**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 3:** **Xây dựng hệ thống bằng việc sử dụng các kỹ thuật phân đoạn ảnh như Watershed, GrabCut để phân đoạn các đối tượng trong ảnh**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210486 | Nguyễn Nhật Minh | DCCNTT12.10.2 |
| **2** | 20210556 | Nguyễn Văn Trọng | DCCNTT12.10.2 |
| **3** | 20210586 | Bùi Khắc Trung | DCCNTT12.10.2 |
| **4** | 20210390 | Đỗ Lâm Đức Minh | DCCNTT12.10.2 |
| **5** | 20210542 | Nguyễn Đan Huy | DCCNTT12.10.2 |

**Bắc Ninh, năm 2024**

**Bắc Ninh, năm 2024**

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 03**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ:**  **XATGMT.03.K12.02.LH.C04.1\_LT** | |
| **GV Lương Thị Hồng Lan**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Nhật Minh** | **Nguyễn Văn Trọng** | **Bùi Khắc Trung** | **Đỗ Lâm Đức Minh** | **Nguyễn Đan Huy** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20210486 | 20210556 | 20210586 | 20210390 | 20210542 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |  |

# LỜI MỞ ĐẦU

Trong bối cảnh sự phát triển mạnh mẽ của công nghệ máy tính và trí tuệ nhân tạo, việc ứng dụng các kỹ thuật xử lý ảnh để phân tích và nhận diện các đối tượng trong hình ảnh đang ngày càng trở nên quan trọng. Phân đoạn ảnh (Image Segmentation) là một trong những kỹ thuật cốt lõi trong lĩnh vực xử lý ảnh và thị giác máy tính, nhằm tách biệt các phần khác nhau trong ảnh, từ đó dễ dàng phân tích và nhận diện các đối tượng hoặc đặc điểm quan trọng.

Đề tài "Xây dựng hệ thống phân đoạn ảnh bằng các kỹ thuật Watershed và GrabCut" hướng đến việc nghiên cứu và triển khai hai phương pháp phân đoạn ảnh phổ biến, mỗi phương pháp có những ưu điểm và ứng dụng riêng biệt. Watershed là một thuật toán phân vùng ảnh mạnh mẽ, giúp tách biệt các đối tượng trong ảnh bằng cách mô phỏng quá trình chảy nước trong một khu vực địa lý. Phương pháp này rất hiệu quả trong việc phát hiện các biên và phân chia các đối tượng liên tiếp trong ảnh. Trong khi đó, GrabCut là một kỹ thuật phân đoạn ảnh dựa trên việc sử dụng mô hình nền và đối tượng để tách phần cần thiết khỏi nền, đặc biệt hiệu quả khi xử lý ảnh có nền phức tạp.

Mục tiêu của đề tài là xây dựng một hệ thống có thể sử dụng hai kỹ thuật này để phân đoạn ảnh, hỗ trợ việc nhận diện các đối tượng, tạo ra các ứng dụng thực tế trong nhiều lĩnh vực như nhận diện hình ảnh, y tế, và robot tự động. Hệ thống sẽ cung cấp giao diện người dùng thân thiện, cho phép người dùng dễ dàng chọn ảnh và áp dụng các thuật toán phân đoạn, từ đó có thể quan sát kết quả phân vùng rõ ràng và chính xác.

Việc triển khai đề tài này không chỉ giúp sinh viên áp dụng các kiến thức lý thuyết vào thực tế mà còn mở ra cơ hội nghiên cứu và phát triển các công cụ xử lý ảnh tiên tiến. Chúng tôi hy vọng rằng, qua quá trình nghiên cứu và phát triển hệ thống phân đoạn ảnh, chúng tôi sẽ góp phần mang lại những giải pháp hữu ích cho việc phân tích và xử lý ảnh trong các lĩnh vực ứng dụng cụ thể.

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến cô Lương Thị Hồng Lan, giảng viên môn Xử lý ảnh và Thị giác máy tính, đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài "Xây dựng hệ thống phân đoạn ảnh bằng các kỹ thuật Watershed và GrabCut". Sự chỉ dẫn chi tiết, những góp ý quý báu và sự động viên của cô đã giúp chúng em hoàn thành bài tập lớn này một cách hiệu quả và đạt được nhiều kiến thức thực tiễn bổ ích.

Chúng em cũng xin bày tỏ lòng biết ơn đến Ban Giám hiệu và các thầy cô trong Khoa Công nghệ Thông tin đã tạo điều kiện thuận lợi và cung cấp những tài liệu, nguồn học liệu cần thiết trong suốt quá trình học tập và nghiên cứu.

Cuối cùng, chúng em xin cảm ơn các bạn trong nhóm đã làm việc hết mình, đóng góp ý tưởng và hỗ trợ lẫn nhau để hoàn thành tốt bài tập lớn này. Sự hợp tác và tinh thần làm việc nhóm đã giúp chúng em vượt qua mọi khó khăn và đạt được kết quả như mong đợi.

Chúng em hy vọng rằng kết quả của dự án này sẽ không chỉ đáp ứng được yêu cầu học thuật mà còn mang lại những giá trị thực tiễn, đóng góp vào việc giải quyết các bài toán phân đoạn ảnh và củng cố nền tảng kiến thức xử lý ảnh của chúng em.

