

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

—o0o—



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**MỘT THUẬT TOÁN HIỆU QUẢ DỰA TRÊN CẠNH
ĐỂ TỐI ƯU LỀ PHÂN LOẠI TRONG MẶT PHẪNG**

Giảng Viên Hướng Dẫn:	TS. Nguyễn Kiều Linh
Sinh viên thực hiện:	Nguyễn Văn A
Mã sinh viên:	BXXDCCNYYY
Lớp:	DXHTTTY
Niên khóa:	20xx-20xx
Hệ đào tạo:	Đại học chính quy

Hà Nội, 12/2023

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

—o0o—



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**BAO LỒI XẤP XỈ VÀ ỨNG TRONG VIỆC PHÁT HIỆN
ĐỊNH HƯỚNG VÀ ĐÓNG GÓI ĐỐI TƯỢNG.**

Giảng Viên Hướng Dẫn:	TS. Nguyễn Kiều Linh
Sinh viên thực hiện:	Trần Xuân Độ
Mã sinh viên:	B19DCCN183
Lớp:	D19CNPM04
Niên khóa:	2019-2023
Hệ đào tạo:	Đại học chính quy

Hà Nội, 12/2023

Mục lục

Danh sách hình vẽ	iv
Danh sách bảng	v
Danh sách các ký hiệu và chữ viết tắt	vi
Mở đầu	1
1 Giới thiệu bài toán phát hiện, định hướng và đóng gói đối tượng	3
1.1 Giới thiệu	3
1.2 Xây dựng tập bao lồi đầu tiên	4
1.3 Tích chập biến dạng	6
1.4 Thuật toán Jarvis March	7
1.5 Định nghĩa công thức Convex Intersection over Union (CIoU) . .	9
1.6 Hàm mất mát	9
1.7 Thích ứng bao lồi	10
1.7.1 Xây dựng tập các bao lồi	10
1.7.2 Chiến lược phân đoạn tập các bao lồi	11
2 Thuật toán tính bao lồi xấp xỉ	14
2.1 Outer convex approximation	14
2.2 Sơ lược về	17
3 Ứng dụng	19
3.1 Thặng dư bậc hai và ứng dụng	19
3.1.1 Phương trình đồng dư bậc hai.....	19

4 Một số kết quả tính toán	21
Kết luận	23
Tài liệu tham khảo	24

LỜI CẢM ƠN

Đây là mục tùy chọn

.....

Hà Nội, tháng năm 20....

Sinh viên

Nguyễn Văn A

Danh sách hình vẽ

1.1	Minh hoạ vấn đề. (Bên trên) Khi sử dụng cách biểu diễn dạng hộp, các đối tượng có hướng phân bố dày đặc gây ra hiện tượng đặc trưng răng cưa tại các vùng giao của trường tiếp nhận giữa các đối tượng. (Bên dưới) Với cách biểu diễn bao lồi, phương pháp CFA xử lý tốt các đặc trưng nằm trên lưới tích chập thông thường của các đối tượng có hướng phân bố không đều, giải quyết hiện tượng răng cưa đặc trưng hiệu quả	4
1.2	Biểu đồ luồng quy trình thực hiện của bộ phát hiện CFA.	5
1.3	So sánh biểu diễn hộp có hướng (bên trên) so với biểu diễn bao lồi (ở dưới).	5
1.4	So sánh tích chập thông thường và tích chập biến dạng.	6
1.5	a) lấy mẫu tích chập với 9 điểm lấy mẫu. b) để có tích chập biến dạng, thêm độ dịch chuyển vào mỗi điểm lấy mẫu (mũi tên xanh). c) phép biến đổi tỷ lệ. d) phép quay	7
1.6	Quá trình thực hiện tích chập biến dạng.	7
1.7	mô tả các bước thực hiện thuật toán Jarvis March	8
1.8	xây dựng tập các bao lồi để biểu diễn các đối tượng, đặc biệt với những đối tượng phân bố dày đặc	11
1.9	Phân chia tập bao lồi theo hướng dẫn của nguyên tắc nhất quán độ dốc.	12
2.1	Quy trình tự động nhận dạng biển số xe.	17

Danh sách bảng

4.1	Thời gian chạy tính bao lỗi dưới (đơn vị: giây).	22
-----	--	----

Danh mục các ký hiệu và chữ viết tắt

$\text{conv}(D)$	bao lồi của tập hợp D
$\partial(\text{conv}(D))$	biên của bao lồi $\text{conv}(D)$
$\text{CH}(D)$	tập các hình tròn cực biên của tập D
V_C	tập đỉnh của bao lồi $\text{conv}(C)$
$[p, q]$	đoạn thẳng nối p với q
pq	đường thẳng đi qua p và q
.....

Mở đầu

Trong nhiều thập kỷ, chúng ta đã chứng kiến quá trình phát triển đáng kể của bài toán nhận diện đối tượng. Đóng góp vào sự phát triển này chính là việc sử dụng mạng học sâu kết hợp với đa dạng các cách biểu diễn đặc trưng, các database có kích thước lớn hơn, và việc đào tạo trước các mô hình để tiết kiệm thời gian và chi phí. Tuy nhiên, phần lớn các bộ phát hiện đối tượng đều gặp phải vấn đề khi biểu diễn các đối tượng quan sát từ trên xuống, có hướng tùy ý, hoặc các đối tượng có bố cục khác nhau trong quá trình đào tạo. Vấn đề trở nên nghiêm trọng hơn khi các đối tượng phân bố dày đặc, gây ra hiện tượng răng cưa ở các vùng giao nhau của trường tiếp nhận.

Một trong những giải pháp để phát hiện đối tượng có hướng, đó là sử dụng phương pháp làm giàu đặc trưng/mở neo, cung cấp nhiều hơn các đặc trưng để đào tạo các bộ phát hiện. Giải pháp này tuy nhiên lại gây ra sự phức tạp trong tính toán, dễ gây ra sai sót. Một giải pháp khác là định nghĩa bộ biến đổi RoI, áp dụng phép biến đổi không gian RoIs trong khi tiếp tục học các tham số dưới sự giám sát của hộp bao có hướng. Các bộ biến đổi này được cho là linh hoạt, nhạy bén, hoạt động mượt mà, cho phép trường tiếp nhận thích nghi với các đối tượng có hướng. Tuy nhiên, vấn đề về cách điều chỉnh lưới đặc trưng cho đối tượng có bố cục bất kỳ vẫn chưa được giải quyết. Đây chính là nguyên nhân gây ra hiện tượng feature aliasing, xảy ra khi các đối tượng phân bố dày đặc trong khung hình.

Do đó, ta đề xuất một cách tiếp cận khác: sử dụng phương pháp điều chỉnh các đặc trưng bằng bao lồi(convex-hull) dành cho các đối tượng có hướng và phân bố dày đặc. Mục tiêu để điều chỉnh các đặc trưng nằm trong lưới tích chập thông thường với các đối tượng có bố cục phân bố không đều. Ta xây dựng bộ

cục đối tượng thành một bao lồi, có lợi thế hơn so với sử dụng bố cục hình chữ nhật, giúp bao phủ toàn bộ đối tượng nhưng giảm thiểu tối đa diện tích vùng nền của đối tượng. Mỗi bao lồi là tập hợp các điểm đặc trưng định nghĩa đường biên của đối tượng, biểu thị tỷ lệ Convex Intersection over Union để các định vị trí đối tượng. Bên trong bao lồi, các đặc trưng khác nhau biểu thị cho sự xuất hiện của các đối tượng được phân loại khác nhau.

