TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG MÔ HÌNH HỌC SÂU TRONG PHÂN LOẠI HẠT LÚA GIỐNG

NGUYỄN TRỌNG KHANG

AN GIANG, 5/2025

TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN

ỨNG DỤNG MÔ HÌNH HỌC SÂU TRONG PHÂN LOẠI HẠT LÚA GIỐNG

NGUYỄN TRỌNG KHANG

DTH215955

GIẢNG VIÊN HƯỚNG DẪN: TS. NGUYỄN VĂN HÒA

AN GIANG, 5/2025

Khóa luận "Ứng dụng mô hình học sâu trong phân loại hạt lúa giống" do sinh viên Nguyễn Trọng Khang thực hiện dưới sự chỉ dẫn của Ts.Nguyễn Văn Hòa đã báo cáo kết quả nghiên cứu và được Hội đồng Khoa học và Đào tạo thông qua ngày.....

Phản biện 1	**Phản biện 2**
(Ký và ghi rõ chức danh, họ tên)	*(Ký và ghi rõ chức danh, họ tên)*

Giảng viên hướng dẫn

(Ký và ghi rõ chức danh, họ tên)

LỜI CẢM ƠN

Sau bốn năm học tập và rèn luyện tại Trường Đại học An Giang, em xin bày tỏ lòng biết ơn sâu sắc đến các quý thầy giáo, cô giáo, quý lãnh đạo nhà trường, các anh chị cán bộ nhân viên và các bạn sinh viên đã luôn quan tâm, giúp đỡ tôi trong suốt thời gian qua.

Đặc biệt, tôi xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Văn Hòa, người thầy đã trực tiếp hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài khóa luận tốt nghiệp.

Em cũng xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn yêu thương, động viên, ủng hộ em trong suốt quá trình học tập và thực hiện khóa luận. Nhờ sự quan tâm, giúp đỡ của mọi người, em đã có thêm động lực để vượt qua khó khăn, hoàn thành tốt khóa luận tốt nghiệp của mình.

Tuy nhiên, do năng lực còn hạn chế, nên chuyên đề nghiên cứu khoa học không tránh khỏi những thiếu sót. Kính mong nhận được sự thông cảm, đóng góp ý kiến của các quý thầy cô để bài nghiên cứu khóa luận được hoàn thiện hơn.

Xin chân thành cảm ơn!

An Giang, ngày 14 tháng 5 năm 2025

Sinh viên thực hiện

Nguyễn Trọng Khang

TÓM TẮT

Đề tài "Ứng dụng mô hình học sâu trong phân loại hạt lúa giống" nhằm mục đích phát triển các phương pháp tiên tiến, hiện đại nhằm cải thiện khả năng phân tích các đặc điểm nhận dạng và phân loại của hạt lúa giống.

Đề tài tập trung xây dựng phương pháp tiếp cận và lựa chọn kỹ thuật phù hợp để phát triển mô hình phân loại hạt lúa giống. Nghiên cứu đề xuất hai hướng triển khai: mô hình tích hợp một giai đoạn và mô hình hai giai đoạn riêng biệt. Trong quá trình thực hiện, đề tài đã lựa chọn và so sánh các kiến trúc mạng nơ-ron sâu tiêu biểu thuộc các hướng mở rộng (model scaling) khác nhau, bao gồm: ResNet50 (mở rộng theo chiều sâu), InceptionV3 (mở rộng theo chiều rộng), EfficientNet (mở rộng đồng thời theo chiều sâu, chiều rộng và độ phân giải), và Vision Transformer (dựa trên kiến trúc Transformer). Ngoài ra, các mô hình nổi bật trong bài toán phát hiện đối tượng như YOLO và Faster R-CNN cũng được xem xét và đánh giá.

Từ kết quả nghiên cứu, đề tài đã tiến hành so sánh hiệu quả giữa các phương pháp xây dựng mô hình. Kết quả đánh giá cho thấy mô hình tích hợp một giai đoạn đạt hiệu suất vượt trội với độ chính xác lên đến 98% và thời gian suy luận chỉ 17.6ms. Điều này khẳng định tầm quan trọng của việc lựa chọn kiến trúc mô hình phù hợp với bài toán cụ thể nhằm tối ưu hóa cả độ chính xác và hiệu năng xử lý.

MỤC LỤC

DANH SÁCH HÌNH ẢNH

DANH SÁCH BIỂU ĐỒ

DANH SÁCH BẢNG

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1 ĐẶT VẤN ĐỀ

1.1.1 Tính cấp thiết của đề tài

Trong bối cảnh nông nghiệp hiện đại, việc nâng cao chất lượng và đa dạng hóa giống lúa đang trở thành yếu tố then chốt, không chỉ nhằm đảm bảo an ninh lương thực mà còn góp phần thúc đẩy sự phát triển bền vững của ngành. Chính vì thế, nhu cầu phân loại hạt giống lúa một cách chính xác và hiệu quả ngày càng trở nên cấp thiết. Tuy nhiên, phương pháp phân loại hiện nay chủ yếu vẫn dựa trên cách làm thủ công, vừa tốn nhiều thời gian và công sức, vừa phụ thuộc lớn vào trình độ chuyên môn của người thực hiện.

Trước xu thế cách mạng công nghiệp 4.0, việc ứng dụng các tiến bộ khoa học kỹ thuật, đặc biệt là trí tuệ nhân tạo, vào lĩnh vực nông nghiệp là xu hướng tất yếu, không chỉ ở Việt Nam mà còn tại nhiều quốc gia nông nghiệp phát triển như Thái Lan, Trung Quốc. Trong đó, công nghệ học sâu nổi bật với khả năng trích xuất đặc trưng hình ảnh mạnh mẽ, cho phép tự động hóa quá trình nhận dạng và phân loại hạt giống với độ chính xác cao.

Xuất phát từ nhu cầu thực tiễn và tiềm năng ứng dụng mạnh mẽ của công nghệ, đề tài "Ứng dụng mô hình học sâu trong phân loại hạt lúa giống" hướng đến việc nâng cao hiệu quả và chất lượng trong sản xuất nông nghiệp. Việc áp dụng mô hình học sâu không chỉ hỗ trợ tự động hóa quy trình phân loại hạt giống mà còn có thể mở rộng ứng dụng sang nhiều khâu khác trong chuỗi giá trị nông nghiệp, qua đó giúp tiết kiệm thời gian, tối ưu hóa nguồn lực và giảm sự phụ thuộc vào lao động thủ công. Bên cạnh đó, đề tài còn mở ra một hướng tiếp cận mới trong xây dựng hệ thống nông nghiệp thông minh, góp phần thúc đẩy nền nông nghiệp hiện đại, bền vững và hiệu quả hơn.

1.1.2 Phạm vi

Nghiên cứu này tập trung vào việc ứng dụng các mô hình học sâu, để xây dựng phương pháp phân loại hạt lúa giống thông qua hình ảnh. Hai hướng tiếp cận chính được triển khai bao gồm: mô hình phân loại một giai đoạn và mô hình hai giai đoạn. Trên cơ sở đó, nghiên cứu tiến hành đánh giá và so sánh hiệu suất giữa các phương pháp, nhằm xác định hướng tiếp cận tối ưu cho bài toán phân loại hạt giống trong thực tiễn.

1.1.3 Mục tiêu nghiên cứu

Xây dựng các phương pháp tiếp cận bài toán và các mô hình phân loại hạt lúa giống dựa trên hình ảnh.

Kiểm thử và đánh giá hiệu quả của các phương pháp, mô hình thông qua phân tích kết quả thực nghiệm nhằm xác định hướng tiếp cận tối ưu.

Thông qua đề tài, tôi kỳ vọng sẽ góp phần thúc đẩy việc ứng dụng các tiến bộ của khoa học và công nghệ, đặc biệt là trí tuệ nhân tạo và thị giác máy tính, vào lĩnh vực sản xuất nông nghiệp không chỉ tại tỉnh An Giang mà còn trên toàn vùng Đồng bằng sông Cửu Long, từ đó góp phần nâng cao chất lượng, hiệu quả sản xuất và hướng đến phát triển nông nghiệp hiện đại, bền vững.

1.1.4 Phương pháp nghiên cứu

Tìm hiểu và nghiên cứu cơ sở lý thuyết liên quan đến học sâu, thị giác máy tính, cùng với các tài liệu, bài báo khoa học về bài toán phân loại nói chung và bài toán phân loại hạt giống lúa nói riêng.

Thu thập và xây dựng dữ liệu thông qua các hình ảnh được chụp bằng scan.

Nghiên cứu về thuật toán được các mô hình học sâu được sử dụng. Cách áp dụng Transfer Learning để huấn luyện, xây dựng và tích hợp mô hình theo từng phương pháp.

Kiểm thử, đánh giá và cải tiến mô hình để mang lại độ ổn định và sự chính xác cao nhất có thể trong quá trình xây dựng và sử dụng.

CHƯƠNG 2.
TỔNG QUAN ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT

2.1 KHÁI NIỆM VỀ LÚA

2.1.1 Khái niệm về lúa

Lúa (*Oryza sativa*) đóng vai trò thiết yếu trong đời sống con người, đặc biệt là ở Việt Nam, một quốc gia có nền nông nghiệp lúa nước lâu đời. Lúa-gạo không chỉ là lương thực chính mà còn là nguồn thu nhập quan trọng cho người nông dân. Ngày nay, nhiều giống lúa được lai tạo ra với chất lượng đa dạng và năng suất để đáp ứng nhu cầu ngày càng cao của con người đặc biệt là vấn đề về an ninh lương thực.

Hình 1: Cây lúa

2.1.2 Phân loại hạt lúa giống truyền thống

Phân loại hạt lúa giống có thể được thực hiện thông qua nhiều phương pháp và tùy thuộc vào mục tiêu cụ thể của việc phân loại, bao gồm phân loại theo chất lượng hạt giống, năng suất, đặc tính sinh học, giá trị và các yếu tố khác. Dưới đây là một số tiêu chí để phân loại hạt lúa giống phổ biến [1].