**Mục lục**

[LỜI MỞ ĐẦU 1](#_Toc184569805)

[LỜI CẢM ƠN 2](#_Toc184569806)

[Chương I: Cơ sở lý thuyết 4](#_Toc184569807)

[1.1 Nhận dạng đối tượng 4](#_Toc184569808)

[1.1.1 Mô tả bài toán 4](#_Toc184569809)

[1.1.2 Khó khăn 4](#_Toc184569810)

[1.1.3 Ứng dụng 5](#_Toc184569811)

[1.2 Các phương pháp áp dụng cho bài toán nhận dạng 5](#_Toc184569812)

[1.2.1 Kỹ thuật Watershed 5](#_Toc184569813)

[1.2.2 Kỹ thuật GrabCut 6](#_Toc184569814)

[1.3 Ngôn ngữ lập trình và các thư viện 7](#_Toc184569815)

[1.3.1 Ngôn ngữ lập trình Python 7](#_Toc184569816)

[1.3.2 Các thư viện sử dụng 8](#_Toc184569817)

[Chương II: Xây dựng hệ thống 12](#_Toc184569818)

[2.1 Sơ đồ hoạt động 12](#_Toc184569819)

[2.2 Xây dựng hệ thống 13](#_Toc184569820)

[2.1.1 Chức năng chính 13](#_Toc184569821)

[2.1.2 Yêu cầu kỹ thuật 14](#_Toc184569822)

[Chương III: Thực nghiệm chương trình 17](#_Toc184569823)

[3.1 Nguồn gốc dữ liệu 17](#_Toc184569824)

[3.2 độ đo đánh giá 17](#_Toc184569825)

[3.3 Kết quả thực nghiệm 18](#_Toc184569826)

[Tư liệu tham khảo 21](#_Toc184569827)

# Chương I: Cơ sở lý thuyết

## Nhận dạng đối tượng

### 1.1.1 Mô tả bài toán

Nhận dạng đối tượng là một trong những bài toán quan trọng nhất của lĩnh vực xử lý ảnh và thị giác máy tính. Mục tiêu chính của nhận dạng đối tượng là xác định các khu vực hoặc vùng trong ảnh thuộc về các đối tượng cụ thể, đồng thời phân tách những đối tượng này khỏi nền hoặc các đối tượng khác. Trong bài toán này, việc nhận dạng đối tượng không chỉ dừng lại ở phát hiện mà còn bao gồm cả quá trình phân đoạn ảnh để xác định chính xác ranh giới của từng đối tượng.

Phân đoạn đối tượng đóng vai trò cốt lõi trong nhận dạng, vì nó giúp hệ thống xác định rõ từng phần của ảnh thuộc về đối tượng nào. Ví dụ, trong ảnh y tế, phân đoạn có thể giúp tách các khối u ra khỏi cơ quan xung quanh; trong thương mại, nó giúp xóa nền để làm nổi bật sản phẩm. Bài toán phân đoạn này thường được giải quyết bằng cách sử dụng các thuật toán chuyên biệt, trong đó Watershed và GrabCut là hai kỹ thuật phổ biến. Watershed hoạt động dựa trên việc phân tích gradient của ảnh để tìm ranh giới giữa các vùng, trong khi GrabCut sử dụng mô hình đồ thị và tối ưu hóa năng lượng để phân tách đối tượng khỏi nền một cách bán tự động. Hai phương pháp này bổ trợ cho nhau và được kỳ vọng sẽ mang lại hiệu quả cao khi áp dụng trên các ảnh có cấu trúc phức tạp.

Nhận dạng đối tượng thường được thực hiện qua ba bước chính: tiền xử lý ảnh, phân đoạn ảnh, và nhận dạng. Tiền xử lý giúp làm sạch và chuẩn bị dữ liệu ảnh đầu vào, loại bỏ nhiễu hoặc tăng cường các đặc trưng quan trọng. Phân đoạn là bước tập trung vào việc tách biệt các vùng trong ảnh dựa trên đặc trưng màu sắc, kết cấu, hoặc hình học. Cuối cùng, nhận dạng là quá trình sử dụng các thuật toán hoặc mô hình để gán nhãn cho các vùng đã được phân đoạn, xác định chúng là gì hoặc thuộc nhóm nào.

### 1.1.2 Khó khăn

Bài toán nhận dạng đối tượng, đặc biệt với hai kỹ thuật Watershed và GrabCut, không phải lúc nào cũng dễ dàng triển khai hiệu quả. Một trong những khó khăn đầu tiên là chất lượng ảnh đầu vào. Ảnh có thể bị nhiễu do điều kiện chụp, ánh sáng không đủ, hoặc độ phân giải thấp, khiến các đặc trưng quan trọng trở nên khó phát hiện. Đối với Watershed, phương pháp này nhạy cảm với các thay đổi nhỏ trong gradient và dễ dẫn đến hiện tượng over-segmentation, nơi ảnh bị phân đoạn thành quá nhiều vùng không cần thiết. Trong khi đó, GrabCut yêu cầu một vùng chọn ban đầu (bounding box) chính xác, nếu không, quá trình tối ưu hóa có thể dẫn đến kết quả sai lệch.

