Mục lục

[**I. Tổng quan đề tài** 3](#_Toc106699034)

[**II. Lưu đồ thuật toán** 4](#_Toc106699035)

[**III. Quy trình thực hiện** 5](#_Toc106699036)

[**1. Tách hình ảnh đầu vào trên thành hai phần riêng biệt.** 5](#_Toc106699037)

[**2. Xử lý hình ảnh thước.** 7](#_Toc106699038)

[**3. Xử lý hình ảnh gạo** 9](#_Toc106699039)

[**+ Tiền xử lý hình ảnh gạo** 9](#_Toc106699040)

[**+ Chuyển đổi thành hình ảnh xương, tìm các điểm endpoint và branchpoint (điểm ngã 3)** 10](#_Toc106699041)

[**+ Tìm hướng phát triển của đường thẳng qua endpoint và branchpoint** 12](#_Toc106699042)

[**+ Phát triển endpoint theo hướng đã xác định để tìm kiếm điểm kết nối.** 13](#_Toc106699043)

[**+ Lọc ra từng hạt và đưa ra phân tích dựa theo diện tích và chiều dài hạt** 15](#_Toc106699044)

[**IV. Đánh giá kết quả và phát triển** 15](#_Toc106699045)

[**1. Hình ảnh thước** 15](#_Toc106699046)

[**2. Tiền xử lý hình ảnh gạo** 16](#_Toc106699047)

[**3. Ảnh khung xương** 16](#_Toc106699048)

[**4. Tìm điểm kết nối** 17](#_Toc106699049)

[**5. Lọc hạt** 17](#_Toc106699050)

[**6. Kết quả chung** 18](#_Toc106699051)

# **I. Tổng quan đề tài**

- Với một hình ảnh đầu vào như bên chúng ta cần đưa ra kết quả đánh giá chất lượng hạt gạo, cụ thể như:

+ Đánh giá theo kích thước hạt gạo, đưa ra các tiêu chuẩn hạt dài, hạt ngắn và hạt vỡ.

+ Đánh giá theo màu sắc hạt gạo từ đó đưa ra đánh giá.

# **II. Lưu đồ thuật toán**

Diagram

Description automatically generated

# **III. Quy trình thực hiện**

## **1. Tách hình ảnh đầu vào trên thành hai phần riêng biệt.**

**A picture containing text, measuring stick, blue

Description automatically generated+ Bức ảnh thước.**

**+ Bức ảnh gạo.**

Đầu tiên từ bức ảnh đầu vào, tìm phần hình ảnh chiếm diện tích lớn nhất, đó chính là thước.

Sau đó lấp đầy phần diện tích foreground, ta được một bức ảnh foreground của thước.

Shape

Description automatically generated with low confidence

A picture containing text, measuring stick

Description automatically generated

Hình 1a: Ảnh đầu vào

Hình 1.1a: Hỉnh ảnh thước

Hình 1.1b: Hỉnh ảnh foreground thước

A picture containing text, fireworks, outdoor object

Description automatically generated Đè hình ảnh foreground thước lên hình ảnh ban đầu, ta thu được hình ảnh gạo không chứa thước.

Hình 1.1c: Hình ảnh gạo

**\* Kết quả**

A picture containing text, fireworks, outdoor object

Description automatically generated**A picture containing text, measuring stick, blue

Description automatically generated**A picture containing text, measuring stick

Description automatically generated

Hình 1.1: Các hình ảnh trên bỏ qua một số bước trung gian như lọc nhiễu và chuyển thành ảnh đen trắng

Hình 1.1d: Input

Hình 1.1e: Output

## **2. Xử lý hình ảnh thước.**

Từ bức ảnh thước chúng ta cần trích xuất ra được kích thước pixel dựa theo kích thước thực ( ví dụ như 1cm=… pixel).

Để có thể trích xuất được kích thước từ những vạch kẻ chia là vô cùng khó khăn như, kích thước vạch kẻ nhỏ, mờ, bị bẩn hoặc bị xước trong thực tế.

=> Em thực hiện một cách đơn giản hơn mà vẫn giữ lại được sự hiệu quả đó là từ hình ảnh trên chỉ giữ lại các số trên thước, từ khoảng cách giữa các số có thể tính gần đúng khoảng cách pixel thực tế.

