

TRƯỜNG ĐẠI HỌC AN GIANG  
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG MÔ HÌNH MÁY HỌC TRONG  
PHÂN TÍCH HÌNH ẢNH HẠT GẠO

HUỲNH PHÚC DUY

AN GIANG, 05-2024

TRƯỜNG ĐẠI HỌC AN GIANG  
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG MÔ HÌNH MÁY HỌC TRONG  
PHÂN TÍCH HÌNH ẢNH HẠT GẠO

HUỲNH PHÚC DUY  
DTH205748

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN VĂN HÒA

AN GIANG, 05-2024

Khoá luận “Ứng dụng mô hình máy học trong phân tích hình ảnh hạt gạo” do sinh viên Huỳnh Phúc Duy thực hiện dưới sự hướng dẫn của TS. Nguyễn Văn Hòa. Tác giả đã báo cáo kết quả nghiên cứu và được Hội đồng Khoa học và Đào tạo thông qua ngày .....

**Phản biện 1**

*(Ký và ghi rõ chức danh, họ tên)*

**Phản biện 2**

*(Ký và ghi rõ chức danh, họ tên)*

**Giảng viên hướng dẫn**

*(Ký và ghi rõ chức danh, họ tên)*

## LỜI CẢM ƠN

Sau những ngày tháng học tập và rèn luyện tại Trường Đại học An Giang, bằng sự biết ơn và kính trọng em xin gửi lời cảm ơn chân thành đến các quý Thầy Cô trường Đại học An Giang, đặc biệt là quý Thầy Cô và anh chị Khoa Công Nghệ Thông Tin và Viện Biến đổi Khí Hậu đã tạo mọi điều kiện thuận lợi giúp đỡ em trong suốt quá trình học tập, nghiên cứu và hoàn thiện quá trình nghiên cứu.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới Thầy Nguyễn Văn Hòa đã trực tiếp hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài. Ngoài ra em xin cảm ơn Thầy Lê Thanh Phong đã hỗ trợ về quá trình thu thập dữ liệu mẫu để tạo tập dữ liệu trong đề tài này. Nhờ những lời khuyên hữu ích và sự giúp đỡ tận tình của hai thầy giúp em hoàn thiện chủ đề nghiên cứu

Em cũng biết ơn sự quan tâm và ủng hộ từ gia đình và bạn bè đã luôn khích lệ, động viên, giúp đỡ em trong quá trình học tập và nghiên cứu.

Lời cuối cùng em xin chúc quý Thầy Cô, chúc Thầy Nguyễn Văn Hòa, Thầy Lê Thanh Phong và các bạn lời chúc sức khỏe và luôn thành công.

Một lần nữa em xin chân thành cảm ơn!

An Giang, ngày 21 tháng 05 năm 2024

Sinh viên thực hiện

**Huỳnh Phúc Duy**

## TÓM TẮT

Cuộc cách mạng công nghiệp 4.0 đang làm thay đổi toàn diện các ngành nghề trên toàn thế giới, trong đó có nông nghiệp ở Việt Nam, đặc biệt là trong sản xuất lúa gạo, một ngành cực kỳ quan trọng với đất nước. Trong quá trình sản xuất, từ khâu trồng trọt đến thu hoạch và chế biến, công nghệ mới như hệ thống cảm biến IoT và tự động hóa đã được áp dụng để cải thiện hiệu quả và đảm bảo chất lượng. Đáng chú ý, công nghệ phân tích hạt gạo hiện đã được tự động hóa, sử dụng máy học và xử lý ảnh để đánh giá các đặc tính như kích thước, màu sắc, và hình dạng của hạt gạo.

Trước đây, việc phân tích hạt gạo thường được thực hiện thủ công, tốn nhiều thời gian và công sức. Nhưng nhờ vào sự tiến bộ của công nghệ, quá trình này đã được tự động hóa, giúp cải thiện độ chính xác và hiệu quả. Các kỹ thuật học máy và xử lý ảnh như Detectron2, YOLOv8,... được sử dụng để phát triển các mô hình phân tích hạt gạo. Những mô hình này được huấn luyện trên dữ liệu lớn và đa dạng, giúp họ nhận diện và phân tích các đặc điểm của hạt gạo như kích thước, hình dạng, màu sắc, phân loại các hạt gạo gãy và không đạt chất lượng. Việc này không chỉ giúp tiết kiệm thời gian và công sức mà còn nâng cao độ tin cậy và chính xác của kết quả phân tích nhằm hỗ trợ tốt hơn cho quyết định sản xuất và chế biến.

## MỤC LỤC

<b>CHƯƠNG 1: ĐẶT VẤN ĐỀ .....</b>	<b>1</b>
<b>1.1 TÍNH CẤP THIẾT CỦA ĐỀ TÀI.....</b>	<b>1</b>
<b>1.2 PHẠM VI NGHIÊN CỨU.....</b>	<b>2</b>
<b>1.3 MỤC TIÊU CỦA ĐỀ TÀI.....</b>	<b>2</b>
<b>1.4 PHƯƠNG PHÁP NGHIÊN CỨU.....</b>	<b>3</b>
<b>CHƯƠNG 2: TỔNG QUAN CƠ SỞ LÝ THUYẾT .....</b>	<b>4</b>
<b>2.1 MÔ HÌNH HỌC MÁY .....</b>	<b>4</b>
2.1.1. Tổng quan mô hình học máy .....	4
2.1.2. Mô hình YOLOv8 (You Only Look Once version 8).....	4
2.1.3. Mô hình Detectron2 .....	10
<b>2.2 PHÂN TÍCH HẠT GẠO .....</b>	<b>12</b>
<b>2.3 GIỚI THIỆU CÔNG CỤ VÀ MÔI TRƯỜNG TRIỂN KHAI.....</b>	<b>13</b>
2.3.1. Ngôn ngữ lập trình Python .....	13
2.3.2. OpenCV .....	15
2.3.3. Môi trường Google Colab .....	16
2.3.4. Công cụ Roboflow.....	17
2.3.5. K-Fold Cross Validation.....	20
2.3.6. Framework Flask.....	21
<b>CHƯƠNG 3: MÔ HÌNH MÁY HỌC TRONG PHÂN TÍCH HÌNH ẢNH HẠT GẠO.....</b>	<b>23</b>
<b>3.1 MÔI TRƯỜNG VÀ CÔNG CỤ SỬ DỤNG .....</b>	<b>23</b>
<b>3.2 XỬ LÝ VÀ THU THẬP DỮ LIỆU.....</b>	<b>23</b>
3.2.1. Thu thập dữ liệu.....	23
3.2.2. Tạo tập dữ liệu với Roboflow.....	25
<b>3.3 PHÂN TÍCH HẠT GẠO VỚI MÔ HÌNH HỌC MÁY YOLOv8 VÀ DETECTRON2 .....</b>	<b>27</b>
3.3.1. Huấn luyện mô hình .....	27
3.3.2. Đánh giá mô hình bằng phương pháp K-Fold Cross Validation .....	30
3.3.3. Phân tích hạt gạo với Detectron2 và YOLOv8.....	33
<b>3.4 KẾT QUẢ .....</b>	<b>37</b>
3.4.1. Số lượng hạt gạo.....	37

3.4.2 Kích thước hạt gạo.....	41
3.4.3 Phân tích sự khác biệt kích thước giữa các vị trí bông của hạt gạo .....	43
3.4.4 Một số giao diện Web được thực hiện với Flask .....	47
3.5 HƯỚNG PHÁT TRIỂN .....	49
TÀI LIỆU THAM KHẢO .....	50

## DANH SÁCH HÌNH

Hình 1. Sơ đồ kiến trúc mạng YOLO .....	5
Hình 2. Layer trong darknet-53 .....	5
Hình 3. Các feature map của mạng YOLO.....	6
Hình 4. Xác định anchor box cho một vật thể .....	7
Hình 5. Mô tả AoO và AoU .....	8
Hình 6. Công thức ước lượng bounding box từ anchor box.....	9
Hình 7. Non-max suppression .....	9
Hình 8. Dòng thời gian của mô hình YOLO .....	10
Hình 9. Ứng dụng của YOLOv8.....	10
Hình 10. Kiến trúc Detectron2.....	11
Hình 11. TOP 10 Ngôn ngữ lập trình phổ biến nhất thế giới 2023.....	14
Hình 12. Quá trình phân chia K-Fold Cross Validation.....	21
Hình 13. Ảnh đại diện 10 giống gạo thu thập.....	24
Hình 14. Các hạt gạo gãy.....	24
Hình 15. Các hạt gạo không đạt chất lượng .....	25
Hình 16. Bounding hạt gạo tạo tập dữ liệu huấn luyện .....	26
Hình 17. Quy trình phân tích hạt gạo .....	33
Hình 18. Hàm findContours chuyển đổi mặt nạ nhị phân .....	34
Hình 19. Hàm minAreaRect tìm hộp giới hạn nhỏ nhất.....	35
Hình 20. Góc xoay của hàm minAreaRect .....	35
Hình 21. Sự khác biệt mặt nạ giữa hai mô hình .....	42
Hình 22. Minh họa ba vị trí bông .....	43
Hình 23. Giao diện xử lý File .....	47
Hình 24. Giao diện xử lý thư mục .....	47
Hình 25. Giao diện thực thi xử lý File .....	48
Hình 26. Giao diện thực hiện xử lý thư mục .....	48



## DANH SÁCH BẢNG

Bảng 1. Tài nguyên CPU, GPU, TPU của Google Colab.....	17
Bảng 2. Bảng xuất dữ liệu của Roboflow.....	19
Bảng 3. Số hạt gạo của 10 giống gạo được gán nhãn thủ công.....	23
Bảng 4. Thông số thiết lập huấn luyện mô hình Detectron2 .....	28
Bảng 5. Thông số và mô hình thiết lập huấn luyện mô hình YOLOv8 .....	29
Bảng 6. Tổng quan điểm khác biệt các mô hình YOLOv8-seg.....	30
Bảng 7. Số liệu hạt của các loại gạo theo từng Fold trong các tập dữ liệu.....	30
Bảng 8. Thông số Average Precision của hai mô hình Detectron2 và YOLOv8 .....	31
Bảng 9. Số lượng gạo đo được của mô hình YOLOv8.....	37
Bảng 10. Số lượng gạo đo được của mô hình Detectron2.....	38
Bảng 11. Bảng tỉ lệ sai số trong quá trình đếm hạt của hai mô hình .....	39
Bảng 12. Dữ liệu kích thước trung bình hạt gạo trong tập Test qua hai mô hình .....	41
Bảng 13. Số liệu ảnh và số hạt gạo đạt ở ba vị trí đầu, giữa và cuối bông.....	44
Bảng 14. Số liệu kích thước của đầu, giữa và cuối bông của mỗi giống.....	45
Bảng 15. Sai số kích thước của các vị trí bông.....	45

## CHƯƠNG 1: ĐẶT VẤN ĐỀ

### 1.1 TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Cuộc cách mạng công nghiệp 4.0 không chỉ đơn thuần là một xu hướng, mà còn là một biểu tượng của sự chuyển đổi toàn diện trong thế giới hiện đại. Nó mang lại tác động to lớn không chỉ đến môi trường kinh doanh mà còn đến mọi khía cạnh của cuộc sống và xã hội. Trong bối cảnh mà một quốc gia đang phát triển như Việt Nam, với nền nông nghiệp vẫn đóng vai trò cực kỳ quan trọng trong kinh tế, việc sử dụng các công nghệ 4.0 như một công cụ thúc đẩy sự phát triển không chỉ là một lựa chọn mà là một yếu tố không thể thiếu. Trong nền nông nghiệp của Việt Nam, việc áp dụng công nghệ vào sản xuất lúa gạo đang trở thành một xu hướng không thể phủ nhận.

Gạo là lương thực chính của hơn 3,5 tỷ người trên thế giới, chiếm khoảng 60% dân số thế giới.[1] Gạo là một nguồn cung cấp carbohydrate quan trọng, cung cấp năng lượng cho cơ thể. Nó cũng là một nguồn cung cấp chất xơ, vitamin và khoáng chất tốt. Tại Việt Nam gạo có giá trị đặc biệt quan trọng, gạo Việt Nam được đánh giá là có chất lượng cao, thơm ngon dẻo dai. Gạo Việt được xuất khẩu sang hơn 150 quốc gia và vùng lãnh thổ trên thế giới.[2] Gạo Việt Nam có nhiều chủng loại khác nhau về phương pháp nấu, hương vị, mùi vị, năng suất,... phù hợp với khẩu vị và nhu cầu của nhiều người như gạo nếp, gạo tẻ, gạo thơm, gạo lứt,...

Cả hai giai đoạn chính của quá trình sản xuất lúa gạo (gieo trồng và chăm sóc, xử lý sau thu hoạch) đều có thể ứng dụng một cách hiệu quả công nghệ hiện đại, từ sử dụng hệ thống cảm biến IoT giúp nông dân có thể theo dõi và quản lý đất đai và cây trồng trong giai đoạn trồng trọt đến tự động hóa các quy trình sản xuất và kiểm tra chất lượng của giai đoạn thu hoạch và chế biến.

Phân tích hạt gạo là một quá trình cần thiết để đánh giá chất lượng và đặc tính của gạo, đóng vai trò quan trọng trong ngành công nghiệp gạo. Nhiệm vụ này không chỉ bao gồm xác định chất lượng thông qua hàm lượng dinh dưỡng, mà còn qua các yếu tố như màu sắc, kích thước và hình dạng của hạt gạo. Trong quá khứ, phân tích này thường được thực hiện thủ công, một quá trình tốn nhiều công sức và thời gian. Tuy nhiên, nhờ vào những tiến bộ trong công nghệ, bây giờ việc phân tích hạt gạo có thể được tự động hóa với sự trợ giúp của máy móc, giúp cải thiện độ chính xác và hiệu quả của quá trình phân tích.

Khả năng nhận diện đối tượng với máy học đã đạt được những tiến bộ đáng kể trong những năm gần đây. Các mô hình máy học hiện đại có thể nhận diện các đối tượng trong ảnh và video với độ chính xác cao như Fast R-CNN, YOLO, SSD, Bayes,...[3] Các mô hình máy học được sử dụng để nhận diện đối tượng

thường được đào tạo trên một tập dữ liệu lớn các hình ảnh và video có nhãn. Tập dữ liệu này chứa các hình ảnh và video của các đối tượng khác nhau, cùng với nhãn xác định từng đối tượng. Máy học sau đó học cách xác định các đặc điểm của các đối tượng trong tập dữ liệu này và sử dụng các đặc điểm này để nhận dạng các đối tượng trong các hình ảnh và video mới. Vì vậy các kỹ thuật xử lý ảnh được sử dụng để phân loại các hạt gạo không đạt chất lượng và gạo gãy, nứt, đồng thời phân tích các đặc điểm của hạt gạo như màu sắc, kích thước, hình dạng,...

Các kỹ thuật học máy hiện được ứng dụng để phát triển các mô hình phân tích hạt gạo, dựa trên những dữ liệu đã được thu thập sẵn. Những mô hình này được huấn luyện để nhận diện và phân tích các đặc điểm khác nhau của hạt gạo, như kích thước, hình dạng, màu sắc, và thậm chí là các dấu hiệu của bệnh tật hay hư hỏng. Sử dụng dữ liệu từ các hình ảnh và thông tin chi tiết khác, mô hình học máy có thể tự động đánh giá chất lượng gạo, đảm bảo đáp ứng các tiêu chuẩn chất lượng và nâng cao hiệu quả sản xuất. Công nghệ này giúp giảm đáng kể thời gian và công sức cần thiết cho công tác phân tích, đồng thời tăng cường độ tin cậy của kết quả phân tích.

Từ các thông tin trên, ta thấy được tiềm năng đáng kể của xử lý ảnh và mô hình học máy trong việc phân tích gạo từ đó nâng cao chất lượng của quá trình phân tích và giúp các nhà sản xuất đưa ra quyết định tốt hơn trong sản xuất và chế biến gạo.

## **1.2 PHẠM VI NGHIÊN CỨU**

Nghiên cứu ứng dụng mô hình máy học, học sâu trong phân tích, nhận diện, đếm số lượng hạt gạo thông qua hình ảnh.

## **1.3 MỤC TIÊU CỦA ĐỀ TÀI**

Ứng dụng hai mô hình máy học, học sâu trong phân tích, nhận diện, đếm số lượng hạt gạo thông qua hình ảnh. Cụ thể:

- Tìm hiểu các mô hình máy học để ứng dụng phân tích hình ảnh hạt gạo.
- Thu thập, tạo tập dữ liệu hạt gạo và huấn luyện mô hình máy học với tập dữ liệu hạt gạo.
- Đánh giá hiệu quả của kết quả thực nghiệm khi ứng dụng mô hình máy học trong phân tích hạt gạo.

## **1.4 PHƯƠNG PHÁP NGHIÊN CỨU**

Phương pháp tổng hợp tài liệu:

- Thu thập, tìm hiểu, phân tích, nghiên cứu các tài liệu về các mô hình máy học trong nhận diện đối tượng.

Phương pháp nghiên cứu thực nghiệm:

- Thu thập, tạo tập dữ liệu hạt gạo.
- Ứng dụng huấn luyện mô hình máy học với tập dữ liệu hạt gạo.
- Kiểm thử, đánh giá hiệu quả kết quả.

## CHƯƠNG 2: TỔNG QUAN CƠ SỞ LÝ THUYẾT

### 2.1 MÔ HÌNH HỌC MÁY

#### 2.1.1. Tổng quan mô hình học máy

Học máy là một nhánh ngành đang phát triển của các thuật toán tính toán được thiết kế để mô phỏng trí tuệ con người bằng cách học từ môi trường xung quanh.[4] Mô hình học máy là một thuật toán được huấn luyện qua tập dữ liệu khổng lồ để nhận diện, dự đoán và phân loại. Các mô hình này thường được huấn luyện thông qua một quá trình tối ưu hóa, trong đó chúng cố gắng giảm thiểu sai số giữa dự đoán và thực tế.

