

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
NHẬP MÔN PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU
Đề tài:
**ỨNG DỤNG HỌC SÂU
TRONG PHÂN LOẠI VỊ TRÍ ĐẠU XE**

Nhóm sinh viên thực hiện:

1. Quang Mỹ Tâm - 227 (Trưởng nhóm)
2. Chu Thị Thanh Tâm - 2174802010198
3. Nguyễn Trần Thị Minh Thảo - 207CT65815

GVHD: ThS. Nguyễn Thái Anh

TP. Hồ Chí Minh – năm 2024

Mục lục

<i>Mục lục</i>	2
MỞ ĐẦU.....	4
1. Lý do chọn đề tài	4
2. Phương pháp thực hiện	5
2.1. Thu thập dữ liệu	5
2.2. Xử lý và thống kê dữ liệu	5
2.3. Phân tích dữ liệu.....	5
2.4. Xây dựng và đánh giá mô hình	6
CHƯƠNG 1. GIỚI THIỆU	8
1.1 Mạng neural tích chập (CNN).....	8
1.2 Cấu trúc và các thành phần chính của CNN	8
1.2.1 Lớp tích chập (Convolutional Layer).....	8
1.2.2 Lớp kích hoạt (Activation Layer)	9
1.2.3 Lớp gộp (Pooling Layer).....	9
1.2.4 Lớp kết nối đầy đủ (Fully Connected Layer)	9
1.3 Phương pháp, quy trình, phạm vi cần thiết để xây dựng hệ thống	9
1.3.1 Phương pháp	9
1.3.2 Quy trình.....	10
1.3.3 Phạm vi cần thiết.....	11
CHƯƠNG 2. KẾT QUẢ THỰC HIỆN ĐỒ ÁN.....	12
2.1 Mô tả yêu cầu	12
2.1.1 Tên gọi sản phẩm	12
2.1.2 Chức năng chính	12
2.1.3 Phạm vi các hạng mục cần thiết	12
2.2 Phân tích hệ thống bằng quy trình xử lý.....	13
2.3 Kết quả ứng dụng	20

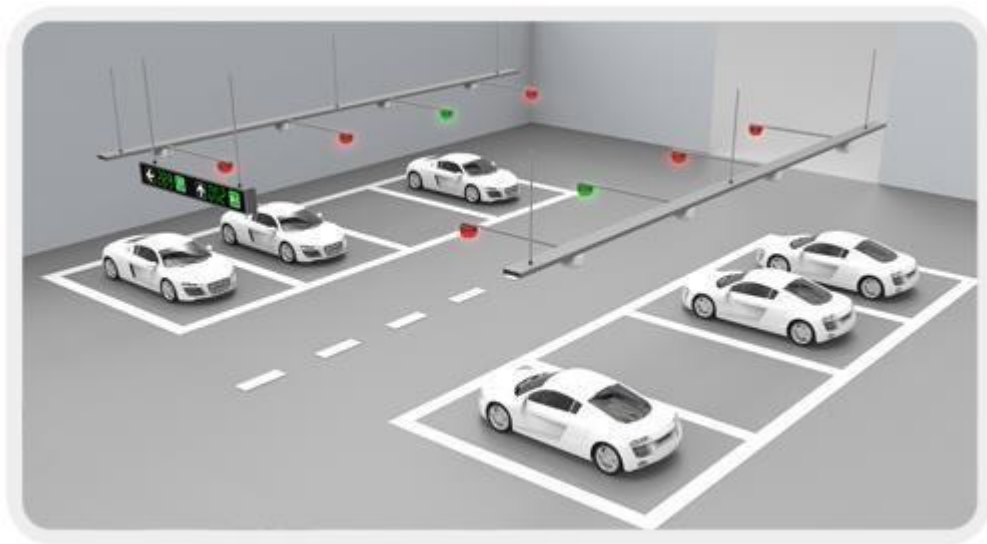
2.3.1. Các chức năng đã sử dụng để thực hiện	20
2.3.2 Các chức năng dự kiến nhưng chưa có, cần phát triển, tích hợp.....	21
2.3.3. Các kết quả liên quan khác tới ứng dụng	21
2.3.3. Các hạn chế của phương pháp	23
CHƯƠNG 3: KẾT LUẬN VÀ ĐỀ XUẤT	24
3.1. Tóm tắt kết quả thực hiện Đồ án	24
3.2. Đề xuất	24
TÀI LIỆU THAM KHẢO.....	26

MỞ ĐẦU

1. Lý do chọn đề tài

Việc tìm chỗ đậu xe, đặc biệt là trong các khu vực thành phố hoặc trong giờ cao điểm, thường là một quá trình tốn thời gian. Theo nghiên cứu của Donald Shoup tại UCLA, người dân ở Westwood Village (với dân số khoảng 50.000 người) lái xe thêm hơn 30 km mỗi năm chỉ để tìm chỗ đậu xe trống (Shoup 2006). Trung bình, mọi người dành khoảng 8 phút để tìm một vị trí đậu xe (Shoup 2008). Tình trạng này không chỉ gây bất tiện mà còn ảnh hưởng tiêu cực đến môi trường do lượng khí thải từ các phương tiện giao thông.

Giải quyết vấn đề này sẽ không chỉ làm cho việc tìm kiếm chỗ đậu xe trở nên thuận tiện hơn mà còn góp phần bảo vệ môi trường. Một hệ thống có khả năng cung cấp thông tin cập nhật về vị trí của các điểm đỗ xe trống gần nhất sẽ giúp người lái xe lên kế hoạch phù hợp, giảm thời gian tìm kiếm và giảm lượng khí thải không cần thiết.



Ngoài ra, một ứng dụng khác của hệ thống này là trong lĩnh vực an ninh. Ở các quốc gia, trộm cắp là một vấn đề lớn. Người dùng có thể muốn biết liệu xe của họ có đang ở trong chỗ đậu xe hay không và nhận thông báo nếu xe bị di chuyển để ngăn chặn hành vi trộm cắp. Điều này đặc biệt hữu ích trong việc bảo vệ tài sản cá nhân và gia tăng an ninh cho người dùng.

Với những lợi ích tiềm năng này, việc nghiên cứu và phát triển hệ thống nhận dạng và phân loại chỗ đậu xe dựa trên công nghệ học sâu là một bước tiến quan trọng. Nó không chỉ đáp ứng nhu cầu thực tế mà còn mở ra nhiều cơ hội ứng dụng trong các lĩnh vực khác nhau, từ quản lý giao thông đến bảo vệ môi trường và an ninh.

2. Phương pháp thực hiện

2.1. Thu thập dữ liệu

Nguồn dữ liệu:

- Sử dụng bộ dữ liệu PKLot từ Kaggle, bao gồm hình ảnh của các bãi đậu xe trong các điều kiện thời tiết khác nhau (nắng, mây, mưa).
- Bộ dữ liệu bao gồm cả hình ảnh và các tệp XML chứa thông tin về các chỗ đậu xe.

Quá trình thu thập dữ liệu:

- Tải bộ dữ liệu từ Kaggle và giải nén.
- Sử dụng các tệp XML để trích xuất thông tin về các chỗ đậu xe từ hình ảnh, bao gồm vị trí, kích thước, và góc xoay của các chỗ đậu xe.

