

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
BỘ MÔN KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

KHOÁ LUẬN TỐT NGHIỆP
CỦY NHÂN CÔNG NGHỆ THÔNG TIN

**NGHIÊN CỨU ỨNG DỤNG DEEP LEARNING VÀO
BÀI TOÁN ĐẾM ĐỐI TƯỢNG TRONG ẢNH TĨNH**



GIẢNG VIÊN HƯỚNG DẪN:
TS. NGÔ ĐỨC THÀNH

SINH VIÊN THỰC HIỆN:
NGUYỄN TRUNG ĐỨC - 13520211

04 - 2017

Xin dành tặng quyển luận văn này cho ...

LỜI CÁM ƠN

Tôi xin chân thành cảm ơn ...

TÓM TẮT

Trong những năm gần đây, ngành công nghệ thông tin đang phát triển rất mạnh mẽ và nhanh chóng. Tiêu biểu là AI - Artificial Intelligence (Trí tuệ nhân tạo), và cụ thể hơn nữa là Machine Learning (Máy Học) nổi lên như một bằng chứng rõ rệt của một cuộc cách mạng công nghiệp lần thứ tư sắp xảy ra (1 - động cơ hơi nước, 2 - điện năng, 3 - công nghệ thông tin). Trí tuệ nhân tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể rất nhiều người trong chúng ta không nhận ra. Hệ thống nhận diện tự tag khuôn mặt của Facebook ¹, hệ thống gợi ý sản phẩm của Amazon ², xe tự hành của Google và Tesla, trợ lý ảo Siri của Apple³, hệ thống gợi ý phim của Netflix ⁴, và mới đây, với một sự kiện chấn động, tháng 5 năm 2016 máy chơi cờ vây AlphaGo ⁵ của Google DeepMind đã chiến thắng cả con người (nhà vô địch cờ vây thế giới Lee Se-dol) trong một trò chơi mà số khả năng biến hóa có thể xảy ra là 10^{761} nhiều hơn tất cả số lượng nguyên tử trong vũ trụ

Khi mà công nghệ ngày càng phát triển, khả năng tính toán của các máy tính được nâng lên một tầm cao mới cộng với một lượng dữ liệu khổng lồ được các hãng công nghệ lớn thu thập. Machine

¹<https://www.facebook.com/notes/facebook/making-photo-tagging-easier/467145887130/>

²<https://www.amazon.com/>

³<https://www.apple.com/ios/siri/>

⁴<https://piay.netflix.com/>

⁵<https://deepmind.com/research/alphago/>

Learning đã tiến thêm một bước tiến dài và một lĩnh vực mới ra đời có tên là Deep learning (Học Sâu). Deep Learnig đã giúp cho máy tính thực thi được những công việc tưởng chừng như không thể vào khoảng 10 năm trước đây như: phân loại đối tượng khác nhau trong một bức ảnh, Giao tiếp với con người, hay thậm chí là sáng tác âm nhạc, văn thơ.

Nhận thấy tính ứng dụng cao, sự hiệu quả và tiềm năng phát triển của Deep Learning sau này. Sinh viên chọn đề tài "Nghiên cứu áp dụng Deep Learning cho bài toán đếm đối tượng trong ảnh tĩnh" với tính ứng dụng cao và có thể áp dụng cho nhiều lĩnh vực khác nhau trong thực tế. Qua đề tài này, sinh viên muốn tìm hiểu về Deep Learning và áp dụng kiến thức đã tìm hiểu được để xây dựng một phương pháp đếm các đối tượng (con người, phương tiện) tự động từ hình ảnh của người dùng cung cấp.

Mục lục

Mục lục	v
Danh sách hình vẽ	vi
Nomenclature	vi
1 TỔNG QUAN	2
1.1 Đặt vấn đề	2
1.2 Đối tượng, phạm vi và mục tiêu	5
1.2.1 Đối tượng	5
1.2.2 Phạm vi	5
1.2.3 Mục tiêu	6
1.3 Cấu trúc khóa luận	7
2 CÁC NGHIÊN CỨU LIÊN QUAN	8
2.1 Đếm dựa trên phát hiện đối tượng	8
2.2 Đếm dựa trên gom cụm	11
2.3 Đếm dựa trên mô hình hồi quy	13
2.4 Kết chương	14
3 DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG TRONG ẢNH TĨNH.	16

MỤC LỤC

3.1	Tổng quan về Deep Learning.	17
3.1.1	Neural networks	17
3.1.1.1	Mạng neural nhân tạo.	17
3.1.1.2	Hoạt động của mạng neural nhân tạo. . . .	23
3.1.2	Tổng quan deep learning	26
3.1.3	Convolutional Neural Networks	32
3.1.3.1	Lớp Convolution	34
3.1.3.2	Lớp Pooling	38
3.1.3.3	Lớp Normalization	40
3.2	Áp dụng Deep Learning cho bài toán đếm đối tượng	41
3.2.1	Mô hình đếm đối tượng	41
3.2.2	Mô hình CCNN	43
3.2.3	Mô hình Hydra CNN	44
3.3	Kết chương	44
4	THỰC NGHIỆM VÀ ĐÁNH GIÁ	45
4.1	Bộ dữ liệu thực nghiệm.	45
4.1.1	Bộ Dữ liệu UCSD	46
4.1.2	Bộ dữ liệu <i>UCF_CC_50</i>	46
4.1.3	Bộ dữ liệu TRANCOS	47
4.2	Phương pháp đánh giá thực nghiệm.	48
4.2.1	Đánh giá thực nghiệm trên bộ UCSD	48
4.2.2	Đánh giá trên bộ dữ liệu UCF	50
4.2.3	Đánh giá trên bộ dữ liệu TRANCOS	50
4.3	Kết quả đánh giá	51
4.3.1	Kết quả đánh giá trên bộ dữ liệu UCSD	51
4.3.2	Kết quả đánh giá trên bộ dữ liệu UCF	51
4.3.3	Kết quả đánh giá trên bộ dữ liệu TRANCOS	51

MỤC LỤC

5 ỨNG DỤNG THỰC NGHIỆM VÀ HƯỚNG PHÁT TRIỂN. 52

Phụ lục A	53
Phụ lục B	54
Tài liệu tham khảo	55

Danh sách hình vẽ

1.1	Dữ liệu hình giám sát đám đông (nguồn: Internet)	3
1.2	(a) là hình ảnh trong cuộc hành hương ở Sabarimala và (b) là hình ảnh trong lễ hội âm nhạc Love Parade	4
2.1	phát hiện dựa trên toàn bộ đối tượng.	10
2.2	phát hiện dựa trên các bộ phận của đối tượng (đầu và vai). .	11
2.3	(a) và (b) minh họa kết quả dựa trên việc gom nhóm các chuyển động liên kết với nhau. (c)	12
2.4	mô hình áp dụng regression: đầu tiên ta xác định khu vực quan tâm cần tính toán và tìm ra sơ đồ chuẩn hóa (normal- isation map) . sau đó sẽ rút trích những đặc trưng tổng thể và huấn luyện ra mô hình hồi quy sử dụng những đặc trưng đã được chuẩn hóa.	15
3.1	Các thành phần cấu trúc của một neural nhân tạo ¹	18
3.2	mô hình kết nối và quá trình truyền tín hiệu giữa hai neural sinh học ²	19
3.3	Cấu trúc của một neural nhân tạo.	21
3.4	Cấu trúc mô hình ANN 1 lớp.	23
3.5	Cấu trúc mạng nhiều tầng truyền thẳng.	24

¹<https://owlcation.com/stem/Structure-of-a-Neuron>

²<https://en.wikipedia.org/wiki/Neuron>

DANH SÁCH HÌNH VẼ

3.6 Các đặc trưng trên khuôn mặt từ tập dữ liệu sẽ được học qua quá trình huấn luyện. các lớp sau có thể tổng hợp và từ dữ liệu đầu ra của lớp trước đó cho ra những đặc trưng tổng quát và phức tạp hơn.	28
3.7 Minh họa mô hình Deep Neural Networks với ba lớp ẩn, một lớp đầu vào và một lớp đầu ra với cùng kiểu liên kết fully-connected.	30
3.8 Minh họa mô hình CNN sử dụng cho bài toán nhận dạng chữ viết tay.	31
3.9 Minh họa kiến trúc mô hình Deep Belief Networks (DBNs). .	32
3.10 Caption for LOF	33
3.11 Caption for LOF	34
3.12 Caption for LOF	35
3.13 Caption for LOF	37
3.14 Caption for LOF	38
3.15 Max-pooling lấy giá trị lớn nhất trên vùng $[2 \times 2]$ trên kích thước $[4 \times 4]$ và thu được đầu ra kích thước $[2 \times 2]$	39
3.16 Minh họa áp dụng Pooling từ lớp Conv	40
3.17 Mô hình giải quyết bài toán đếm đối tượng với đầu vào là một bức ảnh và đầu ra là một bản đồ mật độ đối tượng trong bức ảnh đó.	43
4.1 một vài hình ảnh trong bộ dữ liệu UCF	47
4.2 Mô hình Pyramid với nhiều tỉ lệ trên một patch.	49
4.3 Mô số ví dụ trong bộ dữ liệu test sau khi chạy thực nghiệm với kết quả thể hiện gồm ground truth, kết quả dự đoán và giá trị sai lệch.	50

DANH SÁCH HÌNH VẼ

Chương 1

TỔNG QUAN

1.1 Đặt vấn đề

Ngày nay, công nghệ phát triển không ngừng và ngày càng ảnh hưởng sâu rộng vào đời sống xã hội, dần dần trở thành một phần không thể thiếu trong xã hội loài người. Trong sự phát triển đó thì công nghệ thông tin đóng một vai trò chủ chốt không thể thiếu. Với sự những tiến bộ vượt bậc trong lĩnh vực trí tuệ nhân tạo (Artificial Intelligence) mà điển hình và thị giác máy tính (computer vision). Những công nghệ này đã và đang giúp con người giải quyết được những bài toán thực tế trong đời sống và làm cho xã hội ngày càng văn minh hiện đại hơn.

Với sự phát triển nhanh chóng của khoa học kỹ thuật và công nghệ, hệ thống thiết bị thu phát hình ảnh như camera giám sát ngày càng hiện đại và phô biến với chất lượng ngày càng cao cùng chi phí triển khai ở mức thấp. Các hệ thống này thường được sử dụng để theo dõi và đánh giá tình hình anh ninh trật tự ở các địa điểm công cộng (công viên, trường học, quảng trường, nhà ga, sân bay,...), các cơ quan trọng yếu (tòa nhà, cơ quan công quyền, các khu vực an ninh quốc phòng). Hay những hệ thống giám sát giao thông theo giới tình hình tắc nghẽn tại các điểm nóng giao thông

1. TỔNG QUAN

hay tình hình phương tiện giao thông thời gian cao điểm. Với đối tượng giám sát là con người, một trong những hành vi của con người cần được ưu tiên giám sát đó là hành vi tụ tập đông người (đám đông).



Hình 1.1: Dữ liệu hình giám sát đám đông (nguồn: Internet)

Vào ngày 24 tháng 7 năm 2010, lễ hội âm nhạc Love Parade¹ được tổ chức tại Đức ở một ga vận tải cũ với sức chứa tối đa là 250,000 người, nhưng thực tế đã có hơn nửa triệu khác du lịch đến tham gia lễ hội, dẫn đến xảy ra một vụ thảm họa về đám đông. Vụ việc này đã cướp đi sinh mạng của 21 người và ít nhất 500 người bị thương. Một vụ tương tự cũng xảy ra tại Ấn Độ, vào ngày 14 tháng 1 năm 2011, là ngày Makara Jyothi². Một cuộc hành hương thường niên hàng năm diễn ra ở gần đền thờ Sabarimala cũng đã cướp đi sinh mạng của 106 người hành hương và làm bị thương gần 100 người. Những sự việc trên có thể sẽ không phải xảy ra nếu chúng ta có một hệ thống giám sát và ước lượng đượng mật độ đám đông, từ đó đưa ra những cảnh báo về khả năng xảy ra những thảm họa như trên.

¹https://en.wikipedia.org/wiki/Love_Parade_disaster

²https://en.wikipedia.org/wiki/Love_Parade_disaster

1. TỔNG QUAN



Hình 1.2: (a) là hình ảnh trong cuộc hành hương ở Sabarimala và (b) là hình ảnh trong lễ hội âm nhạc Love Parade

Không những thế việc có thể ước lượng hay tính toán được mật độ của đám đông có thể áp dụng trong rất nhiều bối cảnh như những cuộc biểu tình, những sự kiện chính trị, những sân vận động bóng đá, thông tin cho lĩnh vực quảng cáo, hay cảnh báo và đưa ra chiến lược giải quyết các vấn đề ùn tắc với đối tượng là các phương tiện giao thông.

Với những ứng dụng thiết thực và kể trên. Nhưng trong thực tế, cho đến nay việc giám sát đám đông qua các thiết bị thu phát hình ảnh vẫn chủ yếu dựa vào con người. Tuy nhiên, việc duy trì sự tập trung của con người cho việc giám sát một cách liên tục dễ gây quá tải cho người giám sát dẫn đến những sai sót có thể xảy ra, bên cạnh đó công việc này còn đòi hỏi chi phí vận hành, quản lý cao và khó giám sát. Từ đó, việc xây dựng các hệ thống phát hiện và ước lượng đám đông hay phương tiện giao thông là một nhu cầu cấp thiết hiện nay. Hệ thống này có thể hỗ trợ con người giám sát hiệu quả những hoạt động có liên quan đến tụ tập đám đông hay những điểm nóng giao thông đang gây ùn tắc, để từ đó đưa ra những cảnh báo hay chiến lược giải quyết phù hợp và hiệu quả.

Với tính ứng dụng cao và thiết thực, nhưng đổi lại đây là một bài toán

1. TỔNG QUAN

vẫn còn tồn tại rất nhiều thách thức và khó khăn cần phải giải quyết và cải thiện như: sự che khuất lẫn nhau giữa các đối tượng, kích thước đa dạng của các đối tượng (tùy thuộc vào độ sâu của ảnh), sự biến dạng đối tượng từ góc nhìn, quá ít pixel để biểu diễn cho đối tượng... vv. Qua quá trình tìm hiểu và khảo sát cho bài toán đếm đối tượng, sinh viên nhận thấy hướng tiếp cận giải quyết bài toán theo phương pháp regression đã cho kết quả với tốc độ và độ chính xác cao nhất hiện nay. Kèm theo đó, Deep Learning trong thời gian gần đây đã phát triển rất nhanh chóng và được áp dụng hiệu quả trong nhiều lĩnh vực. Đặc biệt là về Computer Vision, Deep Learning đã giải quyết rất tốt với độ chính xác cao các vấn đề trong lĩnh vực này (bài toán phân lớp ảnh áp dụng deep learning cho độ chính xác thậm chí còn tốt hơn con người). Chính vì những lý do trên, sinh viên chọn thực hiện khóa luận này với đề tài "nghiên cứu ứng dụng deep learning vào bài toán đếm đối tượng trong ảnh tĩnh".