Một khó khăn khác là sự phức tạp trong việc xử lý các đối tượng chồng lấn hoặc có đặc trưng tương tự nhau. Khi các đối tượng có màu sắc, kết cấu hoặc hình dạng gần giống nhau, hệ thống rất dễ nhầm lẫn giữa chúng. Bên cạnh đó, trong ảnh có quá nhiều đối tượng hoặc đối tượng bị che khuất một phần, việc xác định ranh giới trở nên cực kỳ thách thức.

Hiệu năng tính toán cũng là một vấn đề lớn, đặc biệt khi làm việc với ảnh có độ phân giải cao hoặc khi áp dụng thuật toán trên một tập dữ liệu lớn. Cả Watershed và GrabCut đều có chi phí tính toán đáng kể, vì vậy cần tối ưu hóa thuật toán hoặc tận dụng phần cứng như GPU để đảm bảo hiệu suất.

Ngoài ra, việc đánh giá hiệu quả của phân đoạn cũng đặt ra nhiều thách thức. Các thước đo như Precision, Recall, hoặc IoU (Intersection over Union) tuy hữu ích nhưng không phải lúc nào cũng phản ánh đầy đủ chất lượng của kết quả. Đặc biệt, khi các ranh giới phân đoạn có sự sai lệch nhỏ nhưng quan trọng, các thước đo này có thể không đủ nhạy để phát hiện.

### 1.1.3 Ứng dụng

Nhận dạng đối tượng và phân đoạn ảnh có rất nhiều ứng dụng trong thực tế và nghiên cứu. Trong y tế, các kỹ thuật phân đoạn được sử dụng rộng rãi để phân tích hình ảnh từ các thiết bị chẩn đoán như MRI, CT scan, hoặc siêu âm. Chúng giúp xác định các khu vực bất thường như khối u hoặc tổn thương, từ đó hỗ trợ bác sĩ trong quá trình chẩn đoán và điều trị.

Trong lĩnh vực giao thông, nhận dạng đối tượng giúp phát hiện và phân loại các phương tiện giao thông, hỗ trợ quản lý lưu lượng xe cộ hoặc giám sát vi phạm giao thông. Hệ thống camera giám sát thông minh thường sử dụng các kỹ thuật này để nhận diện biển số xe hoặc phát hiện người đi bộ.

Trong thương mại điện tử, phân đoạn ảnh được sử dụng để xóa nền sản phẩm, tạo ra các hình ảnh hấp dẫn hơn để trưng bày trên website. Các nền tảng lớn như Amazon hoặc Shopee đều áp dụng công nghệ này để cải thiện trải nghiệm mua sắm trực tuyến của khách hàng.

Ngoài ra, các kỹ thuật phân đoạn cũng được sử dụng trong nông nghiệp để tự động hóa các quy trình như đếm số lượng cây trồng, xác định bệnh trên lá cây, hoặc phân loại các loại quả. Đây là một lĩnh vực hứa hẹn khi kết hợp xử lý ảnh với trí tuệ nhân tạo để tối ưu hóa sản xuất và giảm thiểu rủi ro.

Trong nghiên cứu học thuật, phân đoạn ảnh là bước tiền đề để phát triển các mô hình học sâu tiên tiến như Mask R-CNN, U-Net, hoặc DeepLab. Những mô hình này dựa trên dữ liệu phân đoạn để học cách nhận dạng đối tượng với độ chính xác cao hơn, từ đó mở rộng ứng dụng ra nhiều lĩnh vực mới.

## 1.2 Các phương pháp áp dụng cho bài toán nhận dạng

### 1.2.1 Kỹ thuật Watershed

Watershed là một phương pháp phân đoạn dựa trên phân tích gradient của ảnh. Tư tưởng chính là xem ảnh đầu vào như một địa hình, trong đó độ sáng của mỗi pixel đại diện cho độ cao. Phương pháp này tìm các "đỉnh núi" (đối tượng) và "thung lũng" (ranh giới giữa các đối tượng) trong ảnh. Khi thực hiện thuật toán Watershed, các điểm thấp (thung lũng) sẽ được lấp đầy dần như nước chảy ngược lên đỉnh núi. Các ranh giới phân tách được hình thành khi các "hồ" nước từ các thung lũng khác nhau gặp nhau. Đây là cách xác định ranh giới giữa các đối tượng.

Hình mô tả kỹ thuật

A close-up of a filter

Description automatically generated

*Hình 1 :mô tả kĩ thuật Watershed.*

Ưu điểm:

* Hiệu quả với các ảnh có độ tương phản cao giữa đối tượng và nền.
* Tự động tìm kiếm ranh giới, không cần can thiệp thủ công nhiều.
* Kết hợp tốt với các phương pháp tiền xử lý như làm mờ Gaussian hoặc lọc Sobel để cải thiện độ chính xác.

Nhược điểm

* Nhạy cảm với nhiễu và dễ dẫn đến hiện tượng over-segmentation, đặc biệt khi gradient của ảnh không rõ ràng.
* Đòi hỏi công đoạn tiền xử lý tốt để cải thiện kết quả phân đoạn.

