

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
---📖---



PHẠM THANH TÙNG

PHẠM THANH TÙNG

NGHIÊN CỨU XÂY DỰNG BẢN ĐỒ SỐ SỬ
DỤNG PHƯƠNG PHÁP HỌC SÂU VÀ THIẾT KẾ
QUỶ ĐẠO CHO ROBOT DI ĐỘNG

CƠ ĐIỆN TỬ

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ CƠ ĐIỆN TỬ

2024

Hà Nội - 2025

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



PHẠM THANH TÙNG

**NGHIÊN CỨU XÂY DỰNG BẢN ĐỒ SỐ SỬ
DỤNG PHƯƠNG PHÁP HỌC SÂU VÀ THIẾT KẾ
QUỸ ĐẠO CHO ROBOT DI ĐỘNG**

Ngành: Cơ điện tử
Mã số: 8520114

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ CƠ ĐIỆN TỬ

**NGƯỜI HƯỚNG DẪN:
PGS.TS NGUYỄN ANH TÚ**

Hà Nội - 2025

LỜI CAM ĐOAN

Tôi tên là Phạm Thanh Tùng, học viên cao học khóa 12 chuyên ngành Kỹ thuật Cơ điện tử. Sau gần 2 năm học tập và nghiên cứu tại Trường Đại học Công nghiệp Hà Nội, được sự giúp đỡ của các thầy, cô giáo đặc biệt là PGS.TS Nguyễn Anh Tú, tôi đã hoàn thành đề án tốt nghiệp thạc sĩ. Tôi xin cam đoan rằng đề tài ***“Nghiên cứu xây dựng bản đồ số sử dụng phương pháp học sâu và thiết kế quỹ đạo chuyển động”*** là kết quả nghiên cứu của cá nhân tôi dưới sự hướng dẫn của PGS.TS Nguyễn Anh Tú. Toàn bộ nội dung, dữ liệu và kết quả trong đề án này là do tôi thực hiện và không sao chép công trình của cá nhân khác dưới bất kỳ hình thức nào.

Trong quá trình thực hiện đề tài, tôi luôn tuân thủ các quy định về đạo đức nghiên cứu và tôn trọng quyền sở hữu trí tuệ. Các tài liệu tham khảo, các nguồn thông tin được sử dụng trong đề án này đều được trích dẫn rõ ràng và đầy đủ.

Tôi xin hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của các nội dung, dữ liệu và kết quả nghiên cứu trong đề án này.

Hà Nội, ngày ... tháng ... năm 2025
Người thực hiện

Phạm Thanh Tùng

LỜI CẢM ƠN

Để hoàn thành được đề án “*Nghiên cứu xây dựng bản đồ số dữ dụng phương pháp học sâu và thiết kế quỹ đạo cho robot di động*”, tôi xin được gửi lời cảm ơn chân thành và sâu sắc tới PGS.TS Nguyễn Anh Tú - người đã tận tình hướng dẫn, chỉ bảo và hỗ trợ tôi trong suốt quá trình thực hiện đề tài.

Qua đây, tôi cũng xin được gửi lời cảm ơn đến các thầy, cô giáo trong Trường Cơ khí - Ô tô, Trung tâm Đào tạo Sau đại học - Trường Đại học Công nghiệp Hà Nội đã truyền đạt kiến thức, kỹ năng, luôn sẵn sàng hỗ trợ và tạo mọi điều kiện thuận lợi cho tôi trong suốt quá trình học tập, nghiên cứu tại trường.

Cuối cùng, tôi xin gửi lời cảm ơn đến gia đình, bạn bè, người thân, các anh chị đồng nghiệp đã luôn quan tâm, động viên, hỗ trợ tôi trong suốt quá trình học tập và thực hiện đề tài tạo động lực cho tôi tiếp tục học tập và phấn đấu trong tương lai.

Hà Nội, ngày ... tháng ... năm 2025

Người thực hiện

Phạm Thanh Tùng

MỤC LỤC

LỜI CAM ĐOAN.....	i
LỜI CẢM ƠN	ii
MỤC LỤC.....	iii
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT.....	v
DANH MỤC BẢNG BIỂU	vi
DANH MỤC HÌNH ẢNH	vii
MỞ ĐẦU	1
CHƯƠNG 1 - TỔNG QUAN VỀ PHƯƠNG PHÁP THIẾT KẾ BẢN ĐỒ VÀ HOẠCH ĐỊNH ĐƯỜNG DẪN CHO ROBOT DI ĐỘNG	5
1.1. Tổng quan về robot di động	5
1.2. Tổng quan về phương pháp thiết kế bản đồ	8
1.2.1. Các loại bản đồ	9
1.2.2. Phương pháp truyền thống.....	11
1.2.3. Phương pháp hiện đại	13
1.2.4. Một số nghiên cứu về thiết kế bản đồ cho robot di động.....	13
1.3. Tổng quan về phương pháp hoạch định đường dẫn (Path Planning).....	14
1.3.1. Hoạch định đường dẫn.....	14
1.3.2. Các thuật toán hoạch định đường dẫn cho robot di động	16
1.4. Phạm vi nghiên cứu.....	21
1.5. Phương pháp nghiên cứu.....	21
Kết luận chương I.....	21
CHƯƠNG 2 - MÔ HÌNH HỌC SÂU VÀ CÁC THUẬT TOÁN THIẾT KẾ QUỸ ĐẠO	22
2.1. Tổng quan chung về học máy (Machine Learning)	22
2.1.1. Khái niệm	22
2.1.2. Các phương pháp học máy.....	22
2.1.3. Các ứng dụng của học máy.....	26
2.2. Học sâu (Deep Learning)	27
2.2.1. Khái niệm học sâu	27
2.2.2. Mạng nơ – ron.....	28
2.2.3. Nguyên lý của học sâu.....	33
2.2.4. Image Segmentation - Thuật toán phân đoạn ảnh	34

2.3. Phương pháp thiết kế quỹ đạo	46
2.3.1. Thiết kế quỹ đạo với Dijkstra	46
2.3.2. Thiết kế quỹ đạo với A^*	49
Kết luận chương II	52
CHƯƠNG 3 - XÂY DỰNG MÔ HÌNH HỌC SÂU VÀ THIẾT KẾ QUỸ ĐẠO CHUYỂN ĐỘNG CHO ROBOT	53
3.1. Quy trình thực hiện	53
3.2. Lựa chọn mô hình học sâu	55
3.3. Các bước huấn luyện mô hình học sâu	58
Kết luận chương III	62
CHƯƠNG 4 - KẾT QUẢ VÀ ĐỀ XUẤT	63
4.1. Kết quả huấn luyện và đánh giá mô hình	63
4.2. Kết quả so sánh giữa các mô hình	67
4.3. Kết quả dự đoán và trích xuất bản đồ đường đi trên tập dữ liệu kiểm tra ...	69
4.4. Kết quả hoạch định đường đi sử dụng các thuật toán tìm kiếm	73
Kết luận chương 4	76
KẾT LUẬN VÀ ĐỀ XUẤT	77
TÀI LIỆU THAM KHẢO	79

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

TT	Ký hiệu, chữ viết tắt	Nguyên nghĩa
1	CNN	Convolutional Neural Network (Mạng nơ-ron tích chập)
2	GPS	Global Positioning System (Hệ thống định vị toàn cầu)
3	SIAM	Simultaneous Localization and Mapping (Định vị và lập bản đồ đồng thời)
4	ROS	Robot Operating System (Hệ điều hành Robot)
5	GIS	Geographic Information System (Hệ thống thông tin địa lý)
6	DQN	Deep Q-Network (Mạng học tăng cường kết hợp Q-learning và Deep Learning)
7	DNN	Deep Neural Network (Mạng nơ-ron sâu)
8	RNN	Recurrent Neural Network (Mạng nơ-ron hồi tiếp)
9	CNN	Convolutional Neural Network (Mạng nơ-ron tích chập)
10	FCN	Fully Convolutional Network (Mạng tích chập hoàn toàn)
11	FPN	Feature Pyramid Network (Mạng kim tự tháp đặc trưng)
12	LiDAR	Light Detection and Ranging (Cảm biến quét laser)
13	RL	Reinforcement Learning (Học tăng cường)
14	DL	Deep Learning (Học sâu)
15	ML	Machine Learning (Học máy)
16	AI	Artificial Intelligence (Trí tuệ nhân tạo)
17	ACO	Ant Colony Optimization (Tối ưu bầy kiến)
18	GA	Genetic Algorithm (Thuật toán di truyền)
19	RRT	Rapidly-exploring Random Tree (Cây khám phá ngẫu nhiên nhanh)

DANH MỤC BẢNG BIỂU

Bảng 3.1: Bảng phân tích các mô hình học sâu trong xây dựng mô hình phân đoạn bản đồ đường đi cho robot di động	57
Bảng 4.1: So sánh giá trị chỉ số giữa các mô hình mạng	69
Bảng 4.2: So sánh ưu và nhược điểm giữa các mô hình.....	72
Bảng 4.3: Số liệu kết quả tìm kiếm đường đi	74
Bảng 4.4: Đánh giá hiệu suất trung bình của các thuật toán đối với 3 kịch bản. 76	

DANH MỤC HÌNH ẢNH

Hình 1.1: Robot Shakey do SRI International phát triển [4]	5
Hình 1.2: Sojourner robot thám hiểm Sao Hỏa đầu tiên được giới thiệu vào năm 1997 [5].....	6
Hình 1.3: Quá trình phát triển của robot di động [6]	7
Hình 1.4: Thiết lập tạo bản đồ cho robot lau nhà sử dụng điện thoại thông minh [8]	8
Hình 1.5: Ví dụ về các loại bản đồ [6]	9
Hình 1.6: Ứng dụng giải thuật Dijkstra tìm đường đi ngắn nhất.....	17
Hình 1.7: A* và các biến thể của thuật toán A* [6].....	18
Hình 1.8: Mô hình cây ngẫu nhiên RRT [23]	20
Hình 2.1: Ví dụ về học máy có giám sát.....	23
Hình 2.2: Phân cụm các điểm dữ liệu [33].....	25
Hình 2.3: Các ứng dụng của học máy [35]	26
Hình 2.4: Mối quan hệ giữa học sâu, học máy và trí tuệ nhân tạo [36].....	27
Hình 2.5: Ví dụ về mạng thần kinh nhân tạo (ANNs)	28
Hình 2.6: Mô hình mạng nơ-ron sâu [36]	29
Hình 2.7: Cấu trúc của một mạng nơ - ron tích chập [38]	30
Hình 2.8: Sơ đồ của một RNN [40]	32
Hình 2.9: Kiến trúc mạng RNN, LSTM và GRU [41].....	33
Hình 2.10: Vật thể được xác định trong bài toán nhận diện vật thể và bài toán phân đoạn ảnh [42].....	34
Hình 2.11: Ví dụ về phân đoạn ảnh sử dụng mạng tích chập hoàn toàn (FCN) [43]	35
Hình 2.12: Kiến trúc một mô hình U-net [42]	36
Hình 2.13: Bộ mã hóa và bộ giải mã của U-Net [44]	38
Hình 2.14: Một ví dụ về phân đoạn sử dụng Mask R-CNN [45].....	39
Hình 2.15: Kiến trúc của mạng Faster R-CNN [46]	39
Hình 2.16: Kiến trúc mạng Mask R-CNN [46].....	40
Hình 2.17: So sánh 2 phương pháp phát hiện vật thể thông thường với FPN [48]	41
Hình 2.18: FPN cho phân đoạn đối tượng [49].....	42
Hình 2.19: Mô hình chung của Deeplab [51].....	43

Hình 2.20: Sự khác biệt giữa môi trường tiếp nhận trong đầu vào của Standard Convolution và Atrous Convolution [53].....	43
Hình 2.21: Mô hình mạng DeepLab v3+ [55]	45
Hình 2.22: Tiến trình của thuật toán Dijkstra [56].....	47
Hình 2.23: Lưu đồ thuật toán Dijkstra	48
Hình 2.24: Tìm kiếm đường đi của giải thuật A* [57]	49
Hình 2.25: Lưu đồ thuật toán A*	51
Hình 3.1: Quy trình thực hiện các nội dung nghiên cứu.....	54
Hình 3.2: Quy trình thực hiện huấn luyện mô hình	58
Hình 3.3: Một hình ảnh vệ tinh trong 1 khu vực tại TP. Hồ Chí Minh	59
Hình 4.1: Kết quả các chỉ số huấn luyện và đánh giá của mô hình DeepLabV3+	63
Hình 4.2: Kết quả các chỉ số huấn luyện mô hình Unet	65
Hình 4.3: Kết quả các chỉ số huấn luyện mô hình FPN	66
Hình 4.4: Kết quả so sánh chỉ số hàm mất mát giữa các mô hình.....	68
Hình 4.5: Kết quả so sánh chỉ số hàm hiệu suất giữa các mô hình.....	68
Hình 4.6: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình DeepLabV3+.....	70
Hình 4.7: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình Unet	71
Hình 4.8: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình FPN	72
Hình 4.9: Mô hình bản đồ ô lưới để thực hiện hoạch định đường dẫn.....	73
Hình 4.10: Kịch bản thứ nhất (Path 1)	74
Hình 4.11: Kịch bản thứ 2 (Path 2)	74
Hình 4.12: Kịch bản thứ 3 (Path 3)	75

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, với sự phát triển của khoa học công nghệ trong tất cả các lĩnh vực của đời sống thì robot di động cũng là một trong những lĩnh vực đang được nghiên cứu và ứng dụng mạnh. Robot di động là robot độc lập, được thiết kế để di chuyển tự chủ trong một môi trường không xác định và thực hiện một số nhiệm vụ cụ thể [1], điển hình có thể kể đến các ứng dụng phổ biến của robot di động như: trong công nghiệp robot di động được sử dụng trong dây chuyền sản xuất tự động, giúp tăng cường hiệu suất và giảm chi phí sản xuất, trong lĩnh vực y tế robot di động có thể được sử dụng trong việc chăm sóc bệnh nhân, vận chuyển hàng hóa y tế, và thậm chí thực hiện các ca phẫu thuật tự động, trong lĩnh vực dịch vụ robot di động có thể được sử dụng để giao hàng tự động, dọn dẹp nhà cửa, và cung cấp dịch vụ khách hàng thông qua trí tuệ nhân tạo,... Để hoàn thành nhiệm vụ được giao, robot cần được tích hợp các giải thuật thông minh để giải quyết các bài toán một cách chủ động và một hệ thống định vị tự động chính xác.

Để robot di động có thể di chuyển được và thực hiện các nhiệm vụ thì chúng cần phải thu thập thông tin, phân tích, lập bản đồ về môi trường và sau đó đưa ra các quyết định trong một môi trường chưa được biết trước. Việc thay đổi và thích ứng với các sự thay đổi của môi trường chính là đặc trưng cho khả năng linh hoạt của robot. Chính vì thế ngày nay rất nhiều phương pháp đã được nghiên cứu và ứng dụng để robot di động có thể kể đến như: GPS (hệ thống định vị toàn cầu) sử dụng tín hiệu từ vệ tinh để xác định vị trí tương đối của robot trên bề mặt Trái Đất, cảm biến gia tốc và cảm biến góc quay để theo dõi chuyển động và hướng di chuyển của robot, bản đồ số (SLAM) sử dụng cảm biến và thuật toán để xác định vị trí của robot và vẽ bản đồ của môi trường xung quanh, cảm biến hình ảnh và máy quét sử dụng máy ảnh hoặc máy quét để thu thập dữ liệu về môi trường và sử dụng các thuật toán xử lý hình ảnh để định vị robot, cảm biến siêu âm và LIDAR sử dụng sóng siêu âm hoặc sóng laser để đo khoảng cách và tạo bản đồ của môi trường xung quanh...

Bên cạnh đó, với sự phát triển các kỹ thuật về Trí thông minh nhân tạo (Artificial Intelligence) cùng với sự hỗ trợ của các giải thuật học máy giúp robot có khả năng tự học, các phương pháp khám phá dựa trên học tập được xem như chìa khóa cho việc giải quyết các vấn đề trên. Trí thông minh nhân tạo giúp robot có khả năng tư duy và có thể thực hiện các nhiệm vụ khác nhau tương tự như con

người. Sự phát triển các thuật toán học máy (Machine Learning) giúp robot thực hiện các nhiệm vụ khác nhau mà không phải thực hiện lập trình lại. Giải pháp này giúp cho bài toán định vị và thiết lập quỹ đạo chuyển động robot linh hoạt, chính xác và tin cậy hơn. Trong nghiên cứu của Krizhevsky và cộng sự [2] đã áp dụng mạng nơ-ron phức hợp sâu (CNN) và thiết kế một mạng phân loại hiệu quả cho dữ liệu quy mô lớn, điều này đã thúc đẩy một làn sóng nghiên cứu về mạng nơ-ron. Ngoài ra, CNN cũng đã tạo ra nhiều đột phá lớn trong lĩnh vực nhận dạng hình ảnh áp dụng trong robot [3]. Trong khi đó phương pháp học sâu (Deep Learning) hiện đang được coi là một trong những phương pháp tiếp cận mạnh mẽ để cải thiện khả năng tự học và tự điều chỉnh của robot. Bằng cách sử dụng mạng nơ-ron sâu, robot có thể học từ dữ liệu cảm biến và trải nghiệm để tự động điều chỉnh hành vi và quyết định trong môi trường thực tế. Điều này giúp robot có thể thích ứng với các tình huống mới, tối ưu hóa hiệu suất và đáp ứng linh hoạt với môi trường xung quanh. Các ứng dụng của phương pháp học sâu trong robot di động bao gồm điều hướng tự động, tránh vật cản và tương tác với con người. Các thuật toán kết hợp với học sâu (DL) và RL đã trở thành một trong những công cụ hiệu quả trong việc ra quyết định thực hiện các nhiệm vụ phức tạp của robot. Các phương pháp Q-learning và khả năng suy luận mạnh mẽ của CNN, Deep Q-network (DQN), phương pháp học DRL cũng đã cho thấy tiềm năng trong ứng dụng để giải quyết nhiều bài toán khác nhau trong lĩnh vực robot như phát hiện và tránh vật cản, thiết kế quỹ đạo, điều khiển chuyển động, nhận diện và phân loại các đối tượng.

Bởi vậy, học viên đề xuất chọn đề tài ***“Nghiên cứu xây dựng bản đồ số sử dụng phương pháp học sâu và thiết kế quỹ đạo cho robot di động”*** làm đề án tốt nghiệp Thạc sĩ.

2. Tính cấp thiết của đề tài

Công nghệ robot được đánh giá là một trong những công cụ phục vụ đắc lực cho cuộc cách mạng 4.0, robot đã được Bộ Khoa học và Công nghệ đưa vào danh mục công nghệ cao, công nghệ chủ chốt của công nghiệp 4.0. Quyết tâm này cũng cho thấy sự kỳ vọng vào một bước đột phá trong nghiên cứu làm chủ và phát triển công nghệ robot cùng nhiều lĩnh vực công nghệ liên quan của Việt Nam. Tuy nhiên, nghiên cứu và ứng dụng robot ở Việt Nam vẫn còn nhiều hạn chế. Robot được chế tạo tại Việt Nam còn rất ít và hầu hết sử dụng công nghệ cũ của thế giới, chưa có đủ khả năng làm chủ công nghệ cũng như phát triển công nghệ phù hợp.

Năm 2022, khi đại dịch Covid-19 bùng phát, robot di động đã được triển khai trong nhiều ứng dụng để hỗ trợ các hoạt động y tế, giảm nguy cơ lây nhiễm và duy trì được các hoạt động của xã hội. Từ thực tế cho thấy việc sử dụng robot di động khi dịch bệnh xảy ra là hết sức cấp thiết để hạn chế tối đa tiếp xúc tránh lây lan dịch bệnh tuy nhiên một số robot ở Việt Nam hiện nay đang được các đơn vị nghiên cứu chế tạo với các giải pháp, công nghệ kinh điển như bám đường theo vạch sơn, sử dụng cột mốc,... nên hoạt động của robot di động còn hạn chế, không linh hoạt.

Cùng với sự phát triển của công nghệ, các ứng dụng của robot di động đã bắt đầu xuất hiện trong nhiều lĩnh vực khác nhau, việc xây dựng bản đồ số sử dụng phương pháp học sâu và thiết kế quỹ đạo cho robot di động mang lại nhiều hiệu quả, giúp robot di động hoạt động hiệu quả và an toàn trong các môi trường phức tạp, cải thiện được khả năng định vị và điều hướng, cải thiện tính linh hoạt và thích nghi khi môi trường có sự thay đổi đột ngột. Với việc áp dụng công nghệ học sâu (deep learning), robot di động có thể tự động xử lý dữ liệu từ các cảm biến và hình ảnh để tạo ra bản đồ số chính xác của môi trường xung quanh. Điều này có thể giúp robot di động trong việc di chuyển an toàn và hiệu quả trong môi trường phức tạp, như các khu vực đô thị hay môi trường tự nhiên. Với việc nghiên cứu và áp dụng các công nghệ mới như học sâu và thiết kế quỹ đạo cho robot di động, Việt Nam có cơ hội phát triển các ứng dụng robot di động tiên tiến, từ đó nâng cao năng lực cạnh tranh và ứng dụng công nghệ trong nhiều lĩnh vực khác nhau.

Ở Việt Nam, trong những năm qua cũng có nhiều công trình khoa học nghiên cứu về robot di động đã được các tác giả thực hiện và công bố. Tuy nhiên các nghiên cứu tập trung vào các loại robot hoạt động trong nhà xưởng sản xuất, thực hiện các nhiệm vụ cụ thể theo một quy trình công nghệ xác định. Khi cần thay đổi nhiệm vụ cần phải thực hiện lập trình lại cho robot hoặc thay đổi kết cấu phần cứng.

Ngoài ra, đất nước ta trong giai đoạn hiện nay là một nước đang phát triển, khiến nhu cầu sử dụng robot công nghiệp tăng nhanh. Đa phần các công ty, nhà máy, tập đoàn lớn đã ứng dụng robot di động vào công tác sản xuất để nâng cao hiệu quả công việc của mình nhưng chủ yếu là các robot nhập khẩu, công nghệ đóng kín, chỉ cho phép người dùng thực hiện những thao tác vận hành cơ bản. Khi cần thay đổi nhiệm vụ công nghệ, quỹ đạo chuyển động phải cần có sự can thiệp từ chính hãng sản xuất. Còn lại các doanh nghiệp vừa, nhỏ và siêu nhỏ chiếm tỷ lệ 97% ở Việt Nam vẫn chưa ứng dụng robot vào trong hoạt động sản xuất của

mình do nguồn kinh phí đầu tư lớn. Điều đó cho thấy thị trường robot tại Việt Nam còn rất rộng lớn, cấp thiết cần thúc đẩy quá trình nghiên cứu, chế tạo robot ở trong nước để giảm giá thành đầu tư ban đầu.

Vì vậy, việc thực hiện đề tài giúp xây dựng cơ sở lý thuyết cho việc xây dựng bản đồ số cho robot di động là một trong những tiền đề cho việc làm chủ được công nghệ, giảm chi phí đầu tư ban đầu và làm chủ được công nghệ đảm bảo chi phí vận hành cũng là nhỏ nhất.

3. Mục tiêu nghiên cứu

Mục tiêu chung: Tìm hiểu và mở rộng nghiên cứu về các công nghệ tiên tiến, giải pháp thông minh ứng dụng trong robot; định hướng làm chủ công nghệ trong chế tạo robot đặc biệt là robot di động phục vụ các lĩnh vực trong đời sống.

Mục tiêu cụ thể:

- Nghiên cứu, ứng dụng phương pháp học sâu cho bài toán trích xuất bản đồ từ ảnh vệ tinh;
- Thiết kế quỹ đạo chuyển động cho robot di động hoạt động trong môi trường xác định;
- Làm chủ công nghệ nền tảng để phát triển các loại robot di động thích nghi và làm việc trong các môi trường thay đổi.

CHƯƠNG 1 - TỔNG QUAN VỀ PHƯƠNG PHÁP THIẾT KẾ BẢN ĐỒ VÀ HOẠCH ĐỊNH ĐƯỜNG DẪN CHO ROBOT DI ĐỘNG

1.1. Tổng quan về robot di động

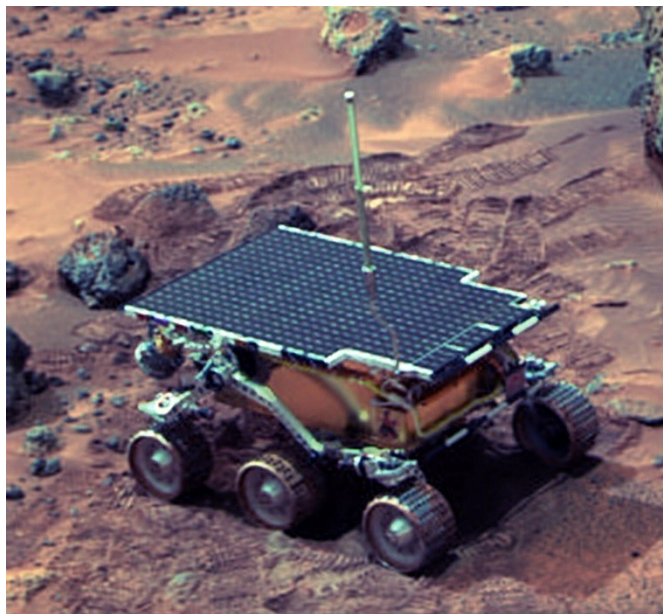
Robot học (Robotics) là một lĩnh vực kỹ thuật và khoa học máy tính, tập trung vào việc thiết kế, chế tạo và vận hành robot. Mục tiêu chính là phát triển những cỗ máy thông minh có khả năng hỗ trợ con người trong nhiều lĩnh vực. Robot có nhiều hình dạng và chức năng khác nhau, một số loại được thiết kế giống con người, trong khi số khác tồn tại dưới dạng ứng dụng tự động hóa, thực hiện các nhiệm vụ lặp lại một cách chính xác, có khả năng thực hiện nhiệm vụ nhanh chóng và hiệu quả với rất ít hoặc không cần sự can thiệp của con người. Trong những năm qua, lĩnh vực robot đã đạt được những bước tiến vượt bậc và hiện nay được xem là một trong các công cụ thúc đẩy đổi mới và chuyển đổi kỹ thuật số.

Robot di động bắt đầu xuất hiện như những hệ thống tự hành cơ bản được phát triển trong các phòng thí nghiệm nghiên cứu từ những năm đầu của thập niên 1940 -1950. Một trong những ví dụ đầu tiên là robot “Elsie” và “Elmer” được thiết kế bởi W. Grey Walter vào năm 1948, đây là các robot rùa nhỏ có khả năng tự di chuyển và phản ứng với ánh sáng. Đến những năm của thập niên 1960 -1970, robot Shakey do SRI International phát triển đã đánh dấu một bước tiến quan trọng, Shakey là robot đầu tiên tích hợp trí thông minh nhân tạo, có khả năng lập kế hoạch và ra quyết định, đây cũng là robot đầu tiên sử dụng cảm biến và bản đồ môi trường để điều hướng.



Hình 1.1: Robot Shakey do SRI International phát triển [4]

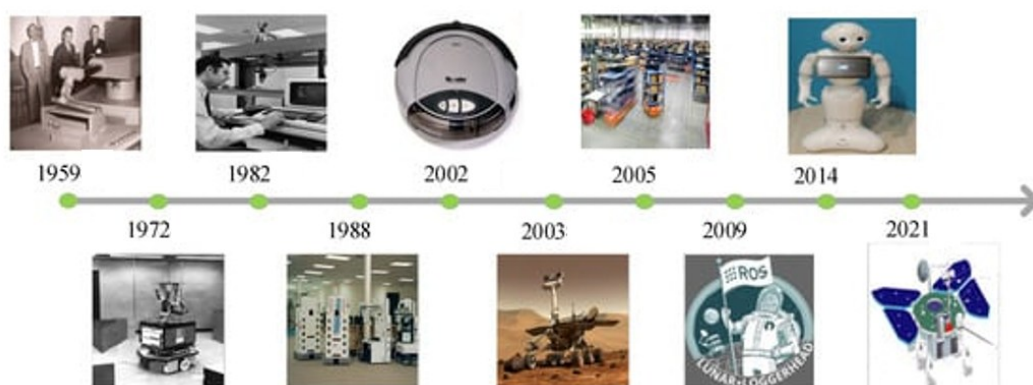
Với sự tiến bộ trong công nghệ máy tính, cảm biến và hệ thống điều khiển robot di động được ứng dụng rộng rãi trong công nghiệp như xe tự hành (AGV) trong các nhà máy. Các tổ chức như NASA cũng bắt đầu phát triển robot di động để khám phá không gian, điển hình là Sojourner robot thám hiểm sao hỏa đầu tiên ra mắt vào năm 1997. Từ những năm 2000, sự phát triển của trí tuệ nhân tạo, cảm biến LIDAR, và máy học đã đưa robot di động vào đời sống hàng ngày. Các ví dụ điển hình bao gồm robot hút bụi (Roomba), xe tự lái của các hãng như Tesla và Waymo, hay robot giao hàng tự hành. Ngoài ra, trong lĩnh vực nghiên cứu, các nền tảng mã nguồn mở như ROS đã giúp việc phát triển và thử nghiệm robot di động trở nên dễ dàng và phổ biến hơn.



Hình 1.2: Sojourner robot thám hiểm Sao Hỏa đầu tiên được giới thiệu vào năm 1997 [5]

Trong lịch sử của robot di động, các phương pháp xác định vị trí cho robot di động đã được đề xuất và thực hiện thành công. Trong nghiên cứu về robot di động, nhiều phương pháp đã được đề xuất nhằm nâng cao khả năng định vị, lập kế hoạch và điều hướng trong các môi trường phức tạp. Một số hướng tiếp cận tập trung vào việc sử dụng hình ảnh màu kết hợp với thông tin bản đồ để xác định vị trí của robot trong môi trường hành lang của tòa nhà. Bên cạnh đó, các kỹ thuật lập kế hoạch quỹ đạo hiện đại đã tính đến cả nội dung thông tin của môi trường lẫn sự phân bố mật độ người, giúp đường đi được thiết kế an toàn và tối ưu hơn. Các phương pháp điều khiển dựa trên hành vi cũng chứng minh tính hiệu quả trong việc cho phép robot hoạt động ổn định trong nhiều tình huống thực tế. Ngoài ra, một số nghiên cứu đã phát triển các cơ chế phát hiện tình huống bí đường nhằm giúp hệ thống điều hướng phản ứng kịp thời, tránh sự cố trong quá trình vận hành.

Những thuật toán tối ưu hóa đa mục tiêu, đặc biệt là dựa trên di truyền, cũng được áp dụng để giải quyết bài toán lập kế hoạch đường đi phức tạp, cân bằng giữa nhiều tiêu chí khác nhau như độ dài quỹ đạo, mức độ an toàn và hiệu quả tính toán. Gần đây, robot di động còn được triển khai trong các môi trường công nghiệp đặc thù, chẳng hạn như kiểm tra trạm biến áp, nơi điều kiện làm việc phức tạp và tiềm ẩn nhiều nguy hiểm. Những nghiên cứu này đã mở ra cái nhìn mới về việc ứng dụng robot vào các nhiệm vụ đòi hỏi sự an toàn cao, đồng thời khẳng định vai trò ngày càng quan trọng của robot trong các hệ thống kỹ thuật hiện đại.



Hình 1.3: Quá trình phát triển của robot di động [6]

Robot di động đang ngày càng trở nên thông minh và linh hoạt hơn, đặc biệt là robot di động để thực hiện được các nhiệm vụ trong các môi trường không kiểm soát ví dụ như: Trong logistics, chúng được sử dụng để vận chuyển hàng hóa trong kho, giao hàng tự động, hoặc di chuyển hành lý tại sân bay; trong nông nghiệp, robot hỗ trợ thu hoạch, gieo hạt, kiểm tra sức khỏe cây trồng với sự trợ giúp của thị giác máy tính và cảm biến IoT; trong y tế, robot đảm nhiệm việc vận chuyển thuốc, khử khuẩn và chăm sóc bệnh nhân; chúng cũng xuất hiện trong các dịch vụ như phục vụ tại nhà hàng, làm hướng dẫn viên tại bảo tàng, hay dọn rác ở các khu vực công cộng.... Ngoài ra, robot quân sự và an ninh đảm bảo tuần tra, xử lý bom mìn và cứu hộ trong điều kiện nguy hiểm. Đặc biệt, trong khám phá không gian, robot được thiết kế để thám hiểm hành tinh, lắp ráp vệ tinh, và hỗ trợ trên các trạm không gian. Các công nghệ như cảm biến thông minh, hệ thống điều hướng chính xác, và vật liệu bền bỉ đã giúp robot di động trở thành giải pháp toàn diện, đáp ứng nhu cầu đa dạng của con người. Có thể thấy robot di động hiện nay đã trở thành công cụ quan trọng trong nhiều lĩnh vực đời sống nhờ tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo (AI), định vị và lập bản đồ đồng thời (SLAM), cảm biến LIDAR, và hệ thống điều khiển tự động.

