1. Phần xử lí ảnh.
2. Tính thực tế của hệ thống xử lí ảnh và định hướng xây dựng hệ thống.
   1. Tính thực tế.

* *Xác định mục tiêu cụ thể, tối ưu hóa cho hệ thống:*

Xây dựng thuật toán xử lí ảnh xung quanh 1 vấn đề cụ thể như sản xuất, y tế, nông nghiệp hay an ninh. Thiết kế 1 hệ thống riêng cho từng vấn đề này nhằm đảm bảo việc tối ưu hóa hiệu suất và tốc độ xử lí.

* *Hệ thống hoạt động độc lập.*

Hệ thống hoạt động độc lập, có thể triển khai trên nhiều nền tảng, và thiết bị khác nhau. Có thể kết nối tương thích với các thiết bị và nền tảng khác.

* *Lưu trữ dưới nhiều định dạng.*

Ảnh số có thể được lưu trữ dưới nhiều định dạng khác nhau như JPEG, PNG, BMP, TIFF, và mỗi định dạng có các ưu và nhược điểm riêng. Việc chuyển đổi giữa các định dạng và lưu trữ ảnh một cách hiệu quả là một vấn đề nghiên cứu cần được giải quyết. Điều này bao gồm việc nén ảnh mà không làm giảm chất lượng quá nhiều, lưu trữ và truy xuất ảnh một cách nhanh chóng và hiệu quả.

* *Bảo mật cho hệ thống.*

Hệ thống được phát triển hoàn toàn dựa trên lý thuyết và không sử dụng những thư viện có sẵn hiện nay. Đảm bảo về mặt bản quyền và tính bảo mật của hệ thống.

Giải quyết các vấn đề nghiên cứu trên không chỉ giúp phát triển một ứng dụng xử lý ảnh xám hiệu quả và chất lượng mà còn đảm bảo rằng ứng dụng của bạn có thể hoạt động ổn định, bảo mật và dễ dàng tích hợp vào các hệ thống khác.

* 1. Định hướng xây dựng hệ thống.
     1. Đối tượng xử lí và yêu cầu đầu ra của hệ thống xử lí ảnh.
* *Đối tượng nhận dạng.*

Đối tượng nhận dạng là những hộp carton chứa đồ vật có trọng lượng từ 500g đến 1kg đã được gán mác.

* *Mục tiêu.*
* Xác định được hình dạng, kích thước, trọng tâm và góc nghiên của vật từ đó có thể ứng dụng cho việc cầm nắm vật chính xác và phân loại sản phẩm.
* Xử lí ảnh nhanh, chính xác bằng cách sử dụng các phương pháp tiên tiến và tối ưu để đảm bảo xử lí ảnh nhanh chóng và đáng tin cậy. Các phương pháp này sẽ được thiết kế để áp dụng các phép biến đổi, phân tích và nhận dạng đối tượng trên ảnh một cách chính xác và hiệu quả.
* Đơn giản hóa quá trình xử lí ảnh, tạo ra một hệ thống tự động hóa, không cần phải nhập thông số cụ thể từ người dùng. Điều này đòi hỏi phải sử dụng thuật toán thông minh, có khả năng xác định từ dữ liệu và tự động tìm ra các đặc trưng quan trọng trong ảnh. Hệ thống sẽ thực hiện các phép biến đổi và phân tích ảnh một cách tự động và đáng tin cậy.
  + 1. Định hướng xây dựng hệ thống xử lí ảnh.

Xử lí ảnh bằng trí tuệ nhân tạo (AI) và xử lí ảnh bằng thuật toán là hai phương pháp phổ biến để xử lí và phân tích dữ liệu hình ảnh. Dưới đây là một số khác biệt chính giữa hai phương pháp này:

* *Đặc điểm.*

|  |  |  |
| --- | --- | --- |
| Các phương pháp xử lí | Phương pháp xử lí ảnh bằng thuật toán. | Phương pháp xử lí ảnh bằng trí tuệ nhân tạo. |
| Thuật toán | Thuật toán xử lí ảnh thường được thiết kế và triển khai theo quy tắc và quy trình cụ thể. | Xử lí ảnh bằng trí tuệ nhân tạo sử dụng các mô hình học máy hoặc mạng nơ-ron nhân tạo để tự động học và trích xuất thông tin từ dữ liệu hình ảnh. |
| Quy trình thực hiện | Các thuật toán xử lí ảnh thông thường dựa trên các phép biến đổi toán học và xử lí tín hiệu, chẳng hạn như lọc thông thấp, lọc nhiễu, phát hiện biên, phân đoạn ảnh, và mô hình màu. | Các thuật toán học máy và mạng nơ-ron nhân tạo có khả năng tự điều chỉnh và học từ dữ liệu, giúp cải thiện độ chính xác theo thời gian. |
| Yêu cầu | Các thuật toán này thường yêu cầu kiến thức chuyên môn về xử lí ảnh và thường được viết và tối ưu cho một mục đích cụ thể. | Xử lí ảnh bằng trí tuệ nhân tạo có thể thực hiện các tác vụ phức tạp như nhận dạng đối tượng, phân loại, phát hiện khuôn mặt, nhận dạng văn bản, và tạo ảnh mới dựa trên dữ liệu huấn luyện. |
| Ứng dụng | Xử lí ảnh bằng thuật toán có thể cung cấp kết quả hiệu quả cho các tác vụ xử lí đơn giản và có cấu trúc rõ ràng. | Phương pháp này có thể tự động học và tùy chỉnh cho từng tác vụ cụ thể và có khả năng khai thác thông tin phức tạp trong dữ liệu hình ảnh. |