Mục tiêu của đề án là tìm hiểu, nghiên cứu phương pháp phát hiện đối tượng dày đặc có hướng, cụ thể là nghiên cứu phương pháp BeyondBoundingBox dành cho vấn đề trên. Ngoài ra, đề án còn tìm hiểu thuật toán phát hiện bao lồi mới, thay thế vào phương pháp trên, kiểm nghiệm và đo lường độ hiệu quả của thuật toán mới này.

Trong đề án em sẽ tập trung trình bày một số nội dung chính như sau:

Chương 1: Giới thiệu bài toán phát hiện, định hướng và đóng gói đối tượng: Nội dung chương 1 sẽ khái quát các vấn đề và phương pháp nhận dạng đối tượng, trình bày về các phương pháp liên quan, nguyên lý và cách thức triển khai, giới thiệu sử dụng thuật toán bao lồi xấp xỉ để thực hiện bài toán.

Chương 2: Trình bày thuật toán bao lồi xấp xỉ: Nội dung của chương 2 sẽ giới thiệu thuật toán, lý thuyết và triển khai thuật toán bằng code C++

Chương 3: Thực nghiệm và kết quả: Nội dung của chương 3 Áp dụng bao lồi xấp xỉ cho bài toán phát hiện, định hướng và đóng gói đối tượng, cách thức triển khai bộ phát hiện, cách thức thay thế thuật toán.

Chương 4: Tổng kết: Tổng kết bài toán, tóm tắt những kết quả đã đạt được và còn chưa đạt được. Từ đó đề xuất mục tiêu hướng tới cũng như hướng nghiên cứu, phát triển tiếp theo.

Chương 1

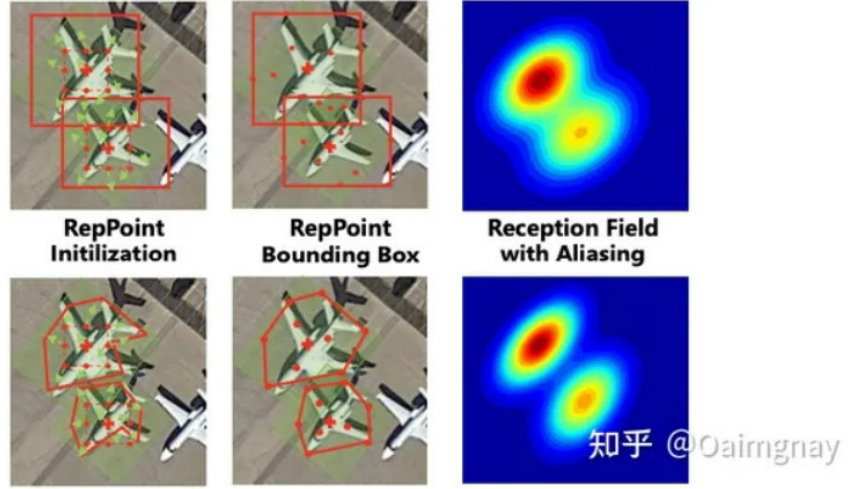
Giới thiệu bài toán phát hiện, định hướng và đóng gói đối tượng

Chương 1 đặt vấn đề về bài toán phát hiện đối tượng, trình bày lý thuyết về phương pháp BeyondBoundingBox.

1.1 Giới thiệu

Trong bài toán phát hiện đối tượng, việc định vị và phát hiện đối tượng dày đặc vẫn còn là một vấn đề thách thức do vấn đề đặc trưng răng cưa (feature aliasing), tức là các pi. Có các giải pháp đã được sử dụng: làm giàu các đặc trưng (enhance features) và sử dụng anchors. Nhược điểm của các phương pháp này: làm cho cấu trúc mạng trở nên phức tạp hơn, tăng thời gian huấn luyện (training) và suy luận (inference). Một phương pháp khác đó là sử dụng biến đổi RoI, nhưng phương pháp này không thích ứng tốt với các đối tượng có hướng bất kỳ, đặc biệt đối với các đối tượng xuất hiện dày đặc trong hình ảnh.

Để giải quyết các vấn đề trên, một phương pháp đã được đưa ra: phương pháp thích ứng bao lồi (convex-hull feature adaptation-CFA). CFA được thực hiện dựa trên phương pháp biểu diễn bao lồi, định nghĩa một tập các điểm đặc trưng, giới hạn phạm vi của đối tượng mục tiêu sử dụng chỉ số CIoU. CFA đạt được sự phân bố đặc trưng tối ưu nhờ vào việc xây dựng tập bao lồi và phân chia linh hoạt các bao lồi thành các bao lồi âm và bao lồi dương. CFA cũng xem xét sự chồng chéo nhau giữa bao lồi dự đoán và bao lồi thực tế, phạt các bao lồi được dùng chung bởi nhiều đối tượng, giảm thiểu hiện tượng đặc trưng răng



Hình 1.1: Minh hoạ vấn đề. (Bên trên) Khi sử dụng cách biểu diễn dạng hộp, các đối tượng có hướng phân bố dày đặc gây ra hiện tượng đặc trưng răng cưa tại các vùng giao của trường tiếp nhận giữa các đối tượng. (Bên dưới) Với cách biểu diễn bao lồi, phương pháp CFA xử lý tốt các đặc trưng nằm trên lưới tích chập thông thường của các đối tượng có hướng phân bố không đều, giải quyết hiện tượng răng cưa đặc trưng hiệu quả

cưa (feature aliasing), đạt được sự thích ứng đặc trưng tối ưu. Nó cũng đạt được kết quả tốt nhất khi thử nghiệm trên tập dữ liệu DOTA và SKUR110KR.

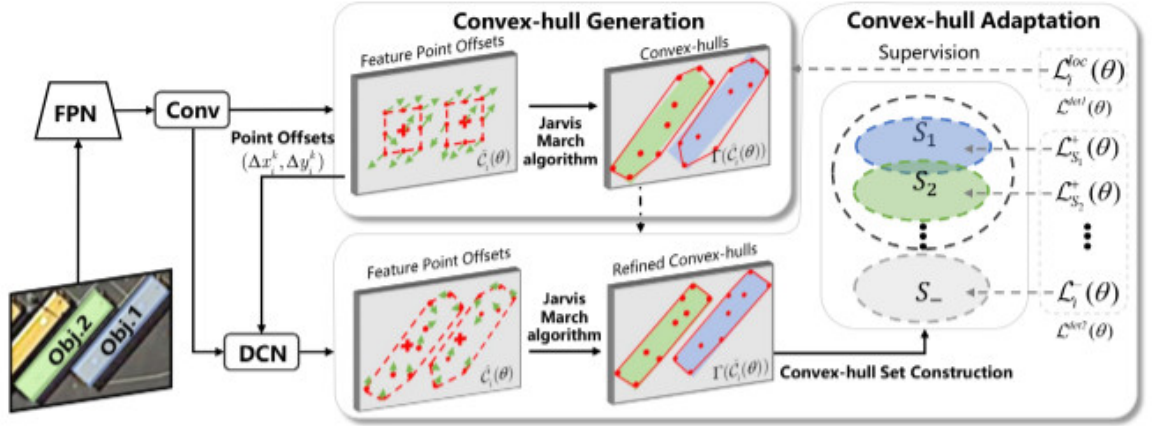
Phương pháp CFA được chia làm 2 giai đoạn thực hiện:

- Giai đoạn 1: tạo tập bao lồi và ước lượng sơ bộ bố cục của bao lồi.
- Giai đoạn 2: chỉnh sửa bao lồi sao cho phù hợp với các đối tượng phân bố dày đặc.