Với tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực, chúng ta có thể thấy được tương lai phát triển của robot di động trong đời sống, nó không chỉ nằm ở việc trở thành công cụ hỗ trợ để nâng cao chất lượng cuộc sống, hỗ trợ con người trong các hoạt động mà còn thay thế con người trong các nhiệm vụ nguy hiểm hoặc lặp lại và mở ra nhiều khả năng mới trong nhiều lĩnh vực khác trong tương lai. Tuy nhiên, để đạt được hiệu quả tối ưu, chúng ta cần tiếp tục giải quyết các thách thức về cảm biến, định vị, và khả năng thích nghi với điều kiện khắc nghiệt. Những tiến bộ trong công nghệ AI và thiết kế phần cứng sẽ đóng vai trò then chốt trong việc đưa robot di động trở thành công cụ không thể thiếu trong nhiều ngành công nghiệp.

1.2. Tổng quan về phương pháp thiết kế bản đồ

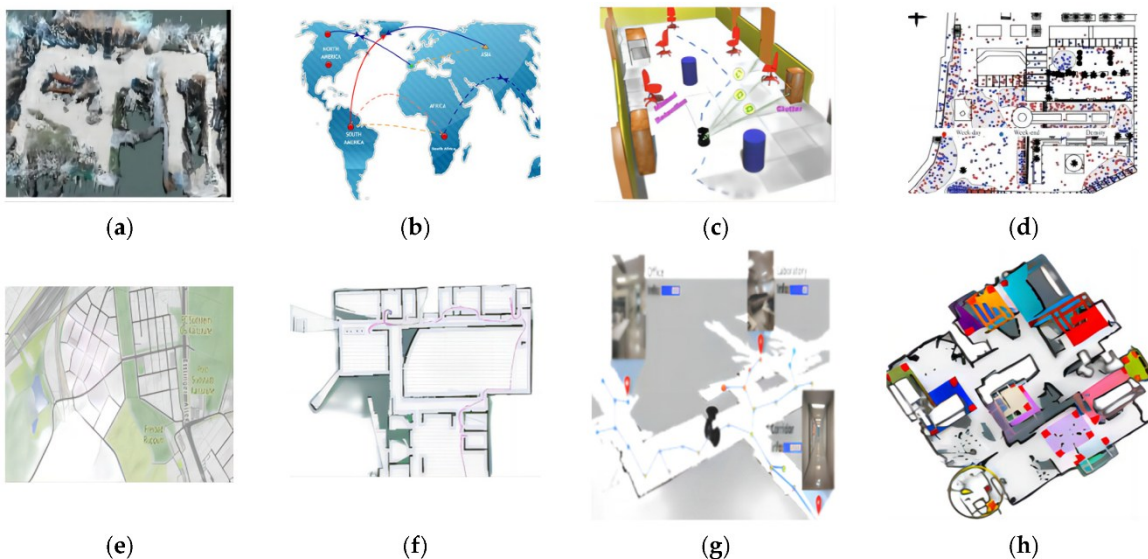
Robot di động là thiết bị thông minh có thể thực hiện các nhiệm vụ cụ thể một cách độc lập trong môi trường phức tạp [7] được ứng dụng phổ biến trong các nhiệm vụ tự hành, không yêu cầu sự giám sát từ con người. Khả năng tự hành của robot trong môi trường chưa được biết trước đòi hỏi robot phải tự nhận diện môi trường xung quanh, xây dựng bản đồ, xác định vị trí, lập kế hoạch và điều khiển quá trình di chuyển. Để robot có thể nhận biết và di chuyển trong các môi trường một cách an toàn và hiệu quả thì thiết kế bản đồ cho robot di động là một trong những bước quan trọng. Việc thiết lập bản đồ cho robot di động giúp cho robot có thể biết được vị trí của mình trong môi trường và định hướng đến các điểm đích. Robot sử dụng bản đồ để xác định được không gian xung quanh, vị trí của các chướng ngại vật, và đường đi an toàn. Điều này đặc biệt quan trọng đối với các robot hoạt động trong các môi trường phức tạp hoặc không biết trước. Bản đồ cung cấp cho robot thông tin cần thiết để lập kế hoạch di chuyển hiệu quả. Với bản đồ, robot có thể tính toán quãng đường ngắn nhất hoặc an toàn nhất để đến đích, tránh các chướng ngại vật, tối ưu hóa việc sử dụng năng lượng và cải thiện đáng kể hiệu suất hoạt động của robot.



Hình 1.4: Thiết lập tạo bản đồ cho robot lau nhà sử dụng điện thoại thông minh [8]

Thiết kế bản đồ môi trường là một trong những khía cạnh quan trọng trong nghiên cứu robot di động, đặc biệt khi giải quyết các vấn đề định vị, xác định vị trí, điều hướng tự động, lập kế hoạch di chuyển và tránh chướng ngại vật cũng như tìm kiếm và cứu hộ. Thành công của quá trình này phụ thuộc vào độ chính xác và độ tin cậy trong việc triển khai, đồng thời cũng dựa vào các cảm biến dùng để thu thập dữ liệu môi trường. [9] Thiết kế bản đồ (mapping) có thể được mô tả như một quá trình minh họa môi trường dựa trên thông tin xung quanh, được biểu diễn dưới dạng hình ảnh đồ họa. Bản đồ có thể được trình bày dưới dạng bản đồ số liệu (metric) hoặc bản đồ địa hình (topological). Thiết kế bản đồ hỗ trợ định vị bằng cách hiển thị vị trí của mục tiêu hoặc đối tượng trong khu vực. Ngoài ra, nó còn giúp mô tả môi trường hoặc sơ đồ mặt bằng. Có nhiều phương pháp thiết kế bản đồ như bản đồ lưới chiếm dụng (occupancy grid map), bản đồ đặc trưng (feature map), bản đồ địa hình (topological map) hoặc bản đồ dựa trên dữ liệu thô từ cảm biến hay phương pháp học sâu trong xây dựng bản đồ. Một trong những thách thức lớn trong thiết kế bản đồ là đảm bảo độ chính xác và độ tin cậy của cảm biến. Các loại cảm biến được sử dụng bao gồm camera ảnh biến đo khoảng cách laser, cảm biến siêu âm và Kinect. Tuy nhiên, mỗi loại đều có những ưu điểm và hạn chế riêng. Chẳng hạn, cảm biến siêu âm có thể gặp phải các vấn đề như khoảng trống tín hiệu (blanking interval), ảnh hưởng của tốc độ âm thanh, sai lệch góc và sự tương tác với bề mặt. Những yếu tố này có thể gây ảnh hưởng tiêu cực đến hệ thống [10], nhưng có thể được khắc phục bằng cách sử dụng đồng thời nhiều cảm biến siêu âm trong một hệ thống.

1.2.1. Các loại bản đồ



Hình 1.5: Ví dụ về các loại bản đồ [6]

Theo [6], trong lĩnh vực nghiên cứu lập kế hoạch đường đi của robot di động, mô hình hóa môi trường trong thế giới thực là bắt buộc. Các bản đồ tỷ lệ, bản đồ tô pô, bản đồ ngữ nghĩa và bản đồ hành vi được mô tả trong Hình 1.5 minh họa các mức độ trừu tượng đa dạng được sử dụng để mô tả và mô hình hóa môi trường. Các loại bản đồ này đóng vai trò là các chiến lược phổ biến và riêng biệt để đại diện môi trường, được mô tả chi tiết như sau:

- Bản đồ tỷ lệ (a): Còn được gọi là bản đồ đo lường, bản đồ số liệu thường dựa vào lưới hoặc pixel để phân chia môi trường thành một loạt các đơn vị rời rạc. Mỗi đơn vị biểu thị một trạng thái và có thể đại diện cho không gian trống hoặc chướng ngại vật. Bản đồ số liệu có thể thể hiện các kích thước vật lý của thế giới thực. Nó bao gồm nhiều loại khác nhau như bản đồ lưới chiếm dụng, bản đồ độ sâu và bản đồ xác suất.

- Bản đồ tô pô (b): Mô tả vị trí của robot thường không liên quan đến các chiều vật lý thực tế của thế giới. Thay vào đó, nó tận dụng kết nối và mối quan hệ khoảng cách giữa các vị trí khác nhau để mô tả vị trí của robot. Loại bản đồ này bao gồm hai loại chính: bản đồ tô pô dựa trên nút và dựa trên cạnh. Bản đồ tô pô tìm thấy tiện ích trong các tình huống mà môi trường thể hiện một số đều đặn nhất định, như đã thấy trong mạng lưới đường bộ thành phố.

- Bản đồ ngữ nghĩa (c): Bản đồ ngữ nghĩa thường dựa trên tính năng, trong đó các thực thể môi trường được dán nhãn để nâng cao nhận thức và hiểu biết của robot. Việc dán nhãn này tạo điều kiện cho sự tự chủ và trí thông minh cao hơn bằng cách truyền đạt ý nghĩa cho nhận thức của robot về môi trường xung quanh.

- Bản đồ hành vi (d): Được sử dụng để thể hiện hành vi và thông tin nhiệm vụ của robot, bản đồ hành vi phác thảo các hành động mà robot cần thực hiện để hoàn thành các nhiệm vụ cụ thể. Bản đồ hành vi thường được tạo bằng cách sử dụng hệ thống điều khiển robot và thuật toán lập kế hoạch, cung cấp tiện ích trong các ứng dụng ra quyết định, thực thi và học tập tự động.

- Bản đồ vệ tinh (e): Tận dụng hình ảnh vệ tinh, bản đồ vệ tinh gói gọn cảnh quan mặt đất để khắc họa sống động địa hình thực tế và các đặc điểm tự nhiên như núi, rừng, hồ và sông.

- Bản đồ trong nhà (f): Được thiết kế đặc biệt để điều hướng trong môi trường hạn chế, bản đồ trong nhà phục vụ cho các không gian như khu phức hợp mua sắm, sân bay, bảo tàng và trung tâm y tế. Họ thường kết hợp các sơ đồ tầng phức

tạp cho các cấp độ khác nhau, các địa danh có chú thích, tín hiệu định hướng, đường đi được chỉ định và vị trí chính xác của các tiện nghi đa dạng.

- Bản đồ giao thông (g): Lĩnh vực của bản đồ giao thông bao gồm đường xá, biển báo giao thông, tín hiệu và cơ sở hạ tầng xe cộ khác. Robot sử dụng thành thạo những dữ liệu quan trọng này để vạch ra các tuyến đường tối ưu, đảm bảo tuân thủ các quy định giao thông và nghi thức đường bộ để điều hướng an toàn và hiệu quả.

- Bản đồ lai (h): Kết nối các thuộc tính của các thể loại bản đồ đa dạng, bản đồ kết hợp có tiềm năng cung cấp một kho thông tin địa lý toàn diện và chính xác. Những bản đồ này đặc biệt vô giá để điều hướng các địa hình phức tạp, tạo điều kiện thuận lợi cho việc lập kế hoạch đường đi phức tạp. Ví dụ: tích hợp bản đồ số liệu với bản đồ tô pô có thể mang lại định vị và điều hướng robot chính xác. Tương tự, kết hợp bản đồ ngữ nghĩa với bản đồ hành vi sẽ đẩy nhanh việc lập kế hoạch và thực hiện nhiệm vụ.

1.2.2. Phương pháp truyền thống

Các phương pháp truyền thống trong thiết kế bản đồ dựa trên nền tảng kỹ thuật số hóa thủ công, xử lý ảnh viễn thám và hệ thống thông tin địa lý (GIS). Những phương pháp này đã hình thành cơ sở cho sự phát triển của bản đồ số trước khi học sâu trở nên phổ biến.

Hệ thống thông tin địa lý (GIS): Là công cụ chính để thu thập, lưu trữ, phân tích và hiển thị dữ liệu không gian. Phương pháp này chuyển đổi bản đồ giấy hoặc ảnh vệ tinh thành dữ liệu vector (điểm, đường, vùng) thông qua phần mềm như ArcGIS hoặc QGIS.

- Ưu điểm:

- + Độ chính xác cao khi dữ liệu đầu vào được kiểm định;
- + Linh hoạt trong việc tích hợp dữ liệu đa nguồn.

- Hạn chế:

- + Tốn thời gian và nhân lực cho quy trình thủ công;
- + Khó xử lý dữ liệu lớn (big data) hoặc cập nhật thời gian thực.

Kỹ thuật viễn thám (Remote Sensing) và Photogrammetry: Viễn thám kết hợp Photogrammetry cho phép tái tạo bề mặt địa hình từ ảnh chụp hàng không hoặc vệ tinh.

- Quá trình diễn hình:

- + Thu thập ảnh: Sử dụng camera đa phổ hoặc LIDAR trên máy bay/vệ tinh.
- + Xử lý ảnh: Loại bỏ biến dạng do góc chụp hoặc độ cao; áp dụng kỹ thuật Structure from Motion (SfM) để dựng hình 3D từ ảnh 2D [11].

- + Phân loại đối tượng: Sử dụng phương pháp phân loại có giám sát (Supervised Classification) để nhận diện lớp phủ (rừng, đô thị, mặt nước,...).

- Ưu điểm:

- + Cung cấp dữ liệu địa hình chi tiết ở quy mô lớn;

- + Hiệu quả trong giám sát biến động môi trường (ví dụ: phá rừng, đô thị hóa).

- Hạn chế:

- + Chi phí cao cho thiết bị và xử lý dữ liệu;

- + Độ chính xác phụ thuộc vào điều kiện thời tiết và độ phân giải ảnh.

Phương pháp khảo sát thực địa (Field Surveying): Thu thập dữ liệu trực tiếp từ hiện trường bằng thiết bị đo đạc như máy toàn đạc (Total Station), GPS- RTK (Real-Time Kinematic).

- Quy trình:

- + Thiết lập hệ tọa độ, chuẩn hóa hệ quy chiếu;

- + Đo đạc chi tiết: Ghi nhận tọa độ các đối tượng (cột mốc, đường ranh giới,...);

- + Xây dựng bản đồ: Số hóa dữ liệu vào phần mềm chuyên dụng.

- Ưu điểm:

- + Độ chính xác cực cao (sai số dưới 5cm với GPS RTK);

- + Phù hợp cho các dự án quy mô nhỏ (xây dựng cơ sở hạ tầng).

- Hạn chế:

- + Không khả thi cho khu vực rộng lớn hoặc địa hình hiểm trở;

- + Yêu cầu chuyên môn cao và thời gian triển khai dài.

Các phương pháp truyền thống đặt nền móng cho việc xây dựng bản đồ, nhưng tồn tại hạn chế về khả năng tự động hóa và xử lý dữ liệu phức tạp. Sự ra đời của học sâu đã mở ra hướng tiếp cận mới, khắc phục những nhược điểm này thông qua việc tích hợp dữ liệu đa chiều và tối ưu hóa thông minh.

1.2.3. Phương pháp hiện đại

Các phương pháp hiện đại dựa trên nền tảng học sâu (deep learning) và xử lý dữ liệu lớn, mang lại khả năng tự động hóa, xử lý thời gian thực và độ chính xác cao, có thể kể đến một số hướng tiếp cận như:

SLAM (Simultaneous Localization and Mapping) kết hợp học sâu: SLAM là công nghệ cho phép robot hoặc thiết bị tự động xây dựng bản đồ môi trường xung quanh đồng thời xác định vị trí của chính nó trong bản đồ đó. SLAM kết hợp học sâu để cải thiện độ chính xác, xử lý nhiễu và tự động hóa quy trình.

Digital Twin và IoT: Kết hợp Digital Twin (bản sao số) và IoT để cập nhật bản đồ thời gian thực.

Các phương pháp hiện đại dựa trên học sâu và công nghệ số đã cách mạng hóa việc xây dựng bản đồ số và thiết kế quỹ đạo, mang lại khả năng tự động hóa, độ chính xác cao và thích ứng với môi trường động.

1.2.4. Một số nghiên cứu về thiết kế bản đồ cho robot di động

Nghiên Cứu Về Bản Đồ Dựa Trên Lưới: Một trong những nghiên cứu nền tảng về thiết kế bản đồ là công trình của Elfes, người đã đề xuất phương pháp sử dụng lưới chiếm chỗ (occupancy grid) để biểu diễn môi trường [12]. Phương pháp này chia không gian thành các ô lưới nhỏ, mỗi ô được gán giá trị xác suất dựa trên dữ liệu từ cảm biến sonar, giúp robot nhận diện vật cản. Tuy đơn giản, cách tiếp cận này đặt nền móng cho các nghiên cứu sau về lập bản đồ xác suất, mặc dù bị hạn chế bởi yêu cầu bộ nhớ lớn khi áp dụng trong không gian rộng.

SLAM Dựa Trên Bộ Lọc Kalman: Nghiên cứu của Smith và Cheeseman đã giới thiệu cách tiếp cận dựa trên đặc trưng (feature-based mapping) kết hợp với bộ lọc Kalman [13]. Họ sử dụng các điểm mốc (landmarks) trong môi trường để xây dựng bản đồ thưa, đồng thời cập nhật vị trí robot và các đặc trưng này. Phương pháp này sau đó được phát triển thêm thành EKF-SLAM (Extended Kalman Filter SLAM) bởi Dissanayake và cộng sự, trở thành một giải pháp phổ biến cho các môi trường có quy mô vừa và nhỏ, dù gặp khó khăn khi xử lý sai số tích lũy ở không gian lớn [14].

SLAM Dựa Trên Tối Ưu Hóa Đồ Thị: Grisetti và cộng sự đã đề xuất một hướng tiếp cận khác với SLAM dựa trên đồ thị (graph-based SLAM) [15]. Phương pháp này mô phỏng quá trình lập bản đồ thành một đồ thị, trong đó các nút là vị trí của robot và các cạnh là ràng buộc không gian từ dữ liệu cảm biến. Các thuật toán tối ưu hóa như G2O được sử dụng để giảm sai số toàn cục, giúp tạo ra bản

đồ chính xác hơn, đặc biệt trong các môi trường phức tạp như tòa nhà hoặc khu vực đô thị.

Visual SLAM Và Ứng Dụng Học Máy: Với sự phát triển của thị giác máy tính, Mur-Artal và cộng sự đã giới thiệu ORB-SLAM, một hệ thống SLAM thị giác dựa trên camera đơn [16]. ORB-SLAM sử dụng các đặc trưng ORB để trích xuất thông tin từ hình ảnh, xây dựng bản đồ thưa hiệu quả trong không gian 3D. Gần đây hơn, Zhou và cộng sự đã tích hợp học sâu vào Visual SLAM, sử dụng mạng nơ-ron để dự đoán độ sâu, nâng cao khả năng tái tạo môi trường mà không cần cảm biến đắt tiền như LiDAR [17].

Các nghiên cứu gần đây, như của Zhang và cộng sự, tập trung vào việc ứng dụng học sâu để cải thiện SLAM trong môi trường động [18]. Bằng cách kết hợp mạng nơ-ron tích chập (CNN) và dữ liệu cảm biến, họ đã phát triển các hệ thống có khả năng thích nghi với sự thay đổi của vật thể trong không gian, mở ra tiềm năng cho robot hoạt động trong các tình huống thực tế như nhà máy hoặc khu vực đông người.

1.3. Tổng quan về phương pháp hoạch định đường dẫn (Path Planning)

1.3.1. Hoạch định đường dẫn

Từ giữa những năm 1960, việc hoạch định đường dẫn đã thu hút sự quan tâm từ rất nhiều học giả. Lập kế hoạch đường đi cho robot di động đề cập đến việc thiết kế đường dẫn an toàn, không va chạm với khoảng cách ngắn nhất và không tốn thời gian nhất từ điểm xuất phát đến điểm kết thúc bởi một robot di động tự hành [7]. Gần đây, rất nhiều nghiên cứu đã được thực hiện trong lĩnh vực lập kế hoạch chuyển động cho robot di động. Vấn đề hoạch định đường dẫn là việc xác định đường đi cho robot di động từ vị trí hiện tại đến vị trí mục tiêu thông qua môi trường chướng ngại vật. Các chướng ngại vật có thể đứng yên hoặc di chuyển. Con đường mong muốn là một con đường thời gian ngắn nhất, nơi không có va chạm giữa robot và các chướng ngại vật. Trong quá trình hoạch định đường dẫn, robot di động luôn tuân thủ theo một nguyên tắc chung để đảm bảo tìm được quãng đường tối ưu nhất, trong thời gian ngắn nhất và có thể tránh được vật cản hiệu quả: (1) Mô hình môi trường robot di động di chuyển được chuyển thành dạng bản đồ để có thể lưu trữ thuận tiện, (2) các tiêu chí tối ưu hóa, (3) Thuật toán tìm kiếm đường đi đáp ứng được các bộ tiêu chí tối ưu hóa trong không gian trạng thái.

Hoạch định đường dẫn (Path Planning) là một lĩnh vực quan trọng trong robot học và điều khiển tự động, nhằm xác định một lộ trình tối ưu cho robot từ vị trí ban đầu đến đích trong không gian làm việc mà không va chạm với chướng ngại vật. Ngày nay, hoạch định đường dẫn còn được ứng dụng rộng rãi trên nhiều lĩnh vực cụ thể như xe tự hành, máy bay không người lái (UAV), robot công nghiệp và các hệ thống tự động khác [19].

Một số phương pháp về hoạch định đường dẫn đã được đề xuất và nghiên cứu. Tuy nhiên, ta có thể chia hoạch định đường dẫn thành 2 loại đó là hoạch định đường dẫn toàn cầu (Global Path Planning), hoạch định đường dẫn cục bộ (Local Path Planning) [20].

Hoạch định Toàn cầu: Là quá trình xác định lộ trình tối ưu từ điểm xuất phát đến đích dựa trên thông tin đầy đủ và tĩnh về môi trường. Phương pháp này yêu cầu bản đồ được xây dựng trước (ví dụ: bản đồ chi tiết của tòa nhà hoặc địa hình) và thường không xét đến các chướng ngại vật động [21].

Hoạch định Cục bộ: Tập trung vào việc điều chỉnh lộ trình trong thời gian thực để tránh chướng ngại vật không xác định hoặc di động, dựa trên dữ liệu từ cảm biến (ví dụ: lidar, camera). Phương pháp này không yêu cầu bản đồ đầy đủ nhưng đòi hỏi khả năng phản ứng nhanh [22].

Đối với hoạch định đường dẫn toàn cầu (Global Path Planning) đòi hỏi một lượng lớn thông tin chi tiết về môi trường từ đó tiếp cận mục tiêu theo một đường đi đã được xác định trước. Trong khi đó, hoạch định đường đi cục bộ (Local Path Planning) robot có thể biết 1 phần hoặc không nắm được thông tin về môi trường mà chủ yếu dựa vào việc thu thập và xử lý thông tin trong quá trình di chuyển. Hoạch định đường dẫn cục bộ (Local Path Planning) có đặc điểm là tính linh hoạt tốt, trong khi nhược điểm là đường đi có thể tối ưu cục bộ nhưng không được đảm bảo trên toàn cục hoặc thậm chí mục tiêu có thể không tiếp cận được. Mặt khác, hoạch định đường dẫn toàn cầu (Global Path Planning) yêu cầu robot di chuyển phải hiểu và xây dựng một mô hình bản đồ toàn cục, sử dụng thuật toán tìm kiếm trên mô hình bản đồ để có được đường đi tối ưu hoặc không tối ưu để hướng dẫn robot di chuyển an toàn đến điểm mục tiêu trong môi trường thực tế [7]. Hoạch định đường đi đề cập đến việc sử dụng các thuật toán để lập kế hoạch đường đi cho robot di động tạo ra một đường đi khả thi dựa trên mô hình môi trường.

Hoạch định đường dẫn (Path Planning) là một lĩnh vực quan trọng trong robot học và trí tuệ nhân tạo, tập trung vào việc xác định lộ trình di chuyển tối ưu cho các

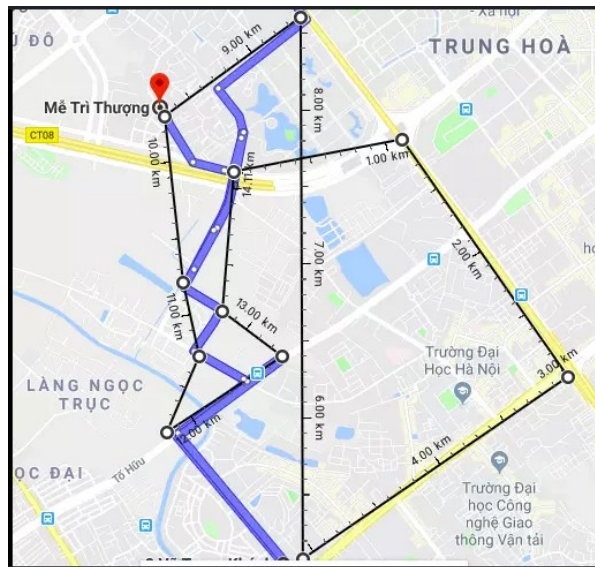
tác nhân như robot di động, phương tiện tự hành và UAV trong môi trường có hoặc không có cấu trúc. Mục tiêu của hoạch định đường dẫn không chỉ dừng lại ở việc tìm kiếm một quỹ đạo hợp lệ, đảm bảo tránh va chạm với các chướng ngại vật và tuân theo các ràng buộc động học, mà còn hướng đến tối ưu hóa các yếu tố như quãng đường, thời gian di chuyển và mức tiêu hao năng lượng. Ngoài ra, trong bối cảnh môi trường động và không chắc chắn, các thuật toán hoạch định đường dẫn cần có khả năng thích ứng linh hoạt, đảm bảo tính khả thi khi triển khai trong thời gian thực. Do đó, nghiên cứu về hoạch định đường dẫn không chỉ đóng vai trò quan trọng trong robot học mà còn có nhiều ứng dụng trong giao thông thông minh, mô phỏng tác nhân và các hệ thống tự động khác.

1.3.2. Các thuật toán hoạch định đường dẫn cho robot di động

Một thuật toán lập kế hoạch đường dẫn hiệu quả cần đáp ứng bốn tiêu chí. Đầu tiên, kỹ thuật lập kế hoạch chuyển động phải có khả năng luôn tìm ra đường đi tối ưu trong môi trường tĩnh thực tế. Thứ hai, nó phải có thể mở rộng cho các môi trường năng động. Thứ ba, nó phải duy trì sự tương thích và nâng cao cách tiếp cận tự tham chiếu đã chọn. Thứ tư, nó phải giảm thiểu độ phức tạp, lưu trữ dữ liệu và thời gian tính toán [23].

Dựa vào các tính năng đặc biệt của các thuật toán hoạch định đường dẫn đơn tác nhân (SAPF), phân loại các thuật toán trong hoạch định đường dẫn gồm 3 loại: thuật toán cổ điển, thuật toán tối ưu hóa thông minh và thuật toán sử dụng trí tuệ nhân tạo [6]. Đối với các thuật toán cổ điển chúng ta có thể xét đến các thuật toán như thuật toán tìm kiếm dựa trên đồ thị (Dijkstra, A*, D và các biến thể), thuật toán dựa trên lấy mẫu (RRT, PRM) và đi vào phân tích đối với các thuật toán cổ điển.

Thuật toán Dijkstra, được phát triển bởi nhà khoa học máy tính người Hà Lan Edsger W. Dijkstra vào năm 1956 và công bố chính thức vào năm 1959, là một trong những thuật toán nền tảng trong lý thuyết đồ thị. Thuật toán này giải quyết bài toán tìm đường đi ngắn nhất từ một đỉnh nguồn đến tất cả các đỉnh khác trong một đồ thị có trọng số không âm [24]. Với tính ứng dụng rộng rãi trong các lĩnh vực như định tuyến mạng, hệ thống định vị toàn cầu (GPS), và tối ưu hóa giao thông, thuật toán Dijkstra đã trở thành một công cụ quan trọng trong khoa học máy tính và kỹ thuật.

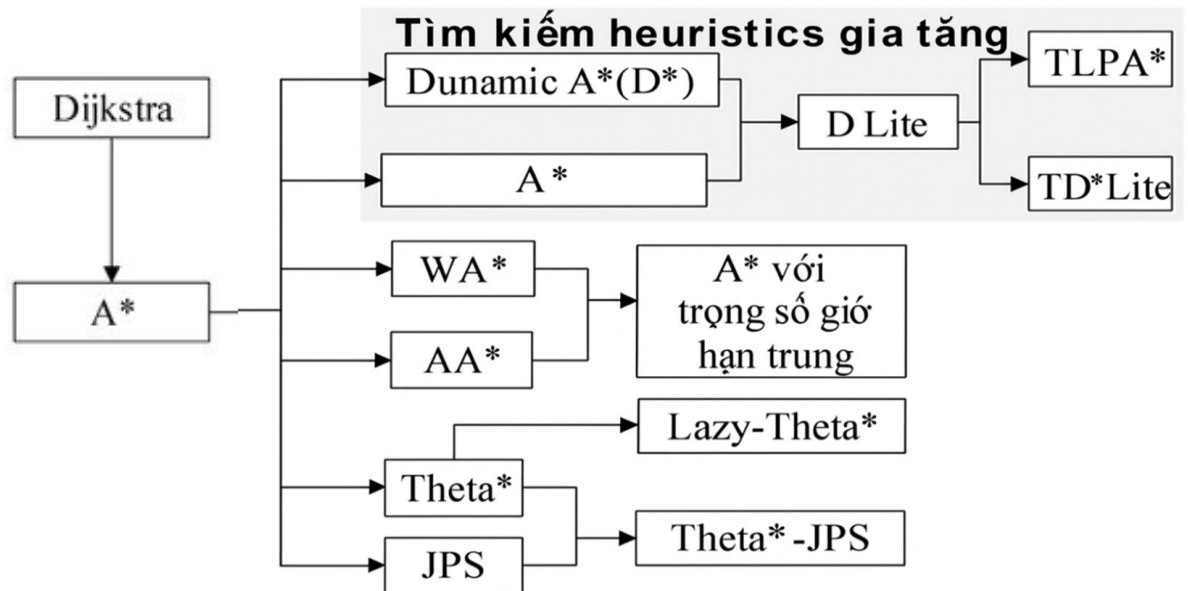


Hình 1.6: Ứng dụng giải thuật Dijkstra tìm đường đi ngắn nhất từ vị trí mốc đến Mễ Trì Thường [25]

Để tăng tốc độ tìm kiếm so với thuật toán Dijkstra, thuật toán A^* đã được đề xuất bởi Peter Hart và các cộng sự lấy cảm hứng từ thuật toán Dijkstra. Thuật toán A^* là một thuật toán Heuristic sử dụng thông tin Heuristic để tìm ra đường tối ưu [23]. Thuật toán này sử dụng thông tin phỏng đoán để hướng dẫn quá trình tìm kiếm đồng thời tính đến cả chi phí để tiếp cận từng nút từ nút bắt đầu và chi phí để đạt được đến nút mục tiêu. Tuy nhiên, thuật toán A^* truyền thống đã gặp phải các vấn đề về hiệu quả tìm kiếm thấp, chi phí lớn và tốn nhiều bộ nhớ khi tìm kiếm trong các môi trường lớn. Do đó, nhiều biến thể của thuật toán A^* đã được phát triển để khắc phục các hạn chế này có thể kể đến như WA^* (Weighted A^*), AA^* (Adaptive A^*), Θ^* , JPS (Jump Point Search) như sơ đồ Hình 1.7. Pu và cộng sự đã giới thiệu một thuật toán A^* thích ứng kép, bao gồm các chức năng phỏng đoán đa mục tiêu thích ứng và chiến lược mở rộng nút thích ứng. Lai và cộng sự đã đề xuất một thuật toán A^* thích ứng bị hạn chế tập trung gán trọng số động cho các nút ở các vị trí khác nhau và kết hợp các ngưỡng thích ứng vào hàm phỏng đoán để tăng cường khả năng thích ứng. Thuật toán Θ^* , ban đầu được giới thiệu bởi Daniel, cho phép tìm kiếm đường dẫn dọc theo các góc tùy ý và sau đó đánh dấu Lazy- Θ^* hiệu quả hơn, Luo đã áp dụng khái niệm kích thước bước thay đổi của thuật toán Θ^* cho thuật toán JPS, cải thiện hơn nữa chất lượng đường dẫn và độ mượt mà của JPS truyền thống [6].