* *Ưu điểm và nhược điểm.*

|  |  |  |
| --- | --- | --- |
|  | Phương pháp xử lí ảnh bằng thuật toán. | Phương pháp xử lí ảnh bằng trí tuệ nhân tạo. |
| Ưu điểm | Quy tắc rõ ràng: Các thuật toán xử lí ảnh dựa trên quy tắc và quy trình cụ thể, giúp hiểu rõ logic và cách hoạt động của từng bước xử lí.  Hiệu suất: Vì được thiết kế và tối ưu cho mục đích cụ thể, các thuật toán xử lí ảnh thông thường thường có thể thực hiện nhanh chóng và hiệu quả cho các tác vụ đơn giản và có cấu trúc rõ ràng. | Khả năng học tự động: Trí tuệ nhân tạo và học máy cho phép mô hình tự động học từ dữ liệu và cải thiện độ chính xác theo thời gian.  Xử lí dữ liệu phức tạp: Xử lí ảnh bằng trí tuệ nhân tạo có khả năng xử lí dữ liệu hình ảnh phức tạp và trích xuất thông tin phức tạp như nhận dạng đối tượng, phân loại, và phát hiện. |
| Nhược điểm | Giới hạn linh hoạt: Các thuật toán xử lí ảnh thông thường thường chỉ áp dụng cho một tác vụ cụ thể và không linh hoạt trong việc tự động học và thích ứng với dữ liệu mới.  Khó khăn trong xử lí dữ liệu phức tạp: Khi đối mặt với dữ liệu hình ảnh phức tạp, các thuật toán xử lí ảnh thông thường có thể gặp khó khăn trong việc trích xuất thông tin và đưa ra kết quả chính xác. | Yêu cầu dữ liệu huấn luyện lớn: Để đạt được độ chính xác cao, mô hình trí tuệ nhân tạo thường yêu cầu một lượng lớn dữ liệu huấn luyện, và việc thu thập và gán nhãn dữ liệu có thể tốn kém và công phu. |

Tóm lại, xử lí ảnh bằng thuật toán phụ thuộc vào quy tắc và quy trình xử lí ảnh, trong khi xử lí ảnh bằng trí tuệ nhân tạo sử dụng khả năng học máy và trí tuệ nhân tạo để tự động học và trích xuất thông tin từ dữ liệu hình ảnh. Xử lí ảnh bằng thuật toán có ưu điểm về quy tắc rõ ràng và hiệu suất, trong khi xử lí ảnh bằng trí tuệ nhân tạo có ưu điểm về khả năng học tự động và xử lí dữ liệu phức tạp. Cả hai phương pháp đều có ứng dụng và ưu điểm riêng, tuỳ thuộc vào tác vụ và mục đích sử dụng cụ thể. Thông qua những định nghĩa và so sánh trên, đối với việc nhận diện những tính chất có độ phức tạp thấp như hình dạng, kích thước, trọng tâm và góc nghiêng của vật thì có thể thấy được việc xử lí bằng thuật toán sẽ phù hợp hơn với các yêu cầu đã được đặt ra.

Thông qua những phân tích về tính chất, ưu và nhược điểm của 2 phương pháp, kết hợp với yêu cầu của đề bài đã đặt ra cùng với định hướng của việc xây dựng hệ thống, quyết định sử dụng thuật toán để xử lí ảnh.

1. Cơ sở lý thuyết.

Ở phần này sẽ xác định và làm rõ nội dung lý thuyết các thuật toán được sử dụng để xây dựng hệ thống xử lí ảnh. Các thuật toán đã được so sánh và chọn lọc từ trước với góc nhìn cá nhân người thực hiện, phù hợp với đề tài thực hiện.

* 1. Thuật toán Hough Transform.

1. Định nghĩa.

Hough Transform (HT) là một kỹ thuật phát hiện các hình dạng cơ bản trong hình ảnh, chẳng hạn như đường thẳng, đường tròn, và đường ellipse, bằng cách biến đổi không gian điểm ảnh thành không gian tham số. Kỹ thuật này đặc biệt hiệu quả trong việc phát hiện các đường thẳng và đường tròn trong ảnh nhiễu. Dưới đây là mô tả chi tiết về tính chất thuật toán Hough Transform.