1.2 Đối tượng, phạm vi và mục tiêu

1.2.1 Đối tượng

Đối tượng nghiên cứu là các phương pháp áp dụng deep learning để ước lượng mật độ đám đông (con người hoặc phương tiện giao thông) trong các hình ảnh tĩnh.

1.2.2 Phạm vi

- Khóa luận tập trung tìm hiểu Deep Learning mà cụ thể là mô hình Convolutional Neural Networks qua đó áp dụng để giải quyết bài toán đếm đối tượng.
- Thực hiện huấn luyện và đánh giá trên ba bộ dữ liệu:

1. TỔNG QUAN

- TRANCOS³ bộ dữ liệu được thu thập từ các video giám sát giao thông ở thành phố Dirección General de Tráfico Tây Ba Nha⁴.
 - UCSD⁵ bộ dữ liệu này thu thập từ dữ liệu video được lấy từ camera giám sát, chưa cảnh người đi bộ trên một con đường ở UCSD (University of California, San Diego⁶).
 - UCF¹ là bộ dữ liệu chứa các hình ảnh đám đông với mật độ cực lớn. Các hình ảnh này được thu thập từ FLICKR².
- xây dựng ứng dụng đánh giá kết quả thực nghiệm và ước lượng đối tượng (người và phương tiện giao thông) trong ảnh tĩnh.

1.2.3 Mục tiêu

- Tập trung tìm hiểu về Deep Learning và mô hình Convolutional Neural Networks.
- Khảo sát về bài toán đếm đối tượng, các chiến lược được đề xuất giải quyết bài toán, tìm hiểu các kiến thức liên quan. Từ đó chọn chiến lược hiệu quả nhất để áp dụng và giải quyết bài toán đếm đối tượng.
- So sánh đánh giá giải pháp xây dựng được với những công trình đã được công bố.
- Xây dựng công cụ đáng giá mô hình và ứng dụng ước lượng đám đông và phương tiện giao thông.

³<http://agamenon.tsc.uah.es/Personales/rlopez/data/trancos/>

⁴<http://www.dgt.es/es/>

⁵<http://www.svcl.ucsd.edu/projects/peoplecnt/>

⁶<https://ucsd.edu/>

¹<http://crcv.ucf.edu/data/crowdcounting.php>

²là một trang web lưu trữ dữ liệu hình ảnh và video được xây dựng bởi Ludicorp trong năm 2004 và sau đó Yahoo mua lại vào 20 tháng 3 năm 2015

1.3 Cấu trúc khóa luận

Chương 1: Giới thiệu tổng quan đề tài.

Chương 2: Trình bày tổng quát các hướng tiếp cận đã được đề xuất nhằm giải quyết bài toán hướng đối tượng đã được đưa ra.

Chương 3: Trình bày các kiến thức cơ sở về Neural Networks, Deep Learning, Convolutional Neural Network và từ đó trình bày mô hình Deep Learning áp dụng cho bài toán.

Chương 4: Trình bày phương pháp đánh giá mô hình trên các bộ dữ liệu và đưa ra so sánh với các phương pháp khác. kèm theo đó trình bày ứng dụng xây dựng công cụ đánh giá và áp dụng mô hình để ước lượng đối tượng.

Chương 5: Trình bày ứng dụng thực nghiệm, kết luận và hướng phát triển của đề tài.

Chương 2

CÁC NGHIÊN CỨU LIÊN QUAN

Từ những vấn đề đặt ra ở trên, Ta có thể thấy được tính ứng dụng và ý nghĩa thực tiễn của bài toán đếm đối tượng (ví dụ như con người). Những chiến lược đã được các nhà nghiên cứu đưa ra để giải quyết bao gồm Counting by Detection, Counting by Clustering và gần đây nhất thì phương pháp Counting by Regression đã đưa đến những kết quả đáng chú ý trong xử lý với các đám đông có mật độ lớn và phức tạp. Trong chương này, sinh viên dựa trên một công trình khảo sát thực hiện bởi Loy et al. [1] và quá trình khảo sát của bản thân sẽ trình bày tổng quát và hệ thống hóa các phương pháp giải quyết bài toán đếm đối tượng theo ba chiến lượng nêu trên. Đặc biệt sẽ tập trung vào chiến lượng Counting by Regression, phương pháp cho thấy sự hiệu quả trong đa dạng các môi trường khác nhau.

2.1 Đếm dựa trên phát hiện đối tượng

Với vấn đề **đếm** thì hướng giải quyết dễ thấy đầu tiên và đơn giản nhất đó là phát hiện ra đối tượng và đếm nó. Mục này sẽ trình bày ngắn gọn về các phương pháp trong chiến lượng đếm dựa trên phát hiện đối tượng

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.

(Counting by Detection).

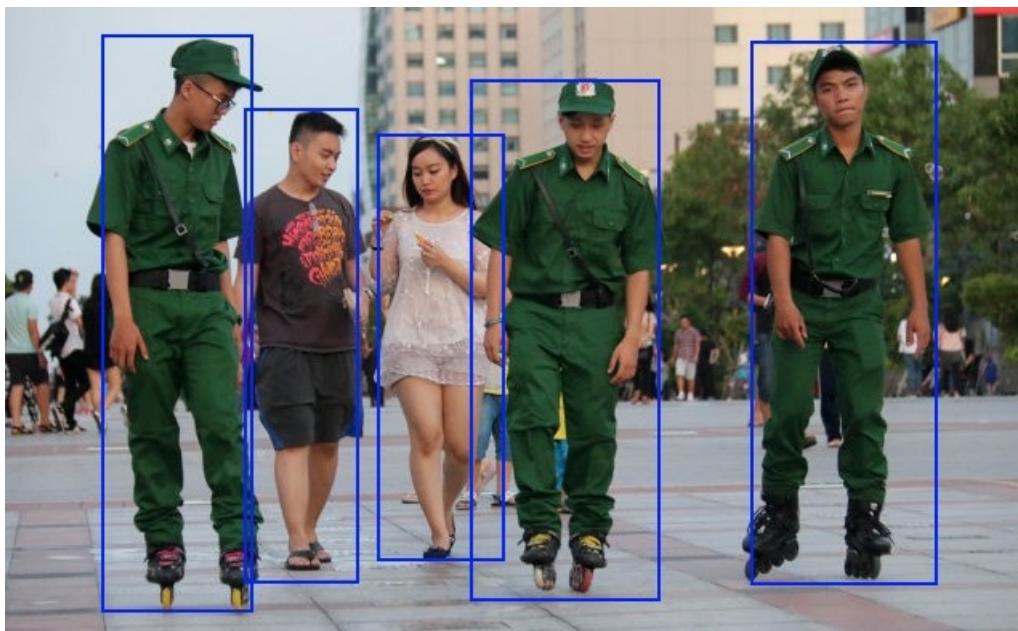
Phát hiện toàn bộ (Monolithic detection): đây là hướng tiếp cận trực quan và trực tiếp nhất đối với bài toán đếm, liệt kê số đối tượng (con người) thông qua phát hiện từng đối tượng. Quá huấn luyện cho phương pháp này sẽ tiến hành phân loại sử dụng toàn bộ các bộ phận của đối tượng trong tập dữ liệu huấn luyện (ví dụ minh họa trong hình 2.1). Những đặc trưng thường được sử dụng trong phương pháp này bao gồm: Haar wavelet¹, đặc trưng HOG(Histogram of gradient)¹, edgelet và shapelets. Việc lựa chọn sử dụng đặc trưng nào để biểu diễn cho đối tượng sẽ ảnh hưởng rất lớn đến độ chính xác và tốc độ của việc phát hiện ra đối tượng. Thông thường, thì việc này sẽ đòi hỏi phải đánh đổi một trong hai, giữa sự chính xác và tốc độ. Cụ thể như phân loại phi tuyến tính như RBF Support Vector Machines (SVMs) là một phương pháp mang lại kết quả với độ chính xác cao, nhưng đổi lại thì tốc độ lạ chậm. Chính vì thế, các phương pháp phân loại tuyến tính như boosting, SVMs tuyến tính hoặc Random/Hough Forests được sử dụng phổ biến hơn.

Quá trình huấn luyện trong giai đoạn phát hiện đối tượng thường áp dụng cửa sổ trượt (sliding window) trên toàn bộ hình ảnh để tìm ra các đối tượng. Công việc này phụ thuộc nhiều và việc chọn kích thước ban đầu cho cửa sổ trượt khi mà kích thước của các đối tượng có sự chênh lệch lớn. Kèm theo đó ta sử dụng một ngưỡng độ tin cậy (thread hold) để có thể loại bỏ những vị trí có độ tin cậy thấp. Cuối cùng ta thu được tổng số lượng của các đối tượng xuất hiện trong bức hình. Việc áp dụng phát hiện dựa toàn bộ đối tượng có thể áp trong trường hợp các đối tượng xuất hiện rời rạc và rõ ràng trong hình ảnh. Tuy nhiên, đối với những trường hợp các đối tượng xuất hiện một cách lộn xộn và bị che khuất thì phương pháp này

¹https://en.wikipedia.org/wiki/Haar_wavelet

¹https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.

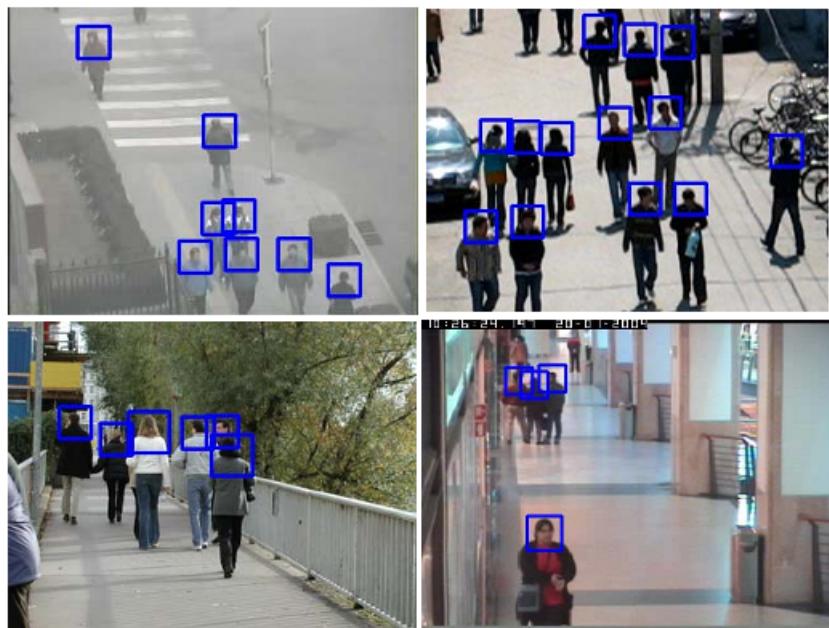


Hình 2.1: phát hiện dựa trên toàn bộ đối tượng.

tỏ ra thiếu hiệu quả và độ chính xác thấp.

Phát hiện bộ phận (Part-based detection): như đã đề cập ở trên thì một thách thức đặt ra cho bài toán đếm đối tượng đó là việc các đối tượng bị che khuất. Để giải quyết được thách thức này, một giải pháp đã được đưa ra đó là thay vì phát hiện toàn bộ đối tượng, ta có thể chỉ cần phát hiện những bộ phận đặc trưng nhất của nó. Cụ thể hơn, đối với con người, ta có thể phân loại đó có phải là người hay không dựa vào những bộ phận đặc trưng như đầu người (ví dụ minh họa trong hình 2.2). Từ đó ước lượng được số lượng người xuất hiện trong hình. Người ta nhận thấy rằng, nếu chỉ sử dụng vùng đầu thì không đủ thông tin và độ tin cậy để có thể phát hiện ra đối tượng (vì có nhiều vật thể có hình dạng tương tự đầu người). Chính vì thế, mô hình phát hiện bao gồm cả đầu và vai có xu hướng mang lại kết quả tốt hơn trong thực tế. Qua đó, có thể nhận thấy so với phương pháp phát hiện toàn bộ đối tượng thì phát hiện dựa vào bộ

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.



Hình 2.2: phát hiện dựa trên các bộ phận của đối tượng (đầu và vai).

phận sẽ giúp giảm nhẹ hơn độ phức tạp của quá trình phát hiện đối tượng và cải thiện độ chính xác trong môi trường phức tạp (có sự che khuất).

2.2 Đếm dựa trên gom cụm

Đếm dựa trên gom cụm (Counting by Clustering) là một hướng tiếp cận dựa trên một giả thiết được đưa ra rằng. Đối với các đối tượng riêng biệt thì thường những đặc tính như hình thức chuyển động hay các đặc điểm bên ngoài có tính tương đồng với nhau. Do đó, khi kết hợp nhằm gom nhóm quỹ đạo của những đặc tính đó với nhau có thể giúp chung ta nhận diện ra các thực thể đó. Cụ thể hơn, các nghiên cứu theo mô hình này bao gồm như [2], nghiên cứu này sử dụng bộ lưu vết Kanade-Lucas-Tomasi (KLT)¹ để thu được một bộ đặc tính từ bộ theo vết ở cấp thấp. Từ đó ta tập hợp

¹https://en.wikipedia.org/wiki/Kanade-Lucas-Tomasi_feature_tracker

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.



Hình 2.3: (a) và (b) minh họa kết quả dựa trên việc gom nhóm các chuyển động liên kết với nhau. (c)

quỹ đạo của các đặc tính đó rồi suy luận ra đối tượng trong bối cảnh cần xét. Cùng với đó, [3] sử dụng các đặc trưng cục bộ, sau đó sử dụng gom nhóm Bayesian¹ để nhóm chúng vào trong các cụm để xử lý. Một phương pháp khác liên quan chặt chẽ với chiến lược này, [4] Đã kết hợp với ý tưởng của các đặc trưng không đổi trong mô hình Counting by Detection ở mục 2.1. Phương pháp này đầu tiên sinh ra một tập các đối tượng người dựa trên giả thuyết phát hiện đầu người. Sau đó quá trình tinh chỉnh tập trên sẽ được lặp đi lặp lại bằng cách tính toán trên các patch nhỏ trong đám đông dựa trên giả thuyết các đặc tính như các trường chuyển động, màu sắc quần áo là không thay đổi.