1.2 Xây dựng tập bao lồi đầu tiên

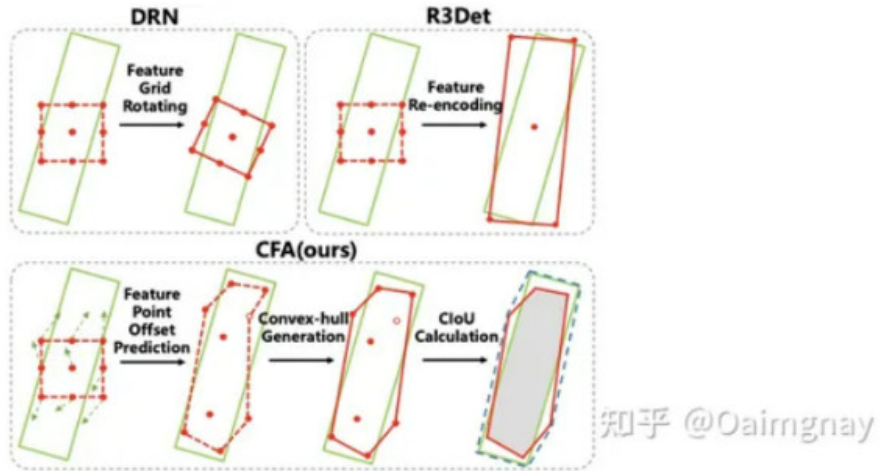
Việc biểu diễn bằng khung hình chữ nhật do bao lồi sinh ra làm giảm khả năng biểu diễn đối tượng. Vì thế phương pháp CFA đã đề xuất biểu diễn phạm vi của đối tượng bằng bao lồi. Mỗi bao lồi là một tập hợp các điểm thoả mãn công thức:

$$C_i = \left\{ (x_i^k, y_i^k) \right\}_i^{k=1 \dots K} \quad (1.1)$$



Hình 1.2: Biểu đồ luồng quy trình thực hiện của bộ phát hiện CFA.

Trong đó: C_i là bao lồi thứ i , (x_i^k, y_i^k) là điểm nằm trên bao lồi thứ k , k là chỉ số của điểm đặc trưng, $K = 9$ tương ứng với 9 điểm được khởi tạo của bao lồi.



Hình 1.3: So sánh biểu diễn hộp có hướng (bên trên) so với biểu diễn bao lồi (ở dưới).

Việc huấn luyện có thể xem như là quá trình dự đoán độ lệch (offset), trong khi hệ số CIoU cần được tối đa hoá để đạt được so khớp tối ưu nhất. Đây là phương pháp sử dụng phép toán tích chập để dự đoán độ lệch: $(\Delta x_i^k, \Delta y_i^k)$ với từng điểm đặc trưng, sau trả về một bản đồ độ bù cho các đặc trưng $O \in$

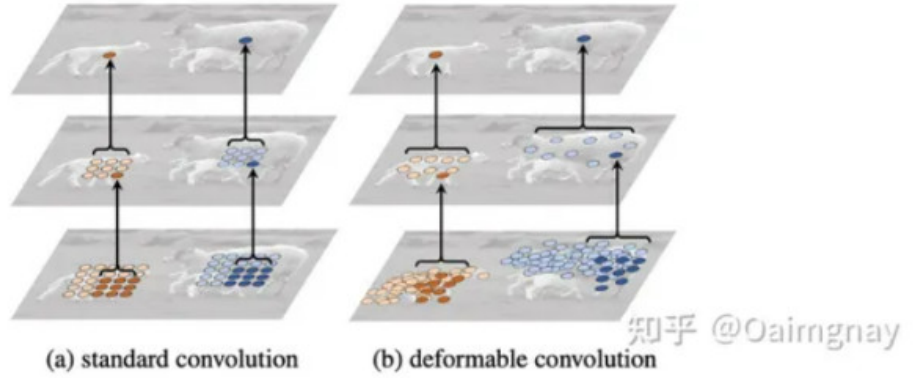
$R^{H \times W \times W}$ (H, W, C lần lượt tương ứng với chiều dài, chiều rộng và số lượng kênh của bản đồ đặc trưng):

$$\hat{C}_i(\theta) \leftarrow \left\{ \left(x_i^k + \Delta x_i^k(\theta), y_i^k + \Delta y_i^k(\theta) \right) \right\}_i^{k=1 \dots K} \quad (1.2)$$

Trong đó: θ biểu thị là các tham số của mạng. Việc dự đoán offset sẽ thực hiện theo công thức trên.

1.3 Tích chập biến dạng

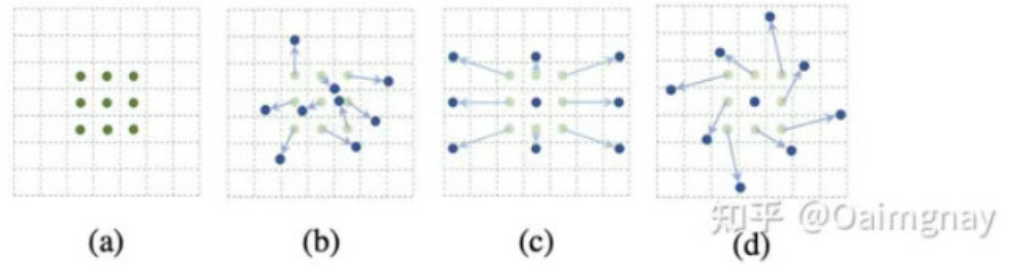
Tích chập biến dạng (Deformable convolution) là dạng tích chập mà vị trí thực hiện tích chập bị biến dạng, không giống tích chập truyền thống là dạng lưới $N \times N$. Ưu điểm của phương pháp này giúp trích xuất các đặc trưng mong muốn được chính xác hơn, lấy mẫu được ở những vị trí đa dạng hơn (Phép tích chập truyền thống chỉ có thể trích xuất các đặc trưng trên một khung hình chữ nhật). (Hình 1.4)



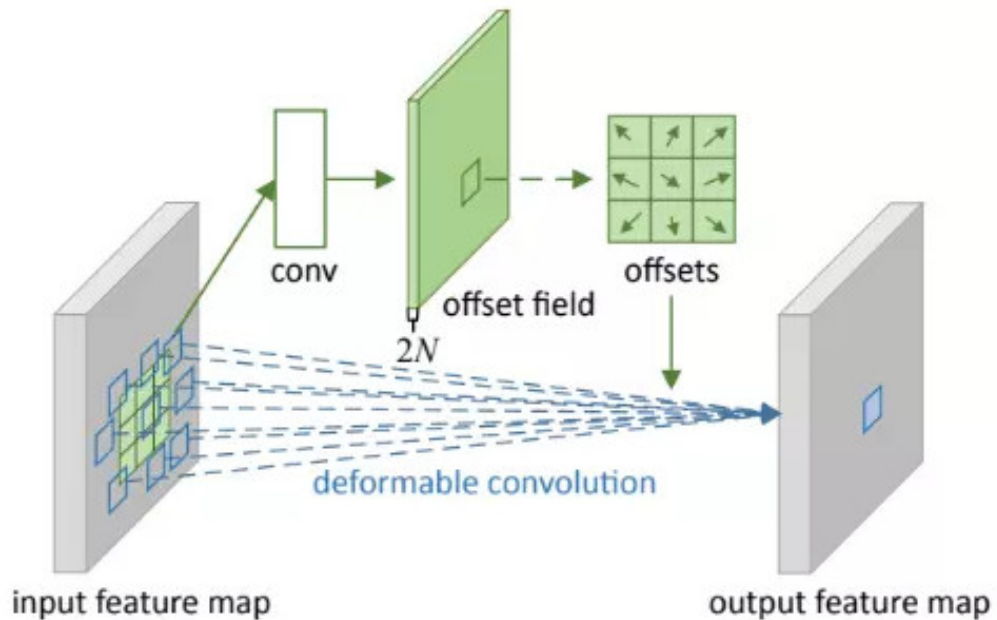
Hình 1.4: So sánh tích chập thông thường và tích chập biến dạng.