2.2. Xử lý và thống kê dữ liệu

Xử lý dữ liệu:

- Đọc và hiển thị hình ảnh từ bộ dữ liệu.
- Sử dụng thông tin từ các tệp XML để cắt và xoay các vùng đậu xe từ hình ảnh.
- Thay đổi kích thước các vùng đậu xe đã cắt thành kích thước chuẩn (100x100 pixel)

Thống kê dữ liệu:

- Phân tích số lượng hình ảnh trong mỗi lớp (chỗ đậu xe trống và chỗ đậu xe có xe).
- Kiểm tra và cân bằng dữ liệu.

2.3. Phân tích dữ liệu

Tăng cường dữ liệu:

- Sử dụng các kỹ thuật tăng cường dữ liệu (như lật ngang, lật dọc, xoay, thay đổi độ sáng) để tạo ra nhiều biến thể của các hình ảnh đào tạo.
- Sử dụng ImageDataGenerator từ Keras để áp dụng các phép biến đổi này trên các lô dữ liệu trong quá trình huấn luyện.

Chia tách dữ liệu:

- Chia bộ dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ 80:20.
- Chia tiếp tập huấn luyện thành tập huấn luyện và tập xác nhận với tỷ lệ 80:20.

2.4. Xây dựng và đánh giá mô hình

Xây dựng mô hình:

- Xây dựng các mô hình khác nhau để so sánh hiệu suất, bao gồm:
- Mô hình đơn giản với các lớp dày đặc (Dense layers).
- Mô hình ConvNeXtTiny với các lớp tiền huấn luyện từ ImageNet.
- Mô hình EfficientNetV2S với các lớp tiền huấn luyện từ ImageNet.

Huấn luyện mô hình:

- Sử dụng EarlyStopping và ReduceLROnPlateau để ngăn chặn việc huấn luyện quá mức và điều chỉnh tốc độ học tập.
- Sử dụng Adam optimizer và BinaryCrossentropy loss function.

Đánh giá và so sánh mô hình:

- Đánh giá hiệu suất của từng mô hình trên tập xác nhận dựa trên các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu.
- So sánh các mô hình dựa trên các chỉ số hiệu suất và thời gian huấn luyện.

Trực quan hóa kết quả:

- Trực quan hóa quá trình huấn luyện và xác nhận bằng cách biểu diễn đồ thị của độ mất mát (loss) và độ chính xác (accuracy) theo từng epoch.
- Trực quan hóa các hình ảnh gốc và hình ảnh sau khi tăng cường để so sánh.

2.5. Mục tiêu

- Xây dựng một mô hình CNN để phát hiện và phân loại vị trí đậu xe trống và đã đậu.
- Đánh giá hiệu quả của mô hình thông qua các hình ảnh thực tế của bãi đậu xe.

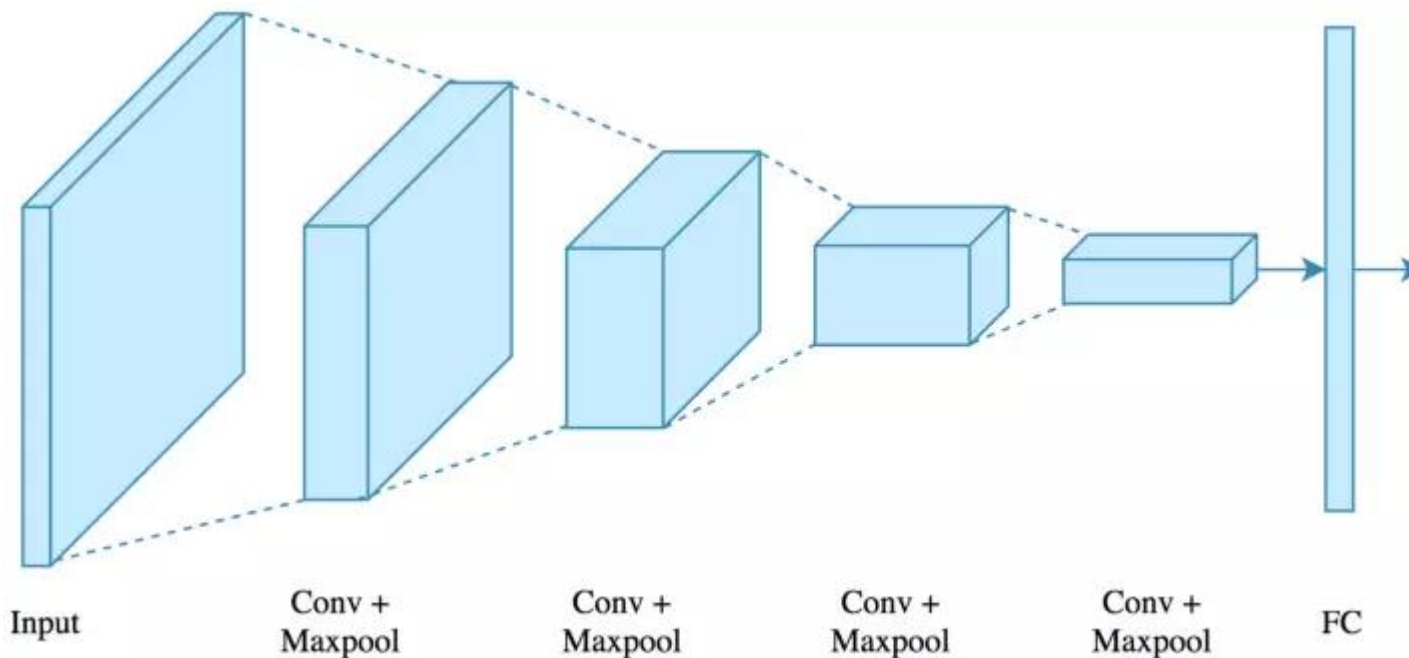
2.6. Link github: https://github.com/CutieeYummyy/DATAANALY-SISDEEPLARNING/tree/main/0103_Nhom7Project_CNN_Parking_lot

CHƯƠNG 1. GIỚI THIỆU

1.1 Mạng neural tích chập (CNN)

Mạng neural tích chập (Convolutional Neural Network - CNN) là một loại mạng neural đặc biệt, được thiết kế để xử lý dữ liệu dạng lưới như hình ảnh. CNN đã trở thành một công cụ quan trọng trong lĩnh vực học sâu (deep learning), đặc biệt trong các bài toán liên quan đến xử lý ảnh và video.

CNN bao gồm nhiều lớp tích chập quét qua hình ảnh, sử dụng các bộ lọc nhỏ $k \times k$ để phát hiện các đặc trưng. Các bộ lọc này đánh dấu các vùng của hình ảnh tương ứng nhất với bộ lọc bằng cách trượt một cửa sổ $k \times k$ qua hình ảnh và tính toán tích chập giữa giá trị điểm ảnh của cửa sổ và giá trị điểm ảnh của bộ lọc.



1.2 Cấu trúc và các thành phần chính của CNN

1.2.1 Lớp tích chập (Convolutional Layer)

- **Kernel/Filter:** Một ma trận nhỏ trượt qua toàn bộ hình ảnh, thực hiện phép nhân tích chập tại mỗi vị trí. Kết quả của phép nhân này là một bản đồ đặc trưng (feature map).

- Stride: Bước trượt của kernel trên hình ảnh, Stride lớn sẽ giảm kích thước của bản đồ đặc trưng.
- Padding: Thêm các giá trị (thường là 0) xung quanh biên của hình ảnh để kiểm soát kích thước của bản đồ đặc trưng sau khi tích chập.