Các phương pháp [2, 3] đề cập ở trên, nhằm tránh việc học có giám

¹<https://www.diva-portal.org/smash/get/diva2:198852/FULLTEXT01.pdf>

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.

sát (Supervised Learning) hay các mô hình được đề cập trong chiến lược Counting by Detection. Tuy nhiên, mô hình này sử dụng giả định một đối tượng xác định dựa vào sự kết hợp là liên kết giữa các chuyển động. Do đó, trong điều kiện những đối tượng đứng yên hoặc hai hay nhiều đối tượng có chung một quỹ đạo chuyển động theo một đơn vị thời gian thì sẽ dẫn phát sinh ra những dự đoán sai. và phương pháp này chỉ có thể áp dụng trong điều kiện ta phải có được một tập các bức hình liên tục (chẳng hạn như dữ liệu video). Nhưng trong thực tế thì bài toán đếm đối tượng đặt ra trong cả bối cảnh cho những bức ảnh tĩnh, chính vì thế chiến lược này chưa đủ độ bao quát để giải quyết bài toán. Chính vì thế, Chiến lược Counting by regression sẽ được trình bày sau đây đã được ra đời để giải quyết được những vấn đề kể trên.

2.3 Đếm dựa trên mô hình hồi quy

Trong những năm gần đây, mặc dù đã có những chiến lược được đề xuất như phát hiện đối tượng hay theo vết đối tượng đã đạt được những tiến bộ đáng kể, nhưng để có thể áp dụng được nó trong thực tế với những bối cảnh động mật độ giày đặc thì đó vẫn là một vấn đề không nhỏ của 2 chiến lược nêu trên. Đếm dựa trên mô hình hồi quy (Counting by regression) là chiến lược đưa ra nhằm tránh các thách thức nan giải trong các chiến lược liên quan đến phát hiện ra những đối tượng riêng lẻ. Thay vào đó, chiến lược này ước lượng mật độ đám đông dựa trên những mô tả hay đặc trưng tổng quát được rút ra từ toàn bộ mô hình đám đông đối tượng được quan tâm với tổ chức và sự bố trí rất đa dạng. Chính vì không đòi hỏi phải phân biệt rõ ràng hay theo dõi các đối tượng được quan tâm một cách riêng lẻ. Điều đó đã làm cho chiến lược này trở thành một phương pháp khả thi nhất cho đến thời điểm hiện tại cho phép áp dụng khá tốt trong những bối

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.

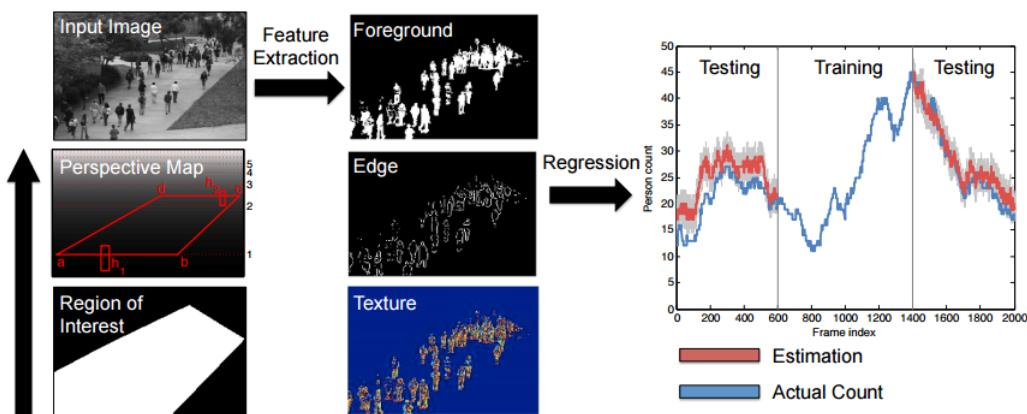
cảnh những đám đông với mật độ lớn và phân bố phức tạp, nơi mà việc phát hiện hay lấy vết các đối tượng riêng lẻ cho thấy việc để áp dụng được vẫn cực kỳ khó khăn.

Một trong những công trình đầu tiên của việc áp dụng phương pháp Regression để ước lượng mật độ của đám đông được đưa ra bởi Davies et al [5]. Đầu tiên, người ta sẽ tiến hành trích xuất những đặc trưng cấp thấp (low-level features) hay các đặc trưng thô như là các pixel ở tiền cảnh (foreground) và các đặc trưng của cạnh từ các khung hình của dữ liệu video. Các thuộc tính tổng quát như diện tích khu vực tiền cảnh (foreground area) và tổng số cạnh sẽ được trích xuất từ tập các đặc trưng thô thu được. Từ đó, một mô hình hồi quy tuyến tính (linear regression) sẽ được sử dụng để liên kết giữa mô hình tổng quát này và hình ảnh đám đông trong thực tế. Cụ thể, với một bức hình hay một frame của dữ liệu video mới là dữ liệu input đưa vào, ta kỳ vọng sẽ thu được mật độ các đối tượng từ những đặc trưng được trích xuất từ những dữ liệu input đó. Từ công trình của Davies et al [5], đã có khá nhiều các phương pháp khác được đề xuất với sự cải tiến về tập các đặc trưng rút ra hoặc với các mô hình hồi quy (regression) tinh vi và phức tạp hơn. Nhưng hầu như tất cả những phương pháp đó đều sử dụng chung một mô hình xử lý như trong [5] (minh họa trong hình 2.4)

Từ những nền tảng đó cùng với những mô hình deep learning mới ra đời. Hướng tiếp cận này đã mang lại những kết quả chính xác hơn và tốc độ cải thiện rõ rệt so với những phương pháp còn lại.

2.4 Kết chương

2. CÁC HƯỚNG TIẾP CẬN VẤN ĐỀ.



Hình 2.4: mô hình áp dụng regression: đầu tiên ta xác định khu vực quan tâm cần tính toán và tìm ra sơ đồ chuẩn hóa (normalisation map) . sau đó sẽ rút trích những đặc trưng tổng thể và huấn luyện ra mô hình hồi quy sử dụng những đặc trưng đã được chuẩn hóa.

Chương 3

DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG TRONG ẢNH TĨNH.

Trong chương này, sinh viên sẽ trình bày tổng quan các kiến thức nền tảng về máy học và Deep Learning (DL). cụ thể hơn trong phần 3.1 sinh viên sẽ trình bày sơ lược về các kiến thức như mô hình Neural Networks, tổng quan về Deep Learning và mô hình đang cực kỳ phát triển mang lại những kết quả rất đáng chú ý là mô hình Convolutional Neural Networks. Từ những kiến thức nền tảng đó, Sinh viên sẽ trình bày chi tiết về kiến trúc mô hình DL được đề xuất bởi Daniel và Roberto [6] cho bài toán đếm đối tượng. Cụ thể đó là mô hình Counting CNN (CCNN) và mô hình cải tiến Hydra CNN. Từ kết quả thực nghiệm có thể nói đây là mô hình State-of-the-art cho bài toán đếm đối tượng.

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

3.1 Tổng quan về Deep Learning.

3.1.1 Neural networks

3.1.1.1 Mạng neural nhân tạo.

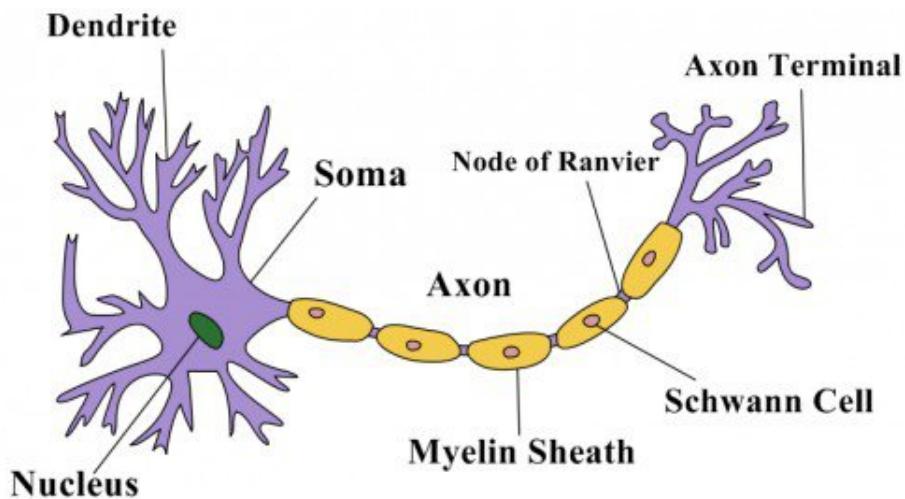
Mạng Neural nhân tạo (Artificial Neural Network: ANN), hay được gọi tắt là Neural Network là một mô hình xử lý thông tin được mô phỏng từ hoạt động của các hệ neural sinh học mà cụ thể hơn là não bộ của con người. Trong đó, thành phần cốt bản của ANN là neural nhân tạo có các cách thức hoạt động và xử lý tương tự như neural sinh học. Do đó, để tìm hiểu về mạng neural nhân tạo, trước tiên ta sẽ tìm hiểu khái quát về mạng neural sinh học.

Cách thức hoạt động của bộ não nói riêng và cả hệ thần kinh nói chung đã được con người quan tâm nghiên cứu từ rất lâu. Nhưng cho đến nay, các nhà khoa học vẫn chưa thực sự hiểu rõ chi tiết về cách thức hoạt động của bộ não và hệ thần kinh con người. Đặc biệt là trong các hoạt động như ghi nhớ, suy nghĩ, học tập, sáng tạo, Tuy nhiên, các nhà khoa học cũng đã thu được một số thông tin cơ bản về bộ não của con người. Theo đó, một bộ não của con người có khối lượng từ 1.2-1.4 kg hoặc khoảng 2% tổng trọng lượng cơ thể với thể tích vào khoảng 1260 cm^3 ở nam và 1130 cm^3 ở nữ¹. Cấu tạo bộ não chia thành nhiều vùng khác nhau và mỗi vùng đảm nhận một hay nhiều hoạt động của cơ thể con người. Quá trình hoạt động của hệ thần kinh bao gồm não bộ và các giác quan như sau: Khi con người tiếp nhận qua các giác quan từ những kích thích từ bên ngoài hoặc bên trong cơ thể. Các kích thích này được chuyển thành xung điện bởi chính các giác quan tiếp nhận nó. Những tín hiệu điện này được đưa về trung ương thần kinh, nơi bộ não sẽ tiến hành xử lý thông tin. Từ những hoạt

¹<https://en.wikipedia.org/wiki/Humanbrain>

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỔI TƯỢNG.

động xử lý, đánh giá, so sánh với các thông tin đã được lưu trữ trước đây. Bộ não sẽ đưa ra quyết định được trả về dưới dạng xung điện. Các tín hiệu điện này sẽ được đưa đến các bộ phận thực thi để thực hiện các hành vi thích hợp như tay, chân, mắt, miệng hay các cơ, ...



Hình 3.1: Các thành phần cấu trúc của một neural nhân tạo².

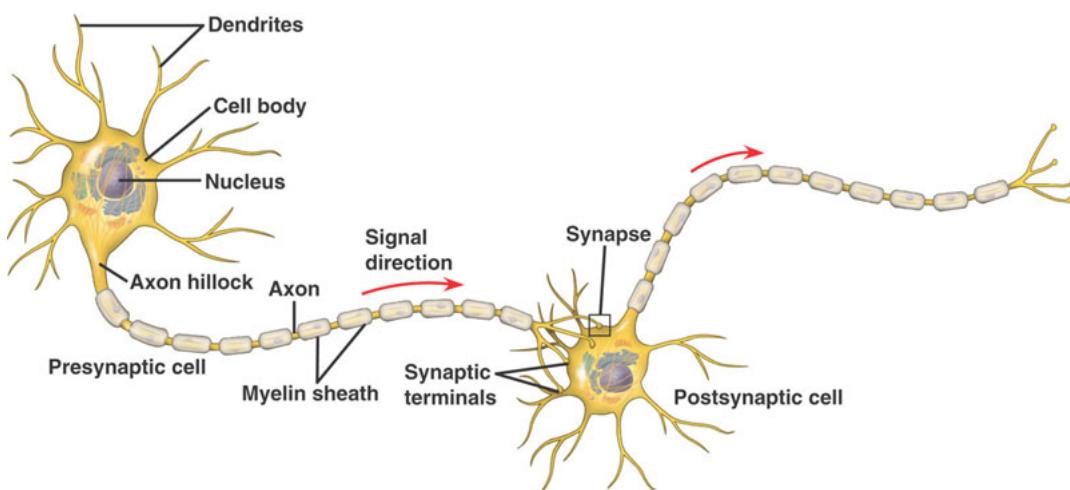
Khi xét ở mức độ tế bào thì bộ não được cấu thành từ 10^{11} phần tử gọi là nơ-ron (hay neural sinh học) và mỗi tế bào có thể kết nối với 10^4 tế bào khác, nghĩa là có gần 10^{15} kết nối giữa chúng³. Ngoài những đặc điểm chung có ở những tế bào khác thì chúng còn có khả năng nhận và xử lý các tín hiệu điện hóa. làm cơ sở cho việc hình thành cho việc xử lý thông tin của não bộ. Mỗi neural sinh học bao gồm 4 thành phần chính:

- Thân neural(cell body) chứa nhân (nucleus) có nhiệm vụ chính là tổng hợp và xử lý các tín hiệu điện hóa từ các đầu vào. Thực chất quá trình này là việc lấy tổng của các tín hiệu mà neural nhận được.
- Các dây dẫn (denrites) là các dây thần kinh truyền tín hiệu đến thân neural.

²<https://en.wikipedia.org/wiki/Neuron>

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

- Sợi trực (axon) có chức năng truyền tín hiệu từ neural sang neural khác. Phần cuối của axon được chia thành những nhánh nhỏ kết thúc tại khớp nối (Synapse)
- Khớp nối (Synapse) là điểm liên kết giữa sợi trực ra của neural này với các nhánh của neural khác. Những liên kết giữa các neural và độ nhạy của mỗi Synapse được xác định bởi quá trình học phức tạp.



Hình 3.2: mô hình kết nối và quá trình truyền tín hiệu giữa hai neural sinh học⁵.

Một cách tổng quát, Neural sinh học hoạt động theo quy trình sau: neural nhận tín hiệu đầu vào từ các denrites. Sau đó nhân neural xử lý các tín hiệu này mà cụ thể là lấy tổng tất cả các tín hiệu chúng nhận được, sau đó phát ra một tín hiệu điện thế. Nếu giá trị điện thế lớn hơn một ngưỡng cho phép nào đó thì sẽ cho ra một tín hiệu đầu ra truyền ra các axon và chính là đầu vào của một neural khác. Dựa trên cấu trúc và cách hoạt động này thì các nhà nghiên cứu đã đề xuất ra mô hình neural nhân tạo hay Neural Networks.

Mạng neural nhân tạo, là một mô hình tính toán được cấu thành từ nhiều phần tử tính toán gọi là neural được liên kết với nhau và cùng hoạt

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỔI TƯỢNG.