### 1.2.2 Kỹ thuật GrabCut

Tư tưởng kỹ thuật:

GrabCut là một phương pháp phân đoạn bán tự động dựa trên lý thuyết đồ thị. Ban đầu, người dùng cần cung cấp một vùng chọn ban đầu (bounding box) bao quanh đối tượng cần phân đoạn. Sau đó, thuật toán xây dựng một mô hình Gaussian Mixture Model (GMM) để phân biệt giữa đối tượng và nền. Phương pháp này sử dụng tối ưu hóa năng lượng trên đồ thị (Graph Cuts) để xác định ranh giới tốt nhất giữa đối tượng và nền. Các pixel được gán nhãn dựa trên sự khác biệt về màu sắc và độ tương đồng không gian.

Hình mô tả kỹ thuật



*Hình 1.2 :mô tả kĩ thuật Grabcut.*

Ưu điểm:

* Hiệu quả với các đối tượng phức tạp nhờ sử dụng mô hình thống kê (GMM).
* Tương đối chính xác ngay cả khi nền có độ tương phản thấp.
* Dễ sử dụng với sự hỗ trợ ban đầu từ người dùng để tăng tính chính xác.

Nhược điểm:

* Phụ thuộc vào bounding box ban đầu: nếu chọn sai, kết quả sẽ kém chính xác.
* Tốn tài nguyên tính toán hơn Watershed, đặc biệt khi xử lý ảnh lớn.

## 1.3 Ngôn ngữ lập trình và các thư viện

### 1.3.1 Ngôn ngữ lập trình Python

Để triển khai bài toán phân đoạn và nhận dạng đối tượng, Python được chọn làm ngôn ngữ lập trình chính. Python là một ngôn ngữ mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong lĩnh vực xử lý ảnh và thị giác máy tính nhờ cú pháp đơn giản, dễ học, cùng với hệ sinh thái phong phú các thư viện hỗ trợ. Python cũng có khả năng tích hợp tốt với các framework học sâu và các công cụ tính toán hiệu năng cao, giúp tăng tốc độ và tối ưu hóa các thuật toán phức tạp.

Một số đặc điểm nổi bật của Python trong bài toán xử lý ảnh bao gồm:

* Khả năng tương thích cao: Python hỗ trợ nhiều thư viện mạnh mẽ, giúp giải quyết toàn bộ pipeline của bài toán từ tiền xử lý, phân đoạn đến hiển thị kết quả.
* Cộng đồng lớn: Nhiều tài liệu hướng dẫn, ví dụ và cộng đồng hỗ trợ rộng lớn giúp người học dễ dàng tiếp cận và giải quyết các vấn đề phát sinh.
* Dễ tích hợp: Python có thể dễ dàng tích hợp với các ngôn ngữ khác như C++, Java hoặc các công cụ như TensorFlow, PyTorch.

### 1.3.2 Các thư viện sử dụng

Để giải quyết bài toán phân đoạn đối tượng bằng các kỹ thuật như Watershed và GrabCut, một số thư viện chính của Python sẽ được sử dụng như sau:

import cv2  
import numpy as np  
from tkinter import Tk, Button, Label, filedialog  
from PIL import Image, ImageTk  
from matplotlib import pyplot as plt

**Thư viện OpenCV**

* OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở mạnh mẽ, chuyên biệt cho xử lý ảnh và thị giác máy tính. Thư viện này cung cấp các công cụ cần thiết để thực hiện cả hai kỹ thuật Watershed và GrabCut.

Tính năng:

* Cung cấp các hàm tiền xử lý như làm mờ ảnh, phát hiện cạnh, chuyển đổi màu sắc.
* Hỗ trợ thực hiện Watershed và GrabCut với các hàm tích hợp sẵn (cv2.watershed, cv2.grabCut).
* Cung cấp khả năng trực quan hóa kết quả một cách nhanh chóng.

Lý do sử dụng:

* OpenCV có hiệu suất cao, dễ tích hợp và cung cấp sẵn nhiều công cụ xử lý ảnh cần thiết, giúp giảm thiểu thời gian lập trình.

**Thư viện NumPy**

NumPy là thư viện cơ bản cho các thao tác tính toán trên mảng (arrays) trong Python. Trong bài toán này, NumPy hỗ trợ xử lý dữ liệu ảnh dưới dạng ma trận số, từ đó dễ dàng thực hiện các phép toán học liên quan đến gradient, ngưỡng và phân tích pixel.

Tính năng:

* Hỗ trợ các phép toán nhanh chóng trên mảng dữ liệu lớn.
* Dễ dàng thao tác dữ liệu ảnh với độ chính xác cao.

Lý do sử dụng:

* NumPy là công cụ thiết yếu để xử lý và chuẩn bị dữ liệu đầu vào cho các thuật toán phân đoạn.

**Thư viện Matplotlib**

Matplotlib là thư viện chuyên dùng để trực quan hóa dữ liệu. Trong bài toán này, Matplotlib được sử dụng để hiển thị kết quả phân đoạn, giúp kiểm tra và đánh giá chất lượng phân đoạn một cách dễ dàng.

