

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG



MÔN HỌC: THỰC TẬP HỌC MÁY VÀ TRÍ TUỆ NHÂN TẠO
BÁO CÁO CUỐI KỲ

ĐỀ TÀI

**PHÂN ĐOẠN ẢNH TRÊN TẬP DỮ LIỆU ẢNH PHỔ TÍN
HIỆU SỬ DỤNG MÔ HÌNH HỌC SÂU**

NGÀNH HỆ THỐNG NHÚNG VÀ IOT

Sinh viên thực hiện:

1. **Ngô Trần Quốc Bảo** 22139004
2. **Võ Xuân Lộc** 22139040

Giảng viên hướng dẫn: TS. Huỳnh Thế Thiện

TP. HỒ CHÍ MINH - 05/2025

Mục lục

DANH SÁCH HÌNH ẢNH	4
DANH SÁCH BẢNG	5
CHƯƠNG 1 TỔNG QUAN VỀ SEMANTIC SEGMENTATION	1
1.1 GIỚI THIỆU VỀ LÝ THUYẾT HỌC SÂU	1
1.1.1 Sự khác biệt giữa học sâu và học máy	2
1.1.2 Mạng nơ-ron sâu	3
1.1.3 Mạng nơ-ron tích chập	5
1.1.4 Cơ chế tự chú ý	9
1.2 CÁC BÀI TOÁN DEEP LEARNING TRONG XỬ LÝ ẢNH	11
1.2.1 Image Classification	11
1.2.2 Object Detection	12
1.2.3 Image Segmentation	12
1.3 GIỚI THIỆU VỀ SEMANTIC SEGMENTATION	13
1.4 PHÂN LOẠI CÁC DẠNG PHÂN ĐOẠN ẢNH	13
1.5 PHÂN ĐOẠN NGỮ NGHĨA TRONG ẢNH DÙNG HỌC SÂU	14
1.6 VAI TRÒ VÀ ỨNG DỤNG TRONG THỰC TẾ	15
CHƯƠNG 2 CÁC THUẬT TOÁN LIÊN QUAN	17
2.1 GIỚI THIỆU VỀ CÁC THUẬT TOÁN TRUYỀN THỐNG	17
2.2 TỔNG QUANG VỀ MẠNG HỌC SÂU TRONG PHÂN ĐOẠN NGỮ NGHĨA HÌNH ẢNH	19
2.2.1 U-Net	19
2.2.2 SegNet	20
2.2.3 DeepLab V1/V2/V3/V3+	21
CHƯƠNG 3 THIẾT KẾ HỆ THỐNG	25

3.1	YÊU CẦU VỀ KIẾN TRÚC MẠNG	25
3.2	TẬP DỮ LIỆU	25
3.3	TRIỂN KHAI THIẾT KẾ KIẾN TRÚC MẠNG	26
3.3.1	Tổng quan về mô hình	26
3.3.2	Multi-Scale Convolution	31
3.3.3	Spatial Attention	32
CHƯƠNG 4	KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ	35
4.1	MÔI TRƯỜNG THỰC NGHIỆM	35
4.2	CÁC TIÊU CHÍ ĐÁNH GIÁ	36
4.3	KẾT QUẢ HUẤN LUYỆN	36
4.4	ĐÁNH GIÁ KẾT QUẢ	37
4.5	NHẬN XÉT VÀ KẾT LUẬN	41
TÀI LIỆU THAM KHẢO		43

Danh sách hình ảnh

Hình 1.1	Mối quan hệ giữa AI,ML và DL.	3
Hình 1.2	Kiến trúc cơ bản của một DNN.	4
Hình 1.3	Cơ chế của tích chập.	6
Hình 1.4	Minh họa việc tính toán tích chập.	7
Hình 1.5	Các hàm kích hoạt trong neural network.	8
Hình 1.6	Minh họa việc tính toán trên lớp Pooling.	8
Hình 1.7	Minh họa lớp kết nối toàn bộ (FC).	9
Hình 1.8	Kiến trúc AM cơ bản.	10
Hình 1.9	Mô hình cơ bản của bài toán Image Classification.	11
Hình 1.10	Sự khác biệt giữa Image Classification và Object Detection.	12
Hình 1.11	Các kỹ thuật phân đoạn hình ảnh.	14
Hình 1.12	Kiến trúc mạng nơ-ron tích chập dạng encoder-decoder.	15
Hình 1.13	Phân đoạn ngữ nghĩa cho xe tự hành.	16
Hình 1.14	Phân đoạn ngữ nghĩa cho y tế	16
Hình 2.1	Các phương pháp phân đoạn hình ảnh truyền thống.	17
Hình 2.2	Minh họa quá trình phân ngưỡng.	18
Hình 2.3	Minh họa cho phương pháp phân đoạn dựa trên biên	18
Hình 2.4	Minh họa cho quá trình phân đoạn dựa trên vùng	19
Hình 2.5	Kiến trúc U-net.	20
Hình 2.6	Kiến trúc SegNet.	21
Hình 2.7	Kết quả thực nghiệm vượt trội so với các phương pháp khác.	22
Hình 2.8	Kiến trúc chung DeepLab.	23
Hình 2.9	ASPP trong Kiến trúc DeepLabv2.	23
Hình 2.10	Kiến trúc DeepLabv3+.	24
Hình 3.1	Bộ dữ liệu SpectrogramSignal.	26
Hình 3.2	Mô hình kiến trúc được thiết kế trong hệ thống.	28

Hình 3.3	Sơ đồ khái minh họa các lớp mạng trong mô hình.	30
Hình 3.4	Sơ đồ khái minh họa Multi-Scale Conv Block.	31
Hình 3.5	Spatial Attention .	33
Hình 4.3	Kết quả huấn luyện .	38
Hình 4.4	Biểu đồ hàm mất mát(Loss).	39
Hình 4.5	Biểu đồ chỉ số IoU (Mean IoU).	39
Hình 4.6	Biểu đồ độ chính xác trung bình (Mean Accuracy).	40

Danh sách bảng

Bảng 3.1	Các thành phần chính và mục tiêu trong kiến trúc mô hình	28
Bảng 3.2	Vai trò chính của các thành phần trong kiến trúc mạng	29
Bảng 4.1	Kết quả huấn luyện và đánh giá mô hình qua các epoch	37

Danh mục các từ viết tắt

Dưới đây là danh mục các từ viết tắt được sử dụng trong luận văn.

Các từ viết tắt	Định nghĩa
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence
ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
AM	Self-Attention Mechanism
FC	Fully Connected Layer
ATM	Attention Mechanism
LTE	Long-Term Evolution
5G	Fifth-Generation

Chương 1

TỔNG QUAN VỀ SEMANTIC SEGMENTATION

1.1 GIỚI THIỆU VỀ LÝ THUYẾT HỌC SÂU

Deep Learning là một nhánh đặc biệt của Machine Learning, có khả năng mô phỏng quá trình tư duy của con người thông qua việc xây dựng và huấn luyện các mạng nơ-ron nhân tạo nhiều lớp (Deep Neural Networks - DNN). Khác với các phương pháp học máy truyền thống chỉ sử dụng một hoặc hai lớp xử lý, các mô hình deep learning thường bao gồm hàng trăm đến hàng nghìn lớp, tạo nên khả năng học hỏi và phân tích dữ liệu ở mức độ phức tạp hơn nhiều.

Sức mạnh cốt lõi của deep learning nằm ở khả năng tự động học các biểu diễn trừu tượng (representation learning) từ dữ liệu thô mà không cần sự can thiệp thủ công. Trong khi các phương pháp machine learning truyền thống đòi hỏi quá trình feature engineering tốn kém thời gian, các mô hình deep learning có thể tự động trích xuất đặc trưng phức tạp qua nhiều lớp xử lý. Mỗi lớp trong mạng sẽ học các biểu diễn ở mức độ trừu tượng cao hơn so với lớp trước đó.

Quá trình học tập của DNN diễn ra thông qua cơ chế lan truyền ngược (backpropagation) và các thuật toán tối ưu hóa như Gradient Descent. Trong quá trình này, mạng tính toán sự chênh lệch giữa kết quả dự đoán và giá trị thực tế (hàm mất mát), sau đó điều chỉnh trọng số của các nơ-ron để giảm thiểu sai số. Điều này cho phép mạng liên tục cải thiện hiệu suất qua nhiều vòng lặp huấn luyện.. Ngoài ra, DNN nổi bật với khả năng học các biểu diễn đặc trưng từ dữ liệu. Các lớp ẩn của mạng nơ-ron sâu có khả năng tạo ra các biểu diễn ngữ cảnh của dữ liệu đầu vào, các biểu diễn này có thể bao gồm thông tin quan trọng và trừu tượng giúp mạng nắm bắt các đặc trưng phức tạp của dữ liệu. Ngày càng có nhiều mô hình mới được xây dựng nhằm đáp ứng yêu cầu cao hơn của con người về độ chính xác, có thể kể đến như mạng nơ-ron tích chập (convolutional neural network – CNN), hay các mô hình được xây dựng hoàn toàn bằng cơ chế tự chú ý.

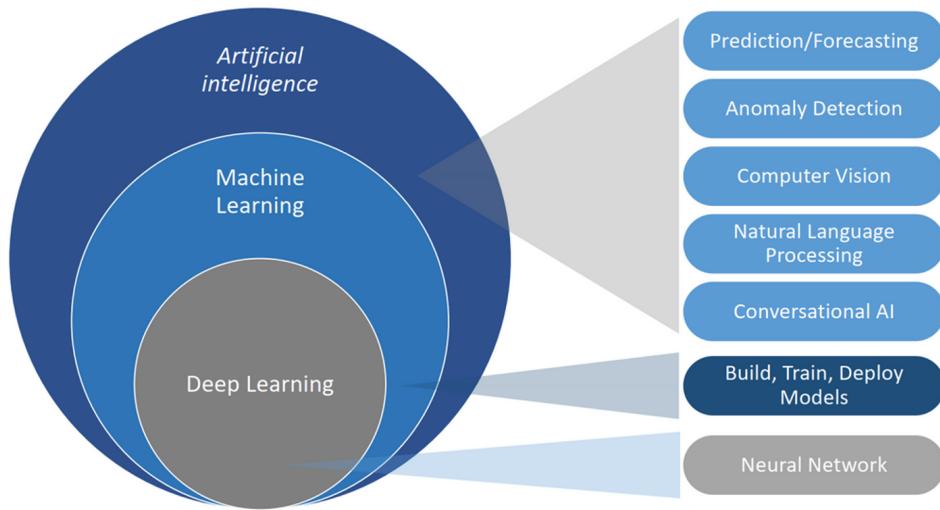
Deep Learning đã chứng minh hiệu quả vượt trội trong nhiều ứng dụng thực tế như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, hệ thống gợi ý, xe tự hành và AI tạo sinh. Tuy nhiên, để đạt hiệu quả, các mô hình deep learning thường đòi hỏi lượng dữ liệu huấn luyện lớn và tài nguyên tính toán cao.

1.1.1 Sự khác biệt giữa học sâu và học máy

Học máy (Machine Learning) và học sâu (Deep Learning) đều là những lĩnh vực quan trọng thuộc trí tuệ nhân tạo (AI), tuy nhiên chúng có những điểm khác biệt cơ bản về cách thức hoạt động, yêu cầu dữ liệu và ứng dụng thực tiễn. Học máy là quá trình giúp máy tính có khả năng học hỏi từ dữ liệu và tự cải thiện mà không cần được lập trình cụ thể cho từng tình huống. Trong học máy truyền thống, con người đóng vai trò lớn trong việc lựa chọn và trích xuất các đặc trưng (features) từ dữ liệu, sau đó sử dụng các thuật toán như Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) hoặc Random Forest để xây dựng mô hình dự đoán hoặc phân loại. Một ví dụ điển hình là khi xây dựng hệ thống dự đoán giá nhà, kỹ sư dữ liệu cần lựa chọn các yếu tố đầu vào như diện tích nhà, số phòng ngủ, vị trí địa lý, sau đó áp dụng thuật toán học máy để dự đoán giá bán phù hợp.

Ngược lại, học sâu là một nhánh mở rộng của học máy, dựa trên các mạng DNN, đặc biệt có khả năng tự động học đặc trưng từ dữ liệu mà không cần sự can thiệp thủ công của con người. Học sâu đòi hỏi lượng dữ liệu khổng lồ và sức mạnh tính toán cao, nhưng có thể giải quyết những bài toán cực kỳ phức tạp mà học máy truyền thống khó đạt hiệu quả cao. Một ví dụ phổ biến về học sâu là việc sử dụng mạng CNN để tự động nhận diện khuôn mặt trong hàng triệu bức ảnh mà không cần lập trình chi tiết về cách phát hiện mắt, mũi, miệng. Ngoài ra, học sâu còn được áp dụng rộng rãi trong xe tự lái, dịch tự động giữa các ngôn ngữ, phát hiện gian lận tài chính, và thậm chí trong y tế để hỗ trợ chẩn đoán bệnh qua hình ảnh y khoa.

Một điểm khác biệt rõ rệt nữa giữa học máy và học sâu là mức độ giải thích của mô hình. Các thuật toán học máy truyền thống như Decision Tree thường dễ dàng phân tích và giải thích, giúp con người hiểu được quyết định của mô hình. Trong khi đó, các mô hình học sâu lại hoạt động như những "hộp đen", rất khó để lý giải tại sao mô hình lại đưa ra một quyết định cụ thể nào đó, mặc dù hiệu suất có thể rất cao. Nhìn chung, việc lựa chọn giữa học máy và học sâu phụ thuộc vào bài toán cụ thể, khối lượng dữ liệu có sẵn, yêu cầu về độ chính xác cũng như khả năng giải thích của hệ thống.



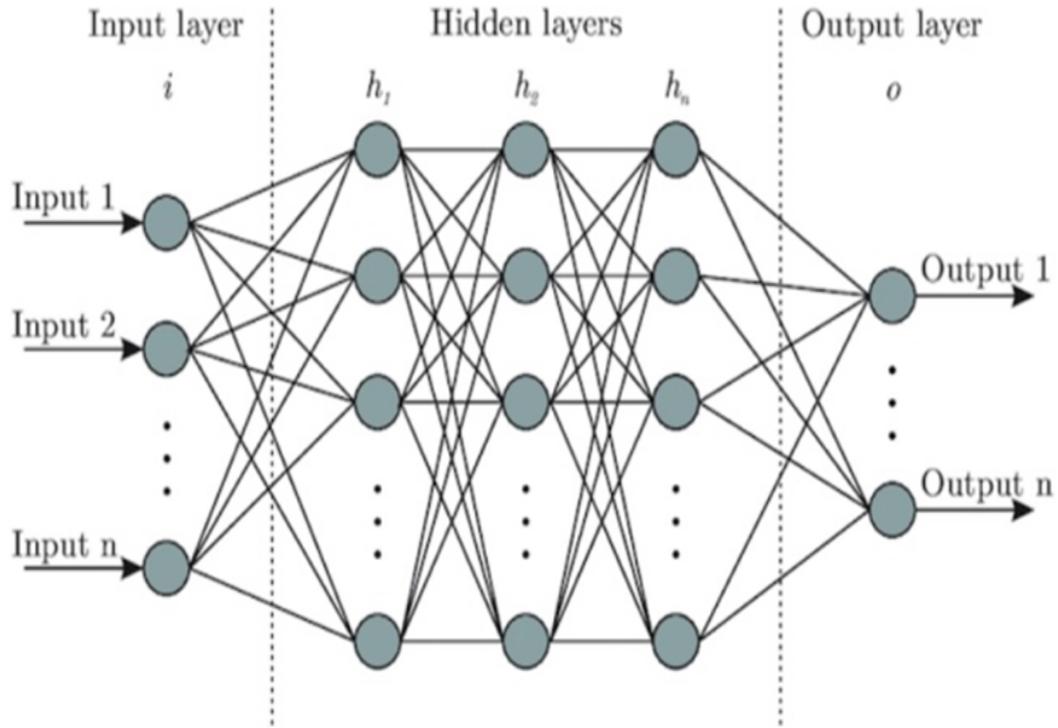
Hình 1.1: Mối quan hệ giữa AI,ML và DL

1.1.2 Mạng nơ-ron sâu

Mạng nơ-ron sâu (DNN) là một kiến trúc máy tính tiên tiến được thiết kế để mô phỏng cách thức não bộ con người xử lý thông tin. Các nơ-ron nhân tạo được thiết kế tương tự như các nơ-ron sinh học. Nó tiếp nhận các tín hiệu đầu vào, ứng với mỗi tín hiệu này sẽ là một trọng số. Các trọng số đại diện cho mức độ quan trọng của tín hiệu đầu vào và sau đó nơ-ron nhân tạo sẽ tổng hợp và đưa vào các hàm kích hoạt. Khác biệt cơ bản giữa DNN và mạng nơ-ron thông thường nằm ở độ sâu của mạng - số lượng lớp xử lý giữa đầu vào và đầu ra. Trong khi mạng nơ-ron truyền thống chỉ có 1-2 lớp ẩn, DNN có thể chứa hàng chục, hàng trăm, thậm chí hàng nghìn lớp xử lý phức tạp.