động song song. Một neural là một môt đơn vị xử lý thông tin và là thành phần cơ bản của mạng ANN. Cấu trúc của một neural được mô tả trong hình 3.3

Trong đó:

- **Các giá trị đầu vào** Trong thực tế, những dữ liệu này thường được đưa vào dưới dạng một véc tơ N chiều.
- **Tập các liên kết:** Có chức năng truyền thông tin. Trong đó một liên kết ứng với một trọng số (hay trọng số liên kết). Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm ban đầu khởi tạo mạng và chúng được cập nhật liên tục trong quá trình học.
- **Hàm kết hợp:** Thường dùng để tính tổng từ các tham số đầu vào với trọng số liên kết tương ứng của nó (một số tài liệu còn gọi là hàm tổng - summing function).

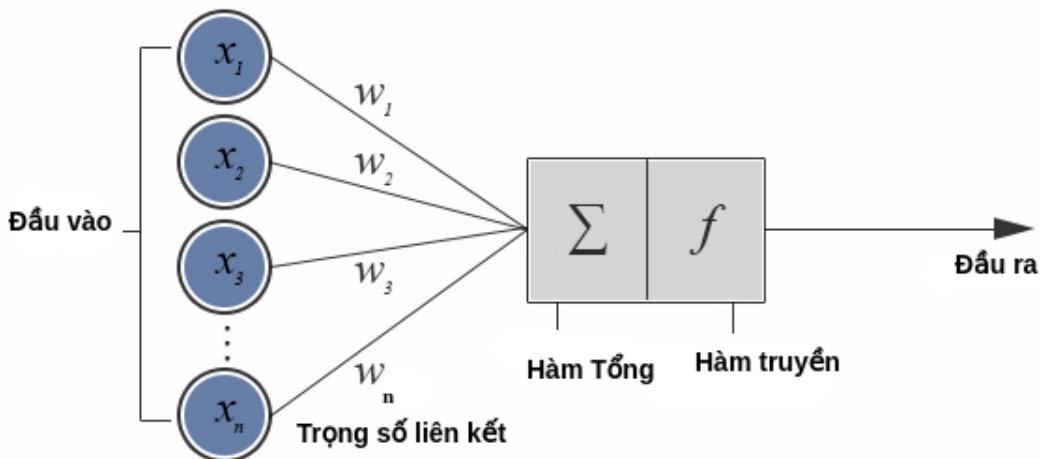
$$a_k = \sum_{i=1}^n W_{ki}x_i + b_k$$

Với W_{ki} là trọng số liên kết giữa các neural đầu vào và b_k là ngưỡng hay độ lệch (bias).

- **Hàm truyền (transfer) hay hàm kích hoạt (active function):** Có chức năng để giới hạn phạm vi đầu ra của mỗi neural. Nó nhận đầu vào là kết quả của hàm kết hợp và ngưỡng đã cho. Thông thường, phạm vi đầu ra của mỗi neural được giới hạn trong đoạn $[0,1]$ hoặc $[-1,1]$. Các hàm truyền rất đa dạng, có thể là hàm truyền tuyến tính hoặc phi tuyến (một số hàm truyền thường hay sử dụng được trình bày trong bảng 3.1). Việc lựa chọn hàm truyền thường phụ thuộc vào từng bài toán và kinh nghiệm của người thiết kế mạng.

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

- **Đầu ra:** là tín hiệu đầu ra sau quá trình tính toán của một neural.



Hình 3.3: Cấu trúc của một neural nhân tạo.

Từ những neural máy tính này được kết hợp với nhau tạo ra một mạng ANN. Tính năng và cách hoạt động của mạng phụ thuộc hoàn toàn vào cấu trúc mạng, trọng số giữa các neural và quá trình xử lý trong mỗi đơn vị neural. Một mạng ANN bao gồm nhiều lớp (layer). Mỗi lớp bao gồm bao gồm nhiều neural có cùng chức năng. Dựa vào Số lớp và sự liên kết giữa các lớp trong mạng mà người ta phân loại ANN thành các nhóm khác nhau.

Phân loại dựa trên số lớp Dựa theo số lớp trong mạng thì ANN được chia thành hai nhóm: mạng một lớp và mạng nhiều lớp.

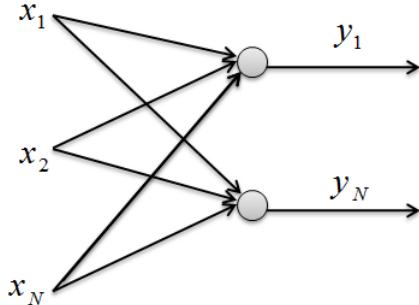
- **Mạng một lớp:** Mạng một tầng tạo từ một lớp neural, nó vừa là tầng đầu vào vừa là tầng đầu ra của mạng.
- **Mạng nhiều lớp:** Đây là một cấu trúc mạng ANN chứa n lớp ($n \geq 2$): trong đó gồm lớp nhận tín hiệu đầu vào (input). Các tín hiệu đầu ra của mạng được cho ra bởi lớp đầu ra của mạng - tầng thứ n. Các lớp nằm ở giữa lớp vào và lớp ra được gọi là lớp ẩn - có ($n-1$) lớp ẩn.

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

Tên Hàm Truyền	Công Thức Toán Học	Đồ thị
Linear	$f(x) = x$	
Tan-sigmoid	$f(x) = \frac{1 - e^x}{1 + e^x}$	
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	
ReLu	$f(x) = \max(0, x)$	

Bảng 3.1: Một số hàm truyền thông thường sử dụng trong mạng neural

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.



Hình 3.4: Cấu trúc mô hình ANN 1 lớp.

Phân loại dựa trên cách liên kết

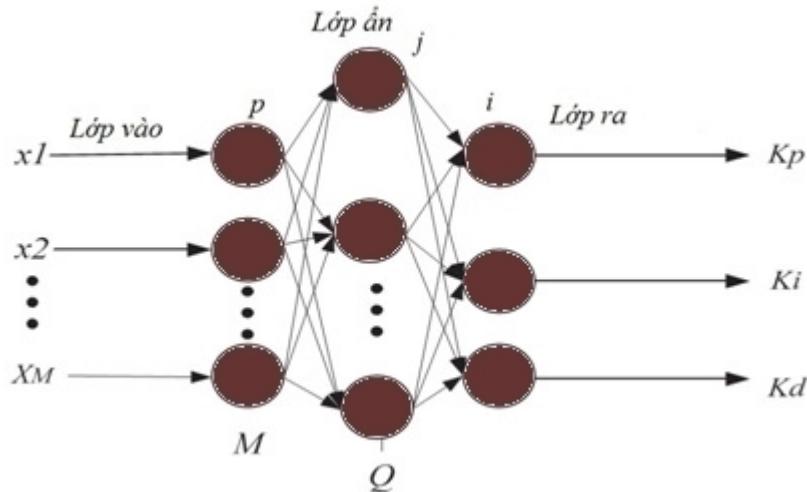
Dựa theo cách liên kết giữa các thành phần trong mạng mà ta có thể chia ANN làm hai nhóm chính:

- **Mạng truyền thẳng** (feedforward neural networks): Đây là một mô hình đơn giản nhất của mạng neural. Các neural trong mạng truyền thẳng liên kết với nhau mà không có liên kết ngược (hay không hình thành nên chu trình). Tín hiệu sẽ truyền từ đầu vào, qua các lớp ẩn và đến đầu ra. Các neural trong cùng một lớp cũng không có liên kết với nhau. ()
- **Mạng hồi quy** (recurrent neural network): Khác với mạng truyền thẳng. Mạng hồi quy chứa các liên kết ngược (hình thành nên chu trình). Mạng có khả năng lưu lại các trạng thái trước đó, và trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào mà còn phụ thuộc vào các trạng thái đầu ra trước đó của nó. ()

3.1.1.2 Hoạt động của mạng neural nhân tạo.

Hoạt động của ANN có thể mô tả một cách cụ thể như sau: tại lớp đầu vào của mạng, các neural sẽ nhận tín hiệu xử lý (tính tổng từ tham số

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.



Hình 3.5: Cấu trúc mạng nhiều tầng truyền thẳng.

đầu vào và truyền kết quả đến hàm truyền) rồi cho ra kết quả (là kết quả cho ra từ hàm kích hoạt). Kết quả này sẽ được truyền tối các neural thuộc tần ẩn thứ nhất, các neural tại đây tiếp nhận các tín hiệu đầu vào rồi xử lý và lại gửi kết quả đến lớp ẩn thứ hai. quá trình này tiếp tục cho đến các neural lớp cuối cùng và cho ra kết quả.

Chức năng của một mạng ANN được quyết định bởi các thành tố như: cấu trúc mạng (số lớp, số neural mỗi lớp và cách các lớp liên kết với nhau) và trọng số của các liên kết bên trong mỗi mạng. Kiến trúc mạng thông thường được dùng cố định cho mỗi bài toán và các trọng số liên kết được tính toán bởi các thuật toán huấn luyện mạng (training algorithm). Quá trình điều chỉnh các trọng số để mạng có thể nhận biết được từ những dữ liệu đầu vào và cho ra kết quả dữ liệu đầu ra như mong muốn gọi là quá trình học (learning) hay huấn luyện (training).

Dễ rõ hơn, giả sử ta có một tập huấn luyện (training set) $(x_i, y_i), \dots, (x_n, y_n)$, với x_i là dữ liệu đầu vào và giá trị mà ta mong muốn thu được sau quá trình tính toán là y_i . Ta gọi y'_i là kết quả thu được thực tế thu được sau

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

quá trình tính toán với đầu và x_i . Khi đó, bản chất của quá trình huấn luyện là thay đổi các trọng số liên kết trong mạng thông qua các bộ huấn luyện sao cho $|y_i - y'_i|$ (sai số) là nhỏ nhất. Một trong những phương pháp huấn luyện ANN là phương pháp lan truyền ngược (Backpropagation).

Backpropagation là thuật toán huấn luyện mạng thường được sử dụng với một phương pháp tối ưu, điển hình là phương pháp Gradient Descent. Phương pháp Gradient Descent được phát triển dựa vào hướng tăng lớn nhất (gradient) tại điểm x thuộc hàm $f(x)$. Gradient tại x của hàm $f(x)$ được tính bởi đạo hàm theo công thức: $\frac{\partial f}{\partial x}$. Gradient tại x sẽ cho biết hướng làm cho hàm $f(x)$ sẽ đạt cực đại. Nếu đi ngược lại hướng đó, ta sẽ thu được điểm cực tiểu của hàm. Từ ý tưởng đó, thuật toán Gradient Descent ra đời.

Thuật toán lan truyền ngược sẽ điều chỉnh các trọng số sao cho thu được bộ giá trị tối ưu nhất (đạt sai số thấp nhất). Lúc ban đầu, ANN sẽ được khởi tạo tập trọng số và tham số học tập α (learning rate). Tập trọng số sẽ được gán ngẫu nhiên và thường là những giá trị nhỏ (trong khoảng $[-1, 1]$). Learning rate là tham số để kiểm soát sự thay đổi của các tham số trong mạng và giúp hàm đạt độ lỗi cực tiểu sau quá trình huấn luyện. Nếu giá trị của α nhỏ, thì giá trị thay đổi trọng số ít và quá trình huấn luyện lâu. Nếu giá trị lớn thì sẽ gây ra hiện tượng sai số vì trọng số thay đổi ở mỗi bước quá nhiều. Do vậy, hiện nay chưa có một kỹ thuật nào có thể tối ưu được việc chọn tham số học tập mà chủ yếu dựa vào thực nghiệm.

Quá trình học mạng bao gồm hai giai đoạn chính được lặp đi lặp lại gồm: **lan truyền** và **Cập nhật trọng số**.

- **Lan truyền:** trong giai đoạn được chia làm hai quá trình.

- Từ bộ dữ liệu đầu vào, chúng đi qua hàm kích hoạt và bộ trọng số được khởi tạo ban đầu để xác định giá trị cho các lớp neural

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỔI TƯỢNG.

tiếp theo. Cho đến khi thu được giá trị đầu tiên của lớp neural cuối cùng.

- Từ giá trị thu được sau cùng, ta tính toán được sai số ở đầu ra và sau đó lan truyền ngược từ lớp đầu ra đến lớp đầu tiên để tính toán gradient tại các lớp ẩn trong mạng.

- **Cập nhật trọng số:** là quá trình điều chỉnh trọng số theo giá trị α và giá trị gradient đã tính toán trong quá trình lan truyền ngược.

Nói tóm lại, việc huấn luyện cho mạng ANN là bài toán xác định các trọng số liên kết trong mạng sao cho sai số đầu ra là nhỏ nhất.

3.1.2 Tổng quan deep learning

Deep Learning (DL) đang là một chủ đề đang thu hút rất nhiều sự quan tâm từ các nhà nghiên cứu cũng như các hãng công nghệ lớn trên thế giới. bởi lý do DL đã thu được những kết quả hết sức nổi bật so với các phương pháp trước đây. Ví dụ như với bài toán phân lớp ảnh trên tập dữ liệu Imagenet⁶, nhờ áp dụng DL mà độ lỗi thu được đã giảm từ 28% năm 2010 xuống còn 4,82% năm 2015. Kết quả này thậm chí còn tốt hơn người với độ lỗi là 5,1%. Không dừng lại ở đó, DL còn đạt được những kết quả ấn tượng trong các cuộc thi như phát hiện đối tượng, nhận diện biển báo giao thông ... và áp dụng trong nhiều lĩnh vực như Xử lý tiếng nói(Spoken Language Processing), xử lý ngôn ngữ tự nhiên(natural language processing - NLP).

Theo Li Deng and Dong Yu và LISA Lab, hiện nay có một vài định nghĩa cho Deep Learning như sau:

- **Định nghĩa 1:** “Deep Learning là một phần của máy học dựa trên việc học qua nhiều cấp độ, tương ứng với hệ thống các đặc trưng hoặc

⁶<http://image-net.org/>

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỔI TƯỢNG.