Θ^* được thiết kế nhằm tạo ra đường đi ngắn hơn và mượt hơn so với A^* truyền thống. Trong khi A^* chỉ cho phép mở rộng các nút kế cận thông qua các bước di chuyển rời rạc và hạn chế hướng đi theo lưới (grid), Θ^* cho phép kết

nối trực tiếp giữa các nút không liên kề, miễn là không có vật cản giữa chúng. Cơ chế này giúp tạo ra các đường đi gần với đường thẳng hơn, phù hợp hơn cho chuyển động của robot thực tế. Điểm đặc biệt của Theta* là việc sử dụng kiểm tra tầm nhìn (line-of-sight check) giữa nút hiện tại và cha của nó. Nếu có thể đi thẳng từ cha của nút hiện tại đến nút đang xét mà không va chạm vật cản, thuật toán sẽ cập nhật đường đi thông qua cha đó thay vì đi vòng theo các bước lưới, từ đó rút ngắn tổng chi phí và tăng tính tối ưu của lộ trình. Về mặt thuật toán, Theta* giữ nguyên cấu trúc dữ liệu ưu tiên (priority queue) và hàm chi phí tương tự như A*, nhưng trong bước cập nhật cha (parent) của các nút, thuật toán thực hiện thêm phép kiểm tra tầm nhìn để tối ưu hóa lộ trình. Nhờ đó, đường đi do Theta* tạo ra thường ít góc gấp khúc hơn, mượt mà hơn, đồng thời rút ngắn tổng quãng đường di chuyển — điều rất quan trọng đối với robot di động trong môi trường thực tế.

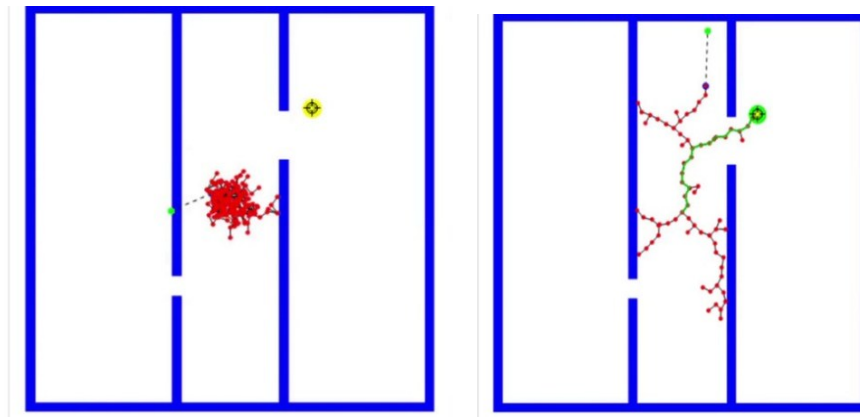


Hình 1.7: A* và các biến thể của thuật toán A* [6]

Trong môi trường động, việc lập kế hoạch đường đi hiệu quả đòi hỏi khả năng thích ứng nhanh với những thay đổi của môi trường. Để hỗ trợ việc lập kế hoạch đường đi cho môi trường động, thuật toán D* (hoặc Dynamic A*) và các biến thể của nó đã được ra đời và là công cụ hiệu quả để lập kế hoạch lại một cách nhanh chóng trong môi trường động [23]. Thuật toán D* lần đầu tiên được Anthony Stentz giới thiệu vào năm 1994 như một phương pháp tìm kiếm gia tăng có thông tin [26]. Không giống A* truyền thống, D* bắt đầu từ đích và tính toán đường đi ngắn nhất từ mọi điểm khởi đầu tiềm năng đến đích. Khi phát hiện chướng ngại vật mới, D* cập nhật bản đồ bằng cách truyền trạng thái RAISE và

LOWER để điều chỉnh chi phí đường đi một cách hiệu quả. Phương pháp này đặc biệt hữu ích trong các môi trường chưa biết trước, chẳng hạn như điều hướng robot trên địa hình gồ ghề, và đã được áp dụng rộng rãi nhờ khả năng lập kế hoạch lại nhanh chóng. Trong thập kỷ 2010-2020, nhiều nghiên cứu đã mở rộng D^* và các biến thể để giải quyết các thách thức mới có thể kể đến công trình của Yahja và cộng sự đã tích hợp D^* với các kỹ thuật lập kế hoạch dựa trên mẫu (sampling-based planning), nhằm cải thiện khả năng xử lý không gian trạng thái cao chiều [27]. Gần đây hơn, vào năm 2020, Zhang và cộng sự đã đề xuất một biến thể của D^* Lite kết hợp học sâu để dự đoán các thay đổi trong môi trường động, nâng cao khả năng thích nghi của thuật toán trong các tình huống thực tế như điều hướng đám đông hoặc môi trường đô thị [28]. Những cải tiến này cho thấy sự linh hoạt của D^* trong việc thích ứng với các tiến bộ công nghệ. Thuật toán D^* và các biến thể của nó có thể được sử dụng cho bất kỳ vấn đề tối ưu hóa chi phí đường dẫn nào trong đó chi phí đường dẫn thay đổi trong quá trình tìm kiếm đường tối ưu đến mục tiêu. D^* hiệu quả nhất khi những thay đổi này được phát hiện gần hơn những nút hiện tại trong không gian tìm kiếm. Hạn chế chính của thuật toán D^* là mức tiêu thụ bộ nhớ của nó cao hơn khi so sánh với các biến thể D^* khác [23].

Bên cạnh phương pháp A^* hay D^* thì các thuật toán dựa trên mẫu cũng được giới thiệu, các thuật toán như A^* có bản chất tĩnh và yêu cầu một đường dẫn được chỉ định trước cho chúng thì các thuật toán động và trực tuyến và không yêu cầu một đường dẫn được chỉ định trước cũng đã được giới thiệu. Năm 1998, LaValle giới thiệu thuật toán về cây ngẫu nhiên khám phá nhanh RRT (Rapidly exploring Random Tree). Ý tưởng cốt lõi đằng sau RRT liên quan đến việc khám phá các con đường trong môi trường chưa biết thông qua lấy mẫu ngẫu nhiên và mở rộng cây dần dần [6]. RRT mở rộng bằng cách nhanh chóng lấy mẫu không gian, phát triển từ điểm bắt đầu và mở rộng cho đến khi cây đủ gần điểm mục tiêu. Trong mỗi lần lặp, cây mở rộng đến đỉnh gần nhất của đỉnh được tạo ngẫu nhiên. Đỉnh gần nhất này được chọn theo số liệu khoảng cách [23]. Để đạt được sự tối ưu trong thuật toán thì các biến thể của thuật toán RRT đã được giới thiệu các biến thể của RRT như RRT-Connect, RRT*,... Và thuật toán RRT phù hợp cho các không gian có chiều cao, mở rộng ngẫu nhiên từ điểm bắt đầu.



Hình 1.8: Mô hình cây ngẫu nhiên RRT [23]

Năm 1996, thuật toán lộ trình xác suất PRM (Probabilistic Roadmap) đã được giới thiệu bởi Kavraki và các cộng sự [6]. Đây là một phương pháp lập kế hoạch đường dẫn dựa trên mẫu, xây dựng một lộ trình (roadmap) trong không gian tự do trước, sau đó sử dụng nó để tìm đường. Tuy nhiên, trong trường hợp các đường đi cần đi qua các chướng ngại vật dày đặc hoặc lối đi hẹp, hiệu quả của PRM có thể giảm [6]. Để khắc phục những hạn chế này, các nhà nghiên cứu đã tiến hành các nghiên cứu sâu rộng để nâng cao hiệu suất của nó, dẫn đến các biến thể khác nhau như PRM * (được giới thiệu vào năm 2010), Lazy PRM (được giới thiệu vào năm 2000), PRM-D * (được giới thiệu vào năm 2023).

Trong nghiên cứu [23] cũng đã đưa ra các thuật toán lập kế hoạch đường dẫn khác điển hình như thuật toán di truyền (GA) hay thuật toán tối ưu hóa đàn kiến (ACO). Thuật toán GA do Holland đề xuất vào năm 1975 [29], mô phỏng các quá trình di truyền, chọn lọc và giao nhau được quan sát thấy trong quần thể tự nhiên để tìm kiếm các giải pháp tối ưu. Thuật toán tối ưu hóa đàn kiến (ACO) được giới thiệu bởi học giả người Ý M. Dorigo vào năm 1992 đã tìm thấy các ứng dụng đáng chú ý trong lĩnh vực lập kế hoạch đường đi của robot di động do các tính năng phản hồi tích cực, tính toán phân tán và độ bền của nó. Giải quyết các vấn đề cố hữu về sự hội tụ chậm và tính nhạy cảm với tối ưu cục bộ trong ACO cơ bản, nhiều nhà nghiên cứu đã thực hiện tối ưu hóa và cải tiến trong thiết kế cấu trúc, lựa chọn tham số và tối ưu hóa pheromone [6].

Trong số các thuật toán đã nêu ra trên thuật toán A* nổi bật trong môi trường tĩnh nhờ tính hiệu quả trong việc tính toán và khả năng tìm đường ngắn nhất của nó, các thuật toán D* và các biến thể thì vượt trội hơn hẳn trong các môi trường động với khả năng lập kế hoạch lại nhanh chóng. Các phương pháp dựa trên mẫu đặc biệt là RRT và PRM cung cấp các giải pháp mạnh mẽ cho không gian cấu hình

phức tạp và chiều cao, với RRT phù hợp truy vấn đơn trong môi trường thay đổi và PRM lý tưởng cho các bài toán đa truy vấn trong không gian tĩnh. Tuy nhiên, các phương pháp này thường tạo ra đường dẫn không tối ưu ban đầu, đòi hỏi các biến thể như RRT* và PRM* để cải thiện. Ngược lại, các thuật toán tối ưu hóa như GA hoặc ACO, dù có khả năng tìm kiếm toàn cục tốt, lại gặp hạn chế về tốc độ hội tụ và tính khả thi trong ứng dụng thời gian thực. Các phương pháp dựa trên AI, với sự phát triển của học sâu và học tăng cường, mở ra tiềm năng thích nghi với môi trường không giám sát, nhưng vẫn cần vượt qua thách thức về dữ liệu huấn luyện và tài nguyên tính toán.

1.4. Phạm vi nghiên cứu

Đề án “Nghiên cứu xây dựng bản đồ số sử dụng phương pháp học sâu và thiết kế quỹ đạo cho robot di động” tập trung vào việc phát triển các giải pháp công nghệ nhằm tập trung vào việc ứng dụng các phương pháp học sâu (Deep Learning) để xử lý dữ liệu hình ảnh thu được từ vệ tinh thông qua Google Map hoặc Google Earth từ đó tập trung vào việc hoạch định đường dẫn (Path Planning) dựa trên bản đồ có được để tìm ra đường đi ngắn nhất cho robot di động sử dụng các thuật toán như A*, D* hay RRT.

1.5. Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

- Nghiên cứu tổng quan về robot di động làm việc trong môi trường ngoài trời;
- Nghiên cứu tổng quan về phương pháp thiết kế bản đồ và hoạch định đường dẫn trong robot di động;
- Nghiên cứu về phương pháp học sâu (deep learning) ứng dụng trong robot di động;
- Nghiên cứu về phương pháp thiết kế quỹ đạo chuyển động cho robot di động.

Phương pháp thực nghiệm: Mô phỏng.

Kết luận chương I

Trong chương này, tác giả đã trình bày một cách tổng quan về robot di động, bao gồm lịch sử phát triển, các loại bản đồ sử dụng trong định vị và dẫn đường, các phương pháp xây dựng bản đồ truyền thống và hiện đại. Ngoài ra, chương cũng nêu rõ vai trò quan trọng của các thuật toán hoạch định đường đi như Dijkstra, A*, RRT trong bài toán robot di động. Cuối cùng, phạm vi nghiên cứu và phương pháp nghiên cứu được xác định rõ, đặt nền móng cho các chương tiếp theo.

CHƯƠNG 2 - MÔ HÌNH HỌC SÂU VÀ CÁC THUẬT TOÁN THIẾT KẾ QUỸ ĐẠO

2.1. Tổng quan chung về học máy (Machine Learning)

2.1.1. Khái niệm

Sự phát triển của các thuật toán và mô hình cho phép máy tính học hỏi từ dữ liệu và nâng cao hiệu suất mà không cần lập trình rõ ràng là trọng tâm của học máy, và học máy có thể xem như là một nhánh quan trọng của AI - trí tuệ nhân tạo. Khái niệm này, được Arthur Samuel định nghĩa lần đầu tiên vào năm 1959 là lĩnh vực "cung cấp cho máy tính khả năng học mà không cần lập trình tường minh", đã trở thành một công cụ thiết yếu trong nhiều lĩnh vực như phân tích dữ liệu lớn, thị giác máy tính và xử lý ngôn ngữ tự nhiên nhờ vào sự tăng trưởng của sức mạnh tính toán và lượng dữ liệu có sẵn. Theo định nghĩa của [30] thì học máy là thuật ngữ chung để chỉ việc máy tính học từ dữ liệu. Đây là một nhánh của Trí tuệ nhân tạo (AI), trong đó các thuật toán được sử dụng để thực hiện một nhiệm vụ cụ thể mà không cần lập trình rõ ràng – thay vào đó, chúng nhận diện các mẫu trong dữ liệu và đưa ra dự đoán dựa trên những gì đã học khi có dữ liệu mới. Máy học (ML) là một cách hiệu quả để tự động hóa các nhiệm vụ phức tạp, vượt xa hơn so với tự động hóa dựa trên quy tắc.

2.1.2. Các phương pháp học máy

Học máy có thể được phân loại thành nhiều phương pháp khác nhau, tùy thuộc vào dữ liệu đầu vào và mục đích học tập và thường được chia thành ba phương pháp chính bao gồm: học có giám sát (supervised learning), học không giám sát (unsupervised learning), và học tăng cường (reinforcement learning). Trong học có giám sát, mô hình được huấn luyện trên một tập dữ liệu đã gắn nhãn để dự đoán đầu ra từ đầu vào mới, chẳng hạn như hồi quy tuyến tính hoặc phân loại. Ngược lại, học không giám sát nhằm khám phá cấu trúc ẩn trong dữ liệu không gắn nhãn, ví dụ như thuật toán phân cụm K-means. Học tăng cường, mặt khác, dựa trên việc tối ưu hóa hành động trong một môi trường thông qua thử nghiệm và phản hồi, thường được áp dụng trong robot hoặc trò chơi điện tử.

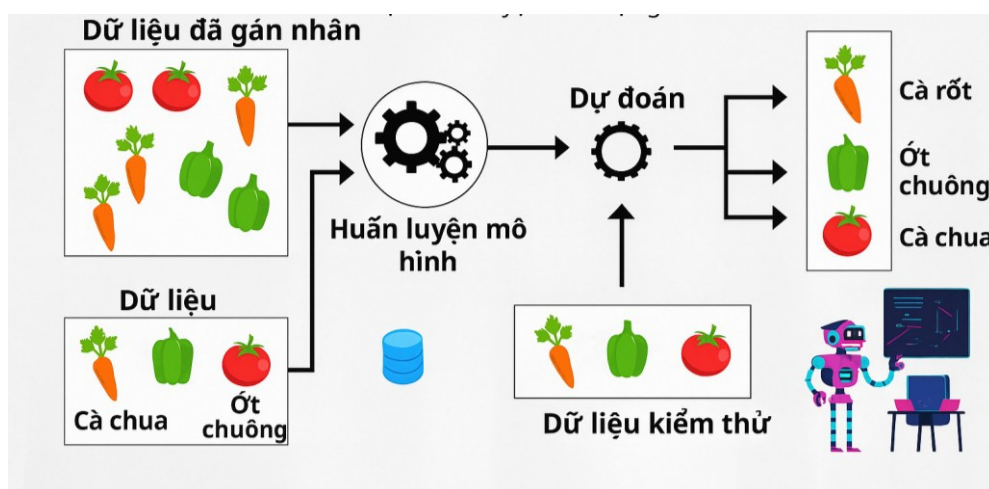
- Học có giám sát: Phù hợp khi có dữ liệu gắn nhãn và mục tiêu là dự đoán chính xác (ví dụ: phân loại hình ảnh).

- Học không giám sát: Hữu ích cho việc khám phá dữ liệu và không cần nhãn (ví dụ: phân nhóm người dùng mạng xã hội).

- Học tăng cường: Tốt cho các bài toán ra quyết định tuần tự trong môi trường động (ví dụ: xe tự hành).

2.1.2.1. Học máy có giám sát (Supervised Machine Learning)

Học có giám sát là một phương pháp trong lĩnh vực học máy, mô hình trong phương pháp này được huấn luyện dựa trên một tập dữ liệu đã được gán nhãn sẵn. Ở đây mỗi điểm dữ liệu đầu vào (input) đều được liên kết với một kết quả đầu ra (output) tương ứng mà mô hình cần học. Mục đích chính là xây dựng một hàm ánh xạ có khả năng dự đoán đầu ra phù hợp cho các dữ liệu đầu vào mới mà mô hình chưa từng gặp trước đó. Tập dữ liệu này bao gồm các ví dụ huấn luyện, mỗi ví dụ chứa một hoặc nhiều đầu vào và đầu ra mong muốn (tín hiệu giám sát). Về mặt toán học, mỗi ví dụ huấn luyện có thể được biểu diễn như một vector đặc trưng và tập dữ liệu huấn luyện là một ma trận các vector này. Thông qua quá trình tối ưu hóa lặp đi lặp lại một hàm mục tiêu, thuật toán học có giám sát tìm ra một hàm có thể ánh xạ đầu vào mới đến đầu ra tương ứng. Một thuật toán học thành công là thuật toán có thể dự đoán chính xác đầu ra cho cả những đầu vào không có trong dữ liệu huấn luyện, và độ chính xác này được cải thiện theo thời gian.



Hình 2.1: Ví dụ về học máy có giám sát

Ta thường chia học máy có giám sát làm hai loại: classification – phân loại và regression – hồi quy. Đối với các bài toán phân loại, đầu ra là các nhãn định danh như “tích cực” hoặc “tiêu cực”. Mục tiêu của phân loại là dự đoán một biến mục tiêu mang tính rời rạc, thường biểu thị dưới dạng nhãn hoặc lớp cụ thể. Ví dụ, có thể kể đến việc xác định xem một email có phải là thư rác, phân loại hình ảnh theo từng loài động vật, hoặc chẩn đoán bệnh lý dựa trên các triệu chứng lâm sàng. Mô hình sẽ học cách ánh xạ dữ liệu đầu vào vào một trong các lớp đã xác định trước. Ngược lại, trong bài toán hồi quy, đầu ra là một giá trị liên tục. Chẳng

hạn, có thể dự đoán giá bán của một ngôi nhà dựa vào diện tích, vị trí và số phòng ngủ; ước lượng nhiệt độ ngày mai từ dữ liệu thời tiết trước đó; hoặc dự báo doanh thu dựa trên mức đầu tư vào quảng cáo.

Các thuật toán phổ biến hay được sử dụng rộng rãi trong mô hình học máy có giám sát [31]:

- Mạng nơ-ron nhân tạo (Neural Networks): Mô phỏng hoạt động của não bộ, đặc biệt hữu ích trong xử lý hình ảnh và ngôn ngữ.

- Mô hình phân lớp Naive Bayes: Dự đoán xác suất dựa trên lý thuyết Bayes, hiệu quả với bài toán phân loại văn bản.

- Hồi quy tuyến tính (Linear Regression): Dự đoán giá trị liên tục dựa trên mối quan hệ tuyến tính giữa các biến đầu vào và đầu ra.

- Hồi quy logistic (Logistic Regression): Thường được sử dụng cho các bài toán phân loại nhị phân (ví dụ: spam hay không spam).

- Rừng ngẫu nhiên (Random Forest): Tập hợp nhiều cây quyết định, tăng cường độ chính xác và giảm thiểu hiện tượng overfitting.

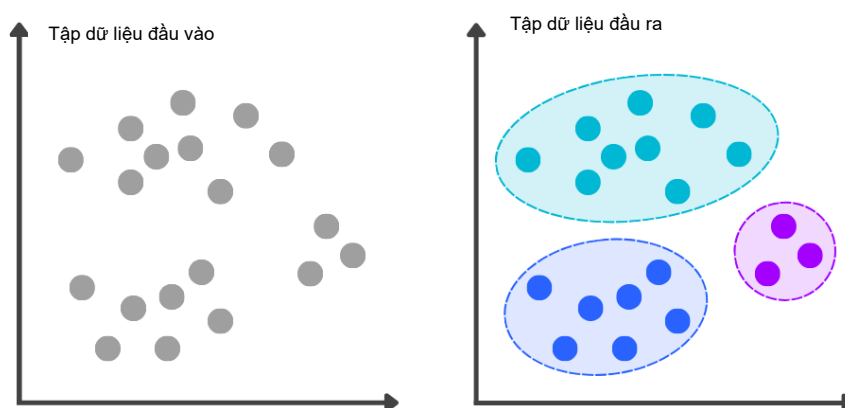
- Máy hỗ trợ vectơ (Support Vector Machine - SVM): Tìm ra đường biên tối ưu để phân loại dữ liệu với khoảng cách lớn nhất giữa các nhóm dữ liệu.

2.1.2.2. Học máy không giám sát (Unsupervised Machine Learning)

Học máy không giám sát là một nhánh quan trọng của học máy, nơi các thuật toán phân tích và phân cụm dữ liệu chưa được gán nhãn. Không giống như học có giám sát, phương pháp này không yêu cầu con người cung cấp nhãn cho dữ liệu. Thay vào đó, hệ thống tự động phát hiện các mẫu, cụm hoặc mối quan hệ ẩn trong tập dữ liệu, từ đó đưa ra những chiến lược kinh doanh phù hợp [31]. Học máy không giám sát đôi khi còn được gọi là khám phá lớp. Một trong những điểm khác biệt chính giữa học máy không giám sát và máy học có giám sát là không có bộ đào tạo nào cho học máy trước và do đó, không có vai trò rõ ràng cho xác nhận chéo. Sự khác biệt quan trọng thứ hai là mặc dù hầu hết các thuật toán được phân cụm diễn đạt theo tiêu chí tối ưu, nhưng thường không có gì đảm bảo rằng giải pháp tối ưu toàn cầu đã thu được [32].

Phân cụm là một phương pháp học máy không giám sát, trong đó các điểm dữ liệu được tổ chức thành các nhóm hoặc cụm, bao gồm các điểm dữ liệu tương tự. Điều này phân chia chúng sao cho mỗi điểm dữ liệu có ít hoặc không có điểm tương đồng với một cụm khác. Phân cụm là một nhiệm vụ được thực hiện bởi các

thuật toán học máy cụ thể, quét một tập dữ liệu và đặt từng điểm dữ liệu trong một cụm các điểm dữ liệu khác có các tính năng tương tự [33]. Các thuật toán thường được sử dụng trong phân cụm tùy theo từng mục đích sử dụng có thể kể đến đó là: Phân cụm K-means là một trong những thuật toán được sử dụng rộng rãi nhất cho khả năng ứng dụng cho nhiều mục đích, tốc độ và dễ thực hiện; phân cụm phân cấp là thuật toán trong hệ thống thực hiện phân cấp các cụm được tạo ra; phân cụm dịch chuyển trung bình khi các điểm dữ liệu được phân cụm dựa trên khoảng cách gần với một tâm, tuy nhiên tâm di chuyển lặp đi lặp lại về phía điểm có mật độ tối đa,...



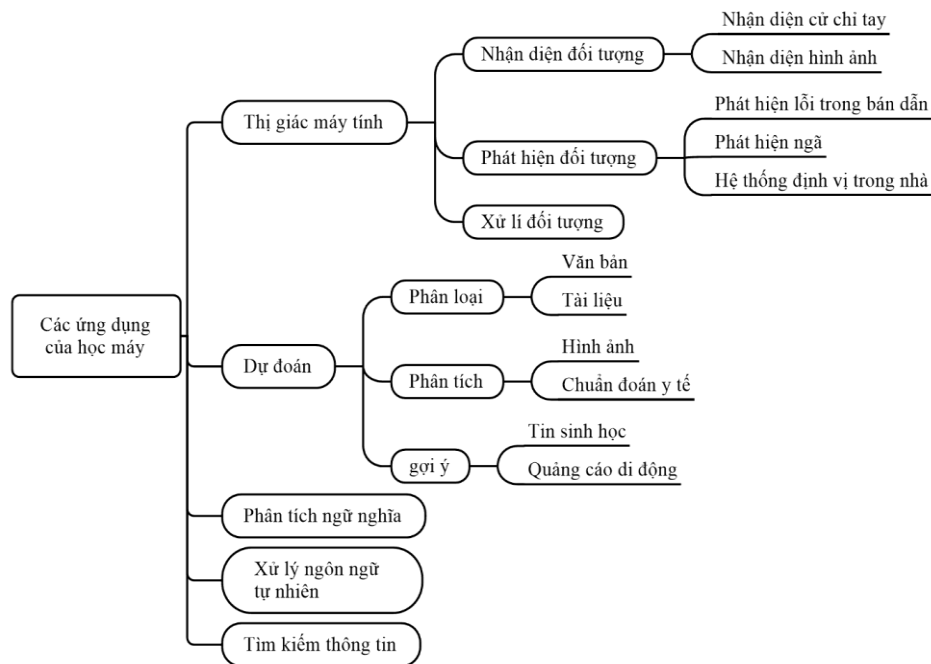
Hình 2.2: Phân cụm các điểm dữ liệu [33]

2.1.2.3. Học tăng cường (Reinforcement Learning)

Học tăng cường là một phương pháp học máy dựa trên việc một tác nhân (agent) tương tác với môi trường, học cách đưa ra quyết định tối ưu thông qua thử nghiệm và nhận phản hồi dưới dạng phần thưởng hoặc hình phạt, từ đó điều chỉnh chiến lược hành động của mình. Không giống học có giám sát hay không giám sát, học tăng cường không yêu cầu dữ liệu huấn luyện được gán nhãn sẵn mà thay vào đó dựa trên chính sách (policy) để tối đa hóa tổng phần thưởng tích lũy theo thời gian. Khái niệm này được mô tả chi tiết bởi Sutton và Barto, với các thành phần chính bao gồm trạng thái (state), hành động (action), và hàm phần thưởng (reward function) [34]. Các thuật toán phổ biến trong học tăng cường bao gồm Q-learning, học sâu tăng cường (Deep Reinforcement Learning), Deep Q-Networks (DQN). Phương pháp này thường được sử dụng trong các bài toán điều khiển, robot học và lý thuyết trò chơi, lý thuyết điều khiển, tối ưu hóa dựa trên mô phỏng, hệ thống đa tác nhân, trí thông minh bầy đàn, thống kê và thuật toán di truyền. Tuy nhiên, học tăng cường đối mặt với thách thức về tính ổn định và thời gian huấn luyện, đặc biệt khi không gian trạng thái hoặc hành động quá lớn.

2.1.3. Các ứng dụng của học máy

Machine Learning – học máy được xem như là một công cụ mạnh mẽ, được ứng dụng rộng rãi trong nhiều lĩnh vực từ công nghệ đến khoa học và xã hội. Sự phát triển của các thuật toán ML, cùng với sự gia tăng về dữ liệu và sức mạnh tính toán, đã thúc đẩy các ứng dụng mang tính đột phá, giải quyết các bài toán phức tạp mà trước đây khó có thể thực hiện. Trong [35] đã đề cập đến các ứng dụng của học máy trong nhiều lĩnh vực có thể kể đến như: thị giác máy tính, dự đoán, phân tích ngữ nghĩa, xử lý ngôn ngữ tự nhiên và truy xuất thông tin.



Hình 2.3: Các ứng dụng của học máy [35]

- Đối với thị giác máy tính (Computer Vision): Sử dụng trong việc nhận diện đối tượng, phát hiện đối tượng và xử lý đối tượng. Các mạng nơ-ron tích chập (Convolutional Neural Networks – CNN) là nền tảng cho nhiều ứng dụng trong lĩnh vực này [3]. Các ứng dụng tiêu biểu có thể kể đến như nhận diện khuôn mặt, Chuẩn đoán y tế, xe tự hành.

- Dự đoán (Prediction): Các tiểu miền khác nhau ở đây là phân loại, phân tích và gợi ý. Phân loại văn bản, phân loại tài liệu, phân tích hình ảnh, chuẩn đoán y tế, dự đoán phát hiện xâm nhập mạng và dự đoán tấn công từ chối dịch vụ đã được thực hiện thành công bằng cách sử dụng học máy [35].

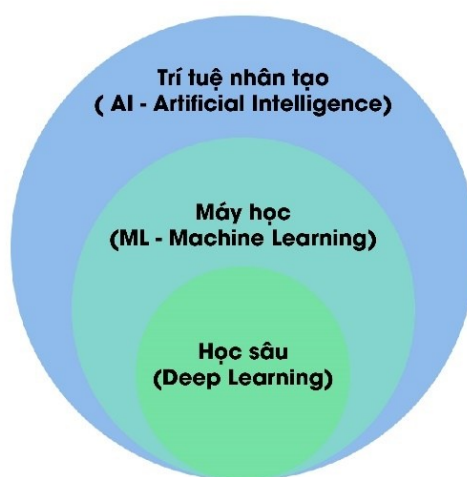
- Phân tích ngữ nghĩa, xử lý ngôn ngữ tự nhiên và tìm kiếm thông tin: Phân tích ngữ nghĩa là quá trình liên kết các cấu trúc cú pháp từ các đoạn văn, câu, từ đến cấp độ có thể hiểu được ý nghĩa tổng thể của văn bản. Xử lý ngôn ngữ tự

nhiên là cách lập trình của máy tính để xử lý chính xác dữ liệu của ngôn ngữ tự nhiên. Tìm kiếm thông tin khoa học là khoa học tìm kiếm thông tin trong tài liệu, tìm kiếm tài liệu và tìm kiếm siêu dữ liệu mô tả dữ liệu cũng như các cơ sở dữ liệu âm thanh và hình ảnh [35].

2.2. Học sâu (Deep Learning)

2.2.1. Khái niệm học sâu

Học sâu là một nhánh của học máy [35]. Học sâu sử dụng các mạng nơ-ron nhân tạo sâu, “sâu” là thuật ngữ chỉ số lớp trong một mạng nơ-ron, như vậy học sâu là mạng nơ-ron với nhiều lớp và tham số. Hầu hết các phương pháp học sâu đều sử dụng cấu trúc mạng nơ-ron. Do đó, nó cũng được gọi là mạng nơ-ron sâu.



Hình 2.4: Mối quan hệ giữa học sâu, học máy và trí tuệ nhân tạo [36]

Học sâu sử dụng các mạng nơ-ron nhân tạo nhiều lớp để mô hình hóa và giải quyết các bài toán phức tạp trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, và phân tích dữ liệu tuần tự. Được xây dựng dựa trên mô phỏng cấu trúc và chức năng của mạng nơ-ron sinh học, học sâu cho phép hệ thống học các biểu diễn đặc trưng phân cấp trực tiếp từ dữ liệu thô mà không cần can thiệp thủ công đáng kể.

Sự khác biệt giữa học sâu và học máy đó là các thuật toán học máy tận dụng dữ liệu có cấu trúc, được gắn nhãn để đưa ra dự đoán. Các tính năng cụ thể được xác định từ dữ liệu đầu vào cho mô hình và được tổ chức thành các bảng. Dữ liệu thường trải qua một số xử lý trước để sắp xếp thành một định dạng có cấu trúc. Còn đối với học sâu nó loại bỏ một số bước xử lý trước dữ liệu. Các thuật toán này có thể nhập và xử lý dữ liệu phi cấu trúc, như văn bản và hình ảnh. Đồng thời, chúng tự động hóa việc trích xuất tính năng, do đó, giảm bớt sự tham gia của con

người trong một số công đoạn. Ví dụ, có một bộ ảnh về các vật nuôi khác nhau cần phân loại. Các thuật toán học sâu có thể tự động xác định đặc điểm nào (ví dụ như tai) là quan trọng nhất để phân biệt từng loài động vật. Trong khi đó, để làm được điều này, học máy cần có sự hỗ trợ của con người [36].