Phép tích chập biến dạng thực ra là thêm phần bù cho các điểm tích chập lấy mẫu. (Hình 1.5)

Cho đầu vào là một bản đồ đặc trưng, giả sử phép tích chập là 3×3 . Để học được phần bù, định nghĩa một lớp tích chập 3×3 khác, chiều của đầu ra là kích thước của bản đồ đặc trưng ban đầu, số kênh $= 2N$. (Hình 1.6) Tiếp theo thực



Hình 1.5: a) lấy mẫu tích chập với 9 điểm lấy mẫu. b) để có tích chập biến dạng, thêm độ dịch chuyển vào mỗi điểm lấy mẫu (mũi tên xanh). c) phép biến đổi tỷ lệ. d) phép quay



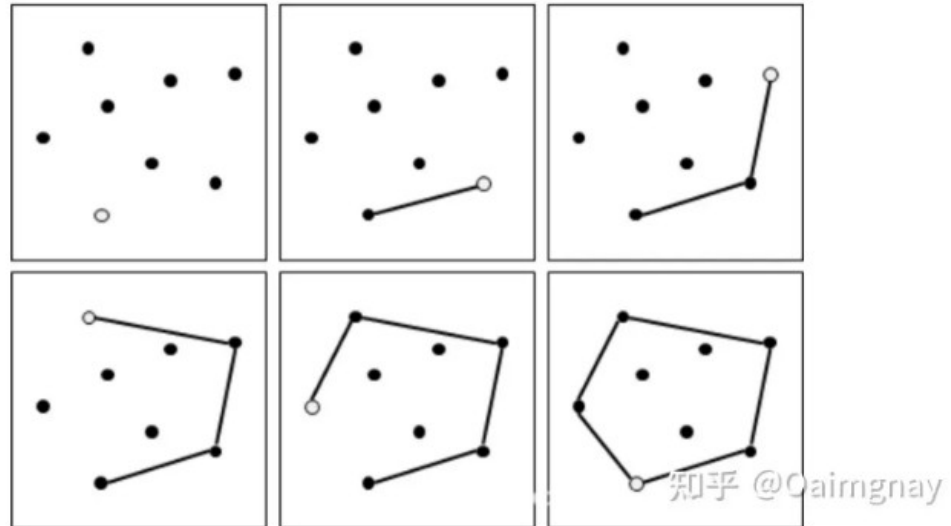
Hình 1.6: Quá trình thực hiện tích chập biến dạng.

hiện tích chập biến dạng, dựa trên độ bù của các phần đã được tính trước đó, sau đó thực hiện phép tích chập như thông thường.

1.4 Thuật toán Jarvis March

Sau khi học được phần bù, việc hoàn thành cập nhật các điểm đặc trưng của bao lồi được thực hiện bởi thuật toán Jarvis March, bao lồi nhỏ nhất phù hợp

với điều kiện sẽ được tạo ra sau mỗi vòng lặp.



Hình 1.7: mô tả các bước thực hiện thuật toán Jarvis March

Tóm tắt các bước thực hiện thuật toán Jarvis March:

- Bước 1: Chọn điểm xuất phát là v_1 nằm trên đường biên. Tìm điểm v_2 tiếp theo sao cho mọi điểm khác trong tập hợp nằm ở phía bên trái đoạn thẳng nối v_1 và v_2 .
- Bước 2: Tìm điểm v_3 tiếp theo trong tập các điểm còn lại, sao cho v_1, v_2, v_3 thỏa mãn góc ngược chiều kim đồng hồ (Counter Clock Wise - CCW). Điều này có nghĩa là v_3 nằm ở bên trái đoạn thẳng nối v_1 và v_2 . Nếu điều kiện này được đáp ứng, tức là v_3 là 1 điểm nằm ở ngoại vi hơn.
- Bước 3: Nếu điều kiện bước trên thỏa mãn, ghi đè giá trị của v_3 lên v_2 . Gán giá trị của v_3 cho v_2 , tức $v_2 = v_3$. Quay lại bước 2 để tìm điểm ngoại vi tiếp theo.
- Bước 4: Lặp lại bước 3 cho đến khi tất cả các điểm đã được duyệt qua, tức là tìm thấy điểm ngoại vi tiếp theo được tìm thấy. Quá trình này tạo ra một chuỗi các điểm trên đường biên của bao lồi.

- Bước 5: Ghi đè giá trị của v2 lên v1. Sau khi tìm thấy điểm ngoại vi tiếp theo, gán giá trị của v2 lên v1, tức $v1 = v2$.
- Bước 6: Lặp lại bước 2 cho đến khi điểm tiếp theo trả về chính là điểm xuất phát. Quá trình này sẽ tiếp tục cho đến khi ta quay trở lại điểm xuất phát ban đầu, hoàn thành toàn bộ quá trình tìm bao lồi.

1.5 Định nghĩa công thức Convex Intersection over Union (CIoU)

Dựa vào mỗi bao lồi dự đoán, có thể tính toán được hàm mất mát vị trí và phân lớp của một đối tượng. Công thức CIoU giữa bao lồi dự đoán thứ i : $C_i(\theta)$ và hộp bao thật sự \mathcal{B}_j của đối tượng thứ j được tính như sau:

$$\text{CIoU}(C_i(\theta), \mathcal{B}_j) = \frac{|C_i(\theta) \cap \mathcal{B}_j|}{|C_i(\theta) \cup \mathcal{B}_j|} - \frac{|\mathcal{R}_j \setminus (C_i(\theta) \cup \mathcal{B}_j)|}{|\mathcal{R}_j|} \quad (1.3)$$

Trong đó: \mathcal{R}_j là hợp của hai đa giác, tức là đa giác nhỏ nhất có thể bao quanh $C_i(\theta)$ và \mathcal{B}_j .

1.6 Hàm mất mát

Theo công thức 1.3, hàm mất mát vị trí CIoU được định nghĩa là:

$$\mathcal{L}_i^{loc}(\theta) = 1 - \text{CIoU}(C_i(\theta), \mathcal{B}_j) \quad (1.4)$$

Cho $f_i^k(\theta)$ là đặc trưng của điểm thứ k , bao lồi đặc trưng $f_i(\theta)$ được tính bởi tổng có trọng số của tất cả các điểm đặc trưng trên bao lồi dự đoán $C_i(\theta)$, tức là bằng công thức: $f_i(\theta) = \sum_k m w_i^k \cdot f_i^k(\theta)$, trong đó, w_i^k biểu thị các trọng số đặc trưng có thể học được từ tích chập biến dạng (DCN). Dựa vào bao lồi đặc trưng, điểm dự đoán $S_i(\theta)$ của bao lồi dự đoán $C_i(\theta)$ được tính bởi phép tích chập, hàm mất mát phân loại của bao lồi dự đoán $C_i(\theta)$ tương ứng với B_j được định nghĩa là:

$$\mathcal{L}_i^{cls}(\theta) = \text{FL}(S_i(\theta), Y_j) \quad (1.5)$$

ở đây Y_j biểu thị là nhãn nhị phân thật sự (ground-truth) và $FL()$ ở đây là hàm mất mát Focal (Focal loss). Kết quả có được là hàm mất mát dành cho bao lỗi dương:

$$\mathcal{L}_i^+(\theta) = \mathcal{L}_i^{cls}(\mathcal{S}_i(\theta), Y_j) + \lambda \mathcal{L}_i^{loc}(\mathcal{C}_i(\theta), \mathcal{B}_j) \quad (1.6)$$