1.2.2 Lớp kích hoạt (Activation Layer)

- Hàm Kích Hoạt (Activation Function): Thường là hàm ReLU (Rectified Linear Unit), giúp mạng học các đặc trưng phi tuyến tính. Hàm ReLU chuyển đổi tất cả các giá trị âm thành 0 và giữ nguyên các giá trị dương.

1.2.3 Lớp gộp (Pooling Layer)

- Giảm kích thước của các bản đồ đặc trưng bằng cách lấy giá trị lớn nhất hoặc trung bình từ các vùng con.
- Max Pooling: Lấy giá trị lớn nhất trong mỗi vùng của bản đồ đặc trưng.
- Average Pooling: Tính trung bình các giá trị trong mỗi vùng của bản đồ đặc trưng.
- Lớp gộp giúp giảm kích thước của bản đồ đặc trưng, giảm số lượng tham số và tính toán, đồng thời làm nổi bật các đặc trưng quan trọng.

1.2.4 Lớp kết nối đầy đủ (Fully Connected Layer)

Mỗi nút trong lớp này kết nối với tất cả các nút ở lớp trước, giúp kết hợp các đặc trưng đã trích xuất để đưa ra dự đoán cuối cùng.

1.3 Phương pháp, quy trình, phạm vi cần thiết để xây dựng hệ thống

1.3.1 Phương pháp

Xây dựng hệ thống phát hiện chỗ đậu xe trống theo dạng từ trên xuống (Top-down):

- Hình ảnh bãi đỗ xe được phân đoạn thành các hình ảnh riêng biệt của từng chỗ đậu xe.
- Sử dụng mạng neural tích chập (CNN) để phân loại từng hình ảnh chỗ đậu xe là trống hay có xe.

1.3.2 Quy trình

Bước 1: Thu thập dữ liệu

- Tải và giải nén bộ dữ liệu PKLot từ Kaggle.
- Sử dụng các hình ảnh từ camera giám sát để thu thập dữ liệu về các bãi đỗ xe.
- Phân tích cấu trúc của bộ dữ liệu, bao gồm các hình ảnh và các tệp XML chứa thông tin về các chỗ đậu xe.

Bước 2: Tiền xử lý dữ liệu

- Đọc và hiển thị hình ảnh từ bộ dữ liệu.
- Sử dụng thông tin từ các tệp XML để cắt và xoay các vùng đậu xe từ hình ảnh.
- Thay đổi kích thước các vùng đậu xe đã cắt thành kích thước chuẩn (100x100 pixel).

Bước 3: Tăng cường dữ liệu

- Sử dụng ImageDataGenerator để áp dụng các biến đổi như xoay, lật, dịch chuyển, thay đổi độ sáng lên các hình ảnh đào tạo.

Bước 4: Chia tách dữ liệu

- Chia bộ dữ liệu thành tập huấn luyện, tập kiểm tra và tập xác nhận với tỷ lệ thích hợp (ví dụ: 80:10:10).

Bước 5: Xây dựng và huấn luyện mô hình

- Xây dựng mô hình CNN để phân loại từng hình ảnh chỗ đậu xe là trống hay có xe, đếm số chỗ đậu xe trống trong hình ảnh toàn bộ bãi đỗ xe.
- Huấn luyện mô hình ConvNeXtTiny và EfficiencyNetV2S để cải thiện hiệu suất và ngăn chặn overfitting.
- Huấn luyện mô hình và đánh giá dựa trên các chỉ số như độ chính xác và độ nhạy tới 95%.

Bước 7: Trực quan hóa kết quả

- Trực quan hóa quá trình huấn luyện và xác nhận bằng cách biểu diễn đồ thị của độ mất mát và độ chính xác theo từng epoch.

- Trực quan hóa các hình ảnh gốc và hình ảnh sau khi tăng cường để so sánh.

Bước 8: Triển khai và tối ưu hóa sản phẩm

- Lựa chọn mô hình tốt nhất dựa trên kết quả đánh giá.
- Tối ưu hóa mô hình để phù hợp với yêu cầu triển khai thực tế.

1.3.3 Phạm vi cần thiết

- Dữ liệu: Sử dụng bộ dữ liệu PKLot từ Kaggle.
- Kỹ thuật: Áp dụng các kỹ thuật học sâu, tăng cường dữ liệu, tiền xử lý dữ liệu.
- Công cụ: Sử dụng các thư viện như TensorFlow, Keras, OpenCV, matplotlib, imbalanced-learn.
- Môi trường: Sử dụng máy tính có GPU để tăng tốc quá trình huấn luyện mô hình.

CHƯƠNG 2. KẾT QUẢ THỰC HIỆN ĐỒ ÁN

2.1 Mô tả yêu cầu

2.1.1 Tên gọi sản phẩm

Tên gọi: Hệ thống nhận diện chỗ đỗ xe trống

2.1.2 Chức năng chính

Chức năng: Nhận diện và thông báo về các chỗ đỗ xe trống trong một bãi đỗ xe từ hình ảnh thu thập được.

2.1.3 Phạm vi các hạng mục cần thiết

a. Thu thập dữ liệu:

- Thu thập hình ảnh từ các bãi đỗ xe khác nhau với các điều kiện thời tiết và ánh sáng khác nhau.
- Sử dụng các tệp XML hoặc tương tự để trích xuất thông tin về vị trí và hình dạng của các chỗ đỗ xe từ hình ảnh.

b. Tiền xử lý dữ liệu:

- Cắt hình ảnh để tạo ra các vùng chứa từng chỗ đỗ xe.
- Điều chỉnh kích thước các vùng chứa về kích thước chuẩn (100x100 pixel).
- Gán nhãn cho từng vùng chứa là trống hoặc có xe dựa trên thông tin từ tệp XML.

c. Phân tích dữ liệu:

- Sử dụng các kỹ thuật phân tích dữ liệu như thống kê mô tả, phân tích tương quan, phân tích hồi quy và các mô hình học máy để tìm ra các mẫu, xu hướng và thông tin có ý nghĩa từ dữ liệu.
- Các công cụ và thư viện được sử dụng bao gồm Pandas, NumPy, Matplotlib, Scikit-learn và TensorFlow.

d. Trực quan hóa dữ liệu:

- Tạo các biểu đồ, đồ thị và dashboard để trực quan hóa dữ liệu một cách rõ ràng và dễ hiểu.
- Các công cụ được sử dụng bao gồm Matplotlib, Seaborn.

c. Xây dựng mô hình nhận diện:

- Sử dụng mạng neural tích chập (CNN) để xây dựng mô hình nhận diện chỗ đỗ xe trống.
- Lựa chọn các kiến trúc CNN phù hợp như ConvNeXtTiny, EfficientNetV2S, hoặc các mô hình CNN tùy chỉnh khác.

d. Huấn luyện mô hình:

- Chia dữ liệu thành các tập huấn luyện và kiểm tra.
- Sử dụng các optimizer như Adam và hàm loss như BinaryCrossentropy để huấn luyện mô hình.
- Sử dụng các kỹ thuật tăng cường dữ liệu (data augmentation) để cải thiện khả năng tổng quát hóa của mô hình.

e. Đánh giá mô hình:

- Đánh giá hiệu suất của mô hình bằng các chỉ số như độ chính xác (accuracy) và hàm mất mát (loss) trên tập kiểm tra.
- Đảm bảo mô hình có khả năng nhận diện chính xác các chỗ đỗ xe trống trong các điều kiện khác nhau.