- Phân loại hình thái học: Phân loại theo kích thước, hình dạng, màu sắc, khối lượng và cấu trúc bên ngoài của hạt giống.
- Tiêu chí về độ thuần giống: Đánh giá mức độ đồng nhất về mặt di truyền của hạt giống, với yêu cầu tỷ lệ hạt khác giống không vượt quá quy định (thường dưới 0.1-0.5% với hạt giống nguyên chủng). Hạt giống thuần có đặc điểm hình thái đồng nhất về màu sắc, kích thước và hình dạng.
- Tỷ lệ nảy mầm: Phân loại dựa trên khả năng nảy mầm, thường yêu cầu trên 85% đối với hạt giống cấp 1, trên 80% với hạt giống thương phẩm. Bao gồm cả đánh giá về tốc độ nảy mầm và đồng đều trong nảy mầm.

- Độ ẩm: Phân loại theo hàm lượng nước trong hạt, thường yêu cầu 12-13% để đảm bảo khả năng bảo quản dài hạn. Độ ẩm cao hơn 14% tăng nguy cơ nấm mốc và giảm sức sống của hạt.
- Đặc tính di truyền và năng suất: Phân loại dựa trên đặc tính di truyền như thời gian sinh trưởng (ngắn, trung bình, dài ngày), tiềm năng năng suất, khả năng kháng sâu bệnh, và các đặc tính chất lượng gạo sau thu hoạch.
- Tuổi của hạt giống: Phân loại theo thời gian thu hoạch và bảo quản. Hạt giống mới thu hoạch thường có sức sống cao hơn, nhưng một số giống lúa cần qua thời gian ngủ nghỉ (2-3 tháng) để đạt tỷ lệ nảy mầm tối ưu.

2.2 TRÍ TUỆ NHÂN TẠO

2.2.1 Học sâu

Học sâu (Deep Learning) là một nhánh của trí tuệ nhân tạo (AI) và học máy (Machine Learning), trong đó các mô hình mạng nơ-ron nhân tạo nhiều lớp được sử dụng để học từ dữ liệu, đưa ra dự đoán và xử lý dữ liệu theo cách hoạt động của bộ não con người. Học sâu đã có lịch sử phát triển lâu dài bắt đầu từ sự ra đời của Perceptron vào năm 1958, trải qua nhiều thăng trầm rồi bùng nổ mạnh mẽ trong các thập kỉ gần đây nhờ sự phát triển của phần cứng đồng thời cùng các mô hình mạng nơ-ron nhiều lớp.

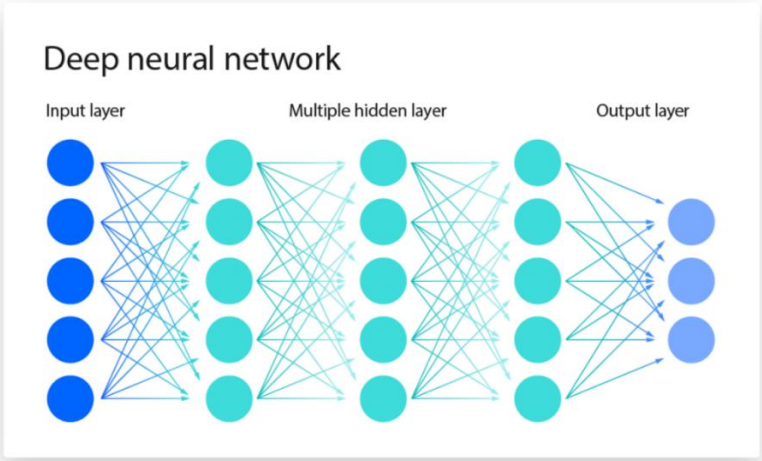
Mô hình học sâu ứng dụng trong nhiều lĩnh vực, từ xử lý ngôn ngữ tự nhiên, nhận dạng hình ảnh và giọng nói, đến chẩn đoán y tế và dự đoán tài chính, chứng minh sự linh hoạt và tiềm năng to lớn của chúng trong việc giải quyết các vấn đề phức tạp.

2.2.1.1 Cấu trúc của một mạng nơ-ron bao gồm:

Tầng đầu vào (Input Layer): Nhận dữ liệu đầu vào và truyền vào mạng, số lượng nơ-ron của tầng này phù thuộc vào số đặc trưng của dữ liệu.

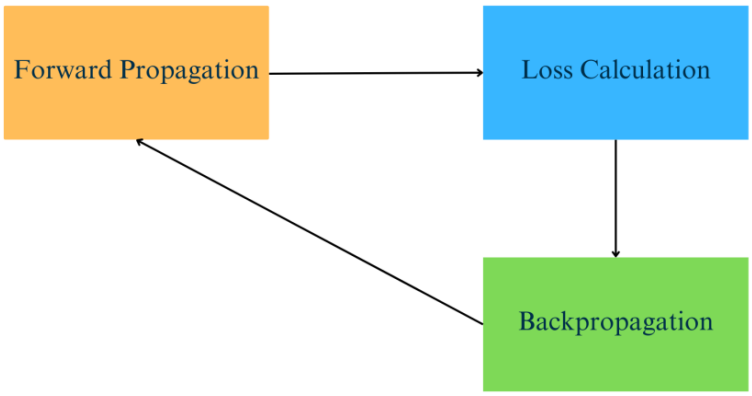
Tầng ẩn (Hidden Layer): Các tầng trung gian thực hiện phép biến đổi dữ liệu trích xuất đặc trưng. Một mạng có thể có nhiều tầng ẩn, càng nhiều tầng thì mô hình càng có khả năng học các đặc trưng phức tạp.

Tầng đầu ra (Output Layer): Cung cấp kết quả cuối cùng của mô hình. Số lượng nơ-ron ở tầng này phụ thuộc vào bài toán.



Hình 2: Cấu trúc của một mạng nơ-ron

2.2.1.2 Quy trình huấn luyện của mô hình học sâu



Hình 3: Quy trình học của mô hình học sâu

Lan truyền tiến (Forward Propagation) là quá trình tạo ra dự đoán từ dữ liệu bằng cách biến đổi dữ liệu đầu vào thông qua các lớp nơ-ron. ại mỗi lớp, dữ liệu đầu vào x được kết hợp với trọng số W và độ lệch b theo công thức:

$$z = W * x + b$$

Sau đó, giá trị z được đưa qua một hàm kích hoạt phi tuyến như ReLU hoặc Sigmoid để tạo ra đầu ra a. Đầu ra của lớp này sẽ tiếp tục trở thành đầu vào cho lớp tiếp theo, và quá trình này được lặp lại cho đến lớp cuối cùng để tạo ra dự đoán $\bar{y}$.

Sau khi có dự đoán, mô hình tiến hành tính toán hàm mất mát (Loss) nhằm đo lường mức độ sai lệch giữa $\bar{y}$ và giá trị thực tế y như là MSE (Mean

Square Error) hay Cross-Entropy Loss. Loss càng nhỏ đồng nghĩa với việc dự đoán càng chính xác, vì vậy nó đóng vai trò là kim chỉ nam để điều chỉnh mô hình trong giai đoạn lan truyền ngược (Backpropagation).

Trong lan truyền ngược, mô hình tính toán Gradient của Loss đối với từng trọng số bằng cách sử dụng quy tắc chuỗi, từ đó xác định mức độ ảnh hưởng của từng trọng số đến sai số. Các tham số sau đó được cập nhật bằng thuật toán Gradient Descent. Ngoài ra, các kỹ thuật hỗ trợ như Dropout và Batch Normalization giúp tăng tốc độ hội tụ và giảm overfitting cũng được sử dụng để nâng cao hiệu quả huấn luyện mô hình.

2.2.2 Thị giác máy tính

Thị giác máy tính là một lĩnh vực của khoa học máy tính và trí tuệ nhân tạo tập trung vào việc cho phép máy móc "nhìn" và hiểu thế giới xung quanh từ hình ảnh và video. Mục tiêu là mô phỏng khả năng nhìn và hiểu của con người, cho phép máy móc thực hiện các nhiệm vụ như nhận dạng đối tượng, theo dõi chuyển động, xử lý hình ảnh, và thậm chí là hiểu được cảnh quan và môi trường xung quanh.

2.2.3 Transfer Learning

Transfer learning là một phương pháp trong học máy mà nó cho phép một mô hình được phát triển cho một nhiệm vụ được sử dụng làm điểm khởi đầu cho một mô hình trên một nhiệm vụ khác.

Nó dựa trên ý tưởng rằng nếu một mô hình được huấn luyện trên một nhiệm vụ lớn và chung chung hơn, mô hình đó có thể được điều chỉnh để thực hiện các nhiệm vụ liên quan nhưng cụ thể hơn mà không cần bắt đầu từ đầu.

2.2.3.1 Fine-tuning

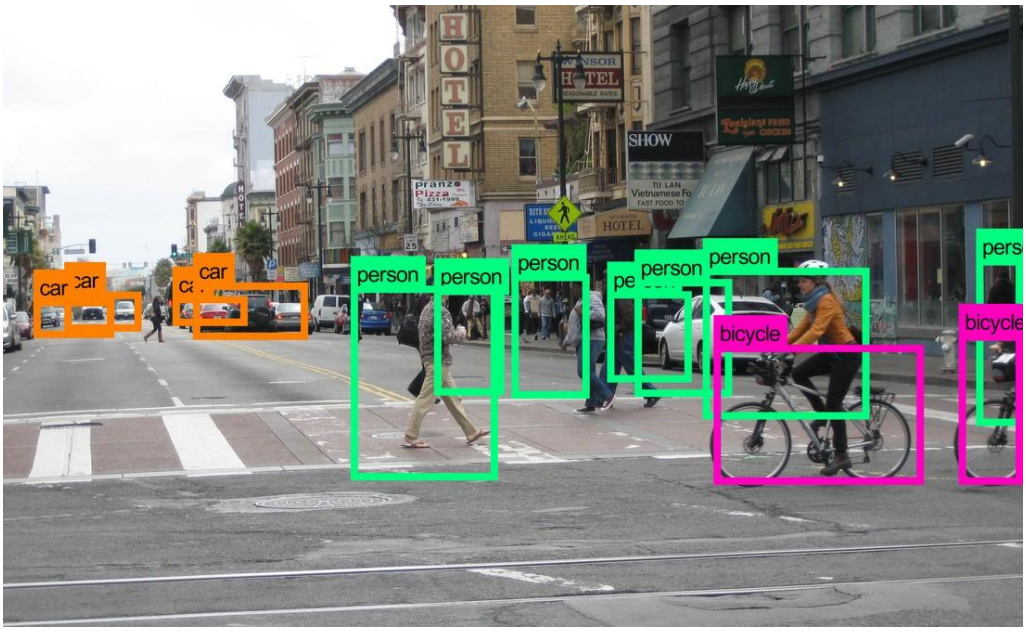
Fine-tuning là quá trình điều chỉnh một mô hình đã được đào tạo trước trên một tập dữ liệu mới hoặc cho một nhiệm vụ cụ thể. Mục tiêu là tận dụng kiến thức đã học được từ một tập dữ liệu lớn và áp dụng nó cho một tập dữ liệu nhỏ hơn hoặc một nhiệm vụ cụ thể. Mục tiêu là tận dụng kiến thức đã học được từ một tập dữ liệu lớn và áp dụng nó cho một tập dữ liệu nhỏ hơn hoặc một nhiệm vụ khác biệt, giúp cải thiện hiệu suất của mô hình mà không cần bắt đầu việc đào tạo từ đầu.