Có nhiều loại mô hình máy học, bao gồm các mô hình học có giám sát (supervised learning) như Mạng Nơ-ron Convolutions (CNNs) và Mạng Nơ-ron Hồi quy (RNNs), mô hình học không giám sát (unsupervised learning) như K-means clustering và mô hình học tăng cường (reinforcement learning) như Q-learning. Mỗi loại mô hình đều có ưu điểm và ứng dụng riêng trong các tác vụ máy học.

Trong lĩnh vực thị giác máy, các mô hình máy học thường được sử dụng để nhận diện, phân loại và theo dõi đối tượng trong ảnh và video. Các mô hình này thường được huấn luyện trên các bộ dữ liệu lớn như ImageNet, COCO, hoặc Pascal VOC để có khả năng nhận diện chính xác trên nhiều loại đối tượng và trong nhiều điều kiện ánh sáng khác nhau. Trong đó nhận dạng đối tượng được chia làm hai nhóm chính:

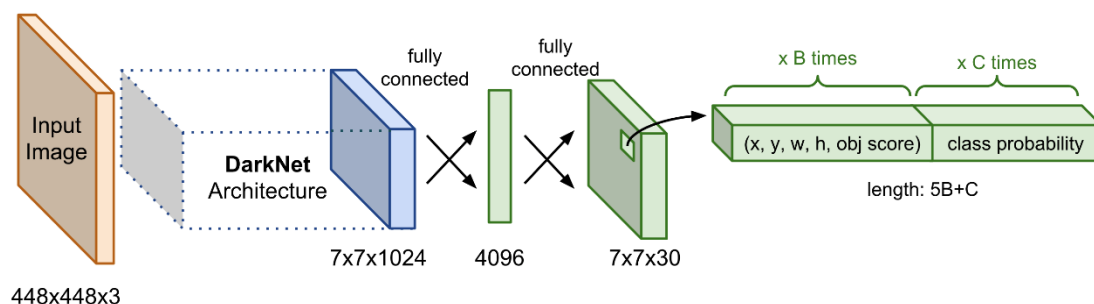
- Họ các mô hình RCNN (Region-Based Convolutional Neural Networks) để giải quyết các bài toán về định vị và nhận diện vật thể.
- Họ các mô hình về YOLO (You Only Look Once) dùng để nhận dạng đối tượng được thiết kế để nhận diện các vật thể real-time.

Đối với chủ đề phân tích hạt gạo thì mô hình học có giám sát để ứng dụng là phù hợp nhất và hai mô hình đại diện cho hai nhóm nhận dạng là YOLOv8 và Detectron2.

#### 2.1.2. Mô hình YOLOv8 (You Only Look Once version 8)

YOLO (You Only Look Once) là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ sự kết hợp các convolutional layer (lớp chập) và connected layers (lớp kết nối). Trong đó các

convolutional layers sẽ trích xuất ra các đặc điểm của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



Hình 1. Sơ đồ kiến trúc mạng YOLO

Trong YOLO version 3 tác giả áp dụng một mạng feature extractor là darknet-53. Mạng này gồm 53 convolutional layers kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả down sample bằng các filter với kích thước là 2. Mạng này có tác dụng giảm thiểu số lượng tham số cho mô hình.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Hình 2. Layer trong darknet-53

Các bức ảnh khi được đưa vào mô hình sẽ được scale để về chung một kích thước phù hợp với input shape của mô hình và sau đó được gom lại thành batch đưa vào huấn luyện.

Mỗi một đầu vào sẽ có một thiết kế các layers riêng phù hợp với shape của input. Sau khi đi qua các layer convolutional thì shape giảm dần theo cấp số nhân là 2. Cuối cùng ta thu được một feature map có kích thước tương đối nhỏ để dự báo vật thể trên từng ô của feature map. Kích thước của feature map sẽ phụ thuộc vào đầu vào. Đối với input 416x416 thì feature map có các kích thước là 13x13, 26x26 và 52x52. Và khi input là 608x608 sẽ tạo ra feature map 19x19, 38x38, 72x72.

Output của mô hình YOLO là một vector gồm:

$$y^T = [p_0, (t_x, t_y, t_w, t_h), (p_1, p_2, \dots, p_c)]$$

Trong đó:

$p_0$  là xác suất dự báo vật thể xuất hiện trong bounding box.

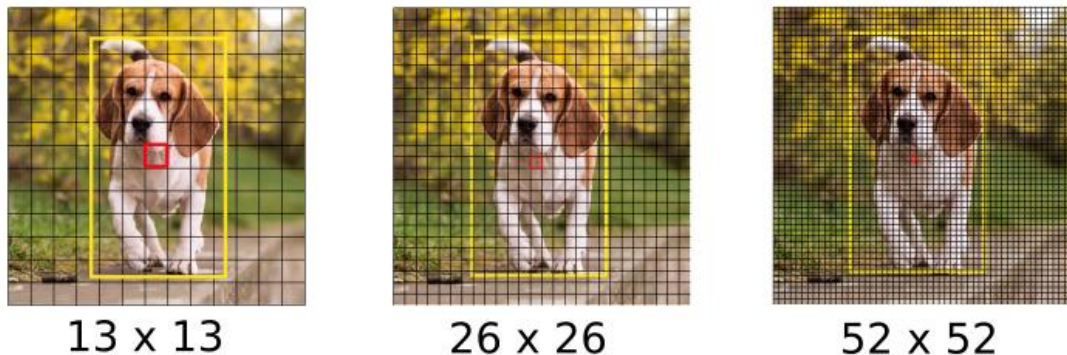
$(t_x, t_y, t_w, t_h)$  giúp xác định bounding box. Trong đó  $t_x, t_y$  là tọa độ tâm và  $t_w, t_h$  là kích thước rộng, dài của bounding box.

$(p_1, p_2, \dots, p_c)$  là vector phân phối xác suất dự báo của các classes.

Output sẽ được xác định theo số lượng classes theo công thức  $(n_{class} + 5)$ , khi huấn luyện 80 classes thì bạn sẽ có output là 85. Trường hợp bạn áp dụng 3 anchors/cell thì số lượng tham số output sẽ là:

$$(n_{class} + 5) \times 3 = 85 \times 3 = 255$$

YOLO dự báo trên nhiều feature map. Những feature map ban đầu có kích thước nhỏ giúp dự báo được các object kích thước lớn. Những feature map sau có kích thước lớn hơn trong khi anchor box được giữ cố định kích thước nên sẽ giúp dự báo các vật thể kích thước nhỏ.



Hình 3. Các feature map của mạng YOLO

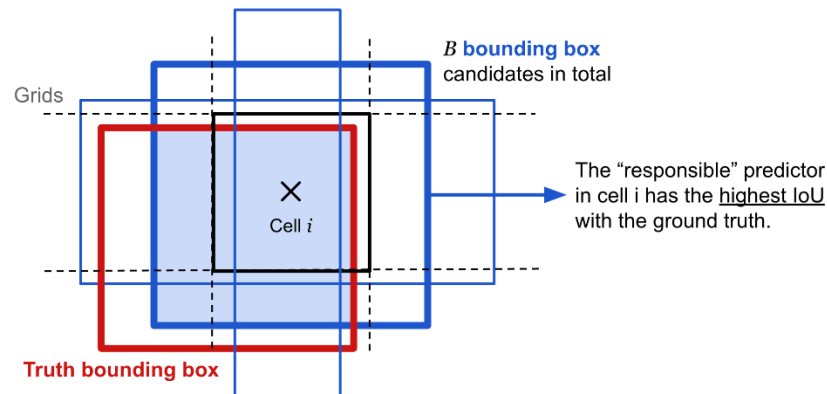
Trên mỗi một cell của các feature map chúng ta sẽ áp dụng 3 anchor box để dự đoán vật thể. Như vậy số lượng các anchor box khác nhau trong một mô hình YOLO sẽ là 9 (3 feature map x 3 anchor box). Đồng thời trên một feature map hình vuông  $S \times S$ , mô hình YOLOv3 sinh ra một số lượng anchor box là:  $S \times S \times 3$ . Như vậy số lượng anchor boxes trên một bức ảnh sẽ là:

$$(13 \times 13 + 26 \times 26 + 52 \times 52) \times 3 = 10647 \text{ (anchor boxes)}$$

Với số lượng anchor boxes vô cùng lớn là nguyên nhân khiến quá trình huấn luyện mô hình YOLO vô cùng chậm khi phải dự báo đồng thời nhãn và bounding box trên đồng thời 10647 bounding boxes.

Để tìm được bounding box cho vật thể, YOLO sẽ cần các anchor box làm cơ sở ước lượng. Những anchor box này sẽ được xác định trước và sẽ bao quanh vật thể một cách tương đối chính xác. Sau này thuật toán regression bounding box sẽ tinh chỉnh lại anchor box để tạo ra bounding box dự đoán cho vật thể.

Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một anchor box. Trong trường hợp có từ 2 anchor boxes trở lên cùng bao quanh vật thể thì ta sẽ xác định anchor box mà có IoU với ground truth bounding box là cao nhất.



Hình 4. Xác định anchor box cho một vật thể

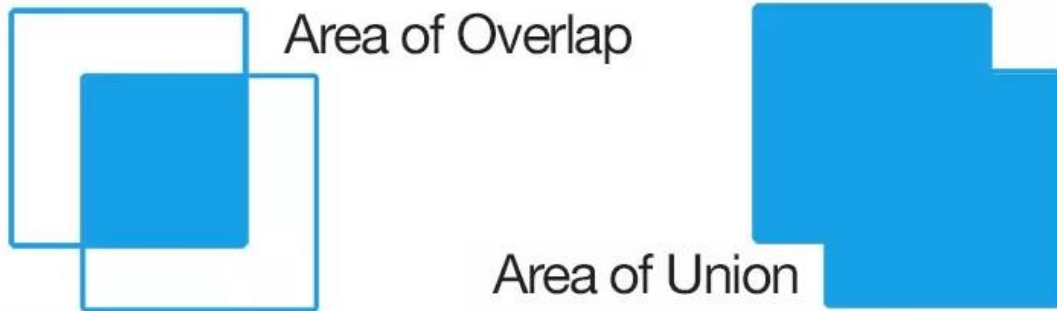
Mỗi một vật thể trong hình ảnh huấn luyện được phân bố về một cell trên feature map mà chứa điểm mid point của vật thể. Khi xác định một vật thể ta sẽ cần xác định 2 thành phần gắn liền với nó là (cell, anchor box). Không chỉ riêng mình cell hoặc chỉ mình anchor box. Một số trường hợp 2 vật thể bị trùng mid point, mặc dù rất hiếm khi xảy ra, thuật toán sẽ rất khó xác định được class cho chúng.

IoU (Intersection over Union) là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể. IoU được tính bằng:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Với Area of Overlap là diện tích phần giao nhau giữa predicted bounding box (hộp giới hạn dự đoán) với growth-truth bounding box (hộp giới hạn thực tế), và Area of Union là diện tích phần hợp giữa predicted bounding box với growth-truth bounding box.



Hình 5. Mô tả AoO và AoU

Những bounding box được đánh nhãn bằng tay trong tập train set và test set. Nếu  $\text{IoU} > 0.5$  thì prediction được đánh giá là tốt.

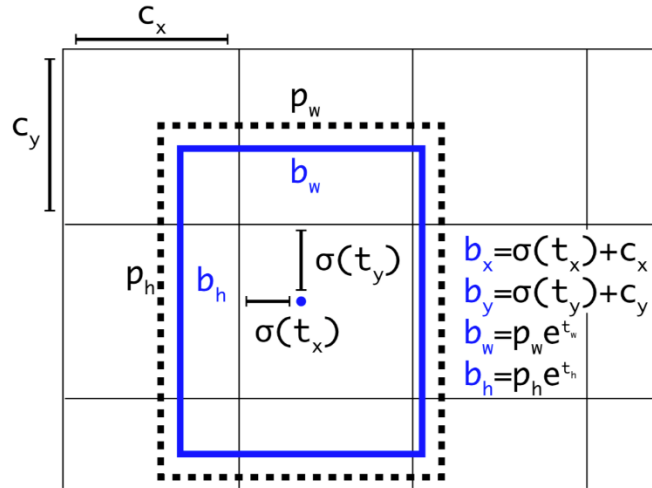
Hàm lỗi (Loss Function) trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính với tổng độ lỗi của 3 thành phần:

- Classification Loss là độ lỗi của việc dự đoán các loại nhãn của object.
- Localization Loss là độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của boundary box  $(x, y, w, h)$ .
- Confidence Loss là độ lỗi của việc dự đoán bounding box chứa object so với nhãn thực tế tại ô vuông đó.

Để dự báo bounding box cho một vật thể chúng ta dựa trên một phép biến đổi từ anchor box và cell. Cho một anchor box có kích thước  $(p_w, p_h)$  tại cell nằm trên feature map với góc trên cùng bên trái của nó là  $(c_x, c_y)$ , mô hình dự đoán 4 tham số  $(t_x, t_y, t_w, t_h)$  trong đó 2 tham số đầu là độ lệch (offset) so với góc trên cùng bên trái của cell và 2 tham số sau là tỷ lệ so với anchor box. Và các tham số này sẽ giúp xác định bounding box dự đoán  $b$  có tâm  $(b_w, b_h)$  thông qua hàm sigmoid và hàm exponential như các công thức:

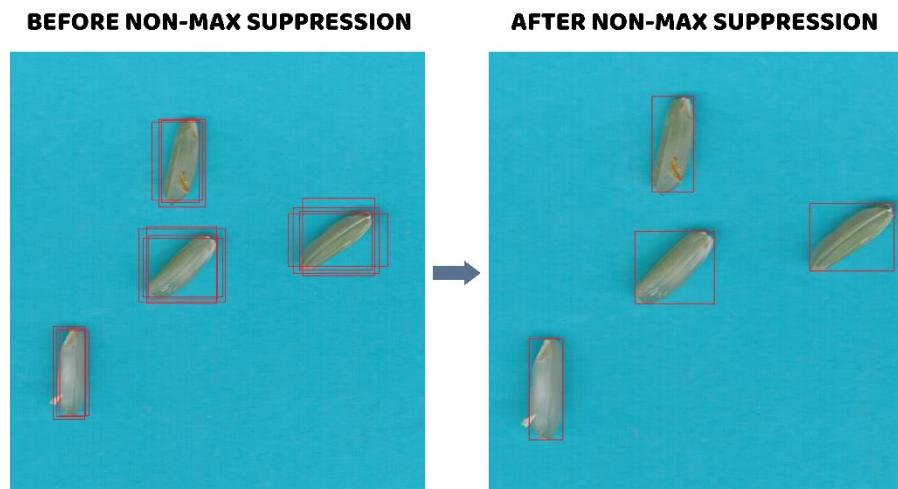
$$b_x = \sigma(t_x) + c_x; b_y = \sigma(t_y) + c_y; b_w = p_w e^{t_w}; b_h = p_h e^{t_h}$$

Các tọa độ đã được hiệu chỉnh theo width và height của bức ảnh nên luôn có giá trị nằm trong ngưỡng  $[0, 1]$ . Khi áp dụng hàm sigmoid giúp ta giới hạn được tọa độ không vượt quá xa các ngưỡng này.



Hình 6. Công thức ước lượng bounding box từ anchor box

Thuật toán YOLO dự báo ra rất nhiều bounding box trên một bức ảnh nên đối với những cell có vị trí gần nhau, khả năng các khung hình bị overlap rất cao. Trong trường hợp đó YOLO cần đến non-max suppression để giảm bớt số lượng các khung hình được sinh ra một cách đáng kể.



Hình 7. Non-max suppression

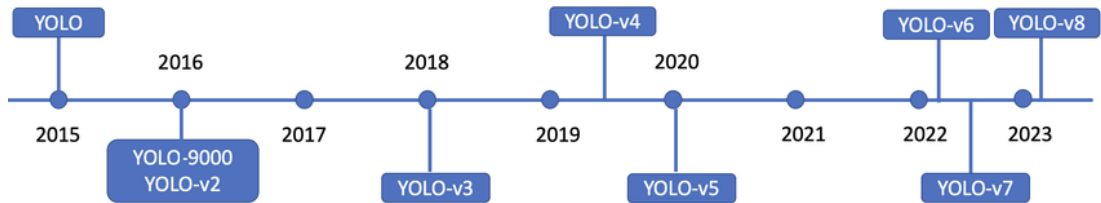
Các bước của non-max suppression:

- Giảm bớt số lượng các bounding box bằng cách lọc bỏ toàn bộ những bounding box có xác suất chứa vật thể nhỏ hơn một ngưỡng threshold, thường là 0.5.
- Với các bounding box giao nhau, non-max suppression chọn ra một bounding box có xác suất chứa vật thể lớn nhất. Sau đó tính toán chỉ số giao thoa IoU với các bounding box còn lại.

Chỉ số giao thoa IoU lớn hơn ngưỡng threshold thì điều đó chứng tỏ 2 bounding boxes đang overlap nhau rất cao. Ta sẽ xóa các bounding có xác suất thấp hơn

và giữ lại bounding box có xác suất cao nhất. Cuối cùng, thu được một bounding box duy nhất cho một vật thể.[5]

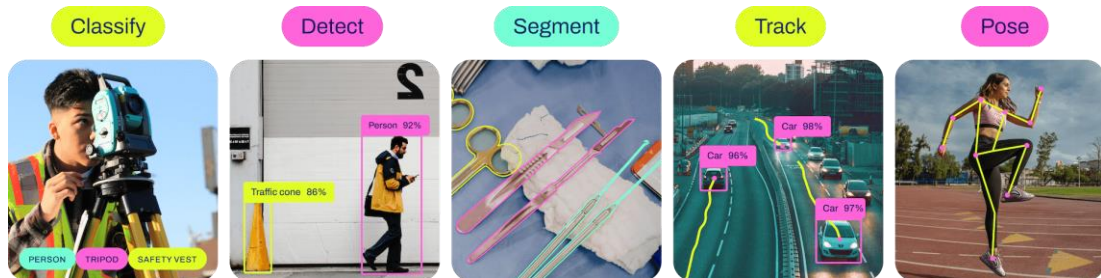
Nhiều phiên bản mới của mô hình đã được giới thiệu kể từ lần phát hành đầu tiên của YOLO vào năm 2015. Mỗi phiên bản được xây dựng nhằm cải tiến phiên bản tiền nhiệm.