Kiến trúc điển hình của một DNN bao gồm:

- Lớp đầu vào (input layer): Lớp này có số nơ-ron tương ứng với số biến không phụ thuộc lẫn nhau, thường được xem là kích thước đầu vào của mô hình và tiếp nhận dữ liệu thô từ mô hình
- Lớp đầu ra (output layer): Đưa ra kết quả cuối cùng, lớp đầu ra có số nơ-ron tương ứng với số ngõ ra trong bài toán phân loại, hay còn có tên khác là số lớp đối tượng.
- Lớp ẩn (hidden layer): Đây là nơi diễn ra quá trình xử lý và trích xuất các đặc trưng, lớp không có quy định cụ thể về số lượng nơ-ron. Thông thường số lượng nơ-ron trong mỗi lớp ẩn thường là một số lũy thừa của 2. Mỗi lớp ẩn bao gồm nhiều nơ-ron, với mỗi nơ-ron kết nối đến tất cả nơ-ron ở lớp trước đó. Số lượng lớp ẩn và số nơ-ron trong mỗi lớp quyết định khả năng học của mạng. Các lớp đầu thường học các đặc trưng đơn giản như đường viền, góc cạnh, trong khi các lớp sau tổng hợp thành các mẫu phức tạp hơn như khuôn mặt, chữ viết, hay các khái niệm trừu tượng.



Hình 1.2: Kiến trúc cơ bản của một DNN.

Hình 1.2 mô tả cấu trúc của một DNN đơn giản chỉ bao gồm 1 lớp đầu vào lấy dữ liệu, 3 lớp ẩn và 1 lớp đầu ra giúp phân loại dữ liệu.

Cơ chế hoạt động và học tập Với khả năng học biểu diễn phức tạp và tự động trích xuất đặc trưng. Trong quá trình huấn luyện, DNN liên tục điều chỉnh các trọng số của các kết nối giữa các nơ-ron thông qua thuật toán "lan truyền ngược"(backpropagation):

- Mạng xử lý dữ liệu đầu vào và đưa ra dự đoán
- Hệ thống tính toán sai số giữa dự đoán và kết quả mong muốn
- Sai số được lan truyền ngược qua các lớp của mạng
- Các trọng số được điều chỉnh để giảm thiểu sai số
- Quá trình lặp lại nhiều lần với nhiều mẫu dữ liệu khác nhau

Thông qua quy trình lặp đi lặp lại này, DNN dần dần tinh chỉnh hàng triệu tham số để phản ánh chính xác các mẫu và mối quan hệ trong dữ liệu.

Ứng dụng của Mạng Nơ-ron Sâu (DNN): DNN đã tạo ra những bước đột phá ấn tượng trong nhiều lĩnh vực, mở rộng khả năng ứng dụng của trí tuệ nhân tạo vào đời sống hàng ngày. Dưới đây là những ứng dụng nổi bật của DNN:

- Xử lý hình ảnh và thị giác máy tính: DNN giúp thực hiện nhận dạng đối tượng, phát hiện khuôn mặt, phân đoạn hình ảnh, tạo hình ảnh từ mô tả văn bản và nâng cao

chất lượng ảnh. Các hệ thống nhận dạng hình ảnh hiện đại như trong điện thoại thông minh, camera an ninh đều dựa trên công nghệ này.

- Y học và chăm sóc sức khỏe: áp ứng dụng trong chẩn đoán hình ảnh y tế, dự đoán bệnh, phát triển thuốc, giám sát sức khỏe và hỗ trợ phẫu thuật robot
- Công nghệ xe tự lái và robot tự động: Sử dụng DNN để nhận diện môi trường, dự đoán hành vi, lập kế hoạch đường đi và điều khiển chuyển động chính xác.
- Trong lĩnh vực tài chính và kinh doanh: DNN giúp dự đoán thị trường chứng khoán, phát hiện gian lận, chấm điểm tín dụng, xây dựng hệ thống gợi ý sản phẩm và dự báo nhu cầu thị trường

Tóm lại, mạng nơ-ron sâu đãi diện cho một phần quan trọng của lĩnh vực trí tuệ nhân tạo và đã mang lại nhiều đóng góp quan trọng cho nhiều lĩnh vực ứng dụng khác nhau. Sự đơn giản và tính linh hoạt của DNN làm cho nó trở thành một công cụ mạnh mẽ trong việc giải quyết các vấn đề phức tạp và không đồng nhất trong thế giới thực.

1.1.3 Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (CNN) là một kiến trúc đặc biệt được thiết kế nhằm nâng cao hiệu quả xử lý dữ liệu hình ảnh. Ưu điểm chính của CNN là khả năng phát hiện các mẫu đặc trưng không phụ thuộc vào vị trí trong hình ảnh, nhờ vào cơ chế tích chập di chuyển trên toàn bộ ảnh đầu vào.

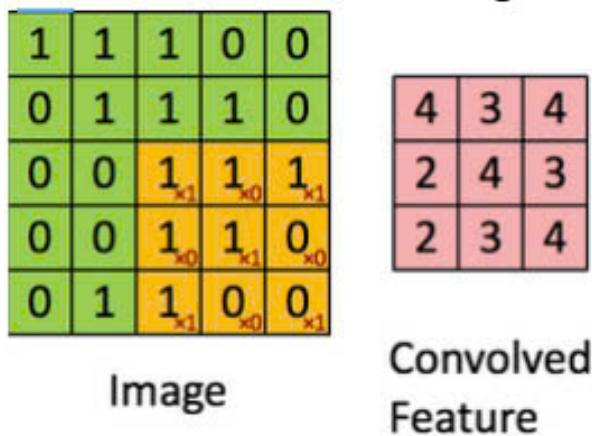
So với mạng nơ-ron truyền thống sử dụng các lớp kết nối đầy đủ, CNN giảm đáng kể số lượng tham số cần huấn luyện. Điều này không chỉ giúp mô hình chạy hiệu quả hơn mà còn cải thiện khả năng tổng quát hóa, đặc biệt quan trọng khi làm việc với dữ liệu hình ảnh có độ phức tạp cao.

Convolution (tích chập)

Tích chập là kỹ thuật được sử dụng đầu tiên trong xử lý tín hiệu số. Nhờ khả năng biến đổi thông tin hiệu quả, các nhà khoa học đã áp dụng kỹ thuật này vào lĩnh vực xử lý ảnh và video số. Về bản chất, tích chập hoạt động như một cửa sổ trượt (sliding window) di chuyển trên một ma trận dữ liệu. Hình 1.3 là 1 ví dụ cụ thể để ta có thể theo dõi cơ chế của tích chập

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature. Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột. Convolution hay tích chập là nhân từng phần tử trong ma trận 3.

Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có



Hình 1.3: Cơ chế của tích chập.

kích thước nhỏ như trong ví dụ trên là 3x3. Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3x3 với ma trận bên trái. Kết quả được một ma trận gọi là Convolved feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5x5 bên trái.

Cấu trúc mạng CNN

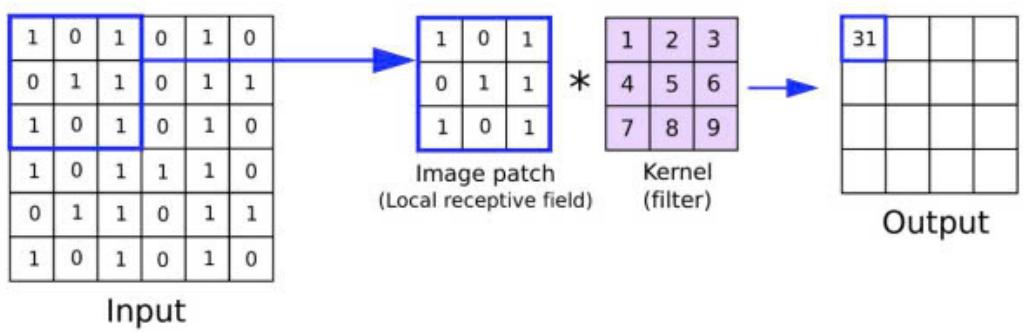
Mạng CNN là một hệ thống nhiều lớp tích chập (convolution) được xếp chồng lên nhau, kết hợp với các hàm kích hoạt phi tuyến như ReLU và các lớp gộp (pooling/subsampling) để kích hoạt các trọng số trong các node. Qua mỗi lớp xử lý, thông tin trở nên trừu tượng hơn, giúp mạng học được các đặc trưng phức tạp.

Khác với mạng nơ-ron truyền thống sử dụng kết nối đầy đủ (fully connected layer) - nơi mỗi nơ-ron đầu vào kết nối với tất cả nơ-ron đầu ra, CNN sử dụng cơ chế kết nối cục bộ thông qua tích chập. Mỗi nơ-ron ở lớp tiếp theo chỉ nhận thông tin từ một vùng nhỏ của lớp trước, được tạo ra bằng cách áp dụng bộ lọc (filter) lên vùng cục bộ đó. Cách tiếp cận này cho phép CNN hiệu quả hơn trong việc xử lý dữ liệu hình ảnh, đồng thời giảm đáng kể số lượng tham số cần huấn luyện.

- **Lớp tích chập (convolutional layer):** Lớp tích chập là thành phần cốt lõi của mạng nơ-ron tích chập (CNN), đóng vai trò quan trọng trong việc trích xuất đặc trưng từ dữ liệu đầu vào. Hoạt động chính của lớp này dựa trên phép toán tích chập giữa dữ liệu đầu vào và các bộ lọc (gọi là filter hoặc kernel). Bộ lọc này hoạt động như các cửa sổ trượt trên ảnh để trích xuất các đặc trưng cục bộ, chẳng hạn như cạnh, góc, và các thông tin đặc biệt. Các lớp tích chập có thể có nhiều bộ lọc khác nhau để trích xuất nhiều đặc trưng khác nhau từ ảnh đầu vào.

Filter hoặc Feature Detector: ma trận lọc, thông thường có kích thước 3x3 hoặc 5x5.

Convolved Feature, Activation Map hoặc Feature map: là đầu ra của ảnh khi



Hình 1.4: Minh họa việc tính toán tích chập

cho bộ lọc chạy qua khi sử dụng phép tính tích vô hướng.

Receptive Field: là các vùng nhỏ trong ảnh để tính tích chập có kích thước giống với bộ lọc.

Depth: số lượng bộ lọc.

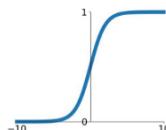
Stride: là khoảng cách dịch chuyển của bộ lọc sau mỗi lần tính tích chập trên 1 receptive field. Ví dụ với stride=1 tương đương việc dịch sang phải hoặc xuống dưới 1 pixel tùy vào vị trí của vùng ảnh vừa tính toán.

Zero padding: là việc thêm các giá trị 0 ở xung quanh biến ảnh để tính toán thêm đặc trưng ở các vùng biên

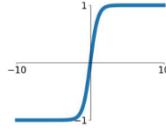
- **Lớp chuẩn hóa (batch normalization):** Là một phần quan trọng của mạng nơ-ron sâu, được sử dụng để chuẩn hóa các giá trị đầu ra của mỗi lớp tích chập trong quá trình huấn luyện bằng cách điều chỉnh giá trị trung bình và phương sai của các batch dữ liệu, giúp tăng tốc quá trình học và giảm hiện tượng quá khớp (overfitting).

- **Hàm kích hoạt (activation layer):** Sau mỗi lớp tích chập đã được chuẩn hóa, một hàm kích hoạt được áp dụng vào các bản đồ đặc trưng. Điều này giúp mạng học các biểu diễn phi tuyến性和 làm tăng khả năng học của nó.

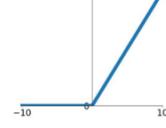
Sigmoid
 $\sigma(x) = \frac{1}{1+e^{-x}}$



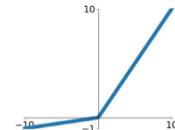
tanh
 $\tanh(x)$



ReLU
 $\max(0, x)$



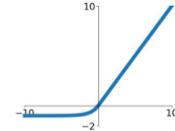
Leaky ReLU
 $\max(0.1x, x)$



Maxout
 $\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

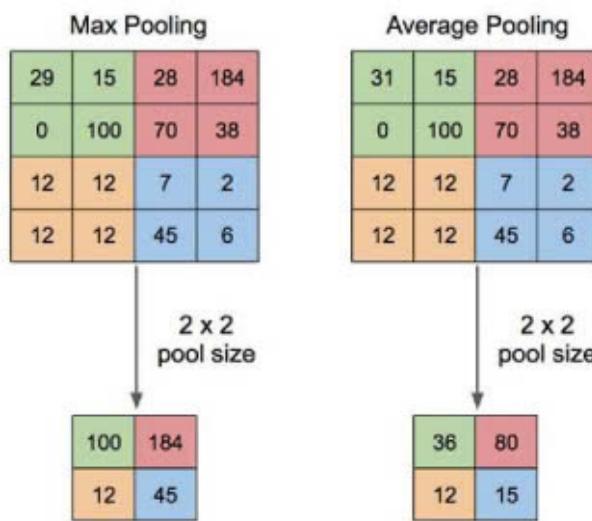
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Hình 1.5: Các hàm kích hoạt trong neural network

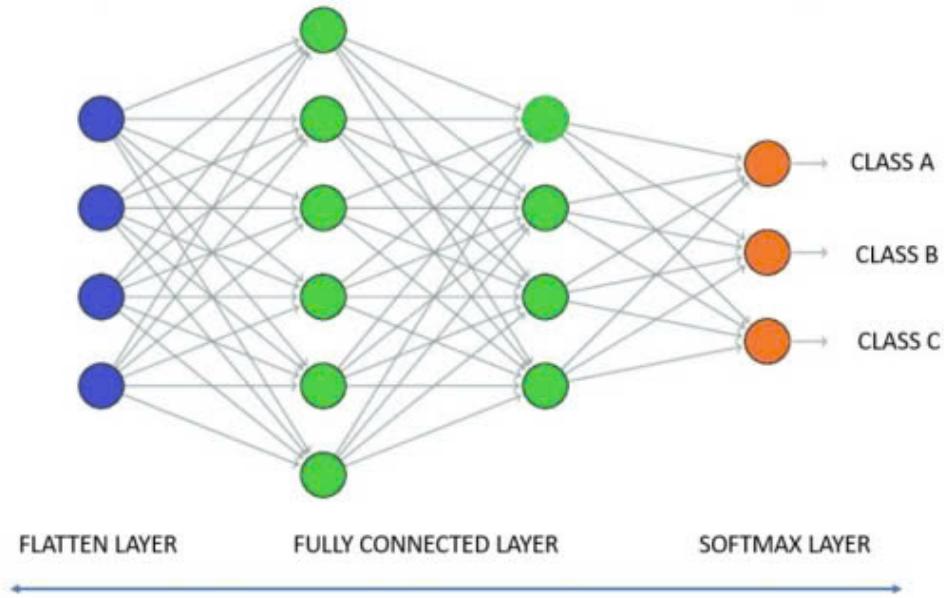
• Lớp tổng hợp (pooling layer): Tác dụng của lớp pooling là giảm kích thước đầu vào là các đặc trưng tích xuất từ lớp tích chập, giúp giảm độ phức tạp tính toán và giúp kiểm soát hiện tượng overfitting. Có 2 phương pháp pooling được sử dụng phổ biến:

- + Max pooling: thay thế giá trị vùng ảnh bởi giá trị pixel lớn nhất trong nó. Ý nghĩa là giúp giữ lại những vùng thông tin lớn nhất đại diện cho vùng ảnh.
- + Average Pooling: thay thế giá trị vùng ảnh bởi giá trị trung bình các pixel trong vùng. Ý nghĩa là giúp lấy thông tin tổng thể để đại diện cho vùng ảnh.