2.2.2. Mạng nơ-ron

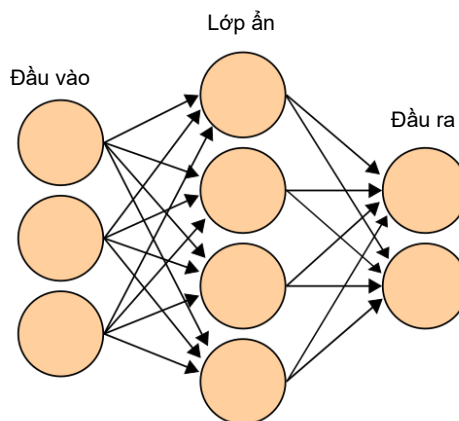
Có nhiều loại mạng nơ-ron khác nhau, mỗi loại đều có cấu trúc và chức năng riêng, phù hợp với các chức năng và tác vụ khác nhau. Một số loại mạng nơ-ron phổ biến:

2.2.2.1. Mạng nơ-ron nhân tạo

Mạng thần kinh nhân tạo (ANNs) là mô hình tính toán được lấy cảm hứng từ cấu trúc sinh học của não bộ. Nó gồm một nhóm các nơ-ron nhân tạo được nối với nhau và xử lý thông tin bằng cách truyền các kết nối và tính giá trị mới tại các nút. Một mạng thần kinh nhân tạo cơ bản bao gồm: Lớp đầu vào (input layer) để tiếp nhận các dữ liệu thô, lớp ẩn (hidden layer) để thực hiện biến đổi phi tuyến và lớp đầu ra (output layer) để tạo kết quả dự đoán. Mỗi quan hệ giữa các lớp được mô tả:

$$y = f\left(\sum_{i=1}^n \omega_i x_i + b\right) \quad (2.1)$$

Trong đó: ω_i là trọng số, x_i là đầu vào, b là độ lệch và f là hàm kích hoạt



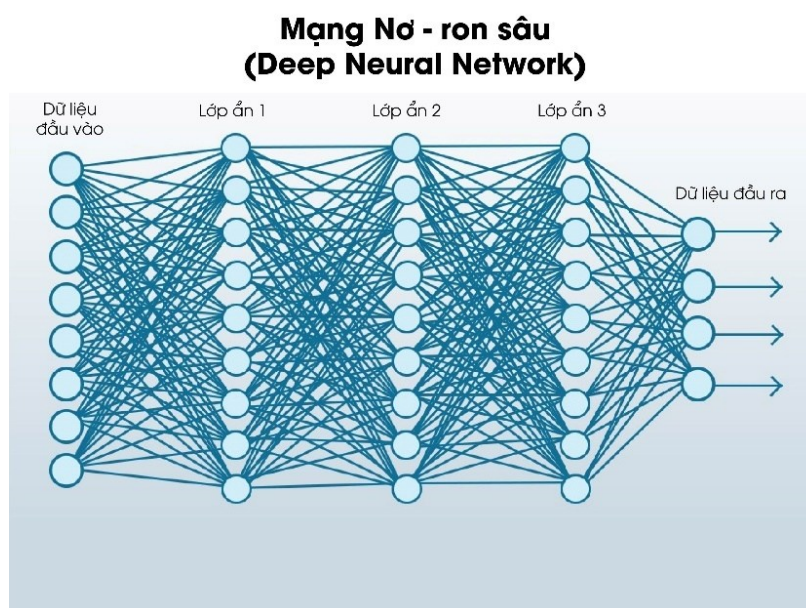
Hình 2.5: Ví dụ về mạng thần kinh nhân tạo (ANNs)

Ban đầu, mục tiêu của phương pháp mạng nơ-ron nhân tạo (ANN) là xử lý các vấn đề theo cách tương tự như cách hoạt động của não người. Tuy nhiên, theo thời gian, trọng tâm đã chuyển sang việc thực hiện các nhiệm vụ cụ thể, khiến cho phương pháp này dần rời xa sự mô phỏng sinh học. Mạng nơ-ron nhân tạo đã được ứng dụng trong nhiều lĩnh vực khác nhau như thị giác máy tính, nhận dạng

tiếng nói, dịch ngôn ngữ tự động, lọc nội dung trên mạng xã hội, trò chơi điện tử và chẩn đoán y tế.

2.2.2.2. Mạng nơ-ron sâu

Deep Neural Networks (DNNs), hay còn gọi là mạng nơ-ron sâu, là sự phát triển nâng cao từ mạng nơ-ron nhân tạo (ANNs) với nhiều lớp ẩn cho phép học các đặc trưng phân cấp từ dữ liệu phức tạp. Sự phát triển của DNNs gắn liền với sự gia tăng sức mạnh tính toán (GPU/TPU) và sự sẵn có của dữ liệu lớn, giúp giải quyết các bài toán mà các phương pháp truyền thống không thể đạt hiệu suất cao [3].



Hình 2.6: Mô hình mạng nơ-ron sâu [36]

Đặc điểm cốt lõi của mạng nơ-ron sâu là có nhiều lớp ẩn, thường có từ 3 lớp ẩn trở lên và mỗi lớp ẩn sẽ học các đặc trưng ở mức độ trừu tượng khác nhau (từ cạnh, hình dạng đến các đối tượng phức tạp). Bên cạnh đó, DNNs không yêu cầu trích chọn đặc trưng thủ công như trong học máy truyền thống và có khả năng mô hình hóa phi tuyến mạnh nhờ kết hợp nhiều hàm kích hoạt phi tuyến (ReLU, Sigmoid, Tanh).

DNN hoạt động cũng giống như việc đang lắp ráp một thứ gì đó phức tạp và ta cần bắt đầu từ những chi tiết, bộ phận nhỏ nhất rồi ghép chúng thành các cụm lớn hơn và cuối cùng là hoàn thiện sản phẩm. Ở các lớp đầu tiên DNN học cách nhận biết các đặc trưng cơ bản (ví dụ một bức ảnh ta có thể học cách để nhận ra các cạnh, góc hoặc các vùng màu sắc). Các lớp sâu hơn sẽ nhận đầu vào từ lớp trước đó và kết hợp các đặc trưng đơn giản này để học các khái niệm phức tạp hơn (ví dụ như kết hợp các góc cạnh để nhận diện ra mắt, mũi). Các lớp cuối

cùng kết hợp các bộ phận này lại để nhận dạng các đối tượng phức tạp như toàn bộ khuôn mặt người hay một con vật cụ thể.

2.2.2.3. Mạng nơ-ron tích chập

Mạng Nơ-ron Tích chập (CNN) là một dạng kiến trúc học sâu chuyên dụng, được phát triển nhằm xử lý hiệu quả các loại dữ liệu có cấu trúc dạng lưới, điển hình là hình ảnh [37]. CNN đã chứng minh khả năng vượt trội trong nhiều nhiệm vụ như phân loại hình ảnh, phát hiện đối tượng và phân đoạn ảnh.

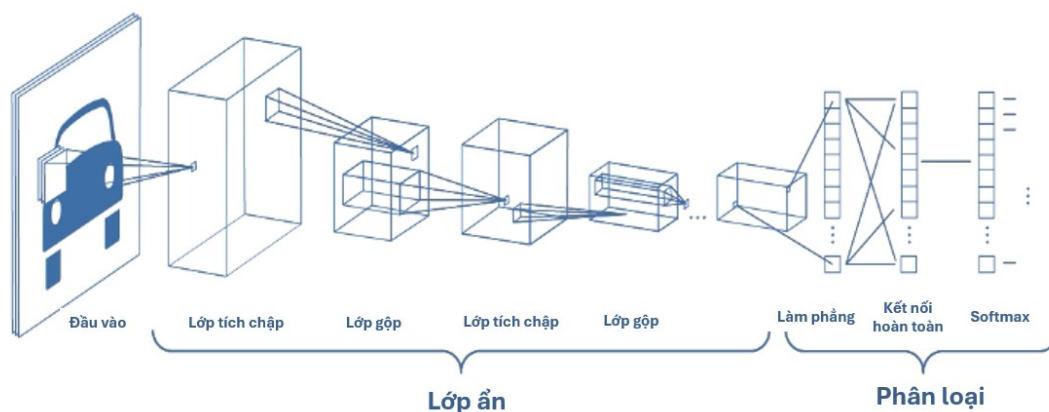
Một mạng CNN tiêu biểu thường bao gồm một số loại lớp chính, trong đó quan trọng nhất là:

- Lớp tích chập (Convolutional Layer): Đây là thành phần then chốt trong cấu trúc CNN. Lớp này sử dụng các bộ lọc (kernels) được học trong quá trình huấn luyện để trượt qua từng vùng nhỏ của ảnh đầu vào — gọi là vùng tiếp nhận (receptive field) — và thực hiện phép tích chập. Kết quả là các bản đồ đặc trưng (feature maps) được tạo ra, giúp mô hình nhận diện các đặc điểm không gian cục bộ như cạnh, màu, hoặc hoa văn.

- Lớp phi tuyến tính (Activation Layer): Thường là hàm ReLU (Rectified Linear Unit), được áp dụng sau lớp tích chập để đưa tính phi tuyến vào mô hình, giúp mạng học được các mối quan hệ phức tạp hơn.

- Lớp gộp (Pooling Layer): Lớp này thực hiện việc giảm chiều không gian của các bản đồ đặc trưng, giúp giảm số lượng tham số, giảm khối lượng tính toán và kiểm soát hiện tượng quá khớp (overfitting). Các kỹ thuật gộp phổ biến nhất là Gộp cực đại (Max Pooling).

- Lớp kết nối đầy đủ (Fully Connected Layer): Sau một chuỗi các lớp tích chập và gộp, các bản đồ đặc trưng cuối cùng được làm phẳng thành một vector và đưa vào các lớp kết nối đầy đủ để thực hiện nhiệm vụ cuối cùng, ví dụ như phân loại.



Hình 2.7: Cấu trúc của một mạng nơ-ron tích chập [38]

CNN được áp dụng rộng rãi trong nhiều lĩnh vực, một số lĩnh vực nổi bật có thể kể đến như:

- Nhận dạng hình ảnh: CNN đạt được hiệu suất vượt trội trong các nhiệm vụ như phân loại hình ảnh và phát hiện đối tượng. Mô hình AlexNet [37] đã đánh dấu bước ngoặt khi giành chiến thắng trong cuộc thi ImageNet năm 2012.

- Xử lý ngôn ngữ tự nhiên (NLP): CNN được sử dụng để trích xuất đặc trưng từ văn bản, chẳng hạn như trong phân tích cảm xúc hoặc phân loại văn bản [39].

- Y học: CNN hỗ trợ chẩn đoán bệnh thông qua phân tích hình ảnh y tế, như phát hiện ung thư từ ảnh chụp X-quang.

Tính chất của mạng nơ - ron tích chập [31]:

- Tính kết nối trượt: Khác với các mạng nơ-ron thông thường, mạng nơ-ron tích chập không kết nối tới toàn bộ hình ảnh mà chỉ kết nối tới từng vùng địa phương (local region) hoặc vùng nhận thức (receptive field) có kích thước bằng kích thước bộ lọc của hình ảnh đó. Các bộ lọc sẽ trượt theo chiều của ảnh từ trái qua phải và từ trên xuống dưới đồng thời tính toán các giá trị tích chập và điền vào bản đồ kích hoạt (activation map) hoặc bản đồ đặc trưng (feature map).

- Tính chia sẻ kết nối và kết nối cục bộ:

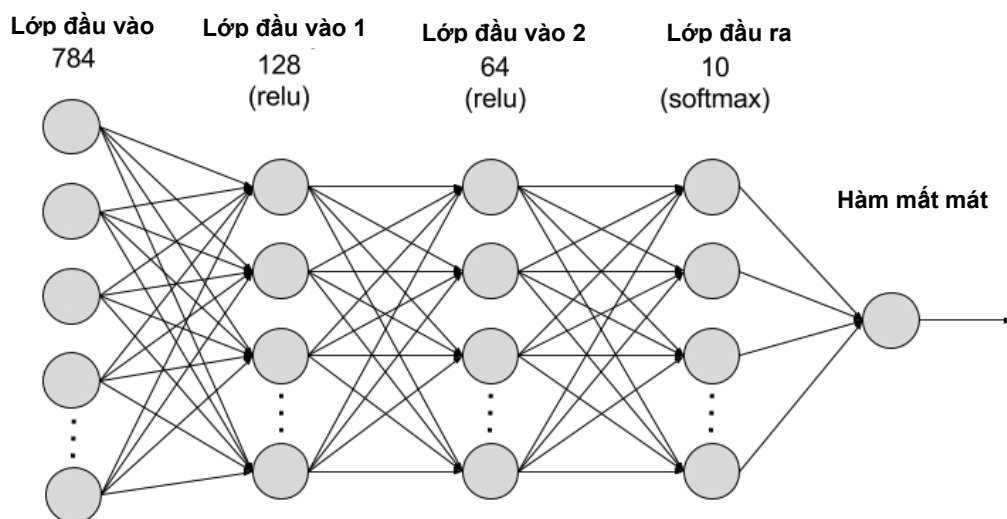
- Tính tổng hợp.

- Độ phức tạp phát hiện hình ảnh tăng dần: Ở tầng đầu tiên, hình ảnh mà chúng ta có chỉ là những giá trị pixels. Sau khi đi qua tầng thứ 2 máy tính sẽ nhận diện được các hình dạng cạnh, rìa và các đường nét đơn giản được gọi là đặc trưng bậc thấp (low level). Càng ở những tầng tích chập về sau càng có khả năng phát hiện các đường nét phức tạp, đã rõ ràng hình thù và thậm chí là cấu thành vật thể, đây được gọi là những đặc trưng bậc cao (high level). Máy tính sẽ học từ tầng cuối cùng để nhận diện nhãn của hình ảnh.

2.2.2.4. Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy (RNN) là một mô hình học sâu được đào tạo để xử lý và chuyển đổi đầu vào dữ liệu tuần tự thành đầu ra dữ liệu tuần tự cụ thể. RNN sở hữu một bộ nhớ nội tại thông qua các kết nối hồi quy cho phép chúng duy trì và tận dụng các thông tin từ các bước thời gian trước đó để thông báo cho các tính toán ở hiện tại. Chính vì thế RNN phù hợp với các dữ liệu như câu từ hoặc dữ liệu chuỗi thời gian trong đó các thành phần tuần tự tương quan với nhau dựa trên ngữ nghĩa phức tạp và quy tắc cú pháp. RNN có thể áp dụng trong các tác vụ liên quan

đến chuỗi thời gian và xử lý ngôn ngữ tự nhiên có thể kể đến như nhận dạng chữ viết tay, nhận dạng tiếng nói có tính chất kết nối không phân đoạn.



Hình 2.8: Sơ đồ của một RNN [40]

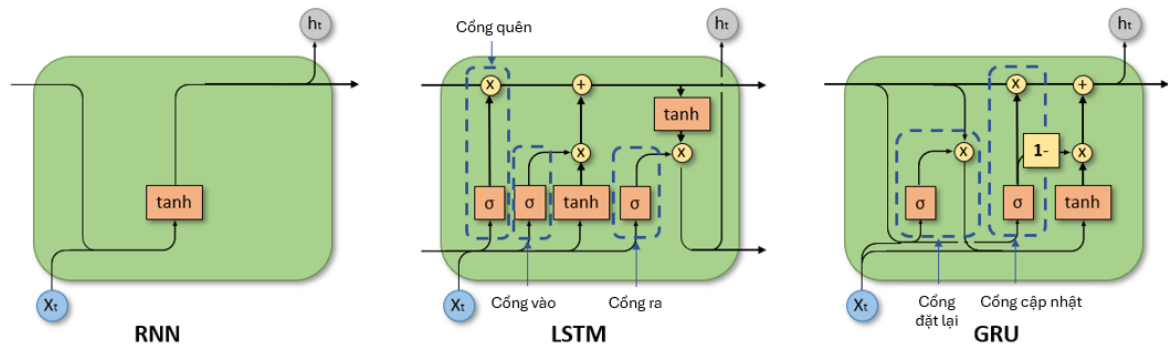
RNN được tạo thành từ các nơ-ron: các nút xử lý dữ liệu kết hợp cùng nhau để thực hiện các tác vụ phức tạp. Các nơ-ron được tổ chức dưới dạng lớp đầu vào, đầu ra và ẩn. Lớp đầu vào nhận thông tin để xử lý và lớp đầu ra cung cấp kết quả. Quá trình xử lý dữ liệu, phân tích và dự đoán diễn ra trong lớp ẩn [40].

Các biến thể nâng cao của RNN: Để giải quyết các hạn chế của RNN đơn giản, hai biến thể nổi bật đã được đề xuất và sử dụng rộng rãi là mạng bộ nhớ dài-ngắn hạn (Long Short-Term Memory - LSTM) và đơn vị hồi quy có cổng (Gated Recurrent Unit - GRU).

- Bộ nhớ dài – ngắn hạn (Long short-term memory – LSTM) là một dạng RNN cải tiến, nó giải quyết vấn đề mất mát ở các RNN thông thường. LSTM giới thiệu một thành phần mới gọi là trạng thái ô nhớ (cell gate) và hoạt động như một băng chuyền thông tin. Mạng có thể đọc, ghi và xóa thông tin khỏi ô nhớ này thông qua các cấu trúc gọi là cổng (gate). Một đơn vị LSTM gồm có 3 cổng chính đó là: cổng quên (forget gate) quyết định thông tin nào sẽ bị loại bỏ khỏi trạng thái bộ nhớ, cổng vào (input gate) quyết định thông tin nào sẽ được lưu trữ trong trạng thái ô nhớ, cổng ra (output gate) quyết định phần nào của ô nhớ sẽ được xuất ra dưới dạng trạng thái ẩn.

- Đơn vị hồi quy có cổng (Gated Recurrent Unit - GRU) là một biến thể của LSTM với kiến trúc đơn giản hơn. Nó kết hợp cổng quên và cổng chính vào thành

một gọi là cổng cập nhật (update gate) duy nhất và cũng gộp trạng thái ô nhớ và trạng thái ẩn là một.



Hình 2.9: Kiến trúc mạng RNN, LSTM và GRU [41]

2.2.3. Nguyên lý của học sâu

Học sâu là một nhánh của học máy, phát triển từ mạng nơ-ron nhân tạo – cấu trúc bao gồm các đơn vị tính toán (nơ-ron) được sắp xếp thành các tầng: tầng đầu vào, các tầng ẩn, và tầng đầu ra. Mô hình này lấy cảm hứng từ cách hoạt động của bộ não con người, với mỗi nơ-ron hoạt động như một tế bào thần kinh nhân tạo. Mỗi đơn vị tính toán sẽ thực hiện một phép biến đổi tuyến tính trên dữ liệu nhận được, sau đó áp dụng một hàm kích hoạt phi tuyến để tạo ra đầu ra. Khi mạng có nhiều tầng ẩn, nó được gọi là mạng "sâu", phản ánh khả năng mô hình hóa các hàm số phức tạp thông qua việc học các đặc trưng phân cấp – từ các mẫu đơn giản ở tầng dưới đến các khái niệm trừu tượng hơn ở tầng trên.

Quá trình huấn luyện deep learning tối ưu hóa hàm mục tiêu thông qua thuật toán lan truyền ngược (Backpropagation) để tối ưu hóa các tham số của mạng (trọng số và độ lệch) dựa trên việc giảm thiểu một hàm mất mát, chẳng hạn như sai số bình phương trung bình hoặc entropy chéo. Thuật toán gradient descent hoặc các biến thể cải tiến (ví dụ: Adam, RMSprop) được áp dụng để cập nhật tham số dựa trên gradient của hàm mất mát. Khả năng học đặc trưng tự động của học sâu loại bỏ nhu cầu thiết kế đặc trưng thủ công, một hạn chế của các phương pháp học máy truyền thống.

Thay vì trích xuất đặc trưng thủ công, các mô hình học sâu có khả năng tự động học các biểu diễn phân cấp của dữ liệu [3]. Đối với robot di động, khả năng này cho phép robot "hiểu" được môi trường xung quanh từ dữ liệu cảm biến thô (như hình ảnh từ camera), hình ảnh thu thập được từ vệ tinh, từ đó xây dựng các bản đồ số có chứa thông tin ngữ nghĩa phong phú.

2.2.4. Image Segmentation - Thuật toán phân đoạn ảnh

Image Segmentation hay phân đoạn hình ảnh là bài toán sẽ phân chia một hình ảnh đầu vào thành nhiều vùng có ý nghĩa sao cho mỗi vùng sẽ đại diện cho một đối tượng hoặc một phần của đối tượng. Khác với bài toán nhận diện vật thể (Object detection) thì bài toán phân đoạn ảnh cho độ chính xác ở mức cao hơn khi nó yêu cầu nhãn dự báo đúng tới từng pixel trong khi bài toán nhận diện vật thể chỉ tập trung vào xác định và định vị các đối tượng trong một hình ảnh và phân loại chúng. Trong thiết kế bản đồ, mục tiêu chính của việc phân đoạn ảnh là trích xuất các khu vực quan trọng giúp hệ thống hiểu rõ hơn về nội dung ảnh để phục vụ cho các phân tích như nhận diện đối tượng. Phân đoạn ảnh cho phép trích xuất chính xác các khu vực có ý nghĩa trong ảnh đầu vào, chẳng hạn như tường, cửa, sàn, vật cản, hoặc lối đi. Các thông tin này giúp robot nhận biết được không gian khả thi để di chuyển, từ đó xây dựng bản đồ môi trường có cấu trúc rõ ràng, hỗ trợ thuật toán thiết kế quỹ đạo hiệu quả và an toàn.

Nhận diện vật thể



Phân đoạn hình ảnh

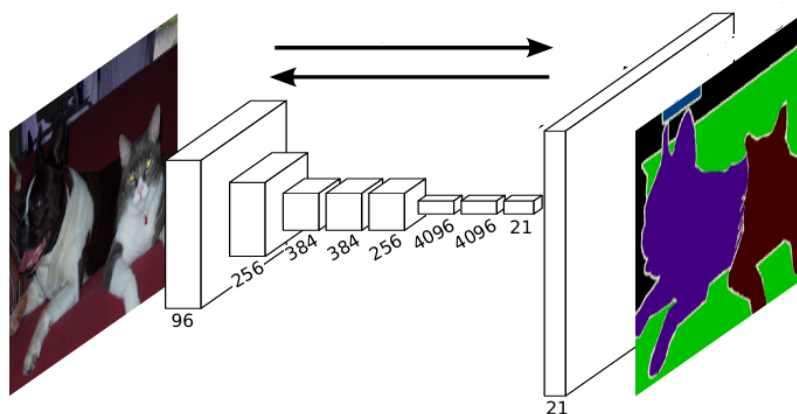


Hình 2.10: Vật thể được xác định trong bài toán nhận diện vật thể và bài toán phân đoạn ảnh [42]

2.2.4.1. Phân đoạn sử dụng mạng tích chập hoàn toàn (Fully Convolutional Network – FCN)

Theo [43], mạng nơ-ron tích chập hoàn toàn (FCN) là một kiến trúc mạng nơ-ron đột phá, đặt nền móng cho các phương pháp học sâu hiện đại trong lĩnh vực phân đoạn ngữ nghĩa. Bằng cách loại bỏ các lớp kết nối đầy đủ (fully connected layers) truyền thống và các lớp gộp (pooling layers) xếp chồng lên nhau, FCN có khả năng xử lý ảnh đầu vào với kích thước bất kỳ và tạo ra một bản đồ

phân đoạn có cùng kích thước, gán nhãn cho từng pixel trên ảnh. Khác với các kiến trúc mạng nơ-ron tích chập truyền thống, mạng nơ-ron tích chập hoàn toàn sử dụng các tầng tích chập chuyển vị để khôi phục kích thước chiều cao và chiều rộng của ảnh xạ đặc trưng trung gian về đúng kích thước ban đầu của ảnh đầu vào. Cách tiếp cận này cho phép tạo ra các dự đoán có sự tương ứng từng điểm theo không gian với ảnh gốc. Ở mỗi vị trí không gian cụ thể, đầu ra theo trục kênh thể hiện lớp dự đoán tương ứng với điểm ảnh tại vị trí đó.



Hình 2.11: Ví dụ về phân đoạn ảnh sử dụng mạng tích chập hoàn toàn (FCN) [43]

Kiến trúc của FCN thường bao gồm hai phần chính:

- Đường xuống (Downsampling Path): Phần này hoạt động như một bộ trích xuất đặc trưng. Nó bao gồm các lớp tích chập và các lớp gộp xếp chồng lên nhau. Khi ảnh đi qua đường xuống, kích thước không gian của bản đồ đặc trưng giảm dần, trong khi chiều sâu tăng lên. Quá trình này giúp mạng học được các đặc trưng ngữ nghĩa trừu tượng của ảnh ví dụ như "cái gì" có trong ảnh.

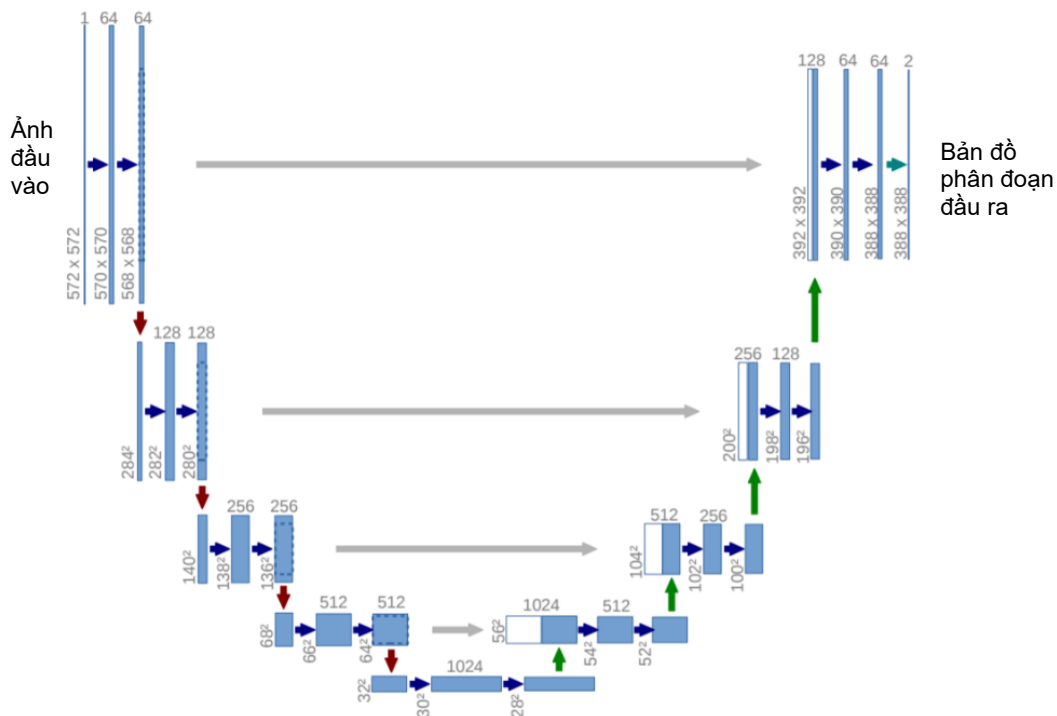
- Đường lên (Upsampling Path): Sau khi thu được bản đồ đặc trưng ở độ phân giải thấp, nhiệm vụ của đường lên là khôi phục lại kích thước không gian ban đầu của ảnh. Để làm được điều này, FCN sử dụng một kỹ thuật gọi là tích chập chuyển vị (transposed convolution) hay giải tích chập (deconvolution). Lớp này thực hiện quá trình ngược lại với lớp tích chập, giúp tăng kích thước của bản đồ đặc trưng một cách thông minh và có thể học được.

Ưu điểm của mạng tích chập hoàn toàn là có thể xử lý được đầu vào linh hoạt, có khả năng nhận ảnh với kích thước bất kỳ mà không bị giới hạn bởi các lớp kết nối đầy đủ; toàn bộ mạng có thể được huấn luyện từ đầu đến cuối một cách hiệu quả; tính toán trên toàn bộ ảnh thay vì từng vùng nhỏ giúp tăng tốc độ xử lý và nó là nguồn cảm hứng cho các kiến trúc phân đoạn tiên tiến sau này có thể kể đến như U-net, DeepLab. Bên cạnh đó, các lớp gộp ở đây làm giảm độ phân

giải từ đó có thể làm mất mát các chi tiết nhỏ trong ảnh và bản đồ phân đoạn có thể không sắc nét ở ranh giới của các đối tượng. Một nhược điểm nữa có thể kể đến đó là FCN chỉ thực hiện phân đoạn ngữ nghĩa, tức là tất cả các đối tượng thuộc cùng một lớp (ví dụ: tất cả các xe ô tô) sẽ được gán cùng một nhãn mà không phân biệt chúng là các cá thể riêng biệt.

2.2.4.2. Phân đoạn ảnh sử dụng kiến trúc học sâu U-net

U-Net là một dạng kiến trúc mạng nơ-ron tích chập (CNN) được phát triển đặc biệt cho các bài toán phân đoạn ảnh, với hiệu quả nổi bật trong lĩnh vực xử lý ảnh y tế. Kiến trúc này lần đầu tiên được đề xuất vào năm 2015 bởi Olaf Ronneberger cùng các cộng sự [42]. Nhờ cấu trúc đối xứng hình chữ U, mô hình này có khả năng học đặc trưng mạnh mẽ đồng thời bảo toàn thông tin vị trí không gian của ảnh gốc, giúp cho kết quả phân đoạn đạt độ chính xác cao.



Hình 2.12: Kiến trúc một mô hình U-net [42]

Kiến trúc U-Net được phát triển dựa trên nền tảng của mạng tích chập hoàn toàn (Fully Convolutional Network – FCN), với mục tiêu giải quyết hiệu quả các bài toán phân đoạn ảnh, đặc biệt trong lĩnh vực xử lý ảnh y tế.

Ý tưởng cốt lõi của U-Net là mở rộng cấu trúc mạng cơ truyền thống bằng cách bổ sung các lớp giải mã đối xứng. Trong các lớp này, thao tác gộp (pooling) được thay thế bằng các kỹ thuật lấy mẫu lại (upsampling), nhằm khôi phục dần

độ phân giải không gian của đặc trưng. Các lớp tích chập sau đó có khả năng học để tái tạo chi tiết đầu ra dựa trên thông tin đã được phục hồi [42].

U-Net có các cải tiến nổi bật có thể kể đến như:

- Cấu trúc mở rộng đối xứng: Khác với các mô hình trước đây như FCN vốn chỉ bao gồm nhánh mã hóa (contracting path), U-Net được thiết kế với kiến trúc đối xứng theo hình chữ U, trong đó nhánh mã hóa có nhiệm vụ trích xuất ngữ cảnh, còn nhánh giải mã (expanding path) đảm nhận việc khôi phục độ phân giải không gian và định vị chính xác các đối tượng trong ảnh. Sự đối xứng này giúp mô hình kết hợp hiệu quả giữa thông tin ngữ nghĩa và thông tin không gian.