1. Tính chất.

* *Phát Hiện Hình Dạng Hình Học Cụ Thể.*
* Đường Thẳng: Hough Transform có thể phát hiện các đường thẳng bằng cách chuyển đổi từ không gian tọa độ Cartesian sang không gian tham số (𝜌,𝜃).
* Đường Tròn: Có thể phát hiện các hình tròn bằng cách sử dụng không gian tham số (𝑎,𝑏,𝑟), trong đó (a,b) là tọa độ tâm và 𝑟 là bán kính.
* Hình Dạng Khác: Có thể mở rộng để phát hiện các hình dạng khác như elip, parabol, và các đường cong tham số khác.
* *Tính Bất Biến Đối Với Nhiễu.*
* Ít Nhạy Cảm Với Nhiễu: Hough Transform tích lũy bằng chứng về sự hiện diện của các hình dạng cụ thể trong không gian tham số, làm cho nó ít nhạy cảm với nhiễu cục bộ trong hình ảnh.
* *Độc Lập Với Tỷ Lệ và Xoay.*
* Bất Biến Với Tỷ Lệ và Xoay: Có khả năng phát hiện các hình dạng mục tiêu bất kể kích thước và hướng của chúng trong hình ảnh. Điều này làm cho Hough Transform trở nên linh hoạt trong nhiều ứng dụng khác nhau.
* *Tốn Tài Nguyên Tính Toán.*
* Yêu Cầu Cao Về Tài Nguyên: Đòi hỏi bộ nhớ và thời gian tính toán lớn, đặc biệt khi tìm kiếm các hình dạng phức tạp trong không gian tham số cao. Điều này có thể hạn chế hiệu quả của nó trong các ứng dụng yêu cầu xử lý thời gian thực hoặc trên các thiết bị tài nguyên hạn chế.
* *Không Phù Hợp Cho Các Hình Dạng Tự Do.*
* Hạn Chế Với Hình Dạng Tự Do: Chủ yếu hiệu quả với các hình dạng hình học có tham số rõ ràng. Không phù hợp cho việc phát hiện các hình dạng tự do hoặc không có tham số xác định.
* *Khả Năng Mở Rộng.*
* Linh Hoạt và Mở Rộng: Có thể được điều chỉnh và mở rộng để phát hiện các hình dạng không tiêu chuẩn bằng cách thay đổi không gian tham số và các phương trình hình học tương ứng.

1. Giải thích thuật toán.

Yêu cầu ảnh đầu vào của thuật toán Hough Transform là ảnh nhị phân thể hiện biên dạng của vật thể, từ đó dựa trên những phép biến đổi để xác định những hình ảnh cạnh đó là đường thẳng hoặc cong.

Trường hợp phổ biến được sử dụng của phép biến đổi Hough Transform là phát hiện ra những đường thẳng. Đường thẳng **y = ax + b** có thể được biểu diễn dưới dạng một điểm (a,b) trong không gian tham số. Tuy nhiên, có một vấn đề là nếu ở dạng này chúng sẽ làm phát sinh ra các giá trị không giới hạn của tham số độ dốc **a**. Do đó, vì lý do tính toán phải sử dụng phương trình đường thẳng dưới dạng hệ tọa độ cực.

r = x. cos(θ) + y.sin(θ) (1)



Hình 1. Đường thẳng được biểu diễn thông qua r và θ.

Với phương trình (1), cho 1 cặp số (x,y) mà tại đó độ sáng I(x,y) = 255 với θ phân bố từ 0 đến 2π, sẽ vẽ được 1 đường hình sin. Ví dụ, đối với x0 = 8 và y0 = 6, khi vẽ trên mặt phẳng (r - θ) nhận được:

A graph with a curve

Description automatically generated

Hình 2.

Thực hiện thao tác trên cho tất cả các điểm ảnh có độ sáng I = 255 trong cùng 1 tấm ảnh. Nếu đường cong của hai điểm đó giao nhau trong mặt phẳng (r - θ), điều đó có nghĩa là cả 2 điểm đều thuộc về cùng 1 đường thẳng . Ví dụ, làm theo ví dụ ở trên và vẽ cho 2 điểm nữa: x1 = 4, y1 = 9 và x2 = 4, y2 = 9, nhận được kết quả sau:

A graph with colored lines

Description automatically generated

Hình 3.

3 đường giao nhau tại (0.925 , 9.6), tọa độ này đại diện cho các tham số (r , θ). Từ cặp tham số (r , θ) có thể đưa vào phương trình (1) để trở thành 1 phương trình đường thẳng đi qua 3 điểm (x0 , y0), (x1 , y1), (x2 , y2).

Tóm lại, một đường có thể được phát hiện bằng cách tìm số giao điểm giữa các đường hình sin ở trên. Càng nhiều đường sin giao nhau tại 1 điểm có nghĩa là đường thẳng đó càng đi qua nhiều điểm. Từ đó có thể sử dụng một ngưỡng xác định cần thiết để phát hiện 1 đường.

Những gì thuật toán Hough Transform làm là nó theo giao điểm giữa các đường sin của mọi điểm ảnh . Nếu số lượng đường giao tại 1 điểm cao hơn *số ngưỡng ,* thì khai báo nó là một đường thẳng với cặp tham số (r , θ) của điểm giao.

* 1. Thuật toán nhận dạng đường biên.

1. Định nghĩa.

Thuật toán Canny là một trong những phương pháp phát hiện biên phổ biến nhất trong xử lý ảnh, được phát triển bởi John F. Canny vào năm 1986. Mục tiêu của thuật toán Canny là xác định các biên rõ ràng trong ảnh bằng cách giảm thiểu ảnh hưởng của nhiễu trong khi tối đa hóa khả năng phát hiện biên.

