficientNet
       1. Ưu điểm ResNet

**Khắc phục hiện tượng mất gradient (vanishing gradient):** ResNet sử dụng residual connection (kết nối tắt) cho phép dòng gradient lan truyền trực tiếp qua lớp sâu, giúp huấn luyện mạng rất sâu (50, 101, 152 lớp) một cách ổn định.

**Khả năng mở rộng độ sâu tốt:** Việc thêm các khối residual không làm giảm hiệu năng, thậm chí còn cải thiện độ chính xác trong nhiều bài toán thị giác máy tính.

**Hiệu năng cao và ổn định:** ResNet đạt kết quả tốt trên nhiều bộ dữ liệu chuẩn như ImageNet, CIFAR, và được xem là kiến trúc nền tảng cho nhiều mô hình cải tiến sau này.

**Dễ tích hợp và tái sử dụng:** ResNet thường được dùng làm backbone trong các bài toán phân loại ảnh, phát hiện đối tượng và phân đoạn ảnh.

* + - 1. Nhược điểm của ResNet

**Số lượng tham số lớn:** Phiên bản ResNet sâu yêu cầu nhiều bộ nhớ và tài nguyên tính toán.

**Hiệu quả tính toán chưa tối ưu:** ResNet không được thiết kế tối ưu cho các thiết bị tài nguyên hạn chế như thiết bị di động hoặc IoT.

**Chưa khai thác tối đa sự cân bằng giữa độ sâu, độ rộng và độ phân giải:** Kiến trúc chủ yếu tập trung vào việc tăng độ sâu của mạng.

* + - 1. Ưu điểm EfficientNet

**Chiến lược mở rộng hợp lý (Compound Scaling):** EfficientNet đồng thời mở rộng độ sâu, độ rộng và độ phân giải ảnh, mô hình độ chính xác cao hơn với số tham số ít hơn.

**Hiệu quả cao về tài nguyên:** So với ResNet, EfficientNet đạt độ chính xác tương đương hoặc cao hơn với ít FLOPs và tham số hơn.

**Phù hợp với thiết bị hạn chế tài nguyên:** Các phiên bản EfficientNet (B0–B7) có thể linh hoạt lựa chọn theo năng lực phần cứng.

**Hiệu năng vượt trội trên ImageNet:** EfficientNet thể hiện sự cải thiện rõ rệt so với các kiến trúc CNN truyền thống.

* + - 1. Nhược điểm EfficientNet

**Cấu trúc phức tạp:** Việc thiết kế và tinh chỉnh EfficientNet phức tạp hơn ResNet, khó tùy chỉnh thủ công.

**Thời gian huấn luyện dài:** Việc sử dụng ảnh có độ phân giải cao làm tăng chi phí huấn luyện.

**Khó giải thích hơn:** Do kiến trúc phức tạp, việc phân tích và diễn giải mô hình trở nên khó khăn hơn.

* 1. THỰC NGHIỆM

Tập dữ liệu thực nghiệm trong đề tài này được xây dựng nhằm phục vụ cho bài toán phân loại phương tiện giao thông đường bộ tại Việt Nam từ ảnh tĩnh. Dữ liệu được thu thập, tiền xử lý và tổ chức một cách thống nhất để đảm bảo phù hợp với việc huấn luyện và đánh giá các mô hình học sâu, cụ thể là ResNet và EfficientNet.

* + 1. Mô tả tập dữ liệu thực nghiệm

Sau quá trình thu thập và làm sạch dữ liệu, tập dữ liệu được đưa qua các bước tiền xử lý nhằm đảm bảo chất lượng và tính đồng nhất của ảnh đầu vào. Cụ thể:

- Mỗi ảnh chỉ chứa một phương tiện giao thông chính, phù hợp với bài toán phân loại ảnh (image classification).

- Tất cả các ảnh được crop và resize về kích thước 224 × 224 pixel, đảm bảo tương thích với kiến trúc đầu vào của cả hai mô hình ResNet và EfficientNet.