2.3 PHÁT HIỆN ĐỐI TƯỢNG

2.3.1 Nhận dạng đối tượng

Nhận dạng đối tượng trong thị giác máy tính là quá trình xác định và xác định vị trí của một hoặc nhiều đối tượng cụ thể trong một hình ảnh hoặc video. Quá trình này bao gồm hai bước chính:

- Object Detection: Xác định vị trí của đối tượng trong hình ảnh. Điều này thường được thực hiện bằng cách vẽ một hộp giới hạn (bounding box) xung quanh mỗi đối tượng được phát hiện.
- Recognition and Classification: Xác định loại hoặc danh mục của đối tượng đó, chẳng hạn như phân biệt một chiếc xe hơi với một con mèo.



Hình 4: Nhận dạng đối tượng

Nhận dạng đối tượng sử dụng một loạt các thuật toán và kỹ thuật máy học, đặc biệt là mạng nơ-ron sâu, để học từ một lượng lớn dữ liệu ảnh được gán nhãn trước. Quá trình học này cho phép mô hình phân biệt các đặc điểm và mẫu vật của đối tượng, giúp nó nhận diện và phân loại đối tượng một cách chính xác trong các hình ảnh mới, ngay cả trong các điều kiện khác nhau về ánh sáng, góc nhìn, và bối cảnh.

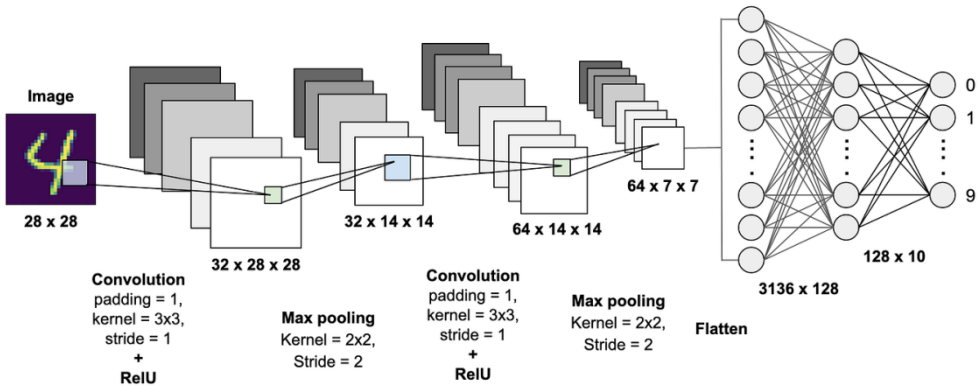
Nhận dạng đối tượng có nhiều ứng dụng thực tế, từ việc tự động gắn thẻ hình ảnh trong các dịch vụ chia sẻ ảnh, hệ thống giám sát an ninh, xe tự lái, đến phân loại và phát hiện bệnh trong ảnh chụp y tế [2].

2.3.2 Phân lớp đối tượng

Phân lớp đối tượng là một lĩnh vực quan trọng và cơ bản của thị giác máy tính, nhằm mục đích nhận diện và gán nhãn cho các đối tượng cụ thể trong ảnh số. Quá trình này thường được thực hiện thông qua các thuật toán học máy đặc biệt là học sâu, với sự hỗ trợ của các mạng nơ-ron tích chập.

Nguyên lý cơ bản của phân lớp đối tượng bắt đầu từ việc chuyển đổi ảnh chứa đối tượng thành dữ liệu số, thường dưới dạng ma trận các giá trị pixel. Các mạng nơ-ron tích chập sau đó sẽ trích xuất đặc trưng từ vùng ảnh chứa đối tượng thông qua các tầng tích chập, sử dụng các bộ lọc để nhận diện

các mẫu đặc trưng của đối tượng như đường nét, hình dạng, góc cạnh hay kết cấu. Các tầng tổng hợp giúp giảm kích thước dữ liệu, đồng thời giữ lại thông tin quan trọng về đối tượng, từ đó tăng hiệu quả tính toán. Cuối cùng, các tầng kết nối đầy đủ và hàm kích hoạt như softmax sẽ phân lớp đối tượng vào các nhóm (nhãn) đã được định nghĩa dựa trên xác suất.



Hình 5: Sử dụng mạng nơ-ron tích chập trong phân lớp đối tượng

Ứng dụng của phân lớp rất đa dạng, từ nhận diện khuôn mặt [3], chẩn đoán y khoa qua ảnh X-quang, đến nhận dạng biển số xe trong giao thông. Tuy nhiên, thách thức lớn bao gồm việc xử lý ảnh có chất lượng thấp, biến đổi ánh sáng, hoặc số lượng lớp phân loại quá lớn, đòi hỏi sự cải tiến liên tục trong thuật toán và tài nguyên tính toán.

2.3.3 Một số khái niệm khác

2.3.3.1 RoI

RoI (Region of Interest) là một khái niệm trung tâm trong các mô hình object detection dựa trên đề xuất vùng (region-based), chẳng hạn như R-CNN, Fast R-CNN, và Faster R-CNN. RoI đại diện cho các vùng hình ảnh được xác định là tiềm năng chứa vật thể, thường được tạo ra bởi một mạng đề xuất vùng (Region Proposal Network – RPN) hoặc phương pháp truyền thống như Selective Search. Những vùng này sau đó được trích xuất đặc trưng và phân loại để xác định chính xác vật thể và vị trí của chúng.

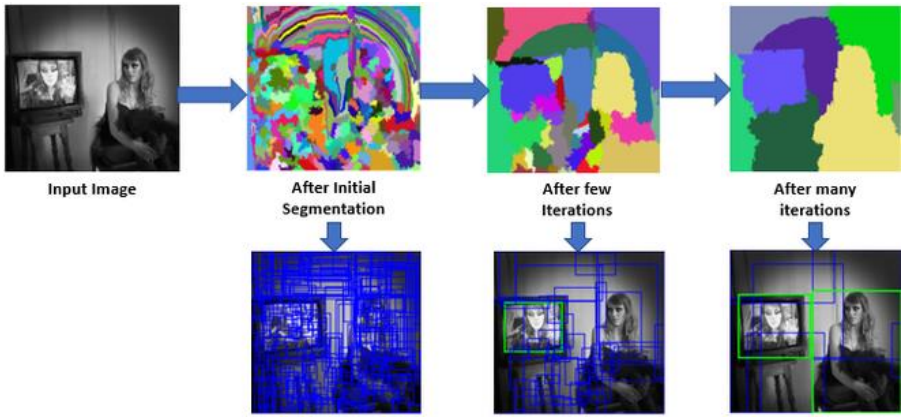
2.3.3.2 Anchor box

Anchor Box là các hộp giới hạn (bounding box) có kích thước và tỷ lệ cố định. Khi quét qua một ảnh đầu vào, tại mỗi vị trí trên feature map, mô hình sẽ gán một hoặc nhiều Anchor Box với các tỷ lệ khác nhau để dự đoán có đối tượng hay không. Nó được sử dụng trong các mô hình hiện đại như Faster R-CNN, YOLO, và SSD, khắc phục hầu hết hạn chế của phương pháp truyền thống Sliding Window.

Anchor Box là một bước quan trọng trong Object Detection, giúp mô hình dự đoán chính xác bounding box mà không cần kiểm tra từng pixel. Nhờ Anchor Box, các mô hình như Faster R-CNN, SSD, và YOLO có thể phát hiện đối tượng hiệu quả và nhanh chóng hơn.

2.3.3.3 Selective Search

Selective Search là một thuật toán đề xuất vùng (region proposal) trong bài toán object detection, được giới thiệu để giảm thiểu số lượng vùng ảnh cần xử lý so với phương pháp trượt cửa sổ (sliding window) truyền thống [4]. Mục tiêu của Selective Search là tạo ra các vùng quan tâm (RoI) có khả năng chứa vật thể bằng cách kết hợp phân vùng ảnh và hợp nhất vùng dựa trên đặc điểm hình ảnh.



Hình 6: Thuật toán Selective Search

2.3.3.4 Non-Max Suppression

Non-Maximum Suppression (NMS) là một kỹ thuật hậu xử lý quan trọng trong các hệ thống phát hiện đối tượng được sử dụng để loại bỏ các dự đoán dư thừa và chồng chéo[5]. Đây là bước không thể thiếu trong quy trình xử lý của hầu hết các thuật toán phát hiện đối tượng hiện đại như R-CNN, Faster R-CNN, SSD, và YOLO.

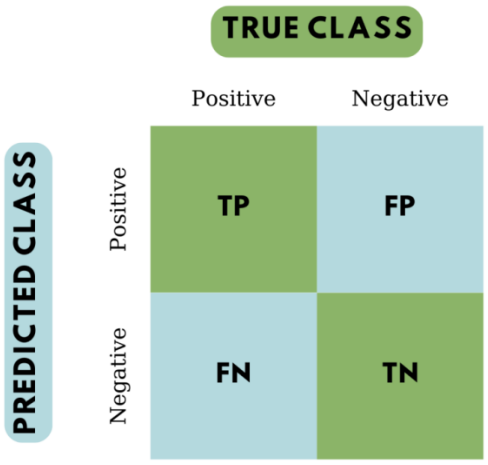
Hình 7: Mô tả Non-Maximum Suppression hoạt động

Non-Maximum Suppression là bước không thể thiếu trong pipeline của object detection, giúp tối ưu hóa kết quả đầu ra và nâng cao độ chính xác của mô hình. Dù có một số hạn chế, NMS vẫn được sử dụng rộng rãi nhờ tính đơn giản và hiệu quả.