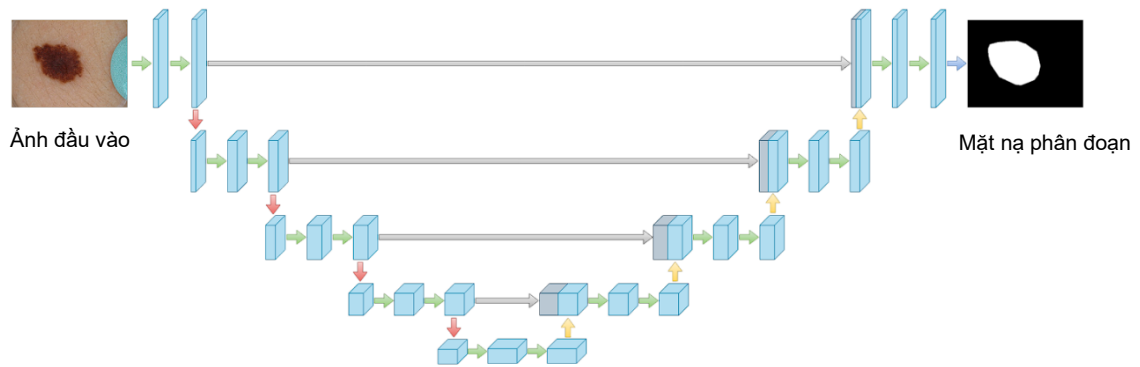
- Kết nối bỏ qua (Skip Connections): U-Net áp dụng các kết nối bỏ qua giữa các tầng tương ứng của nhánh mã hóa và giải mã. Cơ chế này cho phép truyền trực tiếp các đặc trưng không gian từ nhánh co sang nhánh mở rộng, nhằm bù đắp thông tin bị mất trong quá trình downsampling (lấy mẫu xuống), đồng thời cải thiện chất lượng tái tạo chi tiết ở đầu ra.

- Tăng cường dữ liệu (Data Augmentation): Nhằm giải quyết vấn đề thiếu hụt dữ liệu có gán nhãn trong các bài toán phân đoạn ảnh y tế, U-Net khai thác mạnh mẽ các kỹ thuật tăng cường dữ liệu như xoay, lật, dịch chuyển, và biến dạng hình học. Việc này giúp mô hình học được các đặc trưng đa dạng và tổng quát hơn mà không cần phụ thuộc vào một tập huấn luyện lớn.

U-Net bao gồm 2 phần chính đó là phần thu hẹp (Contracting path) ở bên trái và phần mở rộng (Expansive path) ở bên phải.

- Phần thu hẹp: Đóng vai trò như một bộ mã hóa (Encoder) là nhiệm vụ trích lọc đặc trưng để tìm ra bối cảnh của hình ảnh đồng thời làm giảm dần độ phân giải không gian của ảnh (ví dụ từ input kích thước 572×572 chỉ còn 32×32 và đồng thời độ sâu cũng tăng dần từ 3 lên 512). Nó sử dụng mạng tích chập (CNN) với các lớp tích chập và ghép gộp (Pooling) tuần tự.

- Phần mở rộng: Hoạt động như một bộ giải mã (Decoder) thực hiện tái tạo lại ảnh phân đoạn. Bộ giải mã áp dụng phép tăng kích thước không gian (unsampling) thông qua các phép chuyển vị tích chập (transposed convolution) hoặc nội suy từ đó tạo ra ánh xạ đầu ra (output mask) với kích thước tương tự ảnh đầu vào và gán nhãn từng pixel.



Hình 2.13: Bộ mã hóa và bộ giải mã của U-Net [44]

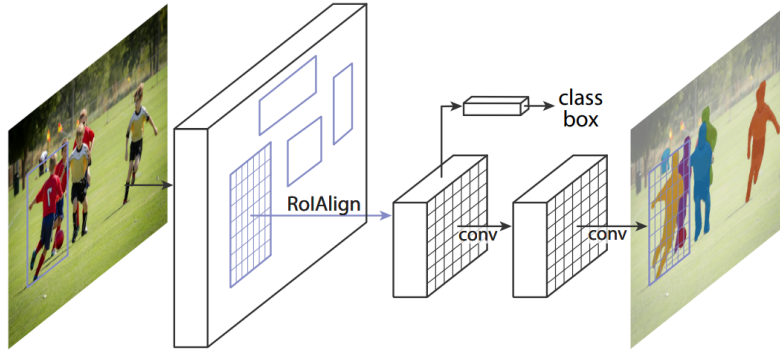
Với độ chính xác vượt trội trong các tác vụ phân đoạn ngữ nghĩa ở cấp độ pixel bằng cách khôi phục lại độ phân giải không gian của bản đồ đặc trưng ở tầng giải mã, kiến trúc này có khả năng định vị và phân định ranh giới đối tượng một cách chi tiết và chính xác. Hơn nữa, với các phiên bản được tối ưu hóa, U-Net còn đạt được tốc độ xử lý nhanh, đáp ứng yêu cầu của nhiều ứng dụng thời gian thực. Nhờ những đặc trưng này mà U-Net được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như:

- Trong y sinh, mô hình này được sử dụng để phân đoạn tế bào, mô và các cấu trúc giải phẫu phức tạp như khối u từ ảnh cộng hưởng từ (MRI) hay cắt lớp vi tính (CT).
- Trong robot học và bản đồ số, U-Net giúp xác định các khu vực có thể di chuyển, vật cản và đường đi từ hình ảnh môi trường.
- Trong lĩnh vực viễn thám, nó được dùng để phân loại và phân đoạn các loại địa hình như đất, sông ngòi, và rừng từ ảnh vệ tinh.
- Trong xử lý ảnh công nghiệp để tự động hóa quy trình kiểm tra lỗi bề mặt sản phẩm và phát hiện đối tượng.

2.2.4.3. Phân đoạn sử dụng kiến trúc Mask R-CNN

Mask R-CNN là phương pháp dò tìm hiệu quả đối với các đối tượng trong một ảnh, đồng thời tạo ra một mặt nạ phân đoạn (segmentation mask) chất lượng cao cho một thực thể và nó là một phương pháp được mở rộng từ phương pháp Faster R-CNN bằng cách thêm một nhánh để dự đoán mặt nạ đối tượng trên mỗi vùng quan tâm (Region of Interest – RoI), song song với nhánh có sẵn dùng để nhận dạng hộp giới hạn (bounding box) [45]. Mask R-CNN tương đối dễ huấn luyện và chỉ làm tăng nhẹ chi phí tính toán so với Faster R-CNN, với tốc độ xử lý đạt khoảng 5 khung hình mỗi giây (fps). Bên cạnh đó, Mask R-CNN có khả năng

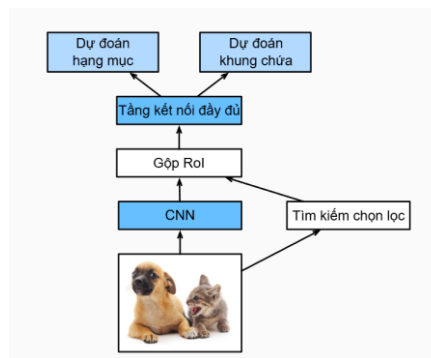
mở rộng tốt sang các nhiệm vụ khác, chẳng hạn như ước lượng tư thế cơ thể người trong cùng một kiến trúc xử lý.



Hình 2.14: Một ví dụ về phân đoạn sử dụng Mark R-CNN [45]

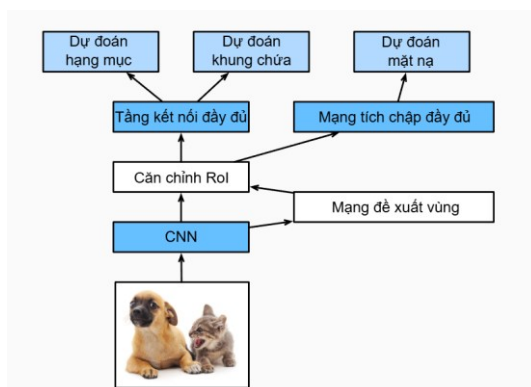
Về bản chất, Mark R-CNN có ý tưởng rất đơn giản, trong khi Faster R-CNN tạo ra hai đầu ra cho mỗi vùng đề xuất bao gồm nhãn lớp và các điều chỉnh cho hộp giới hạn thì Mask R-CNN bổ sung thêm một nhánh thứ ba để dự đoán mặt nạ phân đoạn cho đối tượng. Nhờ đó, kiến trúc này trở nên tự nhiên và dễ hiểu. Tuy nhiên, khác với các đầu ra phân loại và định vị hộp, mặt nạ đòi hỏi mô hình phải nắm bắt được thông tin không gian chi tiết hơn nhằm thể hiện chính xác hình dạng của đối tượng.

- Faster R-CNN: Faster R-CNN hoạt động theo hai giai đoạn chính. Ở giai đoạn đầu, gọi là mạng đề xuất vùng (Region Proposal Network – RPN) được sử dụng để tạo ra các hộp giới hạn ứng viên chứa đối tượng. Giai đoạn tiếp theo, thực hiện trích xuất đặc trưng từ từng hộp đề xuất bằng kỹ thuật RoIPooling, sau đó tiến hành phân loại và điều chỉnh lại hộp giới hạn. Đáng chú ý, cả hai giai đoạn đều có thể dùng chung các đặc trưng được trích xuất từ mạng nơ-ron để tăng tốc độ suy luận [45].



Hình 2.15: Kiến trúc của mạng Faster R-CNN [46]

- Mask R-CNN: có kiến trúc tương tự như Faster R-CNN nhưng bên cạnh đó đã được bổ sung thêm một nhánh mask để thực hiện dự báo nhãn trên từng pixel.



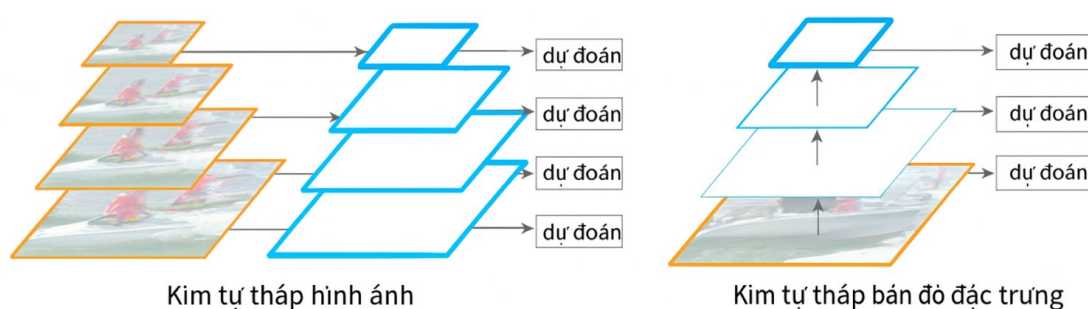
Hình 2.16: Kiến trúc mạng Mask R-CNN [46]

Điều khác biệt ở đây là đối với Faster R-CNN không được thiết kế để có thể căn chỉnh từ pixel tới pixel giữa đầu vào và đầu ra của mạng điều này được thể hiện rõ ở cách RoIPool [45]. RoIPool trong Faster R-CNN thực hiện lượng tử hóa hai lần khi ánh xạ RoI từ ảnh gốc sang feature map, dẫn đến sai lệch vị trí đáng kể giữa các pixel trong RoI và các đặc trưng được trích xuất. Điều này đặc biệt gây hại cho việc phân đoạn chính xác ở cấp độ pixel. Để khắc phục được sự sai lệch này thì một lớp đơn giản, không lượng tử hóa gọi là RoIAlign được đưa ra để bảo đảm sự chính xác của các vị trí trong không gian và giải quyết được thách thức trong việc phân đoạn chính xác từng thể hiện riêng biệt của các đối tượng thuộc cùng một lớp trong một cảnh phức tạp và đây cũng là đóng góp then chốt trong Mark R-CNN để có thể thay thế RoIPooling trong Faster R-CNN.

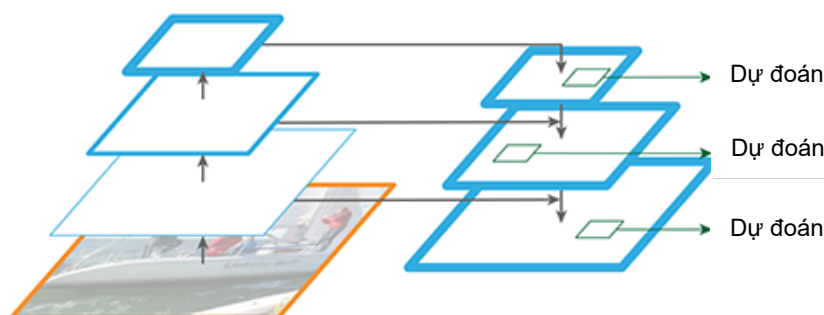
2.2.4.4. Phân đoạn ảnh sử dụng mô hình kiến trúc FPN

Việc phát hiện đối tượng có nhiều kích thước khác nhau, đặc biệt là các vật thể nhỏ, vẫn là một thách thức lớn trong thị giác máy tính. Một giải pháp truyền thống là sử dụng tháp ảnh (image pyramid), trong đó ảnh đầu vào được thay đổi kích thước theo nhiều tỉ lệ để đảm bảo khả năng phát hiện trên nhiều quy mô. Tuy nhiên, phương pháp này thường yêu cầu chi phí tính toán rất cao, dẫn đến tốc độ xử lý chậm, và chỉ phù hợp trong các tình huống mà độ chính xác được ưu tiên hơn hiệu năng, chẳng hạn như trong các cuộc thi.

Thay vì xử lý ảnh ở nhiều độ phân giải khác nhau, một hướng tiếp cận hiệu quả hơn là xây dựng tháp đặc trưng (feature pyramid). Trong cấu trúc này, các tầng đặc trưng ở gần đầu vào thường biểu diễn các đặc trưng mức thấp (low-level features) như cạnh hoặc kết cấu, nhưng lại thiếu thông tin ngữ nghĩa cần thiết cho việc phát hiện đối tượng. Do đó, việc kết hợp đặc trưng từ nhiều tầng với các mức trừu tượng khác nhau là cần thiết để cải thiện hiệu suất phát hiện trên toàn bộ dải kích thước vật thể.



a) Sử dụng Pyramid of image và Pyramid of feature maps để phát hiện vật thể



b) Mô hình mô tả phát hiện vật thể với FPN

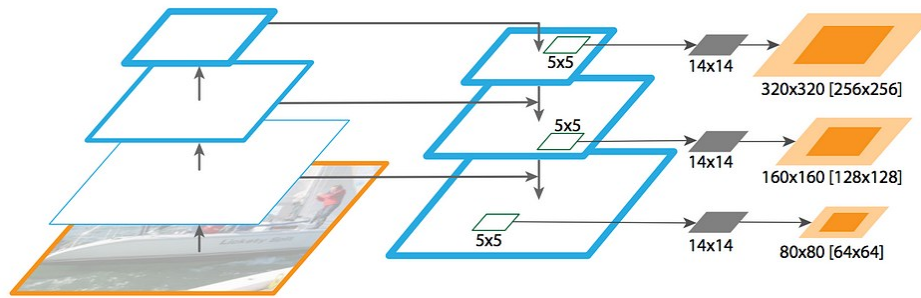
Hình 2.17: So sánh 2 phương pháp phát hiện vật thể thông thường với FPN [47]

Feature Pyramid Network (FPN) là một bộ trích xuất đặc trưng được thiết kế dựa trên khái niệm tháp đặc trưng (feature pyramid), với mục tiêu cải thiện cả độ chính xác và hiệu suất trong các bài toán phát hiện đối tượng. Khác với phương pháp tạo tháp ảnh truyền thống, FPN xây dựng một cấu trúc phân cấp các bản đồ đặc trưng (feature maps) ở nhiều cấp độ độ phân giải ngay trong quá trình trích xuất đặc trưng.

FPN có thể tích hợp trực tiếp vào các kiến trúc phát hiện như Faster R-CNN bằng cách thay thế bộ trích xuất đặc trưng mặc định. Nhờ việc tạo ra các tầng đặc trưng đa tỉ lệ (multi-scale feature maps) với thông tin phong phú hơn, FPN giúp cải thiện đáng kể khả năng phát hiện các đối tượng có kích thước khác nhau, đặc biệt là các vật thể nhỏ mà các phương pháp thông thường thường bỏ sót.

Trong lĩnh vực phân đoạn ảnh, đặc biệt là khi đối mặt với các đối tượng có kích thước và hình dạng đa dạng trong cùng một hình ảnh, việc sử dụng các đặc trưng ở nhiều mức độ phân giải là vô cùng cần thiết. Feature Pyramid Network (FPN) là một kiến trúc mạng học sâu hiệu quả cho mục tiêu này. Mô hình FPN được thiết kế để khai thác các đặc trưng ở nhiều cấp độ của mạng CNN nhằm cải thiện khả năng nhận diện và phân đoạn đối tượng ở các thang đo khác nhau, đặc biệt là các đối tượng nhỏ. Tương tự như Mask R-CNN, Feature Pyramid Network

(FPN) cũng cho thấy hiệu quả cao trong việc trích xuất mặt nạ đối tượng (masks) cho các tác vụ phân đoạn ảnh. Phương pháp này sử dụng một mạng perceptron đa tầng (MLP), trong đó một cửa sổ kích thước 5×5 được trượt trên các bản đồ đặc trưng (feature maps) để tạo ra các phân đoạn đối tượng có kích thước 14×14 . Các mặt nạ được tạo ra ở các tỷ lệ khác nhau sau đó được kết hợp lại nhằm tạo thành mặt nạ cuối cùng, đảm bảo độ chính xác và khả năng bao phủ tốt hơn đối tượng trong ảnh [47].



Hình 2.18: FPN cho phân đoạn đối tượng [48]

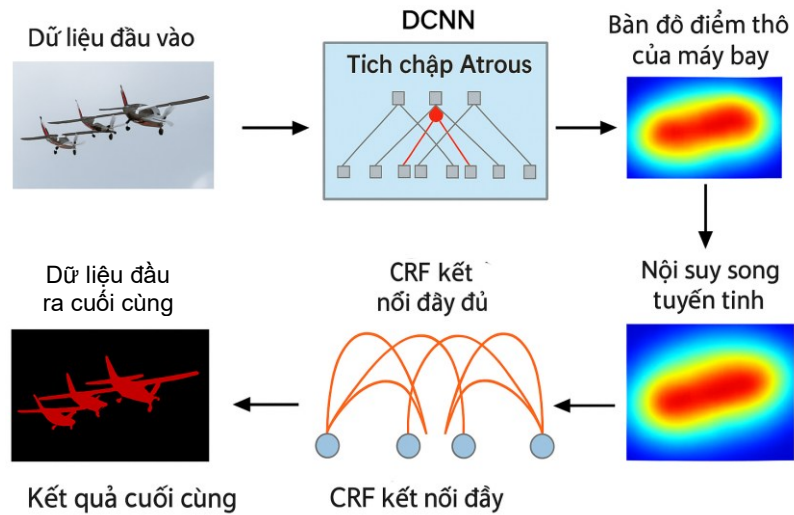
FPN bao gồm hai nhánh chính: nhánh từ dưới lên (bottom-up) trích xuất đặc trưng từ ảnh thông qua các tầng CNN như ResNet, và nhánh từ trên xuống (top-down) kết hợp với các kết nối ngang để tái tạo các đặc trưng có độ phân giải cao với thông tin ngữ nghĩa mạnh. Nhờ đó, mạng có thể tận dụng cả thông tin không gian từ tầng nông và thông tin ngữ nghĩa từ tầng sâu.

Trong các mô hình phân đoạn ảnh như Mask R-CNN, FPN được tích hợp làm phần backbone để cải thiện khả năng xác định biên đối tượng và chi tiết ở các cấp độ khác nhau. Việc sử dụng FPN giúp nâng cao hiệu suất phân đoạn trong các bài toán yêu cầu độ chính xác cao như phân đoạn y tế, bản đồ địa hình, hoặc robot di động định vị trong môi trường phức tạp.

2.2.4.5. Phân đoạn sử dụng mô hình kiến trúc DeepLab

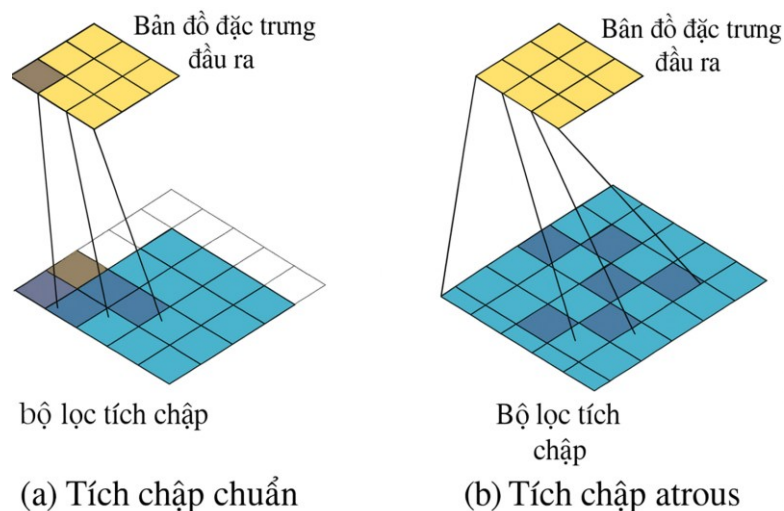
DeepLab là một mô hình mạng học sâu được phát triển bởi nhóm Google Research và là một trong những kiến trúc tiên phong và hiệu quả nhất giải quyết thách thức này thông qua việc kết hợp khéo léo và linh hoạt Atrous Convolution thay vì các phương pháp trước đó là áp dụng Transposed Convolution để mở rộng Receptive Field và sử dụng phương pháp Conditional Random Field để tinh chỉnh kết quả dự báo chuẩn xác hơn. Mô hình DeepLab V1 và DeepLab V2 lần đầu tiên được công bố vào đầu năm 2016 và lần thứ hai vào năm 2017 bởi nhóm tác giả Liang-Chieh Chen và George Panandreu [49]. Qua nhiều phiên bản (DeepLab

v1, v2, v3, v3+), mô hình liên tục được cải tiến để nâng cao độ chính xác và hiệu quả tính toán.



Hình 2.19: Mô hình chung của Deeplab [50]

Tập chập không triệt tiêu hay còn được gọi là Atrous Convolution hoặc Dilated Convolution, là một kỹ thuật trong mạng nơ-ron tích chập, trong đó các phần tử trong bộ lọc được chen thêm các khoảng trống (theo hệ số dilation rate r). Cách làm này cho phép mở rộng trường tiếp nhận (Receptive Field) mà không làm tăng số lượng tham số hoặc giảm độ phân giải không gian của feature map. Atrous Convolution cho phép mô hình duy trì độ phân giải cao của feature map ở các tầng sâu mà không cần sử dụng các phép nội suy (upsampling) hoặc giảm kích thước (downsampling). Điều này giúp giảm thiểu mất mát thông tin và tăng cường khả năng nắm bắt ngữ cảnh trong ảnh đầu vào. Mạng DeepLab là một ví dụ tiêu biểu ứng dụng kỹ thuật này trong phân đoạn ảnh ngữ nghĩa với độ chính xác cao [51].



Hình 2.20: Sự khác biệt giữa môi trường tiếp nhận trong đầu vào của Standard Convolution và Atrous Convolution [52]

Khi ta cho tín hiệu đầu vào $x(i)$, bộ lọc $\omega(k)$ có kích thước K và hệ số dẫn r , đầu ra của tích chập không triệt tiêu tại vị trí i được tính theo công thức:

$$y(i) = \sum_{k=1}^K x(i + r \times k) \times \omega(k) \quad (2.2)$$

Trong không gian 2 chiều, công thức tổng quát hơn sẽ là:

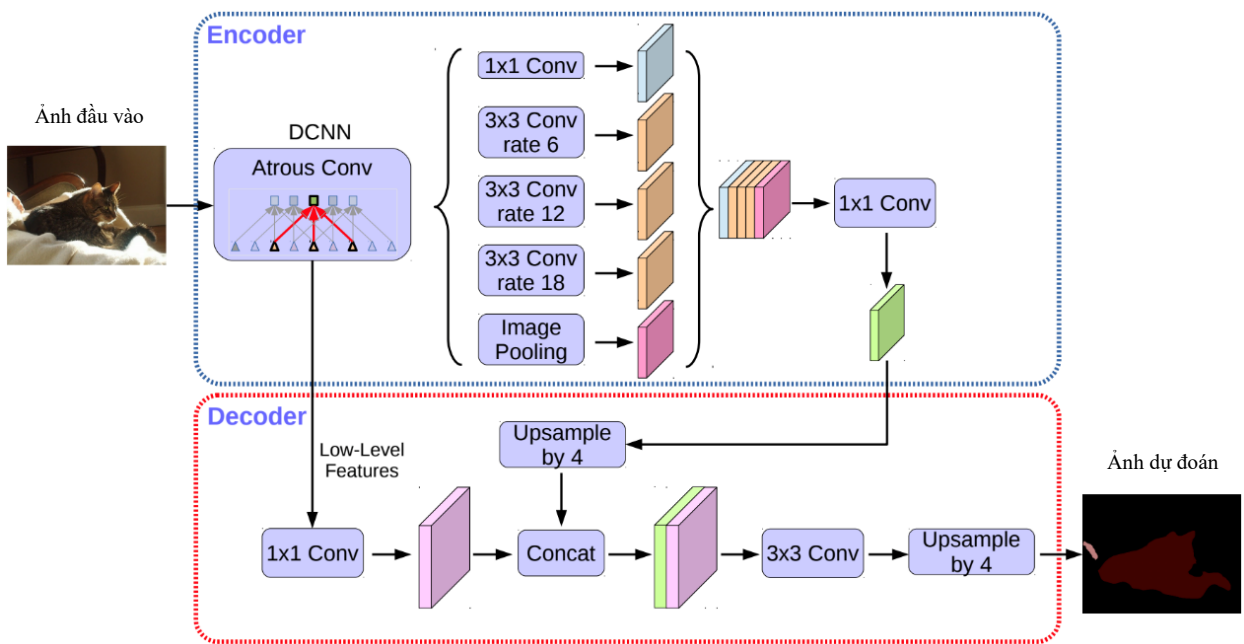
$$y(i, j) = \sum_{m=1}^M \sum_{n=1}^N x(i + r \times m, j + r \times n) \times \omega(m, n) \quad (2.3)$$

Trong đó: r là hệ số giãn (dilation rate), ω là bộ lọc, x là ảnh đầu vào (input feature map), y là kết quả đầu ra (output feature map).

Đối với DeepLab v1 ta chỉ sử dụng 1 phép Dilated duy nhất và sử dụng fully connected CRFs để tinh chỉnh biên đối tượng sau phân đoạn thì ý tưởng của DeepLab V2 ta sử dụng thêm Mô đun ASPP (Atrous Spatial Pyramid Pooling). Mô đun ASPP được thiết kế nhằm nắm bắt thông tin ngữ cảnh ở nhiều tỷ lệ khác nhau (multi-scale context), từ đó cải thiện khả năng mô hình hóa các đối tượng có kích thước và hình dạng đa dạng trong ảnh đầu vào [51]. ASPP khai thác hiệu quả kỹ thuật tích chập không triệt tiêu (Atrous Convolution) bằng cách áp dụng song song nhiều nhánh tích chập với các hệ số dẫn (dilation rates) khác nhau, thường là $r = 6, 12, 18$ trên cùng một feature map. Bên cạnh đó, một nhánh pooling toàn cục (Global Average Pooling) cũng được thêm vào để thu thập thông tin ngữ cảnh toàn cục. Đầu ra của tất cả các nhánh được kết hợp lại (thường bằng phép nối – concatenation), sau đó được xử lý bởi một lớp tích chập 1×1 nhằm giảm số chiều và hợp nhất thông tin. Nhờ khả năng trích xuất đặc trưng ở nhiều tỷ lệ không gian khác nhau mà không làm mất thông tin quan trọng, ASPP giúp mô hình nhận diện hiệu quả các đối tượng có kích thước và vị trí khác nhau trong ảnh. Kỹ thuật này đặc biệt hữu ích trong các bài toán phân đoạn ảnh ngữ nghĩa, điển hình là trong các phiên bản nâng cao của mạng DeepLab [51].

Ở DeepLab v3 đã được loại bỏ Fully Connected CRF thay vào đó cải tiến ASPP với các dilation rate lớn hơn. DeepLab v3+ là một sự kết hợp giữa DeepLabv3 (mạnh về trích xuất ngữ cảnh đa tỷ lệ) và kiến trúc encoder - decoder (mạnh về khôi phục chi tiết không gian) [53]. Encoder (bộ mã hóa) được xây dựng dựa trên backbone mạnh như Xception hoặc ResNet, trích xuất đặc trưng ở mức $\text{stride} = 16$. Đầu ra này sau đó được đưa vào mô-đun Atrous Spatial Pyramid Pooling (ASPP) để thu thập ngữ cảnh đa tỷ lệ một cách hiệu quả mà không làm

mất độ phân giải không gian của feature map. Decoder (bộ giải mã) trong DeepLabv3+ được thiết kế nhằm khôi phục các chi tiết không gian, đặc biệt là viền và biên đối tượng.



Hình 2.21: Mô hình mạng DeepLab v3+ [54]

Cấu trúc encoder-decoder trong DeepLabv3+ giúp mô hình đồng thời khai thác thông tin ngữ cảnh cấp cao (high-level semantic context) và giữ lại chi tiết biên không gian (low-level spatial detail). Nhờ đó, mô hình cải thiện đáng kể độ chính xác tại các ranh giới đối tượng và hiệu suất tổng thể trong các bài toán phân đoạn ngữ nghĩa [51, 53].

DeepLabv3+ đã được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau nhờ khả năng phân đoạn ảnh chính xác và hiệu quả:

- Trong lĩnh vực xe tự hành, mô hình được sử dụng để phân đoạn các thành phần quan trọng như làn đường, biển báo giao thông, người đi bộ và bề mặt đường, góp phần nâng cao khả năng nhận thức môi trường và ra quyết định của hệ thống điều khiển tự động.

- Trong y học, DeepLabv3+ hỗ trợ phân đoạn các cấu trúc mô bệnh học cũng như xác định ranh giới của các khối u trong ảnh chụp CT hoặc MRI, từ đó hỗ trợ chẩn đoán và lập kế hoạch điều trị chính xác hơn.

- Trong nông nghiệp thông minh, mô hình giúp phân vùng khu vực cây trồng từ ảnh vệ tinh hoặc ảnh chụp từ thiết bị bay không người lái (flycam), hỗ trợ quản lý mùa vụ, đánh giá sinh trưởng và giám sát sâu bệnh.

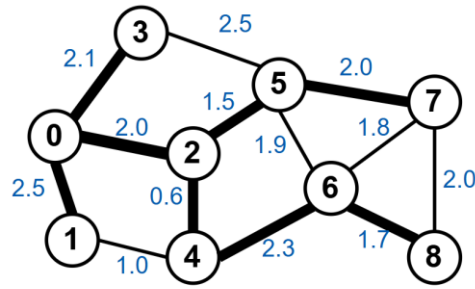
- Trong lĩnh vực robot di động, DeepLabv3+ đóng vai trò quan trọng trong việc xây dựng bản đồ số, nhận diện các khu vực và vật thể trong môi trường xung quanh, phục vụ cho các tác vụ định vị và lập kế hoạch dẫn đường tự động.

2.3. Phương pháp thiết kế quỹ đạo

Trong thiết kế quỹ đạo cho robot tự hành, các thuật toán dựa trên tìm kiếm đồ thị (Graph search-based algorithms) là một phương pháp nền tảng, đặc biệt trong việc hoạch định quỹ đạo toàn cục (Global path planning). Các thuật toán này hoạt động trên một cấu trúc đồ thị bao gồm các nút (nodes) và các cạnh (edges) kết nối các nút này. Bằng cách sử dụng các chiến lược tìm kiếm cụ thể, chúng xác định một đường đi tối ưu hoặc khả thi từ nút bắt đầu đến điểm đích. Hai thuật toán tiêu biểu và phổ biến nhất trong nhóm này là Dijkstra và A* [55].

2.3.1. Thiết kế quỹ đạo với Dijkstra

Thuật toán của Dijkstra là một thuật toán phổ biến để giải các bài toán đường dẫn ngắn nhất nguồn duy nhất có trọng số cạnh không âm trong đồ thị, tức là tìm khoảng cách ngắn nhất giữa hai đỉnh trên đồ thị. Trong quá trình thực thi, thuật toán Dijkstra duy trì một bảng khoảng cách ghi lại độ dài đường đi ngắn nhất hiện tại từ nút xuất phát đến mỗi nút khác. Bắt đầu từ nút gốc, thuật toán sẽ dần dần lựa chọn nút có khoảng cách nhỏ nhất để mở rộng, cập nhật khoảng cách của các nút lân cận và thêm chúng vào tập hợp các nút đã có đường đi ngắn nhất. Quá trình này được lặp lại cho đến khi tất cả các nút đều được duyệt qua, qua đó tìm ra đường đi ngắn nhất từ điểm bắt đầu đến tất cả các nút còn lại. Thuật toán Dijkstra có thể giải quyết bài toán tìm đường đi ngắn nhất trên đồ thị vô hướng lẫn có hướng miễn là trọng số không âm. Đối với các đồ thị có hướng, ở mỗi cạnh có hướng sẽ cho biết hướng di chuyển giữa các đỉnh với nhau bằng các cạnh và thuật toán sẽ tuân thủ theo các hướng của các cạnh khi tìm kiếm đường đi ngắn nhất. Còn đối với các đồ thị vô hướng, khi đó thuật toán sẽ di chuyển cá tiến và lùi dọc theo các cạnh để tìm kiếm được đường đi ngắn nhất.