- Phương pháp làm nét ảnh Unsharp Masking được áp dụng nhằm tăng cường độ sắc nét của biên và chi tiết phương tiện, đặc biệt hữu ích đối với các ảnh bị mờ nhẹ do chuyển động hoặc điều kiện chụp không lý tưởng.

- Tăng cường dữ liệu (Data Augmentation) đối với tập huấn luyện: sử dụng một số phép biến đổi dữ liệu nhẹ nhằm tăng khả năng tổng quát hóa của mô hình, bao gồm:

+ Lật ngang ngẫu nhiên (Random Horizontal Flip

+ Xoay ảnh ngẫu nhiên với góc nhỏ (Random Rotation)

+ Điều chỉnh độ sáng và màu sắc (Color Jitter).

Đối với tập validation và test, ảnh chỉ được resize và chuẩn hóa, không áp dụng augmentation để đảm bảo tính khách quan trong đánh giá mô hình.

Sau khi hoàn tất quá trình tiền xử lý, tập dữ liệu thực nghiệm gồm 2.000 ảnh được chia thành ba tập con phục vụ cho quá trình huấn luyện và đánh giá mô hình, bao gồm:

+ Tập huấn luyện (Training set)

+ Tập xác thực (Validation set)

+ Tập kiểm thử (Test set)

Việc phân chia dữ liệu được thực hiện theo tỷ lệ 80% – 10% – 10%, đồng thời đảm bảo sự phân bố tương đối đồng đều giữa các lớp phương tiện trong từng tập dữ liệu. Cụ thể, số lượng ảnh trong mỗi tập được trình bày như sau:

|  |  |
| --- | --- |
| **Tập dữ liệu** | **Số lượng ảnh** |
| Training | 1600 |
| Validation | 200 |
| Test | 200 |
| Tổng cộng | 2000 |

Cách phân chia này giúp mô hình có đủ dữ liệu để học các đặc trưng của phương tiện giao thông, đồng thời vẫn đảm bảo có tập validation để theo dõi quá trình huấn luyện và tập test độc lập để đánh giá hiệu năng cuối cùng.

Trong quá trình huấn luyện mô hình, dữ liệu được nạp theo từng mini-batch thông qua DataLoader với các tham số cấu hình như sau:

- Kích thước batch (Batch size): 32

- Số luồng xử lý song song (Number of workers): 4

****

Hình 4. 1. Kết quả hình ảnh sau khi thực hiện tiền xử lý

Một số ảnh mẫu sau tiền xử lý đã được trực quan hóa trong quá trình thực nghiệm nhằm kiểm tra chất lượng dữ liệu đầu vào. Các ảnh minh họa cho thấy:

- Kích thước ảnh đã được đồng nhất về 224 × 224 pixel,

- Phương tiện giao thông được thể hiện rõ ràng sau khi áp dụng Unsharp Masking,

- Dữ liệu phản ánh đa dạng góc chụp, bối cảnh và điều kiện ánh sáng của môi trường giao thông thực tế tại Việt Nam.

Việc minh họa ảnh mẫu giúp xác nhận tính hợp lệ của quy trình tiền xử lý và đảm bảo dữ liệu phù hợp trước khi tiến hành huấn luyện mô hình.

* + 1. Quy trình thực hiện

Quy trình thực nghiệm của đề tài được xây dựng theo các bước chính sau:

- Thu thập dữ liệu ảnh phương tiện giao thông

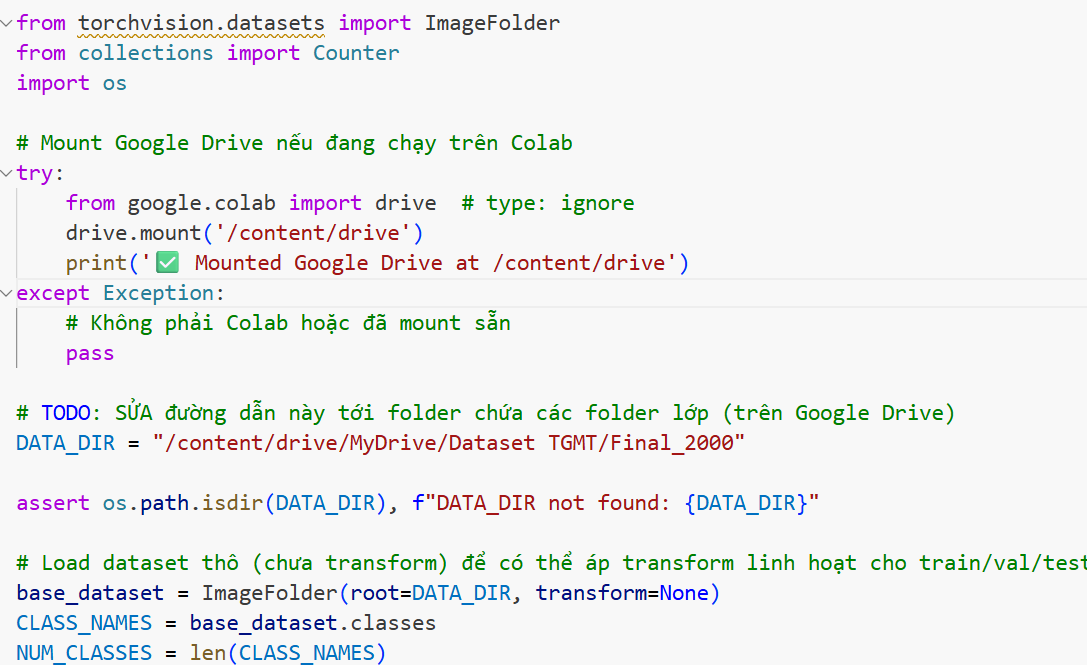
- Tiền xử lý ảnh (resize và làm nét bằng Unsharp Masking)

- Huấn luyện mô hình (ResNet và EfficientNet)

- Đánh giá và so sánh kết quả

**\*Tổ chức dữ liệu**

Dữ liệu ảnh được tổ chức theo cấu trúc thư mục lớp và được nạp vào chương trình thông qua lớp ImageFolder của thư viện torchvision.