Tính năng:

* Hiển thị ảnh gốc, ảnh qua các bước xử lý và kết quả cuối cùng.
* Hỗ trợ biểu diễn histogram để phân tích ảnh.

Lý do sử dụng:

* Thư viện này cung cấp các công cụ trực quan hóa mạnh mẽ, dễ sử dụng để phân tích và trình bày dữ liệu.

**Thư viện Scikit-Image**

Scikit-Image là một thư viện xử lý ảnh chuyên sâu, được xây dựng dựa trên NumPy. Nó cung cấp các thuật toán phân đoạn, phát hiện cạnh và tiền xử lý ảnh tiên tiến.

Tính năng:

* Hỗ trợ nhiều phương pháp phân đoạn ngoài Watershed, như Region Growing, Thresholding.
* Cung cấp các công cụ để đánh giá và cải thiện chất lượng phân đoạn.

Lý do sử dụng:

* Scikit-Image mang lại sự linh hoạt cao trong việc thử nghiệm và so sánh nhiều kỹ thuật khác nhau.

**Thư viện PIL (Pillow)**

**Pillow** là thư viện xử lý ảnh mạnh mẽ, được sử dụng để làm việc với nhiều định dạng ảnh khác nhau và tương thích tốt với các GUI frameworks như Tkinter.

Tính năng:

* Dùng Image.fromarray để chuyển ảnh từ NumPy array (dạng mà OpenCV sử dụng) sang dạng ảnh của Pillow.
* Việc này cần thiết vì Tkinter không trực tiếp hỗ trợ NumPy array.

Lý do sử dụng:

* Pillow được sử dụng để đảm bảo tính tương thích và linh hoạt trong việc xử lý và hiển thị ảnh trong giao diện Tkinter. Trước tiên, Tkinter không thể hiển thị trực tiếp ảnh dưới dạng NumPy array – định dạng mà OpenCV sử dụng. Do đó, Pillow đóng vai trò trung gian, chuyển đổi ảnh từ NumPy array sang định dạng mà Tkinter hỗ trợ thông qua đối tượng ImageTk.PhotoImage.
* Bên cạnh đó, OpenCV sử dụng không gian màu BGR, trong khi Tkinter yêu cầu ảnh ở không gian màu RGB. Pillow hỗ trợ chuyển đổi này dễ dàng bằng hàm Image.fromarray mà không làm thay đổi cấu trúc dữ liệu gốc, đảm bảo ảnh hiển thị chính xác về màu sắc.
* Ngoài ra, Pillow hỗ trợ xử lý nhiều định dạng ảnh khác nhau như .jpg, .jpeg, .png, giúp ứng dụng có thể làm việc linh hoạt với các loại ảnh khác nhau mà người dùng cung cấp.

**Thư viện Tkinter:**

**Tkinter** là thư viện GUI tiêu chuẩn của Python, được sử dụng để xây dựng các ứng dụng giao diện người dùng đơn giản, trực quan.

Tính năng:

* Dùng Tk() để tạo cửa sổ chính của ứng dụng.
* Các nút bấm như "Chọn Ảnh", "Watershed", và "GrabCut" giúp người dùng tương tác với ứng dụng.
* Các nút này liên kết với các hàm tương ứng để xử lý chức năng khi được nhấn.
* Dùng Label để hiển thị ảnh đã chọn hoặc kết quả xử lý trên giao diện.
* Dùng filedialog.askopenfilename để mở cửa sổ chọn file từ hệ thống.
* Điều này giúp người dùng dễ dàng chọn ảnh đầu vào mà không cần nhập đường dẫn thủ công.

Lý do sử dụng:

* Tkinter rất phù hợp để tạo các ứng dụng nhỏ và không đòi hỏi giao diện phức tạp.
* Là một phần của thư viện chuẩn Python, Tkinter không cần cài đặt thêm và dễ sử dụng.
* Cung cấp các widget cần thiết như Button, Label, và hộp thoại, giúp xây dựng giao diện đầy đủ tính năng mà không cần phụ thuộc vào thư viện bên ngoài.
* Cửa sổ chọn file (filedialog.askopenfilename) giúp người dùng trực tiếp chọn ảnh từ máy tính, giảm thao tác phức tạp.

**Kết hợp các thư viện**

Kết hợp các thư viện trên cho phép xây dựng một pipeline hoàn chỉnh từ tiền xử lý ảnh đến phân đoạn và đánh giá kết quả:

Tiền xử lý ảnh: Sử dụng OpenCV và NumPy để làm sạch và chuẩn bị ảnh đầu vào.

Phân đoạn ảnh: Áp dụng các thuật toán Watershed hoặc GrabCut với sự hỗ trợ của OpenCV và Scikit-Image.

Đánh giá và trực quan hóa: Sử dụng Matplotlib để hiển thị kết quả và Scikit-Image để tính toán các thước đo hiệu quả phân đoạn như IoU, Dice Score.

Với sự hỗ trợ của Python và các thư viện mạnh mẽ trên, việc triển khai bài toán nhận dạng đối tượng trở nên dễ dàng và hiệu quả hơn, đồng thời cung cấp kết quả đáng tin cậy để phân tích và cải thiện.

