**Image Segmentation**

1. **Giới thiệu:**

Image Segmentation là bài toán phân chia một hình ảnh thành nhiều vùng ảnh khác nhau. Mục tiêu của phân đoạn là đơn giản hóa hoặc thay đổi hình ảnh đại diện thành một thứ gì đó có ý nghĩa hơn và dễ phân tích hơn. Phân đoạn hình ảnh thường được sử dụng để định vị các đối tượng và ranh giới (đường thẳng, đường cong, v.v.) trong hình ảnh. Chính xác hơn, phân đoạn hình ảnh là quá trình gán nhãn cho mọi pixel trong một hình ảnh sao cho các pixel có cùng nhãn chia sẻ các đặc điểm nhất định.

* Cách tiếp cận tương đồng (Similarity approach), có nghĩa là phát hiện sự tương đồng giữa các pixel hình ảnh để tạo thành một phân đoạn, dựa trên một ngưỡng. Các thuật toán học máy như phân cụm thường dựa trên kiểu tiếp cận này để phân đoạn một hình ảnh.
* Cách tiếp cận gián đoạn (Discontinuity approach): Cách tiếp cận này dựa trên sự gián đoạn của các giá trị cường độ pixel trong hình ảnh. Các kỹ thuật phát hiện đường, điểm và cạnh sử dụng kiểu tiếp cận gián đoạn để thu được các kết quả phân đoạn trung gian. Kết quả này sau đó có thể được xử lý để cho ra hình ảnh được phân đoạn cuối cùng.

1. **Ứng dụng**

Trong Thị giác máy tính, phân đoạn ảnh là một kỹ thuật quan trọng, giúp giải nhiều bài toán thuộc các lĩnh vực khác nhau như xử lý ảnh y tế, phát hiện và nhận dạng đối tượng, hệ thống camera thông minh… Đây là tiền đề của quá trình xử lý dữ liệu hình ảnh. Kết quả phân đoạn tốt sẽ tạo điều kiện thuận lợi cho các khâu xử lý về sau, tiết kiệm thời gian, gia tăng mức độ chính xác, đồng thời giảm thiểu nguồn lực tính toán.

* Y học: phát hiện khối u, phân đoạn và phát hiện các cơ quan, bộ phận,…
* Giao thông: xe tự hành, nhận diện làn đường, xe cộ, người đi bộ, đèn giao thông, biển báo,…
* Xử lý ảnh vệ tinh: phát hiện sông, núi, tuyến đường,…

1. **Một số kỹ thuật phân đoạn ảnh**

Có 05 kỹ thuật phân đoạn ảnh cơ bản, bao gồm:

* Phân đoạn dựa trên ngưỡng (Threshold Based Segmentation)
* Phân đoạn dựa trên cạnh (Edge Based Segmentation)
* Phân đoạn dựa trên khu vực (Region-Based Segmentation)
* Phân đoạn dựa trên kỹ thuật phân cụm (Clustering Based Segmentation)
* Phân đoạn dựa trên mạng nơron nhân tạo (Artificial Neural Network Based Segmentation)

# Edge based segmentation

Cạnh trong ảnh đánh dấu những vị trí hình ảnh không liên tục về mức xám, màu sắc, kết cấu, v.v. Khi di chuyển từ vùng này sang vùng khác, mức xám có thể thay đổi. Vì vậy, nếu tìm thấy sự gián đoạn đó, ta có thể tìm thấy cạnh. Thực tế, có nhiều toán tử phát hiện cạnh, nhưng hình ảnh thu được là kết quả phân vùng trung gian, và không nên nhầm lẫn với hình ảnh được phân vùng cuối cùng. Để ra được kết quả cuối, cần thực hiện một số bước bổ sung bao gồm: kết hợp các phân vùng cạnh thu được làm một, để giảm số lượng phân vùng và có được một đường viền liền mạch của đối tượng.

Như vậy, có thể thấy, phân vùng cạnh đưa ra một kết quả phân vùng trung gian. Kết quả này sau đó có thể áp dụng theo vùng hoặc bất kỳ kiểu phân đoạn nào khác, nhằm có được hình ảnh được phân vùng cuối.

Shape

Description automatically generated with medium confidenceCác cạnh thường được liên kết với “Độ lớn” và “Hướng”. Một số toán tử phát hiện cạnh cung cấp cả hai yếu tố này, chẳng hạn như Sobel edge operator, canny edge detector, Kirsch edge operator, Prewitt edge operator, Robert’s edge operator,….

# Region-based segmentation

#### **Phát triển khu vực (Region growing method)**

Đối với kỹ thuật này, chúng ta bắt đầu với một số pixel làm pixel hạt giống và sau đó kiểm tra các pixel liền kề. Nếu các pixel liền kề tuân theo các quy tắc được xác định trước, thì pixel đó sẽ được thêm vào vùng của pixel gốc và quá trình sẽ tiếp tục cho đến khi không còn điểm tương đồng nào. Phương pháp này thực hiện theo cách tiếp cận từ dưới lên. Trong trường hợp khu vực đang phát triển, quy tắc ưu tiên có thể được đặt làm ngưỡng.