A screenshot of a graph

Description automatically generated with low confidenceA picture containing text, remote, control

Description automatically generated

Hình 1.1f: Hình ảnh thước chỉ giữ lại các số trên thước

Hình 1.1g: Ví dụ về hình ảnh đang lọc ra chữ số 0 trên thước

Với các chữ số quá gần nhau nhỏ hơn một khoảng cách nhất định, ta hiểu rằng đó là 1 số (ví dụ 1 và 0 gần nhau thì đó là số 10).

Khi này ta đã có được vị trí của tất cả các số theo tọa độ, ta có thể dễ dàng tính toán ra khoảng cách giữa 2 số liền kề nhau hay chính là khoảng cách 1cm theo pixel.

## **3. Xử lý hình ảnh gạo**

Xử lý hình ảnh gạo đòi hỏi một quy trình phức tạp hơn, với những bức ảnh mà các hạt gạo chạm nhau cần một quy trình xử lý để tách hạt gạo trước khi đánh giá chúng.

### **+ Tiền xử lý hình ảnh gạo**

Đây là một bước vô cùng quan trọng để lấy dữ liệu cho các bước xử lý sau, sai số tiềm ẩn lớn.

Logo

Description automatically generated with medium confidence Trực tiếp lọc hạt gạo để thu được một số thông số mẫu như chiều dài hạt, diện tích hạt. Để tránh sai số quá lớn, ta chỉ thực hiện trên khoảng 20% số hạt, do thuật toán lọc hạt chạy từ trên xuống dưới nên các hạt cần độ chính xác cao chính ta các hạt ở bên trên.

# Nếu các hạt này thưa thớt và tách nhau thì độ chính xác sẽ cao hơn, nếu số hạt chạm sau nhỏ hơn 5% thì thuật toán vẫn xử lý được, >10% sai số sẽ xảy ra.

Hình 1.1h: 20% số hạt cần lọc

=> Ta thu được hai thông số mẫu là:

+ Diện tích trung bình hạt gạo

+ Chiều dài trung bình hạt gạo

A picture containing text, measuring stick

Description automatically generated

Ta tìm được diện tích thông qua phần pixel trắng và độ dài hạt thông qua đường chéo của phần hình chữ nhật đóng khung trên hạt gạo bên cạnh.

Hình 1.1i: Một hạt mẫu khi đã lọc

### **+ Chuyển đổi thành hình ảnh xương, tìm các điểm endpoint và branchpoint (điểm ngã 3)**

Từ hình ảnh gạo 1.1e bên trên ta trực tiếp mở rộng viền hạt gạo đồng thời thu nhỏ hạt gạo, sau đó apply hình ảnh thu nhỏ hạt gạo lên trên hình ảnh mở rộng viền, ta thu được hình ảnh 1.1j bên dưới.

Sử dụng thuật toán skeletonize, ta thu được hình ảnh xương bên dưới.

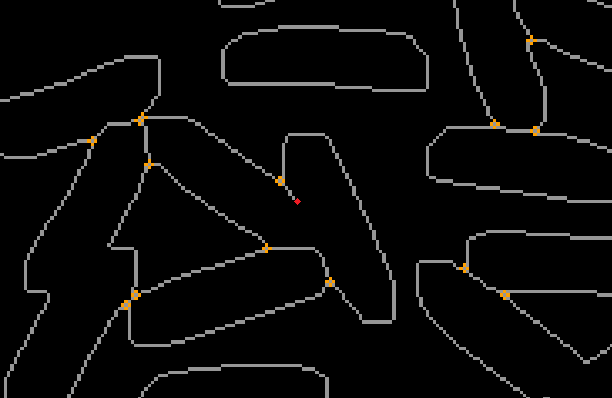
A picture containing text

Description automatically generatedA picture containing text, blackboard

Description automatically generatedThuật toán skeletonize giúp cho phần viền bao quanh hạt gạo chỉ là các đường nối 1 pixel do đó dễ dàng để tìm các điểm endpoint và branchpoint hơn.