2.4 PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH

2.4.1 Ma trận hỗn độn (Confusion Matrix)

Ma trận hỗn độn là một công cụ đánh giá hiệu suất quan trọng trong học máy, đặc biệt cho các bài toán phân loại. Nó cung cấp cái nhìn tổng quan về khả năng dự đoán của mô hình bằng cách so sánh kết quả dự đoán với giá trị thực tế.



Hình 8: Ma trận hỗn độn

Đối với bài toán phân loại nhị phân (binary classification), ma trận hỗn độn là một bảng 2×2 gồm:

- *True Positive* (TP): Số lượng mẫu dương tính được dự đoán đúng là dương tính.
- *False Positive* (FP): Số lượng mẫu âm tính được dự đoán sai là dương tính.
- *False Negative* (FN): Số lượng mẫu dương tính được dự đoán sai là âm tính.
- *True Negative (TN): Số lượng mẫu âm tính được dự đoán đúng là âm tính.*

Từ ma trận hỗn độn, có thể tính toán nhiều chỉ số đánh giá quan trọng:

- *Độ chính xác* (Accuracy): Tỷ lệ mẫu được phân loại đúng trên tổng số mẫu.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- *Độ chính xác dương tính* (Precision): Trong số các dự đoán dương tính, bao nhiêu là đúng.

$$Precision = \frac{TP}{TP + FP}$$

- *Độ nhạy* (Recall/Sensitive): Trong số các mẫu thực sự dương tính, bao nhiêu được phát hiện đúng.

$$Recall = \frac{TP}{TP + FN}$$

- *F1-Score*: Trung bình điều hòa của Precision và Recall, cân bằng cả hai chỉ số.

$$F1 = \frac{(Recall * Precision)}{(Recall + Precision)}$$

Ma trận hỗn độn có thể mở rộng cho bài toán đa lớp (multi-class) với cấu trúc n×n, trong đó n là số lượng lớp. Ma trận hỗn độn có ý nghĩa rất quan trọng trong việc đánh giá các mô hình phân loại. Nó hiển thị số lượng dự đoán đúng và sai dưới dạng một bảng, giúp hiểu rõ cách mô hình phân loại từng lớp.

2.4.2 IoU

Intersection over Union (IoU), là một trong những chỉ số đánh giá quan trọng nhất trong bài toán phát hiện đối tượng [6]. Một cách đơn giản IoU được định nghĩa là tỷ số giữa diện tích phần giao nhau (intersection) và diện tích hợp (union) của hai hình chữ nhật, giá trị của nó sẽ nằm từ 0 đến 1.

$$\text{IoU} = \frac{Diện\ tích\ phần\ giao}{Diện\ tích\ phần\ hợp}$$

Trong đó:

- Diện tích phần giao (Intersection): là phần diện tích chung giữa hộp dự đoán và hộp thực tế là bounding box dự đoán.
- Diện tích phần hợp (Union): là tổng diện tích của hai hộp, trừ đi phần giao nhau.

IOU là một chỉ số cơ bản và quan trọng trong object detection, đóng vai trò trung tâm trong cả huấn luyện, đánh giá và hậu xử lý. Dù có một số hạn chế, nó vẫn là tiêu chuẩn phổ biến nhờ tính đơn giản và hiệu quả.

2.4.3 Mean Average Precision (mAP)

Mean Average Precision (mAP) là chỉ số đánh giá chính cho các mô hình phát hiện đối tượng. Nó được tính bằng giá trị trung bình của Average Precision (AP) trên tất cả các lớp. AP đo lường sự cân bằng giữa Precision và Recall cho từng lớp, thường được tính là diện tích dưới đường cong Precision-Recall. mAP cung cấp một đánh giá tổng thể về hiệu suất phát hiện đối tượng của mô hình trên toàn bộ các lớp.

2.5 CÔNG CỤ VÀ MÔI TRƯỜNG TRIỂN KHAI

2.5.1 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình cấp cao, đa mục đích, được Guido van Rossum tạo ra vào cuối những năm 1980 và được phát hành lần đầu vào năm 1991. Ngôn ngữ này được thiết kế với mục tiêu chính là đọc và viết mã dễ dàng, với cú pháp rõ ràng và sạch sẽ.

Python hỗ trợ nhiều lĩnh vực lập trình khác nhau, từ phát triển web, phát triển phần mềm, cho đến khoa học dữ liệu, trí tuệ nhân tạo và hơn thế nữa.

2.5.2 Pytorch

PyTorch là một thư viện học máy mã nguồn mở được phát triển bởi Facebook AI Research vào năm 2016, đã nhanh chóng trở thành một trong những framework phổ biến nhất trong lĩnh vực học sâu (deep learning). Điểm mạnh nổi bật của PyTorch là cách tiếp cận tính toán đồ thị động (dynamic computational graph), cho phép người dùng xây dựng, chỉnh sửa và thực thi các mô hình neural network một cách linh hoạt trong quá trình chạy, đây là ưu điểm so với các framework sử dụng đồ thị tĩnh như TensorFlow phiên bản đầu.

Ngoài ra, PyTorch còn cung cấp nhiều module tiện ích như torchvision, torchaudio và torchtext để hỗ trợ các ứng dụng trong xử lý ảnh, âm thanh và ngôn ngữ tự nhiên, tạo điều kiện thuận lợi cho việc nghiên cứu và phát triển các mô hình học sâu tiên tiến.

2.5.3 Thư viện OpenCV

OpenCV là thư viện phần mềm nguồn mở cho thị giác máy tính và học máy, hỗ trợ việc áp dụng nhận thức máy tính vào sản phẩm thương mại. Với hơn 2500 thuật toán, OpenCV giúp nhận diện khuôn mặt, theo dõi đối tượng,

và nhiều hơn nữa. Có cộng đồng lớn với hơn 47 nghìn người dùng và hơn 18 triệu lượt tải.

OpenCV được sử dụng bởi Google, Sony và nhiều startup. Hỗ trợ nhiều ngôn ngữ lập trình và hệ điều hành, nghiêng về ứng dụng thời gian thực và đang phát triển giao diện CUDA và OpenCL.

2.5.4 Ultralytics

Ultralytics nổi tiếng với chuỗi mô hình nhận diện đối tượng YOLO, đã đưa ngành thị giác máy tính tiến lên nhiều bước phát triển đáng kể. Phiên bản mới nhất của công ty, YOLOv11, nổi bật với khả năng nhận diện đối tượng và phân đoạn hình ảnh thời gian thực, được xây dựng trên nền tảng những tiến bộ mới nhất của học sâu và thị giác máy tính. Mô hình này đạt hiệu suất vượt trội về cả tốc độ và độ chính xác, khiến nó trở thành giải pháp lý tưởng cho nhiều ứng dụng khác nhau và có khả năng thích nghi với đa dạng nền tảng phần cứng.

2.5.5 Môi trường Google Colab

Google Colab, hay Colaboratory, là một dịch vụ sổ ghi chép Jupyter được lưu trữ bởi Google, cho phép sử dụng mà không cần cài đặt và cung cấp quyền truy cập miễn phí vào các nguồn tài nguyên tính toán bao gồm GPU và TPU. Colab đặc biệt phù hợp cho máy học, khoa học dữ liệu và giáo dục. Colab cho phép tạo sổ ghi chép mới hoặc tải lên và chọn từ các nguồn khác nhau như GitHub, Google Drive hoặc máy tính cá nhân.

2.5.6 Công cụ Roboflow

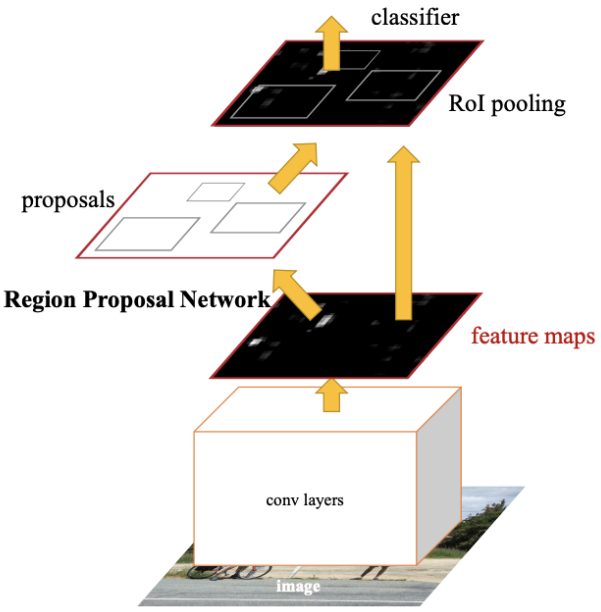
Roboflow là một nền tảng mạnh mẽ cho phép phát triển và triển khai các mô hình thị giác máy tính. Nền tảng này hỗ trợ từ việc tạo dataset, huấn luyện mô hình cho đến việc triển khai và cải thiện chúng.

Roboflow cung cấp khả năng làm việc với hơn 50.000 mô hình mã nguồn mở qua Roboflow Universe, và còn cho phép sử dụng các mô hình cơ sở như BLIP, DETIC, CLIP để cải thiện độ trễ thông qua việc tinh chỉnh với dữ liệu tùy chỉnh của bạn. Hơn nữa, nền tảng này hỗ trợ triển khai mô hình trên nhiều nền tảng khác nhau như NVIDIA Jetson, iOS, máy ảnh OAK, Raspberry Pi, trình duyệt web và đám mây riêng của bạn, với cơ sở hạ tầng được quản lý để sử dụng mô hình tùy chỉnh hoặc mô hình cơ sở dưới dạng điểm cuối API được lưu trữ. Hiện tại Roboflow hỗ trợ rất nhiều cho việc tạo ra dữ liệu cho việc huấn luyện các mô hình sử dụng file YOLO, COCO, JSON,…

2.6 MÔ HÌNH HỌC SÂU

2.6.1 Faster R-CNN

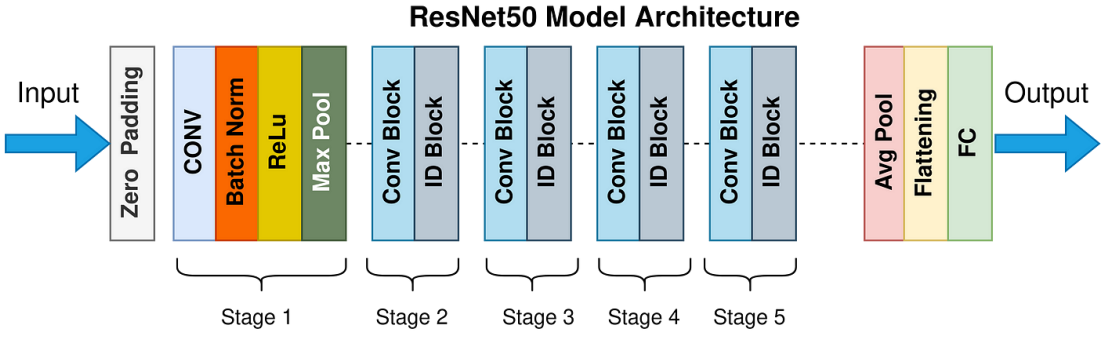
Faster R-CNN là mô hình tốt nhất của họ nhà R-CNN, được công bố đầu tiên vào năm 2015 [7]. So với các phương pháp trước, Faster R-CNN với cải tiến sử dụng một mạng tích chập được gọi là RPN (Region Proposal Network) để trích xuất các vùng đề xuất, đã cải thiện đáng kể tốc độ và độ chính xác của mô hình họ nhà R-CNN.