Hàm mất mát 1.6 là tổng của hàm mất mát vị trí 1.4 và hàm mất mát phân loại 1.5. Hàm mất mát dành cho bao lỗi âm là:

$$\mathcal{L}_i^-(\theta) = \mathcal{L}_i^{cls}(\mathcal{S}_i(\theta), Y_j) \quad (1.7)$$

Ngoài ra, trong quá trình huấn luyện (Hình 1.2), vì các bao lỗi được ban đầu chỉ được sinh ra bằng cách tối ưu CIoU, một hàm loss cần được định nghĩa cho việc giám sát:

$$\mathcal{L}^{\det 1}(\theta) = \frac{1}{J} \sum_i \mathbb{I}_{(x_i, y_i)} \mathcal{L}_i^{loc}(\theta) \quad (1.8)$$

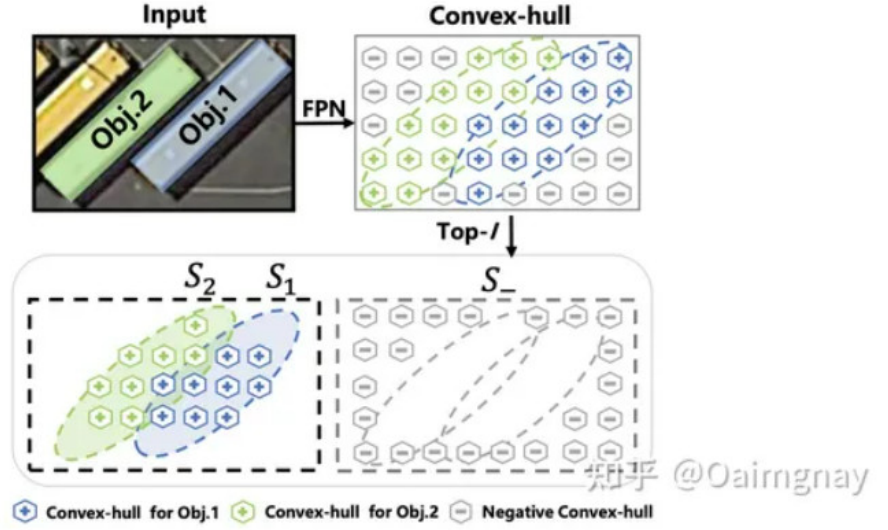
Nhìn chung, trong giai đoạn đầu tiên của việc sinh bao lỗi, hàm mất mát $\mathcal{L}^{\det 1}$ bên trên là hàm cần được quan tâm. Trong giai đoạn 2, hai hàm mất mát phân lớp (1.7 và 1.6) sẽ được sử dụng để phân loại bao lỗi

1.7 Thích ứng bao lỗi

Phương pháp biểu diễn bằng bao lỗi giúp định vị đối tượng ở bất kỳ hình dạng nào, tuy nhiên vẫn có một vấn đề, làm thế nào để định vị một cách chính xác các đối tượng dày đặc, đặc biệt là các đối tượng có đặc trưng răng cưa. Vì vậy bộ phát hiện CFA đã đề xuất một phương pháp thích ứng mới để tinh chỉnh các bao lỗi sinh ra ở giai đoạn 1 để đạt được vị trí chính xác hơn và phân lớp hiệu quả hơn.

1.7.1 Xây dựng tập các bao lỗi

Bộ phát hiện cho xây dựng một tập các bao lỗi với mỗi đối tượng, để một đối tượng mục tiêu có thể khớp với nhiều bao lỗi phù hợp để cùng nhau tối ưu các đặc trưng của các đối tượng dày đặc.



Hình 1.8: xây dựng tập các bao lồi để biểu diễn các đối tượng, đặc biệt với những đối tượng phân bố dày đặc

Với mỗi mục tiêu, ý tưởng xây dựng tập bao lồi tương ứng là: theo như hệ số CIoU giữa bao lồi dự đoán và bao lồi thực tế, chọn I bao lồi đứng đầu là ứng cử viên cho bao lồi dương để xây dựng tập các bao lồi. Ngoài ra cũng có thể xây dựng tập bao lồi sử dụng ngưỡng CIoU xác định bằng thực nghiệm. Các bao lồi còn lại không thuộc vào bất kỳ tập bao lồi nào sẽ được gộp lại thành tập bao lồi âm S .

$$\mathcal{L}_{S_j}^+(\theta) = \frac{1}{|S_j|} \sum_{i \in S_j} \omega_i \mathcal{L}_i^+(\theta) \quad (1.9)$$

Khi nhiều đối tượng tập hợp lại cùng nhau, không phải tất cả các bao lồi nằm trong tập bao lồi đều phù hợp với đối tượng, và các bao lồi có đặc trưng rằng cửa sẽ phải được phân loại thành tập các bao lồi âm. Cùng thời điểm đó, các bao lồi được chia sẻ bởi nhiều đối tượng phải phải có độ tin cậy thấp hơn.

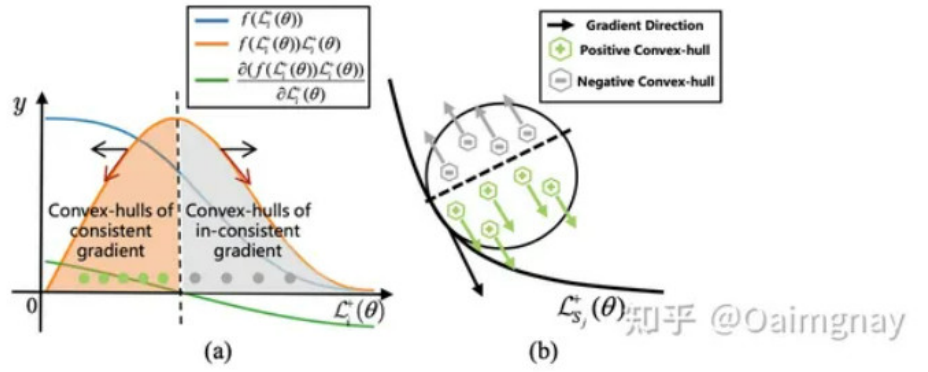
1.7.2 Chiến lược phân đoạn tập các bao lồi

Để giải quyết vấn đề đặc trưng rằng cửa ở các đối tượng dày đặc, bộ phát hiện CFA đề xuất chiến lược phân đoạn tập các bao lồi để đánh giá động các mẫu bao lồi âm và mẫu dương, chuyển đổi trọng số ω_i thành $f(L_i^+(\theta))$. Sau khi

thay thế, được công thức sau:

$$\mathcal{L}_{s_j}^+(\theta) = \frac{1}{|S_j|} \sum_{i \in S_j} f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta) \quad (1.10)$$

Trong đó: f là hàm lỗi đơn điệu giảm phân phối Gaussian: $f(x) = 1.0 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$, có nghĩa là giá trị mất càng nhỏ, độ tin cậy càng cao.



Hình 1.9: Phân chia tập bao lỗi theo hướng dẫn của nguyên tắc nhất quán độ dốc.