Hình 1.6: Minh họa việc tính toán trên lớp Pooling

- Lớp kết nối toàn bộ (FC): Thường thì sau các lớp Conv+Pooling thì sẽ là 2 lớp Fully connected, một layer để tập hợp tất cả các feature layer mà đã được học ở các lớp trước, chuyển đổi dữ liệu từ 3-D, hoặc 2-D thành 1-D, tức chỉ còn là 1 vector. Lớp cuối cùng này tổng hợp tất cả các feature để phân loại hoặc dự đoán cuối cùng. Số neuron của layer này phụ thuộc vào số output mà ta muốn tìm ra. Lớp kết nối đầy đủ còn được biết đến là các lớp ẩn trong DNN, được CNN kế thừa từ DNN.



Hình 1.7: Minh họa lớp kết nối toàn bộ

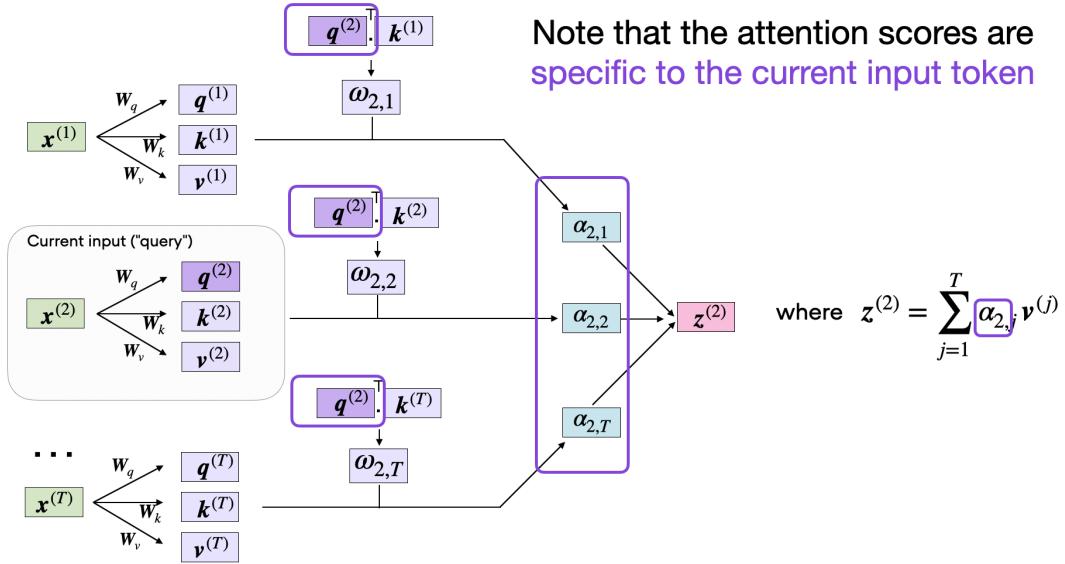
1.1.4 Cơ chế tự chú ý

Cơ chế tự chú ý (self attention mechanism – AM) là một kỹ thuật tiên tiến trong học sâu, nó cho phép mô hình học các mối quan hệ giữa các phần khác nhau của dữ liệu đầu vào, không phụ thuộc vào khoảng cách không gian giữa chúng.

Cấu trúc của AM: Cấu trúc cơ bản của cơ chế tự chú ý bao gồm ba thành phần chính được cấu thành từ ba vector có kích thước tương đồng:

- Vector Query: Biểu diễn cho câu hỏi hoặc mục tiêu mà mô hình muốn tập trung vào, được tính toán từ đầu vào của mô hình, thể hiện mong muốn của mô hình về thông tin quan trọng cần được chú ý.
- Vector Key: Biểu diễn cho bối cảnh hoặc thông tin mà mô hình cần tham khảo để trả lời câu hỏi. Vector Key được tính toán từ đầu vào của mô hình, cung cấp bối cảnh cho mô hình để xác định phần nào của đầu vào có liên quan nhất đến câu hỏi.
- Vector Value: Biểu diễn cho thông tin liên quan đến câu hỏi và bối cảnh, được tính toán từ đầu vào của mô hình, cung cấp thông tin cụ thể để mô hình trả lời câu hỏi.

Sau khi biến đổi, ba tập hợp vector biểu diễn này được sử dụng để tính toán một điểm tương đồng giữa mỗi cặp vector Query và Key. Điểm tương đồng này thể hiện mức độ liên quan giữa hai vector. Tiếp theo, điểm tương đồng được chuẩn hóa bằng một hàm softmax để đảm bảo tổng của các điểm tương đồng cho mỗi Key bằng 1. Cuối cùng, các điểm tương đồng được sử dụng để tạo trọng số cho các vector Value, từ đó tạo ra vector đầu ra là tổng hợp có trọng số của các vector Value.



Hình 1.8: Kiến trúc AM cơ bản. Trên hình ảnh đang biểu diễn cơ chế self-attention trong mô hình Transformer

Lợi ích của AM: AM mang lại nhiều lợi ích quan trọng trong các mô hình phân đoạn ngữ nghĩa hình ảnh. AM cho phép mô hình nắm bắt ngữ cảnh toàn cục bằng cách xem xét mối quan hệ giữa các vùng xa nhau trong ảnh, vượt qua giới hạn trường tiếp nhận cục bộ của CNN truyền thống. Điều này giúp mô hình học được các mối quan hệ phụ thuộc dài hạn, xác định liên hệ giữa các cấu trúc xa nhau và hỗ trợ nhận diện đối tượng phức tạp có cấu trúc phân tán.

Một ưu điểm quan trọng khác là khả năng điều chỉnh trọng số thích ứng theo ngữ cảnh, tự động tập trung vào các vùng quan trọng và lọc bỏ thông tin không liên quan. AM cũng cải thiện đáng kể độ chính xác tại các ranh giới đối tượng và phân biệt tốt hơn giữa các lớp dễ gây nhầm lẫn.

Cơ chế chú ý còn gia tăng khả năng tổng quát hóa của mô hình, giúp xử lý tốt hơn các trường hợp chưa từng gặp trong tập huấn luyện và thích ứng với biến thể về hình dạng, kích thước của đối tượng. Tính phân cấp và đa tỷ lệ của AM cho phép kết hợp thông tin từ nhiều mức trừu tượng khác nhau, hỗ trợ xử lý đối tượng ở nhiều kích thước trong cùng một ảnh.

Cuối cùng, AM mang lại hiệu quả tham số cao, đạt hiệu suất tốt hơn mà không cần tăng đáng kể số lượng tham số, đồng thời cung cấp khả năng diễn giải thông qua việc trực quan hóa các ma trận chú ý để hiểu quá trình ra quyết định của mô hình

1.2 CÁC BÀI TOÁN DEEP LEARNING TRONG XỬ LÝ ẢNH

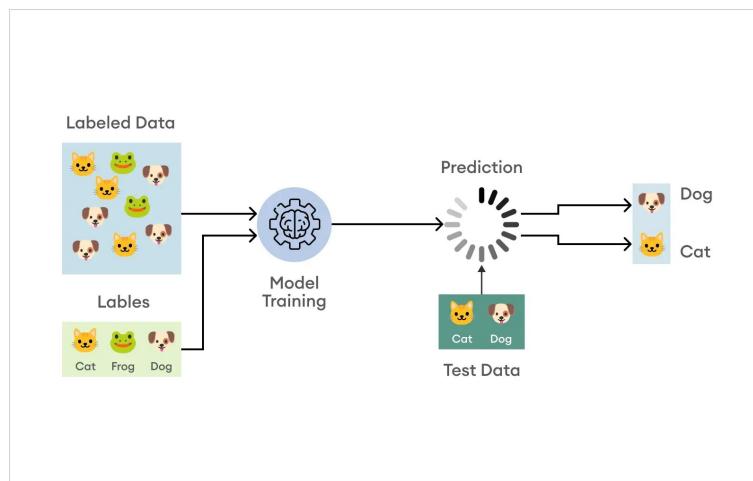
1.2.1 Image Classification

Image Classification (Phân loại ảnh) là bài toán gán nhãn cho toàn bộ hình ảnh dựa trên nội dung chính của nó. Đây là bài toán cơ bản nhất trong thị giác máy tính sử dụng deep learning [1].

Trong bài toán này, mạng neural được huấn luyện để nhận dạng đối tượng chính xuất hiện trong ảnh và gán nó vào một lớp xác định. Đặc điểm quan trọng là mô hình chỉ xác định lớp của toàn bộ ảnh mà không xác định vị trí hay ranh giới của đối tượng.

Kiến trúc mạng neural thường dùng cho bài toán này là Convolutional Neural Networks (CNNs) với các mô hình tiêu biểu như AlexNet, VGG, ResNet, Inception, EfficientNet. Các mô hình này thường bao gồm nhiều lớp tích chập để trích xuất đặc trưng, tiếp theo là các lớp fully connected để thực hiện phân loại [1].

Ứng dụng của Image Classification bao gồm nhận dạng khuôn mặt, phân loại tài liệu, phân loại sản phẩm, phân tích hình ảnh y tế, và nhiều lĩnh vực khác.



Hình 1.9: Mô hình cơ bản của bài toán Image Classification. Bên trái là quá trình huấn luyện, với dữ liệu có nhãn (Labeled Data) gồm các biểu tượng mèo, ếch và chó cùng với nhãn tương ứng (Cat, Frog, Dog). Các dữ liệu này được đưa vào huấn luyện mô hình (Model Training). Bên phải là quá trình dự đoán, khi mô hình đã huấn luyện nhận dữ liệu kiểm tra (Test Data) chứa biểu tượng mèo và chó mới. Mô hình xử lý và đưa ra kết quả dự đoán (Prediction) chính xác rằng đây là "Cat" (mèo) và "Dog" (chó).

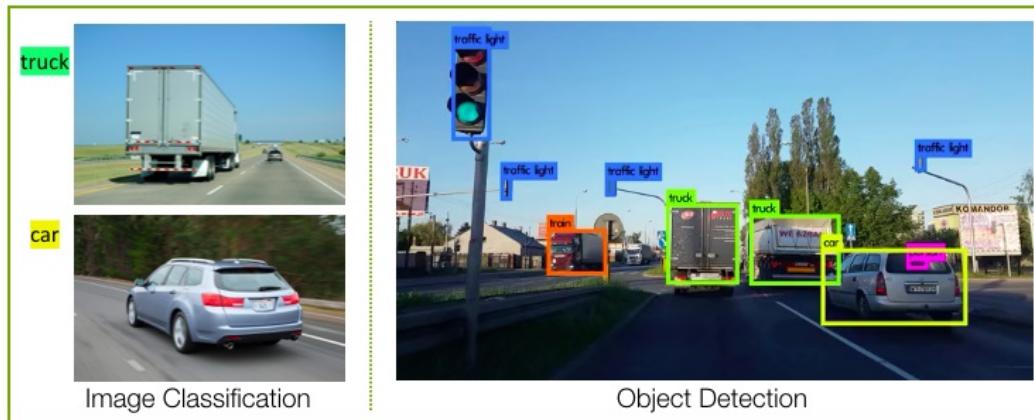
1.2.2 Object Detection

Object Detection (Phát hiện đối tượng) phức tạp hơn so với phân loại ảnh. Ngoài việc xác định lớp của đối tượng, bài toán này còn xác định vị trí của đối tượng trong ảnh thông qua bounding box (khung hình chữ nhật).

Đầu ra của mô hình Object Detection bao gồm danh sách các đối tượng được phát hiện cùng với tọa độ bounding box và nhãn lớp tương ứng. Mô hình có thể phát hiện nhiều đối tượng thuộc các lớp khác nhau trong cùng một ảnh. Các kiến trúc phổ biến cho bài toán này được chia thành hai nhóm chính:

Two-stage detectors: R-CNN, Fast R-CNN, Faster R-CNN
One-stage detectors: YOLO, SSD, RetinaNet

Object Detection được ứng dụng rộng rãi trong xe tự lái, giám sát an ninh, phân tích hình ảnh y tế, và hệ thống thực tế tăng cường.



Hình 1.10: Sự khác biệt giữa Image Classification và Object Detection. Object Detection phát hiện nhiều đối tượng trong cùng một ảnh, vẽ bounding box xung quanh mỗi đối tượng và gắn nhãn cho từng đối tượng ví dụ truck, car, traffic light.

1.2.3 Image Segmentation

Image Segmentation (Phân đoạn ảnh) là bài toán phức tạp nhất, với mục tiêu phân loại từng pixel trong ảnh. Bài toán này yêu cầu mô hình hiểu ảnh ở mức độ chi tiết nhất. Image Segmentation được chia thành các loại sau:

- Semantic Segmentation
- Instance Segmentation
- Panoptic Segmentation

Image Segmentation được ứng dụng trong nhiều lĩnh vực như y học (phân đoạn cơ

quan, khối u), xe tự lái (hiểu môi trường xung quanh), thực tế ảo, và chỉnh sửa ảnh. Mỗi bài toán trên đại diện cho một cấp độ phân tích ảnh ngày càng chi tiết và phức tạp, từ nhận dạng nội dung tổng thể (classification), đến vị trí cụ thể (detection), và cuối cùng là phân tích ở cấp độ pixel (segmentation).

1.3 GIỚI THIỆU VỀ SEMANTIC SEGMENTATION

Trong vài thập kỷ qua, một trong những bài toán thách thức nhất trong lĩnh vực thị giác máy tính là phân đoạn ảnh (image segmentation). Bài toán này đóng vai trò quan trọng trong việc hiểu và phân tích nội dung ảnh ở cấp độ pixel – từ ảnh tĩnh 2D, video cho tới dữ liệu không gian 3D hay thể tích (volumetric data) [2].

Trong đó, phân đoạn ngữ nghĩa (semantic segmentation) nổi lên như một nhánh quan trọng, đóng vai trò then chốt trong việc hiểu ngữ cảnh toàn diện của một cảnh (scene understanding). Thuật toán này có khả năng gán một nhãn ngữ nghĩa (class label) cho mỗi pixel trong ảnh, qua đó phân biệt rõ ràng giữa các vùng mang ý nghĩa khác nhau như con đường, người, xe cộ, cây cối,...

Trước thời kỳ của deep learning, các phương pháp phân đoạn chủ yếu dựa trên các kỹ thuật học máy cổ điển như phân cụm (clustering), phát hiện biên (edge detection), hay mô hình Markov ngẫu nhiên (Markov Random Field). Tuy nhiên, hiệu quả các phương pháp này bị hạn chế khi xử lý những ảnh phức tạp, đa đối tượng.