1. Tính chất.

* *Độ Chính Xác Tốt.*
* Thuật toán Canny được thiết kế để tối đa hóa khả năng phát hiện các biên thực sự trong ảnh. Nó đảm bảo rằng tất cả các biên quan trọng đều được phát hiện.
* *Định Vị Tốt .*
* Thuật toán Canny xác định vị trí của các biên với độ chính xác cao. Vị trí của các biên được tính toán một cách chính xác mà không bị dịch chuyển bởi nhiễu.
* *Đáp Ứng Đơn Lập.*
* Thuật toán Canny đảm bảo rằng mỗi biên chỉ được phát hiện một lần. Điều này giúp loại bỏ các biên giả và giảm thiểu nhiễu.
* *Khả Năng Chống Nhiễu.*
* Bộ lọc Gaussian trong bước đầu tiên của thuật toán giúp giảm thiểu nhiễu trong ảnh, do đó cải thiện chất lượng của biên được phát hiện.

1. Giải thích thuật toán.

Thuật toán phát hiện cạnh Canny bao gồm 5 bước:

* Giảm nhiễu.
* Tính toán Gradient độ xám của ảnh.
* Áp dụng Non-maximum suppression.
* Ngưỡng kép.
* Theo dõi cạnh bằng độ trễ.

Một điều quang trọng cần lưu ý khi thực hiện thuật toán phát hiện cạnh Canny là bức ảnh đầu vào phải ở thang độ xám trước khi thực hiện.

* Bước 1: Giảm nhiễu.

Có nhiều cách giúp làm giảm nhiễu, cách phổ biến thường gặp là sử dụng bộ lọc Gaussian giúp làm mịn ảnh. Để thực hiện bộ lọc Gaussian, kỹ thuật tích chập hình ảnh được áp dụng với Gaussian Kernel. Kích thước kernel phụ thuộc vào hiệu ứng làm mờ mong muốn. Về cơ bản, kernel càng nhỏ hiệu quả làm mờ càng ít.

Phương trình cho một kernel bộ lọc Gaussian có kích thước (2k +1, 2k+1):

Hij = exp(−) ; 1< i,j<(2k+1)

A comparison of a person

Description automatically generated

Hình 4. Ảnh gốc (trái) – Hình ảnh bị làm mờ với bộ lọc Gaussian (sigma = 1.4 và kích thước kernel là 5×5)

* Bước 2: Tính toán Gradient độ xám.
* Sau khi làm mờ, tính Gradient của ảnh để phát hiện các thay đổi đột ngột trong cường độ sáng, từ đó xác định được các biên.
* Sử dụng các bộ lọc Sobel để tính Gradient theo hướng x và y:

Gx = . I(x,y) Gy = . I(x,y)

Với: Gx: Gradient theo hướng x.

Gy: Gradient theo hướng y.

I(x,y): giá trị độ sáng tại điểm có tọa độ (x,y)

* Tính độ lớn và hướng Gradient:

G =

𝜃 = arctan ()

A comparison of a person

Description automatically generated

Hình 5. Hình ảnh bị mờ (trái) – Cường độ gradient (phải)

* Bước 3: Áp dụng Non-maximum suppression.
* Mục tiêu của bước này là làm mỏng cách cạnh biên bằng cách giữ lại các điểm cực đại của gradient và loại bỏ các điểm không phải là biên.
* Với mỗi điểm ảnh, kiểm tra các điểm lân cận theo hướng gradient(𝜃).
* Giữ lại nếu nó là điểm cực đại so với điểm lân cận.

A comparison of a person's face

Description automatically generated

Hình 6. Kết quả của thuật toán Non-maximum suppression

* Bước 4: Ngưỡng kép.
* Ngưỡng kép giúp phân loại các điểm ảnh thành biên mạnh, biên yếu và không phải là biên.
* Áp dụng hai ngưỡng: ngưỡng cao (TH ​) và ngưỡng thấp (TL​).
  + Biên mạnh: điểm ảnh có giá trị gradient lớn hơn TH ​.
  + Biên yếu: điểm ảnh có giá trị gradient giữa TL và TH .
  + Không phải là biên: điểm ảnh có giá trị gradient nhỏ hơn TL ​.

A screenshot of a black and white image of a person

Description automatically generated

Hình 7. Hình ảnh trước non-maximum suppression (trái) – Kết quả (phải): pixel yếu có màu xám và pixel mạnh có màu trắng.

* Bước 5: Theo dõi cạnh bằng độ trễ.
* Mục tiêu của bước này là liên kết các biên yếu với biên mạnh nếu chúng liền kề, đảm bảo tính liên tục của biên.
* Duyệt qua tất cả các điểm ảnh.
  + Nếu điểm ảnh là biên mạnh, giữ lại.
  + Nếu điểm ảnh là biên yếu và có ít nhất một điểm ảnh lân cận là biên mạnh, giữ lại.

A black and white image of a person

Description automatically generated

Hình 8

Thuật toán phân ngưỡng Otsu.

1. Định nghĩa.