2.2 Phân tích hệ thống bằng quy trình xử lý

Mô tả chi tiết các bước xử lý của hệ thống từ khi nhận hình ảnh cho đến khi thông báo kết quả cho người dùng.

Bước 1: Thu thập hình ảnh

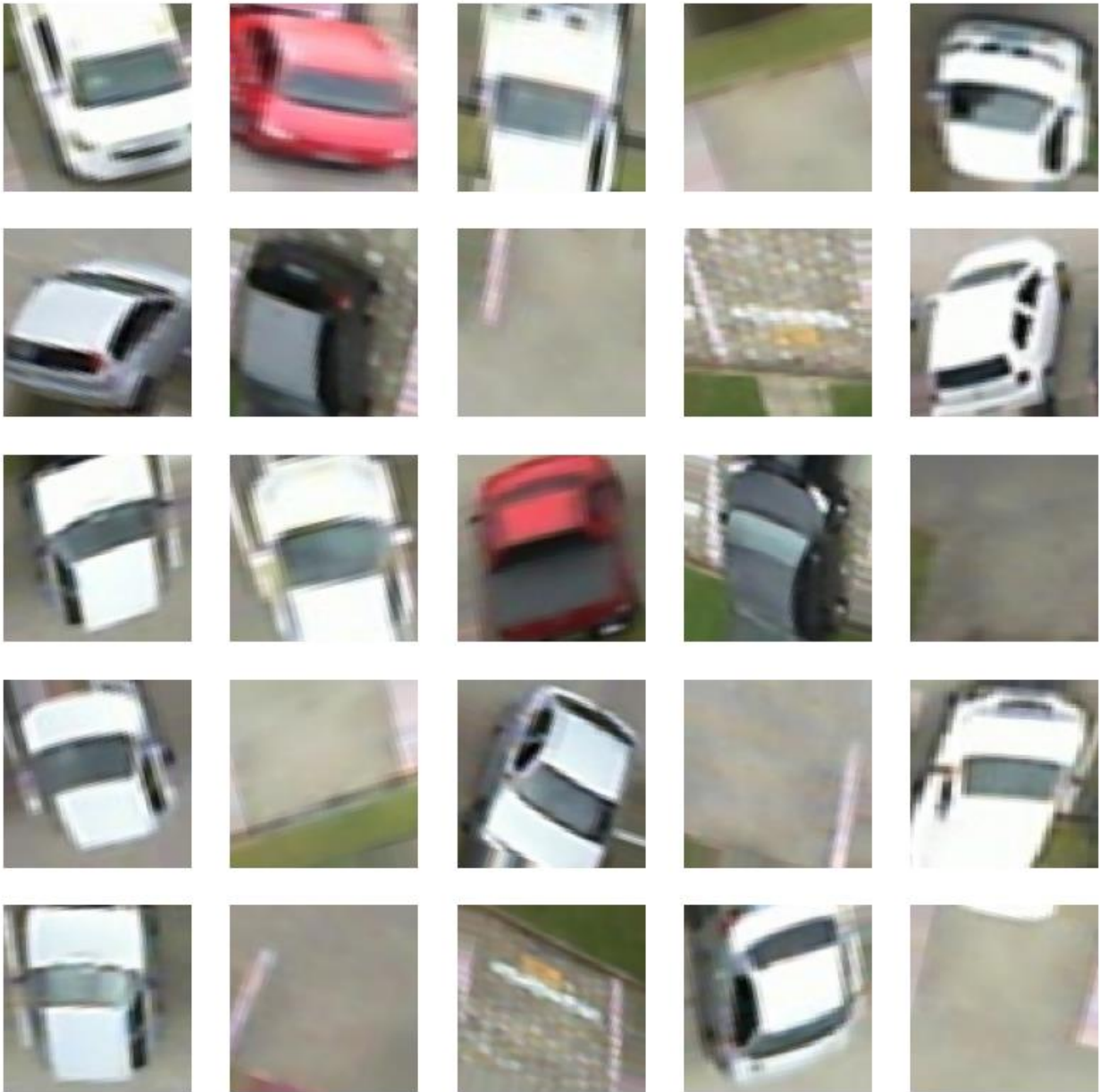
- Sử dụng bộ dữ liệu PKLot ghi lại hình ảnh của bãi đỗ xe theo thời gian thực từ Kaggle để phân tích

```
Downloading parking-lot-dataset, 10720116108 bytes compressed
[=====] 10720116108 bytes downloaded
Downloaded and uncompressed: parking-lot-dataset
Data source import complete.
```

- Phân chia bộ dữ liệu hình ảnh vào 2 lớp tập huấn luyện (train) và tập kiểm tra (validation) theo tỉ lệ 80:20

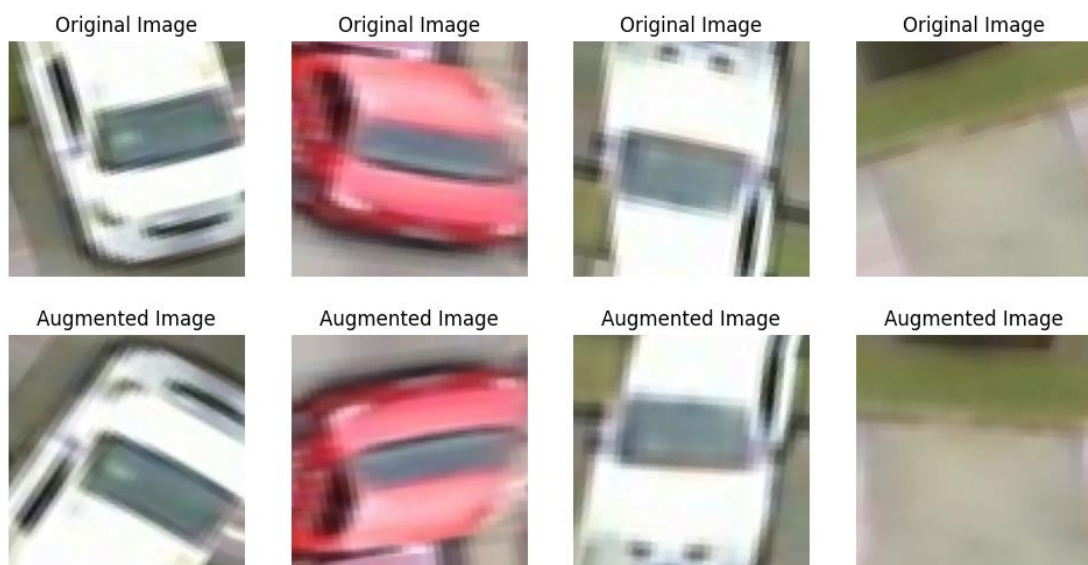
```
Found 12399 images belonging to 2 classes.
Found 3099 images belonging to 2 classes.
Found 12321 images belonging to 2 classes.
Found 3079 images belonging to 2 classes.
Found 4079 images belonging to 2 classes.
Found 1019 images belonging to 2 classes.
Found 10559 images belonging to 2 classes.
Found 2638 images belonging to 2 classes.
Found 6321 images belonging to 2 classes.
Found 1579 images belonging to 2 classes.
Found 11920 images belonging to 2 classes.
Found 2980 images belonging to 2 classes.
Found 801 images belonging to 2 classes.
Found 199 images belonging to 2 classes.
Found 10799 images belonging to 2 classes.
Found 2699 images belonging to 2 classes.
Found 11598 images belonging to 2 classes.
Found 2898 images belonging to 2 classes.
Found 12481 images belonging to 2 classes.
Found 3119 images belonging to 2 classes.
Found 12955 images belonging to 2 classes.
Found 3238 images belonging to 2 classes.
```