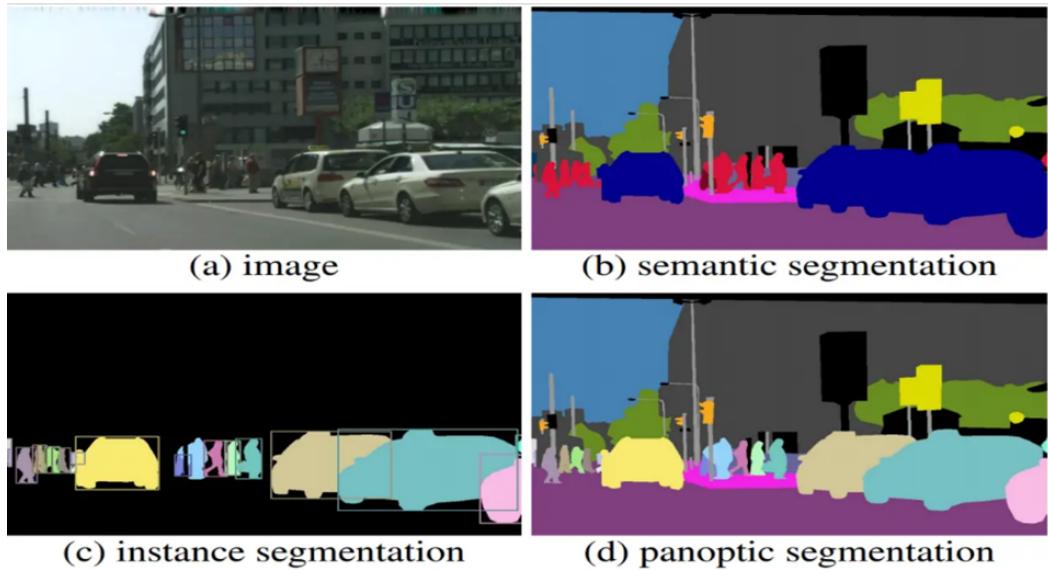
Sự xuất hiện của học sâu (deep learning), đặc biệt là mạng nơ-ron tích chập (CNN), đã tạo ra bước ngoặt lớn. Các mô hình học sâu hiện đại có khả năng tự học đặc trưng, phân tích cấu trúc không gian và mối quan hệ ngữ nghĩa giữa các vùng ảnh, nhờ đó đạt được kết quả vượt trội trên nhiều bộ dữ liệu chuẩn.

1.4 PHÂN LOẠI CÁC DẠNG PHÂN ĐOẠN ẢNH

Về bài toán phân vùng ảnh (Image Segment) không chỉ gói gọn trong semantic segmentation. Tùy theo mục tiêu và độ chi tiết, có thể chia thành các nhánh sau:

- Semantic Segmentation: Gán nhãn lớp cho từng pixel, nhưng không phân biệt các cá thể trong cùng lớp. Ví dụ, tất cả các xe ô tô trong ảnh đều là “xe” – cùng màu.
- Instance Segmentation: Kết hợp giữa nhận dạng đối tượng và phân đoạn, giúp phân biệt từng thực thể riêng biệt trong cùng một lớp. Ví dụ, 4 chiếc xe trong ảnh được tô 4 màu khác nhau
- Panoptic Segmentation: Tác vụ phân đoạn được phát triển gần đây nhất, là sự kết hợp cả semantic và instance segmentation, mỗi pixel được gán một nhãn lớp và một ID

đối tượng – hỗ trợ hiểu cảnh toàn diện và chi tiết.



Hình 1.11: Các kỹ thuật phân đoạn hình ảnh

1.5 PHÂN ĐOẠN NGỮ NGHĨA TRONG ẢNH DÙNG HỌC SÂU

Phân đoạn ngữ nghĩa là một công nghệ trong computer vision nhằm gắn nhãn cho từng điểm ảnh trong một bức hình. Khác với kỹ thuật nhận diện vật thể thông thường, phân đoạn ngữ nghĩa xác định chính xác biên giới và vị trí của các đối tượng, không chỉ đơn thuần là sự hiện diện của chúng.

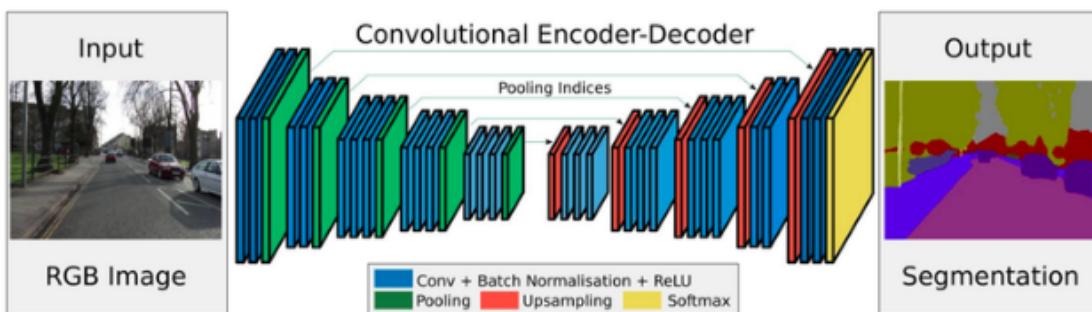
Học sâu, một nhánh của trí tuệ nhân tạo, sử dụng mạng nơ-ron nhiều tầng để nhận biết các mối liên hệ phức tạp giữa dữ liệu. Trong lĩnh vực phân đoạn ngữ nghĩa, hai kiến trúc chính được ưa chuộng là mạng nơ-ron tích chập (CNN) và kiến trúc Transformer với các khối cơ chế chú ý (Attention Mechanism). Mạng CNN có cấu trúc nhiều lớp tích chập để rút trích đặc điểm cục bộ, kết hợp với các lớp kết nối toàn phần để phân loại. Trong khi đó, Transformer tập trung vào việc xác định mối quan hệ không gian giữa các thành phần trong ảnh, làm nổi bật sự khác biệt giữa các đối tượng.

Hầu hết các mô hình phân đoạn hiện đại đều xây dựng dựa trên cấu trúc "mã hóa - giải mã" (encoder-decoder), một cấu trúc nổi tiếng qua mô hình U-Net và các biến thể:

Mã hóa (encoder): là phần đầu của kiến trúc, thường bao gồm một mạng nơ-ron sâu để biểu diễn hình ảnh đầu vào thành các đặc trưng có mức độ trừu tượng cao hơn. Các lớp trong bộ mã hóa thường bao gồm các lớp tích chập và các lớp tổng hợp, giúp giảm dần kích thước không gian của ảnh và tập trung vào các đặc trưng quan trọng. Việc này không chỉ giúp giảm số lượng thông tin cần thiết để xử lý mà còn giúp trích xuất các đặc trưng hữu ích từ hình ảnh. Trong xử lý hình ảnh, phần mã hóa giúp trích xuất thông

tin quan trọng từ hình ảnh, tạo nên một bản đồ đặc trưng chứa đựng các thông tin cần thiết cho quá trình phân đoạn. Các đặc trưng này sau đó được truyền đến phần giải mã.

Giải mã (decoder): là phần thứ hai của kiến trúc và thường bao gồm một mạng nơ-ron để tạo ra đầu ra dự đoán dựa trên thông tin từ bộ mã hóa. Bộ giải mã thường bao gồm các lớp tích chập chuyển vị (transposed convolutional layers) hoặc các lớp tăng kích thước (upsampling layers), giúp khôi phục lại kích thước không gian của ảnh về kích thước ban đầu. Mục tiêu của bộ giải mã là sử dụng các đặc trưng trùu tượng từ bộ mã hóa để tái tạo lại một bản đồ phân đoạn chính xác, nơi mỗi pixel được gán nhãn theo đối tượng tương ứng. Nhờ có sự kết hợp hiệu quả của cả hai phần, kiến trúc mã hóa – giải mã đã chứng minh tính hiệu quả vượt trội trong các nhiệm vụ phân đoạn ngữ nghĩa. Điều này đã thúc đẩy nhiều nghiên cứu và cải tiến trong lĩnh vực này, từ việc sử dụng các kiến trúc CNN cơ bản đến những cải tiến phức tạp hơn.



Hình 1.12: Hình này minh họa một kiến trúc mạng nơ-ron tích chập dạng encoder-decoder (mã hóa-giải mã) cho bài toán phân đoạn ngữ nghĩa ảnh

1.6 VAI TRÒ VÀ ỨNG DỤNG TRONG THỰC TẾ

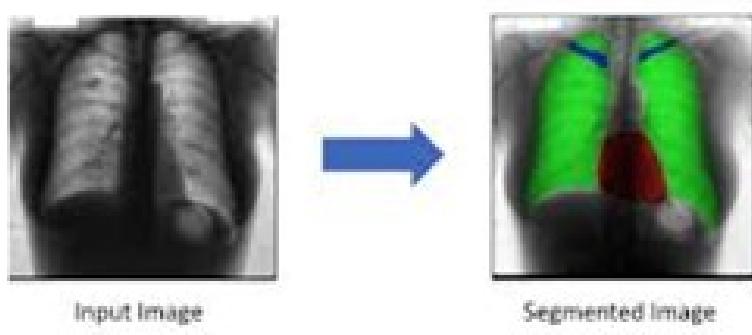
Phân đoạn ngữ nghĩa không chỉ là một bài toán học thuật mà còn có giá trị thực tiễn cao trong cuộc sống và nhiều lĩnh vực, ví dụ:

- **Xe tự hành (autonomous driving):** Semantic Segment cung cấp các thông tin về không gian ở trên đường, Nhận diện chính xác các đối tượng trên đường đi, biển báo, người đi bộ để đưa ra quyết định lái xe an toàn.



Hình 1.13: Sử dụng semantic segmentation cho xe tự hành nhằm phân biệt biển báo, cây cối, đường đi, vỉa hè, xe và người trên lề đường

- **Y tế – Chẩn đoán ảnh y khoa:** Phân vùng các vùng tổn thương trong ảnh MRI, CT, ảnh X-quang,... giúp việc chẩn đoán trở nên nhanh và tiết kiệm thời gian hơn



Hình 1.14: Sử dụng semantic segmentation để phân vùng và nhận dạng các cơ quan trong lồng ngực từ ảnh X-quang và tô màu các cơ quan khác nhau để dễ phân biệt, phổi được tô màu xanh lá cây và tim được tô màu đỏ giúp bác sĩ dễ dàng phân tích hơn

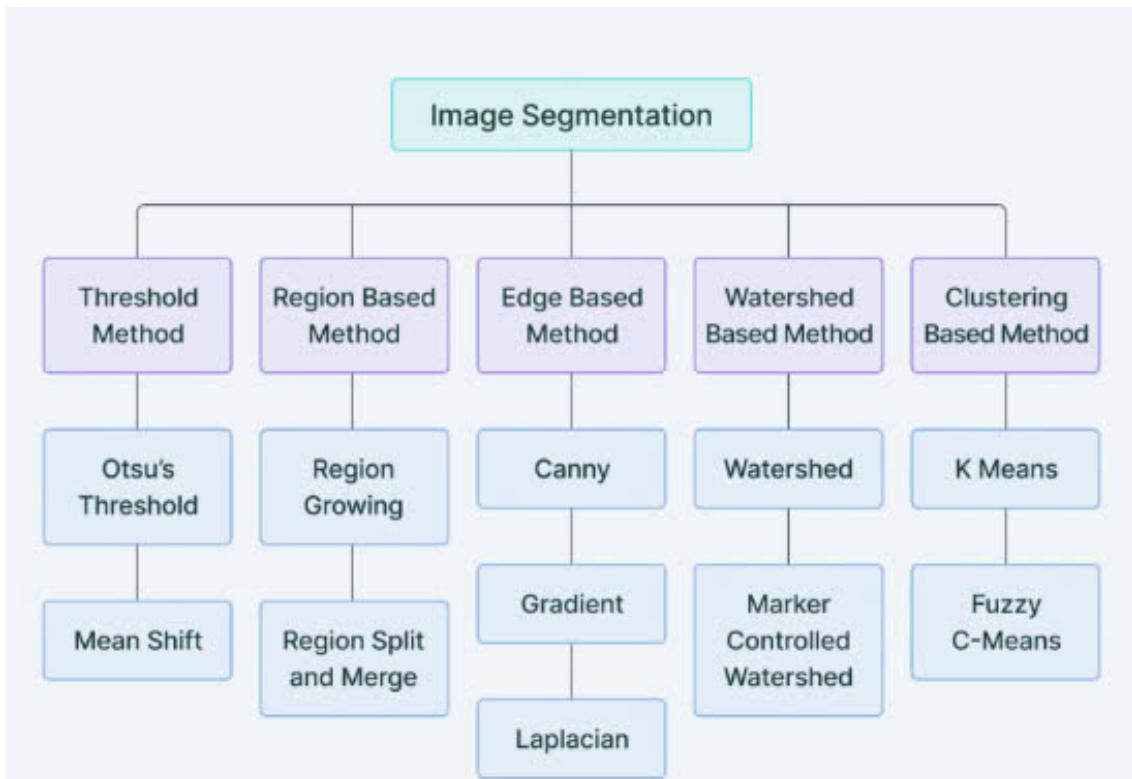
- **Trong nông nghiệp:** Phân tích ảnh chụp từ UAV (drone) để phân biệt cây trồng, cỏ dại, vùng đất khô,... ví dụ cụ thể Drone bay trên cánh đồng dùng Semantic Segmentation để xác định vùng bị sâu bệnh hoặc thiếu nước.
- **Robot công nghiệp và dịch vụ:** Robot trong kho hàng có thể xác định và phân biệt từng hộp sản phẩm trên kệ để lấy đúng mặt hàng.

Chương 2

CÁC THUẬT TOÁN LIÊN QUAN

2.1 GIỚI THIỆU VỀ CÁC THUẬT TOÁN TRUYỀN THÔNG

Phân đoạn hình ảnh bắt nguồn từ Xử lý hình ảnh kỹ thuật số cùng với các thuật toán tối ưu. Phân đoạn hình ảnh ban đầu được phát triển để giải quyết các vấn đề trong xử lý ảnh y tế, viễn thám, nhận dạng đối tượng và nhiều ứng dụng khác. Từ những thuật toán đơn giản như region growing and snakes đến các kỹ thuật phức tạp hơn như active contours và level sets, phân đoạn hình ảnh đã trải qua sự phát triển đáng kể trước khi bước vào kỷ nguyên deep learning với các mạng nơ-ron tích chập (CNN) và các kiến trúc hiện đại khác.

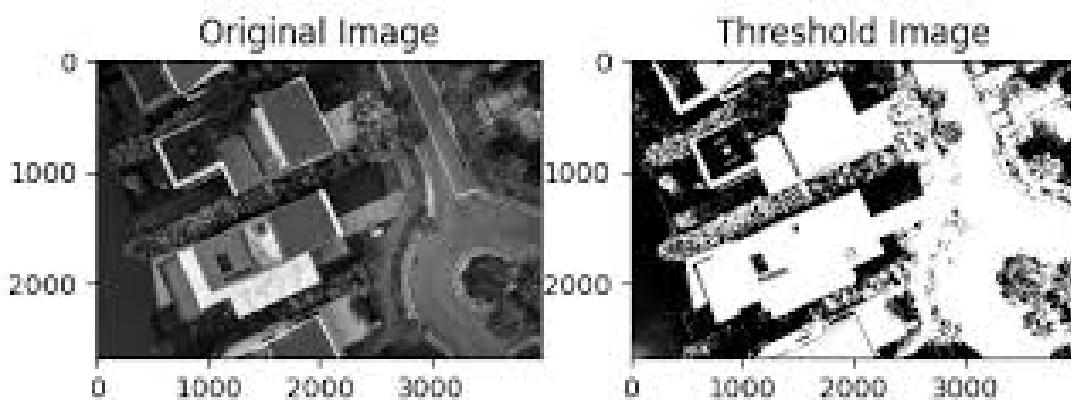


Hình 2.1: Các phương pháp phân đoạn hình ảnh truyền thống

Phân ngưỡng (Thresholding): Là kỹ thuật phân đoạn hình ảnh đơn giản và được sử dụng rộng rãi. Nguyên lý của phương pháp này là so sánh giá trị cường độ của chúng với một giá trị ngưỡng để chia pixel thành 2 lớp. Các pixel có giá trị lớn hơn giá trị ngưỡng thì được gán là 1 và các pixel có giá trị nhỏ hơn giá trị ngưỡng thì được gán là 0

Do đó, phương pháp này chuyển đổi một hình ảnh thang xám thành hình ảnh nhị

phân bìng cách phân loại các điểm ảnh dựa trên giá trị ngưỡng.