VD: Thresh < 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 3 | 6 | 3 | 2 | 1 |
| 2 | 1 | 1 | 6 | 2 | 5 |
| 7 | 1 | 9 | 8 | 7 | 7 |
| 1 | 3 | 2 | 7 | 8 | 1 |
| 6 | 2 | 9 | 6 | 1 | 2 |
| 3 | 1 | 1 | 2 | 3 | 4 |

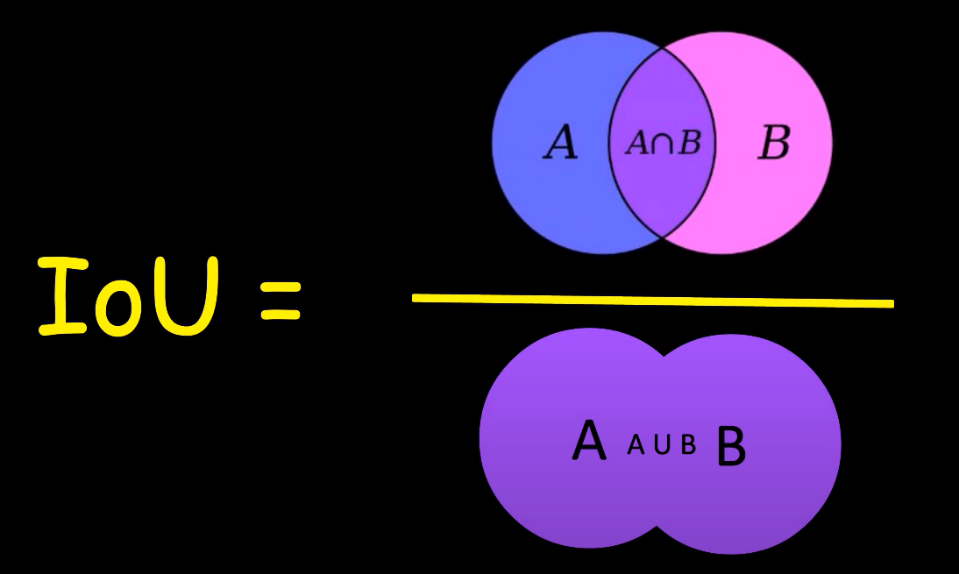
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 3 | 6 | 3 | 2 | 1 |
| 2 | 1 | 1 | 6 | 2 | 5 |
| 7 | 1 | 9 | 8 | 7 | 7 |
| 1 | 3 | 2 | 7 | 8 | 1 |
| 6 | 2 | 9 | 6 | 1 | 2 |
| 3 | 1 | 1 | 2 | 3 | 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 3 | 6 | 3 | 2 | 1 |
| 2 | 1 | 1 | 6 | 2 | 5 |
| 7 | 1 | 9 | 8 | 7 | 7 |
| 1 | 3 | 2 | 7 | 8 | 1 |
| 6 | 2 | 9 | 6 | 1 | 2 |
| 3 | 1 | 1 | 2 | 3 | 4 |

1. **Độ đo IoU**

IoU (Intersection-Over-Union) là một thước đo đánh giá phổ biến để phân đoạn hình ảnh.

IoU = Diện tích phần GIAO /Diện tích phần HỢP



1. **Các bước thực nghiệm**
   1. **Edge base**

Trong phần báo cáo các bước xử lí để đưa ra một phân vùng ảnh cuối cùng như sau:

* Tiền xử lý: làm mờ ảnh bằng Bộ lọc gaussian được khởi tạo thông qua hàm cv2.GaussianBlur()
* Sử dụng toán tử sobel với ddepth = cv2.CV\_64F thông qua hàm sobel với hai trục x và y
* Ảnh sau khi tìm cạnh chưa thể hiện rõ các cạnh do vẫn còn nằm trong những mức xám khác nhau. Để làm rõ, ta sử dụng ngưỡng (threshold) để chia ảnh thành 2 màu trắng và đen, đối với các pixel có mức xám bé hơn ngưỡng ta cho thành màu đen và ngược lại là màu trắng
* Sau khi có ảnh trắng đen, để có được ảnh phân vùng cuối, ta sử dụng hàm cv2.findcontour() để tìm các pixel liên tiếp với nhau. Sau đó, tìm contour có diện tích bao quát vật thể cần được mask.
* Để mask các contour, sử dụng cv2.drawcontour(), ảnh đầu ra là ảnh phân vùng vật thể cần tìm
* Đối với mỗi ảnh có chủ đề, màu sắc, độ tương phản khác nhau, cần chỉnh các thông số phù hợp

1. **Kết quả thực nghiệm**

Sau khi thực nghiệm trên nhiều ảnh, nhóm em đưa ra được một số nhận xét

|  |  |
| --- | --- |
| **Edge-based** | **Region-based (region growing)** |
| Phù hợp với ảnh mà vật thể cần phân đoạn có cạnh rõ ràng, background ít chi tiết cạnh.  Vật thể càng ít chi tiết càng tốt, có các cạnh quan trọng là viền ngoài, | Phù hợp với ảnh mà vật thể cần phân đoạn có màu gần giống nhau và tách biệt với màu của background.  Vật thể nên có màu gần giống nhau, và vùng màu liên tục không bị chia cắt bởi các màu khác. |
| Tốt hơn region-based khi vật thể cần phân đoạn có nhiều màu khác nhau, nhiều màu sắc gây nhiễu. | Tốt hơn edge với các ảnh nhiễu, phát hiện cạnh khó |
|  |  |

**Đọc thêm**

Khi nhắc đến việc làm mờ hình ảnh, bạn sẽ nghĩ ngay đến việc tại mình phải làm điều đó? Ảnh đang rõ vậy làm mờ chi cho khó nhìn ta?!

Thực ra trong xử lý ảnh, phép làm mờ ảnh được dùng rất nhiều và có nhiều vai trò quan trọng. Hiệu ứng làm mờ mang lại (áp dụng trong các trường hợp):

* Giảm nhiễu (noise) trong ảnh
* Làm trơn ảnh (smooth). Việc làm trơn ảnh sẽ giảm sắc nét của cạnh, thay vào đó, vùng trơn sẽ lan ra

Trong các bài báo về xử lý ảnh, ở bước tiền xử lý họ hay dùng 2 phép biến đổi sau:

* Cân bằng sáng
* Làm mờ