Hình 9: Mô hình Faster R-CNN

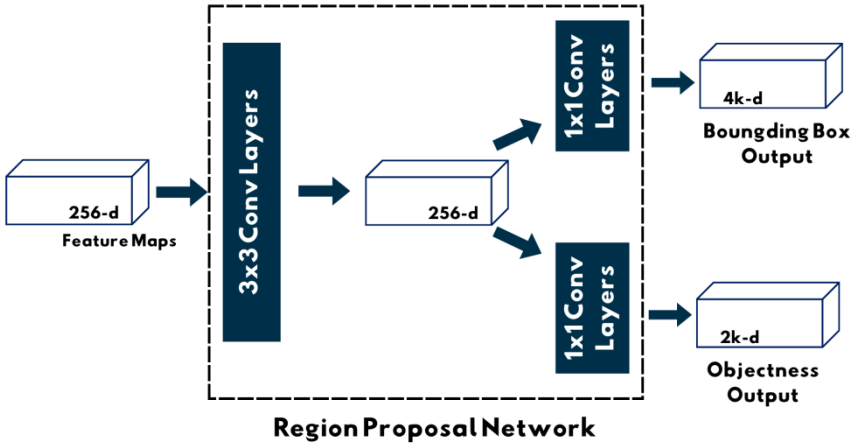
2.6.1.1 Kiến trúc của Faster R-CNN

Sự khác nhau lớn nhất giữa Faster R-CNN và mô hình tiền nhiệm trước đó là Fast R-CNN nằm ở phương pháp trích xuất, tạo các vùng đề xuất. Do đó backbone của Faster R-CNN cũng không có sự thay đổi quá nhiều. Vì thế VGG16, Resnet50 hay EfficientNet vẫn là sự lựa chọn tốt và tối ưu để trích xuất đặc trưng của hình ảnh.



Hình 10: Kiến trúc ResNet50

Faster R-CNN đã giới thiệu cho chúng ta một mạng tích chập có khả năng "học", trích xuất các vùng đề xuất được gọi là Region Proposal Network thay cho thuật toán Selective Search vốn tốn nhiều thời gian và không có khả năng linh động như mạng RPN.



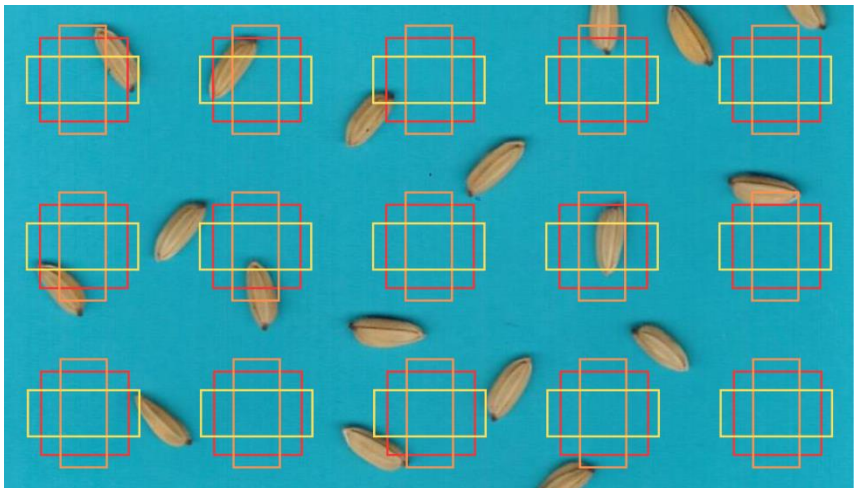
Hình 11: Cấu trúc của Region Proposal Network

Ảnh sau khi qua backbone để trích xuất đặc trưng và qua mạng Ragion Proposal để lấy các vùng đề xuất sẽ tiếp tục đi qua RoI pooling để chuyển đổi các vùng đề xuất khác nhau thành một kích thước cố định, cho phép mạng nơ-ron có thể xử lý hiệu quả các đặc trưng của từng vùng ảnh. Cuối cùng các đặc trưng sẽ được đưa qua các lớp kết nối đầy đủ (fully connected) để thực hiện hai nhiệm vụ chính: phân loại đối tượng và điều chỉnh vị trí chính xác của bounding box.

2.6.1.2 Cơ chế hoạt động của Faster R-CNN

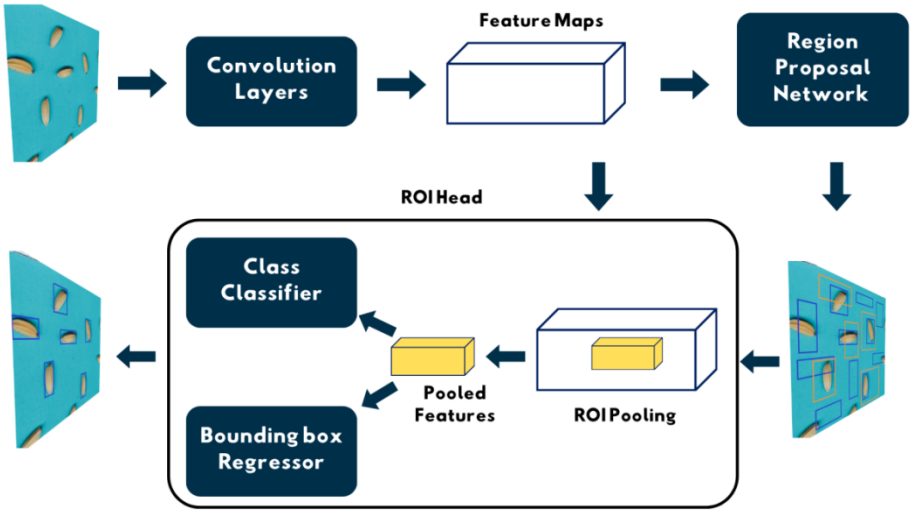
Hình ảnh đầu vào được đưa qua mạng nền (Backbone) như ResNet50 hoặc VGG16 để tạo feature map - một bản đồ đặc trưng phản ánh thông tin hình ảnh ở mức trừu tượng. Feature map này giữ lại các đặc điểm quan trọng như hình dạng, cạnh, hoặc kết cấu, làm cơ sở cho các bước tiếp theo.

Từ Feature map, RPN sinh ra các vùng đề xuất (proposals) bằng cách tạo k anchor box với tỷ lệ và kích thước khác nhau. Các anchor box này có vai trò dự đoán vật thể, các anchor box chứa vật thể sẽ được tiến hành điều chỉnh tọa độ (Hồi quy tọa độ) để khớp với vật thể thực. Thuật toán Non-Maximum Suppression được dùng để giữ lại các vùng có độ tin cậy cao nhất, loại bỏ các vùng trùng lặp.



Hình 12: Minh họa các anchor box được sinh ra

Qua RPN ta được các proposal có kích thước khác nhau, sau đó chúng được ánh xạ lên Feature map và chuẩn hóa về kích thước cố định qua ROI Pooling và tiến hành phân loại.



Hình 13: Quá trình phân loại và phát hiện đối tượng

Kết quả hệ thống trả về các bounding box đã tối ưu, nhãn lớp và độ tin cậy tương ứng, tạo thành kết quả phát hiện vật thể hoàn chỉnh.

2.6.2 YOLOv8

YOLOv8 là một trong những phiên bản đột phá của dòng mô hình phát hiện đối tượng thời gian thực YOLO, được Ultralytics phát hành vào năm 2023. Mô hình này mang lại hiệu suất vượt trội về cả độ chính xác và tốc độ xử lý, đồng thời mở rộng khả năng ứng dụng sang nhiều tác vụ thị giác máy tính khác

Kế thừa và phát triển từ nền tảng các phiên bản YOLO trước đó, YOLOv8 giới thiệu nhiều cải tiến quan trọng như backbone mạng mới, đầu

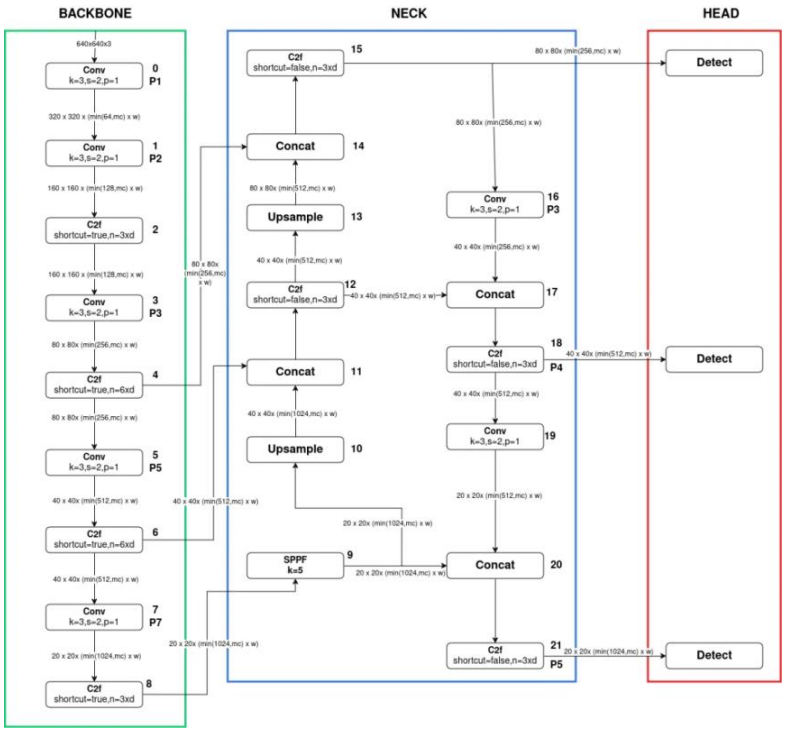
phát hiện được tối ưu hóa, và các kỹ thuật huấn luyện tiên tiến. Đặc biệt, YOLOv8 không chỉ giới hạn ở nhiệm vụ phát hiện đối tượng mà còn hỗ trợ đầy đủ các tác vụ phân đoạn hình ảnh (segmentation), ước tính tư thế (pose estimation), phân loại (classification), theo dõi đối tượng (tracking) và phát hiện đối tượng (detection).