Hình 2.2: Minh họa quá trình phân ngưỡng (thresholding) trong xử lý ảnh. Bên trái là ảnh gốc (original image). Bên phải là kết quả sau khi áp dụng phương pháp phân ngưỡng (threshold image), biến đổi ảnh thành dạng nhị phân (binary) với chỉ hai màu đen và trắng.

Phân đoạn dựa trên biên (Edge-based segmentation); Phân đoạn dựa trên biên là phương pháp phân đoạn hình ảnh dựa vào việc phát hiện các đường biên giữa các vùng khác nhau trong ảnh. Nguyên lý cơ bản là xác định các điểm mà tại đó cường độ sáng thay đổi đột ngột, từ đó xây dựng các đường biên để phân chia hình ảnh thành các vùng riêng biệt.

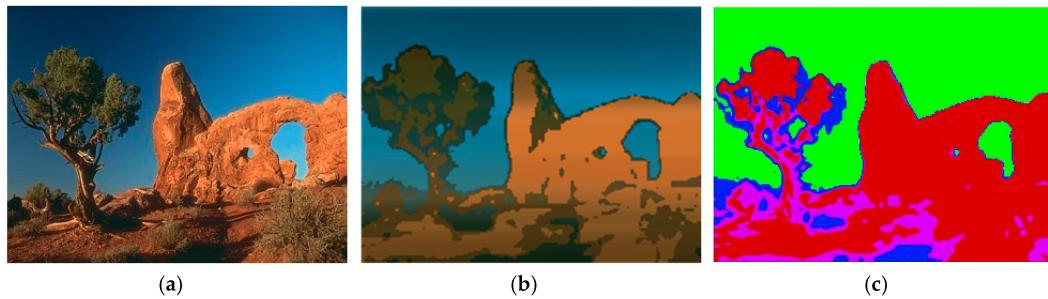
Phát hiện biên thường được thực hiện bằng cách sử dụng các bộ lọc đặc biệt cung cấp cho ta biết các biên của hình ảnh khi tích chập. Các thuật toán phát hiện biên phổ biến: Sobel, Prewitt, Canny.



Hình 2.3: Minh họa cho phương pháp phân đoạn dựa trên biên. Phát hiện biên trích xuất các cạnh hoặc đường viền từ ảnh gốc, giữ lại những thay đổi đột ngột về cường độ sáng trong hình ảnh

Phân đoạn dựa trên vùng (Region-based segmentation): Phân đoạn dựa trên vùng là phương pháp phân chia hình ảnh bằng cách nhóm các điểm ảnh có đặc điểm tương tự thành các vùng có ý nghĩa. Khác với phương pháp dựa trên biên, phương pháp này tập trung vào tính đồng nhất bên trong vùng thay vì sự thay đổi đột ngột tại ranh giới. Thuật toán phân đoạn dựa trên vùng hoạt động theo nguyên tắc, bắt đầu từ một hoặc nhiều điểm hạt giống (seed pixels) sau đó kiểm tra các điểm ảnh lân cận dựa trên tiêu chí tương tự và mở rộng vùng bằng cách thêm các điểm ảnh tương tự, lặp lại quá trình cho đến khi không thể mở rộng thêm. Các phương pháp chính: Region Growing (Phát triển vùng), Region Splitting and Merging (Tách và gộp vùng), Watershed Segmentation.

Phân cụm (Clustering): Phân cụm là phương pháp phân đoạn hình ảnh dựa trên việc phân nhóm các điểm ảnh có đặc điểm tương tự vào cùng một cụm, không phụ thuộc vào vị trí của chúng trong hình ảnh. Các phương pháp này xem xét hình ảnh trong không gian đặc trưng thay vì không gian hình ảnh. Các thuật toán phân cụm chính: K-means Clustering, Fuzzy C-means (FCM), Mean-shift Clustering



Hình 2.4: Minh họa quá trình phân đoạn dựa trên phân vùng sử dụng thuật toán K-means clustering. Hình (c) là kết quả phân cụm K-means với các vùng được phân loại bằng các màu sắc khác biệt rõ ràng (đỏ, xanh lá, tím...) - thể hiện cách thuật toán phân chia hình ảnh thành các cụm khác nhau.

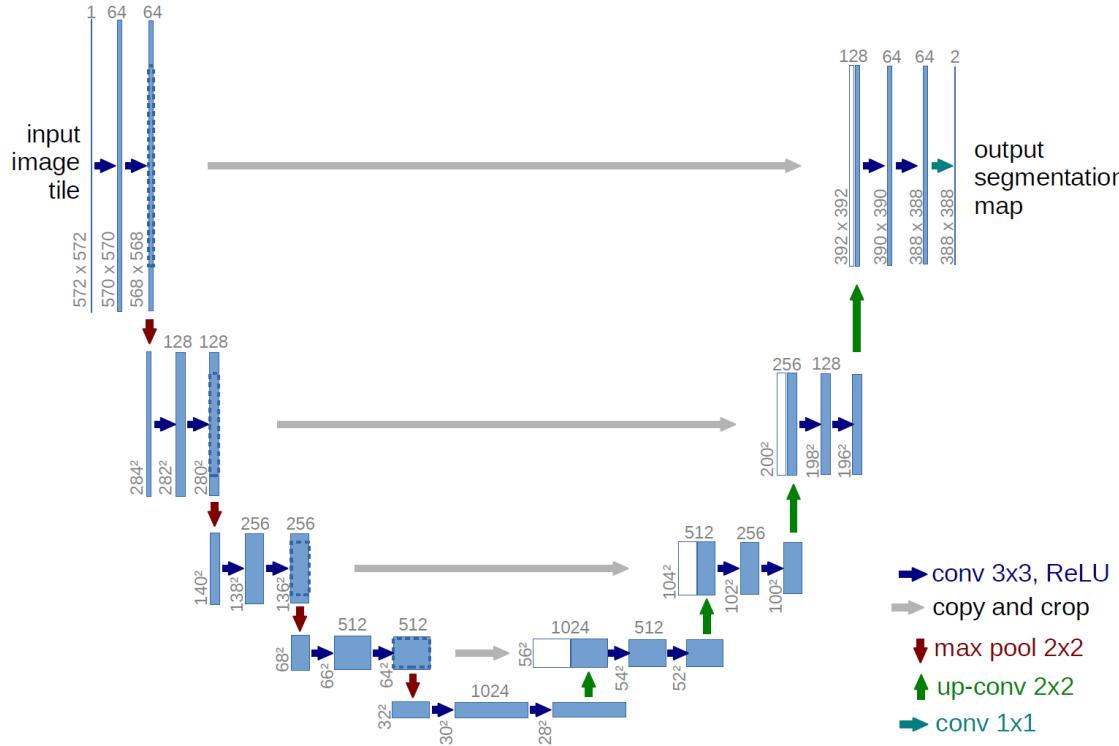
2.2 TỔNG QUANG VỀ MẠNG HỌC SÂU TRONG PHÂN ĐOẠN NGỮ NGHĨA HÌNH ẢNH

2.2.1 U-Net

U-Net là một kiến trúc mạng nơ-ron tích chập (CNN) được giới thiệu lần đầu tiên vào năm 2015 bởi Olaf Ronneberger và các cộng sự, ban đầu được thiết kế để thực hiện phân đoạn ảnh y tế. Tên gọi "U-Net" xuất phát từ hình dạng chữ "U" đặc trưng của kiến trúc mạng, với một nhánh mã hóa (encoder) và một nhánh giải mã (decoder) được kết nối bằng các đường "skip connection".

Skip connections là kết nối đi trực tiếp từ bộ mã hóa đến bộ giải mã mà không đi

qua bottleneck. Điều này giúp giảm mất mát dữ liệu bởi aggressive pooling và down-sampling như được thực hiện trong các khối mã hóa của kiến trúc bộ mã hóa-giải mã.



Hình 2.5: Kiến trúc U-net.

Ứng dụng:

- Phân đoạn ảnh y tế: Ví dụ như tách khối u, mô tế bào, cơ quan nội tạng trong ảnh chụp MRI, CT hoặc ảnh hiển vi [3].
- Phân đoạn ảnh vệ tinh: Sử dụng để phân tích bản đồ vệ tinh, xác định vùng đất, thực vật, hoặc khu dân cư.
- Phân đoạn ảnh nông nghiệp: Nhận diện cây trồng, khu vực bệnh hại hoặc phân tích đất đai từ ảnh flycam hoặc vệ tinh.
- Phân đoạn ảnh giao thông: Phát hiện làn đường, phương tiện, hoặc người đi bộ phục vụ cho xe tự hành.
- Phân đoạn ảnh công nghiệp và sinh học: Kiểm tra chất lượng sản phẩm công nghiệp hoặc nhận diện cấu trúc sinh học trong nghiên cứu.

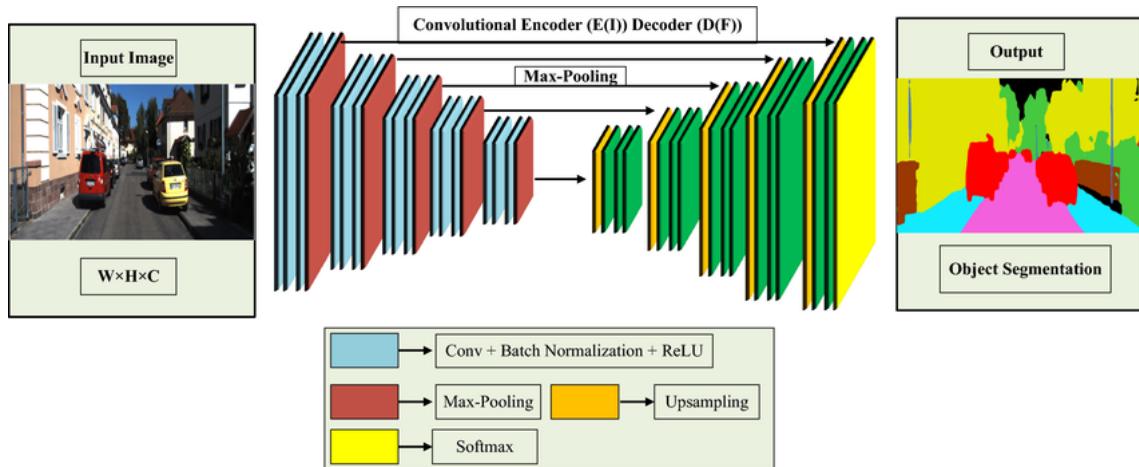
2.2.2 SegNet

SegNet là một kiến trúc học sâu được thiết kế cho phân đoạn ngữ nghĩa, sử dụng bộ mã hóa và giải mã sâu để thực hiện phân đoạn ngữ nghĩa ảnh hiệu quả [4], trong đó

mục tiêu là phân loại từng pixel trong một hình ảnh thành một danh mục được xác định trước. Đây là một mạng nơ-ron mã hóa-giải mã được thiết kế riêng cho phân đoạn hình ảnh theo từng pixel, khiến nó trở nên cực kỳ hiệu quả đối với các tác vụ đòi hỏi phân đoạn hình ảnh chi tiết và chính xác.

SegNet hoạt động bằng cách học cách dán nhãn từng pixel trong hình ảnh dựa trên danh mục tương ứng, cung cấp hiểu biết toàn diện về nội dung của hình ảnh.

SegNet đặc biệt hữu ích trong các ứng dụng như lái xe tự động, phân tích hình ảnh y tế và hiểu rõ bối cảnh đô thị, nơi mà việc phân đoạn chính xác là rất quan trọng.



Hình 2.6: Kiến trúc Segnet.

2.2.3 DeepLab V1/V2/V3/V3+

Theo sau UNet, DeepLab của Facebook đóng vai trò như một cột mốc quan trọng, cung cấp các kết quả tiên tiến nhất về phân đoạn ngữ nghĩa.

DeepLab đã sử dụng các kết cấu phức tạp thay thế các hoạt động tổng hợp đơn giản và ngăn ngừa mất thông tin đáng kể trong khi downsampling. Ngoài ra, DeepLab tiếp tục giới thiệu trích xuất đặc trưng đa tầng (multi-scale feature extraction) với sự trợ giúp của Atrous Spatial Pyramid Pooling để giúp mạng phân đoạn các đối tượng bất kể kích thước của chúng.

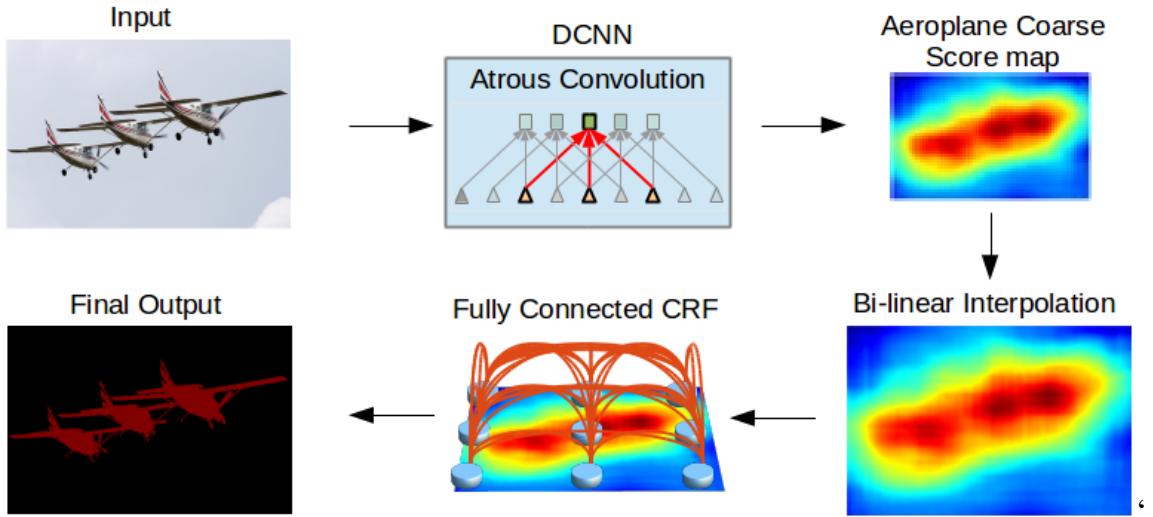
Để khôi phục thông tin về ranh giới, một trong những phần quan trọng nhất của semantic hay instance segmentation, DeepLab đã sử dụng Conditional Random Fields (CRF) được kết nối đầy đủ.

Sự kết hợp giữa độ chính xác khi khoanh vùng chi tiết của CRF và khả năng nhận dạng của CNN đã khiến DeepLab cung cấp bản đồ phân đoạn có độ chính xác cao, đánh bại các phương pháp như FCN và SegNet với biên độ rộng.

Method	mIOU
<i>pre-release version of dataset</i>	
Adelaide_Context [40]	66.4
FCN-8s [14]	65.3
<i>DeepLab-CRF-LargeFOV-StrongWeak [58]</i>	
DeepLab-CRF-LargeFOV [38]	64.8
<i>CRF-RNN [59]</i>	
DPN [62]	59.1
Segnet basic [100]	57.0
Segnet extended [100]	56.1
<i>official version</i>	
Adelaide_Context [40]	71.6
Dilation10 [76]	67.1
DPN [62]	66.8
Pixel-level Encoding [101]	64.3
DeepLab-CRF (ResNet-101)	70.4

Hình 2.7: Kết quả thực nghiệm vượt trội so với các phương pháp khác.