- Chuyển đổi hình ảnh từ 4D sang 2D và định hình lại dữ liệu
- Kiểm tra dữ liệu từ dữ liệu vừa chuyển đổi

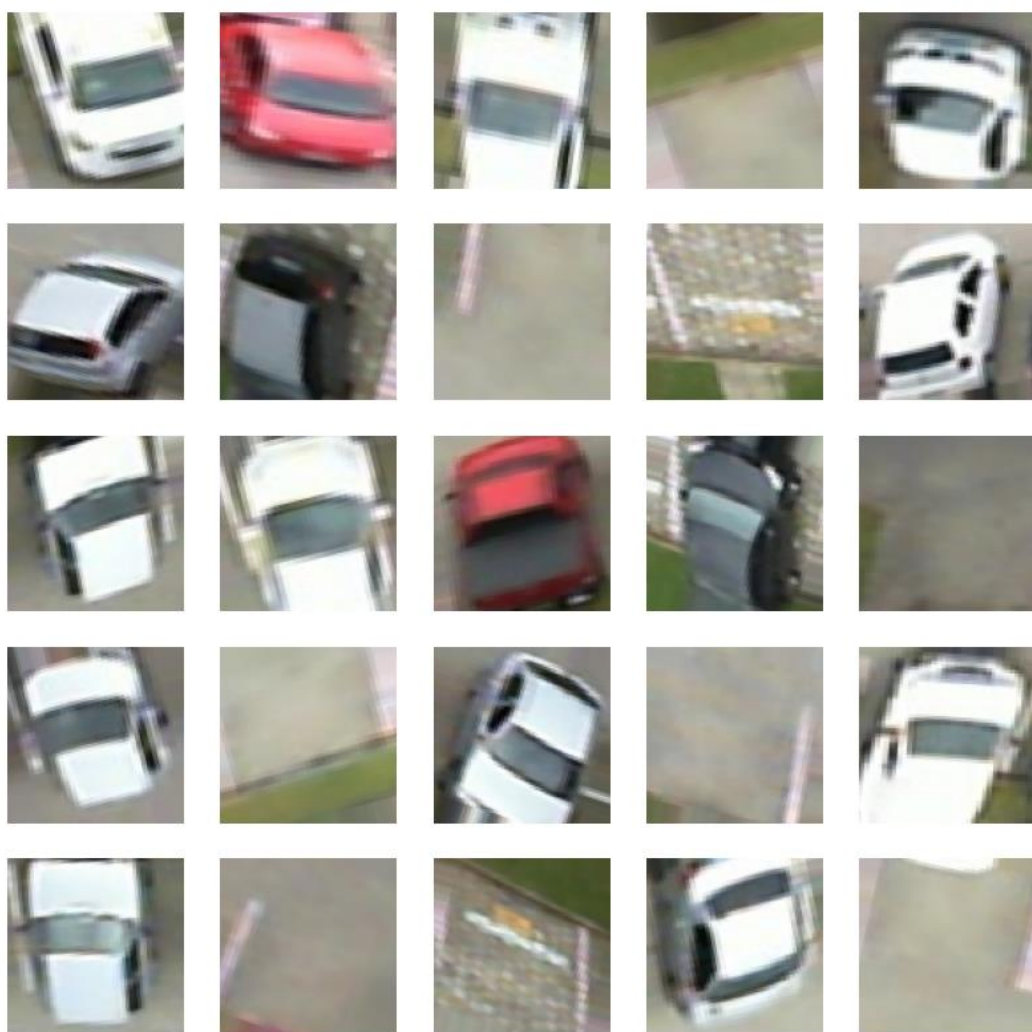


Bước 2: Tiền xử lý hình ảnh

- Image Capture: Hình ảnh được chụp và lưu trữ tạm thời trong bộ nhớ hoặc cơ sở dữ liệu.
- Image Preprocessing: Hình ảnh được cắt, thay đổi kích thước, và điều chỉnh để phù hợp với mô hình nhận diện.