Nguyên tắc phân chia tập bao lỗi là nguyên tắc nhất quán độ dốc. Bằng cách lấy đạo hàm của công thức công thức 1.10, ta có được:

$$\frac{\partial \mathcal{L}_{s_j}^+(\theta)}{\partial \theta} = \frac{1}{|S_j|} \sum_{i \in S_j} \frac{\partial (f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta))}{\partial \mathcal{L}_i^+(\theta)} \frac{\partial \mathcal{L}_i^+(\theta)}{\partial \theta} \quad (1.11)$$

Tiêu chí để thực hiện phân đoạn tập các bao lỗi là: đạo hàm của mỗi một bao lỗi dương $\frac{\partial \mathcal{L}_i^+(\theta)}{\partial \theta}$ yêu cầu đạo hàm của toàn bộ tập bao lỗi $\frac{\partial \mathcal{L}_{s_j}^+(\theta)}{\partial \theta}$ là nhất quán. Điều này có nghĩa là: bao lỗi nào có độ dốc không nhất quán được xem là bao lỗi âm, tức là những bao lỗi này sẽ dẫn đến hiện tượng răng cưa đặc trưng. Xem xét công thức 1.11, nếu $\frac{\partial (f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta))}{\partial \mathcal{L}_i^+(\theta)}$ mang giá trị dương, thì bao lỗi \mathcal{L}_i được xếp là bao lỗi dương, hoặc ngược lại bao lỗi sẽ là âm. Xem xét hình 1.9, khi sắp xếp các giá trị mất mát $\frac{\partial \mathcal{L}_i^+(\theta)}{\partial \theta}$ theo thứ tự tăng dần, $f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta)$ (đường màu cam) là một hàm lỗi hướng lên với một cực trị duy nhất, trong khi đường $\frac{\partial (f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta))}{\partial \mathcal{L}_i^+(\theta)}$ (màu xanh lục) chia các bao lỗi thành tập các bao lỗi dương S_j và tập bao lỗi âm S_- .

Cùng thời điểm đó, để xử lý đặc trưng răng cưa, tác giả cũng giới thiệu hệ số chống đặc trưng răng cưa:

$$p_i = \gamma \frac{\text{CIoU}(\mathcal{C}_i, \mathcal{B}_j)}{\sum_{m=1}^M \text{CIoU}(\mathcal{C}_i, \mathcal{B}_j)} \quad (1.12)$$

Hệ số này thể hiện mức độ mà một đối tượng thuộc về một đối tượng duy nhất, khi nó chồng lên M đối tượng khác. γ là hệ số chống đặc trưng răng cưa. Hàm mất mát được cập nhật thành:

$$\mathcal{L}_{s_j}^+(\theta) = \frac{1}{|S_j|} \sum_{i \in S_j} p_i f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta) \quad (1.13)$$

Giai đoạn 2 của quá trình tối ưu được điều khiển bởi hàm mất mát trên tập bao lỗi, hàm này được xác định bằng cách kết hợp hàm mất mát phân loại và hàm mất mát vị trí:

$$\begin{aligned} \mathcal{L}^{\text{det}2}(\theta) = & \frac{1}{J} \sum_{j=1}^J \frac{1}{|S_j|} \sum_{i \in S_j} p_i f(\mathcal{L}_i^+(\theta)) \mathcal{L}_i^+(\theta) \\ & + \frac{1}{|S_-|} \sum_{i \in S_-} \mathcal{L}_i^-(\theta) \end{aligned} \quad (1.14)$$

Hàm mất mát này xem xét sự tương ứng về đặc trưng của nhiều đối tượng, tiến hành phạt các bao lỗi được chia sẻ bởi nhiều đối tượng, và giảm thiểu đặc trưng răng cưa để đạt được thích ứng đặc trưng tối ưu. Cuối cùng hàm mất mát của toàn bộ bộ phát hiện CFA là:

$$L^{\text{det}1}(\theta) + L^{\text{det}2}(\theta) \quad (1.15)$$

Chương 2

Thuật toán tính bao lồi xấp xỉ

giới thiệu chung về thuật toán, cách dùng, có bao nhiêu loại thuật toán

2.1 Outer convex approximation

Cho:

$$X := \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^2 \quad (2.1)$$

Giả sử không mất tính tổng quát rằng

$$x_1, x_2, \dots, x_n \text{ không cùng nằm trên cùng một đường thẳng.} \quad (2.2)$$

Ta viết tất cả các vector thành dạng vector hàng, những vector này sẽ có chuyển vị của chúng được ký hiệu bởi chỉ số trên T , và sử dụng chỉ số trên để chỉ định các thành phần của chúng, ví dụ: $x = (x^1, x^2) \in \mathbb{R}^2$. Cho $x, x' \in \mathbb{R}^2$, chúng ta có:

$$\begin{aligned} [x, x'] &:= \{(1 - \lambda)x + \lambda x' \mid \lambda \in [0, 1]\} \\ (x, x') &:= \{(1 - \lambda)x + \lambda x' \mid \lambda \in (0, 1)\} \end{aligned} \quad (2.3)$$

Cho X thỏa mãn (2.1) - (2.2) và $\delta \geq 0$, trong phần này em muốn tìm một bao lồi xấp xỉ của X , nghĩa là:

$$\text{một đa giác lồi } \mathcal{P}^{outer} \text{ thỏa mãn bao lồi } X \subset \mathcal{P}^{outer} \quad (2.4)$$

sao cho

$$dist_H(\text{conv } X, \mathcal{P}^{outer}) \leq \delta \quad (2.5)$$

P^{outer} được xác định bởi:

$$P^{outer} := \{x \in \mathbb{R}^2 | dx^T \leq \beta_d \text{ với tất cả } d \in D\} \quad (2.6)$$

trong đó $D \subset \mathbb{R}^2$ biểu thị tập các hướng tối đa và $\beta_d \in \mathbb{R}$ biểu thị ngưỡng tương ứng với hướng $d \in D$. Với D cho trước, P^{outer} là bao lồi xấp xỉ phù hợp nhất chứa X nếu:

$$\beta_d := \max_{x \in X} dx^T \text{ với tất cả } d \in D \quad (2.7)$$

Cho P là tập các đỉnh của P^{outer} . Ta bắt đầu quá trình xác định bao lồi xấp xỉ ngoài với hình chữ nhật nhỏ nhất có chứa X , tập X này có chứa cạnh song song với trục tọa độ. Theo công thức (2.5) - (2.6), hình chữ nhật P^{outer} được xác định bởi:

$$D := \{(1, 0), (0, 1), (-1, 0), (0, -1)\} \quad (2.8)$$

và:

$$\begin{aligned} \beta_{(1,0)} &:= \max \{x^1 \mid (x^1, x^2) \in X\}, \\ \beta_{(0,1)} &:= \max \{x^2 \mid (x^1, x^2) \in X\}, \\ \beta_{(-1,0)} &:= \max \{-x^1 \mid (x^1, x^2) \in X\}, \\ \beta_{(0,-1)} &:= \max \{-x^2 \mid (x^1, x^2) \in X\}. \end{aligned} \quad (2.9)$$

Theo công thức (2.2) ta có:

$$\beta_{(-1,0)} < \beta_{(1,0)} \quad \text{và} \quad \beta_{(0,-1)} < \beta_{(0,1)}$$

Vì vậy, P^{outer} là một hình chữ nhật phù hợp với 4 đỉnh phân biệt, có tập đỉnh là:

$$P := \{r_1, r_2, r_3, r_4\} \quad (2.10)$$

Trong đó:

$$\begin{aligned} r_1 &:= (\beta_{(1,0)}, \beta_{(0,1)}), \\ r_2 &:= (\beta_{(-1,0)}, \beta_{(0,1)}), \\ r_3 &:= (\beta_{(-1,0)}, \beta_{(0,-1)}), \\ r_4 &:= (\beta_{(1,0)}, \beta_{(0,-1)}). \end{aligned} \quad (2.11)$$