Thuật toán Otsu là một phương pháp tự động xác định ngưỡng phân đoạn trong xử lý ảnh để tách một ảnh xám thành hai vùng: vật thể và nền ảnh. Thuật toán này dựa trên việc tối ưu hóa phương sai giữa các lớp (inter-class variance) để tìm ngưỡng tốt nhất phân biệt giữa hai vùng trong ảnh. Nobuyuki Otsu đã phát triển phương pháp này vào năm 1979 và nó đã trở thành một trong những kỹ thuật phân ngưỡng phổ biến nhất.

1. Tính chất.

* Tự Động Tìm Ngưỡng

Thuật toán Otsu tự động xác định ngưỡng tối ưu để phân tách các pixel trong ảnh thành hai lớp (foreground và background) mà không cần bất kỳ thông tin bên ngoài nào. Điều này làm cho nó trở thành một công cụ hữu ích cho nhiều ứng dụng xử lý ảnh.

* Dựa Trên Phương Sai Giữa Các Lớp

Thuật toán Otsu hoạt động dựa trên việc tối ưu hóa phương sai giữa các lớp. Nó tìm ngưỡng sao cho phương sai giữa các lớp (background và foreground) được tối đa hóa. Cụ thể:

* Phương sai tổng thể (​) của ảnh được tính bằng cách sử dụng toàn bộ các giá trị cường độ của ảnh.
* Phương sai trong lớp (​) đo lường sự phân tán trong mỗi lớp (background và foreground).
* Phương sai giữa các lớp (​) đo lường sự phân tán giữa hai lớp này và được tối đa hóa trong thuật toán Otsu.
* Không Yêu Cầu Các Tham Số Ban Đầu

Một trong những ưu điểm lớn của thuật toán Otsu là nó không yêu cầu bất kỳ tham số ban đầu nào từ người dùng. Ngưỡng tối ưu được tính toán trực tiếp từ histogram của ảnh.

* Hiệu Quả Tính Toán

Thuật toán Otsu tính toán ngưỡng tối ưu dựa trên histogram của ảnh, do đó nó rất hiệu quả về mặt tính toán. Việc tính toán này có thể được thực hiện một cách nhanh chóng và dễ dàng ngay cả với các ảnh có kích thước lớn.

* Áp Dụng Cho Các Ảnh Cấp Xám

Thuật toán Otsu được thiết kế chủ yếu cho các ảnh cấp xám (grayscale images). Đối với các ảnh màu, có thể cần áp dụng các biến thể của thuật toán hoặc chuyển đổi ảnh màu sang ảnh cấp xám trước khi áp dụng Otsu.

* Khả Năng Ứng Dụng Cho Phân Ngưỡng Đa Cấp

Mặc dù thuật toán Otsu gốc chỉ xác định một ngưỡng để phân chia ảnh thành hai lớp, nó có thể được mở rộng để xác định nhiều ngưỡng (multi-thresholding) nhằm phân chia ảnh thành nhiều lớp. Điều này được thực hiện bằng cách tối ưu hóa tổng phương sai giữa các lớp cho nhiều ngưỡng.

* Nhạy Cảm Với Histogram Của Ảnh

Hiệu suất của thuật toán Otsu phụ thuộc vào histogram của ảnh. Đối với các ảnh có histogram rõ ràng với hai đỉnh riêng biệt, thuật toán Otsu hoạt động rất tốt. Tuy nhiên, đối với các ảnh có histogram phẳng hoặc có nhiễu, kết quả có thể không tối ưu.

* Độ Ổn Định Cao

Thuật toán Otsu thường cho kết quả ổn định và nhất quán, đặc biệt là khi histogram của ảnh có sự phân biệt rõ ràng giữa các lớp.

1. Giải thích thuật toán.

* Bước 1: Tính histogram và xác suất xuất hiện của các mức xám
* Duyệt qua tất cả các pixel trong ảnh để tính histogram.
* Tính xác suất xuất hiện của mỗi mức xám P(i):

P(i) =

trong đó ni​ là số lượng pixel có mức xám i và N là tổng số pixel trong ảnh.

* Bước 2: Khởi tạo các giá trị ban đầu
* Tính giá trị trung bình toàn cục của ảnh

μT =

* Bước 3: Duyệt qua tất cả các cặp ngưỡng có thể (từ 0 đến 255)
* Với mỗi cặp ngưỡng t1​ và t2​ (với 0 ≤ t1 < t2 ≤ 255), chia ảnh thành ba lớp:
  + Lớp 1: [0, t1​]
  + Lớp 2: [t1 + 1, t2​]
  + Lớp 3: [t2 + 1, 255]
* Tính xác suất và giá trị trung bình của mỗi lớp:
  + Tổng xác suất của lớp 1:

ω0 =

* + Tổng xác suất của lớp 2:

ω1 =

* Tổng xác suất của lớp 3:

ω2 =

* Giá trị trung bình của lớp 1:

μ0 =

* Giá trị trung bình của lớp 2:

μ1 =

* + Giá trị trung bình của lớp 3:

μ2 =

* Bước 4: Tính tổng phương sai giữa các lớp
* Tính tổng phương sai giữa các lớp ​:

​ = ω0 . + ω1 . + ω2 .