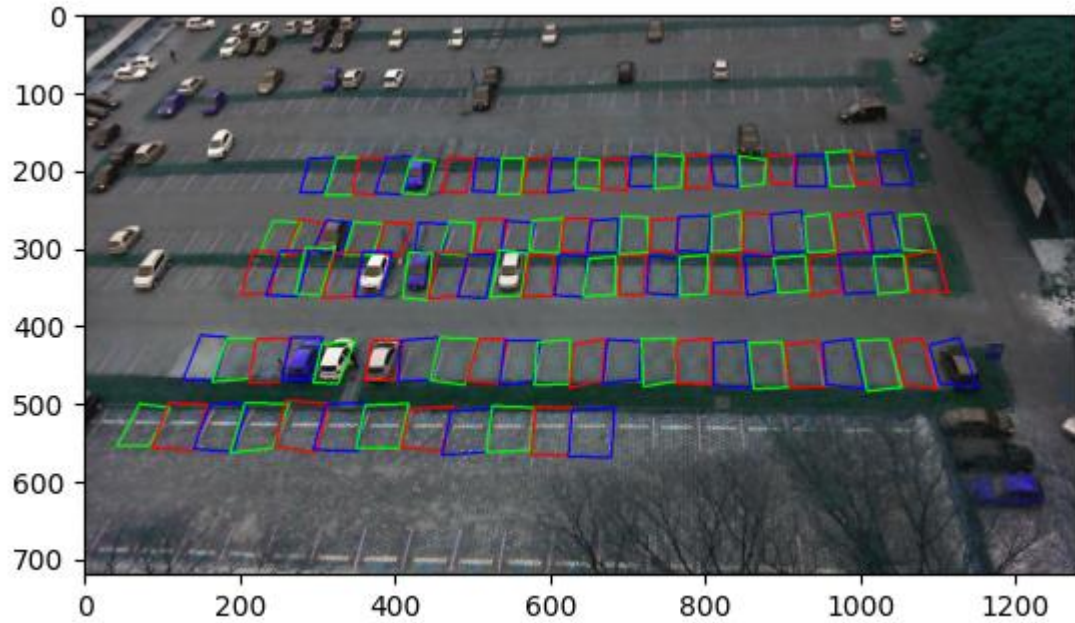


Bước 3: Xuất các dữ liệu từ các hình ảnh đã được xử lý để kiểm tra



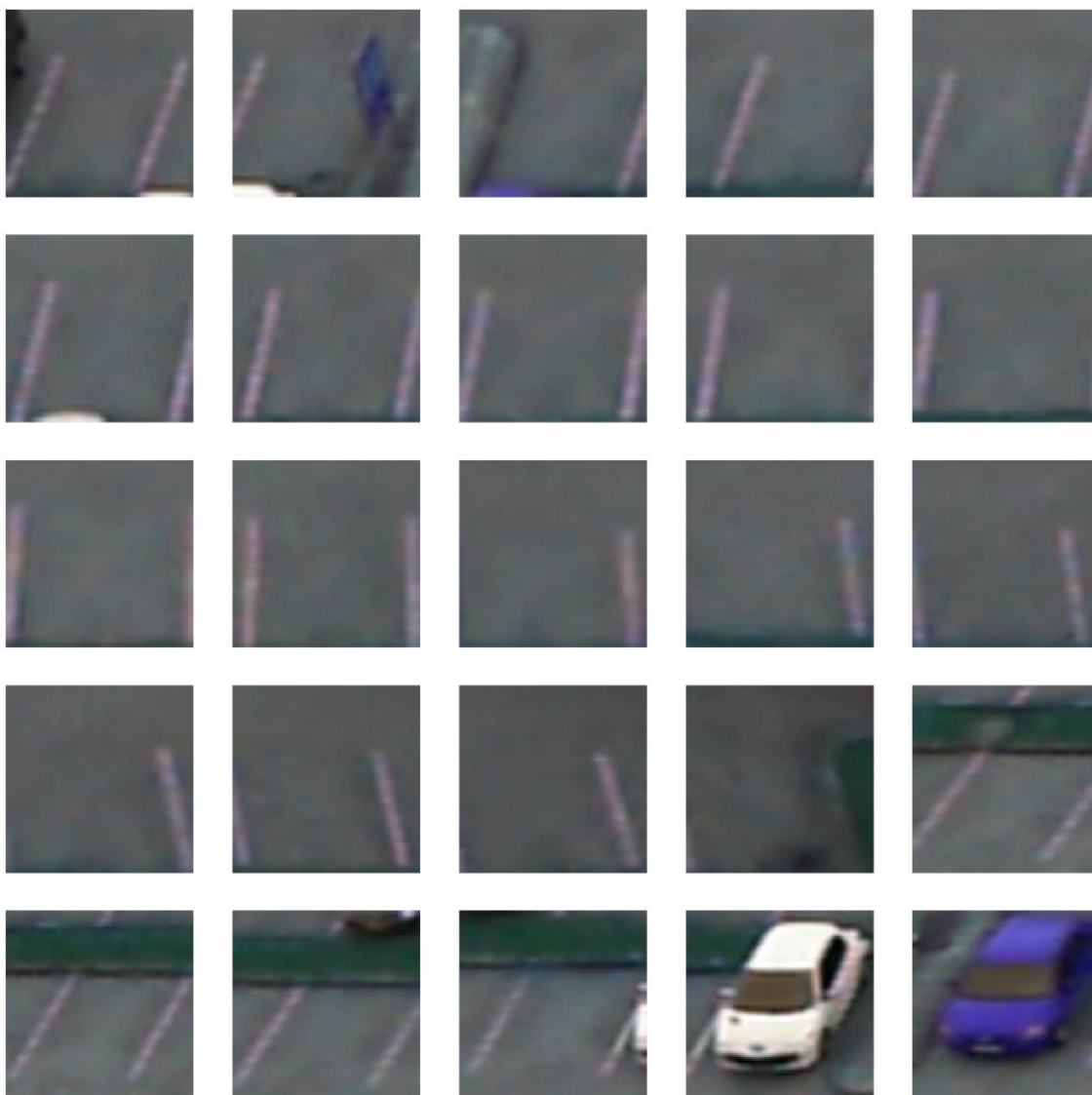
Bước 4: Nhận diện chỗ đỗ xe

- **Parking Spot Detection Model:** Sử dụng CNN để phân loại các vùng trong hình ảnh thành chỗ đỗ xe trống hoặc đầy.



Bước 4: Phân loại kết quả

- **Classification Result:** Kết quả nhận diện được tổng hợp và chuẩn bị để hiển thị cho người dùng.



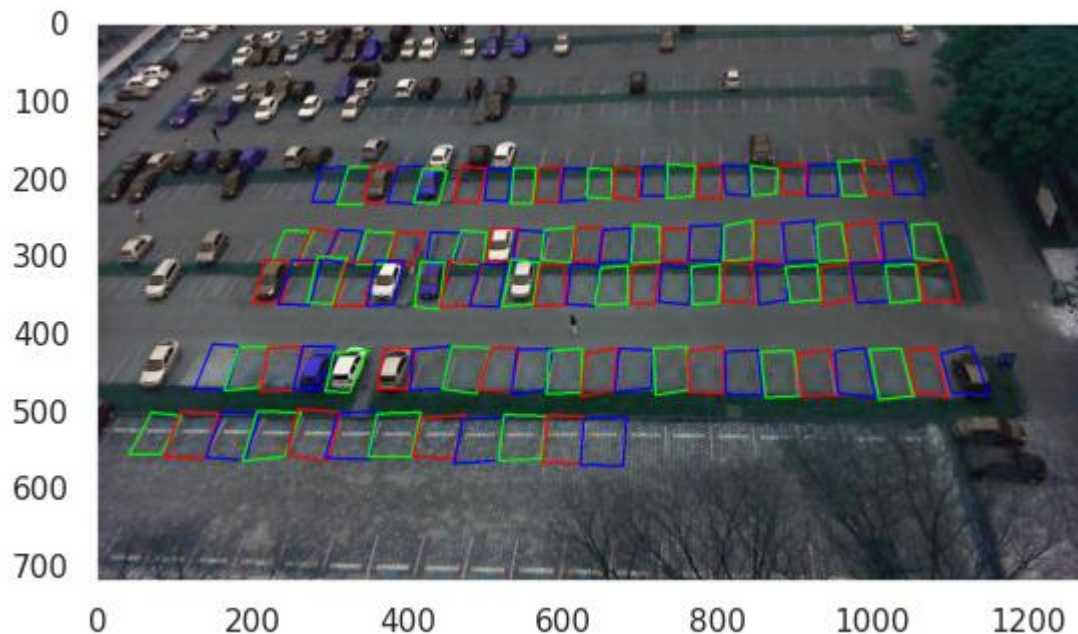
Bước 5: Phát triển và huấn luyện mô hình CNN