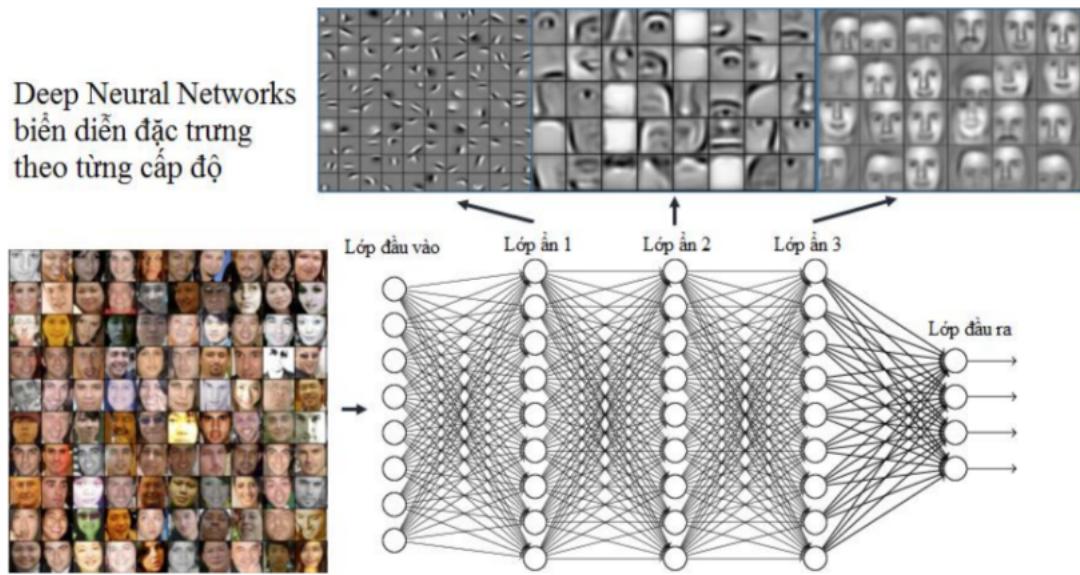
các khái niệm. Những khái niệm ở cấp cao hơn được xây dựng từ khái niệm ở cấp thấp và những khái niệm cùng cấp có thể dùng để định nghĩa những khái niệm cấp cao hơn.”

- **Định nghĩa 2:** “Deep Learning là một nhánh của máy học bao gồm tập các thuật toán nhằm trừu tượng hóa mô hình dữ liệu bậc cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, được tạo nên bởi các mô hình biến đổi phi tuyến.”
- **Định nghĩa 3:** “Deep Learning là tập hợp các kỹ thuật máy học khai thác nhiều lớp xử lý thông tin phi tuyến nhằm biến đổi và trích xuất đặc trưng giám sát hoặc không giám sát, và phục vụ cho phân tích mẫu và phân loại.”
- **Định nghĩa 4:** “Deep Learning là hướng nghiên cứu mới của lĩnh vực Máy Học, được phát triển với mục đích mang Máy Học tiến gần mới mục đích ban đầu của nó: Trí Tuệ Nhân Tạo. Deep Learning trừu tượng hóa nhiều cấp độ của dữ liệu và giúp tìm ra ý nghĩa ẩn chứa trong dữ liệu.”

Từ đó, một cách tổng quát DL là một khái niệm chỉ các thuật toán máy học dựa trên ANN để xây dựng mô hình hay phát hiện ra các quan hệ phức tạp trong dữ liệu bằng những phương pháp học theo nhiều tầng. Kết quả của tầng trước là dữ liệu đầu vào cho tầng kế tiếp, mô hình được học ở tầng sau có tính tổng quát hơn so với mô hình trước.

Các kiến trúc cấu Deep Learning có khác biệt so với kiến trúc của ANN thông thường là bởi vì chiều sâu hay số lớp của chúng. Một ví dụ cho Deep Learning là mô hình cấu trúc Deep Neural Networks (DNNs). DNNs thông thường sẽ có số lớp ẩn nhiều hơn so với ANN. Trong Deep Learning, mỗi lớp sẽ đóng vai trò huấn luyện một đặc trưng nhất định dựa vào đầu ra

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.



Hình 3.6: Các đặc trưng trên khuôn mặt từ tập dữ liệu sẽ được học qua quá trình huấn luyện. các lớp sau có thể tổng hợp và từ dữ liệu đầu ra của lớp trước đó cho ra những đặc trưng tổng quát và phức tạp hơn.

của lớp trước đó. Các lớp càng về sau sẽ học được các thông tin quan trọng hơn và càng mang tính tổng quát được tổng hợp từ những lớp trước đó. Có thể hiểu DL là mô hình tự học các đặc trưng theo từng cấp bậc (feature hierarchy). Đặc trưng được tự học qua quá trình huấn luyện DL. Đặc trưng ở mức càng cao thì độ phức tạp và tính tổng quát càng cao (hình 3.6) là một ứng dụng minh họa trong việc áp dụng Deep Neural Networks cho bài toán nhận dạng khuôn mặt).

Đối với các mô hình DL với số lượng lớp lớn, thì lượng tham số có thể lên đến hàng tỷ tham số. Điều này làm cho DL phức tạp hơn rất nhiều và có khả năng xử lý lượng dữ liệu lớn. Ngoài ra, Deep Learning có khả năng xử lý trên cả dữ liệu không gián nhãn (unlabeled) và cả dữ liệu không có cấu trúc (unstructured data). Đối với dữ liệu không có cấu trúc, Deep Learning cho phép tìm ra những thông tin hoặc cấu trúc bị từ dữ liệu huấn

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

luyện. Với dữ liệu không gián nhãn, thì DL sẽ liên tục tiến hành tái cấu trúc đầu vào tại đầu ra của mạng và so sánh với dữ liệu huấn luyện nhằm cố gắng cực tiểu hóa độ sai lệch giữa chúng.

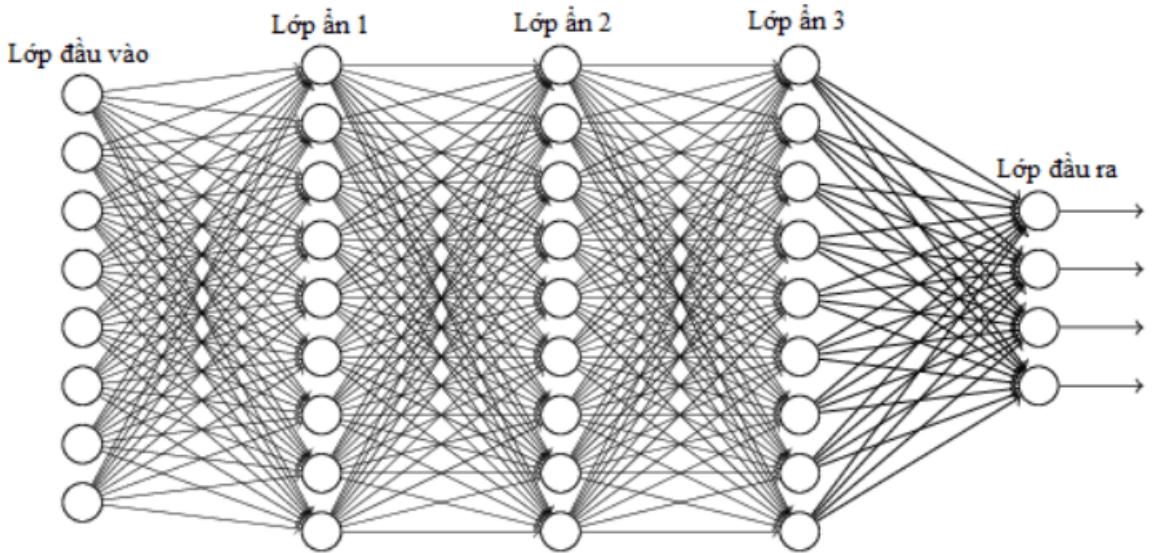
Từ những phương pháp huấn luyện khác nhau, ta có thể phân loại Deep Learning thành ba nhóm cơ bản: Deep Learning cho học có giám sát, Deep Learning cho học không giám sát và Hybrid Deep Learning - Deep Learning có kết hợp cả hai phương pháp học có giám sát và không giám sát.

Deep Learning cho học có giám sát (Supervised Learning). Là một phương pháp trong máy học với mục đích nhằm tạo ra một mô hình có khả năng dự đoán kết quả từ một tập dữ liệu đã được gán nhãn. Tập dữ liệu chứa các mẫu dữ liệu huấn luyện, mỗi mẫu huấn luyện có cấu trúc là một cặp giá trị đầu vào và các nhãn tương ứng. Thuật toán học có giám sát sẽ tiến hành học từ tập dữ liệu được gán nhãn tạo ra một mô hình hoàn chỉnh và từ mô hình đó để áp dụng xử lý cho những mô hình mới. Chức năng chủ yếu của phương pháp học có giám sát đó là giải quyết các bài toán phân lớp (Classification) và toàn toán hồi quy (Regression).

Trong Deep Learning, Phương pháp học có giám sát chủ yếu sẽ được áp dụng để giải quyết hai nhóm bài toán chính đó là bài toán nhận dạng và bài toán dự đoán. Dưới đây sinh viên sẽ giới thiệu hai kiến trúc Deep Networks nổi bật nhất trong phương pháp học có giám sát là Deep Neural Networks và Convolutional Neural Networks.

- **Deep Neural Networks (DNNs):** Là một phiên bản mô hình mạng neural nhiều lớp của một ANN truyền thống. Các lớp trong Deep Neural Networks kết nối với nhau theo kiểu fully-connected, tức là một neural ở lớp này sẽ kết nối tới toàn bộ các neural ở lớp tiếp theo. Điều này sẽ dẫn đến số lượng liên kết neural rất lớn. Vì kiến trúc đa lớp nên việc huấn luyện sẽ khó khăn và phức tạp hơn so với các mô

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.



Hình 3.7: Minh họa mô hình Deep Neural Networks với ba lớp ẩn, một lớp đầu vào và một lớp đầu ra với cùng kiểu liên kết fully-connected.

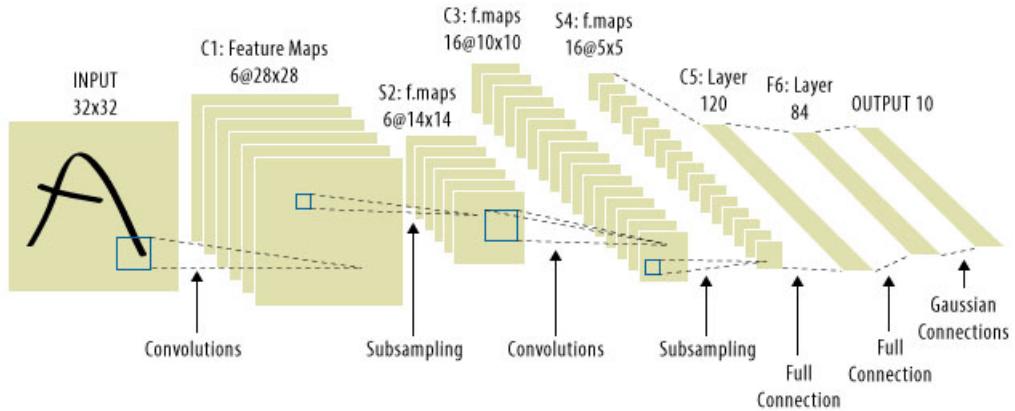
hình ANN truyền thống.

- **Convolutional Neural Networks (CNN):** hiện tại là thuật toán đang cho kết quả tốt nhất trên các bài toán thuộc lĩnh vực thị giác máy tính. Hình 3.10 là một ví dụ về việc sử dụng CNN cho bài toán nhận diện chữ viết tay mang tên LeNet5 được phát minh bởi giáo sư Yan LeCun năm 1998. Mô hình LeNet5 được huấn luyện trên tập MNIST⁷ và có sai số rất nhỏ (gần 1%). Lý thuyết và những thành phần cụ thể của mô hình CNN sẽ được trình bày trong phần tiếp theo của chương này.

Deep Learning cho học không giám sát (Unsupervised Learning): là kỹ thuật học từ dữ liệu không được dán nhãn. Mục đích của học không giám sát là tìm ra các thông tin hoặc cấu trúc bị ẩn dưới dữ liệu.

⁷<http://yann.lecun.com/exdb/mnist/>

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.



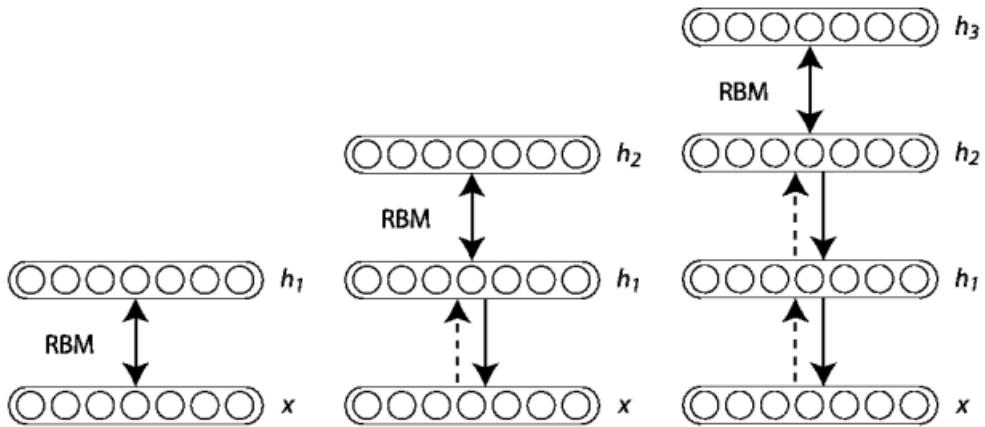
Hình 3.8: Minh họa mô hình CNN sử dụng cho bài toán nhận dạng chữ viết tay.

Trong Deep Learning, các mô hình học không giám sát thường được áp dụng để giải quyết các bài toán như gom nhòm dữ liệu, phát hiện bất thường, hay khởi tạo trọng số cho các mô hình học có giám sát. Một số thuật toán DL tiêu biểu cho việc học không giám sát đó là Deep Belief Networks (DBNs) hay Deep Autoencoders (DAs). Dưới đây sẽ trình bày sơ lược về kiến trúc của 2 thuật toán này.

- **Deep Belief Networks (DBNs):** có thể định nghĩa như một cấu trúc chứa nhiều mạng Restricted Boltzmann Machine (RBMs)⁸. Các lớp ẩn của các RBMs sẽ là lớp đầu vào của RBMs tiếp theo trong mạng DBNs. Công việc huấn luyện DBNs rất phức tạp và tốn nhiều thời gian hơn so với các kiến trúc DL khác. DBNs được huấn luyện bằng cách huấn luyện từng RBMs thành phần. Kết quả đầu ra của RBMs đầu tiên sẽ làm đầu vào cho RBMs tiếp theo.
- **Deep Autoencoders (DAs):** phát triển dựa trên mô hình autoencoder - mô hình mạng neural được huấn luyện bằng thuật toán lan

⁸https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.



Hình 3.9: Minh họa kiến trúc mô hình Deep Belief Networks (DBNs).

truyền ngược mà trong đó không sử dụng dữ liệu được gián nhãn mà thực hiện bằng cách cho giá trị đầu ra bằng giá trị đầu vào. hay nói các khác là tìm một hàm xấp xỉ sao cho đầu ra bằng đầu vào.