0	1	2	3	4	5	6	7	8
0	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)
-	(2.5, 0)	(2.0, 0)	(2.1, 0)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)
-	(2.5, 0)	-	(2.1, 0)	(2.6, 2)	(3.5, 2)	(∞ , -)	(∞ , -)	(∞ , -)
-	(2.5, 0)	-	-	(2.6, 2)	(3.5, 2)	(∞ , -)	(∞ , -)	(∞ , -)
-	-	-	-	(2.6, 2)	(3.5, 2)	(∞ , -)	(∞ , -)	(∞ , -)
-	-	-	-	-	(3.5, 2)	(4.9, 4)	(∞ , -)	(∞ , -)
-	-	-	-	-	-	(4.9, 4)	(5.5, 5)	(∞ , -)
-	-	-	-	-	-	-	(5.5, 5)	(6.6, 6)
-	-	-	-	-	-	-	-	(6.6, 6)

Hình 2.22: Tiến trình của thuật toán Dijkstra [55]

Thuật toán này không có một công thức toán học nhất định mà nó dựa trên một nguyên tắc cập nhật trọng số, ý tưởng của thuật toán có thể được diễn giải như sau:

- Xét các thành phần chính bao gồm:

+ $d(v)$ là một mảng lưu trữ độ dài đường đi ngắn nhất (tạm thời) từ điểm nguồn s đến đỉnh v . Ban đầu $d(s) = 0$ và $d(v) = \infty$ cho tất cả các đỉnh v khác s .

+ $prev(v)$ là một mảng lưu trữ đỉnh nằm ngay trước đỉnh v trên đường đi ngắn nhất từ nguồn s . Mảng này dùng để truy vết lại đường đi sau khi kết thúc thuật toán.

+ Tập các đỉnh đã xét (S) dùng để lưu những đỉnh đã tìm được đường đi ngắn nhất.

+ Tập các đỉnh chưa xét (Q) dùng để lưu những điểm còn lại.

- Công thức và khái niệm cốt lõi của thuật toán: Với mỗi cạnh từ đỉnh u đến đỉnh v có trọng số $w(u, v)$ thì khi đó ta thực hiện so sánh: Nếu $d(u) + w(u, v) < d(v)$ thì cập nhật $d(v) = d(u) + w(u, v)$, $prev(v) = u$. Nghĩa là khi nếu đi từ nguồn s đến u rồi đến v mà ngắn hơn đường đi thẳng đến v đã biết trước đó thì ta cập nhật lại đường đi ngắn nhất đến v chính là con đường mới này.

- Các bước thực hiện thuật toán:

Khởi tạo:

- + Tạo mảng khoảng cách d và gán $d(s) = 0$, tất cả các phần tử còn lại là ∞
- + Tạo mảng truy vết $prev$ và gán tất cả các phần tử là rỗng.
- + Tạo một tập Q chứa tất cả các đỉnh của đồ thị.

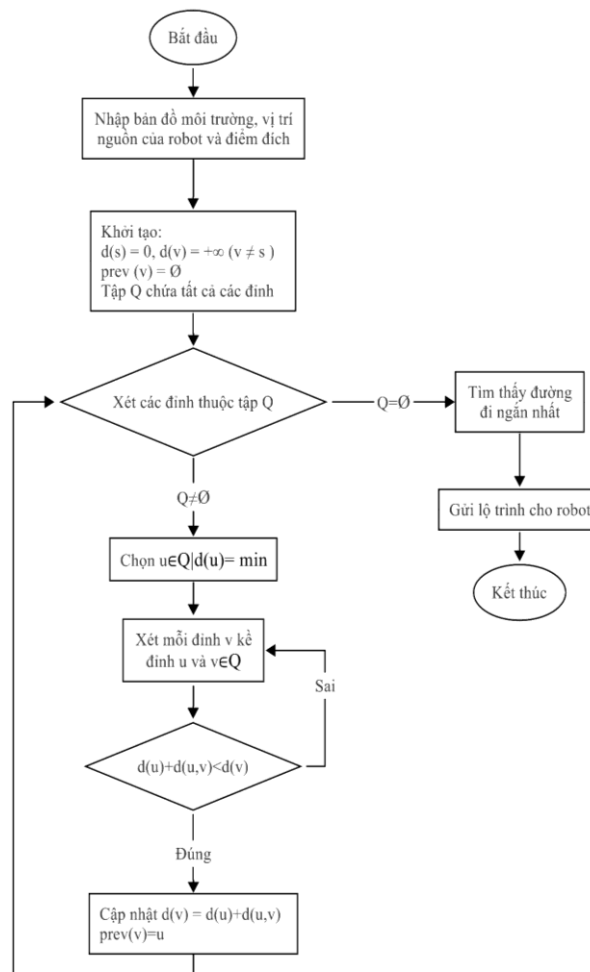
Lặp: Chừng nào tập Q vẫn còn phần tử

- + Chọn đỉnh u trong Q có giá trị $d(u)$ là nhỏ nhất.
- + Loại u ra khỏi Q .
- + Với mỗi đỉnh v kề với u mà vẫn còn trong Q ta thực hiện xét:

Nếu $d(u) + w(u, v) < d(v)$ thì cập nhật $d(v) = d(u) + w(u, v)$ và $prev(v) = u$.

Kết thúc: Khi tập Q rỗng, thuật toán kết thúc. Mảng d sẽ chứa độ dài đường đi ngắn nhất từ s đến mọi đỉnh khác, và mảng $prev$ có thể được dùng để tái tạo lại các đường đi đó.

Theo các bước thực hiện thuật toán được mô tả ở trên, ta có lưu đồ thuật toán Dijkstra cụ thể như sau:



Hình 2.23: Lưu đồ thuật toán Dijkstra

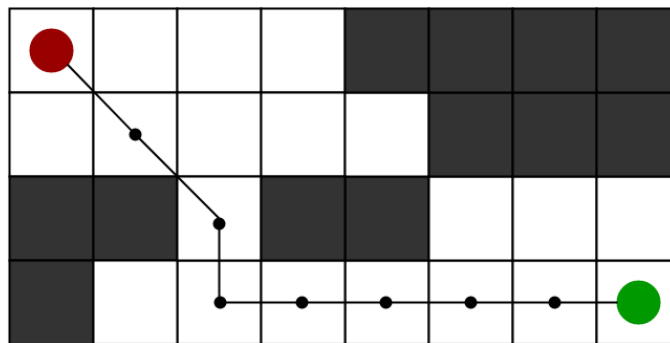
Mặc dù hiệu quả, thuật toán Dijkstra truyền thống tồn tại một số nhược điểm như không thể xử lý các cạnh có trọng số âm và phản ứng chậm trong các môi trường thay đổi động. Để khắc phục những hạn chế này, nhiều cải tiến đã được đề xuất [55]:

- Định hướng tìm kiếm: Một phương pháp cải tiến là xem xét cả khoảng cách từ nút đang xét đến điểm đích để định hướng cho việc tìm kiếm, đồng thời sử dụng mô hình elip để giới hạn phạm vi tìm kiếm, giúp giảm thiểu các phép tính không cần thiết.

- Mở rộng hướng tìm kiếm: Thay vì tìm kiếm theo bốn hướng lân cận (Four-neighborhood search) trong bản đồ lưới, một số nghiên cứu đề xuất phương pháp tìm kiếm bát giác (Octagonal search), cho phép tìm kiếm theo tám hướng. Cách tiếp cận này giúp hoạch định được một đường đi ngắn hơn, mượt mà hơn, đồng thời giảm số lần và góc rẽ.

2.3.2. Thiết kế quỹ đạo với A^*

Thuật toán A^* là một phiên bản mở rộng của thuật toán Dijkstra. Điểm cải tiến cốt lõi của A^* là nó giới thiệu thêm một hàm Heuristic, ký hiệu là $h(n)$ để ước tính được chi phí từ nút hiện tại đến nút đích. Việc kết hợp thêm hàm chi phí ước tính này giúp đẩy nhanh quá trình tìm kiếm. Trong khi Dijkstra có xu hướng khám phá tất cả các nút có thể tiếp cận thì A^* lại tìm kiếm một cách có chọn lọc hơn bằng cách ưu tiên các hướng tiến về phía đích.



Hình 2.24: Tìm kiếm đường đi của giải thuật A^* [56]

Cơ sở lý thuyết của thuật toán A^* là kết hợp giữa hai hàm đánh giá đó là hàm $g(n)$ và hàm $h(n)$, trong đó $g(n)$ là chi phí thực tế từ điểm nút xuất phát đến điểm nút hiện tại (nút n) và $h(n)$ là chi phí Heuristic ước lượng từ nút hiện tại (nút n) đến điểm đích trong không gian trạng thái. Hàm tổng chi phí $f(n) = g(n) + h(n)$ được tối ưu hóa để lựa chọn đường đi ngắn nhất và đảm bảo hiệu quả trong không gian tìm kiếm phức tạp, $f(x)$ thường có giá càng thấp thì độ ưu tiên càng cao. Heuristic thường sử dụng khoảng cách Euclid hoặc Manhattan, tùy thuộc vào đặc

thù bài toán. Khi giá trị của $g(n)$ không đổi, giá trị của $f(n)$ chủ yếu bị ảnh hưởng bởi giá trị của $h(n)$. Khi nút ở gần nút đích, giá trị của $h(n)$ nhỏ, giá trị của $f(n)$ tương đối nhỏ. Do đó, nó đảm bảo việc tìm kiếm đường đi ngắn nhất luôn diễn ra theo hướng của điểm đích. Thuật toán A* xem xét thông tin vị trí của điểm mục tiêu của rô-bốt di động và tìm kiếm dọc theo điểm đích. So với thuật toán Dijkstra, hiệu quả tìm kiếm đường dẫn của thuật toán A* cao hơn.

- Khoảng cách Euclid (Euclidean Distance) là độ dài đường thẳng ngắn nhất giữa hai điểm trong không gian liên tục, phản ánh được đúng khoảng cách vật lý trong môi trường liên tục và phù hợp với các hệ thống di chuyển tự do. Bên cạnh đó Heuristic theo khoảng cách Euclid giúp thuật toán ưu tiên các hướng tiến gần đến thẳng đích và giảm số lượng nút cần duyệt, được tính theo công thức:

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.4)$$

Trong đó $(x_2 - x_1)$ và $(y_2 - y_1)$ là tọa độ của nút hiện tại và nút đích.

- Khoảng cách Manhattan (Manhattan Distance) là tổng độ lệch theo các trục tọa độ, tính toán đơn giản hơn khoảng cách Euclid do không sử dụng phép khai căn và thường phù hợp với các hệ thống yêu cầu tốc độ cao. Đặc biệt hiệu quả trong môi trường lưới khi robot di chuyển theo 4 hoặc 8 hướng cố định và đảm bảo hàm Heuristic có thể phản ánh được đúng chi phí thực tế khi di chuyển theo các cạnh lưới. Được tính theo công thức:

$$h(n) = |x_2 - x_1| + |y_2 - y_1| \quad (2.5)$$

Thuật toán chi tiết:

Khởi tạo:

- + *Open*: Danh sách chứa các nút đang xét, ban đầu chỉ chứa nút bắt đầu.
- + *Closed*: Danh sách các nút đã xét, ban đầu rỗng.
- + $G(start) = 0$.
- + $H(start) = heuristic(start)$
- + $F(start) = G(start) + H(start)$

Lặp cho đến khi Open rỗng hoặc tìm thấy đích:

- + Chọn nút n trong *Open* và $F(n)$ nhỏ nhất
- + Nếu n là đích, thoát và trả về đường đi
- + Xóa n ra khỏi *Open* và di chuyển sang *Closed*
- + Với mỗi nút kế tiếp m của n :
 - Nếu m trong *Closed*, bỏ qua

- Tính $G(\text{tentative}) = G(n) + \text{cost}(n, m)$
- Nếu m không trong Open , thêm m vào Open
- Nếu m trong Open và $G(\text{tentative}) \geq G(m)$ thì bỏ qua
- Cập nhật:

$$\text{Parent}(m) = n$$

$$G(m) = G(\text{tentative})$$

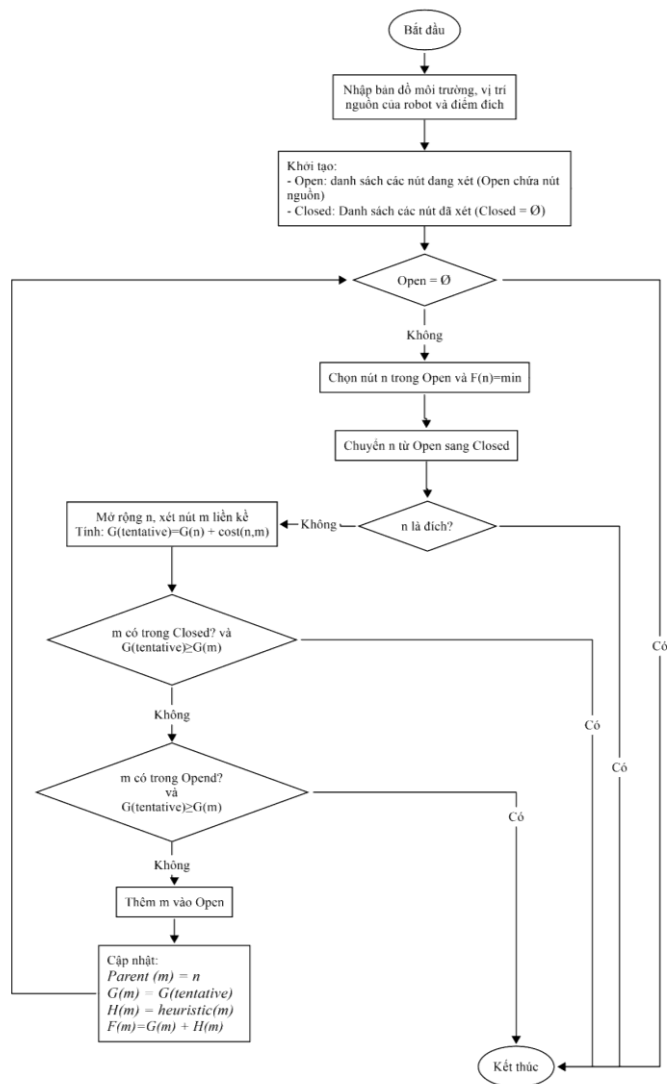
$$H(m) = \text{heuristic}(m)$$

$$F(m) = G(m) + H(m)$$

Trong đó:

- $G(\text{tentative})$: Chi phí từ điểm bắt đầu đến điểm tạm thời
- $\text{Parent}(m)$: Nút cha, dùng để truy vết đường đi ngược

Theo các bước thực hiện thuật toán được mô tả ở trên, ta có lưu đồ thuật toán A^* cụ thể như sau:



Hình 2.25: Lưu đồ thuật toán A^*

Kết luận chương II

Chương này cung cấp kiến thức nền tảng về học máy và học sâu, giới thiệu các kiến trúc mạng nơ-ron phổ biến như CNN, FCN, U-Net, DeepLabV3+ và các thuật toán thiết kế quỹ đạo như Dijkstra và A*. Các phương pháp phân đoạn ảnh bằng học sâu được trình bày như một công cụ quan trọng trong việc xây dựng bản đồ số cho robot di động. Đây là cơ sở lý thuyết quan trọng cho việc lựa chọn và triển khai mô hình ở chương sau.

CHƯƠNG 3 - XÂY DỰNG MÔ HÌNH HỌC SÂU VÀ THIẾT KẾ QUỸ ĐẠO CHUYỂN ĐỘNG CHO ROBOT

3.1. Quy trình thực hiện

Mục tiêu chính của đề án là ứng dụng các thuật toán học sâu để tự động trích xuất bản đồ đường đi từ ảnh vệ tinh. Bản đồ đường đi được trích xuất sau đó sẽ là cơ sở dữ liệu đầu vào cho các thuật toán hoạch định đường đi (path planning), nhằm mục đích tìm kiếm lộ trình tối ưu cho robot trong môi trường đã được định nghĩa. Phương pháp tạo bản đồ truyền thống, thường sử dụng cảm biến LIDAR, mặc dù hiệu quả, nhưng lại đối mặt với những hạn chế đáng kể về thời gian và chi phí. Đặc biệt, khi thực hiện các thử nghiệm trong nhiều môi trường đa dạng, việc triển khai LIDAR trở nên kém hiệu quả do yêu cầu cao về tài nguyên và công sức. Để khắc phục những hạn chế trên, đề án đề xuất một hướng tiếp cận mới: ứng dụng học sâu để tự động hóa hoàn toàn quá trình tạo bản đồ. Điều này không chỉ giúp tăng cường hiệu quả mà còn đẩy nhanh tốc độ nghiên cứu và phát triển, tạo tiền đề cho việc triển khai robot tự hành một cách linh hoạt và tiết kiệm hơn trong các khu vực.

Quy trình thực hiện nội dung nghiên cứu trong đề án có thể chia làm 3 giai đoạn cơ bản như sau:

- Giai đoạn 1: Tạo bản đồ số

- + Thu thập dữ liệu đa nguồn: Sử dụng ảnh vệ tinh độ phân giải cao từ Google Earth Pro và kết hợp dữ liệu địa hình từ OpenStreetMap, bổ sung dataset công cộng như SpaceNet cho khu vực đô thị.

- + Tiền xử lý dữ liệu: Chuẩn hóa kích thước hình ảnh và xử lý cải thiện chất lượng hình ảnh

- + Trích xuất bản đồ từ dữ liệu thu thập được sử dụng phương pháp học sâu

- Giai đoạn 2: Tối ưu hóa đường đi

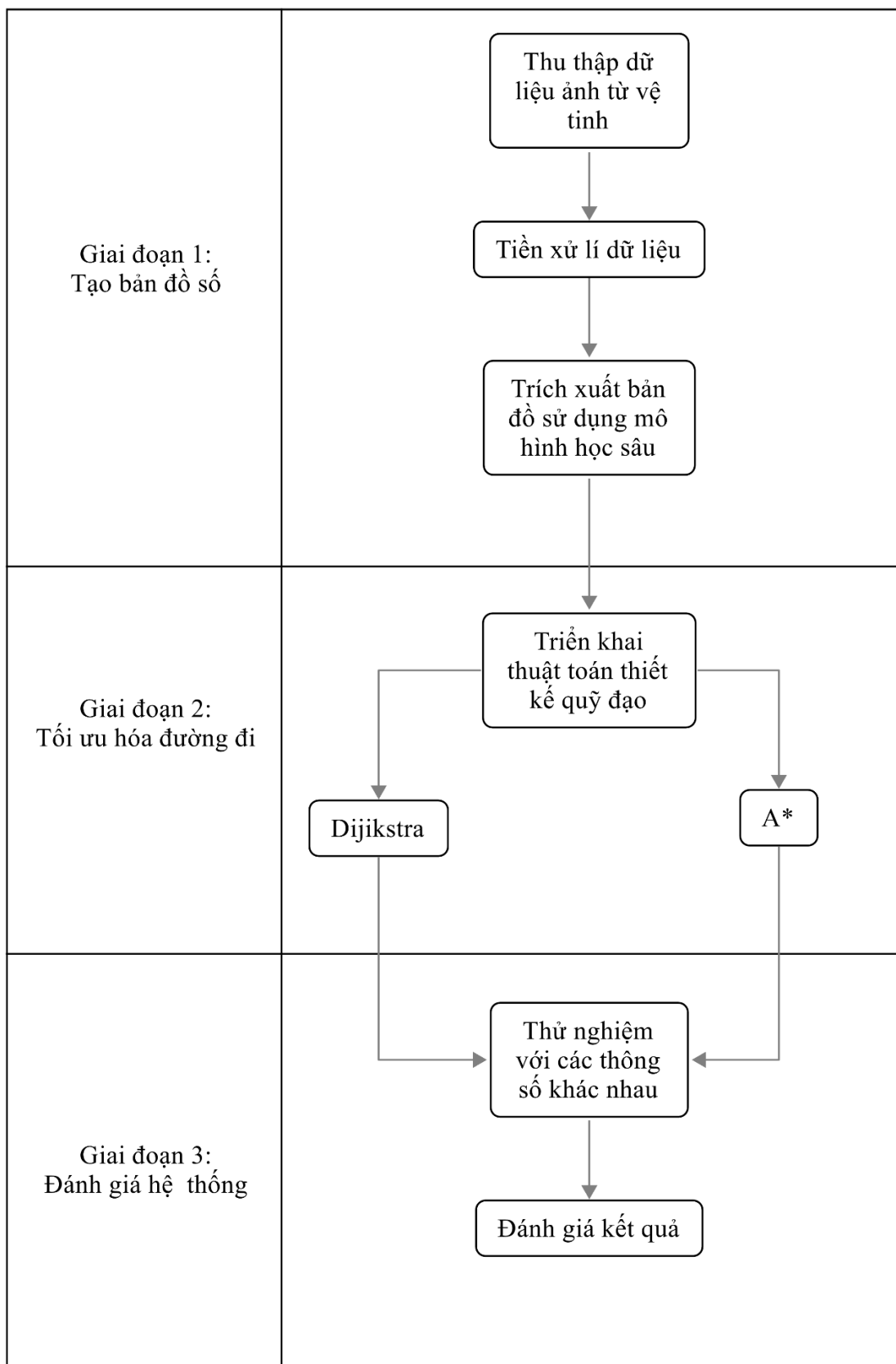
Phát triển các thuật toán như Dijkstra, A* để tìm đường đi ngắn nhất từ điểm xuất phát đến điểm đích mong muốn trên bản đồ.

- Giai đoạn 3: Đánh giá hệ thống

- + So sánh với phương pháp truyền thống

- + Đánh giá thời gian thực thi giữa các thuật toán với các thông số khác nhau

Quy trình trên được sơ đồ hóa thành các bước như sau:



Hình 3.1: Quy trình thực hiện các nội dung nghiên cứu

3.2. Lựa chọn mô hình học sâu

Trong các hệ thống định vị và dẫn đường cho robot di động, thiết kế bản đồ số đóng vai trò then chốt để mô phỏng, nhận diện và trích xuất thông tin từ môi trường. Một trong những bước quan trọng trong quá trình thiết kế bản đồ là phân đoạn ảnh (image segmentation) – nhằm phân chia, huấn luyện hình ảnh thành các vùng tương ứng để phát hiện đường đi cho robot di động. Việc lựa chọn mô hình học sâu hiệu quả trong bài toán này ảnh hưởng lớn đến độ chính xác và độ tin cậy của bản đồ số được xây dựng. Trong phạm vi nghiên cứu này, ba mô hình học sâu được lựa chọn gồm: U-Net, DeepLabV3+ và Feature Pyramid Network (FPN).

DeepLab v3+

DeepLab v3+ là phiên bản cải tiến của các mô hình DeepLab trước đó (DeepLab v1, v2, v3), được giới thiệu trong bài báo “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. Mô hình này kết hợp các kỹ thuật như tích chập Atrous (Atrous Convolution), mô-đun không gian kim tự tháp Atrous (Atrous Spatial Pyramid Pooling - ASPP), và kiến trúc mã hóa-giải mã (Encoder-Decoder) để cải thiện độ chính xác và hiệu quả trong việc phân đoạn hình ảnh. Chính vì thế, DeepLab v3+ được sử dụng để phân loại từng pixel trong hình ảnh thành các danh mục như đường, tòa nhà, cây cối, hoặc vùng nước, từ đó tạo ra các bản đồ phân vùng chính xác.

Trích xuất bản đồ từ hình ảnh vệ tinh hoặc dữ liệu không gian đòi hỏi khả năng phân đoạn chính xác các thực thể địa lý ở nhiều tỷ lệ và ngữ cảnh khác nhau. DeepLab v3+ đặc biệt phù hợp với bài toán này nhờ vào các đặc điểm sau:

- Xử lý hình ảnh độ phân giải cao: Hình ảnh vệ tinh thường có kích thước lớn và chứa chi tiết phong phú. DeepLab v3+ sử dụng tích chập atrous để duy trì độ phân giải không gian, giúp bảo toàn thông tin chi tiết về ranh giới của các đối tượng như đường xá hoặc tòa nhà.

- Biểu diễn ngữ cảnh đa tỷ lệ: Mô-đun ASPP trong DeepLab v3+ cho phép thu thập thông tin ngữ cảnh ở các trường thụ cảm khác nhau, phù hợp để phân đoạn các đối tượng có kích thước và hình dạng đa dạng.

- Khôi phục chi tiết ranh giới: Kiến trúc mã hóa-giải mã kết hợp đặc trưng cấp cao và cấp thấp, giúp cải thiện độ chính xác trong việc xác định ranh giới giữa các vùng như khu dân cư và khu vực xanh.

Kiến trúc DeepLabV3+

- *Backbone: Resnet50*
- *Atrous Convolution*
- *ASPP (Atrous Spatial Pyramid Pooling)*
- *Decoder*
- *Output Stride: 16*
- *Loss: DiceLoss*

U-Net

U-Net là một mô hình mạng nơ-ron tích chập được thiết kế đặc biệt cho các bài toán phân đoạn ảnh, ban đầu trong lĩnh vực y sinh học. Kiến trúc của U-Net có dạng hình chữ U, gồm hai phần chính: bộ mã hóa (encoder) và bộ giải mã (decoder), với các kết nối skip-connection giúp truyền thông tin từ tầng nông sang tầng sâu. Mô hình này có ưu điểm trong việc bảo toàn thông tin không gian và hoạt động hiệu quả với lượng dữ liệu huấn luyện không lớn và phù hợp với các cấu trúc địa lý như đường, sông, nhà cửa nhờ các đặc điểm sau:

- Kiến trúc đối xứng mã hóa – giải mã với kết nối tắt (skip connection): U-Net duy trì thông tin không gian từ tầng mã hóa sang tầng giải mã thông qua các kết nối tắt, cho phép khôi phục chi tiết ranh giới giữa các đối tượng địa lý như ranh giới giữa sông và đất, hoặc đường giao thông và khu dân cư.

- Hiệu quả với tập dữ liệu nhỏ: Trong nhiều trường hợp xây dựng bản đồ, dữ liệu gán nhãn có thể hạn chế do chi phí chú thích cao. U-Net có khả năng huấn luyện tốt ngay cả với bộ dữ liệu huấn luyện nhỏ, rất thích hợp trong bối cảnh bản đồ hóa địa phương hoặc theo vùng nhỏ.

- Xử lý ảnh đầu vào kích thước thay đổi: Nhờ kiến trúc đơn giản và khả năng mở rộng, U-Net có thể linh hoạt xử lý ảnh vệ tinh có kích thước thay đổi mà không cần tái thiết kế kiến trúc mạng.

Kiến trúc của U-Net

- *Backbone: Resnet50*
- *Decoder*
- *Skip connections*
- *Loss: DiceLoss*
- *Tích chập chuyển vị (Transposed Convolution)*

Feature Pyramid Network (FPN)

FPN là một kiến trúc phụ trợ mạnh mẽ thường được tích hợp vào các hệ thống phân đoạn ảnh để nâng cao khả năng nhận diện đối tượng ở nhiều kích thước và tỉ lệ khác nhau. FPN sử dụng cấu trúc mạng kim tự tháp gồm nhánh từ dưới lên (bottom-up) và nhánh từ trên xuống (top-down), kết hợp với các kết nối ngang giúp tổng hợp đặc trưng không gian và ngữ nghĩa từ nhiều tầng. Khi được kết hợp với mô hình như Mask R-CNN hoặc Faster R-CNN, FPN cho phép hệ thống phân đoạn chi tiết các cấu trúc nhỏ như biển báo, đường viền hoặc lề đường, các yếu tố rất quan trọng trong việc tạo bản đồ số chính xác cho điều hướng robot nhờ các yếu tố có thể kể đến:

- Tổng hợp đặc trưng từ nhiều cấp độ: FPN sử dụng kiến trúc dạng kim tự tháp với nhánh top-down kết hợp lateral connections, cho phép mô hình khai thác cả đặc trưng ngữ nghĩa từ tầng sâu và thông tin không gian từ tầng nông. Điều này giúp phát hiện đồng thời các đối tượng nhỏ (cột mốc, xe, ngã tư) và lớn (đồng ruộng, khu đô thị) trong ảnh vệ tinh.

- Tăng độ nhạy với đối tượng có kích thước nhỏ: Việc phân đoạn các thực thể nhỏ như biển báo giao thông, cầu vượt hoặc lối đi phụ thường bị bỏ sót nếu mô hình không có cơ chế đa tỉ lệ. FPN giải quyết điều này thông qua việc kết hợp các tầng đặc trưng có độ phân giải khác nhau.

- Dễ tích hợp vào hệ thống segmentation lớn: FPN thường được sử dụng như một khối nền (backbone) trong các mô hình mạnh như Mask R-CNN hoặc PANet, cho phép mở rộng linh hoạt trong hệ thống thiết kế bản đồ tự động và phân tích cảnh quan đô thị.

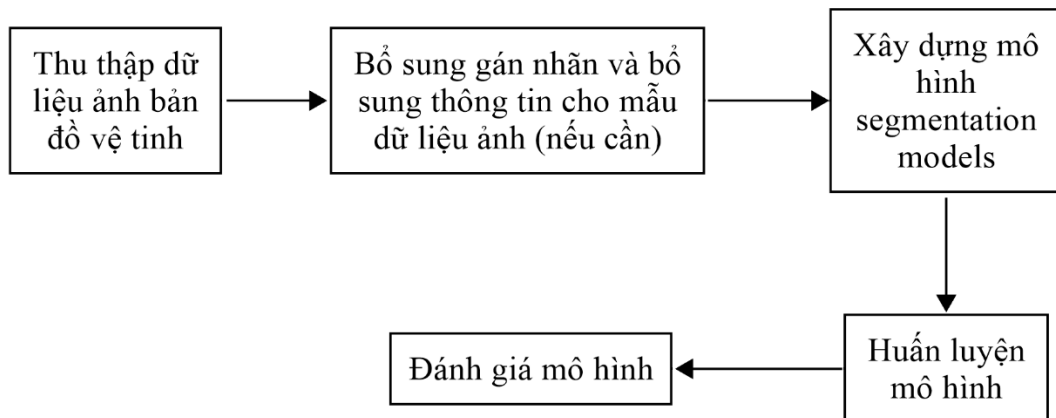
Bảng 3.1: Bảng phân tích các mô hình học sâu trong xây dựng mô hình phân đoạn bản đồ đường đi cho robot di động

Tiêu chí	U-Net	FPN	DeepLabV3+
Kiến trúc	Mạng encoder–decoder đối xứng, có skip connection giữ lại thông tin không gian	Kiến trúc kim tự tháp với bottom-up, top-down và lateral connections	Encoder–decoder với Atrous Spatial Pyramid Pooling (ASPP)
Khả năng phân đoạn đa tỉ lệ	Hạn chế, do đặc trưng chỉ được trích xuất ở một cấp độ	Rất tốt, kết hợp đặc trưng ở nhiều tầng độ phân giải khác nhau	Rất tốt, sử dụng ASPP để trích xuất ngữ cảnh đa tỉ lệ
Hiệu quả trên dữ liệu nhỏ	Tốt – hoạt động hiệu quả với tập dữ liệu huấn luyện nhỏ	Trung bình – yêu cầu nhiều dữ liệu hơn do tích hợp	Trung bình đến cao – huấn luyện tốt nhưng cần lượng dữ liệu tương đối

Tiêu chí	U-Net	FPN	DeepLabV3+
		nhiều tầng trích xuất	
Phân đoạn đối tượng nhỏ	Tốt với ranh giới rõ ràng	Rất tốt – kết hợp đặc trưng nhiều tầng giúp phát hiện chi tiết nhỏ	Tốt – ASPP mở rộng trường tiếp nhận để bao quát vùng nhỏ
Phù hợp với ảnh vệ tinh độ phân giải cao	Trung bình – cần chia nhỏ ảnh hoặc giảm độ phân giải	Tốt – có thể tích hợp vào hệ thống lớn như Mask R-CNN	Rất tốt – duy trì độ phân giải không gian nhờ atrous convolution
Khả năng khôi phục ranh giới chi tiết	Tốt – nhờ skip connection giữa encoder và decoder	Tốt – kết hợp đặc trưng cấp thấp và cao	Rất tốt – kết hợp đặc trưng cấp thấp trong decoder để tái tạo ranh giới chính xác
Mức độ phức tạp	Thấp – dễ cài đặt và huấn luyện	Trung bình – yêu cầu cấu trúc backbone và multi-path	Cao – sử dụng ASPP và cấu trúc phức tạp hơn

3.3. Các bước huấn luyện mô hình học sâu

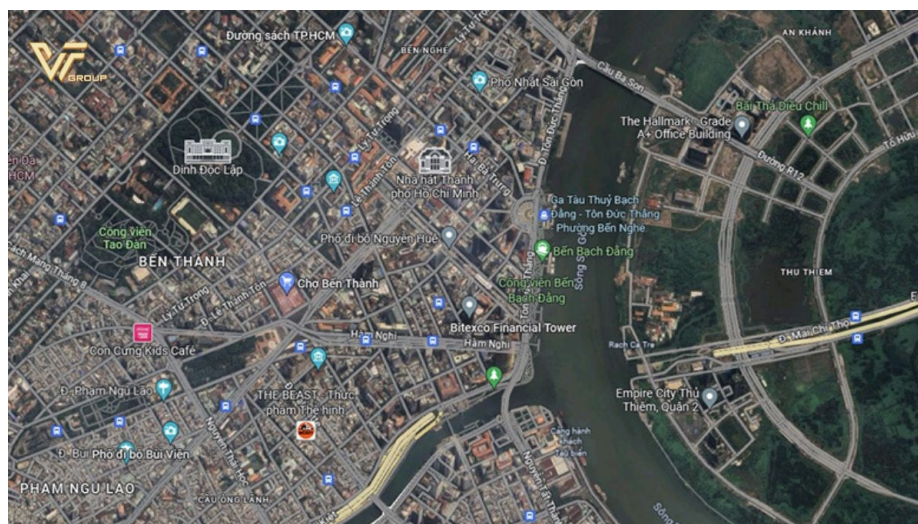
Để thực hiện huấn luyện mô hình học sâu trong trích xuất bản đồ ta thực hiện các bước như sơ đồ Hình 3.3.