Hình 8. Dòng thời gian của mô hình YOLO

Tiếp nối thành công của các phiên bản YOLO trước đây, YOLOv8 ra đời với những cải tiến đột phá, đưa mô hình này lên tầm cao mới trong lĩnh vực nhận diện và theo dõi đối tượng. Phiên bản mới sở hữu nhiều ưu điểm nổi bật như tốc độ xử lý nhanh chóng, độ chính xác cao và dễ sử dụng.

Mô hình YOLOv8 có nhiều ứng dụng đa dạng như nhận dạng, theo dõi đối tượng, phân đoạn ảnh, phân lớp, ước lượng tư thế.[6]



Hình 9. Ứng dụng của YOLOv8

Phân đoạn ảnh là ứng dụng quan trọng nhất của mô hình YOLOv8 cho đề tài phân tích hạt gạo nhằm tách biệt được đối tượng hạt gạo trong ảnh nhằm xác định độ dài, rộng và phân loại chất lượng gạo.

### 2.1.3. Mô hình Detectron2

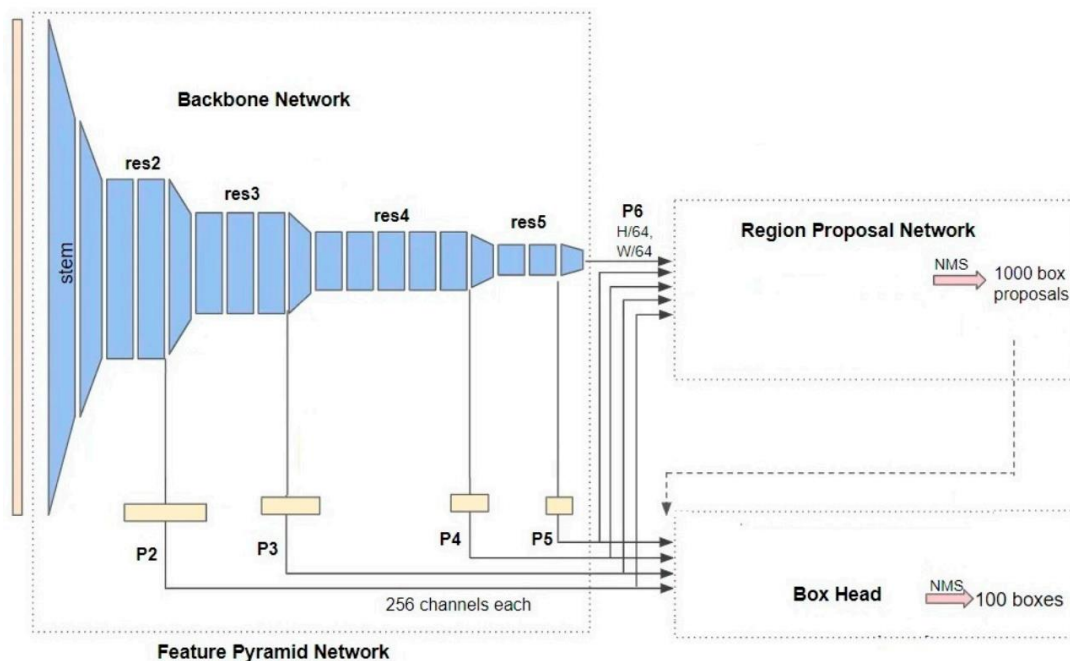
Detectron được Facebook AI Research phát hành vào ngày 9 tháng 10 năm 2019, như một bản nâng cấp cho framework detectron2 gốc. Framework này chủ yếu được thiết kế cho các nhiệm vụ phát hiện đối tượng và phân đoạn thực thể. Detectron2 thực hiện các thuật toán phát hiện tiên tiến như Mask R-CNN, RetinaNet, Faster R-CNN, RPN, TensorMask, PointRend, DensePose và nhiều thuật toán khác. Detectron2 có những ưu điểm vượt trội như tốc độ nhanh hơn, độ chính xác cao hơn, có tính module, có thể tùy chỉnh hơn các thuật toán phát hiện tiên tiến và được xây dựng trên nền tảng PyTorch.

Mô hình nhanh hơn và chính xác hơn vì trong quá trình thực hiện lại các thuật toán tiên tiến, có khả năng Detectron2 sẽ phát hiện ra các phần triển khai không tối ưu hoặc các tính năng lỗi thời từ các phiên bản cũ hơn của các thuật toán này và thực hiện lại chúng. Về tính module, Detectron2 chia việc triển khai của mình thành các phần nhỏ. Các phần bao gồm dữ liệu đầu vào, mạng backbone, các mạng đề xuất vùng và các head dự đoán. Detectron2 có thể tùy chỉnh các thành phần của mô hình có các triển khai tích hợp sẵn, các thành phần đó có thể được tùy chỉnh bằng cách gọi các triển khai mới.

Detectron2 cung cấp các mô hình được tiền huấn luyện với kết quả phát hiện tiên tiến cho các nhiệm vụ thị giác máy tính. Những mô hình này được huấn luyện với nhiều hình ảnh trên các tài nguyên tính toán cao tại phòng thí nghiệm nghiên cứu của Facebook có thể không có sẵn ở các tổ chức khác.

Những mô hình được tiền huấn luyện này giúp phát triển các ứng dụng thị giác máy tính điển hình một cách nhanh chóng mà không cần thu thập, chuẩn bị nhiều hình ảnh hoặc yêu cầu tài nguyên tính toán cao để huấn luyện các mô hình mới. Khi cần phát triển một nhiệm vụ thị giác máy tính trên một lĩnh vực cụ thể với một bộ dữ liệu tùy chỉnh, những mô hình hiện có sẽ là trọng số khởi đầu, và toàn bộ mô hình Detectron2 có thể được huấn luyện lại trên bộ dữ liệu tùy chỉnh.[7]

Detectron2 bao gồm bốn thành phần chính: backbone, neck, Mạng Đề Xuất Vùng (Region Proposal Network) và head.



Hình 10. Kiến trúc Detectron2

Trong kiến trúc của detectron2. Backbone chịu trách nhiệm trích xuất đặc trưng từ ảnh đầu vào, sử dụng các kiến trúc khác nhau như ResNet, ResNeXt và MobileNet. Những kiến trúc này thường được tiền huấn luyện trên các bộ dữ liệu ảnh quy mô lớn như ImageNet. Quá trình tiền huấn luyện bao gồm việc huấn luyện mạng trên nhiều ảnh có nhãn từ ImageNet để học các biểu diễn thị giác chung. ImageNet là một bộ dữ liệu rộng lớn, tận dụng sự tổ chức phân cấp do WordNet, một cơ sở dữ liệu từ vựng, cung cấp. Mạng backbone bao gồm nhiều lớp tích chập được tổ chức theo cấp bậc. Những lớp này dần dần giảm kích thước không gian của các bản đồ đặc trưng trong khi tăng số lượng kênh.

Thành phần neck, được thực hiện như Mạng Lưới Đặc Trưng Hình Kim Tự Tháp (Feature Pyramid Network), tinh chỉnh các bản đồ đặc trưng nhận được từ backbone. FPN kết hợp các đặc trưng từ các quy mô khác nhau để tạo ra một kim tự tháp đặc trưng đa quy mô, cho phép phát hiện các đối tượng ở nhiều kích cỡ và quy mô khác nhau trong ảnh. Kim tự tháp đặc trưng là một yếu tố cơ bản trong các hệ thống nhận dạng giúp phát hiện đối tượng ở nhiều quy mô. Tuy nhiên, một số hệ thống phát hiện đối tượng dựa trên học sâu hiện đại, như YOLO, lại không chứa phần này và được gọi là bộ nhận diện đơn lần.

Sau khi phân tích các đặc trưng đa quy mô, Mạng Đề Xuất Vùng (Region Proposal Network) tạo ra khoảng 1.000 hộp đề xuất, mỗi hộp với một điểm số tin cậy. Những đề xuất này đại diện cho các hộp giới hạn tiềm năng bao quanh các đối tượng quan tâm trong ảnh.

Detectron2 sử dụng một head hộp để cắt và bao bọc các bản đồ đặc trưng thành nhiều đặc trưng cố định kích thước. Tối đa 100 hộp được lọc ra sử dụng non-max suppression.[8]

## 2.2 PHÂN TÍCH HẠT GẠO

Phân tích hạt gạo là quá trình nghiên cứu và đánh giá các thành phần, tính chất, và đặc tính của hạt gạo. Quá trình này thường bao gồm việc phân tích hóa học và vật lý của hạt gạo.

Phân tích hóa học: Bao gồm xác định thành phần dinh dưỡng như protein, carbohydrate, lipid và khoáng chất trong hạt gạo. Các phương pháp phổ biến có thể bao gồm phương pháp hóa lý hoặc phân tích hóa học phức tạp như phổ phân tích hoặc phân tích hóa học định lượng.

Phân tích vật lý: Bao gồm đo lường các tính chất vật lý của hạt gạo như kích thước, hình dạng, độ ẩm, mật độ, độ cứng, và khả năng hấp thụ nước.

Phân tích hạt gạo có thể được thực hiện cho nhiều mục đích khác nhau, từ nghiên cứu khoa học đến kiểm soát chất lượng trong quá trình sản xuất và

chế biến. Các kết quả từ quá trình phân tích có thể cung cấp thông tin quan trọng để cải thiện chất lượng, hiệu suất, và giá trị dinh dưỡng của sản phẩm gạo.

Trong quá trình tham gia thu thập mẫu tại Viện Biến đổi Khí Hậu của trường Đại học An Giang, việc phân tích hạt gạo đã được tiến hành một cách thủ công, từ việc đếm đến đo đạt, bởi các sinh viên thực hiện nghiên cứu tại viện. Tuy nhiên, sự tiến hành này không chỉ tốn kém thời gian mà còn về công sức và nguồn lực, tạo ra các thách thức và rủi ro về độ chính xác và hiệu suất. Điều này không chỉ gây lãng phí tài nguyên mà còn ảnh hưởng đến tính ứng dụng và khả năng mở rộng của quy trình nghiên cứu.

Trong số các ứng dụng của mô hình học máy, phân đoạn ảnh là một phương pháp vô cùng tiềm năng trong việc phân tích hạt gạo. Ứng dụng nhận diện đối tượng của mô hình có thể hoạt động một cách tự động và chính xác. Tuy nhiên, thách thức lớn nhất khi áp dụng phương pháp này là việc xác định bounding box, nơi mà vật thể được giới hạn bởi một hộp chữ nhật. Đối với các hạt gạo có hình dạng không đều và có thể xoay trái phải, việc xác định bounding box theo hướng chính xác của vật thể trở nên khó khăn và không hiệu quả.

Để giải quyết vấn đề này, phân đoạn ảnh vừa có khả năng nhận diện đối tượng và cả việc sử dụng mặt nạ đối tượng. Thay vì chỉ xác định bounding box dựa trên thông tin từ ảnh, mặt nạ đối tượng cho phép chúng ta xác định được ranh giới chính xác của đối tượng, bao gồm cả các chi tiết nhỏ nhất và các biến thể hình dạng. Điều này giúp chúng ta có thể xác định kích thước và hình dạng của hạt gạo một cách chính xác, tăng cường tính đồng nhất và độ tin cậy của quy trình phân tích.

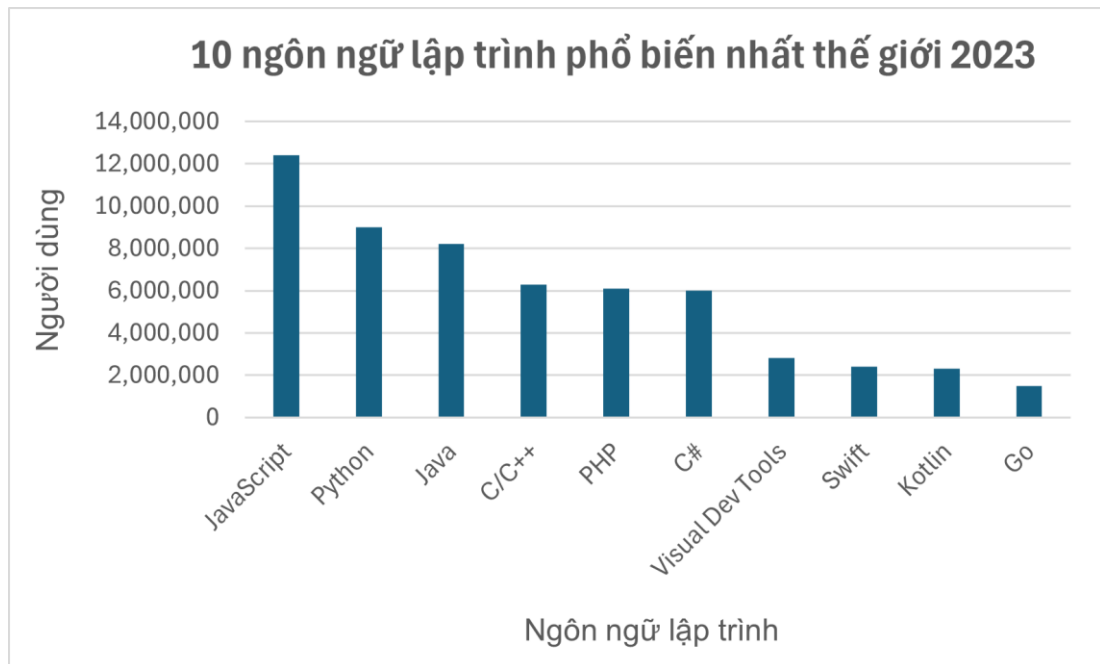
## **2.3 GIỚI THIỆU CÔNG CỤ VÀ MÔI TRƯỜNG TRIỂN KHAI**

### **2.3.1. Ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình bậc cao hướng đối tượng được Guido van Rossum cùng các cộng sự tạo ra năm 1991 dành cho mục đích lập trình đa năng. Python được thiết kế với ưu điểm mạnh là câu lệnh ngắn gọn, dễ nhớ, dễ hiểu. Cấu trúc chương trình của Python rõ ràng, dễ đọc và viết hơn rất nhiều so với những ngôn ngữ lập trình khác (như C, C++, Java, C#, PHP). Do đó Python được coi là một trong những ngôn ngữ thuận tiện nhất cho người mới học lập trình. Đây là ngôn ngữ lập trình thông dịch, có thể chạy trên nhiều hệ điều hành khác nhau như: Windows, Mac OS, OS/2, Linux và các hệ điều hành họ Unix.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python là ngôn ngữ có mã nguồn mở, và đang có cộng đồng người dùng lớn.[9] Thể hiện sự phổ biến

với việc nằm ở vị trí thứ hai trong mười ngôn ngữ lập trình phổ biến nhất thế giới năm 2023.



Hình 11. TOP 10 Ngôn ngữ lập trình phổ biến nhất thế giới 2023

Python có nhiều đặc điểm nổi bật như đơn giản, dễ học; miễn phí, mã nguồn mở; khả năng mở rộng và khả năng nhúng; ngôn ngữ thông dịch cấp cao; thư viện ứng dụng lớn giúp giải quyết công việc phổ biến; hướng đối tượng.[9]

**Đơn giản, dễ học:** Python là một ngôn ngữ đơn giản, không gò bó với cú pháp dễ đọc, học và viết. Giúp lập trình viên có thể tập trung vào giải pháp.

**Miễn phí, mã nguồn mở:** Mang đến sự linh hoạt cho người dùng trong việc sử dụng và chỉnh sửa mã nguồn chương trình, đáp ứng nhu cầu các nhân hóa và phát triển ứng dụng. Cộng đồng người dùng Python rộng lớn luôn sẵn sàng hỗ trợ, chia sẻ kiến thức và kinh nghiệm, tạo môi trường học tập và phát triển hiệu quả cho mọi lập trình viên.

**Khả năng chuyển đổi:** Việc ứng dụng và triển khai Python dễ dàng và thuận tiện nhờ vào khả năng chuyển đổi linh hoạt giữa các hệ điều hành phổ biến như Windows, Mac OS, Linux mà không cần thay đổi cấu trúc mã.

**Khả năng mở rộng và khả năng nhúng:** Lập trình viên có thể kết hợp với mã Python với các ngôn ngữ khác như C, C++ để tăng cường hiệu quả cho ứng dụng giúp phát triển các ứng dụng phức tạp và chuyên biệt.

**Ngôn ngữ thông dịch cấp cao:** Tràn bộ nhớ, dọn dẹp rác hay xóa dữ liệu vô nghĩa sau khi chạy chương trình không còn là vấn đề bận tâm khi sử dụng

Python. Chương trình dịch của Python sẽ tự động dọn dẹp, giải phóng bộ nhớ giúp chúng ta sau khi hoàn thành chương trình, giúp tối ưu hóa và giảm thiểu rủi ro trong quá trình phát triển, đặc biệt là các hoạt động cấp thấp.

Thư viện ứng dụng lớn giúp giải quyết các công việc phổ biến: Các thư viện tiêu chuẩn hỗ trợ đa dạng các công việc phổ biến, giúp tiết kiệm thời gian và công sức cho lập trình viên. Việc sử dụng mã lệnh được kiểm tra kỹ lưỡng và tin cậy bởi cộng đồng phát triển cũng góp phần nâng cao chất lượng và độ ổn định cho chương trình.

Hướng đối tượng: Python hỗ trợ người dùng phương pháp lập trình hướng đối tượng (Object Oriented Programming – OOP), phương pháp này giúp giải quyết các công việc phức tạp một cách trực quan và hiệu quả. Lập trình viên có thể phân tích vấn đề và tạo ra các lớp, đối tượng để giải quyết công việc một cách bài bản và dễ dàng quản lý.