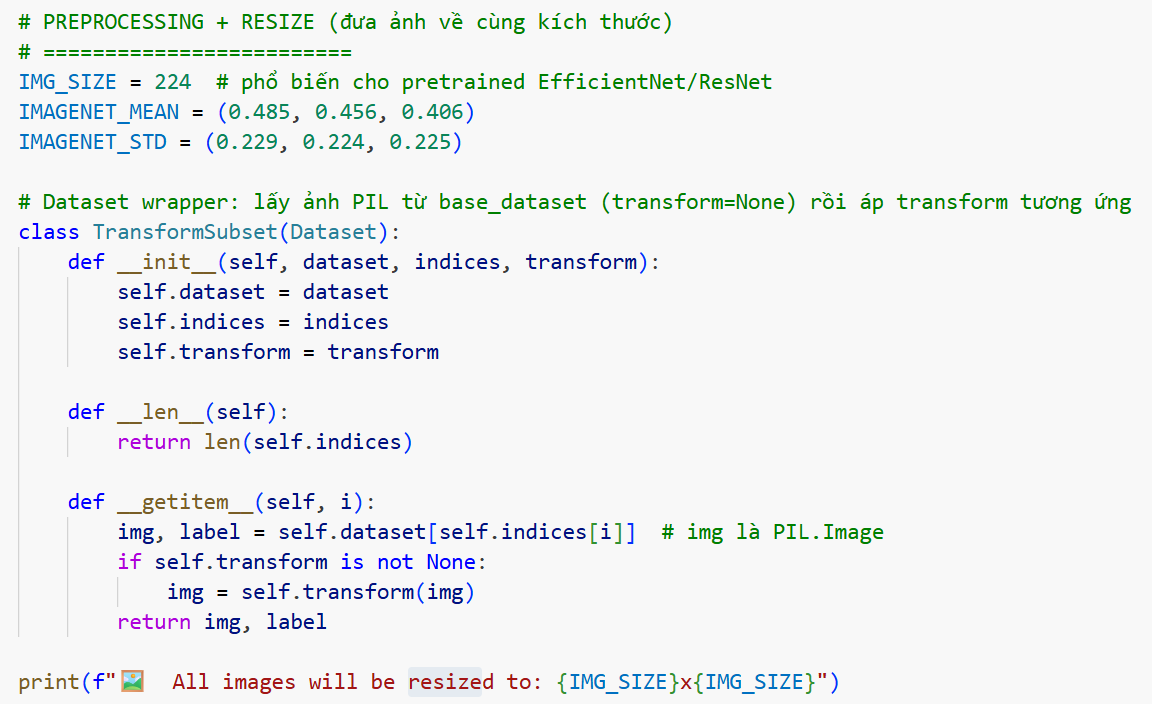


Hình 4. 2. Đoạn code thực hiện load dataset

Cách tổ chức này cho phép hệ thống tự động gán nhãn ảnh dựa trên tên thư mục, giúp đơn giản hóa quá trình quản lý dữ liệu.

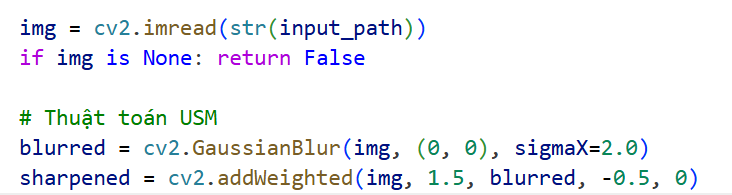
**\*Tiền xử lý ảnh:**

Tất cả ảnh đầu vào được chuẩn hóa kích thước về 224 × 224 pixel, phù hợp với đầu vào của các mô hình ResNet và EfficientNet.



Hình 4. 3. Đoạn code thực hiện resize ảnh

Làm nét ảnh bằng Unsharp Masking: Để tăng cường các chi tiết biên của phương tiện giao thông, phương pháp Unsharp Masking được áp dụng trong giai đoạn tiền xử lý.



Hình 4. 4. Đoạn code thực hiện làm nét ảnh

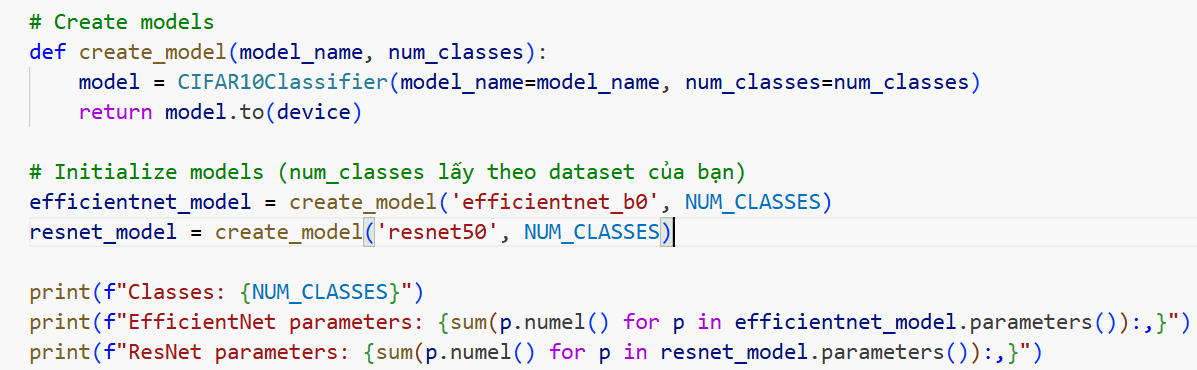
Phương pháp này giúp cải thiện độ sắc nét của ảnh, đặc biệt đối với các ảnh bị mờ nhẹ do chuyển động hoặc điều kiện chụp không lý tưởng.

**\*Chuẩn hóa dữ liệu và augmentation:** Đối với tập huấn luyện, các phép tăng cường dữ liệu được áp dụng nhằm giảm hiện tượng overfitting. Đối với tập validation và test, dữ liệu chỉ được resize và chuẩn hóa.