Trong các bước xấp xỉ tiếp theo, việc xây dựng đa giác P^{outer} lần lượt được cải thiện như sau:

Với 1 đỉnh $p \in P$, cho $p^- \in P$ và $p^+ \in P$ lần lượt là điểm liền trước ngược chiều kim đồng hồ
(2.12)

=====

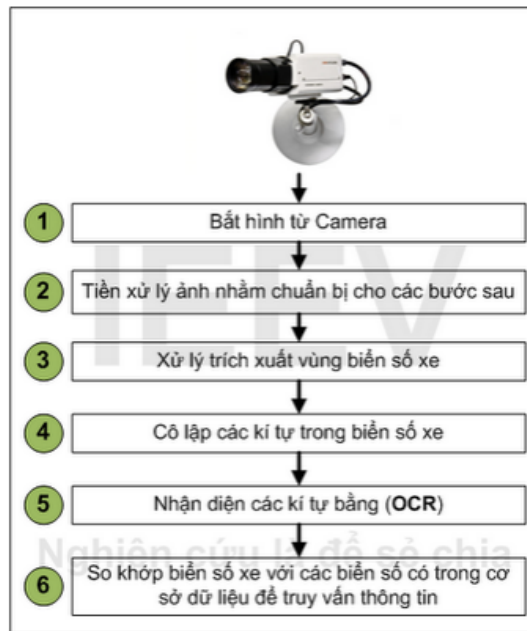
Trong các thuật toán tìm bao lồi thì thuật toán Quickhull [7, 8] được biết đến là một thuật toán hiệu quả có độ phức tạp tính toán trung bình là $O(n \log n)$ nhưng trong trường hợp xấu nhất thì độ phức tạp là $O(n^2)$, trong đó n là số điểm của tập điểm đầu vào của thuật toán. Độ phức tạp của thuật toán Quickhull tương tự như thuật toán Quicksort - một thuật toán sắp xếp hiệu quả dựa trên việc phân chia mảng dữ liệu thành các nhóm phần tử nhỏ hơn và thuật toán này chạy trong thực tế nhanh hơn rất nhiều so với trường hợp xấu nhất. Để giải thích cho điều này, V. Damerow và C. Sohler đã đánh giá thành công số điểm cực biên của một tập hợp theo nghĩa Smoothed analysis [6]. Từ đó có thể đánh giá độ phức tạp tính toán của thuật toán Quickhull theo nghĩa này trong hai trường hợp là:

- Nhiễu ngẫu nhiên của tập điểm được chọn từ phân bố chuẩn $N(0, \delta)$:

$$O\left(n \left(\frac{1}{\delta}\right)^2 \log^2 n\right).$$
- Nhiễu ngẫu nhiên được chọn từ phân bố đều trong hình vuông kích thước 2ϵ :

$$O\left(n \left(\frac{n \log n}{\epsilon}\right)^{2/3}\right).$$

Bài toán tìm bao lồi có ứng dụng rất quan trọng trong lĩnh vực nhận dạng. Các ứng dụng phổ biến của nhận dạng trong thực tế là nhận dạng tiếng nói tự động, phân loại văn bản thành nhiều loại khác nhau (ví dụ những thư điện tử spam/non-spam), nhận dạng tự động các mã bưu điện viết tay trên các bao thư, hay hệ thống nhận dạng mặt người, nhận dạng biển số xe, v.v... Sau đây ta xét một ứng dụng cụ thể của bài toán tìm bao lồi trong nhận dạng biển số xe.



Hình 2.1: Quy trình tự động nhận dạng biển số xe.

2.2 Sơ lược về

Định nghĩa 2.2.1. Phương trình đồng dư đại số bậc n là một đồng dư thức có dạng

$$f(x) = a_0x^n + a_1x^{n-1} + \dots + a_n \equiv 0 \pmod{m} \quad (2.13)$$

trong đó x là ẩn, $a_i \in \mathbb{Z}$ (với $i = 1, 2, \dots, n$) và $a_0 \not\equiv 0 \pmod{m}$.

Chú ý 2.2.2. (i) Giải phương trình (2.13) là tìm tất cả các giá trị nguyên của x thỏa mãn đồng dư thức (2.13). Nếu $x = x_0$ thỏa mãn phương trình (2.13) thì mọi số $x \equiv x_0 \pmod{m}$ đều thỏa mãn (2.13); trong trường hợp này tập hợp $\{x \in \mathbb{Z} \mid x \equiv x_0 \pmod{m}\}$ được gọi là một *nghiệm* của phương trình đồng dư (2.13), kí hiệu là $\overline{x_0}$ hoặc $x \equiv x_0 \pmod{m}$.

(ii) Số nghiệm của phương trình (2.13) là số các phần tử trong một hệ thặng dư đầy đủ theo modulo m mà thỏa mãn (2.13).

(iii) Hai phương trình đồng dư được gọi là tương đương nếu tập hợp các số nguyên thỏa mãn các phương trình đó là trùng nhau.

Ví dụ 2.2.3. Xét phương trình $x^2 \equiv 1 \pmod{5}$.

Giải. Ta thấy trong các số 0, 1, 2, 3, 4 của hệ thặng dư không âm bé nhất theo modulo 5, có hai số 1 và 4 thỏa mãn phương trình đã cho. Vậy phương trình có hai nghiệm là $x \equiv 1 \pmod{5}$ và $x \equiv 4 \pmod{5}$. \square

Ví dụ 2.2.4. Giải phương trình đồng dư $x^4 + 7x + 4 \equiv 0 \pmod{9}$.

Giải. Dễ thấy phương trình $x^4 + 7x + 4 \equiv 0 \pmod{3}$ có nghiệm là $x \equiv 1 \pmod{3}$ (hay $x = 3t + 1$ với $t \in \mathbb{Z}$). Thay x vào phương trình cần giải và bỏ đi những số hạng chia hết cho 9 ta được

$$\begin{aligned} 6t + 3 &\equiv 0 \pmod{9} \\ \Leftrightarrow 2t + 1 &\equiv 0 \pmod{3} \\ \Leftrightarrow t &\equiv 1 \pmod{3} \\ \Leftrightarrow t &= 3k + 1. \end{aligned}$$

Vậy phương trình có nghiệm là $x = 3(3k + 1) + 1$ hay $x \equiv 4 \pmod{9}$. \square

Định nghĩa 2.2.5. Phương trình đồng dư $ax \equiv b \pmod{m}$ được gọi là *phương trình đồng dư tuyến tính* với a, b, m là các số nguyên đã biết. Khi đó $x \equiv x_0 \pmod{m}$ là một nghiệm của phương trình khi và chỉ khi $ax_0 \equiv b \pmod{m}$.

Chương 3

Ứng dụng

Chương này trình bày định nghĩa và các tính chất của thặng dư bậc hai: cách tính thặng dư bằng định nghĩa, cách tính thặng dư thông qua ký hiệu Legendre, cách tính thông qua luật thuận nghịch bậc hai. Sau đó ứng dụng thặng dư bậc hai và luật thuận nghịch bậc hai để tính toán và giải một số bài toán chứng minh, tìm căn nguyên thủy, kiểm tra tính nguyên tố.