Với những đặc điểm vượt trội và nổi bật đó thì Python là một ngôn ngữ lý tưởng để có thể ứng dụng mô hình máy học và phát triển đề tài nghiên cứu này.

### **2.3.2. OpenCV**

OpenCV (Thư viện thị giác máy tính mã nguồn mở) của phòng thí nghiệm nghiên cứu Intel được chính thức ra mắt vào năm 1999 với mục tiêu là sự tiến bộ trong nghiên cứu về thị giác máy tính và phổ biến kiến thức về thị giác máy tính, là một thư viện với các hàm và mã được tối ưu hóa, dễ dàng sử dụng và miễn phí.

Phiên bản alpha đầu tiên của OpenCV được phát hành công khai tại Hội nghị IEEE về Computer Vision và Nhận dạng mẫu (IEEE Conference on Computer Vision and Pattern Recognition) vào năm 2000, tiếp đó là năm bản beta được phát hành trong giai đoạn từ 2001 đến 2005. Phiên bản 1.0 chính thức đầu tiên được phát hành vào năm 2006. Sau đó, phiên bản 1.1 "bản xem trước" được phát hành vào tháng 10 năm 2008. OpenCV được viết bằng ngôn ngữ lập trình C++, và giao diện chính của nó cũng là C++. Tuy nhiên, OpenCV vẫn duy trì một giao diện C cũ, ít toàn diện hơn mặc dù vẫn rộng rãi. Tất cả các phát triển và thuật toán mới đều xuất hiện trong giao diện C++. OpenCV cung cấp các ràng buộc ngôn ngữ (language bindings) cho Python, Java và MATLAB.[10]

Vào năm 2012, thư viện có hơn 2500 thuật toán được tối ưu hóa và được sử dụng rộng rãi trên toàn thế giới, có hơn 2,5 triệu lượt tải xuống và hơn 40 nghìn người trong nhóm người dùng.[11]



OpenCV có thể ứng dụng vào các lĩnh vực như:

- Bộ công cụ tính năng 2D và 3D
- Ước tính Egomotion (chuyển động của máy ảnh so với ảnh cứng)
- Hệ thống nhận diện khuôn mặt
- Nhận dạng cử chỉ
- Tương tác giữa người và máy (HIC - Human-computer interaction)
- Robot di động
- Hiểu chuyển động
- Phát hiện đối tượng
- Phân đoạn và nhận dạng
- Tầm nhìn âm thanh nổi lập thể (cảm nhận chiều sâu từ 2 camera)
- Cấu trúc từ chuyển động (SFM - Structure from motion)
- Theo dõi chuyển động

Phân đoạn và nhận dạng ảnh là lĩnh vực chính của đề tài vì vậy OpenCV là một công cụ mạnh mẽ và linh hoạt cho ứng dụng phân tích hình ảnh hạt gạo. Tuy nhiên, để sử dụng hiệu quả, người dùng cần có kiến thức chuyên môn về xử lý ảnh, học máy và lập trình.

### **2.3.3. Môi trường Google Colab**

Jupyter Notebook là nền tảng mà Google Colaboratory (Google Colab) dựa trên. Jupiter là một công cụ mã nguồn mở chạy trên trình duyệt tích hợp các ngôn ngữ được thông dịch, thư viện và các công cụ để trực quan hóa dữ liệu. Jupyter Notebook có thể hoạt động cục bộ hoặc trên đám mây. Mỗi tài liệu được cấu thành từ nhiều ô, mỗi ô chứa mã hoặc mã đánh dấu. Kết quả đầu ra là các văn bản, bảng biểu, biểu đồ và đồ họa. Sử dụng công nghệ này giúp việc chia sẻ và sao chép các công trình khoa học dễ dàng hơn vì các thí nghiệm và kết quả được trình bày một cách khép kín.

Google Colaboratory (Google Colab) là một dự án nhằm mục đích phổ biến giáo dục và nghiên cứu về học máy. Google Colab được hoạt động như một đối tượng Google Docs: có thể chia sẻ và người dùng có thể cộng tác trên cùng sổ tay. Google Colab cung cấp cả môi trường chạy Python 2, Python 3 và được cấu hình sẵn với các thư viện học máy và trí tuệ nhân tạo cần thiết như TensorFlow, Matplotlib và Keras. Máy ảo (Virtual Machine) có thời gian chạy và bạn sẽ được cấp khoản thời gian chạy nhất định trong ngày. Sau khi hết thời gian chạy tất cả dữ liệu và cấu hình của người dùng sẽ bị mất. Tuy nhiên, sổ tay người dùng sẽ được bảo lưu và cũng có thể chuyển các tệp từ ổ cứng của máy ảo sang tài khoản Google Drive của người dùng.[12]

Việc ứng dụng mô hình máy học qua phần cứng hạn chế của máy tính cá nhân sẽ không đem lại hiệu quả cao. Bằng việc sử dụng Google Colab không yêu cầu cài đặt hoặc cấu hình máy tính, chỉ yêu cầu bạn có một tài khoản Google thì ta có thể thực thi các chương trình máy học, khoa học dữ liệu thông qua trình duyệt web cung cấp tài nguyên từ máy ảo như CPU tốc độ cao và cả GPUs, TPUs. Dịch vụ cung cấp nhiều loại GPUs, TPUs thường là Nvidia K80s, T4s, P4s and P100s. Tuy nhiên đây là dịch vụ miễn phí nên Google Colab có một số giới hạn nhất định như thứ tự ưu tiên trong việc sử dụng tài nguyên và thời gian sử dụng cho mỗi phiên làm việc tối đa là 12 giờ/ phiên và phải chờ đợi khoảng 24 giờ để khởi động lại thời gian sử dụng GPU, TPUs.

Bảng 1. Tài nguyên CPU, GPU, TPU của Google Colab

CPU	GPU	TPU
Intel Xeon Processor with two core @ 2.30 GHz and 13 GB RAM	Up to Tesla K80 with 12 GB of GDDR5 VRAM, Intel Xeon Processor with two core @ 2.20 GHz and 13 GB RAM	Cloud TPU with 180 teraflops of computation, Intel Xeon Processor with two core @ 2.30 GHz and 13 GB RAM

Ngoài ra Google Colab hiện cũng cung cấp một nền tảng trả phí có tên Google Colab Pro, có giá 9,99\$/tháng. Khi sử dụng, có thể nhận được GPU Tesla T4 hoặc Tesla P100 và tùy chọn chọn một phiên bản có RAM cao khoảng 27 GB. Ngoài ra, thời gian sử dụng tối đa của bạn được nhân đôi từ 12 giờ lên 24 giờ.

Việc sử dụng Google Colab là một lựa chọn thông minh và tiện lợi cho việc ứng dụng mô hình máy học trong phân tích hình ảnh hạt gạo. Không chỉ giúp giảm bớt gánh nặng về phần cứng và cấu hình môi trường làm việc, mà còn cung cấp một môi trường linh hoạt và dễ chia sẻ cho các nghiên cứu và dự án khoa học. Dựa trên sức mạnh của máy ảo cung cấp bởi Google, ta có thể thực thi các chương trình máy học một cách hiệu quả, sử dụng các nguồn tài nguyên như CPU, GPUs và TPUs. Mặc dù có một số hạn chế như thời gian sử dụng và ưu tiên tài nguyên, nhưng với sự linh hoạt và tiện ích mà nó mang lại, Google Colab vẫn là một công cụ quan trọng trong lĩnh vực học máy và phân tích hình ảnh.

#### 2.3.4. Công cụ Roboflow

Roboflow là một hệ thống quản lý quy trình làm việc dựa trên đám mây để phát triển robot tập trung vào dữ liệu và tăng cường AI.[13] Roboflow có sẵn các tập dữ liệu được công khai và quyền truy cập để người dùng tải lên dữ liệu tùy chỉnh của riêng mình. Công cụ hỗ trợ định dạng khác nhau khi xuất dữ liệu và các bước tiền xử lý như hướng hình ảnh, thay đổi kích cỡ và tăng cường dữ liệu.

Cùng với đó Roboflow có một loạt mô hình để huấn luyện, triển khai một cách trực quan và hiện đại như EfficientNet, MobileNet, Yolo, TensorFlow, PyTorch.

Việc đầu tiên để có thể huấn luyện một mô hình máy học đó là cần tập dữ liệu chuẩn xác với mô hình đó. Có thể chọn bất kỳ tập dữ liệu nào phù hợp với đề tài từ những tập dữ liệu có sẵn trên dữ liệu đám mây Roboflow hoặc tải lên tập dữ liệu được thu thập của bản thân từ phần máy tính cá nhân.

Tiếp theo, gán nhãn cho tập dữ liệu đã chọn như vẽ các hộp chữ nhật hoặc các hình đa giác bao quanh đối tượng mà chúng ta muốn nhận dạng hoặc phân đoạn. Sau đó Roboflow sẽ tăng cường và phân chia tập dữ liệu thành “Train, Valid, Test” để tiến hành huấn luyện cho mô hình.

Kế tiếp là bước tiền xử lý ảnh, các bước xác định được thực hiện đối với tất cả hình ảnh trước khi đưa vào mô hình, như kích thước hoặc chuyển đổi hình ảnh sang thang độ xám, tăng hình ảnh (lật, xoay, làm mờ hoặc thêm nhiễu vào hình ảnh) để tổng quát hóa dữ liệu tốt hơn khi triển khai mô hình.

Cuối cùng xuất tập dữ liệu đã được gán nhãn với nhiều định dạng khác nhau dưới dạng file nén hoặc đoạn mã tải dữ liệu. Có rất nhiều định dạng khác nhau cho các mô hình khác nhau được liệt kê cụ thể dưới bảng sau:

Bảng 2. Bảng xuất dữ liệu của Roboflow

Loại File	Định dạng
JSON	COCO Segmentation
	COCO
	COCO-MMDetection
	CreateML
TXT	YOLOv5 Oriented Bounding Boxes
	YOLOv8 Oriented Bounding Boxes
	YOLO Darknet
	YOLO v3 Keras
	YOLO v4 Pytorch
	Scaled – YOLOv4
	meituan/YOLOv6
	YOLO v5 Pytorch
	YOLO v7 Pytorch
	YOLOv8
	YOLOv9
XML	Pascal VOC
CSV	Tensorflow Object Detection
	RetinaNet Keras
	Multi-Label Classification
Khác	Tensorflow TFRecord
	Server Benchmark
	OpenAI Clip Classification

Roboflow hỗ trợ xây dựng một mô hình huấn luyện một các dễ dàng. Với hai tùy chọn chính:

- Train from Scratch: Mô hình sẽ được xây dựng từ đầu với tập dữ liệu được cung cấp.
- Start from a Checkpoint: Yêu cầu một mô hình liên quan với tập dữ liệu, bắt đầu từ một điểm kiểm tra cho phép sử dụng mô hình hiện có làm điểm xuất phát. Tập dữ liệu mẫu sẽ được đào tạo thêm với tập dữ liệu hình ảnh cá nhân. Điều này sẽ tăng tốc quá trình đào tạo và hiệu suất cải thiện, có thể được gọi là huấn luyện chuyển tiếp. Tuy nhiên yêu cầu bắt buộc là phải có mô hình liên quan với tập dữ liệu bản thân.

Roboflow hỗ trợ để tạo tập dữ liệu và huấn luyện nhiều mô hình khác nhau giúp tiết kiệm thời gian, nâng cao hiệu suất để thực hiện. Đặc biệt phù hợp với việc ứng dụng nhiều hơn một mô hình trong phân tích hình ảnh hạt gạo.

### **2.3.5. K-Fold Cross Validation**

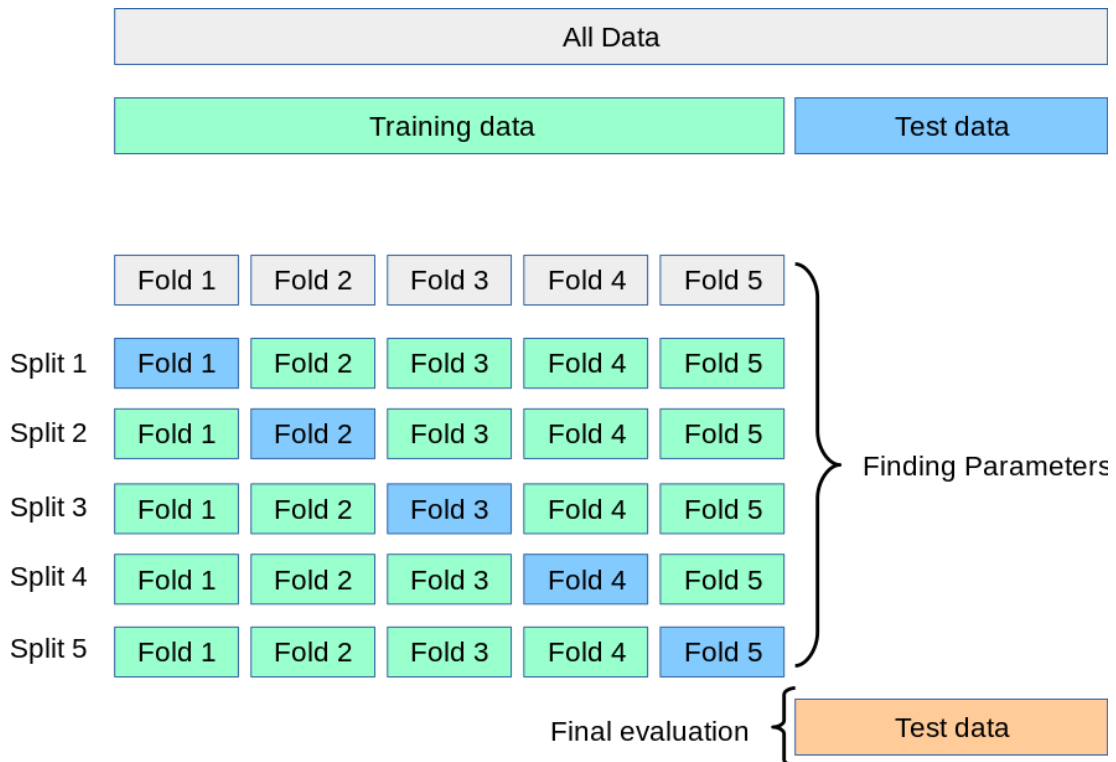
Overfitting là hiện tượng xảy ra khi một mô hình được huấn luyện quá tốt trên dữ liệu huấn luyện, dẫn đến việc nó không thể tổng quát hoá và thực hiện kém trên dữ liệu mới, chưa được quan sát trước. Điều này có thể dẫn đến mô hình bị quá mức phụ thuộc vào các điểm dữ liệu cụ thể trong tập huấn luyện, và không thể đưa ra dự đoán chính xác cho dữ liệu mới. Để ngăn chặn hiện tượng này, các kỹ thuật như cross-validation và regularization thường được sử dụng.

Xác thực chéo (Cross Validation) là một kỹ thuật được sử dụng trong machine learning để đánh giá hiệu suất của một mô hình trên dữ liệu chưa được quan sát trước. Nó bao gồm việc chia dữ liệu có sẵn thành nhiều fold hoặc tập con, sử dụng một trong những fold này làm tập validation, và huấn luyện mô hình trên các fold còn lại. Quá trình này được lặp lại nhiều lần, mỗi lần sử dụng một fold khác nhau làm tập validation. Cuối cùng, kết quả từ mỗi bước validation được trung bình để tạo ra một ước lượng mạnh mẽ hơn về hiệu suất của mô hình.

Mục đích chính của cross validation là ngăn chặn hiện tượng overfitting, xảy ra khi một mô hình được huấn luyện quá tốt trên dữ liệu huấn luyện và thực hiện kém trên dữ liệu mới, chưa được quan sát trước. Bằng cách đánh giá mô hình trên nhiều tập validation, cross validation cung cấp một ước lượng thực tế hơn về hiệu suất tổng quát của mô hình, tức là khả năng của nó trong việc hoạt động tốt trên dữ liệu mới, chưa được quan sát trước.

Có một số loại kỹ thuật cross-validation, bao gồm kỹ thuật chia làm k phần (k-fold cross-validation), kỹ thuật loại bỏ một quan sát (leave-one-out cross-validation), kỹ thuật phân chia ngẫu nhiên (holdout validation), và kỹ thuật phân chia theo loại (stratified cross-validation).[14]

Trong đó K-Fold Cross Validation một trong những kỹ thuật phổ biến nhất. Dữ liệu được chia thành K tập con (folds) có kích thước gần như bằng nhau.



Hình 12. Quá trình phân chia K-Fold Cross Validation

Dữ liệu Test sẽ được tách ra một cách riêng biệt và dành cho bước đánh giá cuối cùng, nhằm kiểm tra "phản ứng" của mô hình khi đối mặt với dữ liệu chưa từng được quan sát trước. Phần dữ liệu Training sẽ được chia ngẫu nhiên thành K fold (K là một số nguyên, thường là 5 hoặc 10). Sau đó, mô hình sẽ được huấn luyện K lần, mỗi lần sử dụng một phần làm dữ liệu validation và K-1 fold còn lại làm dữ liệu huấn luyện. Kết quả đánh giá cuối cùng của mô hình sẽ được tính là trung bình của các kết quả đánh giá từ K fold huấn luyện. Quy trình này giúp đảm bảo rằng chúng ta có được một đánh giá mô hình toàn diện và chính xác hơn, từ đó tạo ra sự khách quan trong việc đánh giá hiệu suất của mô hình.