* Bước 5: Tìm ngưỡng tối ưu
* Tìm cặp ngưỡng t1​ và t2​ làm cho tổng phương sai giữa các lớp ​ lớn nhất.
  1. Thuật toán BFS.

1. Định nghĩa.

Tìm kiếm theo chiều rộng (BFS) là một thuật toán để duyệt đồ thị hoặc cây. BFS áp dụng cho cây và đồ thị gần như giống nhau. Sự khác biệt duy nhất là đồ thị có thể chứa các chu trình, vì vậy chúng ta có thể duyệt lại cùng một nút. Để tránh xử lý lại cùng một nút, chúng ta sử dụng mảng boolean đã truy cập, mảng này sẽ đánh dấu các đỉnh đã truy cập. BFS sử dụng cấu trúc dữ liệu hàng đợi (queue) để tìm đường đi ngắn nhất trong biểu đồ.

1. Tính chất.

* Đảm Bảo Liên Thông

BFS đảm bảo rằng tất cả các pixel thuộc về cùng một vùng liên thông sẽ được gán cùng một nhãn. Thuật toán này duyệt qua toàn bộ các pixel liên kề theo chiều rộng, đảm bảo không bỏ sót bất kỳ pixel nào trong vùng.

* Độ Phức Tạp Tính Toán

Độ phức tạp thời gian của BFS là O(V+E), trong đó V là số lượng pixel và E là số cạnh giữa các pixel. Trong trường hợp xử lý ảnh, E thường tỷ lệ với V, do đó, độ phức tạp thời gian là O(n⋅m)O, với nnn là chiều rộng và mmm là chiều cao của ảnh. Điều này đảm bảo BFS hoạt động hiệu quả trên các ảnh có kích thước lớn.

* Không Gian Bộ Nhớ

BFS sử dụng hàng đợi để lưu trữ các pixel chờ xử lý. Trong trường hợp xấu nhất, hàng đợi có thể chứa tất cả các pixel của ảnh, do đó không gian bộ nhớ cần thiết là O(n⋅m).

* Tính Chất Hàng Đợi

BFS sử dụng hàng đợi để xử lý các pixel theo thứ tự, đảm bảo rằng các pixel gần nhau được xử lý trước, giúp gán nhãn một cách hệ thống và chính xác.

* Tính Ổn Định

BFS đảm bảo tính ổn định trong việc phân loại vùng liên thông. Mỗi lần chạy thuật toán BFS với cùng một ảnh đầu vào sẽ cho ra cùng một kết quả phân loại vùng liên thông.

* Khả Năng Xử Lý Nhiều Thành Phần

BFS có thể xử lý nhiều thành phần liên thông trong một ảnh duy nhất. Mỗi khi phát hiện một pixel chưa được gán nhãn, thuật toán sẽ bắt đầu một cuộc tìm kiếm mới để gán nhãn cho toàn bộ vùng liên thông liên quan.

1. Giải thích thuật toán.

 **Khởi tạo**.

* Tạo một ma trận nhãn để đánh dấu các pixel đã được gán nhãn, tất cả các phần tử ban đầu bằng 0. Khởi tạo nhãn bắt đầu từ 1.

 **Duyệt qua từng pixel trong ảnh**.

* Nếu pixel chưa được gán nhãn và thuộc foreground, bắt đầu một cuộc tìm kiếm BFS từ pixel này.

 **BFS để gán nhãn cho vùng liên thông**.

* Sử dụng hàng đợi để duyệt các pixel lân cận và gán nhãn cho các pixel thuộc cùng một vùng liên thông.

 **Gán nhãn cho toàn bộ vùng liên thông**.

* Tiếp tục duyệt cho đến khi tất cả các pixel được gán nhãn.

1. Thực hiện xây dựng hệ thống xử lí ảnh.

Sau khi làm rõ các phần lý thuyết cốt lõi trong việc xây dựng hệ thống. Cho ra lưu đồ giải thuật sau.

A diagram of a flowchart

Description automatically generated

Hình 9. Lưu đồ giải thuật hệ thống xử lí ảnh.

* *Mô tả cụ thể các bước*

Bước 1: Nhận dữ liệu từ camera:

* Mô tả: Camera chụp ảnh và truyền dữ liệu dưới dạng mảng một chiều chứa các giá trị điểm ảnh.
* Mục tiêu: Đảm bảo dữ liệu ảnh được nhận đầy đủ và chính xác để chuẩn bị cho các bước xử lý tiếp theo.

Bước 2: Chuyển đổi dữ liệu từ mảng 1 chiều thành mảng 2 chiều:

* Mô tả: Chuyển dữ liệu từ mảng một chiều thành mảng hai chiều với kích thước 640x480, đại diện cho bức ảnh xám.
* Mục tiêu: Tạo cấu trúc dữ liệu thích hợp để dễ dàng thực hiện các bước xử lý ảnh.

A square object in the dark

Description automatically generated

Hình 9. Ảnh đầu vào.

Bước 3: Áp dụng thuật toán Otsu để phân biệt vật thể và nền:

* Mô tả: Thuật toán Otsu được sử dụng để tìm ngưỡng phân loại giữa vật thể và nền. Các điểm ảnh của vật thể được gán giá trị 255 (trắng), và nền được gán giá trị 0 (đen).
* Mục tiêu: Phân biệt rõ ràng giữa vật thể và nền trong ảnh.