Hình 3.2: Quy trình thực hiện huấn luyện mô hình

Bước 1: Thu thập và chuẩn hóa dữ liệu ảnh bản đồ vệ tinh

Tìm và thống kê các nguồn dataset về dữ liệu ảnh vệ tinh



Hình 3.3: Một hình ảnh vệ tinh trong 1 khu vực tại TP. Hồ Chí Minh

- Nguồn dữ liệu: Thu thập ảnh vệ tinh từ các nguồn như DeepGlobe, SpaceNet, Inria Aerial, Camvid, TGRS Road và Massachusetts Roads... hoặc các nhà cung cấp thương mại (Maxar, Planet). Đảm bảo ảnh có độ phân giải phù hợp với bài toán (ví dụ: độ phân giải không gian 10m cho Sentinel-2 hoặc cao hơn cho ảnh thương mại).

Ở đây ta sử dụng 2 tập dữ liệu TGRS Road và Massachusetts Roads:

- Tập dữ liệu TGRS Road được thu thập từ ảnh vệ tinh độ phân giải cao, cung cấp các cặp ảnh đầu vào và mặt nạ phân đoạn. Tập dữ liệu bao gồm các khu vực đô thị, nông thôn với độ đa dạng cao về địa hình và cấu trúc đường xá.

- Massachusetts Roads: Nó bao gồm hàng ngàn bức ảnh chụp từ trên không của các con đường ở Massachusetts. Điểm đặc biệt là mỗi bức ảnh này đều có một "đáp án" đi kèm: một bản đồ đen trắng chỉ rõ chính xác đâu là đường đi.

Mục đích chính là dùng bộ "đề và đáp án" này để huấn luyện và kiểm tra các mô hình AI, xem chúng có thể "học" cách tự nhận dạng đường đi một cách chính xác hay không.

Tiền xử lý:

- Cắt ảnh: Chia ảnh vệ tinh kích thước lớn thành các ô nhỏ (tiles) với kích thước cố định (ví dụ: 256x256 hoặc 512x512 pixel) để phù hợp với đầu vào của mô hình.

- Đồng nhất: đồng nhất các giá trị điểm ảnh hay pixel (ví dụ: chia cho 255 để đưa về khoảng $[0, 1]$); áp dụng các phép biến đổi dữ liệu để chuẩn hóa và tăng cường dữ liệu (augmentation), bao gồm: Thay đổi kích thước ảnh (resize); chuẩn

hóa theo thống kê (mean, std); các phép biến đổi ngẫu nhiên như xoay, lật, thay đổi độ sáng nhằm tăng tính đa dạng của tập huấn luyện.

- Loại bỏ nhiễu: Xử lý các vùng bị che phủ nếu cần.

Bước 2: Bổ sung gán nhãn và bổ sung thông tin cho mẫu dữ liệu ảnh (nếu cần)

Phân chia dữ liệu:

Việc xử lý các tập dữ liệu khổng lồ là một trong những thách thức lớn, đặc biệt là giới hạn về bộ nhớ RAM. Một trong những phương pháp hiệu quả và được áp dụng rộng rãi để giải quyết vấn đề này là sử dụng Data Generator (Trình tạo dữ liệu). Data Generator là một kỹ thuật cho phép tải dữ liệu theo từng phần nhỏ (gọi là "batch") một cách linh hoạt trong quá trình huấn luyện mô hình. Thay vì nạp toàn bộ hàng Gigabyte hay thậm chí Terabyte dữ liệu vào bộ nhớ cùng một lúc dẫn đến tình trạng tràn bộ nhớ thì ta sẽ tạo ra một luồng dữ liệu. Khi bắt đầu quá trình huấn luyện, thay vì đọc tất cả ảnh và nhãn vào RAM, Data Generator sẽ chỉ cung cấp một lượng dữ liệu vừa đủ cho một vòng lặp huấn luyện. Sau khi mô hình hoàn thành việc học trên gói dữ liệu thứ nhất, bộ nhớ được giải phóng và Data Generator sẽ tiếp tục tải gói dữ liệu kế tiếp. Quá trình này lặp lại cho đến khi mô hình đã học trên toàn bộ tập dữ liệu.

Bổ sung gán nhãn và bổ sung thông tin cho mẫu dữ liệu ảnh

Việc bổ sung gán nhãn và thông tin cho dữ liệu ảnh vệ tinh là một bước quan trọng để nâng cao chất lượng và tính ứng dụng của dữ liệu, đặc biệt trong các nhiệm vụ như phân loại, phân đoạn, hoặc phát hiện đối tượng. Quy trình này không chỉ tăng cường khả năng phân tích mà còn đảm bảo dữ liệu phù hợp với các tiêu chuẩn khoa học và kỹ thuật.

Định nghĩa lớp: Xác định các lớp cần phân đoạn (ví dụ: đường, nhà, cây, nước, đất trống). Mỗi lớp được gán một giá trị số duy nhất (ví dụ: đường = 1, nhà = 2, cây = 3, ...).

Annotations:

+ Công cụ gán nhãn: Sử dụng các công cụ như QGIS, LabelMe, hoặc các nền tảng như Labelbox, Supervisely để tạo mặt nạ phân đoạn (segmentation mask). Mỗi pixel trong ảnh ground truth sẽ được gán một giá trị tương ứng với lớp của nó.

+ Định dạng mặt nạ: Mặt nạ thường là ảnh đơn kênh (grayscale) hoặc ảnh RGB với mỗi lớp được mã hóa bằng một màu hoặc giá trị cụ thể.

+ Độ chính xác: Đảm bảo annotations được thực hiện chính xác, đặc biệt với các đối tượng có ranh giới phức tạp như đường hoặc centerline.

- Ảnh centerline (nếu cần)

Centerline thường được sử dụng trong các bài toán phân đoạn đường hoặc mạng lưới tuyến tính (như đường giao thông, sông). Centerline là một biểu diễn dạng đường mảnh (1-pixel rộng) của các đối tượng tuyến tính.

Cách tạo:

+ Từ mặt nạ phân đoạn: Sử dụng các thuật toán như skeletonization (trong thư viện OpenCV hoặc scikit-image) để chuyển đổi mặt nạ đường thành centerline.

+ Gán nhãn thủ công: Nếu độ chính xác cao là cần thiết, có thể sử dụng công cụ như QGIS để vẽ centerline thủ công.

Định dạng: Centerline thường được lưu dưới dạng ảnh nhị phân (binary mask) với giá trị 1 cho centerline và 0 cho nền.

- Kiểm tra chất lượng annotations

Kiểm tra các lỗi như nhãn sai, ranh giới không chính xác, hoặc các vùng bị bỏ sót.

Đảm bảo ảnh đầu vào và ảnh ground truth có cùng kích thước và căn chỉnh chính xác.

Bước 3: Xây dựng mô hình segmentation models

Bước 4: Huấn luyện mô hình

Bước 5: Đánh giá mô hình

Sử dụng các chỉ số đánh giá như:

- IoU (Intersection over Union) hay còn được gọi là Jaccard Index, là chỉ số đo lường mức độ trùng khớp giữa vùng dự đoán và vùng thực tế. IoU được tính bằng tỉ số giữa phần giao (intersection) và phần hợp (union) của hai tập hợp pixel. Chỉ số IoU có giá trị từ 0 đến 1, trong đó giá trị càng cao cho thấy mức độ khớp giữa dự đoán và nhãn thực càng tốt. Đây là chỉ số phổ biến trong các bài toán segmentation và object detection.

- Dice Coefficient cũng đo lường mức độ trùng khớp giữa hai vùng ảnh, Dice Coefficient có giá trị từ 0 đến 1, trong đó 1 là mức độ trùng khớp hoàn hảo. Chỉ

số này thường được sử dụng nhiều trong phân đoạn ảnh y tế vì nó nhạy với các vùng nhỏ, nơi mà độ chênh lệch nhỏ có thể ảnh hưởng lớn đến kết quả tổng thể.

- Pixel Accuracy: là tỷ lệ giữa số lượng pixel được phân loại đúng và tổng số pixel trong ảnh. Chỉ số này đơn giản và dễ tính, tuy nhiên không phản ánh tốt hiệu suất mô hình nếu dữ liệu bị mất cân bằng (ví dụ khi số pixel nền chiếm ưu thế).

Kết luận chương III

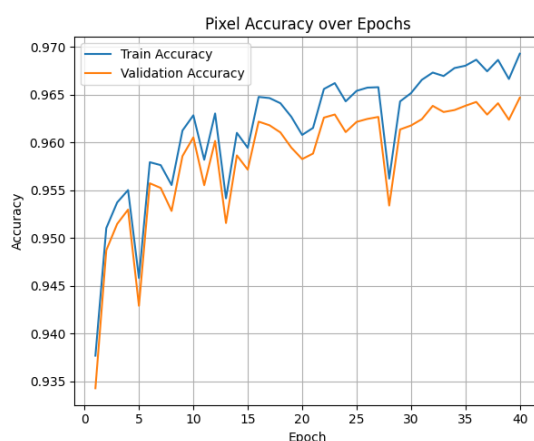
Chương 3 mô tả quy trình triển khai chi tiết đề tài, từ thu thập dữ liệu ảnh vệ tinh, xử lý tiền dữ liệu, lựa chọn mô hình học sâu (U-Net, DeepLabV3+, FPN), đến quá trình huấn luyện và đánh giá. Các mô hình được áp dụng để trích xuất bản đồ số làm cơ sở cho việc thiết kế đường đi. Ngoài ra, các thuật toán tìm đường như A* và Dijkstra được áp dụng để tối ưu hóa quỹ đạo.

CHƯƠNG 4 - KẾT QUẢ VÀ ĐỀ XUẤT

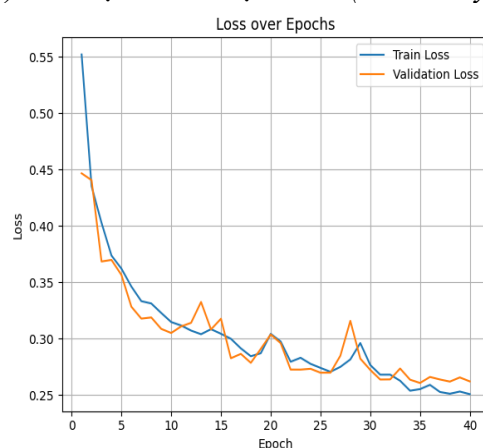
4.1. Kết quả huấn luyện và đánh giá mô hình

Sau khi triển khai huấn luyện dữ liệu ảnh trên các mô hình khác nhau đã đề cập, kết quả của quá trình huấn luyện và đánh giá được trình bày từ hình 4.1 đến 4.3.

Mô hình DeepLabV3+



a) Đồ thị chỉ số hiệu suất (Accuracy)



b) Đồ thị chỉ số hàm mất mát (Loss)



c) Đồ thị chỉ số IoU trung bình (Mean IoU)

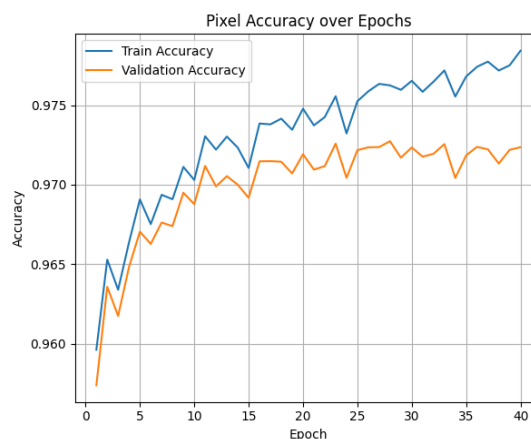
Hình 4.1: Kết quả các chỉ số huấn luyện và đánh giá của mô hình DeepLabV3+

Với chỉ số hàm mất mát (Loss), xu hướng đồ thị trên cả hai tập giảm nhanh chóng trong 10 epoch đầu tiên, từ mức 0.5 xuống dưới 0.3. Sau đó, tiếp tục giảm chậm hơn và ổn định dần về mức 0.25 vào cuối quá trình huấn luyện (ở epoch thứ 40). Đường cong đồ thị của chỉ số hàm mất mát không có dao động lớn, chứng tỏ quá trình tối ưu khi huấn luyện mô hình diễn ra tương đối tốt. Tuy vậy, cũng xuất hiện vài dao động nhẹ ở đồ thị của tập đánh giá, cho thấy mô hình thiếu ổn định cục bộ ở mức độ nào đó.

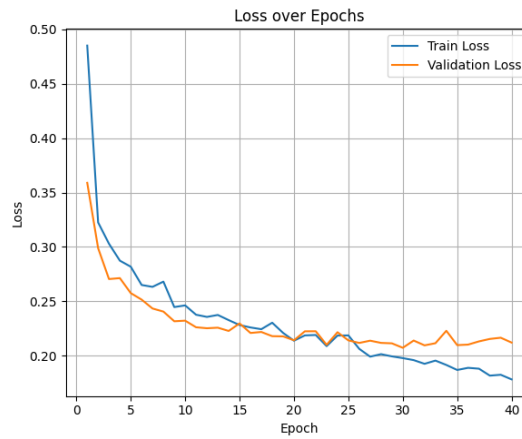
Với chỉ số IoU trung bình, giá trị đồ thị tăng đều đặn qua các epoch, đạt ngưỡng khoảng 0.62 trên tập huấn luyện và 0.58 trên tập đánh giá. Có thể thấy, mô hình thể hiện rõ sự cải thiện liên tục trong việc xác định đúng vùng phân đoạn đường đi trên tập ảnh bản đồ. Bên cạnh đó, đồ thị xuất hiện sự hội tụ bắt đầu rõ ràng sau khoảng epoch 35 cho thấy mô hình đã gần đạt ngưỡng học tối đa với bộ dữ liệu đầu vào.

Mô hình Unet

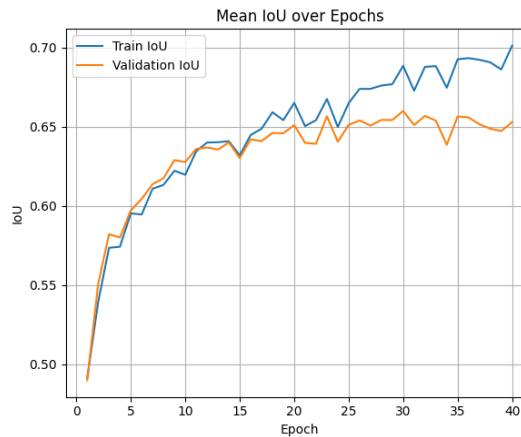
Hình 4.2 biểu diễn kết quả các chỉ số huấn luyện và đánh giá của mô hình Unet. Trên tập huấn luyện, độ chính xác tăng ổn định và đạt ngưỡng khoảng 0.978, cho thấy mô hình học tốt trên dữ liệu huấn luyện. Tuy nhiên, trên tập đánh giá, sau khi tăng nhanh đến khoảng epoch 15, đồ thị bắt đầu dao động mạnh và không còn cải thiện rõ rệt, dao động quanh mức 0.972. Do đó, mô hình có khả năng bắt đầu bị hiện tượng “overfitting” từ rất sớm (khoảng sau epoch 15).



a) Đồ thị chỉ số hiệu suất (Accuracy)



b) Đồ thị chỉ số hàm mất mát (Loss)



c) Đồ thị chỉ số IoU trung bình (Mean IoU)

Hình 4.2: Kết quả các chỉ số huấn luyện mô hình Unet

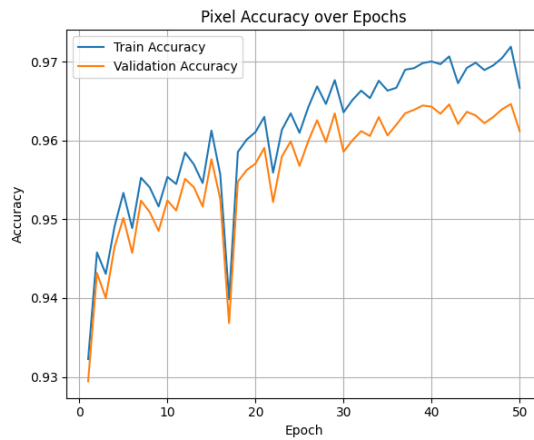
Với kết quả chỉ số hàm mất mát của mô hình Unet, xu hướng đồ thị trên tập huấn luyện giảm đều đặn và đạt mức thấp khoảng 0.18, phản ánh việc tối ưu hoá hoạt động hiệu quả. Ngược lại, xu hướng đồ thị trên tập đánh giá lại dao động khá mạnh, mặc dù đạt mức thấp nhất khoảng tại epoch 15–17, nhưng sau đó không ổn định và không tiếp tục giảm. Sự dao động này cho thấy mô hình không tổng quát hoá tốt trên tập dữ liệu đánh giá.

Với chỉ số IoU trung bình, đồ thị trên tập huấn luyện tăng đều và đạt mức khá cao, khoảng 0.70. Kết quả này cho thấy mô hình học tốt về phân đoạn vùng trên ảnh huấn luyện. Tuy nhiên, đồ thị của chỉ số này trên tập đánh giá cũng dao động mạnh sau epoch 15, chỉ đạt khoảng 0.65, và không còn tăng rõ rệt. Điều này chứng tỏ mô hình xảy ra tình trạng không ổn định trong quá trình kiểm nghiệm trên tập đánh giá.

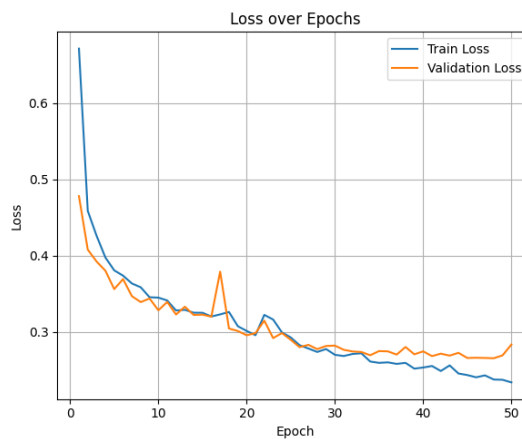
Nhìn chung, thông qua ba chỉ số đánh giá trên tập huấn luyện, có thể thấy kết quả đều dao động và không ổn định sau khoảng epoch 15. Hay nói cách khác,

mô hình Unet có xảy ra hiện tượng “over fitting” tương đối đáng kể, cho thấy mô hình bắt đầu ghi nhớ thay vì học khái quát trên tập dữ liệu ảnh đầu vào.

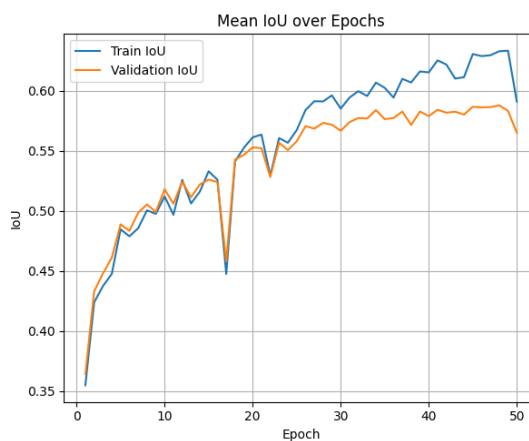
Mô hình FPN



a) Đồ thị chỉ số hiệu suất (Accuracy)



b) Đồ thị chỉ số hàm mất mát (Loss)



c) Đồ thị chỉ số IoU trung bình (Mean IoU)

Hình 4.3: Kết quả các chỉ số huấn luyện mô hình FPN

Kết quả các chỉ số huấn luyện của mô hình FPN được trình bày như trên hình 4.3. Đồ thị chỉ số hiệu suất trên cả tập huấn luyện và đánh giá tăng ổn định qua các epoch và đạt khoảng trên 0.96, cho thấy mô hình học đều và hiệu quả. Đặc

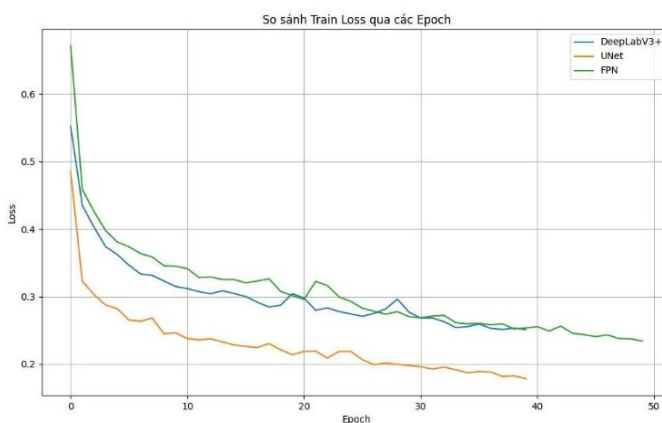
biệt xu hướng đồ thị trên tập đánh giá di chuyển gần với đường của tập huấn luyện, phản ánh quá trình học cân bằng và ổn định. Khoảng cách giữa hai đường nhỏ, cho thấy hiện tượng overfitting thấp hơn so với hai mô hình DeepLabV3+ và UNet.

Với chỉ số của hàm mất mát, các giá trị trên cả đồ thị của tập huấn luyện và đánh giá đều đạt mức thấp khoảng 0.23, không có dấu hiệu dao động bất thường. Giá trị trên đồ thị của tập đánh giá cao hơn một chút (khoảng 0.28) nhưng biến động không lớn và giảm dần theo thời gian, phản ánh mô hình có khả năng tổng quát hoá khá tốt. Nhìn chung, điểm khác biệt giữa hai đường đồ thị của chỉ số hàm mất mát cho thấy quá trình huấn luyện ổn định và đáng tin cậy.

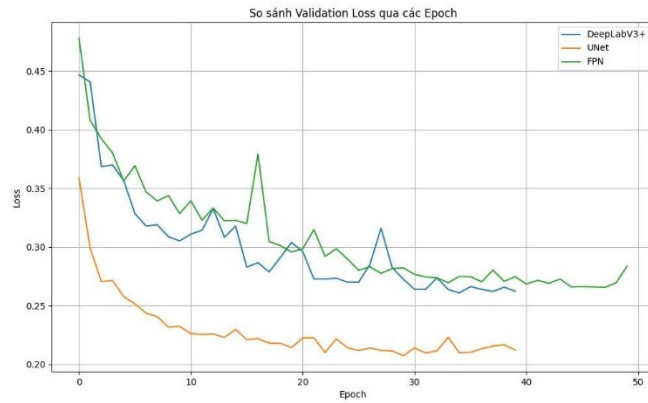
Với chỉ số cuối cùng là IoU trung bình, đường đồ thị trên tập huấn luyện có xu hướng tăng đều, đạt ngưỡng khoảng 0.63. Kết quả này chứng tỏ mô hình học tốt khả năng phân đoạn. Tương tự, đường đồ thị trên tập đánh giá tăng theo và đạt khoảng 0.59, ít dao động. Sự hội tụ bắt đầu rõ rệt sau khoảng 35-40 epoch, cho thấy mô hình FPN học từ từ nhưng chắc chắn.

4.2. Kết quả so sánh giữa các mô hình

Hình 4.4 đến hình 4.6 biểu diễn kết quả so sánh trên các chỉ số khác nhau giữa các mô hình.



a) So sánh chỉ số hàm mất mát trên tập huấn luyện giữa các mô hình

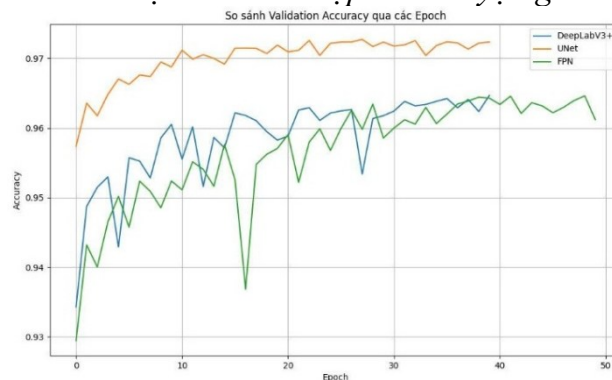


b) So sánh chỉ số hàm mất mát trên tập đánh giá giữa các mô hình
 Hình 4.4: Kết quả so sánh chỉ số hàm mất mát giữa các mô hình

Kết quả trên hình 4.4 cho thấy UNet tiếp tục dẫn đầu với chỉ số hàm mất mát thấp nhất (0.18 trên tập huấn luyện và 0.21 trên tập đánh giá). Đồ thị của mô hình FPN và DeepLabV3+ có giá trị cao hơn so với Unet (FPN: 0.23 trên tập huấn luyện và 0.28 trên tập đánh giá; DeepLabV3+: 0.25 trên tập huấn luyện và 0.26 trên tập đánh giá). Tuy nhiên, biểu đồ cho thấy chỉ số của UNet dao động nhiều ở tập đánh giá, phản ánh mô hình dễ bị hiện tượng “overfitting”. FPN thể hiện tăng nhẹ nhưng ổn định trên tập đánh giá, cho thấy mô hình học đều và tránh được overfitting.



a) So sánh chỉ số hiệu suất trên tập huấn luyện giữa các mô hình



b) So sánh chỉ số hiệu suất trên tập đánh giá giữa các mô hình
 Hình 4.5: Kết quả so sánh chỉ số hàm hiệu suất giữa các mô hình

Dựa vào hình 4.5 có thể thấy mô hình UNet có chỉ số hiệu suất cao nhất trên cả tập huấn luyện (0.978) và tập đánh giá (0.972). Tuy mô hình UNet vượt trội về

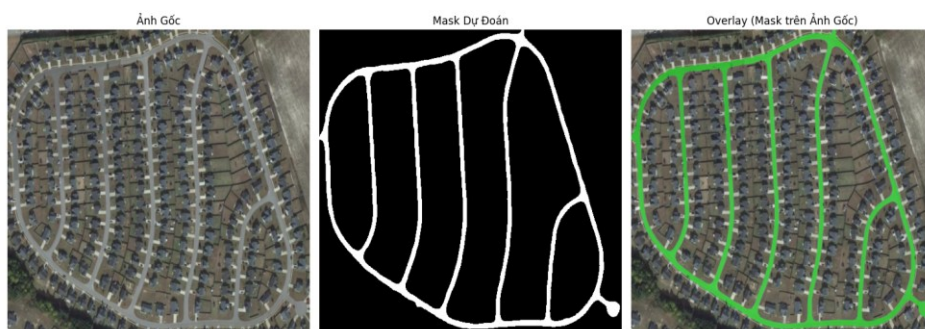
con số, nhưng các dao động lớn trong quá trình training (đã phân tích trước) khiến kết quả không thật sự đáng tin cậy. Mô hình DeepLabV3+ và FPN đều đạt mức hiệu suất 0.970, thấp hơn nhưng ổn định hơn. Mặc dù UNet có chỉ số cao nhất, nhưng mô hình FPN và DeepLabV3+ đáng tin cậy hơn nhờ tính ổn định và hội tụ đều đặn. Kết quả phân tích sự so sánh giữa các mô hình mạng được tác giả trình bày như trên bảng 4.2.