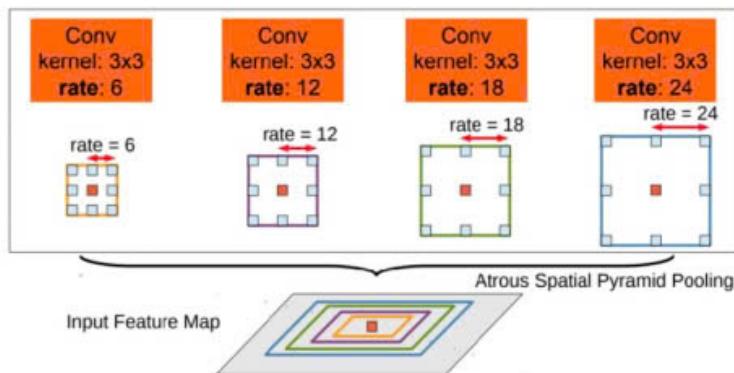
DeepLabV1 là phiên bản đầu tiên của chuỗi mô hình DeepLab, sử dụng VGG- 16 làm backbone và giới thiệu kỹ thuật Atrous Convolution (Dilated Convolution) nhằm mở rộng receptive field mà không làm giảm độ phân giải của feature map. Nhờ đó, mô hình có thể nắm bắt được thông tin không gian rộng hơn mà không cần dùng pooling quá nhiều. Tuy nhiên, để cải thiện khả năng phân đoạn chính xác ở các vùng biên, DeepLabV1 kết hợp thêm Conditional Random Field (CRF) như một bước hậu xử lý. CRF giúp làm mượt các biên và phân đoạn rõ ràng hơn các vật thể trong ảnh. Mặc dù đạt được kết quả tốt, DeepLabV1 vẫn còn hạn chế trong việc tổng hợp ngữ cảnh đa tầm nhìn (multi-scale).



Hình 2.8: Kiến trúc chung DeepLab.

DeepLabV2 kế thừa các ưu điểm từ V1 nhưng cải tiến mạnh mẽ về khả năng học đặc trưng đa tầm nhìn thông qua kỹ thuật Atrous Spatial Pyramid Pooling (ASPP). ASPP sử dụng nhiều nhánh tích chập song song với các tỷ lệ atrous khác nhau (dilated rates), cho phép mô hình nhìn vật thể ở các kích thước khác nhau cùng lúc. DeepLabV2 cũng chuyển sang sử dụng ResNet-101 làm backbone, giúp cải thiện khả năng học sâu và chính xác hơn. CRF vẫn được giữ lại để xử lý hậu kỳ. Nhờ những cải tiến này, DeepLabV2 đạt được hiệu quả cao hơn rõ rệt trên các bộ dữ liệu segmentation như Pascal VOC và Cityscapes.

Atrous Spatial Pyramid Pooling (ASPP)



Hình 2.9: ASPP trong Kiến trúc DeepLabv2.

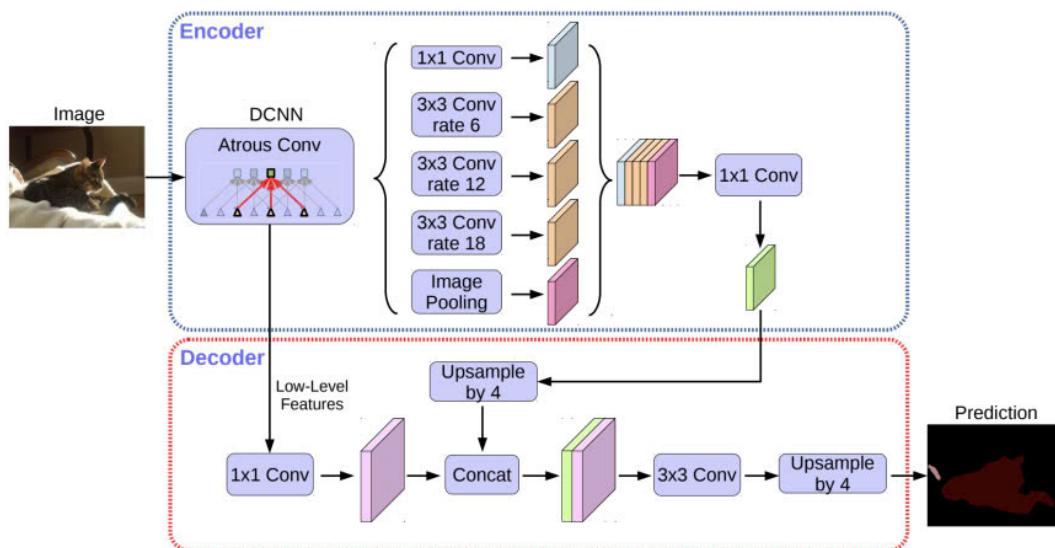
DeepLabV3 đánh dấu một bước tiến vượt bậc bằng cách loại bỏ hoàn toàn CRF,

thay vào đó tăng cường khả năng học của mạng nhờ mở rộng ASPP. ASPP trong V3 không chỉ sử dụng nhiều atrous rate mà còn thêm Global Average Pooling để nắm bắt thông tin ngữ cảnh toàn cục (global context). Các thành phần trong ASPP được chuẩn hóa bằng Batch Normalization, giúp quá trình huấn luyện ổn định hơn. DeepLabV3 sử dụng backbone mạnh như ResNet-101 hoặc Xception, cho phép trích xuất đặc trưng tốt hơn. Nhờ vậy, dù không có bước hậu xử lý CRF, mô hình vẫn đạt được kết quả chính xác cao, tốc độ nhanh hơn và xử lý tốt các đối tượng có hình dạng và kích thước đa dạng.

DeepLabV3+ (2018) là một phiên bản cải tiến của DeepLabV3, mục tiêu chính là cải thiện khả năng phân đoạn chi tiết biên (boundary details). Kiến trúc của DeepLabV3+ kết hợp giữa Encoder (DeepLabV3) [5] và Decoder, giúp đạt được sự kết hợp giữa thông tin ngữ cảnh sâu (từ encoder) và chi tiết biên (từ decoder).

Encoder sử dụng các phương pháp như Atrous Convolution và ASPP để thu thập thông tin từ nhiều tỷ lệ không gian khác nhau, giúp mô hình nhận diện các đối tượng ở nhiều kích thước khác nhau. Decoder sử dụng các đặc trưng từ các tầng nông (low-level) của backbone để giữ lại chi tiết biên và sau đó thực hiện upsampling và kết hợp với các đặc trưng từ encoder. Quá trình này giúp cải thiện độ chính xác của phân đoạn biên mà không cần sử dụng CRF.

Với sự kết hợp này, DeepLabV3+ có thể phân đoạn ảnh chính xác hơn, đặc biệt là trong các ứng dụng cần phân đoạn biên rõ ràng như phân đoạn ảnh y tế, ảnh vệ tinh, hay tự lái. Đồng thời, nó có thể triển khai trên các nền tảng có phần cứng yếu hơn nhờ vào khả năng sử dụng các backbone như MobileNetV2.



Hình 2.10: Kiến trúc DeepLabv3+.

Chương 3

THIẾT KẾ HỆ THỐNG

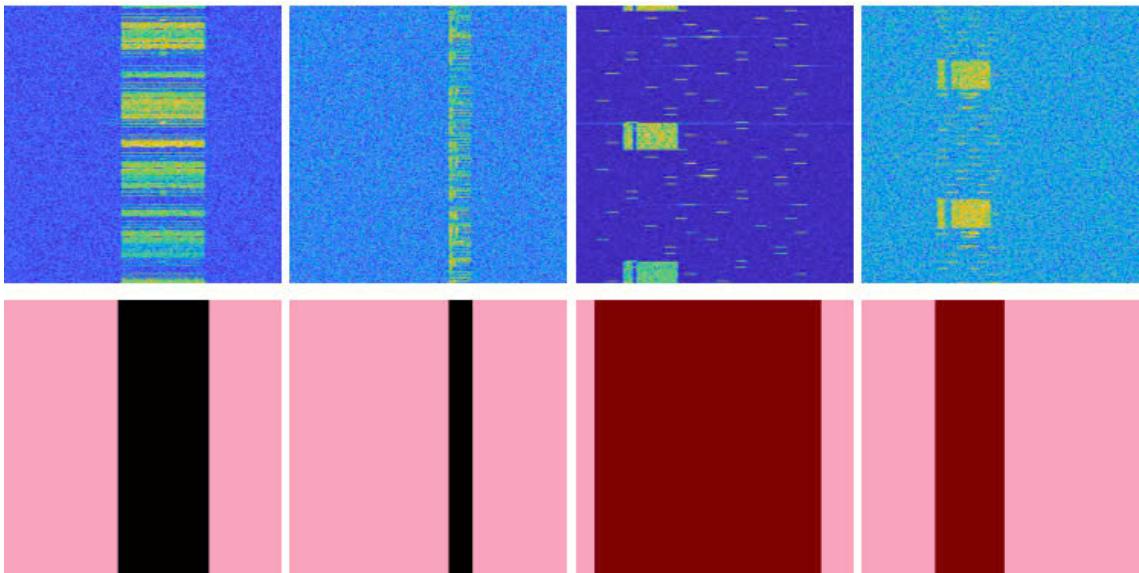
3.1 YÊU CẦU VỀ KIẾN TRÚC MẠNG

Hệ thống đề xuất đề xuất được thiết kế để đảm bảo đáp ứng được các nguyên tắc cụ thể của một hệ thống hoàn chỉnh và tuân thủ các yêu cầu sau:

- Phải tự thiết kế kiến trúc mạng, không được sử dụng các kiến trúc mẫu như U-Net, DeepLabv3+ (trừ khi đã có cải tiến đáng kể).
- Mô hình được cải tiến phải có sự vượt trội so với các mô hình cơ sở. Sử dụng các kỹ thuật tiên tiến về kết nối, tối ưu hóa (như skip connections, attention mechanisms, hoặc multi-scale feature extraction).
- Cần thiết lập các tham số tối ưu như Learning rate, Batch size, Optimizer,..
- Tổng số lượng trọng số huấn luyện của mô hình cải tiến phải < 300.000 để đảm bảo hiệu quả.
- Mô hình phải có khả năng tổng quát hóa tốt, tránh overfitting.
- Hiệu suất của mô hình được đánh giá dựa trên các phép đo về độ chính xác phải có sự cải thiện so với mô hình cơ sở.

3.2 TẬP DỮ LIỆU

Bộ dữ liệu được sử dụng để phân loại các pixel trong báo cáo lần này là bộ dữ liệu về tín hiệu quang phổ. Dữ liệu bao gồm các ảnh spectrogram thu được từ tín hiệu của các công nghệ mạng khác nhau, tiêu biểu là LTE (Long Term Evolution – mạng 4G) và NR (New Radio – mạng 5G). Mỗi ảnh biểu diễn sự thay đổi của phổ tần theo thời gian, giúp làm nổi bật đặc trưng riêng của từng loại tín hiệu bao gồm 6000 ảnh input đầu vào với kích thước $3 \times 256 \times 256$ và 6000 ảnh label tương ứng được phân thành 3 loại tín hiệu ứng với màu hồng, đen và đỏ. Bộ dữ liệu này phù hợp để huấn luyện các mô hình học sâu nhằm phát hiện, nhận dạng và phân loại tín hiệu không dây trong các ứng dụng như giám sát phổ tần, phân tích môi trường vô tuyến, và phát triển các hệ thống truyền thông thông minh.



Hình 3.1: Bộ dữ liệu SpectrogramSignal.

3.3 TRIỂN KHAI THIẾT KẾ KIẾN TRÚC MẠNG

3.3.1 Tổng quan về mô hình

Dựa trên yêu cầu của đề tài, nhóm đã xây dựng một mô hình mạng nơ-ron tích chập (CNN) cải tiến, hướng tới bài toán **semantic segmentation** dựa trên kiến trúc **Encoder-Decoder (Unet)** với các cải tiến tự thiết kế, nhằm đảm bảo hiệu quả cao đồng thời tối ưu hóa số lượng tham số để vẫn tuân thủ yêu cầu tổng số lượng tham số nhỏ hơn 300000.

Kiến trúc của hệ thống được thiết kế theo dạng **Encoder-Decoder** với các cải tiến cụ thể nhằm tăng khả năng trích xuất và tổng hợp đặc trưng ở nhiều cấp độ không gian. Cụ thể, mô hình được triển khai như sau:

Input

- Ảnh đầu vào có kích thước ($\mathbf{H} \times \mathbf{W} \times 3$).

Encoder: Giai đoạn mã hóa có nhiệm vụ trích xuất đặc trưng từ ảnh đầu vào.

- Đầu tiên, ảnh đầu vào (RGB) được đưa qua một khối tích chập đơn giản (Conv2d(3×3) + BatchNorm + ReLU) để trích xuất đặc trưng ban đầu thành 16 kênh.
- Sau đó, tại các tầng sâu hơn, mô hình sử dụng khối **Multi-Scale Convolution Block** với 3 nhánh (Conv2d(3×3), Conv2d(5×5), và Conv2d(7×7)), sau đó được nối lại theo chiều kênh nhằm trích xuất đặc trưng ở nhiều kích thước khác nhau (3×3 , 5×5 , 7×7), tăng khả năng nhận diện đối tượng ở nhiều tỷ lệ.

- Ở tầng tiếp theo, mạng tích hợp thêm một **Spatial Attention Block** giúp làm nổi bật các vùng quan trọng trong không gian feature map.

Bottleneck:

- Sau quá trình downsampling, các đặc trưng được xử lý sâu hơn thông qua một khối Multi-Scale Convolution thứ hai (không có Spatial Attention) để tiếp tục trích xuất đặc trưng đa tỷ lệ, tăng cường khả năng học ngữ cảnh sâu hơn giúp tăng cường khả năng nắm bắt ngữ cảnh rộng.

Decoder:

- Các feature map sau bottleneck được **Upsample** đặc trưng (Upsampling $\times 2$) để tăng kích thước chiều không gian đảm bảo khôi phục kích thước không gian về kích thước ảnh ban đầu sau khi trích xuất đặc trưng ở các Layer trước.
- Đồng thời thực hiện **Skip Connections** từ các tầng Encoder tương ứng được thực hiện bằng **phép nối (concatenation)**, giúp mạng giữ lại thông tin chi tiết từ các tầng nồng.
- Sau mỗi phép nối, các feature map được tinh chỉnh qua Conv2d + BatchNorm + ReLU.
- Lặp lại quá trình để đưa kích thước trở lại như ban đầu.

Output:

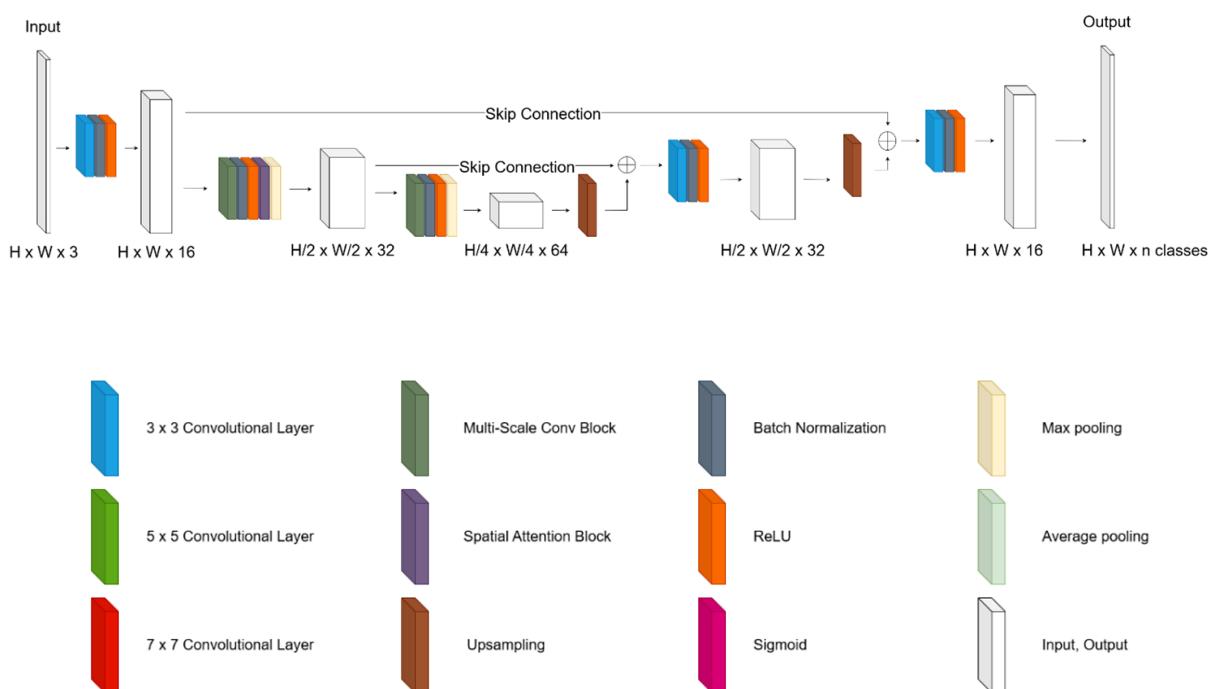
- Đầu ra của mạng sử dụng một lớp Conv2d(kernel_size=1) để đưa số lượng kênh về đúng với số lượng lớp cần phân đoạn (n_classes).
- Output có kích thước ($H \times W \times n_classes$), phù hợp với yêu cầu bài toán semantic segmentation (phân đoạn từng pixel).