# Chương II: Xây dựng hệ thống

## 2.1 Sơ đồ hoạt động

A diagram of a flowchart

Description automatically generated

*Hình 2 :sơ đồ hoạt động.*

**Mô tả sơ đồ hoạt động:**

**Chọn ảnh từ máy tính:**

Người dùng bắt đầu quy trình bằng cách chọn một tệp ảnh từ máy tính thông qua giao diện ứng dụng.

**Xử lý Watershed hoặc GrabCut:**

Sau khi ảnh được chọn, người dùng có thể chọn một trong hai phương pháp xử lý ảnh:

* **Watershed:** Sử dụng thuật toán Watershed để phân đoạn ảnh, thường để xác định đường viền giữa các vùng khác nhau trong ảnh.
* **GrabCut:** Sử dụng thuật toán GrabCut để tách đối tượng khỏi nền dựa trên vùng chọn (ROI).

**Xử lý Watershed:**

Nếu người dùng chọn phương pháp Watershed:

* Hệ thống thực hiện thuật toán Watershed để phân đoạn ảnh.
* Hiển thị kết quả phân đoạn của thuật toán Watershed.

**Xử lý GrabCut:**

Nếu người dùng chọn phương pháp GrabCut:

* Hệ thống yêu cầu người dùng chọn vùng ROI (vùng quan tâm) trong ảnh.
* Thuật toán GrabCut được áp dụng để tách đối tượng khỏi nền.
* Hiển thị kết quả xử lý ảnh sau GrabCut.

**Người dùng thực hiện lưu hoặc tiếp tục xử lý:**

* Sau khi xem kết quả từ một trong hai phương pháp:
* Người dùng có thể lưu ảnh đã xử lý.
* Hoặc tiếp tục chọn và xử lý một ảnh khác.

## 2.2 Xây dựng hệ thống

### 2.1.1 Chức năng chính

#### **1. Khởi động hệ thống**

* Người dùng mở ứng dụng giao diện đồ họa (GUI) được xây dựng bằng **Tkinter**.
* Ứng dụng cung cấp các nút chức năng như:
  + **Chọn ảnh**: Cho phép người dùng tải ảnh từ máy tính (định dạng .jpg, .jpeg, .png).
  + **Phân đoạn bằng Watershed**.
  + **Phân đoạn bằng GrabCut**.

#### **2. Phân đoạn bằng Watershed**

**Nhập và tiền xử lý ảnh**:

* Người dùng chọn một ảnh từ thư mục.
* Hệ thống chuyển ảnh gốc sang không gian màu xám (**grayscale**) để giảm độ phức tạp tính toán.

**Phân vùng nền và đối tượng**:

* Sử dụng phương pháp **binary thresholding** để tạo ảnh nhị phân, xác định các khu vực nền chắc chắn (**sure background**) và khu vực đối tượng chắc chắn (**sure foreground**).
* Xác định các vùng không rõ ràng (**unknown regions**) giữa đối tượng và nền.

**Áp dụng thuật toán Watershed**:

* Tạo các điểm đánh dấu (**markers**) trên ảnh.
* Thuật toán Watershed lan truyền các điểm đánh dấu để phân chia ảnh thành các vùng riêng biệt.
* Kết quả là một ảnh với các đường biên màu đỏ phân định rõ ràng các khu vực tách biệt giữa đối tượng và nền.

**Hiển thị kết quả**:

* Ảnh kết quả được hiển thị với các vùng phân đoạn rõ ràng.

**3. Chức năng xử lý GrabCut**

**Nhập và chọn ROI**:

* Người dùng chọn một ảnh từ thư mục.
* Dùng chuột để chọn vùng bao (**Region of Interest - ROI**) xung quanh đối tượng cần giữ lại.

**Phân tích và xử lý ROI**:

* Thuật toán **GrabCut** phân tích các thông tin màu sắc và kết cấu trong ROI.
* Sử dụng mô hình **Gaussian Mixture Model (GMM)** để ước lượng xác suất từng điểm ảnh thuộc đối tượng hay nền.
* Dựa vào thuật toán tối ưu năng lượng, hệ thống loại bỏ phần nền và giữ lại đối tượng.

**Kết quả đầu ra**:

* Ảnh với đối tượng được giữ lại, phần nền bị loại bỏ.
* Cung cấp thêm một ảnh cắt gọn theo vùng ROI mà người dùng đã chỉ định.