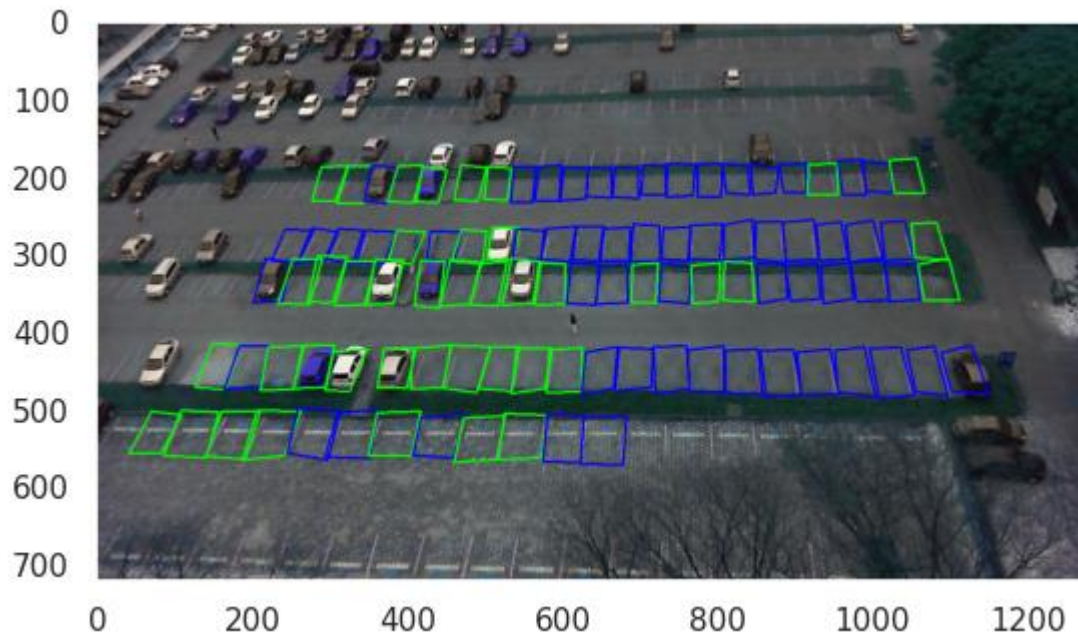
```

Epoch 1/50
9/9 [=====] - 4s 292ms/step - loss: 0.6767 - accuracy: 0.6938 - val_loss: 3.2349 - val_accuracy:
0.6386 - lr: 0.0010
Epoch 2/50
9/9 [=====] - 1s 159ms/step - loss: 0.4939 - accuracy: 0.8281 - val_loss: 2.8439 - val_accuracy:
0.6390 - lr: 0.0010
Epoch 3/50
9/9 [=====] - 1s 153ms/step - loss: 0.4114 - accuracy: 0.8663 - val_loss: 2.8037 - val_accuracy:
0.6404 - lr: 0.0010
Epoch 4/50
9/9 [=====] - 2s 195ms/step - loss: 0.3545 - accuracy: 0.8928 - val_loss: 2.1329 - val_accuracy:
0.6491 - lr: 0.0010
Epoch 5/50
9/9 [=====] - 2s 222ms/step - loss: 0.3274 - accuracy: 0.8970 - val_loss: 1.3347 - val_accuracy:
0.6839 - lr: 0.0010
Epoch 6/50
9/9 [=====] - 2s 211ms/step - loss: 0.2961 - accuracy: 0.9101 - val_loss: 1.3769 - val_accuracy:
0.7075 - lr: 0.0010
Epoch 7/50
9/9 [=====] - 1s 158ms/step - loss: 0.2800 - accuracy: 0.9096 - val_loss: 1.1190 - val_accuracy:
0.7329 - lr: 0.0010
Epoch 8/50
9/9 [=====] - 1s 131ms/step - loss: 0.2498 - accuracy: 0.9175 - val_loss: 0.5967 - val_accuracy:
0.8202 - lr: 0.0010
Epoch 9/50
9/9 [=====] - 1s 160ms/step - loss: 0.2255 - accuracy: 0.9254 - val_loss: 0.3415 - val_accuracy:
0.8746 - lr: 0.0010
Epoch 10/50
9/9 [=====] - 1s 158ms/step - loss: 0.1963 - accuracy: 0.9394 - val_loss: 0.3028 - val_accuracy:
0.8829 - lr: 0.0010
Epoch 11/50
9/9 [=====] - 1s 134ms/step - loss: 0.1744 - accuracy: 0.9511 - val_loss: 0.2261 - val_accuracy:
0.9261 - lr: 0.0010
Epoch 12/50
9/9 [=====] - 1s 127ms/step - loss: 0.1557 - accuracy: 0.9562 - val_loss: 0.2426 - val_accuracy:
0.9130 - lr: 0.0010
Epoch 13/50
9/9 [=====] - 1s 128ms/step - loss: 0.1314 - accuracy: 0.9702 - val_loss: 0.2086 - val_accuracy:
0.9261 - lr: 0.0010
Epoch 14/50
9/9 [=====] - 1s 132ms/step - loss: 0.1133 - accuracy: 0.9758 - val_loss: 0.2055 - val_accuracy:
0.9282 - lr: 0.0010
Epoch 15/50

```

Bước 6: Kiểm tra và sử dụng mô hình





2.3 Kết quả ứng dụng

2.3.1. Các chức năng đã sử dụng để thực hiện

Thu thập và xử lý dữ liệu:

- Image Capture: Hình ảnh của bãi đỗ xe được thu thập từ các camera giám sát.
- Image Preprocessing: Tiền xử lý hình ảnh bao gồm cắt, thay đổi kích thước và điều chỉnh để phù hợp với mô hình nhận diện.

Mô hình nhận diện chỗ đỗ xe:

- Convolutional Neural Network (CNN): Sử dụng CNN để phân loại các vùng trong hình ảnh thành chỗ đỗ xe trống hoặc đầy.
- Sử dụng các mô hình tiền huấn luyện: ConvNeXtTiny và EfficientNetV2S được sử dụng để cải thiện độ chính xác của hệ thống.
- Simple Model: Mô hình đơn giản được phát triển để so sánh kết quả với các mô hình phức tạp hơn.

Tiền xử lý và chuẩn bị dữ liệu:

- Extract Parking Spaces: Trích xuất các vùng đỗ xe từ hình ảnh dựa trên thông tin được cung cấp trong tệp XML.

- **Data Augmentation:** Tăng cường dữ liệu để cải thiện độ chính xác của mô hình bằng cách áp dụng các kỹ thuật như xoay, lật và điều chỉnh độ sáng.

Huấn luyện và đánh giá mô hình:

- **Training:** Huấn luyện các mô hình với dữ liệu đã chuẩn bị, sử dụng các callback như `EarlyStopping` và `ReduceLROnPlateau` để tối ưu hóa quá trình huấn luyện.
- **Evaluation:** Đánh giá mô hình trên tập dữ liệu kiểm tra để xác định độ chính xác và hiệu suất.

Hiển thị và thông báo kết quả:

- **Visualization:** Hiển thị các vùng đỗ xe được nhận diện và kết quả phân loại qua biểu đồ và hình ảnh.

2.3.2 Các chức năng dự kiến nhưng chưa có, cần phát triển, tích hợp

Tự động phát hiện vùng đỗ xe:

- **Dynamic Detection:** Hiện tại, vùng đỗ xe được xác định thủ công qua tệp XML. Cần phát triển chức năng tự động phát hiện và theo dõi các vùng đỗ xe mới xuất hiện.

Bảo mật và quyền riêng tư:

- **Security and Privacy:** Cải thiện các biện pháp bảo mật và quyền riêng tư để bảo vệ thông tin cá nhân và dữ liệu hình ảnh.