Mục tiêu thiết kế của các thành phần trong mô hình được mô tả trong bảng 3.1 :

Để trực quan hóa quá trình xử lý, Hình 3.2 dưới đây mô tả pipeline kiến trúc của mô hình mạng CNN cải tiến. Mô hình bao gồm ba giai đoạn chính: giai đoạn mã hóa (Encoder) với các khối tích chập đa tỷ lệ (Multi-Scale Convolution Block) kết hợp attention theo không gian (Spatial Attention), giai đoạn giải mã (Decoder) sử dụng skip connections để khôi phục kích thước không gian, và cuối cùng là lớp đầu ra (Output Layer) đưa về số lượng lớp phân đoạn mong muốn.

Thành phần	Mục tiêu
Multi-Scale Convolution Block	Trích xuất đặc trưng ở nhiều tỷ lệ không gian (fine-scale + coarse-scale)
Spatial Attention Block	Tập trung vào các vùng quan trọng trong ảnh
Skip Connection	Kết hợp thông tin chi tiết từ các tầng nông với đặc trưng sâu để giữ chi tiết hình ảnh trong quá trình giải mã
Upsampling	Khôi phục kích thước không gian về kích thước ảnh ban đầu
Mạng nhẹ	Số lượng tham số tối ưu hóa dưới 300K để phù hợp yêu cầu bài toán

Bảng 3.1: Các thành phần chính và mục tiêu trong kiến trúc mô hình



Hình 3.2: Pipeline kiến trúc mô hình CNN cải tiến với Multi-Scale Feature Extraction và Spatial Attention. Mô hình sử dụng kết hợp giữa các lớp tích chập đa tỉ lệ (kernel $3 \times 3, 5 \times 5, 7 \times 7$) và cơ chế attention theo không gian nhằm tăng khả năng trích xuất đặc trưng, đồng thời duy trì chi tiết qua các skip connections trong quá trình giải mã.

Tóm tắt các thành phần trong sơ đồ:

Thành phần	Vai trò chính
Conv2d + BatchNorm + ReLU	Trích xuất và chuẩn hóa đặc trưng
Multi-Scale Conv Block	Trích xuất đa tỷ lệ (multi-scale feature extraction)
Spatial Attention Block	Làm nổi bật vùng quan trọng trong không gian
Max Pooling	Downsampling (giảm kích thước không gian)
Upsampling	Khôi phục kích thước không gian trong decoder
Skip Connection	Kết hợp đặc trưng nông (chi tiết) với đặc trưng sâu (ngữ nghĩa)
Conv2d(1×1)	Chuyển sang số lượng lớp phân đoạn

Bảng 3.2: Vai trò chính của các thành phần trong kiến trúc mạng



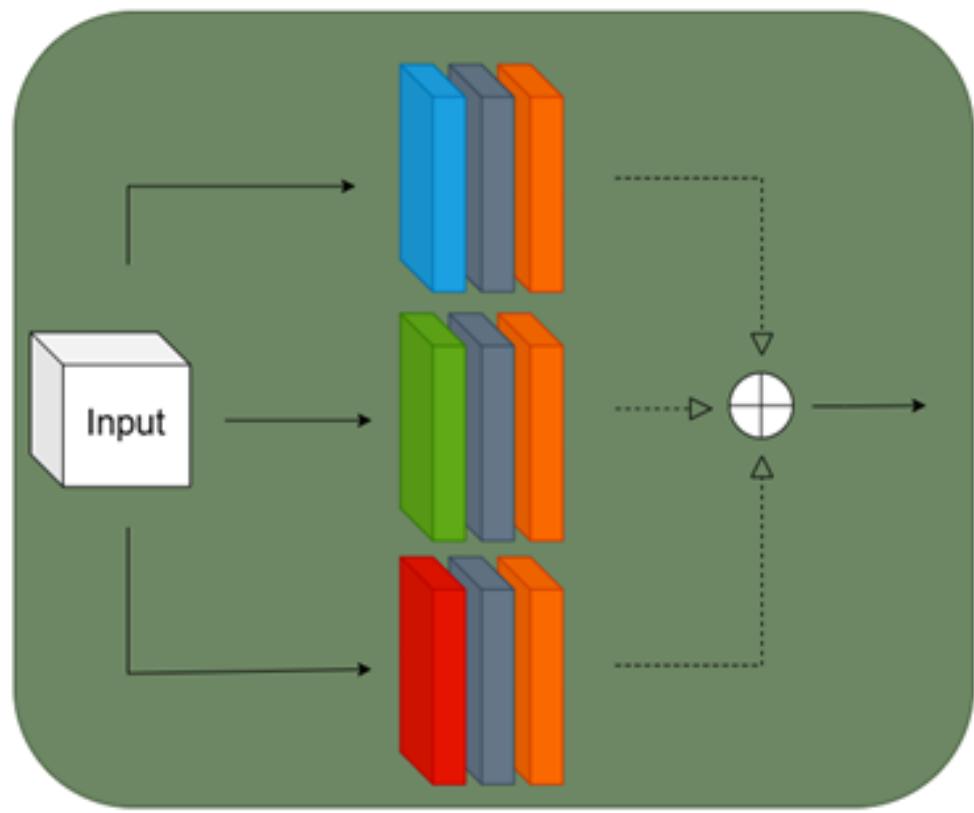
ENCODER

DECODER

Hình 3.3: Sơ đồ khái minh họa các lớp mạng trong mô hình.

3.3.2 Multi-Scale Convolution

Nhằm cải thiện khả năng trích xuất đặc trưng ở nhiều cấp độ khác nhau, nhóm đã thiết kế và tích hợp một khối Multi-Scale Convolution Block (MSCB) vào kiến trúc mô hình. Khối này cho phép mô hình học được các đặc trưng từ nhiều receptive field khác nhau, giúp mô hình nhận diện tốt cả các chi tiết nhỏ và các cấu trúc lớn trong ảnh. Hình 3.4 dưới đây minh họa cấu trúc tổng quát của một khối Multi-Scale Convolution:



Hình 3.4: Sơ đồ khái niệm khối Multi-Scale Conv Block.

Mô tả chi tiết hoạt động của khối Multi-Scale Convolution:

- **Input:** Đầu vào là một tensor đặc trưng có chiều (C, H, W) , trong đó C là số kênh, H là chiều cao và W là chiều rộng của đặc trưng.
- **Nhánh đa tỉ lệ:** Tensor đầu vào được đưa qua ba nhánh song song, mỗi nhánh sử dụng các bộ lọc với kích thước kernel khác nhau (trong mô hình nhóm sử dụng

ba bộ lọc lần lượt là 3×3 , 5×5 , và 7×7) nhằm thu nhận thông tin ở nhiều tỷ lệ không gian khác nhau, giúp mô hình nắm bắt được các đặc trưng có kích thước khác nhau trong dữ liệu.

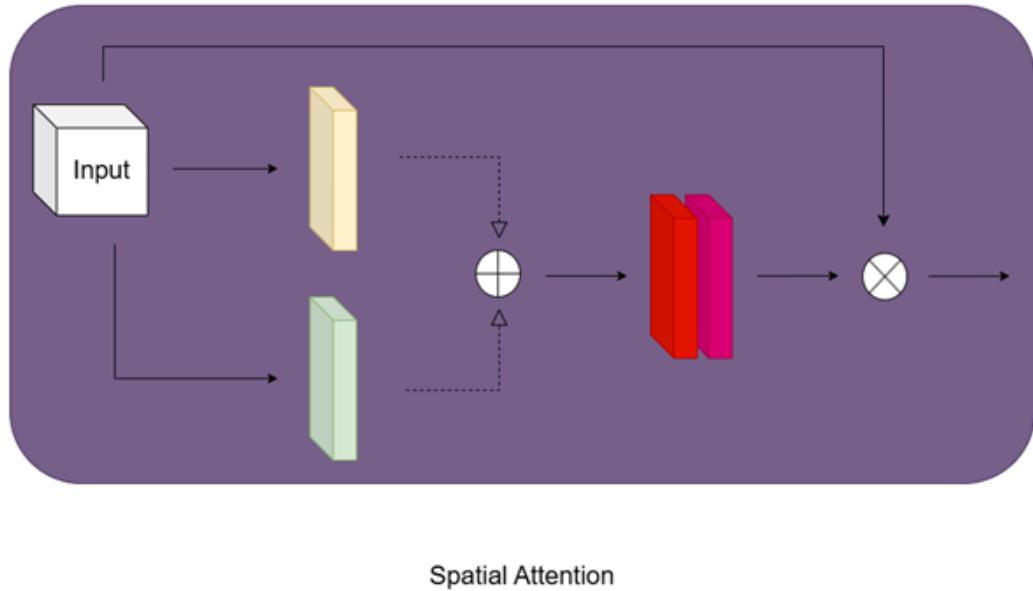
- **Tích chập tuần tự:** Bên trong mỗi nhánh, có thể có nhiều lớp tích chập được xếp chồng lên nhau. Sau mỗi lớp tích chập thường sẽ có một lớp *Batch Normalization* để ổn định quá trình huấn luyện và một hàm kích hoạt phi tuyến (trong mô hình, nhóm sử dụng hàm ReLU) nhằm tăng khả năng biểu diễn của mô hình.
- **Kết hợp đặc trưng:** Các đặc trưng đầu ra từ ba nhánh song song sẽ được kết hợp lại. Phép kết hợp này có thể được thực hiện thông qua phép cộng các tensor tương ứng hoặc bằng cách nối các kênh của chúng lại (*concatenation*). Kết quả của phép kết hợp sẽ tiếp tục được đưa qua các tầng xử lý tiếp theo trong mô hình.

Tác dụng của khối trong mô hình:

- **Khả năng trích xuất đặc trưng mạnh mẽ hơn so với các block tích chập đơn lẻ:** Bằng cách xử lý thông tin ở nhiều tỷ lệ không gian khác nhau, khối này có khả năng nắm bắt được các đặc trưng phức tạp và đa dạng hơn so với việc chỉ sử dụng các lớp tích chập đơn lẻ với một kích thước kernel cố định.
- **Hỗ trợ mô hình học được thông tin đa dạng, từ cả chi tiết nhỏ (*fine detail*) và cấu trúc lớn (*global context*):** Việc sử dụng các nhánh với kích thước kernel khác nhau cho phép mô hình đồng thời học được các chi tiết nhỏ thông qua các kernel nhỏ và các cấu trúc lớn thông qua các kernel lớn hơn.
- **Hiệu quả trong bài toán phân đoạn ảnh:** Kiến trúc này rất hữu ích trong các bài toán phân đoạn ảnh, nơi các đối tượng cần phân loại có thể xuất hiện với nhiều kích thước và hình dạng khác nhau trong ảnh. Khả năng thu nhận thông tin đa tỷ lệ giúp mô hình đưa ra dự đoán chính xác hơn trên các đối tượng này.

3.3.3 Spatial Attention

Bên cạnh việc sử dụng Multi-Scale Convolution Block thì để cải thiện khả năng mô hình tập trung vào các vùng không quan trọng trong ảnh, nhóm đã tích hợp Spatial Attention Module vào kiến trúc mô hình. Dưới đây là hình 3.5 minh họa cho cấu trúc khối Spatial Attention



Hình 3.5: Spatial Attention

Như minh họa ở Hình 3.5, Spatial Attention hoạt động qua các bước sau:

1. **Xử lý input feature map:** Đầu vào của quá trình là một *feature map*. Feature map này sẽ trải qua hai phép toán tổng hợp không gian khác nhau, cụ thể là:
 - **Average Pooling:** Tính giá trị trung bình của các phần tử theo chiều kênh.
 - **Max Pooling:** Lấy giá trị lớn nhất của các phần tử theo chiều kênh.
 Hai phép toán pooling này giúp thu thập thông tin về sự phân bố của các đặc trưng trên không gian.
2. **Tạo attention map:** Kết quả của hai phép pooling ở bước trên sẽ được nối lại với nhau (*concatenate*). Tensor kết hợp này sau đó được đưa qua một khối gồm các lớp tích chập (*Convolution*) để học bản đồ trọng số không gian (*spatial attention map*). Bản đồ này thể hiện mức độ quan trọng của từng vị trí không gian trong feature map đầu vào.
3. **Khuếch đại đặc trưng không gian:** Attention map thu được sẽ được nhân điểm (*element-wise multiplication*) với *input feature map* để làm nổi bật (tăng trọng số) các vùng không gian quan trọng được xác định bởi attention map và làm mờ (giảm trọng số) các vùng không liên quan.

Vai trò của Spatial Attention trong mô hình:

- **Tăng cường tập trung vào vùng đặc trưng:** Cơ chế này cho phép mô hình tập trung mạnh hơn vào các vùng đặc trưng quan trọng trong ảnh, ví dụ như ranh giới giữa các vật thể hoặc các chi tiết nhỏ thường bị bỏ sót.
- **Giảm nhiễu và tăng độ rõ nét cho các feature map, giúp cải thiện kết quả phân đoạn:** Bằng cách giảm sự chú ý đến các vùng không liên quan, Spatial Attention giúp giảm thiểu tác động của nhiễu và làm tăng độ rõ nét cho các feature map. Điều này dẫn đến việc các tầng xử lý tiếp theo trong mô hình nhận được thông tin chất lượng hơn, từ đó cải thiện kết quả phân đoạn cuối cùng.

Chương 4

KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ

4.1 MÔI TRƯỜNG THỰC NGHIỆM

Quá trình huấn luyện và đánh giá mô hình được thực hiện trên nền tảng **Kaggle Notebook** với cấu hình phần cứng và phần mềm như sau:

Phần cứng:

- GPU: NVIDIA Tesla T4 × 2
- RAM: 13 GB
- Disk: 20 GB

Phần mềm:

- Python 3.8
- PyTorch 2.0
- CUDA 11.8

Thông số huấn luyện:

- Batch size: 32
- Learning rate: 0.001
- Optimizer: Adam
- Loss function: CrossEntropyLoss
- Epochs: 40
- Tổng số parameters: 108005
- Dataset: spectrogramsignal
- Số lượng ảnh đầu vào: 6000 ảnh

- Số lượng nhãn (label images): 6000 ảnh
- Kích thước chia tập:
 - Train set: 4800 ảnh (80%)
 - Validation set: 1200 ảnh (20%)

Mô hình được huấn luyện và lưu dưới định dạng TorchScript (.pt) để đảm bảo khả năng inference linh hoạt sau huấn luyện.

4.2 CÁC TIÊU CHÍ ĐÁNH GIÁ

Để đánh giá hiệu quả của mô hình, nhóm sử dụng các tiêu chí sau:

- **Loss Function:** Sử dụng hàm mất mát CrossEntropyLoss, phổ biến trong các bài toán phân đoạn đa lớp.
- **Mean Accuracy (mAcc):** Trung bình độ chính xác trên từng lớp (pixel-level classification accuracy).
- **Mean Intersection over Union (mIoU):** Mean IoU được tính bằng cách trung bình IoU của tất cả các lớp. Cụ thể, được tính theo công thức:

$$IoU = \frac{TP}{TP + FP + FN} \quad (4.1)$$

với TP (True Positive), FP (False Positive) và FN (False Negative) được tính trên từng lớp.