Bảng 4.1: So sánh giá trị chỉ số giữa các mô hình mạng

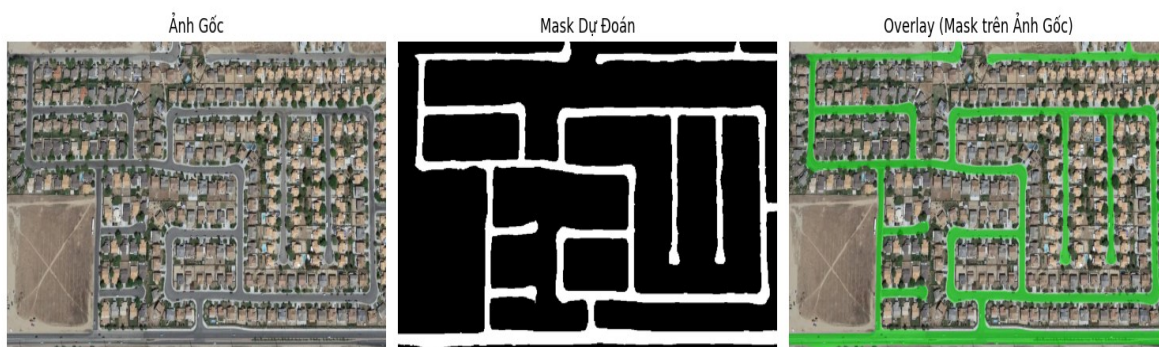
Chỉ số	Tập dữ liệu	Mô hình		
		DeepLabV3+	UNet	FPN
Accuracy	Train	0.970	0.978	0.970
	Validation	0.965	0.972	0.965
Loss	Train	0.25	0.18	0.23
	Validation	0.26	0.21	0.28
Mean IoU	Train	0.62	0.70	0.63
	Validation	0.58	0.65	0.59

4.3. Kết quả dự đoán và trích xuất bản đồ đường đi trên tập dữ liệu kiểm tra

Các mô hình sau khi huấn luyện sẽ được triển khai dự đoán trên hình ảnh dữ liệu thực để đánh giá khả năng vận hành trong các điều kiện khác nhau. Hình 4.6 thể hiện kết quả dự đoán bản đồ trích xuất đường đi của mô hình DeepLabV3+ trên các khu vực riêng biệt.



a) Kết quả dự đoán trên bản đồ địa hình vòng cung



b) Kết quả dự đoán trên bản đồ địa hình ô lưới



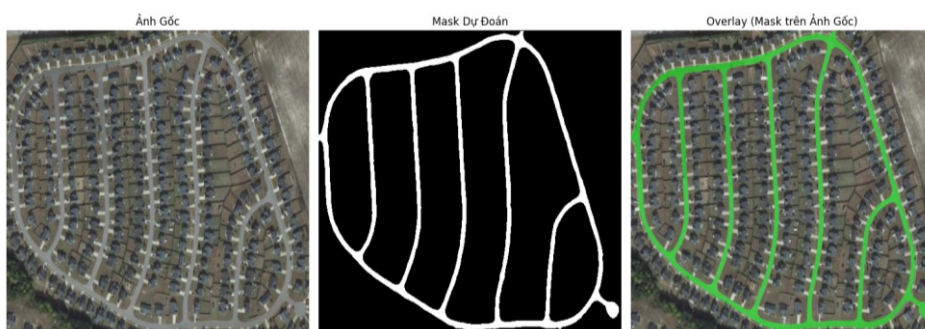
c) Kết quả dự đoán trên bản đồ địa hình thành phố



d) Kết quả dự đoán trên bản đồ địa hình phân nhánh phức tạp

Hình 4.6: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình DeepLabV3+

Hình 4.7 thể hiện kết quả dự đoán bản đồ trích xuất đường đi của mô hình Unet trên các khu vực riêng biệt.



a) Kết quả dự đoán trên bản đồ địa hình vòng cung



b) Kết quả dự đoán trên bản đồ địa hình ô lưới



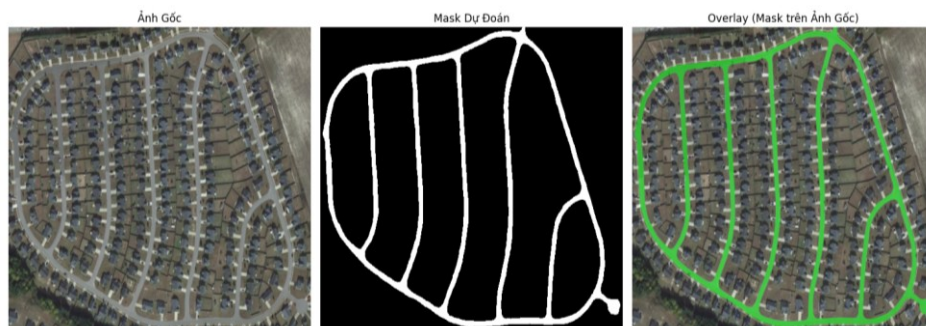
c) Kết quả dự đoán trên bản đồ địa hình thành phố



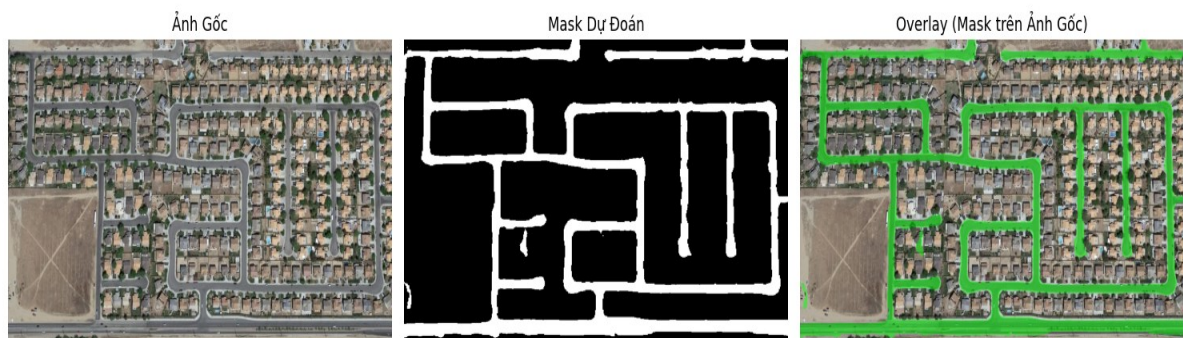
d) Kết quả dự đoán trên bản đồ địa hình phân nhánh phức tạp

Hình 4.7: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình Unet

Hình 4.8 thể hiện kết quả dự đoán bản đồ trích xuất đường đi của mô hình FPN trên các khu vực riêng biệt.



a) Kết quả dự đoán trên bản đồ địa hình vòng cung



b) Kết quả dự đoán trên bản đồ địa hình ô lưới



c) Kết quả dự đoán trên bản đồ địa hình thành phố



d) Kết quả dự đoán trên bản đồ địa hình phân nhánh phức tạp

Hình 4.8: Kết quả dự đoán và trích xuất đường đi trên bản đồ của mô hình FPN

Kết quả dự đoán trên các dạng bản đồ khác nhau của ba mô hình thiết kế được tổng hợp và phân tích như trên bảng 4.2:

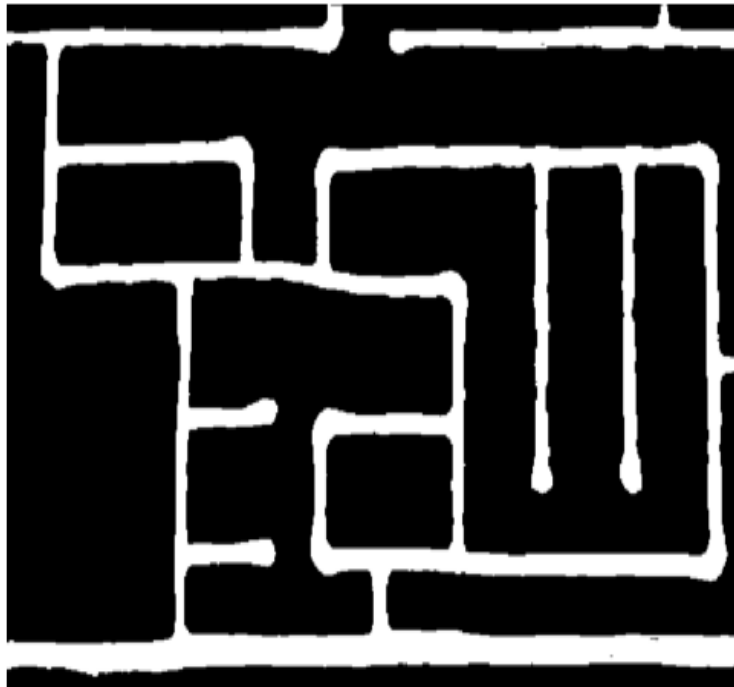
Bảng 4.2: So sánh ưu và nhược điểm giữa các mô hình

Mô hình	Ưu điểm	Nhược điểm	Mức độ phù hợp triển khai thực tế
DeepLabV3+	Đường phân đoạn rõ, sắc nét, độ chính xác cao	Bỏ sót đường nhỏ, giảm hiệu suất ở ảnh có nhiễu	Có, nhưng cần kiểm soát hiện tượng overfitting. Cần khử nhiễu tốt cho ảnh đầu vào.
Unet	Nhận diện được đường lớn bao quanh khu vực	Nhận diện yếu, không ổn định, dễ nhầm lẫn, hay xảy ra hiện tượng đứt nét.	Hiệu suất không cao, không nên dùng trực tiếp
FPN	Phân đoạn mượt, ổn định, bao quát tốt đường nhỏ, trung bình và có độ rõ nhánh phức tạp	Độ sắc nét không cao bằng DeepLabV3+	Thích hợp và ổn định nhất, phù hợp nhiều loại ảnh đầu vào.

4.4. Kết quả hoạch định đường đi sử dụng các thuật toán tìm kiếm

Sau khi bản đồ số được trích xuất thành công từ ảnh vệ tinh bằng mô hình học sâu, trong các bước tiếp theo của đề tài là triển khai và đánh giá các thuật toán hoạch định đường đi (Path Planning) trên bản đồ này. Mục tiêu của giai đoạn này là tìm ra một quỹ đạo di chuyển tối ưu (ngắn nhất) cho robot từ một điểm xuất phát đến điểm đích cho trước. Ở đây, ta lựa chọn mô hình bản đồ địa hình ô lưới đã trích xuất được ở trên và để đánh giá hiệu quả, ba thuật toán hoạch định đường đi kinh điển và các biến thể của chúng đã được lựa chọn để thực nghiệm, bao gồm:

- Thuật toán Dijkstra: Một thuật toán nền tảng đảm bảo tìm ra đường đi ngắn nhất trong đồ thị có trọng số không âm.
- Thuật toán A*: Một phiên bản cải tiến của Dijkstra, sử dụng hàm Heuristic để tăng tốc độ tìm kiếm bằng cách ưu tiên các hướng về phía đích.
- Thuật toán Theta*: Một biến thể của A* cho phép tạo ra các đường đi any-angle (mọi góc), giúp lộ trình trở nên mượt mà và ngắn hơn so với các đường đi bị giới hạn trên lưới.



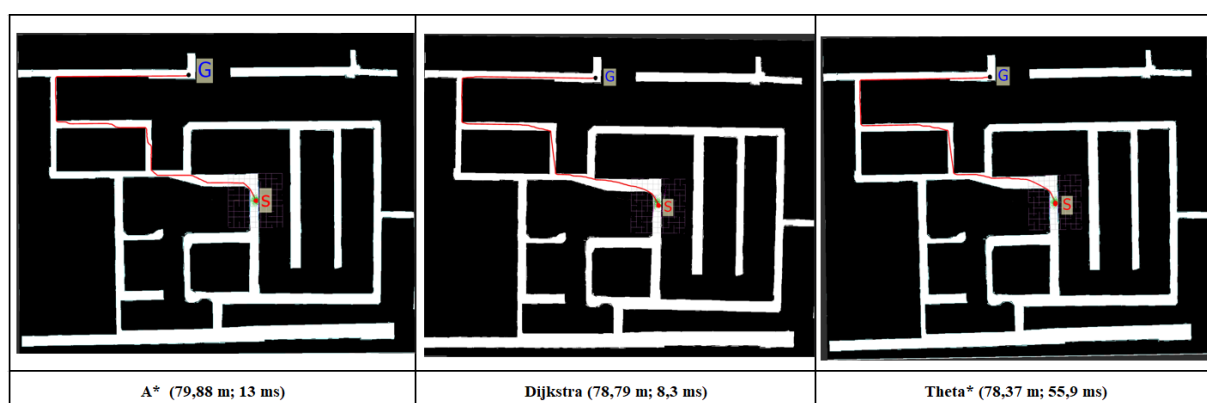
Hình 4.9: Mô hình bản đồ ô lưới để thực hiện hoạch định đường dẫn

Các thuật toán được thử nghiệm trên ba kịch bản (Path 1, Path 2, Path 3) với các cặp điểm xuất phát (Start Point) S và điểm đích (Goal Point) G khác nhau. Tiêu chí đánh giá chính là tổng chiều dài đường đi tìm được (tính bằng mét) và thời gian tìm kiếm đường đi (tính bằng mili giây) sử dụng 3 thuật toán trên để đảm bảo tính hiệu quả và được mô tả ở các bảng số liệu dưới đây:

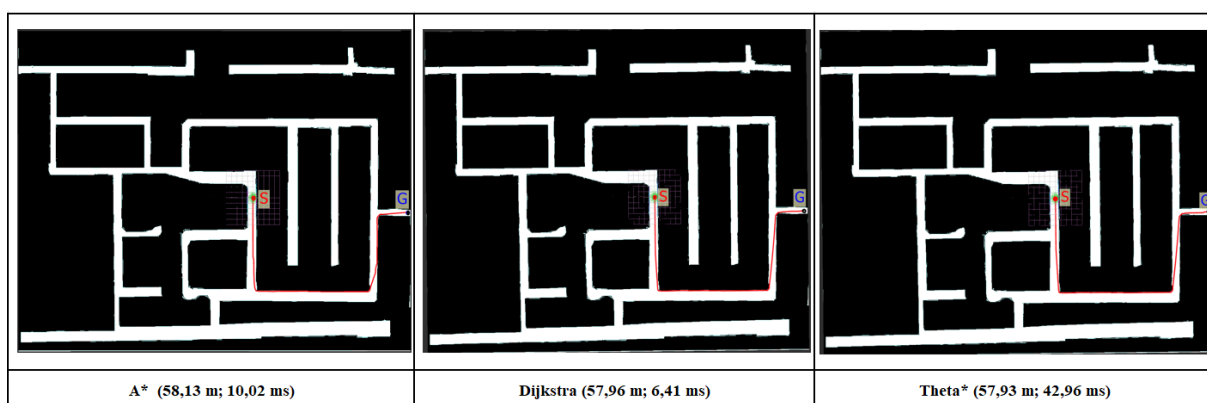
Bảng 4.3: Số liệu kết quả tìm kiếm đường đi

Kịch bản	Path 1		Path 2		Path 3	
Thuật toán	Quãng đường (m)	Thời gian (ms)	Quãng đường (m)	Thời gian (ms)	Quãng đường (m)	Thời gian (ms)
A*	79,88	13	58,13	10,02	57,96	9,7
Dijkstra	78,79	8,3	57,96	6,41	57,36	6,2
Theta*	78,37	55,9	57,93	42,96	57,35	41,5

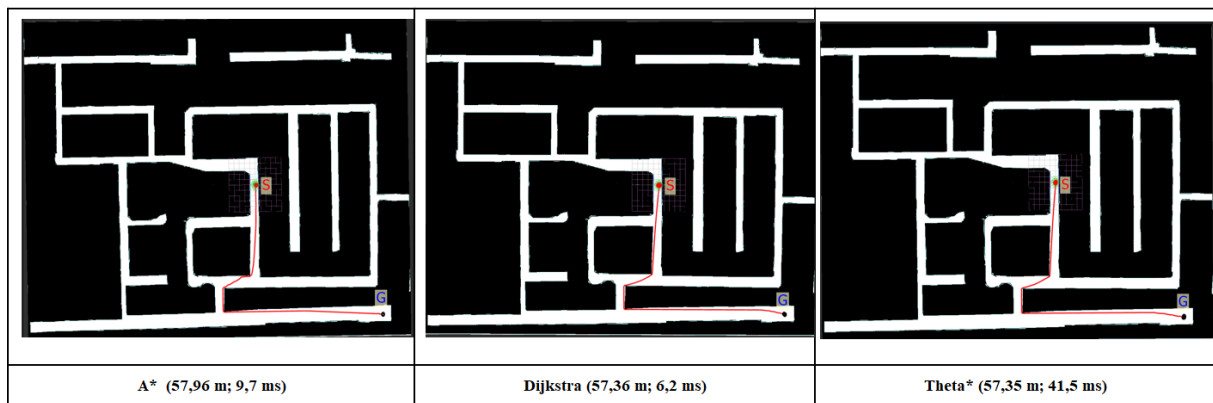
Sự khác biệt về hiệu suất này bắt nguồn từ bản chất của từng thuật toán. Dijkstra luôn tìm kiếm đường đi ngắn nhất từ điểm xuất phát đến mọi đỉnh khác, trong khi A* sử dụng hàm Heuristic để tăng tốc quá trình tìm kiếm nhưng không đảm bảo đường đi mượt nhất. Theta*, một biến thể của A*, cải tiến bằng cách cho phép các đường đi any-angle (mọi góc), không bị giới hạn bởi các cạnh của lưới, thông qua việc kiểm tra tầm nhìn (line-of-sight), từ đó tạo ra các lộ trình thẳng và ngắn hơn.



Hình 4.10: Kịch bản thứ nhất (Path 1)



Hình 4.11: Kịch bản thứ 2 (Path 2)



Hình 4.12: Kịch bản thứ 3 (Path 3)

Về độ dài đường đi: Thuật toán Theta* cho kết quả tốt nhất ở cả ba trường hợp, lần lượt là 78.37 m, 57.93 m và 57.35 m. Trong khi đó, thuật toán A* có xu hướng tạo ra các đường đi dài hơn với các giá trị tương ứng là 79.88 m, 58.13 m và 57.96 m. Dijkstra cho kết quả trung gian, chỉ nhỉnh hơn Theta* một chút về mặt độ dài. Sự khác biệt này chủ yếu xuất phát từ đặc điểm hình học của các thuật toán. Cụ thể, Theta* là biến thể của A* cho phép tạo ra các đoạn thẳng trực tiếp giữa các điểm không bị vật cản ngăn cách, nhờ đó giảm thiểu số bước rẽ và tổng độ dài đường đi.

Về thời gian tính toán: Ngược lại với hiệu quả về quãng đường, thuật toán Theta* cho thời gian xử lý lâu nhất so với hai thuật toán còn lại. Cụ thể, thời gian xử lý của Theta* lần lượt là 55.9 ms, 42.96 ms và 41.5 ms cao hơn đáng kể so với A* và Dijkstra. Dijkstra là thuật toán có thời gian xử lý thấp nhất trong cả ba tình huống, cho thấy hiệu quả cao về mặt tính toán, mặc dù nó không tối ưu về độ dài đường đi do không xem xét yếu tố Heuristic. A* đứng giữa về mặt thời gian xử lý nhờ việc sử dụng hàm Heuristic để dẫn hướng tìm kiếm, qua đó cân bằng giữa tốc độ và chất lượng đường đi.

Về đặc điểm hình học của đường đi: Quan sát từ các Hình 4.10 đến Hình 4.12 trong 3 kịch bản, ta có thể thấy A* và Dijkstra tạo ra các đường đi với nhiều khúc gấp khúc, tuân theo lưới ô vuông của bản đồ. Trong khi đó, Theta* cho phép các đoạn đường đi chéo, gần giống với đường thẳng hơn, phản ánh đúng khả năng cải tiến của thuật toán này trong việc tạo ra các đường đi mượt và ngắn hơn. Tuy nhiên, chính vì việc kiểm tra liên tục giữa các điểm không liền kề mà Theta* cần nhiều thời gian xử lý hơn.

Bảng 4.4: Đánh giá hiệu suất trung bình của các thuật toán đối với 3 kịch bản

Thuật toán	Độ dài trung bình (m)	Thời gian trung bình (ms)
A*	65.99	10.91
Dijkstra	64.70	6.97
Theta*	64.55	46.12

Từ các kết quả trung bình thu được, ta có thể thấy:

- Theta* phù hợp trong các tình huống yêu cầu đường đi ngắn nhất và chất lượng di chuyển cao, nhưng không giới hạn về thời gian tính toán.
- Dijkstra là lựa chọn tối ưu trong các hệ thống thời gian thực, tối ưu thời gian tìm kiếm hơn so với đường đi.
- A* là thuật toán đảm bảo được cả hai yếu tố về quãng đường và thời gian tìm kiếm, thích hợp cho các ứng dụng yêu cầu sự cân bằng giữa độ chính xác và tốc độ.

Kết luận chương 4

Trong chương 4, đề tài đã trình bày chi tiết kết quả thực nghiệm của quá trình huấn luyện mô hình học sâu trong trích xuất bản đồ và so sánh hiệu quả của các thuật toán thiết kế quỹ đạo di chuyển cho robot di động. Thông qua các thử nghiệm trên tập dữ liệu ảnh vệ tinh, mô hình DeepLabV3+ cho thấy khả năng phân đoạn vượt trội hơn so với U-Net và FPN, cả về độ chính xác và mức độ hội tụ. Các thuật toán lập kế hoạch đường đi như A*, Dijkstra và Theta* cũng được triển khai trên bản đồ số thu được. Kết quả phân tích đã làm rõ ưu nhược điểm của từng mô hình và thuật toán, từ đó khẳng định tính khả thi và hiệu quả của việc kết hợp học sâu trong xây dựng bản đồ với các thuật toán lập kế hoạch đường đi trong bài toán điều hướng robot di động.

KẾT LUẬN VÀ ĐỀ XUẤT

1. Kết luận

Đề tài “*Nghiên cứu xây dựng bản đồ số sử dụng phương pháp học sâu và thiết kế quỹ đạo cho robot di động*” đã tập trung vào việc tích hợp các phương pháp trong lĩnh vực trí tuệ nhân tạo và các thuật toán tìm kiếm, cụ thể là học sâu (deep learning), với các thuật toán hoạch định đường đi sử dụng thuật toán tìm kiếm để nâng cao khả năng tự hành của robot di động trong môi trường thực tế. Trong quá trình thực hiện, đề tài đã đạt được các kết quả nổi bật như sau:

- Đã nghiên cứu tổng quan lý thuyết về học sâu, các mô hình mạng phân đoạn ảnh (U-Net, FPN, DeepLabV3+), cũng như các thuật toán lập kế hoạch quỹ đạo (Dijkstra, A*, Theta*).

- Xây dựng và huấn luyện thành công mô hình học sâu để trích xuất bản đồ đường đi từ ảnh vệ tinh.

- Triển khai và đánh giá hiệu quả của các thuật toán thiết kế đường đi trên bản đồ số.

- Đánh giá hiệu suất các mô hình qua các chỉ số định lượng, từ đó đề xuất mô hình kết hợp hiệu quả cho bài toán điều hướng robot di động.

Từ những kết quả đạt được, đề tài đã góp phần khẳng định tính khả thi và hiệu quả của việc áp dụng học sâu trong xây dựng bản đồ số và điều hướng robot di động, đồng thời tạo nền tảng cho việc phát triển các hệ thống robot thông minh hơn trong tương lai.

2. Hướng phát triển của đề tài

Bên cạnh các kết quả đã đạt được, đề tài vẫn còn nhiều khía cạnh có thể được nghiên cứu và phát triển thêm trong tương lai:

- Mở rộng mô hình học sâu: Thử nghiệm các kiến trúc mạng tiên tiến hơn như Transformer-based segmentation (SegFormer, Mask2Former) để nâng cao độ chính xác và khả năng tổng quát trên nhiều môi trường khác nhau.

- Tối ưu hóa thời gian xử lý sử dụng các kỹ thuật hoặc thuật toán khác nhau để phù hợp với các hệ thống robot có tài nguyên tính toán hạn chế.

- Tích hợp SLAM thời gian thực: Áp dụng học sâu vào hệ thống SLAM trực tiếp để robot có thể vừa xây dựng bản đồ, vừa định vị và lập kế hoạch đường đi trong môi trường động.

- Triển khai thực nghiệm trên robot thực tế: Chuyển từ mô phỏng sang thử nghiệm thực tế để đánh giá tính ứng dụng của mô hình trong môi trường thực như trường học, nhà xưởng, bệnh viện, hoặc khu đô thị.

- Phát triển hệ thống đa robot: Mở rộng nghiên cứu sang hệ thống phối hợp nhiều robot, chia sẻ dữ liệu bản đồ và cùng nhau thực hiện nhiệm vụ một cách tối ưu.

Từ kết quả của đề tài và những định hướng phát triển trên sẽ góp phần nâng cao tính ứng dụng của đề tài trong thực tiễn, đóng góp vào sự phát triển chung đối với lĩnh vực robot di động nhất là trong bối cảnh cách mạng công nghiệp 4.0 và chuyển đổi số hiện nay.

TÀI LIỆU THAM KHẢO

- [1] C. R. Akupati, "Intelligent Outdoor Navigation of a Mobile Robot Platform Using a Low Cost High Precision RTK-GPS and Obstacle Avoidance System," 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceeding of Advances in Neural Information Processing Systems (NIPS)*, pp. 1097-1105, 2012.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning Nature," 2015.
- [4] M. G. Manhas, "JustAI," 28 October 2023. [Online]. Available: <https://justai.in/shakey-the-worlds-first-ai-bot/>.
- [5] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Sojourner_\(rover\)](https://en.wikipedia.org/wiki/Sojourner_(rover)).
- [6] P. L. S. Q. H. Q. J. M. M. L. Y. H. a. E. M. Liwei, "Path Planning Technique for Mobile Robot: A Review," 2023.
- [7] Lixing Liu , Xu Wang , Xin Yang, Hongjie Liu, Jianping Li and Pengfei Wang, "Path planning techniques for mobile robots: Review and prospect," p. 1, 2023.
- [8] "Siêu thị điện máy - Nội thất," [Online]. Available: <https://dienmaycholon.com/kinh-nghiem-mua-sam/cach-luu-ban-do-robot-hut-bui-xiaomi-cuc-don-gian-ban-khong-nen-bo-qua>.
- [9] A. H. A. E. S. M. M. T. S. A. A. S. a. N. A. R. Marni Azira Markom, "A Mapping Mobile Robot using RP Lidar Scanner," 2015.
- [10] S. R. M. T. T. Y. G. K. M. M. A. G. a. D. Dominik Nuss, "A random finite set approach for dynamic occupancy grid map with real-time application," *The International Journal of Robotic Research* , 2018.
- [11] A. N. M. Uysal, "DEM generation with UAV Photogrammetry and analysis in Sahitler hill," *Measurement*, vol. 73, pp. 539-543.
- [12] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," pp. 46-57, 1989.

- [13] R. C. S. a. P. Cheeseman, "On the representation and estimation of spatial uncertainty," vol. 5, pp. 56-68, 12/1986.
- [14] P. N. S. C. H. F. D.-W. a. M. C. M. W. M. G. Dissanayake, "A solution to the simultaneous localization and map building (SLAM) problem," vol. 17, pp. 229-241, 2001.
- [15] R. K. C. S. a. W. B. G. Grisetti, "A tutorial on graph-based SLAM," vol. 2, pp. 31-43, 2010.
- [16] J. M. M. M. a. J. D. T. R. Mur-Artal, "ORB-SLAM: A versatile and accurate monocular SLAM system," vol. 31, 2015.
- [17] B. U. a. T. B. H. Zhou, "DeepTAM: Deep tracking and mapping with convolutional neural networks," 2020.
- [18] X. L. a. Y. W. J. Zhang, "Deep learning-based SLAM for dynamic environments," 2021.
- [19] P. J. L. M. L.E. Kavraki, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," 1996.
- [20] H. L. X. F. W. L. Yinjing Guo, "Research Progress of Path Planning Methods for Autonomous Underwater Vehicle," 2021.
- [21] J.-C. Latombe, Robot Motion Planning, 1991.
- [22] D. W. .. Fox, "The Dynamic Window Approach to Collision Avoidance," 1997.
- [23] N. S. C. D. a. J. E. S. Karthik Karur, "A Survey of Path Planning Algorithms for Mobile Robots," 2021.
- [24] T. H. Krzysztof R Apt, Edsger Wybe Dijkstra: His Life, Work, and Legacy, ACM Books.
- [25] duongthao.1610, "VIBLO," 29 8 2019. [Online]. Available: <https://viblo.asia/p/bai-toan-tim-duong-di-ngan-nhat-voi-giai-thuat-dijkstra-eW65GRxLlDO>.
- [26] A. Stentz, "Optimal and efficient path planning for partially-known environments," *IEEE*, 1994.
- [27] S. S. a. A. S. A. Yahja, "An efficient sampling-based approach to D* planning," *IEEE Trans. Robot*, vol. 26, pp. 512-526, 2010.

- [28] L. L. a. Y. W. J. Zhang, "Deep-enhanced D* Lite for dynamic environment navigation," 2020.
- [29] J. H. Holland, "Genetic Algorithms and the Optimal Allocation of Trials," 1973.
- [30] F. Digital, "FPT Digital," 25 9 2024. [Online]. Available: <https://digital.fpt.com/chuyen-doi-so/tri-tue-nhan-tao-ai-vi/may-hoc-la-gi-machine-learning-la-gi-cac-dang-may-hoc-va-khai-niem-lien-quan-den-machine-learning.html>.
- [31] N. M. Thái, "VNPT AI," 03 02 2025. [Online]. Available: <https://vnptai.io/vi/blog/detail/machine-learning-la-gi>.
- [32] R. G. a. V. J. Carey, "Unsupervised Machine Learning," in *Bioconductor Case Studies*, Springer, pp. 137-157.
- [33] P. T. đổi, "Pi.Exchange," 4 12 2023. [Online]. Available: <https://www.pi.exchange/blog/clustering-in-machine-learning>.
- [34] R. S. S. a. A. G. Barto, in *Reinforcement Learning: An Introduction*, Cambridge University Press, 2018.
- [35] P. P. S. a. D. S. Shah, "A Review of Machine Learning and Deep," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBE)*, 2018.
- [36] "elcom," elcom, 24 10 2022. [Online]. Available: <https://www.elcom.com.vn/deep-learning-la-gi-ung-dung-cua-hoc-sau-trong-thuc-tien-1666582364>.
- [37] I. S. a. G. E. H. A. Krizhevsky, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems* 25, 2012.
- [38] phamdinhkhanh, "DS-boosting your data exploration," 22 8 2019. [Online]. Available: <https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html>.
- [39] Y. Kim, "Convolutional Neural Networks for Sentence Classification," 2014.
- [40] "Amazon Web Services (AWS)," [Online]. Available: <https://aws.amazon.com/vi/what-is/recurrent-neural-network/>.

- [41] S. Maity, "Linkedin," 21 March 2024. [Online]. Available: <https://www.linkedin.com/pulse/rnn-lstm-gru-why-do-we-need-them-suvankar-maity-joege/>.
- [42] P. F. a. T. B. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015.
- [43] E. S. T. D. Jonathan Long, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431-3440, 2015.
- [44] N. Klingler, "viso.ai," 23 April 2024. [Online]. Available: <https://viso.ai/deep-learning/u-net-a-comprehensive-guide-to-its-architecture-and-applications/>.
- [45] G. G. P. D. a. R. G. Kaiming He, "Mask R-CNN," 2018.
- [46] N. V. Q. L. K. H. P. N. M. H. L. P. Đ. K. N. L. Q. N. P. H. V. N. V. C. Đoàn Võ Duy Thanh, "Đắm mình vào học sâu," [Online]. Available: https://d2l.aivivn.com/chapter_computer-vision/rcnn_vn.html.
- [47] T. V. Huy, "Tran Van Huy Artificial Intellegence," 24 July 2021. [Online]. Available: <https://huytranvan2010.github.io/Feature-Pyramid-Network/>.
- [48] P. D. R. G. K. H. B. H. a. S. B. Tsung-Yi Lin, "Feature Pyramid Networks for Object Detection," 2017.
- [49] phamdinhkhanh, "DS boosting your data exploration," 18 June 2020. [Online]. Available: <https://phamdinhkhanh.github.io/2020/06/18/DeepLab.html>.
- [50] [Online]. Available: <http://liangchiehchen.com/projects/DeepLab.html>.
- [51] L. P. G. K. I. M. K. & Y. A. L. Chen, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 834–848, 2017.
- [52] Y. L. D. S. B. W. Gang Li, "Automatic pixel-level bridge crack detection using learning context flux field with convolutional feature fusion," *Journal of Civil Structural Health Monitoring*, pp. 1-17, 2024.

- [53] L. Z. Y. P. G. S. F. & A. H. Chen, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *Lecture notes in computer science*, pp. 833-851, 2018.
- [54] facebookresearch, "paperswithcode," 19 February 2021. [Online]. Available: <https://paperswithcode.com/lib/detectron2/deeplabv3-1#>.
- [55] M. A. Z. a. M. Y. Yuexia Tang, "Path Planning Trends for Autonomous Mobile Robot Navigation: A review," *MDPI*, 2025.
- [56] "Geeksforgeeks," Geeksforgeeks, 30 July 2024. [Online]. Available: <https://www.geeksforgeeks.org/a-search-algorithm/>.
- [57] H. Durrant-Whyte và T. Bailey, Simultaneous localization and mapping (SLAM), 2006, pp. 99-110.
- [58] Han-ye Zhang, Wei-ming Lin and Ai-xia Chen, "Path Planning for the Mobile Robot: A Review," 2018.
- [59] Christos Alexopoulos and Paul M. Griffin, "Path Planning for a Mobile Robot," 1992, pp. 318-322.
- [60] M. Goodchild, "Citizens as sensors: The world of volunteered geography," *GeoJournal*, 2007.
- [61] Wikipedia, "Wikipedia," 27 May 2025. [Online]. Available: https://en.wikipedia.org/wiki/A*_search_algorithm.
- [62] R. Ghinire, "Medium," 10 February 2023. [Online]. Available: <https://medium.com/@r1j1nghimire/semantic-segmentation-using-deeplabv3-from-scratch-b1ff57a27be>.
- [63] P. Đ. Khánh, "DS - Boosting your data exploration," 10 June 2020. [Online]. Available: <https://phamdinhhkhanh.github.io/2020/06/10/ImageSegmentation.html#12-b%C3%A0i-to%C3%A1n-image-segmentation>.
- [64] Viethoangeng, "VIBLO," 16 April 2020. [Online]. Available: <https://viblo.asia/p/maskrcnn-cac-buoc-trien-khai-mask-r-cnn-cho-bai-toan-image-segmentation-6J3Zg4VPlmB>.