Hình 4. 5. Đoạn code thực hiện chuẩn hóa dữ liệu và augmentation

**\*Huấn luyện mô hình:** Hai mô hình ResNet và EfficientNet được khởi tạo với trọng số pretrained trên ImageNet, sau đó thay thế lớp phân loại cuối để phù hợp với số lượng lớp trong bài toán. Quá trình huấn luyện được thực hiện với các tham số chính, batch size: 32, number of workers: 4



Hình 4. 6. Đoạn code thực hiện khởi tạo model

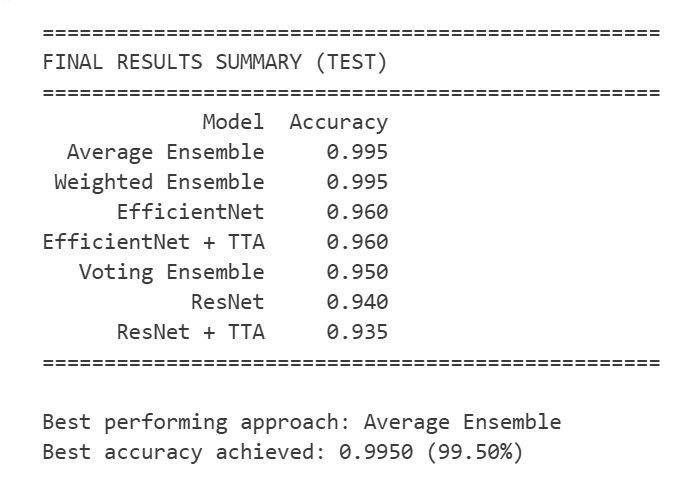


Hình 4. 7. Đoạn code thực hiện huấn luyện mô hình

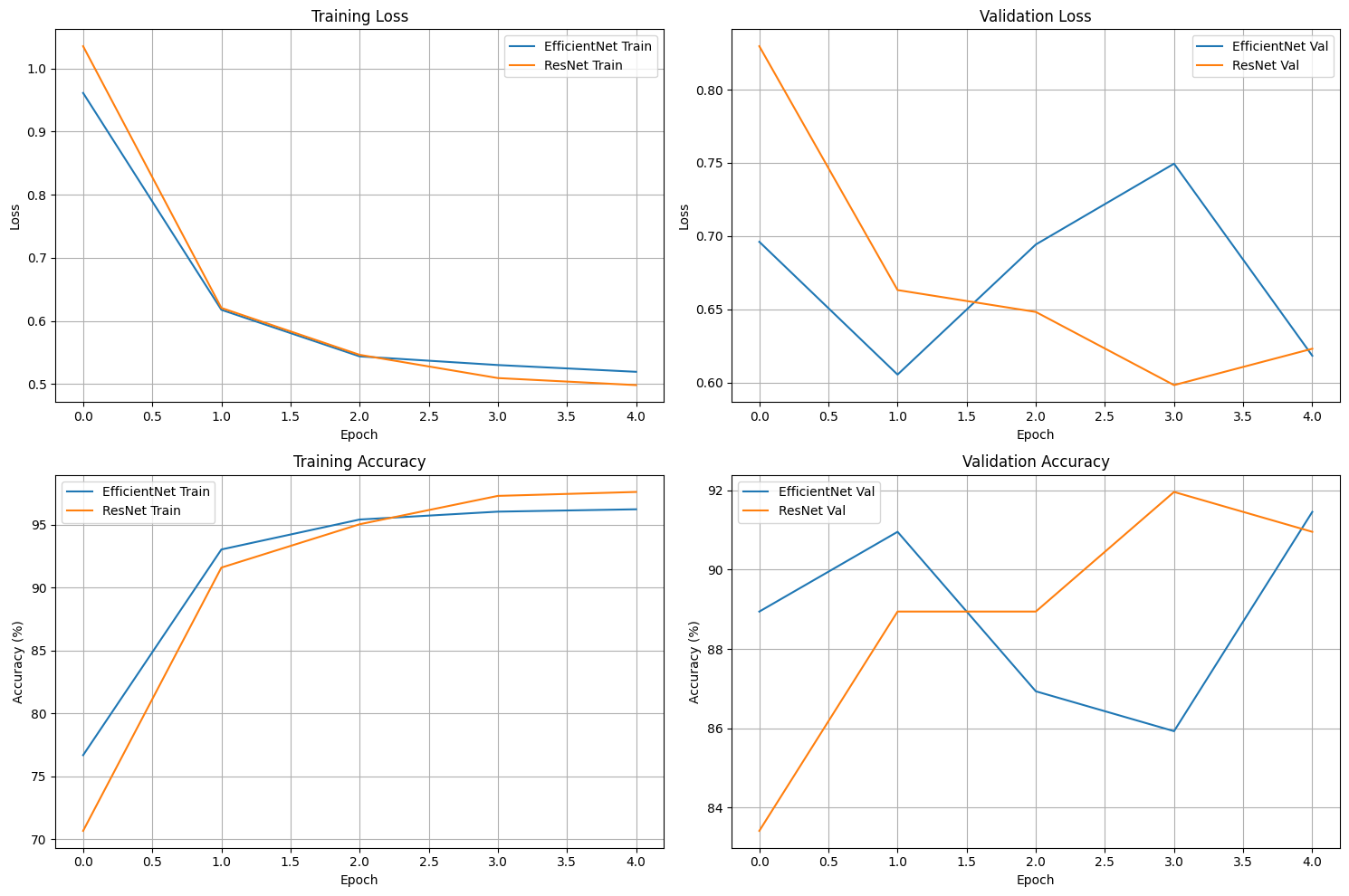
**\*Đánh giá kết quả:** Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm thử độc lập. Độ chính xác của mô hình được tính dựa trên số lượng dự đoán đúng trên tổng số mẫu kiểm thử.