### 2.3.6. Framework Flask

Flask là một framework nhỏ cho việc phát triển web bằng ngôn ngữ Python. Một framework, đơn giản nhất, là một thư viện hoặc bộ sưu tập các thư viện nhằm giải quyết một phần của một vấn đề chung thay vì một vấn đề cụ thể hoàn chỉnh. Khi xây dựng ứng dụng web, có một số vấn đề mà luôn cần phải giải quyết, như định tuyến từ các URL đến các tài nguyên, chèn dữ liệu động vào HTML, và tương tác với người dùng cuối.[15]

Flask là một framework nhỏ vì nó chỉ triển khai các chức năng cốt lõi (bao gồm cả định tuyến) nhưng để lại các chức năng nâng cao hơn (bao gồm xác thực và ORM của cơ sở dữ liệu) cho các phần mở rộng. Kết quả của điều này là ít hơn sự thiết lập ban đầu cho người dùng lần đầu và nhiều lựa chọn và linh hoạt hơn

cho người dùng có kinh nghiệm. Điều này đối lập với các framework "đầy đủ" hơn, như Django, mà quy định các công nghệ ORM (Ánh xạ đối tượng mã nguồn với bảng) và xác thực của riêng chúng.

Flask có nhiều ưu điểm mà các nhà phát triển có thể xem xét khi quyết định chọn framework này cho dự án của mình. Đầu tiên, nguyên lý microframework và cách thức triển khai giúp Flask Python dễ dàng mở rộng khi cần thiết cho ứng dụng web. Nhờ phần core chạy độc lập và ít phụ thuộc, Flask vẫn có thể đáp ứng được yêu cầu của các ứng dụng web, dù mở rộng ở mức quy mô nào.

Tính linh hoạt là một trong những điểm cốt lõi và ưu điểm của Flask Python. Sự đơn giản của core và các thành phần khác giúp giảm sự phụ thuộc vào nhau, từ đó, dễ dàng điều chỉnh ứng dụng web theo yêu cầu của chủ sở hữu doanh nghiệp. Core của Flask Python rất dễ hiểu với các nhà phát triển và triển khai cũng rất nhanh chóng. Nhà phát triển chỉ cần tập trung vào việc viết code cho đúng với yêu cầu của ứng dụng web. Ngoài ra, Flask được đánh giá là một framework nhẹ, với tài liệu đầy đủ và dễ hiểu cho cả những người mới bắt đầu.

Tuy nhiên, cũng không tránh khỏi nhược điểm, một số là hậu quả của sự đơn giản của nó. Do phần core của Flask được giữ đơn giản, việc cài đặt các thành phần bổ sung khi cần có thể gặp phải một số khó khăn. Mặc dù việc cài đặt từng thành phần khi cần thường không phải là vấn đề, nhưng nếu ứng dụng web yêu cầu nhiều thành phần, việc này có thể tốn thời gian của các nhà phát triển.

## CHƯƠNG 3: MÔ HÌNH MÁY HỌC TRONG PHÂN TÍCH HÌNH ẢNH HẠT GẠO

### 3.1 MÔI TRƯỜNG VÀ CÔNG CỤ SỬ DỤNG

Môi trường: Google Colab, ngoài ra còn có các thư viện như OpenCV, Numpy, Math, OS, Ultralytic, Image, Detectron2, Torch, Matplotlib, Yaml, Skimage

Công cụ: Roboflow

Cấu hình: GPU Intel Xeon Processor with two core @ 2.30 GHz and 13 GB RAM; CPU T4s with 15 GB of GDDR5 VRAM

Framework: Flask

Ngôn ngữ: Python

### 3.2 XỬ LÝ VÀ THU THẬP DỮ LIỆU

#### 3.2.1. Thu thập dữ liệu

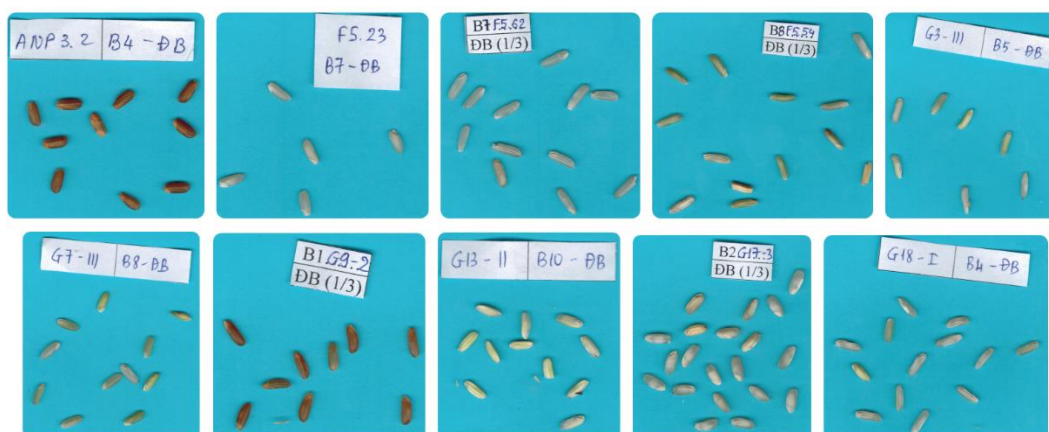
Tập dữ liệu hạt gạo được thu thập tại Viện Biến đổi Khí Hậu tại trường đại học An Giang, các giống gạo được thu thập từ ngân hàng lúa gạo tại viện.

Bảng 3. Số hạt gạo của 10 giống gạo được gán nhãn thủ công

Giống	Số ảnh	Tổng số hạt đạt	Tổng số hạt không đạt	Tổng số hạt gãy
ANP 3.2	30	909	2	11
F5.23		818	10	41
F5.54		489	125	43
F5.62		994	45	42
G3-III		799	24	37
G7-III		829	43	48
G9-II		743	34	65
G13-II		729	15	35
G17-III		657	58	9
G18-1		684	55	23

Tập dữ liệu gạo gồm 10 giống gạo có tất cả 300 ảnh với mỗi giống gồm 30 ảnh và có tất cả 8416 hạt gạo, trong đó có 7651 hạt gạo đạt, 411 hạt gạo không đạt và 354 hạt gạo gãy. Các giống gạo gồm ANP 3.2, F5.23, F5.54, F5.62, G3-III, G7-III, G9-II, G13-II, G17-III, G18-I.





Hình 13. Ảnh đại diện 10 giống gạo thu thập

Việc chọn nhiều giống gạo trong quá trình huấn luyện mô hình máy học là một quyết định có ý nghĩa quan trọng trong nghiên cứu. Bằng cách sử dụng nhiều giống gạo khác nhau có thể tạo ra một tập dữ liệu đa dạng, bao gồm hình dáng, kích thước và màu sắc khác nhau giúp mô hình máy học có thể tổng quát hóa tốt hơn khi đối mặt với dữ liệu mới.

Mô hình sẽ xử lý về khả năng nhận dạng các hạt gạo không đạt chất lượng và gạo gãy trong quá trình xử lý vì vậy cũng cần thu thập cả những loại gạo này trong quá trình lấy mẫu tại viện nghiên cứu.



Hình 14. Các hạt gạo gãy

Gạo gãy (Break) thường là những hạt đã bị vỡ hoặc nứt trong quá trình xử lý hoặc vận chuyển. Điều này có thể xảy ra do các yếu tố như áp lực, va đập, hoặc sự cố trong quá trình sản xuất. Gạo gãy thường không được coi là chất lượng cao vì có thể gây mất mỹ quan và có thể ảnh hưởng đến trải nghiệm ẩm thực.



Hình 15. Các hạt gạo không đạt chất lượng

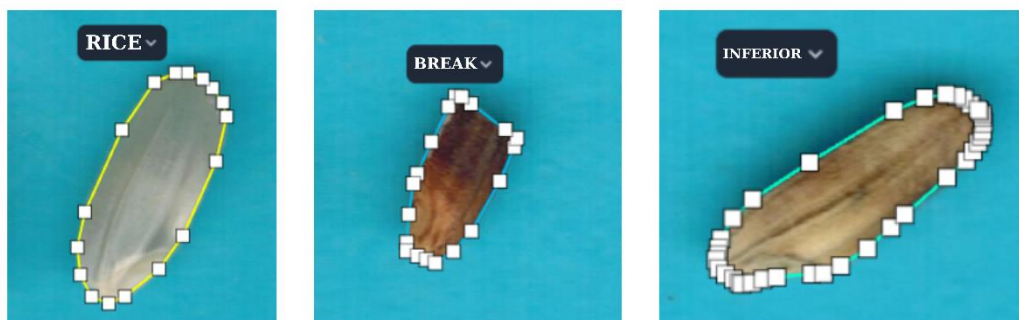
Gạo không đạt (Inferior), ngược lại, là các hạt gạo có chất lượng thấp và không đạt tiêu chuẩn để được coi là gạo tốt. Các tiêu chí để phân biệt gạo không đạt có thể bao gồm hạt bị hỏng, không đều màu, không có vẻ đẹp mỹ quan, hoặc không đạt yêu cầu về kích thước và hình dáng. Nguyên nhân của gạo không đạt có thể là do quá trình sản xuất không hoàn hảo, xử lý không đúng cách, hoặc do các yếu tố khác như điều kiện thời tiết hay quy trình vận chuyển.

Qua quá trình thu thập dữ liệu này đã giúp tạo được một tập dữ liệu đủ đa dạng và cung cấp cho mô hình thông tin đầy đủ để có thể phân biệt giữa các hạt gạo đạt chất lượng với hạt gạo gãy và hạt gạo kém chất lượng. Điều này đã cung cấp một cơ sở tốt cho mô hình để học và hiểu được các đặc điểm quan trọng để phân loại chính xác.

### 3.2.2. Tạo tập dữ liệu với Roboflow

Sau khi thu thập tập dữ liệu tất cả ảnh sẽ được tiến hành tải lên Roboflow. Bước đầu trong quá trình xử lý là phân loại và gán nhãn cho từng hạt gạo trong ảnh. Gạo gãy và gạo không đạt là hai loại hạt gạo có chất lượng không tốt, thường được phân biệt dựa trên các tiêu chí như hình dáng, màu sắc, kích thước, và độ nguyên vẹn của hạt.

Cả hai loại này đều là kết quả của quá trình sản xuất và xử lý không hoàn hảo. Đối với các nhà sản xuất và người tiêu dùng, vì vậy ta phải phân loại trong quá trình gán nhãn và bounding tập dữ liệu để huấn luyện mô hình.



Hình 16. Bounding hạt gạo tạo tập dữ liệu huấn luyện

Khác với bounding box của việc ứng dụng detection, để có thể ứng dụng segmentation cần tạo tập dữ liệu phù hợp để huấn luyện mô hình bằng bounding theo hình dáng của đối tượng ở đây là hạt gạo. Việc này giúp xác định được rõ ràng phạm vi của các đối tượng cần phân đoạn trong hình ảnh. Điều này không chỉ giúp định rõ ranh giới giữa các vật thể, mà còn tạo ra một tập dữ liệu đa dạng với các vật thể có kích thước, hình dạng và đặc tính khác nhau nhằm đảm bảo rằng mỗi đối tượng trong hình ảnh được định vị và đánh dấu đúng vị trí, từ đó giảm thiểu nhiều và sai sót trong tập dữ liệu huấn luyện. Điều này làm tăng độ chính xác và hiệu suất của mô hình sau khi huấn luyện.

Khi đã hoàn thành quá trình bounding vật thể một cách thành công, bước tiếp theo là xuất tập dữ liệu cần thiết để huấn luyện mô hình. Cấu trúc chính của tập dữ liệu này được tổ chức trong các thư mục riêng biệt, được chia thành các phần quan trọng như train, valid và test đại diện cho thành phần chính của tập dữ liệu như tập huấn luyện (training set), tập xác thực (validation set), và tập kiểm tra (test set). Mỗi thư mục này sẽ chứa các file ảnh cùng với các file metadata, chứa các thông tin quan trọng về vị trí và đặc điểm của các vật thể trong hình ảnh.

Các file metadata này thường bao gồm tọa độ hộp giới hạn, nhãn lớp, và mặt nạ của mỗi đối tượng trong từng file ảnh. Dữ liệu này rất quan trọng để mô hình có thể hiểu được các đối tượng trong hình ảnh và học được từ chúng.

Mặc dù có thể sử dụng các mô hình khác nhau như YOLO hay Detectron2, cấu trúc của các file metadata vẫn giữ nguyên các thông tin cần thiết. Ví dụ, đối với YOLO, các thông tin này thường được lưu trong các file có đuôi txt, trong khi đối với Detectron2, chúng có thể được lưu dưới định dạng json. Tuy nhiên, bất kể định dạng nào được sử dụng, các file metadata vẫn chứa các thông tin quan trọng như tọa độ, nhãn lớp và mặt nạ, giúp mô hình hiểu rõ hơn về dữ liệu và cải thiện khả năng nhận diện và phân loại các đối tượng.

Tập dữ liệu được chia thành ba phần chính: tập huấn luyện (training set), tập xác thực (validation set), và tập kiểm tra (test set).

Tập huấn luyện được sử dụng để "dạy" hay huấn luyện mô hình. Thông qua quá trình học từ tập dữ liệu này, mô hình tìm cách hiểu và nắm bắt các mối quan hệ hay mô hình tiềm ẩn giữa các đặc điểm đầu vào và đầu ra mong muốn. Chiếm tỉ lệ lớn trong tập dữ liệu để đảm bảo đủ thông tin trong quá trình học.

Tập xác thực được sử dụng để tinh chỉnh các tham số của mô hình (như tốc độ học) và lựa chọn kiến trúc mô hình tốt nhất. Quá trình này giúp phát hiện và ngăn chặn tình trạng quá mức (overfitting), nơi mô hình học "gắn kết" quá mức với dữ liệu huấn luyện mà không thể áp dụng được với dữ liệu mới.

Tập kiểm thử được sử dụng để đánh giá hiệu suất của mô hình sau khi nó đã được huấn luyện, cung cấp một đánh giá khách quan về hiệu suất của mô hình trên dữ liệu chưa từng thấy. Điều này giúp kiểm tra khả năng tổng quát hóa của mô hình đối với dữ liệu mới. Tập kiểm thử được tách biệt hoàn toàn từ quá trình huấn luyện và không được sử dụng ở bất kỳ giai đoạn nào trong việc huấn luyện mô hình.

### **3.3 PHÂN TÍCH HẠT GẠO VỚI MÔ HÌNH HỌC MÁY YOLOv8 VÀ DETECTRON2**

#### **3.3.1. Huấn luyện mô hình**

Thiết lập thông số là quá trình quan trọng nhất trong quá trình huấn luyện mô hình detectron2. Sử dụng module DefaultTrainer để tạo ra một bộ huấn luyện mặc định từ thư viện detectron2.engine và chỉ định vị trí lưu OUTPUT của quá trình huấn luyện. Nạp cấu hình của mô hình mask\_rcnn\_R\_50\_FPN\_3x từ Model Zoo của Detectron2 vào đối tượng cấu hình cfg. Tiếp theo chỉ định tập dữ liệu sử dụng để huấn luyện mô hình.

Thiết lập trọng số khởi tạo cho mô hình., dùng trọng số đã được đào tạo trước của mô hình mask\_rcnn\_R\_50\_FPN\_3x trên tập dữ liệu COCO. Việc sử dụng trọng số đã được đào tạo trước có thể giúp mô hình bắt đầu từ một điểm xuất phát tốt hơn và hội tụ nhanh hơn trong quá trình huấn luyện. Trọng số đã được đào tạo trước được huấn luyện trên một tập dữ liệu lớn do đó chúng chứa thông tin quý báu về các đặc trưng của các đối tượng trong hình ảnh và cách phân biệt chúng từ các phần khác của hình ảnh. Điều này giúp mô hình dễ dàng học được cách phát hiện và phân loại các đối tượng trong ảnh mới mà nó chưa từng thấy trước đó.

Các thông số thiết lập sẽ thể hiện trong bảng sau:

Bảng 4. Thông số thiết lập huấn luyện mô hình Detectron2

Thông số liệu lập	Ý nghĩa	Số liệu
cfg.DATALOADER.NUM_WORKERS	Số lượng các tiến trình worker được sử dụng để tải dữ liệu trong quá trình huấn luyện.	2
cfg.SOLVER.IMS_PER_BATCH	Số lượng hình ảnh trên mỗi batch trong quá trình huấn luyện.	2
cfg.SOLVER.BASE_LR	Thiết lập tốc độ học ban đầu cho quá trình tối ưu hóa trong quá trình huấn luyện.	0.00025
cfg.SOLVER.MAX_ITER	Số lượng vòng lặp tối đa cho quá trình huấn luyện.	5000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE	Thiết lập kích thước mẫu (batch size) cho mỗi hình ảnh trong mỗi lô (batch) cho phần đầu của mô hình đầu nhận dạng vùng quan trọng (ROI).	512
cfg.MODEL.ROI_HEADS.NUM_CLASSES	Đặt số lượng lớp cần dự đoán trong phần đầu nhận dạng vùng quan trọng (ROI) của mô hình. Ở đây có hai hộp chính do ở lưu file huấn luyện ở dạng COCO có một supper category là rice-segmentation và lớp nhận dạng là Rice, Inferior và Break nên số lớp là 4.	4

Sau khi thiết lập các thông số, một bộ huấn luyện mới được tạo ra từ các cấu hình đã được xác định, và quá trình huấn luyện sẽ bắt đầu.

Đối với mô hình YOLOv8 quá trình huấn luyện không phức tạp như Detectron2 khi phải thiết lập nhiều thông số. Việc cần chú ý là các đường dẫn đến thư mục chứa trọng số sẵn có của YOLOv8 để tiếp tục huấn luyện, nhiệm vụ (task), chế độ (mode), số lượng vòng lặp (epoch), kích thước ảnh và vị trí của file “data.yaml”, một file cấu hình định nghĩa các thông số và đường dẫn cho quá trình huấn luyện và kiểm tra mô hình giúp định rõ cấu trúc và thông tin về bộ dữ liệu, giúp quá trình huấn luyện và đánh giá mô hình được thực hiện một cách tự động và hiệu quả.