Bảng 1: Bảng số liệu hiệu năng công khai chính thức của YOLOv8-Detection (trên tập dữ liệu COCO)

Mô hình	Kích cỡ (pixels)	mAP_{50-95}	Tốc độ CPU ONNX (ms)	Tốc độ A100 TensorRT (ms)	Tham số (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

2.6.2.1 Kiến trúc của YOLOv8

Kiến trúc của mô hình YOLOv8 gồm 3 thành phần: Backbone, Neck và Head.

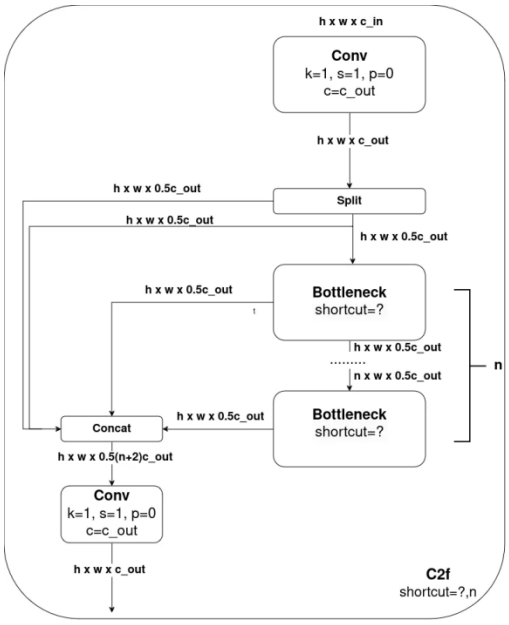


Hình 14: Kiến trúc của YOLOv8

Backbone của YOLOv8 là phiên bản cải tiến của CSPDarknet, được gọi là CSP-Darknet:

- Block cơ bản: Khối C2f
- Tầng đầu vào: Conv2d 3×3, stride=2, theo sau là SPPF (Spatial Pyramid Pooling - Fast).
- Các tầng chính: Một chuỗi các khối C2f với số lượng khối và kênh tăng dần theo độ sâu.
- Cấu trúc đa quy mô: Sử dụng các tầng Downsample để giảm kích thước không gian và tăng số kênh.
- Activation Function: YOLOv8 sử dụng SiLU (Swish) thay vì ReLU để tăng khả năng học đặc trưng phi tuyến tính.

Khối C2f cải tiến sử dụng kết nối dư (residual connections) và các thành phần bottleneck để tăng hiệu quả tính toán, đồng thời duy trì khả năng biểu diễn mạnh mẽ.



Hình 15: Kiến trúc của khối C2f trong Backbone của YOLOv8

Neck là phần tổng hợp và truyền đặc trưng từ Backbone đến Head. YOLOv8 kết hợp Feature Pyramid Network (FPN) và Path Aggregation Network (PAN).

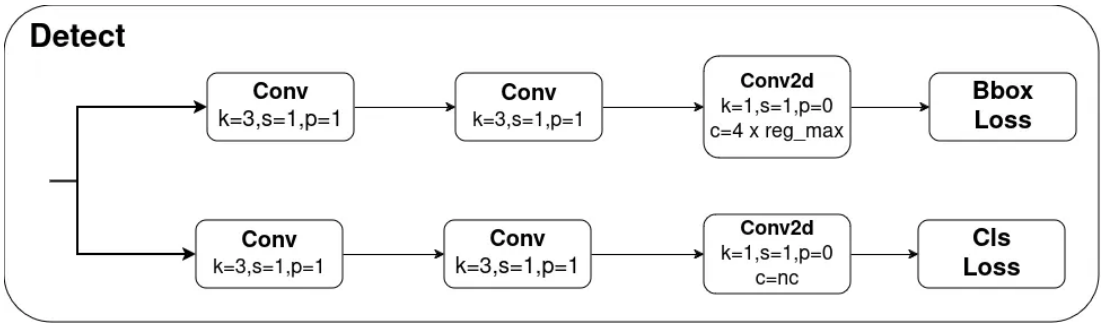
FPN (top-down): Truyền ngữ nghĩa từ các tầng sâu đến các tầng nông.

- Các tầng mở rộng (Upsample) được sử dụng để tăng độ phân giải không gian.
- Kết nối skip connections để kết hợp thông tin từ các mức đặc trưng khác nhau.

PAN (bottom-up): Truyền thông tin chi tiết từ các tầng nông đến các tầng sâu.

- Các khối C2f được sử dụng để kết hợp đặc trưng từ các tầng khác nhau.
- Các tầng giảm mẫu (Downsample) được sử dụng để giảm độ phân giải.

Head là phần dự đoán tọa độ bounding box và lớp của vật thể Khác với YOLOv5 sử dụng anchor-based, YOLOv8 chuyển sang anchor-free, loại bỏ sự phụ thuộc vào anchor boxes được định nghĩa trước.



Hình 16: Kiến trúc Detect của phần head

2.6.3 YOLOv11

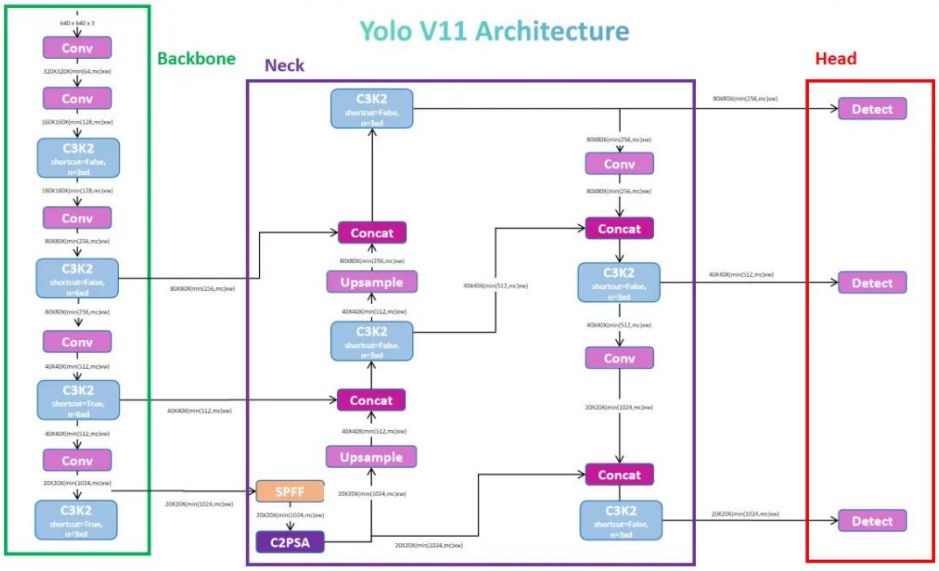
YOLOv11, được giới thiệu tại hội nghị YOLO Vision 2024 (YV24), là phiên bản gần đây nhất trong dòng mô hình phát hiện đối tượng thời gian thực YOLO. Mô hình này kế thừa các ưu điểm của các phiên bản trước, đồng thời thiết kế linh hoạt và kỹ thuật tối ưu hóa mang lại những cải tiến đáng kể về độ chính xác, tốc độ và hiệu quả cho nhiều tác vụ thị giác máy tính[8] .

Bảng 2: Bảng số liệu hiệu năng công khai chính thức của YOLOv11-Detection (trên tập dữ liệu COCO)

Mô hình	Kích cỡ (pixels)	mAP_{50-95}	Tốc độ CPU ONNX (ms)	Tốc độ A100 TensorRT (ms)	Tham số (M)	FLOPs (B)
YOLOv11n	640	39.5	56,1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLOv11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLOv11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLOv11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLOv11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

2.6.3.1 Kiến trúc của YOLOv11

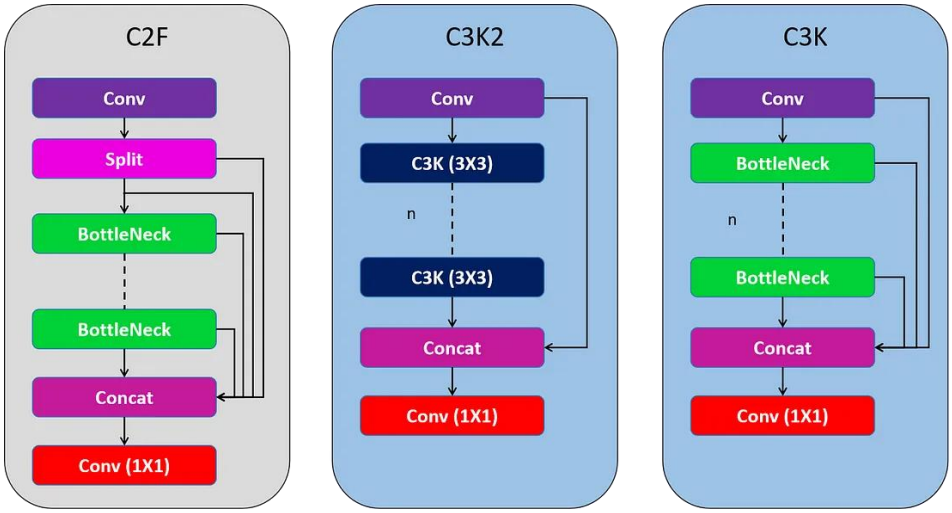
Tương tự với YOLOv8, phiên bản YOLO này cũng có kiến trúc gồm thành phần chính: Backbone, Neck và Head.