Hình 4. 8. Đoạn code thực hiện đánh giá mô hình



Hình 4. 9. Kết quả nhận được sau thực thi đánh giá mô hình



Hình 4. 10. Biểu đồ các thông số sau training, accuracy giữa 2 mô hình

**Môi trường thực nghiệm:** Phần cứng sử dụng GPU Tesla T4 của Google Colab, phần mềm và famework: sử dụng python, pytorch, openCV,…

* + 1. So sánh kết quả giữa mô hình ResNet và EfficientNet

Trong mục này, hiệu năng của các mô hình ResNet và EfficientNet được đánh giá và so sánh dựa trên kết quả thực nghiệm thu được từ tập kiểm thử (test set). Ngoài ra, các phương pháp cải tiến như Test-Time Augmentation (TTA) và Ensemble cũng được xem xét nhằm đánh giá khả năng nâng cao độ chính xác của hệ thống.

Các mô hình được đánh giá trên tập kiểm thử gồm 200 ảnh, sử dụng độ chính xác (Accuracy) làm chỉ số đánh giá chính. Bên cạnh đó, kiểm chứng trực quan các mẫu dự đoán đúng/sai và phân tích lỗi theo từng lớp cũng được thực hiện để đánh giá sâu hơn khả năng tổng quát hóa của mô hình.

Từ kết quả nhận được từ Hình 4.9, ta nhận được:

- EfficientNet đạt độ chính xác 96.0%, cao hơn đáng kể so với ResNet (94.0%) trong cùng điều kiện huấn luyện.

- Việc áp dụng TTA không cải thiện đáng kể hiệu năng của EfficientNet và thậm chí làm giảm nhẹ độ chính xác đối với ResNet.

- Các phương pháp Ensemble, đặc biệt là Average Ensemble và Weighted Ensemble, đạt độ chính xác cao nhất 99.5%, cho thấy khả năng kết hợp nhiều mô hình giúp cải thiện mạnh mẽ hiệu năng phân loại.

Phương pháp đạt hiệu năng tốt nhất trong thí nghiệm là Average Ensemble, với độ chính xác cao nhất 99.50% trên tập kiểm thử.