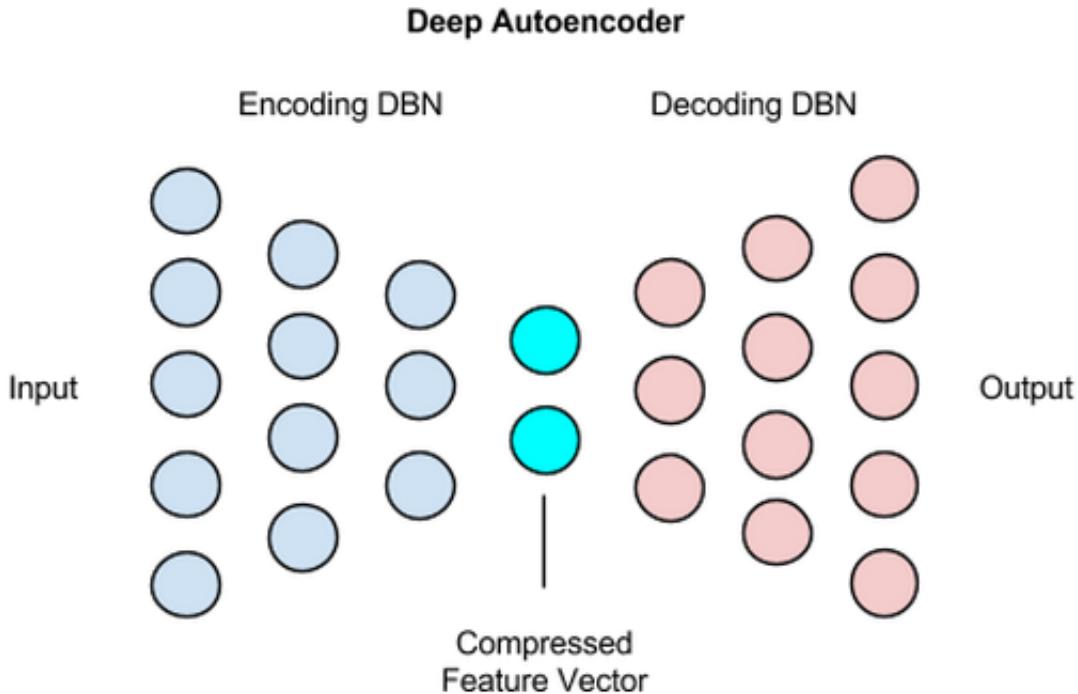
Hybird Deep Learning (HDLs) hay còn được gọi với một tên khác là mô hình Deep Learning "lai". Mô hình này kết hợp cả hai phương pháp học có giám sát (Supervised Learning) và học không giám sát (Unsupervised Learning) nhằm mục tiêu nâng cao độ chính xác. Cụ thể hơn, trong huấn luyện mạng thường sử dụng mô hình học không giám sát để tiến hành khởi tạo các trọng số cho mô hình học có giám sát. Ví dụ như ta có thể sử dụng DBNs để khởi tạo trọng số cho mô hình CNN. Khi đó, ta nói rằng mô hình kết hợp DBN-CNN là mô hình HDLs.

3.1.3 Convolutional Neural Networks

CNN là một trong những thuật toán DL cho kết quả tốt nhất hiện nay trong hầu hết các bài toán về thị giác máy tính. Nó là một kiểu của mạng

⁹<https://deeplearning4j.org/deepautoencoder>

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỔI TƯỢNG.



Hình 3.10: Minh họa kiến trúc mạng DAs⁹

neural nhân tạo với các kết nối giữa các neural được lấy ý tưởng dựa trên sự tổ chức của các neural trong vỏ não thị giác của động vật. Vào năm 1968, hai nhà thần kinh học **Hubel**¹⁰ và **Wiesel**¹¹ đã tìm ra rằng, trong vỏ não thị giác của động vật có chứa các cấu trúc thần kinh phức tạp. Nó chịu trách nhiệm phát hiện ra ánh sáng tại những vùng nhỏ chồng chéo của vùng thị giác (visual field) những vùng này còn được gọi là vùng tiếp nhận (receptive fields). Từ ý tưởng đó, CNN cũng chứa nhiều lớp neural với những vùng kết nối giữa hai lớp neural với nhau. Điều này giúp cho CNN tìm ra những thể hiện tốt hơn. Về cơ bản, CNN là một kiểu mạng ANN truyền thống với kiến trúc chính gồm nhiều thành phần được ghép

¹⁰https://en.wikipedia.org/wiki/David_H._Hubel

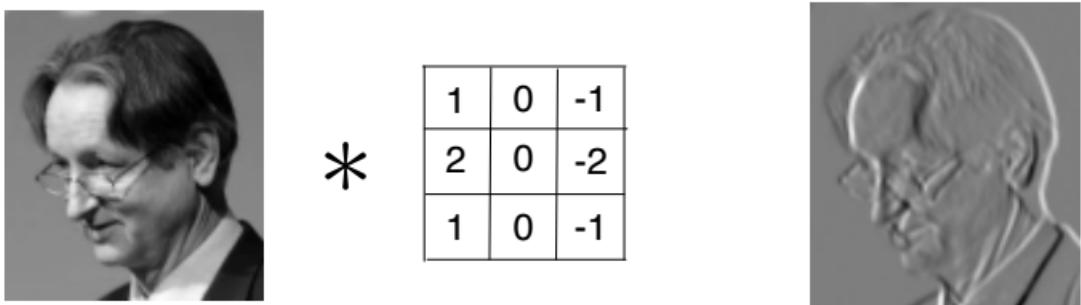
¹¹https://en.wikipedia.org/wiki/Elie_Wiesel

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

nối với nhau theo cấu trúc lớp bao gồm: Convolution, Pooling, ReLU và Fully connected. Trong phần này sẽ trình bày chi tiết về các thành phần của CNN.

3.1.3.1 Lớp Convolution

Tầng Convolution (Conv) là tầng quan trọng nhất trong kiến trúc của CNN. Conv dựa trên lý thuyết xử lý tín hiệu số, việc lấy tích chập sẽ giúp trích xuất được những thông tin quan trọng từ dữ liệu. Conv hoạt động dựa trên đầu vào được biểu diễn bằng một ma trận hai chiều. Ta có thể hình dung phép tính này được thực hiện bằng cách dịch chuyển một cửa sổ mà ta gọi là kernel trên ma trận đầu vào. Trong đó, kết quả trước khi mỗi lần dịch chuyển được thu được bằng cách tính tổng tích chập (tích của các giá trị giữa hai ma trận tại vị trí tương ứng), sau khi trượt qua hết toàn bộ ma trận đầu vào, kết quả cho ra một ma trận kết quả có kích thước bé hơn hoặc bằng ma trận đầu vào. Kernel sẽ được sinh ra sau quá trình huấn luyện để có thể trích xuất ra những đặc trưng phù hợp nhất với yêu cầu bài toán.

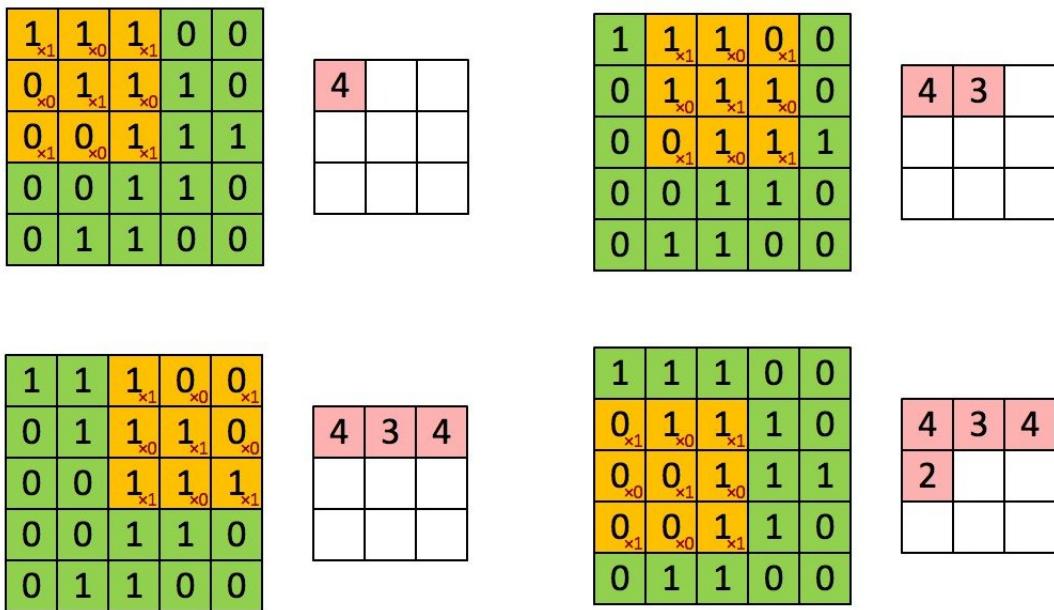


Hình 3.11: minh họa cho việc tính tích chập với ảnh đầu vào ở bên trái, ở giữa là ma trận tích chập (kernel) và bên phải là đặc trưng thu được (feature map)

Khi áp dụng phép tính Conv cho xử lý hình ảnh, Người ta nhận thấy

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỔI TƯỢNG.

rằng Conv sẽ giúp biến đổi các thông tin đầu vào thành các yếu tố đặc trưng (nó tương tự như bộ phát hiện nhằm phát hiện ra các đặc trưng như cạnh, hướng, ...). Hình 3.11 minh họa cho việc áp dụng phép tính Conv trên ảnh và cho ra kết quả là một feature map. Cụ thể hơn, Conv sẽ trích xuất đặc trưng của ảnh đầu vào qua các vùng ảnh nhỏ. Các vùng này được gọi là Local Receptive Field (LRF). Tích chập sẽ tính toán trên các LRF chồng lấp lên nhau. Độ chồng lấp này phụ thuộc vào hệ số trượt S (stride) của từng kiến trúc mạng cụ thể. Nếu sử dụng với hệ số trượt $S = \alpha$, thì tương ứng LRF (bằng kích thước với kernel) sẽ dịch chuyển α đơn vị pixel sau mỗi lần tích chập.



Hình 3.12: minh họa quá trình tính toán Conv với hệ số trượt bằng 1.

Ảnh đầu vào sau khi thực hiện quá trình tích chập sẽ thu được feature map, số LRF ở ảnh đầu vào sẽ tương ứng với số neural ở feature map và kernel sẽ là trọng số liên kết mỗi LRF với một neural ở feature map. Lớp conv có thể chứa một hoặc nhiều feature map. Nếu lớp conv có K feature

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

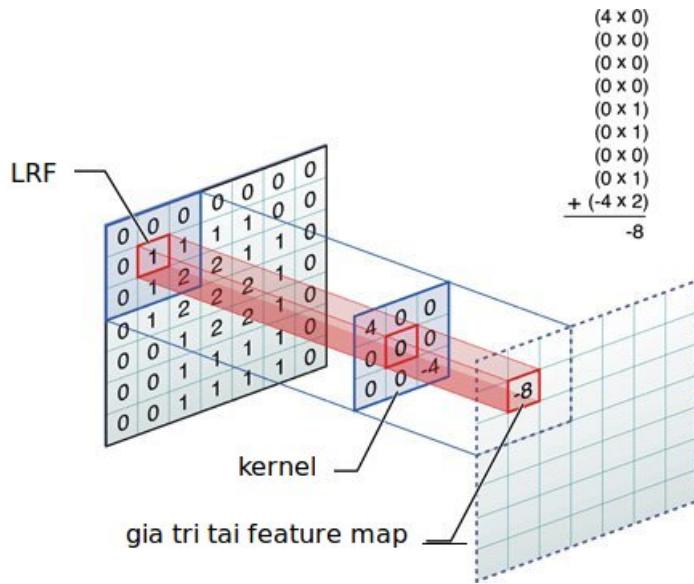
map, thì ta nói lớp conv này có độ sâu là k . Để hình dung rõ hơn về quá trình này, sau đây sẽ minh họa quá trình trích xuất đặc trưng từ ảnh đầu vào cụ thể như sau: thực hiện xử lý tính giá trị đầu ra của một ảnh có kích thước $W_1 \times H_1 \times D_1$ (W_1 và H_1 lần lượt là chiều rộng và chiều cao của ảnh và D_1 là chiều sâu hay thực chất là giá trị tại 3 kênh màu tương ứng của ảnh RGB). Khi đó, một Conv như một cửa sổ trượt (sliding window, còn được gọi là kernel, filter hay feature detector) với kích thước $F \times F$ - giả sử trong trường ta sử dụng K filter. Trong quá trình xử lý, mỗi filter sẽ được tính toán với tất cả các LRF trong hình và $S = \alpha$. Trong một số trường hợp để cân bằng giữa số bước di chuyển và kích thước của ảnh, người ta đã chèn thêm P pixel với một giá trị màu được gán (thông thường là 0) xung quanh viền của ảnh. Sau cùng ta thu được ma trận đầu ra (feature map) với kích thước $W_2 \times H_2 \times D_2$ với giá trị cụ thể như sau:

- $W_2 = \frac{(W_1 - F + 2P)}{\alpha} + 1$
- $H_2 = \frac{(H_1 - F + 2P)}{\alpha} + 1$
- $D_2 = K$

Giá trị tại mỗi ô trong ma trận filter có kích thước là $(F \times F \times D_1) + 1$ (cộng một ô đây là tham số ngưỡng của filter) sẽ tương ứng là trọng số của filter. Các giá trị này sẽ không đổi trong suốt quá trình dịch chuyển tính toán trên toàn bộ hình ảnh. Đây là một đặc tính quan trọng của CNN, bởi nó làm giảm thêm trọng số cần học cho quá trình huấn luyện mạng. Qua đó, ta có thể ước tính tổng số tham số cần học trong quá trình sử dụng Conv là $(F \times F \times D_1) \times K + K$ (ở đây cộng thêm k tham số ngưỡng của k filter).

Áp dụng vào ví dụ cụ thể ở hình 3.14, đầu vào là một ảnh màu với kích thước $(32 \times 32 \times 3)$ ($W_1 = H_1 = 32$ và $D_1 = 3$ chỉ giá trị của các kênh

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.