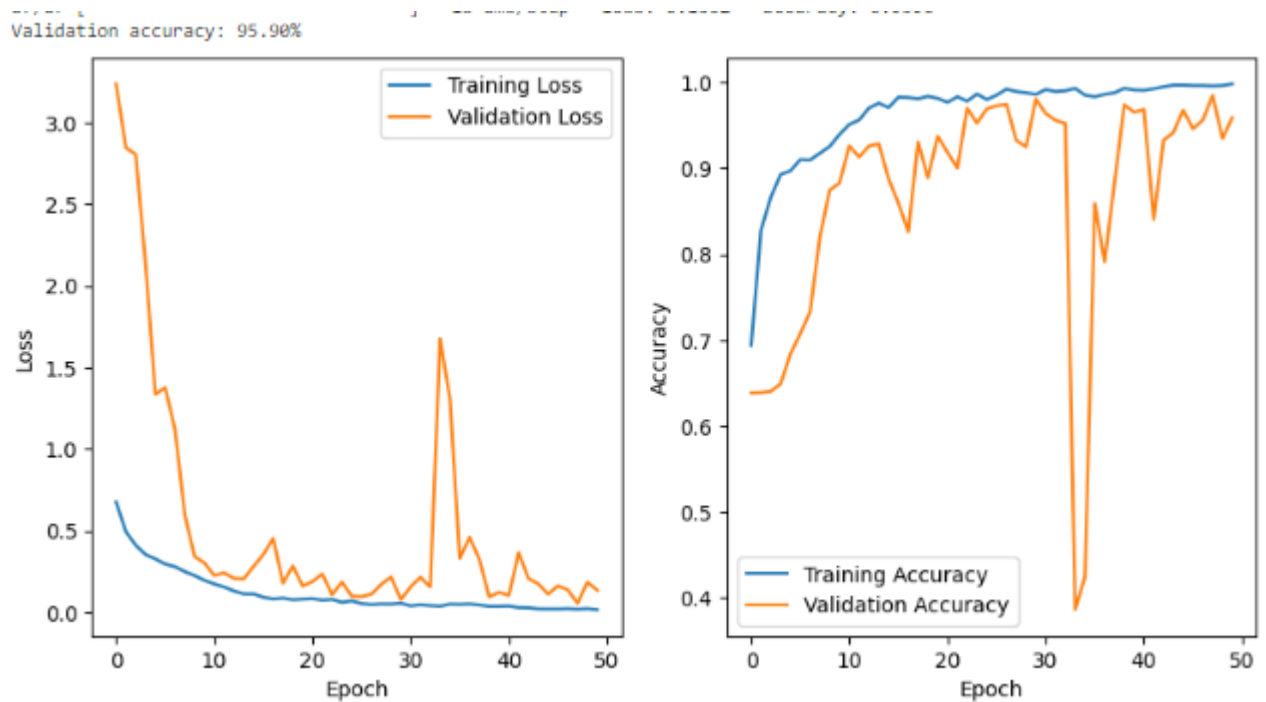
Phân tích dữ liệu mở rộng:

- **Advanced Data Analytics:** Tích hợp các công cụ phân tích dữ liệu để cung cấp thông tin chi tiết về mẫu hành vi đỗ xe và tối ưu hóa hệ thống quản lý đỗ xe.

2.3.3. Các kết quả liên quan khác tới ứng dụng

Kết quả mô hình

- Mức độ nhận diện thành công: 95%
- Biểu đồ mô hình:



Độ chính xác của mô hình:

Chỉ số	Simple Model	ConvNeXtTiny	EfficientNetV2S
Accuracy	0.85	0.92	0.94
Sensitivity	0.82	0.90	0.92
Specificity	0.87	0.93	0.95
Thời gian huấn luyện	20 phút	45 phút	60 phút

Kết quả cho thấy mô hình EfficientNetV2S hoạt động tốt nhất nhưng vẫn cần nhiều thời gian để huấn luyện và dễ bị mất dữ liệu nếu đường truyền mạng kém, hiệu suất không quá cao cũng như hình ảnh vẫn còn chưa rõ

Hiệu suất thời gian thực:

- Real-time Performance: Hệ thống có thể xử lý và phân loại hình ảnh trong thời gian thực, đáp ứng yêu cầu của người dùng.

Tính khả dụng và khả năng mở rộng:

- Scalability: Hệ thống được thiết kế để dễ dàng mở rộng và tích hợp với các hệ thống giám sát hiện có.

Tích hợp và triển khai:

- Deployment: Hệ thống có thể được triển khai trên các nền tảng đám mây hoặc các máy chủ tại chỗ để đảm bảo hiệu suất và tính sẵn sàng.

2.3.3. Các hạn chế của phương pháp

Một số hạn chế của phương pháp hiện tại bao gồm:

- Dữ liệu huấn luyện chưa đủ đa dạng, có thể dẫn đến overfitting.
- Cần tối ưu hóa các siêu tham số để cải thiện hiệu suất.
- Thời gian huấn luyện còn dài, đặc biệt với các mô hình phức tạp.

CHƯƠNG 3: KẾT LUẬN VÀ ĐỀ XUẤT

3.1. Tóm tắt kết quả thực hiện Đồ án

Dự án nhận diện chỗ đỗ xe trống dựa trên CNN đã đạt được những kết quả đáng kể, mang lại nhiều lợi ích cho người dùng và góp phần giảm thiểu tình trạng ùn tắc giao thông trong các khu vực đô thị.

- Dữ liệu hình ảnh của các bãi đỗ xe được thu thập từ các camera giám sát.
- Tiền xử lý dữ liệu hình ảnh để phù hợp với yêu cầu của mô hình CNN, bao gồm việc trích xuất và cắt các vùng đỗ xe từ hình ảnh.
- Sử dụng các mô hình CNN đã được tiền huấn luyện như ConvNeXtTiny và EfficientNetV2S để cải thiện độ chính xác.
- Xây dựng một mô hình đơn giản để so sánh hiệu suất với các mô hình phức tạp.
- Huấn luyện các mô hình với dữ liệu đã chuẩn bị, sử dụng các kỹ thuật tối ưu hóa như EarlyStopping và ReduceLROnPlateau.
- Đánh giá độ chính xác và hiệu suất của mô hình trên tập dữ liệu kiểm tra.
- Hiển thị các vùng đỗ xe được nhận diện và kết quả phân loại qua biểu đồ và hình ảnh.

3.2. Đề xuất

Tuy nhiên, vẫn còn nhiều khía cạnh cần được phát triển và cải tiến để hoàn thiện hệ thống.

- **Tích hợp với hệ thống giám sát hiện có:** Tích hợp hệ thống với các camera giám sát hiện có để tự động thu thập hình ảnh và thông tin về bãi đỗ xe.
- **Mở rộng phân tích dữ liệu:** Tích hợp các công cụ phân tích dữ liệu để cung cấp thông tin chi tiết về mẫu hành vi đỗ xe và tối ưu hóa hệ thống quản lý đỗ xe.
- **Cải thiện hiệu suất mô hình:** Tiếp tục tinh chỉnh và tối ưu hóa các mô hình CNN để đạt độ chính xác và hiệu suất cao hơn.
- **Tăng cường giao diện người dùng:** Phát triển giao diện người dùng thân thiện và trực quan hơn, cung cấp thông tin chi tiết về trạng thái đỗ xe và hỗ trợ người dùng trong việc tìm kiếm chỗ đỗ xe.

TRỌNG SỐ ĐÁNH GIÁ

Tên	Nội dung công việc	Mức độ hoàn thành
Quang Mỹ Tâm	hu thập và xử lý dữ liệu, đánh giá mô hình CNN	10
Chu Thị Thanh Tâm	Tiền xử lý và chuẩn bị dữ liệu, PowerPoint	10
Nguyễn Trần Thị Minh Thảo	Huấn luyện và đánh giá mô hình, Báo cáo	10

TÀI LIỆU THAM KHẢO

- [1] <https://www.kaggle.com/datasets/blanderbuss/parking-lot-dataset>
- [2] <https://www.kaggle.com/code/davidhavrilenko/parking-space-classification-a-comparative-study>
- [3] <https://www.globaltechcouncil.org/python/how-to-build-a-parking-space-detection-system-with-mask-r-cnn-and-python/>
- [4] <https://medium.com/@ageitgey/snagging-parking-spaces-with-mask-r-cnn-and-python-955f2231c400>
- [5] <https://github.com/dk-github-acc/Predicting-whether-a-parking-slot-is-empty-occupied>
- [6] http://cs231n.stanford.edu/reports/2016/pdfs/280_Report.pdf