A white square on a black background

Description automatically generated

Hình 10. Kết quả phân đoạn ảnh bằng ngưỡng Otsu.

Bước 4: Lọc nhiễu bằng thuật toán Openning:

* Mô tả: Sử dụng kỹ thuật Openning, bao gồm Dilation (giãn nở) theo sau là Erosion (xói mòn), để loại bỏ các nhiễu nhỏ và lấp các khoảng trống trong ảnh.
* Mục tiêu: Cải thiện chất lượng ảnh bằng cách loại bỏ các nhiễu không mong muốn.

Bước 5: Lọc các vùng vật thể nhỏ bằng thuật toán BFS:

* Mô tả: Loại bỏ các phần vật thể có diện tích nhỏ và chỉ giữ lại phần có diện tích lớn nhất.
* Mục tiêu: Đảm bảo chỉ các vật thể lớn nhất (có thể là mục tiêu chính) được giữ lại.

A white diamond shaped object on a black background

Description automatically generated

Hình 11. Sau khi lọc nhiễu bằng thuật toán BFS.

Bước 6: Phát hiện cạnh bằng thuật toán Canny:

* Mô tả: Áp dụng thuật toán Canny để phát hiện các cạnh của vật thể trong ảnh.
* Mục tiêu: Xác định các biên rõ ràng của vật thể trong ảnh.

A white outline of a diamond on a black background

Description automatically generated

Hình 12. Nhận diện cạnh bằng phương pháp Canny.

Bước 7: Vẽ ma trận Hough từ ảnh cạnh:

* Mô tả: Sử dụng phương pháp Hough Transform để vẽ ma trận Hough từ ảnh cạnh.
* Mục tiêu: Xác định các đường thẳng có trong ảnh.

A black and white background with a black background

Description automatically generated with medium confidence

Hình 13. Ma trận Hough để chọn ra các cặp (r,θ)

Bước 8: Lấy các điểm đặc trưng từ ma trận Hough:

* Mô tả: Trích xuất các điểm đặc trưng từ ma trận Hough.
* Mục tiêu: Xác định các điểm quan trọng đại diện cho các đường thẳng trong ảnh.

Bước 9: Quy đổi các điểm đặc trưng về phương trình đường thẳng và tìm giao điểm:

* Mô tả: Chuyển đổi các điểm đặc trưng về phương trình đường thẳng và tìm các giao điểm giữa các đường này.
* Mục tiêu: Xác định các giao điểm để nhận dạng hình dạng vật thể.

Bước 10: Phân loại hình dạng của ảnh:

* Mô tả: Dựa trên số lượng phương trình và giao điểm, tiến hành phân loại hình dạng của vật thể:
  + 3 phương trình và 3 giao điểm: Hình tam giác.
  + 4 phương trình, 4 giao điểm và 4 góc vuông: Hình chữ nhật hoặc hình vuông.
  + Nếu không thỏa điều kiện trên, kiểm tra xem có phải hình tròn hay không dựa trên khoảng cách từ tâm tới các điểm biên.
* Mục tiêu: Nhận dạng và phân loại hình dạng của vật thể trong ảnh.

A black and white background with white lines

Description automatically generated

Hình 14. Các đường thẳng được biểu diễn.

Bước 11: Kết luận về nhận dạng:

* Mô tả: Nếu không phải là hình chữ nhật, hình vuông, hoặc hình tròn, kết luận rằng không thể nhận dạng được hình dạng của vật thể.
* Mục tiêu: Đưa ra kết luận cuối cùng về hình dạng, trọng tâm, kích thước của vật thể.
* *Kết quả thực hiện xử lí ảnh theo quy trình ở phần trên.*

A collage of images of diamond shapes

Description automatically generated

Hình 14. Kết quả của quy trình từ 1 bước ảnh đầu vào có thể vẽ lại được biên dạng.

1. Kết quả, đánh giá hệ thống.

* Kết quả nhận diện một số ảnh.

A screenshot of a computer

Description automatically generated

Hình 15. Nhận dạng hình hộp chữ nhật.

A screenshot of a computer screen

Description automatically generated

Hình 16. Nhận dạng hình hộp tam giác.

A screenshot of a computer

Description automatically generated

Hình 17. Nhận dạng hình tròn.

A screenshot of a computer

Description automatically generated

Hình 18. Nhận dạng hình vuông.

* Đánh giá.
* Ưu điểm:
  + Hệ thống nhận diện có thể dự đoán được hình dạng vật thể, trọng tâm và kích thước (px) của vật thể.
  + Có thể nhận dạng được nếu vật thể bị khuất.
  + Độ sai lệch thấp.
* Nhược điểm:
  + Phụ thuộc vào độ sáng.
  + Hạn chế về phần nhận diện đường tròn.