Bảng 5. Thông số và mô hình thiết lập huấn luyện mô hình YOLOv8

Thông số và mô hình thiết lập	
Task	Segment
Mode	Train
Model	YOLOv8s-seg
Epoch	250
Imgsz (kích thước ảnh)	800

Mô hình có nhiều ứng dụng như nhận dạng, phân đoạn, phân lớp, theo dõi, và nhận dạng tư thế. Do mục tiêu chính của việc ứng dụng mô hình vào phân tích hạt gạo nên phân đoạn ảnh là mục tiêu chính để thực hiện tính toán chiều dài và chiều rộng của hạt.

Phân đoạn ảnh là một ứng dụng mở rộng của YOLO từ việc chỉ phát hiện đối tượng thành phân đoạn ảnh. Trong khi phát hiện đối tượng tập trung vào việc xác định vị trí và loại của các đối tượng trong một hình ảnh, phân đoạn tiếp tục để cung cấp thông tin chi tiết hơn về hình dạng và bố cục chính xác của mỗi đối tượng hoặc lớp đối tượng. Phân đoạn ảnh có nhiều loại mô hình như YOLOv8n-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg, và YOLOv8x-seg.

Tổng quan các mô hình:

- YOLOv8n-seg tập trung vào việc cung cấp một mô hình nhỏ gọn để triển khai và sử dụng. Mô hình có kích thước nhỏ, phù hợp cho các ứng dụng yêu cầu tài nguyên tính toán ít và tốc độ xử lý cao.
- YOLOv8s-seg cân bằng giữa kích thước mô hình và hiệu suất. Mô hình vừa vặn, cung cấp mô hình vừa vặn cung cấp một lựa chọn trung bình về tốc độ và độ chính xác.
- YOLOv8m-seg cung cấp sự cân bằng tốt giữa tốc độ và độ chính xác, với thiên hướng nghiêng về độ chính xác. Mô hình yêu cầu phần cứng tốt hơn so với các phiên bản nhỏ hơn và không phải lựa chọn tối ưu cho các ứng dụng IoT.
- YOLOv8l-seg có độ chính xác cao cho các nhiệm vụ phức tạp và phân đoạn chính xác. Tuy nhiên mô hình có tốc độ xử lý chậm hơn so với các phiên bản nhỏ hơn không phù hợp với các ứng dụng thời gian thực.
- YOLOv8x-seg là một mô hình cực kỳ mạnh mẽ với độ chính xác cao nhất và khả năng phân đoạn tốt nhất. Mô hình yêu cầu phần cứng cực kỳ cao và có tốc độ xử lý chậm nhất so với các phiên bản khác.

Bảng 6. Tổng quan điểm khác biệt các mô hình YOLOv8-seg

Mô hình	Kích thước	Tốc độ	Độ chính xác
YOLOv8n-seg	Nhỏ	Nhanh	Thấp
YOLOv8s-seg	Nhỏ	Nhanh	Trung bình
YOLOv8m-seg	Trung bình	Trung bình	Cao
YOLOv8l-seg	Lớn	Chậm	Cao
YOLOv8x-seg	Lớn	Rất chậm	Rất cao

### 3.3.2. Đánh giá mô hình bằng phương pháp K-Fold Cross Validation

Cả hai mô hình đều sẽ được đánh giá thông qua phương pháp K-Fold Cross Validation với  $K=5$ . Tập Test sẽ được để riêng nhằm kiểm tra thực tế sau khi huấn luyện mô hình. Còn tập dữ liệu còn lại được chia thành 5 phần bằng nhau, được gọi là các "fold". Mỗi lượt trong quá trình này, một trong số 5 fold sẽ được dành riêng để kiểm tra hiệu suất của mô hình (Valid Set), trong khi 4 fold còn lại sẽ được sử dụng để huấn luyện mô hình (Train Set).

Tại mỗi lần huấn luyện, mô hình sẽ được điều chỉnh và tinh chỉnh trên tập dữ liệu huấn luyện, sau đó được đánh giá dựa trên tập kiểm tra. Quá trình này được lặp lại 5 lần, mỗi lần sử dụng một fold khác nhau làm Valid Set. Kết quả cuối cùng của mô hình được tính toán dựa trên sự kết hợp của tất cả các lượt đánh giá, đảm bảo rằng mô hình được đánh giá trên một loạt các tập dữ liệu kiểm tra khác nhau.

Kết quả cuối cùng của mô hình là trung bình của kết quả từ tất cả các lượt đánh giá. Điều này đảm bảo rằng mô hình được đánh giá trên một loạt các tập dữ liệu kiểm tra khác nhau, không chỉ dựa trên một lựa chọn cụ thể của dữ liệu kiểm tra. Đồng thời, quá trình này cung cấp một cái nhìn tổng thể về hiệu suất của mô hình trên dữ liệu mới, từ đó giúp nhận biết và điều chỉnh mô hình một cách chính xác, hiệu quả. Thông số của các Fold được thể hiện trong bảng sau:

Bảng 7. Số liệu hạt của các loại gạo theo từng Fold trong các tập dữ liệu

Số hạt các Fold		Spit 1	Spit 2	Spit 3	Spit 4	Spit 5
Train	Rice	5626	5407	5642	5567	5562
	Inferior	298	313	292	344	285
	Break	256	245	259	263	257
Valid	Rice	1325	1544	1309	1384	1389
	Inferior	85	70	91	39	98
	Break	64	75	61	57	63



Test	Rice	700
	Inferior	28
	Break	34

Average Precision là chỉ số được đo bằng tỷ lệ diện tích phủ của các bounding box hoặc phân đoạn đối tượng và diện tích thực. Có ba mức độ IoU được xem xét 0.50, 0.75 và 0.50:0.95 (từ 0.50 đến 0.95 với bước nhảy 0.05). Kết quả của mỗi mức độ IoU được tính riêng cho các đối tượng ở kích thước khác nhau (small, medium, large) và cho tất cả các đối tượng (area=all).

Average Precision (AP) là một phép đo quan trọng để đánh giá hiệu suất của mô hình trong các bài toán phân loại, đặc biệt là khi có sự không cân bằng trong các lớp dữ liệu. Khi ta tính Average Precision của một mô hình, ta có thể biết được mức độ chính xác và độ chính xác trung bình của các dự đoán của mô hình trên toàn bộ các lớp. AP có thể đánh giá chất lượng của dự đoán, hiệu suất của mô hình trên dữ liệu không cân bằng, độ tin cậy của mô hình.

Bảng 8. Thông số Average Precision của hai mô hình Detectron2 và YOLOv8

Average Precision			Spit 1	Spit 2	Spit 3	Spit 4	Spit 5	TB
Detectron2	bbox	Rice	0.91	0.9	0.91	0.92	0.91	0.91
		Inferior	0.57	0.6	0.52	0.46	0.45	0.52
		Break	0.52	0.51	0.64	0.56	0.48	0.54
		All	0.66	0.67	0.69	0.65	0.61	0.66
	segm	Rice	0.92	0.92	0.92	0.93	0.93	0.92
		Inferior	0.56	0.61	0.52	0.45	0.47	0.52
		Break	0.53	0.52	0.64	0.57	0.5	0.55
		All	0.67	0.68	0.7	0.65	0.63	0.67
YOLOv8	bbox	Rice	0.96	0.95	0.96	0.97	0.97	0.96
		Inferior	0.73	0.77	0.77	0.58	0.68	0.71
		Break	0.83	0.77	0.85	0.79	0.83	0.81
		All	0.84	0.83	0.86	0.78	0.83	0.83
	segm	Rice	0.88	0.87	0.88	0.88	0.87	0.88
		Inferior	0.66	0.68	0.72	0.5	0.62	0.64
		Break	0.73	0.68	0.76	0.72	0.75	0.73
		All	0.75	0.74	0.78	0.7	0.74	0.74



Detectron2 thể hiện sự ổn định và hiệu quả trong cả hai mặt công việc. Đối với bounding box, mô hình này có khả năng dự đoán chính xác vị trí của các đối tượng, đặc biệt là trong trường hợp của lớp "Rice" với trung bình AP khoảng 0.91, chỉ ra rằng nó có khả năng xác định hạt gạo trên ảnh một cách đáng tin cậy. Tuy nhiên, trong việc phát hiện các khuyết điểm nhỏ hoặc tổn thương ("Inferior" và "Break"), AP giảm xuống khoảng 0.52 và 0.54, cho thấy mô hình gặp khó khăn hơn. Đối với segmentation, Detectron2 cũng cho thấy một hiệu suất đáng kể, với trung bình AP là khoảng 0.92 cho lớp "Rice" và khoảng 0.52 cho "Inferior" và "Break", cho thấy mô hình có khả năng phân đoạn ảnh một cách chính xác và đáng tin cậy.

Trong khi đó, YOLOv8 thể hiện sự ấn tượng đặc biệt trong việc dự đoán bounding box. Với trung bình AP khoảng 0.96 cho lớp "Rice" và khoảng 0.71 cho "Inferior", cùng với 0.83 cho "Break", mô hình này có khả năng xác định và dự đoán vị trí của các đối tượng một cách chính xác và đáng tin cậy. Đối với segmentation, YOLOv8 cũng cho thấy hiệu suất khá tốt, với trung bình AP là khoảng 0.88 cho lớp "Rice", 0.64 cho "Inferior", và 0.73 cho "Break". Mặc dù AP của nó thấp hơn so với bounding box, nhưng vẫn cho thấy mô hình có khả năng phân đoạn đối tượng một cách đáng kể.

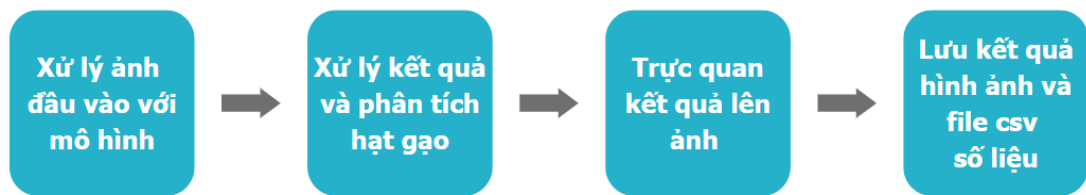
Trong khi hiệu quả trong việc dự đoán đối tượng trên ảnh, có thể gặp khó khăn khi phân biệt các loại hạt gạo với nhau ở mô hình Detectron2. Điều này có thể được thấy qua số liệu hạt, với sự đa dạng trong số lượng và đặc tính của từng loại hạt trong tập dữ liệu. Đối với các loại hạt như "Rice", mô hình thể hiện sự ổn định với mức độ dự đoán cao trên cả training và validation set. Tuy nhiên, đối với các loại hạt "Inferior" và "Break", mô hình có thể gặp khó khăn hơn trong việc phân loại chính xác, nhất là khi mức độ khuyết điểm trong các loại này thấp hơn. Trong khi đó, YOLOv8 thể hiện sự ấn tượng với khả năng dự đoán bounding box và phân loại đối tượng một cách chính xác, nhất là trong trường hợp của lớp "Rice". Mặc dù có một số sự biến động trong dữ liệu hạt, nhưng mô hình vẫn cho thấy sự ổn định và hiệu quả trong việc phân loại các loại hạt trên cả tập training và validation set. Điều này cho thấy YOLOv8 có khả năng tổng quát hóa tốt hơn trên các loại dữ liệu đa dạng và có thể phù hợp tốt hơn với các ứng dụng thực tế.

Sau khi đánh giá xong mô hình ta thực hiện một trong hai cách sau để tạo ra mô hình cuối cùng để sử dụng: Trong quá trình train các fold, lưu lại mô hình tốt nhất và sử dụng mô hình đó cho việc dự đoán. Cách này có ưu điểm là không cần train lại từ đầu, tuy nhiên, mô hình chỉ được huấn luyện trên một phần của dữ liệu và có thể không hoạt động tốt khi đối mặt với dữ liệu mới trong thực tế. Hoặc huấn luyện mô hình một lần nữa với toàn bộ dữ liệu (không chia thành tập

train và validation nữa), sau đó lưu lại mô hình này và sử dụng để dự đoán trên tập test để đánh giá kết quả. Phương pháp này đảm bảo rằng mô hình đã được huấn luyện trên toàn bộ dữ liệu và có thể làm việc hiệu quả với các dữ liệu mới trong thực tế.

### 3.3.3. Phân tích hạt gạo với Detectron2 và YOLOv8

Quy trình phân tích hạt gạo sẽ được thực hiện với mục đích không chỉ đếm chính xác số lượng hạt gạo trong mẫu, mà còn đo kích thước chiều dài và chiều rộng của từng hạt. Tiếp theo, kết quả sẽ được trực quan hóa bằng cách hiển thị tổng số hạt gạo trên hình ảnh, đồng thời đánh dấu số thứ tự lên mỗi hạt và vẽ những đường chỉ rõ chiều dài và chiều rộng của chúng.



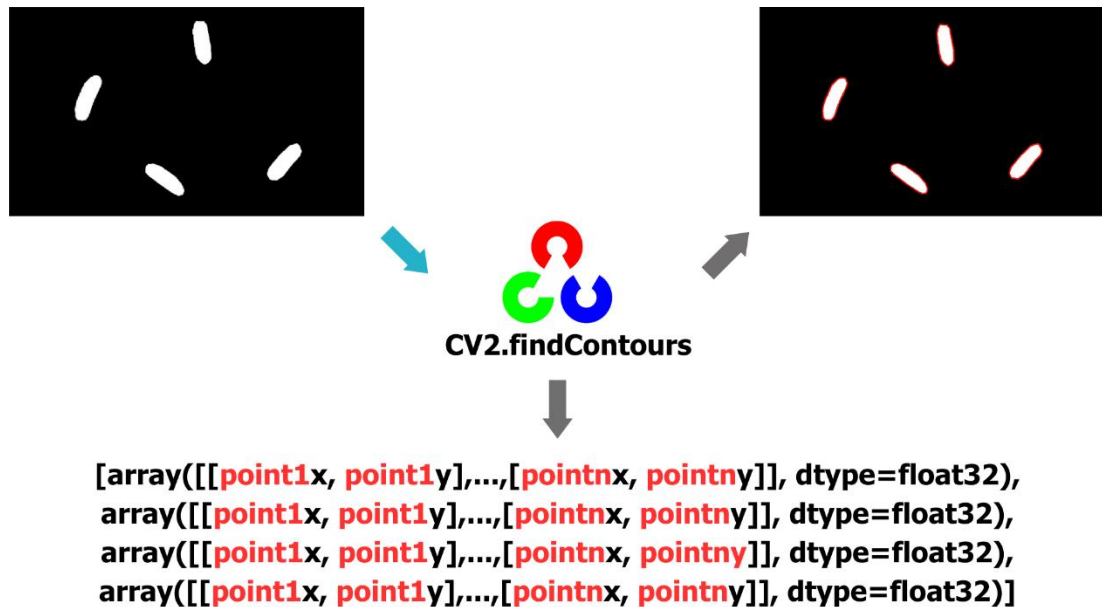
Hình 17. Quy trình phân tích hạt gạo

Ảnh sẽ được tải và đọc thông qua thư viện OpenCV một đường dẫn cụ thể đến tập ảnh. Tiếp theo, ảnh này sẽ được phân tích sử dụng hai mô hình trí tuệ nhân tạo YOLOv8 và Detectron2. Mỗi mô hình sẽ áp dụng các trọng số đã được huấn luyện trước đó trên bộ dữ liệu hạt gạo được chuẩn bị kỹ lưỡng, nhằm đảm bảo độ chính xác cao trong quá trình nhận diện và phân đoạn. Kết quả phân tích, được biểu diễn qua biến predict, sẽ bao gồm dữ liệu về hộp giới hạn và mặt nạ cho mỗi hạt gạo được phát hiện trong ảnh, cung cấp một cơ sở vững chắc cho các bước đánh giá và xử lý sau này.

Đối với các kết quả trả về của mô hình thì dữ liệu hộp giới hạn là các điểm tạo thành hộp giới hạn để thực hiện trực quan số thứ tự của hạt lên ảnh, quan trọng nhất là dữ liệu về mặt nạ của đối tượng trong ảnh. Các mặt nạ phải có dữ liệu là danh sách các điểm bao quanh của từng đối tượng trong tọa độ pixel, trong YOLOv8 đã có hỗ trợ về định dạng mặt nạ này, tuy nhiên trong Detectron2 dữ liệu mặt nạ thường được biểu diễn dưới dạng một mặt nạ nhị phân với các giá trị True và False. Vì vậy, để tiếp tục xử lý dữ liệu cần tinh chỉnh mặt nạ từ định dạng mặt nạ nhị phân sang danh sách các điểm tọa độ pixel có thể xử lý được. Điều này đảm bảo rằng dữ liệu mặt nạ có thể được sử dụng một cách hiệu quả trong các bước xử lý và phân tích tiếp theo.

Hàm `cv2.findContours()` là một trong những hàm quan trọng trong thư viện xử lý ảnh OpenCV (Open Source Computer Vision Library). Hàm được sử dụng để tìm các đường viền (contours) trong ảnh và được sử dụng rất phổ biến trong các

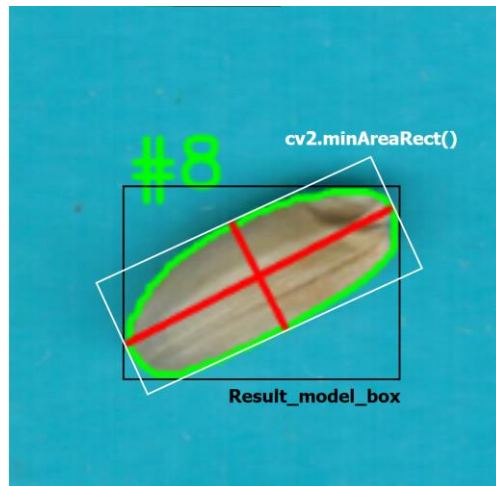
ứng dụng xử lý ảnh, như phân đoạn ảnh, nhận diện đối tượng và phát hiện biên cạnh. Thực hiện hàm với từng mặt nạ nhị phân của danh sách hạt gạo lưu các kết quả danh sách chứa các điểm tạo đường viền của từng hạt sẽ vào danh sách mặt nạ tổng và kết quả sẽ được sử dụng để tiếp tục phân tích.