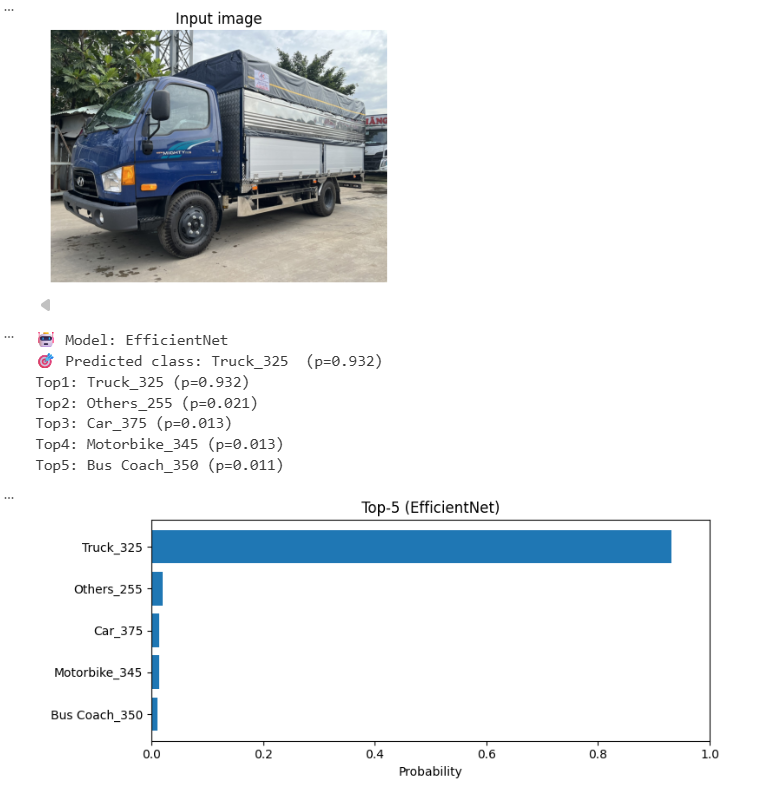
Để đánh giá sâu hơn hành vi của mô hình, các kết quả dự đoán của EfficientNet kiểm chứng trực quan trên tập kiểm thử. Kết quả cho thấy:

- 192 / 200 ảnh được dự đoán đúng.

- 8 / 200 ảnh bị dự đoán sai.



Hình 4. 11. Kết sau thi thực hiện kiểm tra trên bộ dữ liệu test



Hình 4. 12. Kết quả sau khi thực hiện upload bức ảnh ngẫu nhiên không nằm trong bộ test hay val

Ngoài đánh giá trên tập kiểm thử, mô hình còn được kiểm tra khả năng suy luận trên ảnh đơn tải lên từ bên ngoài. Kết quả dự đoán cho thấy mô hình có khả năng xác định đúng lớp phương tiện với xác suất cao, đồng thời cung cấp phân bố xác suất Top-k cho các lớp còn lại. Việc thử nghiệm inference trên ảnh ngoài tập huấn luyện cho thấy mô hình không chỉ hoạt động tốt trên dữ liệu test mà còn có khả năng tổng quát hóa trong các tình huống thực tế.

* 1. KẾT LUẬN

Trong đề tài này, bài toán phân loại phương tiện giao thông đường bộ tại Việt Nam đã được nhóm thực hiện tìm hiểu, thu thập dữ liệu và triển khai bằng các mô hình học sâu dựa trên mạng nơ-ron tích chập, cụ thể là ResNet và EfficientNet. Toàn bộ quá trình từ xây dựng tập dữ liệu, tiền xử lý ảnh, huấn luyện mô hình cho đến đánh giá kết quả đều được thực hiện một cách có hệ thống và khoa học.

* + 1. Kết luận

Trước hết, đề tài đã xây dựng thành công một tập dữ liệu ảnh phương tiện giao thông gồm 2.000 ảnh, phản ánh các điều kiện giao thông thực tế tại Việt Nam. Tập dữ liệu được tiền xử lý thống nhất và chuẩn hóa theo thống kê ImageNet, tạo nền tảng dữ liệu phù hợp cho việc huấn luyện các mô hình học sâu.