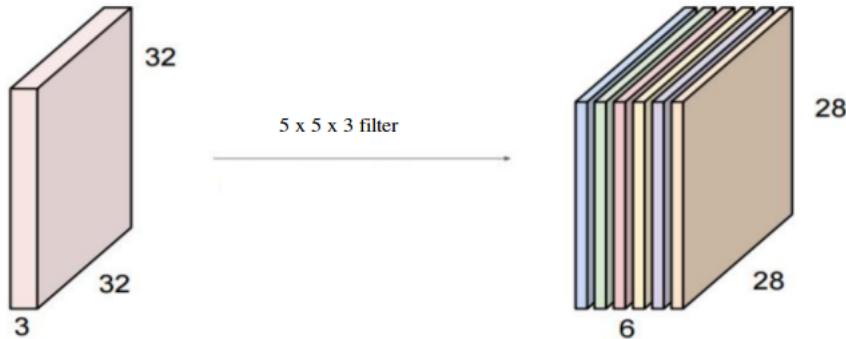
Hình 3.13: các thành phần quá trình tính toán tích chập.¹²

màu RGB). Với số lượng filter $K = 6$, trong đó mỗi filter có kích thước $(5 \times 5 \times 3)$. $F = 3$ với bước di chuyển $S = 1$ và $P = 0$. Tương ứng với mỗi filter, ta thu được 6 feature map khác nhau ở đầu ra. trong đó:

- $W_2 = \frac{32-5}{1} + 1 = 28$
- $W_1 = \frac{32-5}{1} + 1 = 28$
- $D_1 = 6$

Mỗi neural trong một feature map sẽ có số tham số là $(F \times F \times D_1) = 5 \times 5 \times 3 + 1$. nếu không sử dụng tính chất shared weights thì tổng số tham số trong mạng cần phải học cho tất cả các feature map là $(28 \times 28 \times 6) \times (5 \times 5 \times 3 + 1) = 357504$ (tham số). Mặc dù con số đó đã nhỏ hơn nhiều so với việc không sử dụng Conv nhưng con số trên vẫn lớn hơn rất nhiều so với $(F \times F \times D_1) \times K + K = (5 \times 5 \times 3) \times 6 + 6 = 456$ (tham số) khi sử dụng shared weights.

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.



Hình 3.14: Ví dụ kết quả tính toán Conv trong hình ảnh.

Nói tóm lại, Việc sử dụng Conv có những ưu điểm sau:

- Giảm số lượng tham số: trong mạng ANN truyền, mỗi neural ở lớp trước sẽ liên kết tới tất cả các neural ở lớp kế tiếp (fully connected). Dẫn đến tình trạng số lượng các tham số trong mạng rất lớn. Đây là nguyên nhân chính gây việc thời gian huấn luyện mạng rất lâu. Với việc sử dụng Conv, cho phép chia sẻ trọng số liên kết (shared weights), cũng như sử dụng LRF giúp cho số lượng các tham số trong mạng giảm đi đáng kể.
- Các tham số trong quá trình sử dụng Conv hay các giá trị trong các filter sẽ được học trong quá trình huấn luyện mạng. Sau quá trình tính toán, ta sẽ thu được các thông tin đặc trưng như góc, cạnh, Đốm màu trong ảnh, ... Chính vì thế, Conv còn có vai trò xây dựng mô hình tự học ra các đặc trưng.

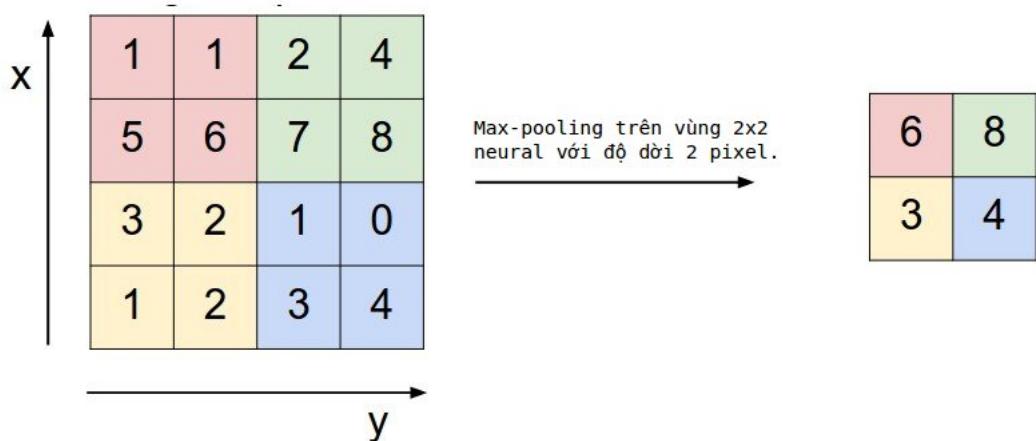
3.1.3.2 Lớp Pooling

Lớp Pooling được sử dụng đúng ngay sau convolution. Pooling giúp cho mạng giảm số lượng tham số, từ đó giúp đơn giản hóa quá trình tính toán

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

của CNN và qua đó góp phần giải quyết vấn đề overfitting khi huấn luyện mạng.

Có nhiều toán tử pooling như Sum-pooling, Max-pooling, L2-pooling nhưng Max-pooling được sử dụng phổ biến nhất trong kiến trúc mạng CNN vì nó cho kết quả hơn so với những toán tử còn lại.



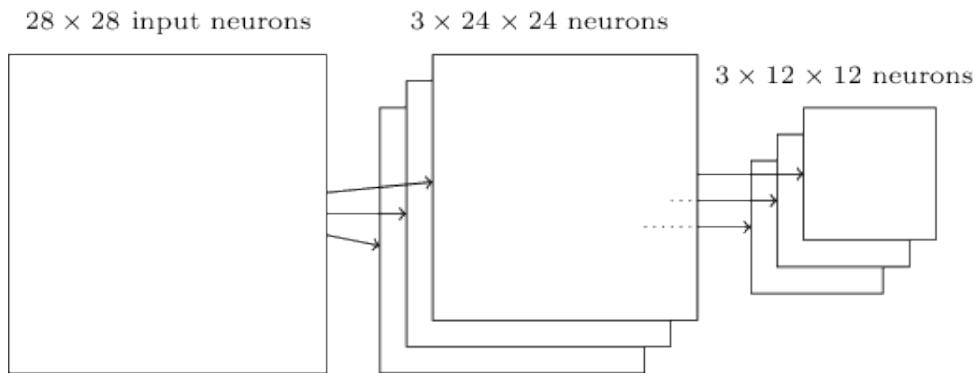
Hình 3.15: Max-pooling lấy giá trị lớn nhất trên vùng $[2 \times 2]$ trên kích thước $[4 \times 4]$ và thu được đầu ra kích thước $[2 \times 2]$.

Ngoài ra, Max-pooling còn giúp tạo ra tính bất biến dịch chuyển (translation invariance) cho đặc trưng. Cụ thể, dù đổi tượng trong hình ảnh đầu vào có sự dịch chuyển nhỏ thì mạng vẫn có khả năng phân lớp chính xác được đổi tượng. Đó là bởi vì max-pooling chọn ra neural có giá trị đầu ra tối đa tại mỗi vùng neural của lớp trước và tổng hợp thành lớp sau. Việc chọn neural có giá trị tín hiệu lớn nhất được xem như chọn ra đặc trưng tốt nhất để xử lý. Chính vì thế, khả năng phân lớp chính xác đổi tượng dựa trên đặc trưng này vẫn không thay đổi và giúp cho CNN đạt được tính ổn định khi đổi tượng di chuyển.

Trong lớp Conv, có thể có nhiều feature map, tương ứng với mỗi feature

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

map sẽ có một lớp max-pooling. Hình 3.16 là một ví dụ minh họa, với lớp đầu vào kích thước $[28 \times 28]$ giả sử ta thu được ba feature map kích thước $[24 \times 24]$ và ba lớp max-pooling kích thước $[12 \times 12]$, ta sử dụng kernel kích thước $[5 \times 5]$ và max-pooling lấy giá trị lớn nhất tại mỗi vùng $[2 \times 2]$ neural.



Hình 3.16: Minh họa áp dụng Pooling từ lớp Conv

3.1.3.3 Lớp Normalization

Lớp Normalization (Norm) là lớp giúp chuẩn hóa dữ liệu đầu ra cho các lớp trong CNN trước khi được truyền đi tiếp. Trong những kiến trúc CNN lớn và phức tạp, Lớp Norm sẽ chuẩn hóa các giá trị neural trước khi chúng được truyền đến hàm ReLU. Hàm ReLU tuy giúp rút ngắn thời gian huấn luyện, nhưng nếu không điều chỉnh trọng số phù hợp, hàm ReLU sẽ rất dễ gặp phải vấn đề "dying ReLU" khiến cho mạng trở nên chậm hơn khi huấn luyện. Lớp Norm lúc này sẽ chuẩn hóa và tạo ra các giá trị tích chập phù hợp để tránh cho ReLU rơi vào giá trị 0. Tránh việc gradient xấp xỉ bằng 0 khiến cho tốc độ học của mạng trở nên rất chậm.

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

3.2 Áp dụng Deep Learning cho bài toán đếm đối tượng

Từ những kiến thức nền tảng đã trình bày ở hai phần trên, phần này sinh viên sẽ tiến hành trình bày một phương pháp áp dụng giải quyết cho bài toán đếm đối tượng. Cụ thể về chi tiết kỹ thuật và các mô hình được sử dụng để giải quyết bài toán.

3.2.1 Mô hình đếm đối tượng

Phương pháp được sử dụng sau đây là một phương pháp thuộc nhóm Counting by Regression. Đầu tiên, Sinh viên sẽ trình bày mô hình tổng quát áp dụng cho bài toán, kèm theo đó là các ký hiệu và công thức tính toán được sử dụng. Đối với phương pháp này, ta định nghĩa lại bài toán đếm đối tượng thành mô hình bài toán ước lượng mật độ đối tượng (density estimation) được đề cập trong công trình của Victor and Zisserman [7].

Quá trình huấn luyện của phương pháp này đòi hỏi cần phải có một tập các hình ảnh gán nhãn. Cụ thể hơn, tất cả các đối tượng trong tập hình ảnh huấn luyện sẽ được đánh dấu bằng một dấu chấm. Giả sử ta có một tập huấn luyện với N bức hình $I_1, I_2, I_3, \dots, I_N$, với mỗi bức hình I_i ta có một tập các điểm đã được đánh dấu bằng các điểm 2 chiều (2D) $P_i = \{P_1, P_2, \dots, P_S\}$. Ở đây, S là tổng số lượng các nhãn (dấu chấm) đã được đánh dấu.

Trong hướng tiếp cận này, ta sẽ sử dụng một hàm mật độ để tính toán ra những giá trị số thực biểu diễn cho giá trị của toàn bộ pixel trong bức hình, sao cho tất cả các giá trị này sẽ biểu diễn phù hợp nhất cho các đối tượng cần đếm. Cụ thể hơn, với một hình ảnh huấn luyện I . Trong quá trình tạo ra ground truth, ta định nghĩa một hàm tính mật độ nhằm tạo ra ground truth D_I cho mỗi ảnh I , bằng việc sử dụng hàm Gaussian hai

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

chiều cho mỗi điểm được đánh dấu.

$$\forall p \in I_i \quad D_i(p) = \sum_{p \in P_i} \mathcal{N}(p; P, \sigma^2 1_{2 \times 2}) \quad (3.1)$$

Ở đây, p đại diện cho pixel đang xử lý và $\mathcal{N}(p; P, \sigma^2 1_{2 \times 2})$ là hàm Gaussian hai chiều để ước lượng giá trị tại điểm p . Từ bản đồ mật độ D_I thu được, ta tính N_I là tổng các đối tượng xuất hiện trong hình bằng cách tính tổng tất cả gác giá trị pixel trên toàn bộ D_I theo công thức.

$$N_I = \sum_{p \in I} D_I(p) \quad (3.2)$$

Bằng cách gán nhãn sử dụng dấu chấm cho mỗi đối tượng và tính tổng giá trị của hàm Gaussian trên toàn bộ pixel của ảnh. Vì thế tất cả các đối tượng sẽ được đếm một cách đầy đủ cả trong trường hợp những đối tượng bị chồng lấp lên nhau.

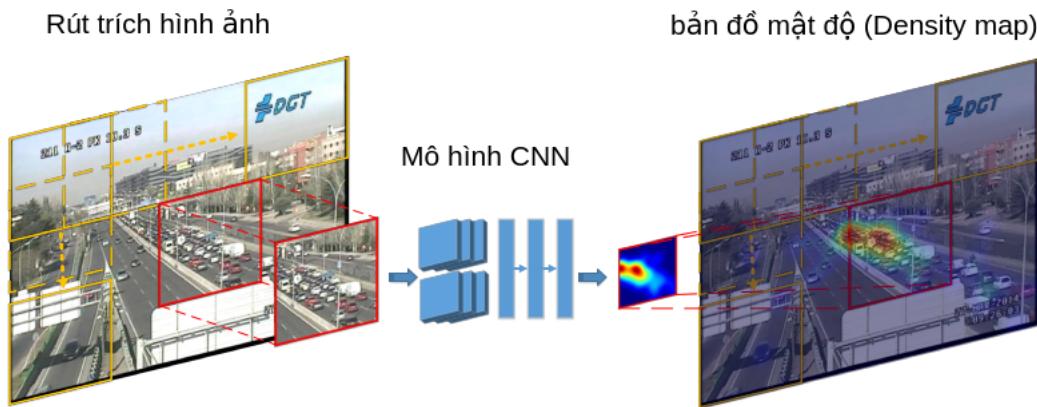
Với mô hình bài toán ước lượng số lượng đối tượng, mục tiêu chính của phương pháp này chính là thiết kế ra một kiến trúc mô hình Deep Learning có khả năng học và tạo ra một hàm hồi quy phi tuyến tính (non-linear regression) \mathcal{R} . Cụ thể, \mathcal{R} sẽ nhận đầu vào là một patch P của ảnh và sau khi xử lý sẽ cho ra một dự đoán ước lượng bản đồ mật độ $D_{pred}^{(P)}$

$$D_{pred}^{(P)} = \mathcal{R}(P|\Omega) \quad (3.3)$$

Trong đó Ω là tập các tham số của mô hình CNN sử dụng. Với mỗi patch hình ảnh $P \in \mathbb{R}^{h \times w \times c}$, trong đó h và w là chiều cao và chiều rộng, c là số kênh màu của patch ảnh. Và kết quả thu được là $D_{pred}^{(P)} \in \mathbb{R}^{h' \times w'}$, trong đó h' và w' lần lượt là chiều cao và chiều rộng của bản đồ mật độ sau khi tính

3. DEEP LEARNING CHO BÀI TOÁN ĐẾM ĐỐI TƯỢNG.

toán. Như vậy, với một hình ảnh đầu vào, đầu tiên ta sẽ tiến hành rút ra những patch của nó, sau đó mô hình CNN sẽ xử lý và cho ra một bản đồ mật độ tương ứng với mỗi patch. và cuối cùng ta sẽ tích hợp tất cả các bản đồ mật độ thu được để cho ra density map của toàn bộ bức hình (minh họa trong hình 3.17). Sau đây, sinh viên sẽ trình bày chi tiết kiến trúc mô hình CNN sử dụng trong mô hình này.



Hình 3.17: Mô hình giải quyết bài toán đếm đối tượng với đầu vào là một bức ảnh và đầu ra là một bản đồ mật độ đối tượng trong bức ảnh đó.