Hình 18. Hàm findCountours chuyển đổi mặt nạ nhị phân

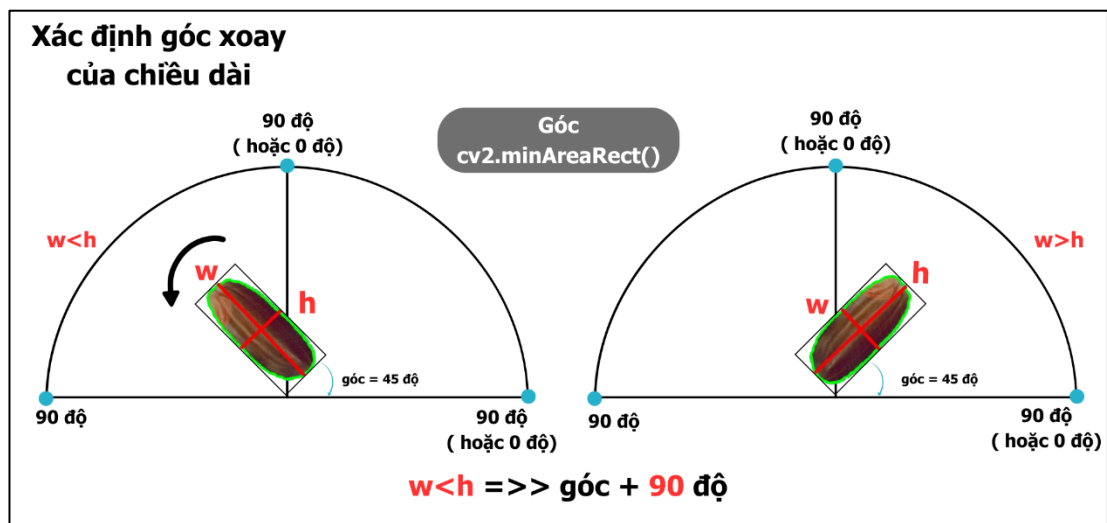
Quá trình đếm hạt gạo sẽ được thực hiện bằng cách đếm số lượng đối tượng trong kết quả trả về của mô hình khi xử lý với ảnh đầu vào và đánh số thứ tự lên từng hạt gạo khi lặp qua tất cả các kết quả trong danh sách kết quả trả về của mô hình.

Tiếp tục xử lý mặt nạ của mô hình, để tìm được chiều dài và chiều rộng ta cần tìm hộp giới hạn nhỏ nhất của mặt nạ. Hộp giới hạn của kết quả mô hình không đạt nhỏ nhất và không theo góc xoay của hạt vì vậy ta phải sử dụng hàm `cv2.minAreaRect()` để tìm tọa độ của trung tâm, kích thước chiều dài và chiều rộng của hộp, cùng với góc xoay của hộp.



Hình 19. Hàm minAreaRect tìm hộp giới hạn nhỏ nhất

Từ hộp giới hạn của mô hình và hộp giới hạn được tạo bởi hàm minAreaRect, chúng ta có thể thu được các thông tin quan trọng như tên lớp, chiều dài và chiều rộng. Đồng thời, áp đặt điều kiện chiều dài lớn hơn chiều rộng để đảm bảo tính chính xác của số liệu. Góc xoay của hộp giới hạn là một thông tin quan trọng từ kết quả của hàm minAreaRect để trực quan dữ liệu lên ảnh.



Hình 20. Góc xoay của hàm minAreaRect

Trong hàm minAreaRect, góc quay được hạn chế trong khoảng từ 0 đến 90 độ. Để xác định góc quay thực tế, ta dựa vào kích thước của hình chữ nhật nhỏ nhất. Khi chiều dài lớn hơn chiều rộng, điều này cho thấy hình chữ nhật đã nghiêng về phía góc lớn hơn 90 độ. Do đó, để đạt được góc quay thực tế, ta cần phải thêm 90 độ vào góc quay.

Như minh họa trên, ngay cả khi hạt gạo quay ngược lại, góc đo trả về của hàm vẫn là 45 độ. Để thu được góc thực tế là 135 độ, ta cần tăng thêm 90 độ vào góc đo ban đầu. Đối với xác định góc độ của chiều rộng, chỉ cần thay đổi điều kiện

kiểm tra để xác định xem chiều rộng có lớn hơn chiều dài hay không, và chỉ khi đó ta cộng thêm 90 độ vào góc quay để thu được góc thực tế.

Chiều dài và rộng của đối tượng được đo với đơn vị pixel vậy nên phải thực hiện chuyển đổi giá trị từ pixel sang đơn vị thực tế (mm). Quá trình chuyển đổi phụ thuộc vào mật độ điểm ảnh (đơn vị pixel trên mỗi inch hoặc DPI) của hình ảnh theo công thức sau:

$$size_{mm} = \frac{size_{pixel}}{DPI} \times 25.4$$

Với:

$size_{mm}$  : kích thước đơn vị mm

$size_{pixel}$  : kích thước đơn vị pixel

$DPI$  : Mật độ điểm ảnh của hình ảnh

25.4 : Số liệu chuyển đổi từ inch sang milimet

Thư viện OpenCV, giúp trực quan hóa các số liệu lên ảnh một cách dễ dàng, từ tổng số lượng hạt gạo cho đến việc vẽ các chiều dài và chiều rộng của chúng và đánh số thứ tự trên ảnh. Việc vẽ chiều dài và chiều rộng của ảnh phụ thuộc vào góc quay của ảnh để hiển thị chính xác kích thước của từng hạt trong ảnh.

Cuối cùng, các thông tin về mỗi hạt như số thứ tự, lớp, chiều dài và chiều rộng theo đơn vị milimet sẽ được ghi lại vào một danh sách chứa thông tin của các hạt. Sau đó, thông tin này sẽ được lưu trữ dưới dạng một bảng dữ liệu trong một file CSV, một định dạng tệp dữ liệu phổ biến được sử dụng để lưu trữ và truyền dữ liệu dưới dạng bảng. Đồng thời, hình ảnh với các số liệu trực quan sẽ được lưu vào thư mục kết quả.

### 3.4 KẾT QUẢ

Để đánh giá và so sánh hiệu quả của hai mô hình nhận dạng hình ảnh tiên tiến, Detectron2 và YOLOv8, một tập dữ liệu thử nghiệm bao gồm 30 tấm ảnh chụp hạt gạo đã được phân tích. Cả hai mô hình này, đã được huấn luyện trên tập dữ liệu hạt gạo, được sử dụng để đếm số lượng hạt trong mỗi bức ảnh và đo kích thước chiều dài và chiều rộng của từng hạt. Phân tích này nhằm cung cấp cái nhìn toàn diện về khả năng và độ chính xác của mỗi mô hình trong việc xử lý và phân tích dữ liệu hình ảnh chi tiết.

#### 3.4.1. Số lượng hạt gạo

Sau khi thực hiện phân tích trên 30 bức ảnh mẫu sử dụng hai mô hình nhận dạng hình ảnh, Detectron2 và YOLOv8 số lượng được tổng hợp trong hai bảng sau:

Bảng 9. Số lượng gạo đo được của mô hình YOLOv8

Số lượng hạt gạo	Số lượng tập Test			Mô hình YOLOv8		
Ảnh	Rice	Inferior	Break	Rice	Inferior	Break
G3 III-12	7	1	2	7	1	2
G3 III-15	4	0	0	4	0	0
G3 III-24	11	0	2	10	1	2
G7 III-6	5	0	0	5	0	0
G7 III-8	10	1	0	11	1	0
G7 III-27	7	1	0	7	1	0
G9 II-3	10	1	0	11	1	0
G9 II-4	7	0	1	8	0	0
G9 II-9	20	1	0	21	1	0
G9 II-10	23	1	4	24	1	4
G9 II-12	16	0	4	17	0	4
G9 II-14	21	1	2	23	0	2
G9 II-16	31	0	2	31	1	1
G9 II-17	30	1	0	31	0	0
G9 II-19	25	0	2	25	1	2
G9 II-20	17	0	1	17	0	1
G9 II-26	38	7	5	45	1	5
G13 II-2	32	0	0	32	0	0

G13 II-3	46	0	0	46	0	0
G13 II-4	29	0	0	29	0	0
G13 II-5	27	0	0	27	0	0
G13 II-6	39	0	0	39	0	0
G13 II-16	11	1	2	11	0	3
G13 II-20	33	1	0	34	0	0
G13 II-26	31	1	2	32	0	2
G17 III-8	25	5	0	25	5	0
G17 III-14	39	0	0	39	0	0
G18 I-4	31	1	3	31	5	1
G18 I-17	28	2	2	28	4	0
G18 I-19	47	2	0	49	1	0
Tổng	700	28	34	719	25	29

Bảng 10. Số lượng gạo đo được của mô hình Detectron2

Số lượng hạt gạo	Số lượng tập Test			Mô hình Detectron2		
	Rice	Inferior	Break	Rice	Inferior	Break
G3 III-12	7	1	2	8	1	2
G3 III-15	4	0	0	4	0	0
G3 III-24	11	0	2	11	0	2
G7 III-6	5	0	0	5	0	0
G7 III-8	10	1	0	10	1	0
G7 III-27	7	1	0	8	0	1
G9 II-3	10	1	0	10	1	0
G9 II-4	7	0	1	7	3	0
G9 II-9	20	1	0	20	1	0
G9 II-10	23	1	4	23	1	4
G9 II-12	16	0	4	18	3	1
G9 II-14	21	1	2	23	0	2
G9 II-16	31	0	2	30	2	1
G9 II-17	30	1	0	31	0	0
G9 II-19	25	0	2	26	0	2

G9 II-20	17	0	1	17	0	1
G9 II-26	38	7	5	45	0	5
G13 II-2	32	0	0	32	0	0
G13 II-3	46	0	0	46	0	0
G13 II-4	29	0	0	28	1	0
G13 II-5	27	0	0	27	0	0
G13 II-6	39	0	0	39	0	0
G13 II-16	11	1	2	11	2	2
G13 II-20	33	1	0	33	1	0
G13 II-26	31	1	2	33	0	1
G17 III-8	25	5	0	26	4	0
G17 III-14	39	0	0	39	0	0
G18 I-4	31	1	3	31	3	1
G18 I-17	28	2	2	28	4	0
G18 I-19	47	2	0	48	1	0
Tổng	700	28	34	717	29	25

Dựa trên kết quả từ các dữ liệu thực tế trên tập Test và hai mô hình, có thể thống kê được một bảng chi tiết, thể hiện phần trăm sai số của hai mô hình phát hiện, Detectron2 và YOLOv8. Bảng số liệu không chỉ là một cái nhìn sâu rộng vào hiệu suất của hai mô hình, mà còn là cơ sở để đánh giá sự đáng tin cậy và hiệu quả của chúng trong các tình huống thực tế.

Bảng 11. Bảng tỉ lệ sai số trong quá trình đếm hạt của hai mô hình

Số lượng hạt gạo	Mô hình YOLOv8			Mô hình Detectron2		
	Rice	Inferior	Break	Rice	Inferior	Break
G3 III-12	0	0	0	14.29	0	0
G3 III-15	0	0	0	0	0	0
G3 III-24	9.09	100	0	0	0	0
G7 III-6	0	0	0	0	0	0
G7 III-8	10	0	0	0	0	0
G7 III-27	0	0	0	14.29	100	100
G9 II-3	10	0	0	0	0	0
G9 II-4	14.29	0	100	0	100	100



G9 II-9	5	0	0	0	0	0
G9 II-10	4.35	0	0	0	0	0
G9 II-12	6.25	0	0	12.5	100	75
G9 II-14	9.52	100	0	9.52	100	0
G9 II-16	0	100	50	3.23	100	50
G9 II-17	3.33	100	0	3.33	100	0
G9 II-19	0	100	0	4	0	0
G9 II-20	0	0	0	0	0	0
G9 II-26	18.42	85.71	0	18.42	100	0
G13 II-2	0	0	0	0	0	0
G13 II-3	0	0	0	0	0	0
G13 II-4	0	0	0	3.45	100	0
G13 II-5	0	0	0	0	0	0
G13 II-6	0	0	0	0	0	0
G13 II-16	0	100	50	0	100	0
G13 II-20	3.03	100	0	0	0	0
G13 II-26	3.23	100	0	6.45	100	50
G17 III-8	0	0	0	4	20	0
G17 III-14	0	0	0	0	0	0
G18 I-4	0	400	66.67	0	200	66.67
G18 I-17	0	100	100	0	100	100
G18 I-19	4.26	50	0	2.13	50	0
Trung bình tỷ lệ sai số	3.36	47.86	12.22	3.19	45.67	18.06

Tỷ lệ sai số trong việc đếm các loại gạo là một chỉ số quan trọng để đánh giá hiệu suất của các mô hình nhận dạng đối tượng. Dựa vào bảng số liệu đã cung cấp, có thể thấy rằng cả hai mô hình YOLOv8 và Detectron2 đều có những điểm mạnh và yếu riêng trong việc đếm các loại gạo. Mô hình Detectron2 và YOLOv8 đều mang lại kết quả rất tốt khi sai số rất nhỏ chỉ bé hơn 4% cho thấy mô hình mang lại hiệu suất rất tốt khi huấn luyện với tập dữ liệu đã cung cấp ở thực tế.

Tuy nhiên, mô hình YOLOv8 cho thấy khi đến loại gạo "Inferior", mô hình này gặp khó khăn nhiều hơn với tỷ lệ sai số lên đến 47.86%. Điều này cho thấy rằng YOLOv8 có thể gặp khó khăn trong việc phân biệt và đếm đúng số lượng của

loại gạo này. Mặc dù tỷ lệ sai số cho loại gạo gãy (12.22%) cũng cao, nhưng vẫn thấp hơn so với gạo kém chất lượng.

Trong khi đó, mô hình Detectron2 lại cho thấy hiệu suất kém hơn đôi chút với loại gạo gãy (18.06%) so với YOLOv8, nhưng hiệu suất khi đo gạo kém chất lượng lại tốt hơn. Tuy vậy sai số của mô hình vẫn rất cao khi đếm loại gạo kém chất lượng (45.67% sai số trung bình).

Tổng quan, cả hai mô hình đều gặp khó khăn trong việc đếm và phân loại loại gạo kém chất lượng với tỷ lệ sai số cao do sự đa dạng về hình dạng, màu sắc và các chi tiết nấm mốc nhỏ trên hạt, cùng với hạn chế về số lượng dữ liệu thu thập được cho loại gạo này. Điều này có thể đề xuất rằng việc cải thiện khả năng nhận diện và phân biệt loại gạo này là một thách thức. Đồng thời, cần tiếp tục nghiên cứu và cải thiện mô hình để giảm thiểu tỷ lệ sai số và cải thiện hiệu suất đếm cho tất cả các loại gạo.

### 3.4.2 Kích thước hạt gạo

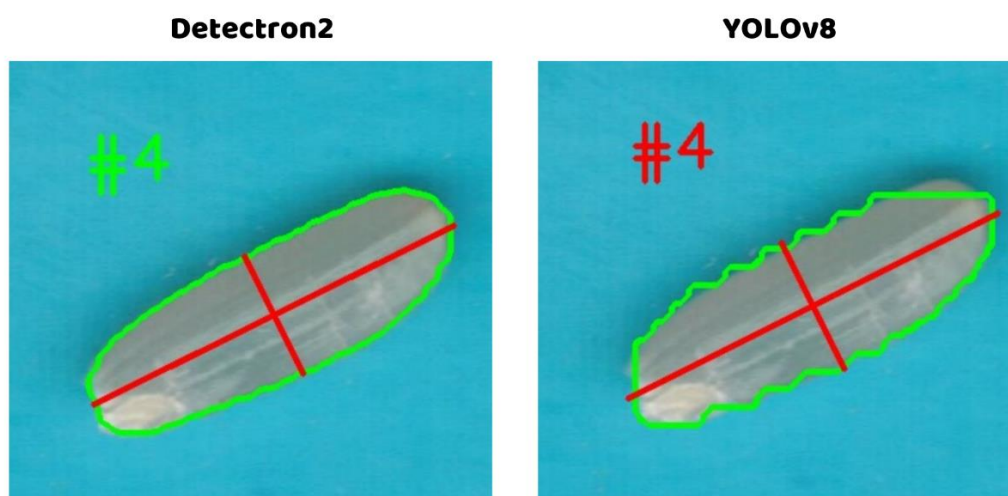
Dưới đây là bảng tổng hợp kết quả phân tích kích thước trung bình của hạt gạo các giống có trong tập dữ liệu Test với hai mô hình nhận dạng hình ảnh, Detectron2 và YOLOv8:

Bảng 12. Dữ liệu kích thước trung bình hạt gạo trong tập Test qua hai mô hình

Giống	Detectron2		YOLOv8		Sai số	
	Rộng (mm)	Dài (mm)	Rộng (mm)	Dài (mm)	Rộng (mm)	Dài (mm)
G3-III	7.22	2.16	7.2	2.32	0.02	0.16
G7-III	6.51	2.24	6.49	2.43	0.02	0.19
G9-II	5.83	2.02	5.66	2.19	0.17	0.17
G13-II	6.52	2.36	6.39	2.55	0.13	0.19
G17-III	6.1	2.69	5.9	2.9	0.2	0.21
G18-I	6.98	2.29	6.8	2.54	0.18	0.25
Sai số cao nhất					0.2	0.25
Sai số thấp nhất					0.02	0.16

Cả hai mô hình Detectron2 và YOLOv8 đều cho thấy kết quả khá chính xác trong việc đo kích thước của hạt gạo. Kích thước của hạt gạo được dự đoán bởi hai mô hình có độ tương đồng cao. Sai số giữa các giá trị tương đối nhỏ, thường dao động trong khoảng từ 0.02mm đến 0.25mm.