Lý do chọn IoU: Vì bài toán phân đoạn yêu cầu đánh giá **sự trùng khớp vùng dự đoán** với vùng thực tế, nên mIoU phản ánh chất lượng phân đoạn tốt hơn so với accuracy thông thường.

4.3 KẾT QUẢ HUẤN LUYỆN

Trong suốt quá trình huấn luyện 40 epoch, mô hình cho thấy sự tiến bộ rõ rệt qua từng epoch. Biểu đồ loss và mIoU trên tập huấn luyện và validation cho thấy quá trình tối ưu diễn ra ổn định, không xảy ra overfitting.

Mô hình tốt nhất được lưu tại epoch 39, với các kết quả:

- Validation Loss: **0.0398**

Epoch	Train Loss	Train mAcc	Train mIoU	Val Loss	Val mAcc	Val mIoU
1	0.4775	0.8548	0.7224	0.2190	0.9161	0.8720
5	0.0872	0.9576	0.9278	0.0757	0.9556	0.9339
10	0.0621	0.9684	0.9458	0.0560	0.9721	0.9517
20	0.0484	0.9750	0.9572	0.0445	0.9766	0.9603
30	0.0409	0.9787	0.9633	0.0409	0.9803	0.9632
40	0.0366	0.9812	0.9673	0.0370	0.9793	0.9665

Bảng 4.1: Kết quả huấn luyện và đánh giá mô hình qua các epoch

- Validation Mean Accuracy: **98.29%**
- Validation Mean IoU: **96.34%**

Mô hình đạt hiệu suất rất cao, đặc biệt là chỉ số mIoU > 96 %, cho thấy khả năng phân đoạn vùng chính xác cao trên bộ dữ liệu thử nghiệm.

4.4 ĐÁNH GIÁ KẾT QUẢ

Để trực quan hóa và đánh giá chi tiết hiệu quả của mô hình, nhóm thực hiện so sánh giữa:

- Ảnh đầu vào (Input Image)
- Nhãn thực tế (Ground Truth)
- Kết quả phân đoạn từ mô hình (Predicted Mask)

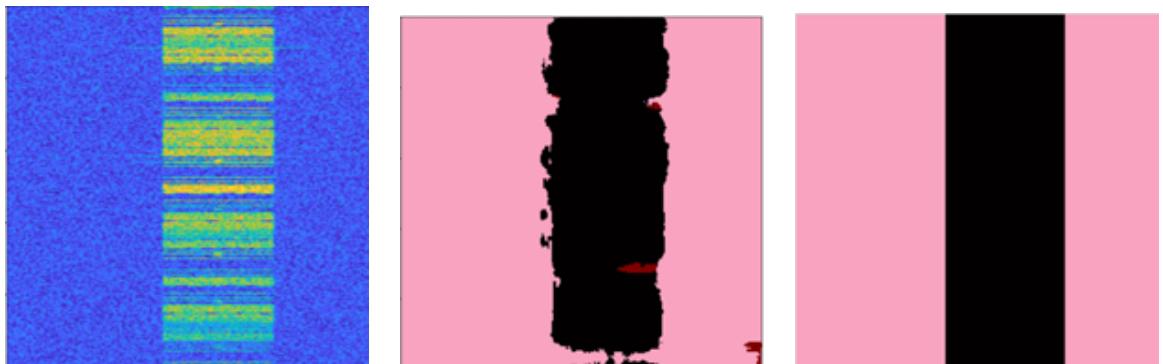
Một số kết quả phân đoạn trên tập validation được trích xuất và thể hiện dưới đây:



(a) Input Image

(b) Ground Truth

(c) Predicted Mask



(a) Input Image



(b) Ground Truth



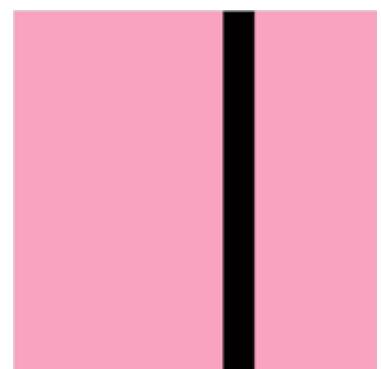
(c) Predicted Mask



(a) Input Image



(b) Ground Truth



(c) Predicted Mask

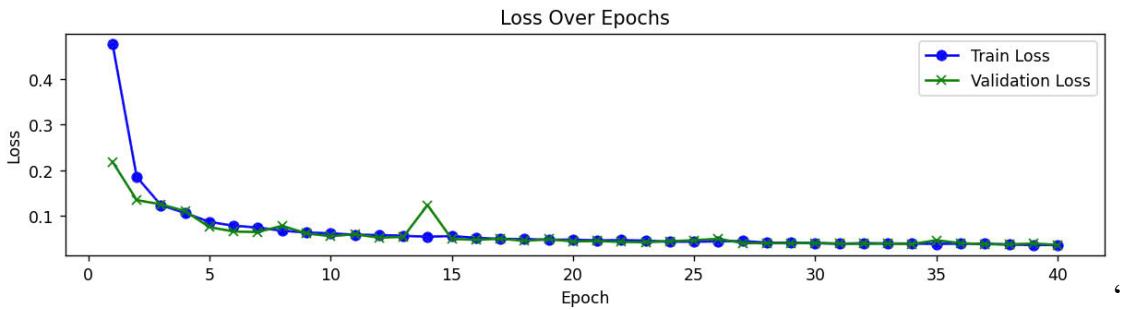
Hình 4.3: Kết quả huấn luyện

Nhận xét:

- Kết quả phân đoạn cho thấy mô hình đã nhận diện chính xác ranh giới và hình dạng của các đối tượng trong ảnh.
- Các khu vực trọng tâm của tín hiệu được phân đoạn đầy đủ, chi tiết, với độ chính xác cao so với nhãn thực tế.

Để đánh giá hiệu quả của mô hình trong quá trình huấn luyện, nhóm đã theo dõi và phân tích ba chỉ số chính bao gồm: Hàm mất mát (Loss), Độ chính xác trung bình (Mean Accuracy) và Chỉ số giao nhau trên hợp (Mean IoU) cho cả tập huấn luyện và tập kiểm tra (validation). Những chỉ số này đóng vai trò quan trọng trong việc đánh giá khả năng hội tụ và độ chính xác của mô hình.

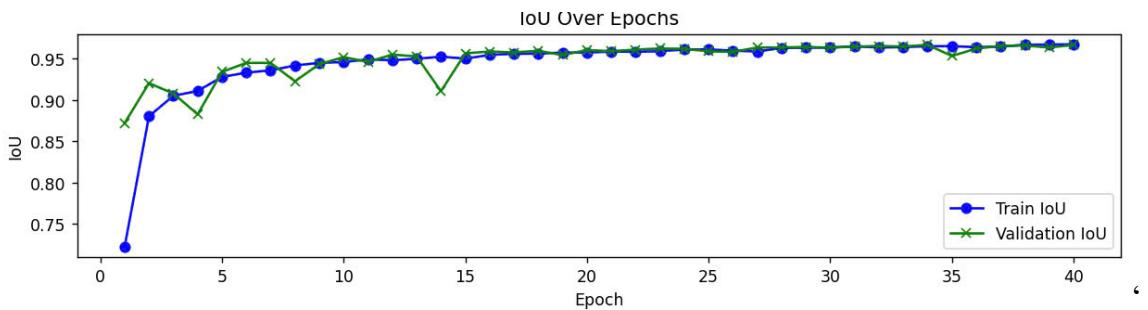
Kết quả được minh họa thông qua ba biểu đồ sau, giúp chúng ta dễ dàng nhận diện sự thay đổi của các chỉ số này qua từng epoch, từ đó rút ra những nhận xét về tiến độ huấn luyện và tình trạng của mô hình. Việc trực quan hóa dữ liệu sẽ giúp nhóm hiểu rõ hơn về hiệu suất của mô hình, đặc biệt là trong việc phát hiện các vấn đề như overfitting hoặc underfitting.



Hình 4.4: Biểu đồ hàm mất mát(Loss).

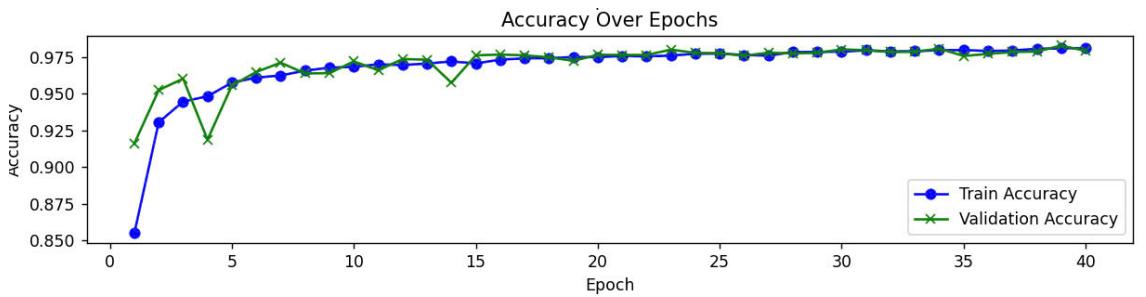
Biểu đồ hàm mất mát (Loss): Train Loss giảm dần qua các epochs từ 0.4775 xuống 0.0366. Điều này cho thấy mô hình đang học tốt và tiến bộ trong việc tối ưu hóa loss trên tập huấn luyện.

Validation Loss cũng giảm từ 0.2190 xuống 0.0370, cho thấy mô hình không chỉ học tốt trên tập huấn luyện mà còn có khả năng tổng quát tốt với tập kiểm tra. Sự giảm nhẹ ở Validation Loss trong những epochs sau cùng có thể cho thấy mô hình đang đạt tới ngưỡng tối ưu và không còn giảm mạnh nữa.



Hình 4.5: Biểu đồ chỉ số IoU (Mean IoU).

Biểu đồ chỉ số IoU (Mean IoU): Train IoU tăng từ 0.7224 lên 0.9673, cho thấy mô hình ngày càng cải thiện khả năng phân vùng chính xác. Validation IoU cũng tăng từ 0.8720 lên 0.9665, điều này cho thấy mô hình không chỉ cải thiện trên dữ liệu huấn luyện mà còn duy trì hiệu suất cao khi đối mặt với tập kiểm tra. Đặc biệt, IoU có sự cải thiện ổn định, chứng tỏ mô hình đang học cách phân vùng và phân loại chính xác hơn qua thời gian.



Hình 4.6: Biểu đồ độ chính xác trung bình (Mean Accuracy)

Biểu đồ độ chính xác trung bình (Mean Accuracy): Train Accuracy tăng từ 85.48% lên 98.12%. Điều này cho thấy mô hình đang học tốt và cải thiện độ chính xác qua các epochs. Validation Accuracy cũng tăng từ 91.61% lên 97.93%, nhưng với sự gia tăng này có một sự dao động nhẹ ở giữa quá trình huấn luyện. Dù vậy, cuối cùng mô hình vẫn đạt được độ chính xác cao trên tập kiểm tra. Nhìn chung, Validation Accuracy và Train Accuracy đều đạt được mức cao, cho thấy mô hình có khả năng tổng quát tốt và không gặp phải hiện tượng overfitting nghiêm trọng.

Nhận xét tổng quan:

Hiệu năng tăng đều:

- Từ Epoch 1 đến Epoch 10, Mean IoU trên tập validation tăng đều từ ~ 0.72 lên ~ 0.95 , cho thấy mô hình dần học được các đặc trưng quan trọng.
- Validation Loss giảm mạnh từ ~ 0.22 xuống ~ 0.05 , cho thấy mô hình đang dần tối ưu và cải thiện khả năng phân loại.

Khả năng tổng quát hóa tốt:

- Độ chính xác (Accuracy) và Mean IoU trên tập validation luôn cao hơn hoặc tương đương với tập huấn luyện, chứng tỏ mô hình có khả năng khái quát tốt và không bị overfitting.
- Mặc dù độ chính xác và IoU trên tập huấn luyện đều tăng mạnh, nhưng độ chính xác và IoU trên tập kiểm tra cũng tăng tương ứng mà không có sự tăng đột biến ở cuối, cho thấy mô hình không bị overfit.
- Các chỉ số bắt đầu ổn định từ Epoch 10 trở đi, nhưng vẫn có những cải thiện nhẹ đến Epoch 18–19, cho thấy mô hình đang hội tụ nhưng vẫn tiếp tục cải thiện.

Dao động nhỏ tại Epoch 14:

- Tại Epoch 14, có sự dao động bất thường: Validation Loss tăng đột ngột và IoU giảm. Nguyên nhân có thể là:
 - Dữ liệu bị nhiễu khiến mô hình mất ổn định.
 - Batch shuffle gây ra phân phối không đồng đều của mẫu.
 - Biến động ngẫu nhiên trong quá trình huấn luyện.

Tối ưu và hiệu quả:

- Dù chỉ có tổng số tham số là 108K (rất thấp), mô hình vẫn đạt Mean IoU > 0.95, một kết quả rất ấn tượng.
- Mô hình hiện tại đã được tối ưu tốt về cả độ chính xác lẫn cấu trúc, phù hợp triển khai trên hệ thống có tài nguyên hạn chế.
- Cả Train Loss, Validation Loss, Train Accuracy, Validation Accuracy, Train IoU và Validation IoU đều cải thiện liên tục qua các epoch, cho thấy mô hình học và tổng quát hóa tốt.

4.5 NHẬN XÉT VÀ KẾT LUẬN

Qua quá trình huấn luyện và thử nghiệm, mô hình mạng nơ-ron tích chập cải tiến mà nhóm xây dựng đã cho thấy hiệu quả rất tốt trong bài toán phân đoạn ngữ nghĩa ảnh.

Các kết quả nổi bật đạt được:

- Mô hình đạt Mean Accuracy (mAcc) lên tới 98.29% và Mean IoU (mIoU) đạt 96.34% trên tập validation, chứng tỏ khả năng phân đoạn chính xác cao.
- Loss trên tập validation giảm ổn định qua các epoch, cho thấy quá trình huấn luyện diễn ra mượt mà, không có dấu hiệu overfitting hay underfitting.
- Tổng số tham số mô hình chỉ khoảng 108.005 parameters, đảm bảo yêu cầu nhẹ, nhanh và phù hợp với các bài toán yêu cầu tối ưu tài nguyên.

Ưu điểm của hệ thống:

- Sử dụng Multi-Scale Feature Extraction giúp mô hình nắm bắt đặc trưng ở nhiều cấp độ không gian, từ chi tiết nhỏ đến cấu trúc lớn.
- Tích hợp Spatial Attention giúp mô hình tập trung vào các vùng thông tin quan trọng, cải thiện chất lượng phân đoạn.

- Thiết kế Encoder-Decoder đơn giản nhưng hiệu quả, dễ dàng mở rộng trong tương lai.
- Huấn luyện ổn định trên nền tảng GPU phổ biến (Tesla T4) với tốc độ nhanh.

Hạn chế và hướng phát triển:

- Mặc dù hiệu quả cao, mô hình có thể bị suy giảm nhẹ độ chính xác khi gặp ảnh có nhiều nhiễu hoặc vùng ranh giới phức tạp.
- Hệ thống hiện tại mới chỉ sử dụng loss đơn thuần (CrossEntropyLoss), chưa áp dụng thêm các kỹ thuật loss phức tạp (như Dice Loss, Focal Loss) để tối ưu hơn cho segmentation.
- Trong tương lai, có thể mở rộng thêm:
 - Áp dụng **data augmentation** mạnh mẽ hơn để tăng độ bền mô hình.
 - Thủ nghiệm với các module attention nâng cao như **CBAM**, **Dual Attention** để cải thiện khả năng học sâu hơn.
 - Áp dụng kỹ thuật **semi-supervised learning** nếu số lượng nhãn hạn chế.

TÀI LIỆU THAM KHẢO

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 3431–3440.
- [3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.
- [4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *arXiv preprint arXiv:1802.02611*, 2018.