Hình 17: Kiến trúc của mô hình YOLOv11

Backbone của YOLOv11 trích xuất các đặc trưng từ ảnh đầu vào thông qua các tầng tích chập (Conv2D), chuẩn hóa (BatchNorm2D) và hàm kích hoạt SiLU tương tự như YOLOv8 nhưng có sự cải tiến.

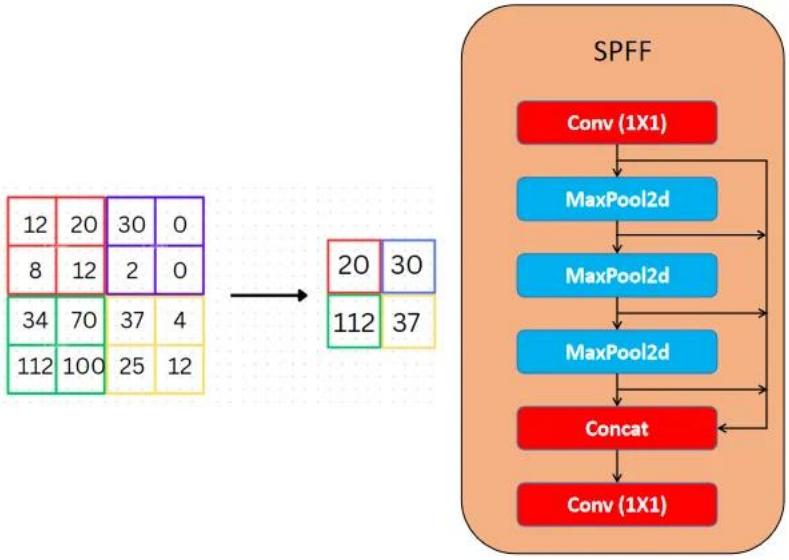
- C3k2 Block: Phiên bản cải tiến của cấu trúc Cross Stage Partial với kích thước kernel 2, giúp giảm số lượng tham số và tăng hiệu quả trích xuất đặc trưng
- SPPF**:** Tăng cường khả năng nhận diện đối tượng ở nhiều kích thước khác nhau mà không làm tăng đáng kể chi phí tính toán.
- C2PSA: Cơ chế chú ý không gian song song, giúp mô hình tập trung vào các vùng quan trọng trong ảnh, cải thiện độ chính xác trong việc phát hiện và phân đoạn đối tượng.



Hình 18: Các khối trong backbone của YOLOv11

YOLOv11 giữ lại mô-đun SPFF (Spatial Pyramid Pooling Fast), được thiết kế để tổng hợp đặc trưng từ các vùng khác nhau của hình ảnh ở nhiều quy mô khác nhau. Ngoài ra còn áp dụng mô-đun attention C2PSA để nâng cao khả năng tập trung vào các đặc trưng quan trọng trong không gian và kênh.

SPFF tổng hợp đặc trưng bằng cách sử dụng nhiều phép toán max-pooling (với các kích thước kernel khác nhau) để thu thập thông tin ngữ cảnh đa quy mô. Mô-đun này đảm bảo rằng ngay cả các đối tượng nhỏ cũng được mô hình nhận diện, vì nó kết hợp hiệu quả thông tin từ các độ phân giải khác nhau.



Hình 19: Neck trong kiến trúc của YOLOv11

Tương tự như các phiên bản YOLO trước đó, YOLOv11 sử dụng một đầu dự đoán đa quy mô để phát hiện các đối tượng ở các kích thước khác nhau. Đầu này xuất ra các hộp phát hiện cho ba quy mô khác nhau (thấp, trung bình, cao) bằng cách sử dụng các bản đồ đặc trưng được tạo ra bởi backbone và neck.

Đầu phát hiện xuất ra các dự đoán từ ba bản đồ đặc trưng (thường là P3, P4 và P5), tương ứng với các mức độ chi tiết khác nhau trong hình ảnh. Phương pháp này đảm bảo rằng các đối tượng nhỏ được phát hiện với chi tiết mịn hơn (P3), trong khi các đối tượng lớn hơn được ghi nhận bởi các đặc trưng cấp cao hơn (P5).

2.6.4 Detectron2

Detectron2 là một framework mạnh mẽ cho phát hiện đối tượng (object detection) được phát triển bởi Facebook AI Research (FAIR) [9], [10]. Đây là phiên bản nâng cấp từ Detectron ban đầu, được viết lại hoàn toàn bằng PyTorch, giúp cho việc nghiên cứu và triển khai các mô hình thị giác máy tính

trở nên dễ dàng hơn.

Với kiến trúc mô-đun hóa, Detectron2 hỗ trợ nhiều thuật toán phát hiện đối tượng tiên tiến như Faster R-CNN, Mask R-CNN, RetinaNet và DETR. Framework này cung cấp một tập hợp các mô hình được huấn luyện sẵn trên các bộ dữ liệu phổ biến như COCO và Pascal VOC, giúp người dùng có thể áp dụng transfer learning một cách hiệu quả.

Điểm mạnh của Detectron2 là khả năng mở rộng cao, cho phép người dùng tùy chỉnh các thành phần như backbone, head, loss function, và augmentation. Nó cũng hỗ trợ nhiều nhiệm vụ thị giác máy tính khác ngoài phát hiện đối tượng như phân đoạn ngữ nghĩa (semantic segmentation), phân đoạn instance (instance segmentation), và theo dõi đối tượng (object tracking).

Bảng 3: Bảng số liệu hiệu năng công khai chính thức của một số mô hình Detectron2 Model_Zoo

Mô hình	Backbone	Lr sched	Tập dữ liệu	AP box	AP_{50}	AP_{75}
Faster R – CNN	R50-FPN	1x	COCO	37.9	58.5	41.2
Faster R – CNN	R50-FPN	3x	COCO	40.2	61.0	43.8
Faster R – CNN	R101-FPN	1x	COCO	40.0	60.5	43.3
Faster R – CNN	R101-FPN	3x	COCO	42.0	62.5	45.9

CHƯƠNG 3.
THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1 MÔ TẢ BÀI TOÁN

Trong bối cảnh nông nghiệp hiện nay, việc phát hiện và phân loại hạt lúa giống có thể giúp cải thiện chất lượng sàng lọc giống, đánh giá chất lượng hạt và tối ưu hóa quá trình sản xuất.

"Ứng dụng mô hình học sâu trong phân loại hạt lúa giống" là bài toán phân loại đối tượng trong đó mô hình học sâu được áp dụng nhằm phân tích hình ảnh và tự động phân loại các loại hạt lúa trong một bức ảnh. Mô hình có khả năng trích xuất và nhận diện các đặc trưng riêng biệt của từng hạt lúa, từ đó hỗ trợ phân loại chính xác mà không cần can thiệp thủ công.

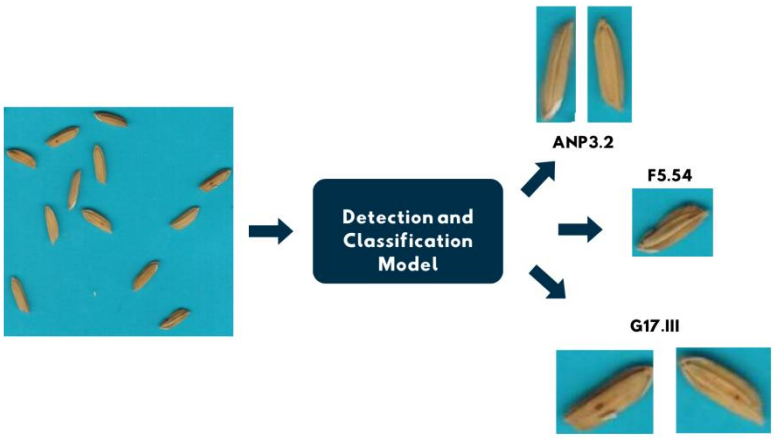
Các mô hình học sâu được sử dụng trong bài toán này dựa trên các kỹ thuật học sâu như mạng nơ-ron tích chập. Ngoài ra đề tài còn xây dựng các phương hướng giải quyết bài toán một giai đoạn với các mô hình riêng biệt và hai giai đoạn từ đó đưa ra sự so sánh khách quan.

Thông qua quá trình huấn luyện, mô hình học cách nhận diện và phân biệt các đặc điểm của hạt lúa giống, cho phép nó tự động thực hiện những công việc tương tự trên tập dữ liệu mới mà không cần sự can thiệp thủ công.

3.2 HƯỚNG TIẾP CẬN BÀI TOÁN

3.2.1 Sử dụng bộ phát hiện và phân loại tích hợp

Phương pháp này sẽ gộp quá trình phát hiện và phân loại vào một giai đoạn hay một mô hình duy nhất. Điều này cho phép hệ thống đồng thời phát hiện vị trí hạt lúa và phân loại trong một lượt suy luận, tối ưu hóa tốc độ xử lý.

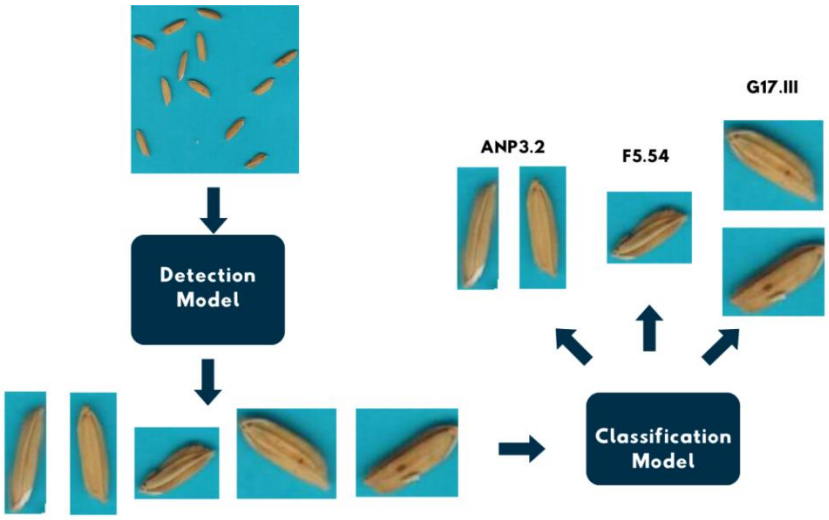


Hình 20: Hướng tiếp cận sử dụng mô hình phân loại và phát hiện tích hợp

3.2.2 Sử dụng bộ phát hiện và phân loại riêng biệt

Ở phương pháp này chúng tiến hành phân chia việc xây dựng mô hình thành 2 giai đoạn:

- Giai đoạn đầu tiên: Sử dụng mô hình YOLO để phát hiện và cắt ảnh hạt lúa được phát hiện.
- Giai đoạn thứ hai: Các ảnh hạt lúa được đưa vào hệ thống phân loại đa lớp dựa trên các mô hình phân loại: ResNet50, Vision Transformer (ViT), InceptionV3 và EffecientNet.