3.2.2 Mô hình CCNN

Trong phần này, sinh viên sẽ trình bày kiến trúc CNN đầu tiên được sử dụng, mô hình Counting Convolutional Neural Networks (CCNN). Kiến trúc của mô hình CCNN bao gồm 6 lớp Convolutional. Lớp Conv1 và Conv2 sử dụng kernel với kích thước 7×7 cùng độ sâu là 32.

3. DEEP LEARNING CHO BÀI TOÁN ĐÊM ĐỐI TƯỢNG.

3.2.3 Mô hình Hydra CNN

3.3 Kết chương

Chương 4

THỰC NGHIỆM VÀ ĐÁNH GIÁ

Để hiểu rõ hơn về phương pháp và đánh giá độ chính xác của mô hình đã trình bày ở trên. Trong chương này sinh viên sẽ tiến hành chạy thực nghiệm các mô hình trình CNN, Hydra CNN cho ba bộ dữ liệu TRANCOS, UCSD và UCF. Từ kết quả thu được sinh viên sẽ tiến hành so sánh với các phương pháp khác đã được đề xuất và bên cạnh đó sẽ xây dựng một công cụ mô hình hóa trực quan dữ liệu nhằm tìm ra những ưu điểm và hạn chế của phương pháp. Để từ đó đặt ra những giả thiết cho các vấn đề còn tồn tại của phương pháp.

4.1 Bộ dữ liệu thực nghiệm.

Trong phạm vi khóa luận này, sinh viên tiến hành đánh giá trên ba bộ dữ liệu tương ứng với ba tiêu chuẩn đánh giá. trong đó, hai bộ được sử dụng để đánh giá cho bối cảnh đám đông: bộ UCSD (dữ liệu người đi bộ), và *UCF_CC_50* (dữ liệu đám đông mật độ lớn). Và bộ còn lại là bộ dữ liệu TRANCOS để đánh giá cho bối cảnh đếm các phương tiện giao thông trong các bối cảnh giao thông tắc nghẽn.

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1.1 Bộ Dữ liệu UCSD

UCSD [8] là một bộ dataset phổ biến thường được sử dụng để đánh giá cho các phương pháp liên quan đến phân tích người đi bộ. Bộ dữ liệu này được thu thập từ một camera giám sát cố định đặt ở lối đi dành cho người đi bộ ở đại học California, San Diego với một giờ dữ liệu video. Video gốc được ghi với tốc độ 30 fps và kích thước mỗi frame là 740×480 , sau đó dữ liệu được hạ xuống 10 fps và kích thước còn 238×158 mỗi frame. Với 2000 frame đầu tiên (200 giây dữ liệu video) được gán nhãn ground true với tổng số 49,855 nhãn được dán. Các hình này được gán nhãn bằng cách gán một dấu chấm đại diện cho mỗi người đi bộ, và một khu vực tính toán (ROI - Region of Interest) được đánh dấu trên mỗi ảnh.

4.1.2 Bộ dữ liệu UCF – CC – 50

Đây là một bộ dữ liệu chứa hình ảnh đám đông với mật độ lớn. Được thu thập chủ yếu từ FLICKR với mục đích cung cấp dữ liệu cho nghiên cứu đám đông (minh họa trong hình 4.1¹). Bộ dataset này bao gồm 50 bức ảnh đám đông chứa số người trong hình từ 94 đến 4543 người, và trung bình là 1280 người mỗi hình. Mỗi người được gán nhãn bằng một dấu chấm. Những hình ảnh này chứa cảnh cực kỳ đông đúc trong các bối cảnh như: các cuộc biểu tình, các sự kiện âm nhạc, Các sân vận động bóng đá hay các sự kiện ngoài trời. Bộ dataset này dùng để đánh giá và kiểm chứng một thách thức quan trọng trong chủ đề đếm đối tượng khi mà số lượng đối tượng rất lớn cùng các khó khăn như sự che khuất, sự biến dạng do góc nhìn hay quá ít pixel biểu diễn cho một đối tượng. bên cạnh đó, số lượng hình ảnh để huấn luyện giảm đáng kể xuống còn rất ít (50 hình) và trong nhiều bối cảnh khác nhau xen lẫn trong bộ dữ liệu. Đó là những thách thức

¹http://crcv.ucf.edu/data/crowd_counting.php

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

mà bộ dataset này đặt ra.



Hình 4.1: một vài hình ảnh trong bộ dữ liệu UCF

4.1.3 Bộ dữ liệu TRANCOS

Bộ dữ liệu TRANCOS (TRaffic ANd COngestionS) [9] là một dataset dùng để đánh giá tiêu chí các đối tượng bị che khuất lớn của các phương tiện giao thông trong bối cảnh giao thông tắc nghẽn. Bộ dataset bao gồm 1244 hình ảnh được thu thập từ các camera giám sát giao thông thực tế ở thành phố Dirección General de Tráfico Tây Ba Nha. Với tổng số 46796 nhãn phương tiện giao thông. Các phương tiện giao thông được gán nhãn trực tiếp bằng tay bằng các dấu chấm đại diện cho mỗi phương tiện. Và kèm theo đó cung cấp thêm vùng tính toán quan tâm (ROI) tương tự như bộ UCSD cho mỗi hình ảnh. Bộ dữ liệu này cung cấp các hình ảnh chứa các bối cảnh rất đa dạng và hơn thế nữa là các camera còn có nhiều góc nhìn khác nhau trong

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

cùng một cảnh.

4.2 Phương pháp đánh giá thực nghiệm.

Trong phần này, sinh viên sẽ tiến hành chạy thực nghiệm đánh giá hai mô hình CCNN và Hydra CNN trên ba bộ dữ liệu kể trên (UCSD, UCF, TRANCOS). Mỗi bộ dữ liệu sẽ có hình thức đánh giá riêng theo yêu cầu của mỗi dataset cùng các độ đo áp dụng cho mỗi bộ. Sau đây sinh viên sẽ trình bày chi tiết quy trình đánh giá thực nghiệm.

4.2.1 Đánh giá thực nghiệm trên bộ UCSD

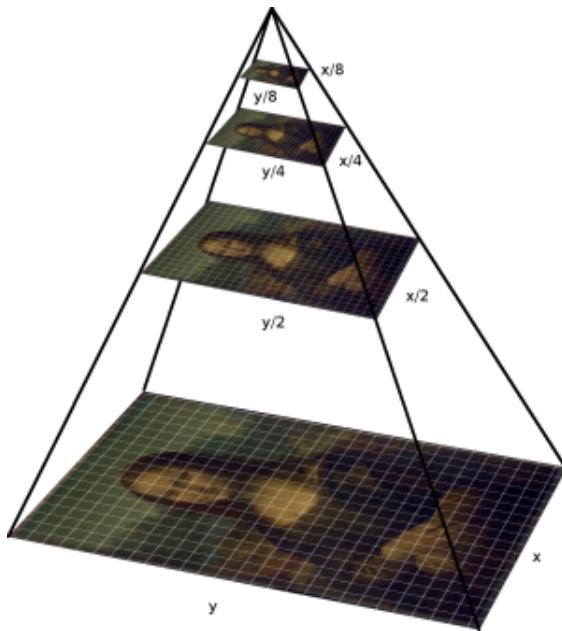
Với bộ dataset UCSD, sinh viên thực hiện quá trình thực nghiệm với cách phân chia dữ liệu sử dụng trong [6, 10, 7, 11, 12]. Bộ dữ liệu được chia thành bốn tập khác nhau để huấn luyện, và tất cả các frame còn lại trong mỗi tập được dùng để test.

- Maximal: được huấn luyện với các frame 600:5:1400
- downscale: huấn luyện với các frame 1205:5:1600
- upscale: huấn luyện với các frame 805:5:1100
- minimal: huấn luyện với các frame 640:80:1360

Trong quá trình huấn luyện cho mô hình CCNN, với mỗi frame hình ảnh, ta tiến hành rút trích ra 800 patch ngẫu nhiên trên toàn bộ hình ảnh với kích thước 72×27 pixel. sau đó ta lật lại lật mỗi patch 180° với mục đích tăng số lượng dữ liệu huấn luyện. Cuối cùng, ta thu được tổng cộng 1600 mẫu dữ liệu huấn luyện với mỗi hình. Những patch với kích thước 72×72 sẽ là đầu vào cho mô hình mạng. Kèm với đó, ta sẽ tạo bản đồ mật độ ground truth bằng phương pháp được sử dụng trong [7]. Ở đây, ta tính

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

toán Gaussian Kernel (với ma trận hiệp phương sai (covariance matrix) $\Sigma = 8.1_{2x2}$) tại điểm chính giữa của mỗi nhãn đối tượng.

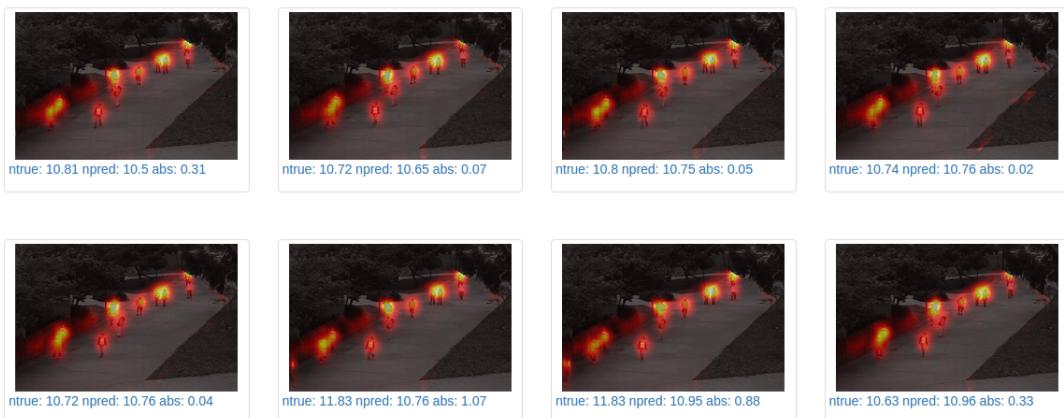


Hình 4.2: Mô hình Pyramid với nhiều tỉ lệ trên một patch.

Để huấn luyện cho mô hình Hydra CNN, ta vẫn sẽ tiến hành theo phương pháp trích xuất các patch được sử dụng trong mô hình CCNN. Điểm khác ở đây chính là với một patch thu được, ta sẽ giải quyết vấn đề tỉ lệ bằng cách áp dụng nhiều tỉ lệ khác cho một patch huấn luyện. Ta xây dựng một mô hình Kim tự tháp (pyramid) với s tỉ lệ khác nhau. Cụ thể hơn, tầng đầu tiên của mô hình Pyramid chưa patch với kích thước ban đầu. Nhưng đến tầng tiếp theo, ta sẽ lấy ra patch từ chính giữa tâm của patch ban đầu với tỉ lệ bằng $\frac{1}{s}$ kích thước của patch ban đầu. Ví dụ, trong trường hợp với mô hình Hydra CNN với hai đầu, tầng đầu tiên chứa patch hình góc ban đầu. và tầng thứ hai chứa patch với kích thước bằng 50% kích thước patch ban đầu. Vì vậy, với 800 patch được trích xuất ngẫu nhiên trong bức hình có kích thước 72×72 pixel. Sau đó tiến hành lấy ra pyramid của mỗi

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

patch như mô tả ở trên. Cuối cùng, khi thu được mô hình cùng bộ tham số cuối cùng, ta sẽ kiểm tra mô hình trên dữ liệu test. Với mỗi hình test, ta sử dụng cửa sổ kích thước 10×10 pixel trượt qua toàn bộ hình ảnh để tính toán cho toàn bộ ảnh. Dưới đây là một số hình ảnh chạy thực nghiệm kiểm tra với độ sai số ít nhất trên bộ dữ liệu () .



Hình 4.3: Một số ví dụ trong bộ dữ liệu test sau khi chạy thực nghiệm với kết quả thể hiện gồm ground truth, kết quả dự đoán và giá trị sai lệch.

4.2.2 Đánh giá trên bộ dữ liệu UCF

Với bộ dữ liệu *UCF_CC_50*, Sinh viên tiến hành thực nghiệm với theo cách thực nghiệm được mô tả trong [?].

4.2.3 Đánh giá trên bộ dữ liệu TRANCOS

Đối với bộ Dữ liệu TRANCOS, sinh viên tiến hành thực hiện đúng theo các chỉ định cài đặt thực nghiệm được sử dụng trong

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.3 Kết quả đánh giá

4.3.1 Kết quả đánh giá trên bộ dữ liệu UCSD

4.3.2 Kết quả đánh giá trên bộ dữ liệu UCF

4.3.3 Kết quả đánh giá trên bộ dữ liệu TRANCOS

Chương 5

ỨNG DỤNG THỰC NGHIỆM VÀ HƯỚNG PHÁT TRIỂN.

Phụ lục A

Đây là phụ lục A

Phụ lục B

Đây là phụ lục B

Tài liệu tham khảo

- [1] C. C. Loy, K. Chen, S. Gong, and T. Xiang, “Crowd counting and profiling: Methodology and evaluation,” in *Modeling, Simulation and Visual Analysis of Crowds*, pp. 347–382, Springer, 2013. [8](#)
- [2] V. Rabaud and S. J. Belongie, “Counting crowded moving objects,” *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 1, pp. 705–711, 2006. [11](#), [12](#)
- [3] G. J. Brostow and R. Cipolla, “Unsupervised bayesian detection of independent motion in crowds,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 594–601, IEEE, 2006. [12](#)
- [4] P. Tu, T. Sebastian, G. Doretto, N. Krahnstoever, J. Rittscher, and T. Yu, “Unified crowd segmentation,” *Computer Vision–ECCV 2008*, pp. 691–704, 2008. [12](#)
- [5] A. C. Davies, J. H. Yin, and S. A. Velastin, “Crowd monitoring using image processing,” *Electronics & Communication Engineering Journal*, vol. 7, no. 1, pp. 37–47, 1995. [14](#)

TÀI LIỆU THAM KHẢO

- [6] D. Onoro-Rubio and R. J. López-Sastre, “Towards perspective-free object counting with deep learning,” in *European Conference on Computer Vision*, pp. 615–629, Springer, 2016. [16](#), [48](#)
- [7] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in *Advances in Neural Information Processing Systems*, pp. 1324–1332, 2010. [41](#), [48](#)
- [8] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–7, IEEE, 2008. [46](#)
- [9] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, “Extremely overlapping vehicle counting,” in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 423–431, Springer, 2015. [47](#)
- [10] C. Zhang, H. Li, X. Wang, and X. Yang, “Cross-scene crowd counting via deep convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 833–841, 2015. [48](#)
- [11] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht, “Learning to count with regression forest and structured labels,” in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 2685–2688, IEEE, 2012. [48](#)
- [12] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, “Crowd counting using multiple local features,” in *Digital Image Computing: Techniques and Applications, 2009. DICTA '09.*, pp. 81–88, IEEE, 2009. [48](#)