Hình 1.1k: Hình ảnh xương của hạt gạo

Hình 1.1j: Hình ảnh mở rộng viền hạt gạo



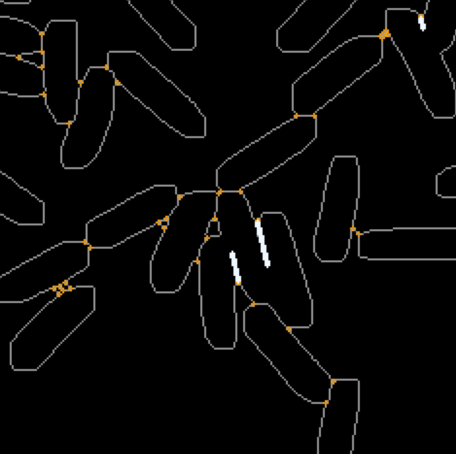
Hình 1.1l: Hình ảnh endpoint màu đỏ và branchpoint màu cam

Dễ dàng có thể nhận thấy các điểm endpoint là các điểm không có điểm nối tiếp theo còn branchpoint chính là các điểm nối ngã 3 đường.

Có thể tìm được các điểm này qua morphology Hit and Miss của opencv nhờ các bộ lọc kernel đã có sẵn.

### **+ Tìm hướng phát triển của đường thẳng qua endpoint và branchpoint**

Có rất nhiều điểm branchpoint trên hình tuy nhiên chúng ta không cần bận tâm đến, chỉ cần quan tâm điểm endpoint sẽ kết nối tới điểm branchpoint nào.

 Sử dụng thuật toán tìm kiếm, xuất phát từ điểm endpoint tìm kiếm theo đường khung xương để tìm điểm branchpoint.

Hình 1.1n: Hình ảnh endpoint kết nối với branchpoint của nó

Vẫn có những trường hợp mà endpoint không thể tìm thấy branchpoint, ta mặc định chỉ cho tìm kiếm trong một khoảng pixel (số lần lặp) nhất định. Theo thử nghiệm thì pixel bằng 0.4 lần chiều dài hạt gạo qua bước tiền xử lý bên trên sẽ cho kết quả tốt nhất. Điểm tìm kiếm cuối cùng ta sẽ mặc định đó là điểm kết nối với endpoint để tìm hướng phát triển.

### **+ Phát triển endpoint theo hướng đã xác định để tìm kiếm điểm kết nối.**

Ta cho endpoint phát triển theo hướng đã tìm trong một khoảng pixel (số lần lặp) nhất định. Theo thử nghiệm thì pixel bằng 0.4 lần chiều dài hạt gạo qua bước tiền xử lý bên trên sẽ cho kết quả tốt nhất.

Shape

Description automatically generated**Diagram

Description automatically generated with medium confidence**

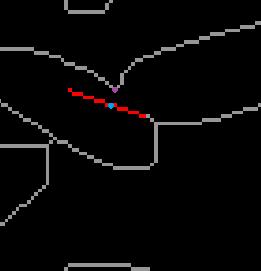
Hình 1.1o: Hình ảnh đường mở rộng gặp đường khung xương

Hình 1.1m: Hình ảnh 2 đường mở rộng gặp nhau (đường thẳng màu trắng sẽ bị loại bỏ trên hình chính)

Với tất cả những đường mở rộng không thể tìm thấy khung xương hay đường mở rộng khác, ta sẽ lưu tất cả các điểm màu xanh (bằng 0.5 nhân số lần lặp) vào trong một mảng.

Với trường hợp 2 đường mở rộng không gặp nhau, tất cả các điểm màu xanh trong mảng đã lưu sẽ tự tìm kiếm các phần tử trong mảng có khoảng cách tới nó < 0.5 nhân số lần lặp. Điểm ở giữa 2 điểm này chính là điểm kết nối. 2 endpoint của 2 đường sẽ kết nối tới điểm này.

Với trường hợp đường mở rộng không gặp được đường khung xương. Điểm màu xanh cũng không tìm kiếm thấy điểm màu xanh nào khác ở gần. Nó sẽ tự động tìm kiếm trong bán kính 0.5 nhân số lần lặp, các điểm có màu đường khung xương gần nhất và kết nối tới đó, trong hình là điểm màu tìm. Điểm endpoint sẽ kết nối tới điểm này.