3.1 Thặng dư bậc hai và ứng dụng

Thặng dư bậc hai đóng vai trò rất quan trọng trong lý thuyết số. Chẳng hạn, thuật toán phân tích số nguyên ra thừa số nguyên tố. Ngoài ra thặng dư bậc hai cũng ứng dụng lớn trong mật mã cũng như trong các giao thức mã hóa.... Ở đây ta xét ứng dụng liên quan đến giải phương trình đồng dư bậc hai trong lý thuyết và thực hành.

3.1.1 Phương trình đồng dư bậc hai.....

Xét phương trình đồng dư bậc hai theo modulo nguyên tố (theo tài liệu [?]).

$$Ax^2 + Bx + C \equiv 0 \pmod{p}, \quad (3.1)$$

trong đó p nguyên tố lẻ và $A, B, C \in \mathbb{Z}$ với $p \nmid A$. (Nếu $p \mid A$ thì quay về phương trình tuyến tính). Vì p nguyên tố lẻ và $p \nmid A$, nên $p \nmid 4A$. Ta nhân hai vế của phương trình với $4A$ ta được

$$4A(Ax^2 + Bx + C) \equiv 0 \pmod{p}. \quad (3.2)$$

Nhưng ta có

$$\begin{aligned} 4A(Ax^2 + Bx + C) &= 4A^2x^2 + 4ABx + 4AC \\ &= (2Ax + B)^2 - (B^2 - 4AC). \end{aligned}$$

Do đó đồng dư thức (3.2) được viết lại thành

$$(2Ax + B)^2 \equiv (B^2 - 4AC) \pmod{p}$$

nó có dạng

$$y^2 \equiv a \pmod{p} \tag{3.3}$$

trong đó $y = 2Ax + B$ và $a = B^2 - 4AC$.

Nếu phương trình $y^2 \equiv a \pmod{p}$ có nghiệm, thì ta suy ra phương trình $2Ax + B = y$ có nghiệm x modulo p (vì $(2A, p) = 1$). Do đó phương trình (3.1) có nghiệm nếu và chỉ nếu phương trình (3.3) có nghiệm.

Ví dụ sau sẽ minh họa điều này.

Thuật toán 1 THUẬT TOÁN TÌM BAO LỒI DƯỚI $\text{conv}_L(P)$

Đầu vào: Cho $P = \{p_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1, \dots, n\}, n \geq 3$.

Đầu ra: Tập \mathcal{Q} tất cả các mặt dưới của $\text{conv}_L(P)$.

- 1: Tìm cạnh đầu tiên e_0 của $\text{conv}_L(P)$.
 - 2: Xét hàng đợi $\mathcal{Q} := \emptyset$ và tập $\mathcal{E}_L(P) := \emptyset$.
 - 3: Gọi **LFRes**(e, P) để nhận được một mặt dưới F_e qua e_0 . Đẩy các cạnh của F_e trừ e_0 vào trong $\mathcal{E}_L(P)$. Đẩy F_e vào trong tập \mathcal{Q} .
 - 4: **while** ($\mathcal{Q} \neq \emptyset$) **do**
 - 5: Lấy F_e từ phía trên của \mathcal{Q} .
 - 6: $T :=$ tập các cạnh F_e .
 - 7: **for each** $e \in T \cap \mathcal{E}_L(P)$ **do**
 - 8: Gọi **LFRes**(e, P) để nhận được mặt F'_e có chung cạnh e với F_e .
 Đưa vào $\mathcal{E}_L(P)$ tất cả các cạnh $e' \neq e_0$ của F'_e chưa xuất hiện trong $\mathcal{E}_L(P)$ và xoá các cạnh đã xuất hiện trong $\mathcal{E}_L(P)$. Đẩy F'_e vào \mathcal{Q} .
-

Chương 4

Một số kết quả tính toán

Trong nội dung này chúng tôi thử nghiệm số cho thuật toán trong [5] và thuật toán vừa được trình bày ở mục trước để so sánh tốc độ của chúng. Các thuật toán này được thực thi bởi chương trình C và chạy trên PC Core i5 1.6 GHz 3M với 4 GB RAM.

Dưới đây Bảng 4.1 minh họa thời gian chạy (đơn vị tính bằng giây) của thuật toán tính bao lồi dưới được giới thiệu bởi P. T. An và D. T. Giang trong [5] và Thuật toán ?? sử dụng kỹ thuật hạn chế của chúng tôi. Cột cuối cùng liệt kê tỉ lệ tăng tốc của thuật toán của chúng tôi so với thuật toán trong [5]. Dữ liệu đầu vào của các thuật toán là các tập điểm được tạo trên bề mặt một paraboloid (tất cả các điểm đều là đỉnh của bao lồi và bao lồi dưới) có dạng $P = \{p_i = (x_i, y_i, z_i) : x_i, y_i, z_i \in \mathbb{R}, z_i = x_i^2 + y_i^2, i = 1, \dots, n\} \subset \mathbb{R}^3$, trong đó tập $\{(x_i, y_i), x_i, y_i \in R, i = 1, \dots, n\}$ được chọn ngẫu nhiên từ phân bố đều trong hình vuông cỡ 200×200 .

Bảng 4.1: Thời gian chạy tính bao lồi dưới (đơn vị: giây).

Đầu vào	Thuật toán trong [5]	Thuật toán 1	Tỉ số tăng tốc
1.000	0,136	0,061	2,23
2.000	0,454	0,248	1,83
5.000	2,684	1,500	1,79
7.000	7,014	3,129	2,24
11.000	14,321	8,484	1,69
17.000	41,307	24,666	1,67
20.000	63,197	37,332	1,69
22.000	70,866	40,235	1,76
30.000	153,240	96,738	1,58
35.000	200,496	125,490	1,60

Kết luận

Trong quá trình thực tập, em đã có cơ hội làm quen một môi trường làm việc mới. Em đã tính lũy những kinh nghiệm về kiến thức trong công việc cũng như các kinh nghiệm về kỹ năng mềm.

Em được rèn luyện kỹ năng giải quyết công việc theo từng giai đoạn, cố gắng hoàn thành công việc trong thời gian cho phép, mạnh dạn trao đổi và chia sẻ kiến thức. Đồng thời cũng bồi dưỡng thêm rất nhiều kiến thức ngoài các kiến thức đã học trên trường.

Tài liệu tham khảo

Tiếng Việt

- [1] Vũ Thị Gái (2016), *Luật thuận nghịch bậc hai và điểm nguyên*, trường Đại học Khoa học, Đại học Thái Nguyên.
- [2] Nguyễn Tiến Hùng (2016), *Luật thuận nghịch bậc hai và hoán vị*, trường Đại học Khoa học, Đại học Thái Nguyên.
- [3] Hà Huy Khoái (1997), *Nhập môn số học thuật toán*, Nhà xuất bản Khoa học Kỹ thuật.
- [4] Lại Đức Thịnh (1977), *Giáo trình số học*, Nhà xuất bản Giáo dục.

Tiếng Anh

- [5] P. T. An and D. T. Giang (2015), “A direct method for determining the lower convex hull of a finite point set in 3D”, *Advances in Intelligent Systems and Computing, Springer, Proceedings of 3rd International Conference on Computer Science, Applied Mathematics and Applications (ICCSAMA, May 11-13, Metz, France)* **358**, pp. 15–26.
- [6] V. Damerow and C. Sohler (2004), “Extreme points under random noise”, *European Symposium on Algorithms* **3221**, pp. 264–274.
- [7] M. M. David (2002), *Computation geometry*, Department of Computer Science.
- [8] J. O’ Rourke (1998), *Computational geometry in C*, 2nd edition, Cambridge University Press, Cambridge.