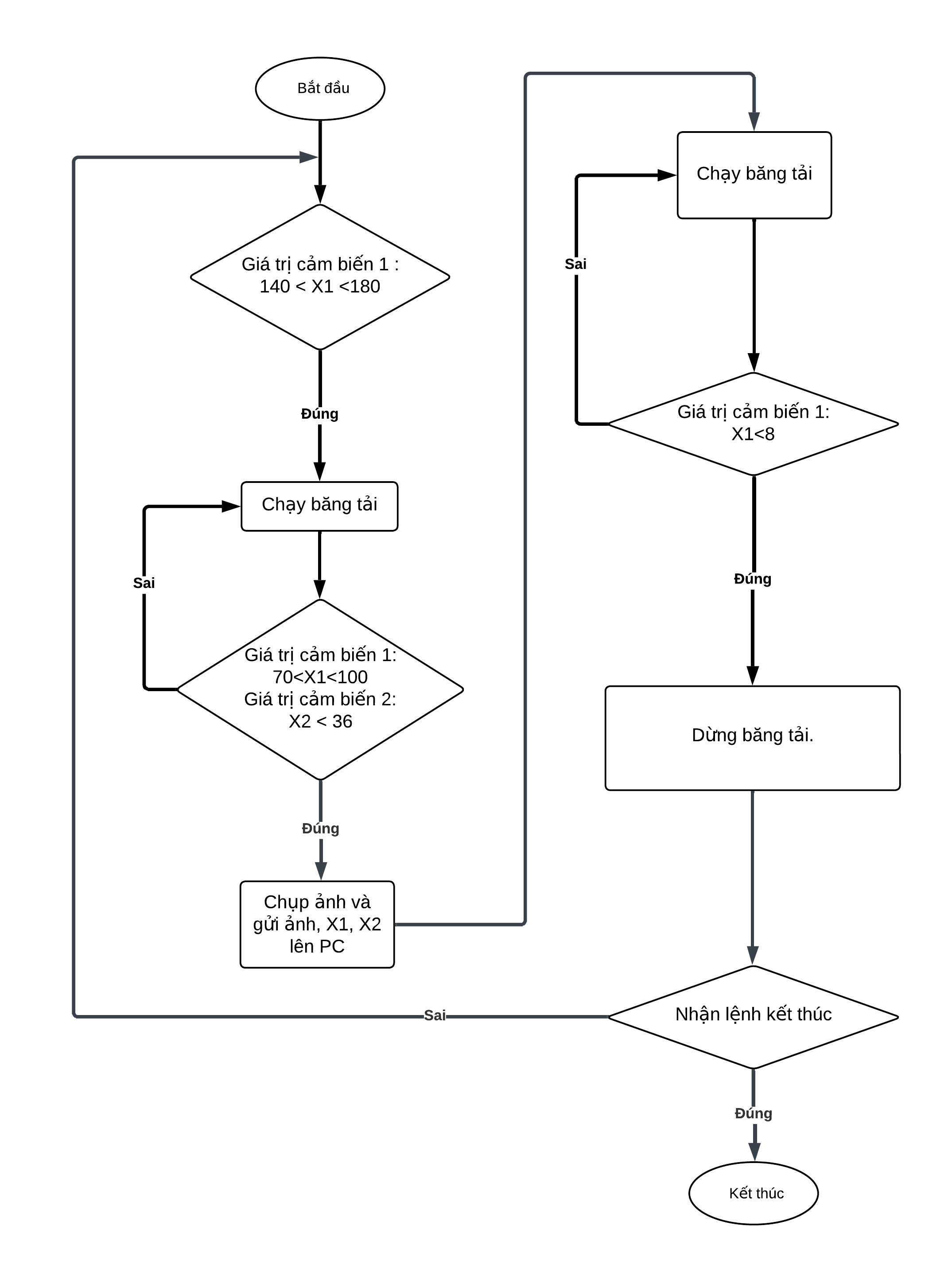
1. Phần điều khiển băng tải và lấy dữ liệu từ cảm biến.

* Sơ đồ kết nối.

Có file riêng trên `zalo

Thiết bị bao gồm:

* Module USART
* PLC S7-1214C
* Module IO
* Module điều khiển băng tải
* Module USART to TTL
* IO-Links AL1100
* Module GT68
* Sensor O5D150
* Lưu đồ điều khiển băng tải.



* Lấy dữ liệu từ cảm biến.

Cảm biến liên kết truyền nhận dữ liệu với PLC thông qua module IO-Link AL1100. Để lấy được dữ liệu trước hết phải khai báo module IO-Link AL1100 và tạo liên kết với PLC.

A screenshot of a computer

Description automatically generated

Khai báo địa chỉ của cảm biến trên module IO-Link AL1100, trong trường hợp này là Port 1 (IW68) và Port 3 (IW71).

A screenshot of a computer

Description automatically generated

Chuyển đổi dữ liệu từ dạng analog sang khoảng cách (cm).

A screenshot of a computer

Description automatically generated

Với:

* IN1: Địa chỉ đầu vào của cảm biến
* IN2: Dữ liệu số bé nhất đo được.
* IN3: Dữ liệu số lớn nhất đo được.
* IN4: Khoản cách lớn nhất đo được.
* IN5: Khoản cách bé nhất đo được.
* IN6: sai số Offset của cảm biến.

Công thức để quy đổi từ dữ liệu sang khoản cách.

D = )