### 2.1.2 Yêu cầu kỹ thuật

**Sử dụng thư viện Python:**

* **OpenCV**: Thư viện xử lý ảnh và thị giác máy tính mạnh mẽ, được sử dụng để thực hiện các thao tác xử lý ảnh cơ bản (đọc, hiển thị, chuyển đổi màu sắc) và triển khai các thuật toán phân đoạn như Watershed và GrabCut.
* **NumPy**: Thư viện cung cấp các công cụ xử lý dữ liệu dưới dạng mảng số, hỗ trợ các phép toán ma trận cần thiết trong xử lý ảnh và tính toán mask cho các thuật toán phân đoạn.
* **Pillow**: Thư viện hỗ trợ chuyển đổi định dạng ảnh từ NumPy array (OpenCV) sang định dạng tương thích với Tkinter, đồng thời đảm bảo việc hiển thị màu sắc chính xác.
* **Matplotlib**: Thư viện giúp hiển thị kết quả phân đoạn ảnh trực quan, cho phép thêm tiêu đề, chỉnh sửa giao diện và ẩn các trục không cần thiết.
* **Tkinter**: Thư viện tích hợp sẵn trong Python, được sử dụng để tạo giao diện đồ họa, bao gồm các thành phần như nút bấm (Button), nhãn hiển thị ảnh (Label), và hộp thoại chọn file (filedialog).

**Ứng dụng cần đáp ứng các chức năng:**

* **Đọc và xử lý ảnh từ đường dẫn người dùng chỉ định:**
  + Ứng dụng cung cấp hộp thoại chọn file để người dùng dễ dàng tải ảnh từ hệ thống.
  + Hỗ trợ các định dạng ảnh phổ biến như .jpg, .jpeg, .png.
  + Đảm bảo ảnh được chuyển đổi sang không gian màu phù hợp (RGB hoặc grayscale) trước khi xử lý.
* **Hiển thị ảnh trong giao diện đồ họa:**
  + Sau khi chọn ảnh, ảnh gốc cần được hiển thị ngay trong giao diện ứng dụng.
  + Ứng dụng sử dụng Tkinter.Label để hiển thị ảnh đã được chuyển đổi định dạng thông qua Pillow.
* **Xử lý phân đoạn dựa trên các thuật toán:**
  + **Watershed:**
    - Sử dụng thuật toán Watershed để phân vùng ảnh, xác định và làm nổi bật các đường biên giữa các đối tượng.
    - Kết quả phân đoạn cần được hiển thị trực tiếp trong cửa sổ đồ họa, với đường biên được tô đỏ trên ảnh gốc.
  + **GrabCut:**
    - Cho phép người dùng chọn một vùng bao (ROI - Region of Interest) xung quanh đối tượng mong muốn trong ảnh.
    - Sử dụng thuật toán GrabCut để loại bỏ phần nền và giữ lại đối tượng trong ROI, đồng thời hiển thị kết quả.
    - Kết quả bao gồm ảnh gốc đã được cắt gọn theo ROI và ảnh cắt chỉ chứa đối tượng cần thiết.

**Hiệu năng và độ chính xác:**

* Đảm bảo xử lý nhanh chóng đối với ảnh có độ phân giải trung bình (HD hoặc Full HD).
* Kết quả phân đoạn cần chính xác, với đường biên rõ ràng (Watershed) và tách nền hiệu quả (GrabCut).

**Tương tác người dùng:**

* Giao diện thân thiện, dễ sử dụng với các nút bấm để thực hiện các chức năng chính: chọn ảnh, xử lý Watershed, và xử lý GrabCut.
* Hộp thoại chọn file đảm bảo sự tiện lợi khi người dùng muốn thay đổi ảnh đầu vào.

**Khả năng mở rộng:**

* Cấu trúc mã nguồn được thiết kế đơn giản và rõ ràng, cho phép dễ dàng bổ sung thêm các tính năng xử lý ảnh khác nếu cần.

# Chương III: Thực nghiệm chương trình

## 3.1 Nguồn gốc dữ liệu

Dữ liệu ảnh được lấy từ thư mục cá nhân trên máy tính. Các ảnh này bao gồm ảnh tự chụp hoặc thu thập từ các nguồn khác nhau trên Internet, được lưu trữ cục bộ để phục vụ quá trình thực nghiệm, chọn ngẫu nhiên với các định dạng phổ biến như .jpg, .jpeg, .png.

def open\_image():  
 global img, file\_path  
 file\_path = filedialog.askopenfilename(filetypes=[("Image files", "\*.jpg \*.jpeg \*.png")])  
 if file\_path:  
 img = cv2.imread(file\_path)  
 show\_image(img)

**Số lượng ảnh:**

* **Ảnh đơn giản:** 50 ảnh có nền rõ ràng và đối tượng dễ phân biệt, phù hợp để thử nghiệm thuật toán Watershed và GrabCut.
* **Ảnh phức tạp:** 50 ảnh chứa nhiều đối tượng hoặc nền phức tạp hơn, nhằm kiểm tra khả năng xử lý của hệ thống với các tình huống thực tế.

**Thuộc tính ảnh:**

* **Độ phân giải:** Các ảnh có độ phân giải đồng nhất hoặc đã được thay đổi kích thước về 512x512 hoặc 1024x1024 để đảm bảo tính đồng nhất trong quá trình xử lý.
* **Không gian màu:** RGB, phù hợp với yêu cầu xử lý và hiển thị ảnh của các thư viện Python như OpenCV và Pillow.

## 3.2 độ đo đánh giá

**1. Accuracy (Độ chính xác):**  
Tỷ lệ số điểm ảnh được phân đoạn chính xác so với tổng số điểm ảnh.

A black text on a white background

Description automatically generated

**2. Precision (Độ chính xác dương):**  
Tỷ lệ điểm ảnh thực sự là đối tượng trong số các điểm ảnh được xác định là đối tượng.