Tuy nhiên, mô hình YOLOv8 có xu hướng cho kết quả có chiều rộng lớn hơn so với Detectron2 cho hầu hết các giống lúa. Trong khi đó, về chiều dài, hai mô hình cho kết quả khá tương đồng,



Hình 21. Sự khác biệt mặt nạ giữa hai mô hình

Nguyên nhân dẫn đến sự khác biệt này do YOLOv8 sử dụng một kiến trúc mạng dựa trên CNN được tối ưu hóa cho tốc độ xử lý. Điều này dẫn đến việc đơn giản hóa cấu trúc mạng, giảm số lượng lớp và độ phức tạp của các phép toán. Kết quả là độ phân giải của điểm ảnh trong mặt nạ thường thấp hơn.

Trong khi đó, Detectron2 sử dụng một mô hình dựa trên FPN (Feature Pyramid Network) và Mask R-CNN. Điều này cho phép mô hình trích xuất và kết hợp các đặc trưng ở nhiều độ phân giải khác nhau. Nhờ vậy, Detectron2 có khả năng tạo ra các mặt nạ có độ phân giải cao và chi tiết hơn, giúp trong việc nhận diện các đối tượng nhỏ và phức tạp hơn. Mục tiêu chính nhằm tối ưu hóa cho tốc độ xử lý và hiệu suất thời gian thực. Do đó, độ chính xác của mặt nạ có thể bị ảnh hưởng để đảm bảo tốc độ khung hình cao.

Dựa trên dữ liệu kích thước của hạt gạo được đo bằng Detectron2 và YOLOv8, có thể thấy rằng cả hai mô hình đều có khả năng đo lường kích thước của hạt gạo một cách đáng tin cậy. Tuy nhiên, việc chọn mô hình phù hợp cần phải cân nhắc mục tiêu cụ thể của phân tích.

Nếu tốc độ phân tích là ưu tiên hàng đầu, YOLO là lựa chọn tốt nhất. Mô hình này được tối ưu hóa để tăng tốc độ xử lý, đồng thời vẫn đảm bảo độ chính xác trong việc đo kích thước của các đối tượng. Điều này làm cho YOLO phù hợp cho các ứng dụng đòi hỏi phản ứng nhanh và xử lý ảnh liên tục, như trong các hệ thống giám sát video.

Tuy nhiên, nếu độ chính xác cao là ưu tiên hàng đầu, Detectron2 là lựa chọn phù hợp nhất. Mặc dù có thể tốn nhiều thời gian hơn để xử lý, nhưng mô hình này sử dụng các kiến trúc phức tạp như FPN và Mask R-CNN để tạo ra các mặt nạ có độ phân giải cao và chi tiết. Điều này giúp Detectron2 phát hiện và đo kích thước của các đối tượng một cách chính xác và chi tiết, làm cho nó phù hợp cho các ứng dụng yêu cầu độ chính xác cao, như trong nghiên cứu khoa học và y học.

Trong tổng thể, việc so sánh và đánh giá kết quả đo lường giữa Detectron2 và YOLOv8 có thể cung cấp thông tin quan trọng để hiểu rõ hơn về hiệu suất của từng mô hình trong ứng dụng thực tế.

### 3.4.3 Phân tích sự khác biệt kích thước giữa các vị trí bông của hạt gạo

Qua quá trình so sánh giữa hai mô hình Detectron2 và YOLOv8, nhận thấy rằng Detectron2 thể hiện sự chính xác đáng kinh ngạc trong việc nhận diện mặt nạ của hạt gạo. Điều này chỉ ra rằng Detectron2 có khả năng vượt trội khi xử lý các tác vụ liên quan đến phát hiện và phân tích về đối tượng nhỏ, phức tạp như hạt gạo. Do đó, quyết định chọn mô hình Detectron2 để tiếp tục nghiên cứu phân tích sự khác biệt đầu, giữa và cuối bông tiếp theo là hoàn toàn hợp lý.



Hình 22. Minh họa ba vị trí bông

Nghiên cứu này nhằm mục đích tìm hiểu sâu hơn về quá trình phân phối và hấp thụ dinh dưỡng của cây lúa, thông qua việc khảo sát kích thước của hạt gạo tại các vị trí khác nhau trên bông. Sử dụng công nghệ Detectron2 trong quá trình phân tích không chỉ giúp tiết kiệm thời gian và công sức mà còn đảm bảo độ chính xác và hiệu suất cao nhất cho quá trình nghiên cứu và ứng dụng thực tiễn. Phương pháp này mở ra cơ hội mới để hiểu rõ hơn về cơ chế phát triển của

cây lúa và tối ưu hóa quản lý dinh dưỡng trong nông nghiệp, đồng thời thúc đẩy sự tiến bộ trong nghiên cứu và ứng dụng công nghệ trong lĩnh vực nông nghiệp.

Bảng 13. Số liệu ảnh và số hạt gạo đạt ở ba vị trí đầu, giữa và cuối bông

Giống	Số ảnh			Số hạt		
	Đầu bông	Giữa bông	Cuối bông	Đầu bông	Giữa bông	Cuối bông
ANP 3.2	10	10	10	195	367	345
F5.23	10	10	10	127	336	368
F5.54	10	10	10	71	209	230
F5.62	10	10	10	119	439	449
G3-III	10	10	10	116	391	302
G7-III	10	10	10	102	382	381
G9-II	11	10	9	147	360	265
G13-II	10	10	10	119	337	283
G17-III	10	10	10	181	335	155
G18-I	10	10	10	116	289	286
Tổng	101	100	99	1293	3445	3064

Trong quá trình thu thập dữ liệu, với mỗi giống gạo, thu thập ảnh ở ba vị trí bông khác nhau là đầu bông, giữa bông và cuối bông, tổng cộng 101 ảnh đầu bông, 100 ảnh giữa bông và 99 ảnh cuối bông. Sau khi thực hiện xử lý tập dữ liệu ảnh ở ba vị trí bông với Detectron2, thu được 1293 hạt đầu bông, 3445 hạt giữa bông và 3064 hạt cuối bông.

Bảng 14. Số liệu kích thước của đầu, giữa và cuối bông của mỗi giống

Giống	Chiều rộng (mm)			Chiều dài (mm)		
	Đầu bông	Giữa bông	Cuối bông	Đầu bông	Giữa bông	Cuối bông
ANP 3.2	2.49	2.53	2.56	5.73	5.73	5.81
F5.23	2.4	2.45	2.47	6	6.05	6.07
F5.54	2.16	2.24	2.27	7.24	7.4	7.43
F5.62	2.28	2.32	2.34	6.99	7	6.99
G3-III	2.09	2.14	2.18	7.1	7.23	7.3
G7-III	2.22	2.35	2.38	6.56	6.69	6.73
G9-II	2.01	2.02	2.06	5.87	5.89	5.94
G13-II	2.27	2.33	2.39	6.34	6.44	6.53
G17-III	2.61	2.67	2.74	6.13	6.14	6.38
G18-I	2.24	2.26	2.29	6.93	6.95	7.02

Bảng 15. Sai số kích thước của các vị trí bông

Giống	Chiều rộng (mm)			Chiều dài (mm)		
	Đầu Giữa	Giữa Cuối	Cuối Đầu	Đầu Giữa	Giữa Cuối	Cuối Đầu
ANP 3.2	0.04	0.03	0.07	0	0.08	0.08
F5.23	0.05	0.02	0.07	0.05	0.02	0.07
F5.54	0.08	0.03	0.11	0.16	0.03	0.19
F5.62	0.04	0.02	0.06	0.01	0.01	0
G3-III	0.05	0.04	0.09	0.13	0.07	0.2
G7-III	0.13	0.03	0.16	0.13	0.04	0.17
G9-II	0.01	0.04	0.05	0.02	0.05	0.07
G13-II	0.06	0.06	0.12	0.1	0.09	0.19
G17-III	0.06	0.07	0.13	0.01	0.24	0.25
G18-I	0.02	0.03	0.05	0.02	0.07	0.09
Max	0.13	0.07	0.16	0.16	0.24	0.25
Min	0.01	0.02	0.05	0	0.01	0

Kết quả nghiên cứu cho thấy kích thước hạt gạo có sự khác biệt nhất định tùy theo vị trí trên bông. Về chiều dài, hạt gạo ở vị trí cuối bông thường nhỏ hơn so với hai vị trí còn lại, với chênh lệch sai số cao nhất đạt 0.25 mm. Chiều dài hạt gạo ở vị trí đầu và giữa bông tương đối đồng đều, với mức chênh lệch sai số cao nhất chỉ đạt 0.16 mm.

Tuy nhiên, xét về chiều rộng, sự khác biệt chỉ thể hiện rõ rệt ở hạt gạo đầu bông. Kích thước chiều rộng của hạt gạo ở vị trí này nhỏ hơn so với hai vị trí còn lại, với mức biến động từ 0.01 đến 0.16 mm. Chiều rộng hạt gạo ở vị trí giữa và cuối bông tương đối tiệm cận, chỉ chênh lệch từ 0.02 đến 0.07 mm.

Dù sự khác biệt về kích thước giữa ba vị trí có thể không đáng kể, nhưng từ những kết quả thu được, có thể suy luận rằng vị trí cuối bông, nằm gần với gốc của cây lúa nhất, có ảnh hưởng đáng kể đến quá trình hấp thụ dinh dưỡng của hạt gạo. Hạt gạo ở vị trí cuối bông có thể hấp thụ lượng dưỡng chất nhiều hơn từ đất so với hạt ở đầu và giữa bông đem lại kích thước lớn nhất cả về chiều dài và chiều rộng của hạt so với hai vị trí còn lại, điều này có thể lý giải thông qua sự gần gũi của vị trí bông với rễ của cây và môi trường dinh dưỡng, tạo điều kiện thuận lợi cho sự hấp thụ dinh dưỡng hiệu quả hơn so với đầu và giữa bông.

Qua quá trình phân tích trên có thể thấy sự khác nhau của việc hấp thụ dinh dưỡng của cây lúa, thông qua việc khảo sát kích thước của hạt gạo ở các vị trí khác nhau trên bông. Điều này đóng vai trò quan trọng về việc hiểu rõ cơ chế cung cấp dưỡng chất cho cây lúa và tối ưu hóa quản lý dinh dưỡng trong nông nghiệp của nước ta.

### 3.4.4 Một số giao diện Web được thực hiện với Flask

YOLO&DETEC

FILE FOLDER

DEMO NHẬN DIỆN HẠT GẠO VỚI MÔ HÌNH  
DETECTRON2 VÀ YOLOV8

Chọn file ảnh để nhận diện...

Các định dạng được hỗ trợ: PNG, JPG và JPEG.

Chọn mô hình

YOLOv8

Thông tin

Trang web được tạo bởi: Huỳnh Phúc Duy

Facebook

Google

Liên hệ

hpduy\_21th@student.agu.edu.vn  
0378108260  
Trường đại học An Giang

Copyright © Your Website 2023

Hình 23. Giao diện xử lý File

YOLO&DETEC

FILE FOLDER

DEMO NHẬN DIỆN HẠT GẠO VỚI MÔ HÌNH  
DETECTRON2 VÀ YOLOV8

Nhập đường dẫn thư mục chứa ảnh...

YOLOv8

THỰC HIỆN

Thông tin

Trang web được tạo bởi: Huỳnh Phúc Duy

Facebook

Google

Liên hệ

hpduy\_21th@student.agu.edu.vn  
0378108260  
Trường đại học An Giang

Copyright © Your Website 2023

Hình 24. Giao diện xử lý thư mục





### 3.5 HƯỚNG PHÁT TRIỂN

Phân tích hạt gạo thông qua hai mô hình trí tuệ nhân tạo tiên tiến, Detectron2 và YOLOv8, đã mang lại những kết quả đáng kể, đặc biệt trong các tác vụ đếm số lượng, đo kích thước của các hạt và phân loại chất lượng gạo. Điều này cho thấy tiềm năng lớn của công nghệ nhận diện hình ảnh trong việc tự động hóa các quy trình kiểm định chất lượng nông sản. Tuy nhiên, mặc dù đã đạt được những thành tựu trên, cả hai mô hình vẫn còn một số hạn chế quan trọng cần được khắc phục.

Cụ thể, khả năng của mô hình trong việc xác định và phân loại các hạt gạo không đạt tiêu chuẩn, hạt gãy, hạt nứt chưa đạt được kết quả cao. Nguyên nhân chính cũng vì sự hạn chế về tập dữ liệu huấn luyện mô hình và sự đa dạng của các loại gạo này. Điều này không chỉ làm ảnh hưởng đến độ chính xác của việc đếm số lượng hạt gạo mà còn gây khó khăn trong việc đánh giá chất lượng tổng thể của lô gạo.

Để nâng cao hiệu quả của hệ thống phân tích và đạt được mục tiêu của việc đảm bảo chất lượng gạo tối ưu, hướng phát triển tiếp theo của nghiên cứu sẽ tập trung vào việc cải tiến và tinh chỉnh tập dữ liệu và thuật toán. Mục tiêu là để chúng có thể nhận diện rõ ràng các loại gạo dựa trên các đặc điểm cụ thể như hình dạng, kích thước và mức độ hư hại của hạt. Bao gồm việc cải thiện khả năng phân biệt gạo gãy, gạo kém chất lượng. Cùng với đó là việc mở rộng khả năng nhận diện hạt gạo bạc bụng (ảnh hưởng đến chất lượng xay chà và thị hiếu của người tiêu dùng), từ đó giúp các nhà sản xuất và các cơ quan kiểm định có được cái nhìn toàn diện hơn về mẻ gạo được sản xuất và kiểm soát chất lượng một cách chính xác hơn.

Thông qua việc áp dụng công nghệ AI hiện đại và tiếp tục nghiên cứu, chúng ta có thể mong đợi một hệ thống phân tích gạo hoàn thiện hơn, góp phần đảm bảo an toàn và chất lượng thực phẩm cho người tiêu dùng, cũng như nâng cao hiệu quả sản xuất và giảm thiểu lãng phí trong ngành công nghiệp gạo.

## TÀI LIỆU THAM KHẢO

- [1] “Gạo Việt Nam trước cơ hội làm chủ thị trường.” Accessed: Mar. 26, 2024. [Online]. Available: <https://nhandan.vn/gao-viet-nam-truoc-co-hoi-lam-chu-thi-truong-post789611.html>
- [2] “Thực trạng sản xuất và xuất khẩu gạo Việt Nam – Các giải pháp phát triển bền vững trong những năm tới - Trang 1 - Tạp chí Kinh tế & Phát triển.” Accessed: Jan. 24, 2024. [Online]. Available: <https://ktpt.neu.edu.vn/tap-chi/so-276/muc-luc-107/thuc-trang-san-xuat-va-xuat-khau-gao-viet-nam-%E2%80%93-cac-giai-phap-phat-trien-ben-vung-trong-nhung-nam-toi.379268.aspx>
- [3] “Object detection: Speed and Accuracy (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3).” Accessed: Jan. 24, 2024. [Online]. Available: <https://viblo.asia/p/object-detection-speed-and-accuracy-faster-r-cnn-r-fcn-ssd-fpn-retinanet-and-yolov3-ByEZkJRWKQ0>
- [4] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.
- [5] P. Đ. Khánh, “Khoa học dữ liệu,” Khanh’s blog. Accessed: Apr. 05, 2024. [Online]. Available: <https://phamdinhkhanh.github.io>
- [6] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO.” Jan. 2023. Accessed: Apr. 05, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [7] “Hands-On Computer Vision with Detectron2.” Accessed: Apr. 06, 2024. [Online]. Available: <https://subscription.packtpub.com/book/data/9781800561625/2/ch02lvl1sec05/an-introduction-to-detectron2-and-its-architecture>
- [8] M. Butt, N. Glas, J. Monsuur, R. Stoop, and A. D. Keijzer, “Using YOLOv8 and Detectron2 Models to Detect Bullet Holes and Calculate Scores from Shooting Cards.” Oct. 11, 2023. doi: 10.20944/preprints202310.0657.v1.
- [9] “Thư viện số Đại Học Công Nghệ Giao Thông Vận Tải: Python Cơ Bản (Tái Bản 2023).” Accessed: Mar. 28, 2024. [Online]. Available: <http://thuvien.utt.edu.vn:8080/jspui/handle/123456789/1613>
- [10] K. Pulli, A. Baksheev, K. Korniyakov, and V. Eruhimov, “Realtime Computer Vision with OpenCV: Mobile computer-vision technology will soon become as ubiquitous as touch interfaces,” *Queue*, vol. 10, no. 4, pp. 40–56, Apr. 2012, doi: 10.1145/2181796.2206309.
- [11] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, “A brief introduction to OpenCV,” in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725–1730.

- [12] T. Carneiro, R. V. Medeiros Da Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, “Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications,” *IEEE Access*, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.
- [13] Q. Lin, G. Ye, J. Wang, and H. Liu, “RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots,” in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, and G. Neumann, Eds., in *Proceedings of Machine Learning Research*, vol. 164. PMLR, Nov. 2022, pp. 1789–1794. [Online]. Available: <https://proceedings.mlr.press/v164/lin22c.html>
- [14] “Cross Validation in Machine Learning - GeeksforGeeks.” Accessed: May 06, 2024. [Online]. Available: <https://www.geeksforgeeks.org/cross-validation-machine-learning/>
- [15] “Flask By Example - Gareth Dwyer - Google Sách.” Accessed: May 11, 2024. [Online]. Available: [https://books.google.com.vn/books?hl=vi&lr=&id=M7PjCwAAQBAJ&oi=fnd&pg=PP1&dq=what+is+flask&ots=pVcBd-8kef&sig=xVlRgF23XKuGUKCc3Ep5ljKu\\_D8&redir\\_esc=y#v=onepage&q=what%20is%20flask&f=false](https://books.google.com.vn/books?hl=vi&lr=&id=M7PjCwAAQBAJ&oi=fnd&pg=PP1&dq=what+is+flask&ots=pVcBd-8kef&sig=xVlRgF23XKuGUKCc3Ep5ljKu_D8&redir_esc=y#v=onepage&q=what%20is%20flask&f=false)