Về mặt mô hình, kết quả thực nghiệm cho thấy cả hai mô hình ResNet và EfficientNet đều có khả năng áp dụng hiệu quả cho bài toán phân loại phương tiện. Tuy nhiên, EfficientNet cho hiệu năng vượt trội hơn so với ResNet trong cùng điều kiện huấn luyện, với độ chính xác đạt 96.0% trên tập kiểm thử, trong khi ResNet đạt 94.0%.

Đặc biệt, khi áp dụng các phương pháp Ensemble, hiệu năng của hệ thống được cải thiện rõ rệt. Phương pháp Average Ensemble và Weighted Ensemble đạt độ chính xác cao nhất 99.5%, cho thấy việc kết hợp nhiều mô hình là một hướng tiếp cận hiệu quả nhằm nâng cao độ tin cậy của hệ thống phân loại.

Ngoài các kết quả định lượng, việc kiểm chứng trực quan các mẫu dự đoán đúng và sai cũng giúp làm rõ hành vi của mô hình, chỉ ra các trường hợp khó như sự nhầm lẫn giữa các lớp phương tiện có hình dạng tương đồng hoặc ảnh có bối cảnh phức tạp.

* + 1. Hạn chế và hướng phát triển

Mặc dù đạt được các kết quả khả quan, đề tài vẫn tồn tại một số hạn chế nhất định:

- Quy mô tập dữ liệu còn tương đối nhỏ so với các bộ dữ liệu lớn quốc tế, có thể ảnh hưởng đến khả năng tổng quát hóa của mô hình trong các kịch bản đa dạng hơn.

- Bài toán chỉ được xét trên ảnh tĩnh, chưa mở rộng sang dữ liệu video – vốn phổ biến trong các hệ thống camera giao thông thực tế.

- Đề tài tập trung vào bài toán phân loại, chưa kết hợp với các bài toán liên quan như phát hiện đối tượng (object detection) hay theo dõi phương tiện (tracking).

Trong tương lai, đề tài có thể được mở rộng và phát triển theo một số hướng sau:

- Mở rộng và làm giàu tập dữ liệu với số lượng ảnh lớn hơn, đa dạng hơn về điều kiện chụp và bối cảnh giao thông.

- Kết hợp bài toán phân loại với phát hiện đối tượng, cho phép xử lý các ảnh hoặc video có nhiều phương tiện xuất hiện đồng thời.

- Áp dụng mô hình cho dữ liệu video thời gian thực trong các hệ thống camera giám sát giao thông.

- Nghiên cứu các mô hình hiện đại hơn hoặc các kiến trúc đa mô hình (multi-model) nhằm nâng cao độ chính xác và độ ổn định của hệ thống trong môi trường thực tế.

Nhìn chung, đề tài đã đạt được mục tiêu đề ra ban đầu, chứng minh tính khả thi và hiệu quả của việc ứng dụng các mô hình học sâu, đặc biệt là EfficientNet, trong bài toán phân loại phương tiện giao thông đường bộ tại Việt Nam. Các kết quả đạt được không chỉ có ý nghĩa về mặt học thuật mà còn có tiềm năng ứng dụng trong các hệ thống giao thông thông minh trong thực tế.

TÀI LIỆU THAM KHẢO

1. Joseph, S., Mewati, A., & Hapnes, T. (2021). Comparative Study of Convolutional Neural Networks-based Algorithm for Fine-grained Car Recognition. <https://doi.org/https://doi.org/10.5220/0010743800003113>
2. Bipul, N., Teerayut, H., & Jagannath, A. (2022). Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network. https://doi.org/https://doi.org/10.3390/s22103813
3. K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
4. M. Tan, Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” Proceedings of the 36th International Conference on Machine Learning (ICML), 2019
5. M. R. Shihab, N. I. Saim, M. U. Mojumdar, D. M. Raza, S. M. T. Siddiquee, S. R. H. Noori & N. R. Chakraborty, “Image dataset for classification of diseases in guava fruits and leaves,” Data in Brief, Vol. 59, 111378, 2025.
6. Richard, S. (2021). Computer vision: algorithms and applications. https://doi.org/https://doi.org/10.5860/choice.48-5140