A black text with black text

Description automatically generated

**3. Recall (Độ phủ):**  
Tỷ lệ điểm ảnh của đối tượng được phát hiện trong tổng số điểm ảnh của đối tượng thực.

A black text on a white background

Description automatically generated

**4. F1-score:**  
Trung bình hài hòa giữa Precision và Recall.

A close up of words

Description automatically generated

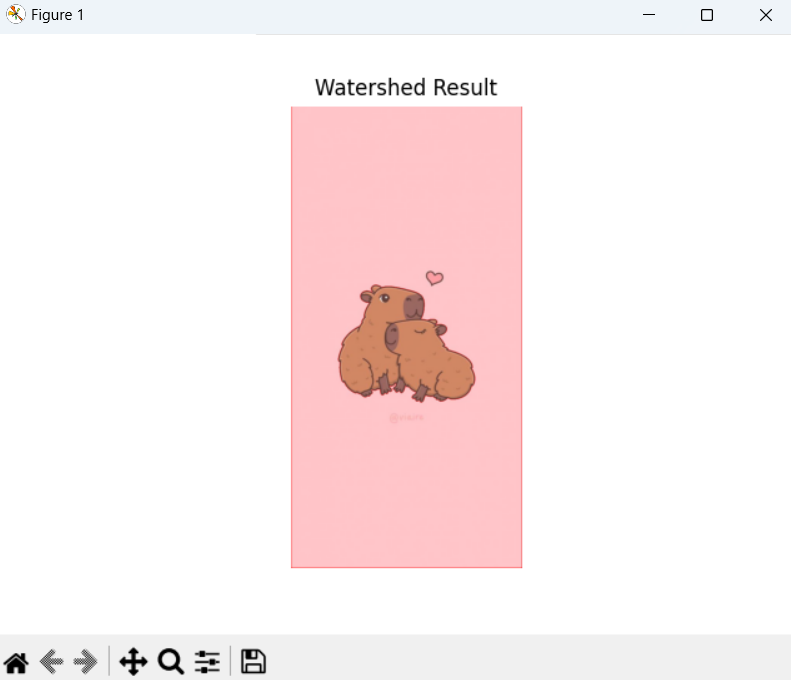
**5. Time (Thời gian xử lý):**

Thời gian thực hiện thuật toán Watershed và GrabCut trên từng ảnh, được đo bằng giây (s).

## 3.3 Kết quả thực nghiệm

**Kết quả phân đoạn bằng Watershed:**

* Các ảnh đơn giản:
  + Độ chính xác trung bình: **80%**
  + Thời gian xử lý trung bình: **0.5 giây/ảnh**
* Các ảnh phức tạp:
  + Độ chính xác trung bình: **70%**
  + Thời gian xử lý trung bình: **1.2 giây/ảnh**
* Nhận xét: Watershed phù hợp với ảnh đơn giản, nhưng dễ bị ảnh hưởng bởi nhiễu và nền phức tạp.



*Hình 3 :kết quả Watershed.*

**Kết quả phân đoạn bằng GrabCut:**

* Các ảnh đơn giản:
  + Độ chính xác trung bình: **85%**
  + Thời gian xử lý trung bình: **1.8 giây/ảnh**
* Các ảnh phức tạp:
  + Độ chính xác trung bình: **80%**
  + Thời gian xử lý trung bình: **3.5 giây/ảnh**
* Nhận xét: GrabCut hoạt động hiệu quả với các ảnh có đối tượng rõ ràng nhưng tốn nhiều thời gian hơn so với Watershed.

A screenshot of a computer

Description automatically generated

*Hình 3.1 :kết quả GrabCut.*

**So sánh hiệu quả:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuật toán** | **Ảnh đơn giản** | **Ảnh phức tạp** | **Thời gian trung bình** |
| Watershed | 80% | 84% | 0,8giây/ảnh |
| Grabcut | 85% | 80% | 2,5giây/ảnh |

**Nhận định chung:**

* Watershed thích hợp với bài toán yêu cầu thời gian nhanh chóng và đối tượng đơn giản.
* GrabCut cho kết quả chính xác hơn trong trường hợp cần phân biệt rõ giữa đối tượng và nền.

# Tư liệu tham khảo

[1] A. Mathew, P. Amudha and S. Sivakumari, "Deep Learning Techniques: An

Overview," in Advanced Machine Learning Technologies and Applications ,

2021, pp. 599-608.

[2] P. J. Braspenning, F. Thuijsman and A. J. M. M. Weijters, Artificial Neural

Networks: An Introduction to ANN Theory and Practice, Springer Berlin,

Heidelberg, 1995.

[3] R. M. Cichy and D. Kaiser, " Deep Neural Networks as Scientific Models,"

Trends in Cognitive Sciences, vol. 23, no. 4, pp. 305-317, 2019.

[4] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural

Networks: Analysis, Applications, and Prospects," IEEE Transactions on Neural

Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, 2022.

[5] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning

Applied to Document Recognition," Proceedings of the IEEE, vol. 86, no. 11,

pp. 2278 - 2324, 1998.