Hình 1.1q: Hình ảnh đường mở rộng không gặp điểm nào

Hình 1.1p: Hình ảnh 2 đường mở rộng không gặp nhau

**\* Kết quả:**

**Logo

Description automatically generated**

Hình 1.1q: Hình ảnh sau khi cắt

Đã có thể cắt được các cạnh hạt dính nhau, tuy nhiên hình ảnh đầu vào do được chụp ở khoảng cách xa dẫn lên các khe giữa hạt khá nhỏ, khuyến nghị chụp ở khoảng cách gần hơn.

### **+ Lọc ra từng hạt và đưa ra phân tích dựa theo diện tích và chiều dài hạt**

# **IV. Đánh giá kết quả và phát triển**

## **1. Hình ảnh thước**

**­**- Hiện tại thuật toán đang sử dụng là tìm vùng diện tích lớn nhất và mặc định đó là thước, gần như sẽ không sai lệch trừ khi phông nền có quá nhiều nhiễu dẫn đến có nhiều mảng diện tích thừa.

- Thuật toán sẽ lọc đi các thanh chia kích thước, nếu thay đổi loại thước khác có thanh chia dày và to hơn có thể ảnh hưởng đến kết quả lọc.

## **2. Tiền xử lý hình ảnh gạo**

- Thuật toán đang lọc 20% số hạt gạo phía bên trên, nếu số hạt gạo này đưa ra một số liệu tham khảo về chiều dài hạt và diện tích hạt sai lệch quá lớn thì sẽ ảnh hưởng đến kết quả chung.

## **3. Ảnh khung xương**

- Hiện tại đang sử dụng hàm có sẵn, khá tiện lợi tuy nhiên có một số vùng mà ảnh khung xương ko thể hiện được chỗ đó là endpoint và branchpoint. Có thể tìm kiếm thuật toán lọc khung xương tốt hơn.



Hình 4a: Không tìm thấy điểm endpoint và branchpoint

## **4. Tìm điểm kết nối**

**A picture containing chart

Description automatically generated**- Ở đây xảy ra trường hợp, do các đường thẳng tìm kiếm chỉ có độ dày là 1 pixel nên có thể xảy ra trường hợp 2 đường thẳng đi qua nhau và không cắt nhau, thậm chỉ cả trường hợp đường thẳng đi qua khung xuong mà không cắt.

- Thuật toán đang sử dụng sẽ mở rộng theo từng pixel, ở mỗi pixel sẽ kiểm tra xem pixel đó có màu đỏ (đường thẳng khác) hay xám (khung xương) và đưa ra phản hồi.

Hình 4b: 2 đường thẳng không cắt nhau

- Nếu tăng độ dày lên thêm 1 pixel sẽ làm thuật toán lập tức tìm thấy điểm màu đỏ của đường thẳng dẫn đến bị lỗi.

=> Chưa thể khắc phục.

## **5. Lọc hạt**

- Đang sử dụng bộ lọc connectedComponentsWithStats của opencv. Vô cùng mạnh mẽ, ưu điểm hơn canny và findcontour. Tuy nhiên vẫn có một vài trường hợp lỗi xảy ra.

A picture containing text, measuring stick

Description automatically generated

Hình 4c: đường cắt giữa 2 hạt quá nhỏ dẫn đến không lọc được 2 hạt

- Thực tế 2 hạt đã tách nhau tuy nhiên phần cắt giữa 2 hạt quá nhỏ chỉ 1 pixel dẫn đến bộ lọc ko thể xử lý để lọc 2 hạt ra riêng được.

## **6. Kết quả chung**

- Do bộ dữ liệu đầu vào quá nhỏ nên không để kiểm nghiệm và đưa ra kết quả tốt nhất. Một vài thông số dựa trên cảm tính ví dụ như độ dài đường thẳng cắt hạt đang để là bằng 0.4 nhân độ dài hạt. Hiện tại bộ số liệu nhỏ mang lại kết quả khả quan nhất tuy nhiên rất có thể sẽ xảy ra lỗi nếu như không tìm được một hệ số ổn định hơn.