Hình 21: Hướng tiếp cận sử dụng mô hình phân loại và phát hiện riêng biệt

3.3 XÂY DỰNG DỮ LIỆU

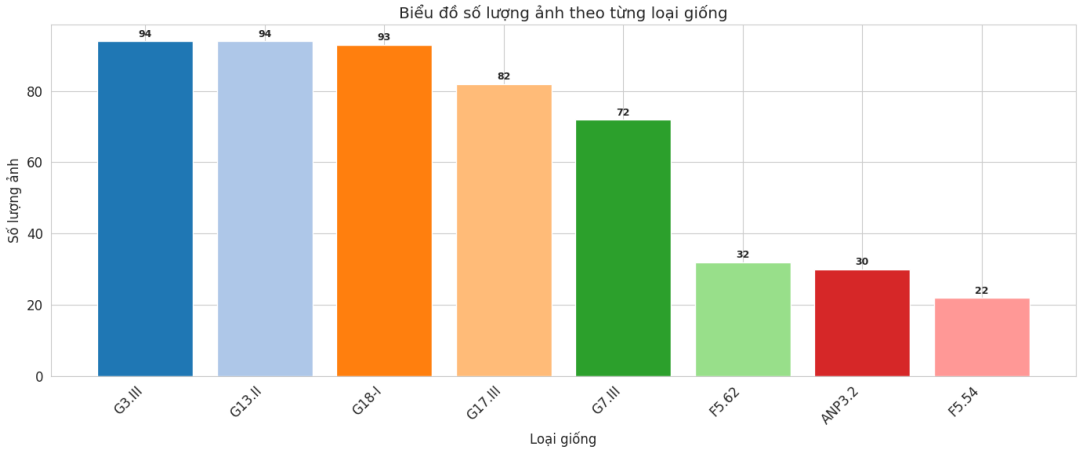
3.3.1 Thu thập dữ liệu

Dữ liệu được thu thập chủ yếu bao gồm hình ảnh của hạt lúa được Viện Biến Đổi Khí Hậu cung cấp. Dữ liệu bao gồm các hình ảnh thuộc 8 giống lúa khác nhau: ANP3.2, F5.54, F5.62, G13.II, G17.III, G18-I, G3.III và G7.III đảm bảo đa dạng về kích thước, màu sắc, và đặc tính khác. Điều này giúp nghiên cứu của chúng tôi có cái nhìn toàn diện và đa chiều về đối tượng nghiên cứu. Mỗi ảnh được thu thập đều có độ phân giải cao 1200 dpi và được đặt tên trùng với từng vị trí của bông lúa để thuận tiện cho việc nhận dạng và truy xuất. Việc sử dụng hình ảnh chất lượng cao góp phần đảm bảo độ chính xác và hiệu quả trong quá trình xử lý và phân tích dữ liệu hình ảnh sau này.

Hình 22: Dữ liệu hình ảnh các loại giống lúa

3.3.2 Tiền xử lý dữ liệu

Tôi bắt đầu chọn lọc lấy 519 ảnh từ 591 ảnh tập dữ liệu ảnh ban đầu trên tất cả giống lúa mà bên Viện Biến Đổi Khí Hậu đã cung cấp. Các ảnh được chọn lọc kỹ càng, chỉ giữ lại những hình ảnh rõ ràng, chứa đối tượng cần nhận diện. Những ảnh có yếu tố gây nhiễu như đối tượng bị chồng lấn quá nhiều, chụp quá gần, hoặc hạt lúa bị hư hỏng, mẻ, không nguyên vẹn... đã bị loại bỏ nhằm đảm bảo độ chính xác khi huấn luyện.

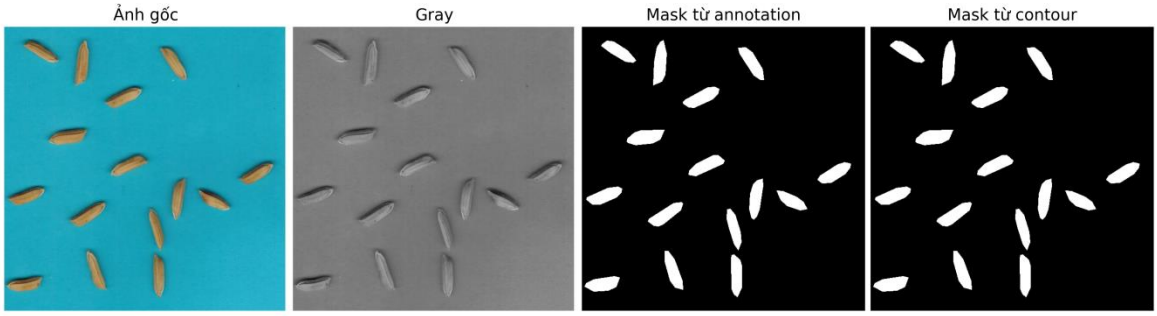


Biểu đồ 1: Biểu đồ dữ liệu ban đầu

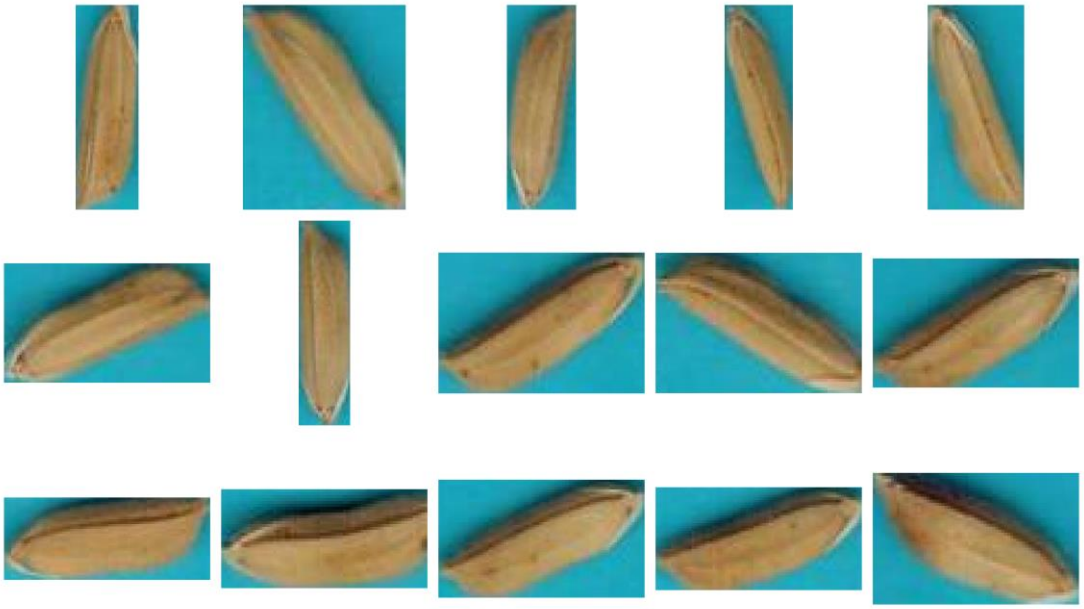
Do ảnh được chụp bằng máy scan nên có độ phân giải cao (1200 dpi), gây khó khăn cho xử lý và yêu cầu phần cứng lớn. Vì vậy, ảnh được giảm độ phân giải về 1024×1024 nhằm tiết kiệm tài nguyên tính toán. Bên cạnh đó, phần lớn ảnh chứa nhiều nền phụ và vật thể không liên quan, làm tăng nhiễu cho mô hình. Nhằm loại bỏ những phần không cần thiết, tôi đã cắt ảnh bằng khung 640×640 để chỉ giữ lại vùng chứa hạt lúa giống.

Hình 23: Ảnh gốc (bên trái), Ảnh đã resize (giữa), Ảnh đã crop (phải)

Đối với giai đoạn phân loại trong mô hình hai bước, ảnh đầu vào yêu cầu kích thước nhỏ hơn (tối đa 300×300 đối với các mô hình CNN truyền thống, 224×224 đối với ResNet50 hoặc ViT). Hơn nữa, ảnh đầu vào cần chỉ chứa một đối tượng duy nhất. Do đó, tôi thực hiện các bước xử lý chuyên sâu như: chuyển ảnh sang thang độ xám (gray scale) để giảm số kênh và đơn giản hóa đặc trưng ảnh; áp dụng kỹ thuật segmentation để tạo mask tách biệt đối tượng và nền; tìm các contour (biên đối tượng) để xác định chính xác vùng chứa hạt lúa.



Hình 24: Quá trình xử lý dữ liệu cho mô hình phân loại



Hình 25: Kết quả của quá trình xử lý dữ liệu cho mô hình phân loại

Từ các contour này, bounding box được tạo ra để trích xuất từng hạt lúa riêng biệt. Dù các ảnh hạt lúa sau khi cắt không có kích thước đồng nhất, nhưng vẫn đảm bảo yêu cầu cơ bản của mô hình phân loại. Sau đó, toàn bộ ảnh được chuyển qua mô-đun transformer tiền xử lý để điều chỉnh kích thước phù hợp với đầu vào của các mô hình học sâu được sử dụng.

3.3.3 Gán nhãn cho ảnh

Việc gán nhãn dữ liệu được thực hiện thông qua công cụ annotation do nền tảng Roboflow cung cấp. Trong tập dữ liệu dành cho bài toán phát hiện và phân loại hạt lúa, quá trình gán nhãn bao gồm việc xác định và đánh dấu từng đối tượng hạt lúa bằng các công cụ như bounding box hoặc polygon, tùy theo hình dạng và độ phức tạp của đối tượng. Mỗi đối tượng được gán nhãn tương ứng với tên giống lúa cụ thể nhằm phục vụ cho mô hình nhận diện chính xác.

Hình 26: Nhãn đã gán của hạt lúa giống với bounding box

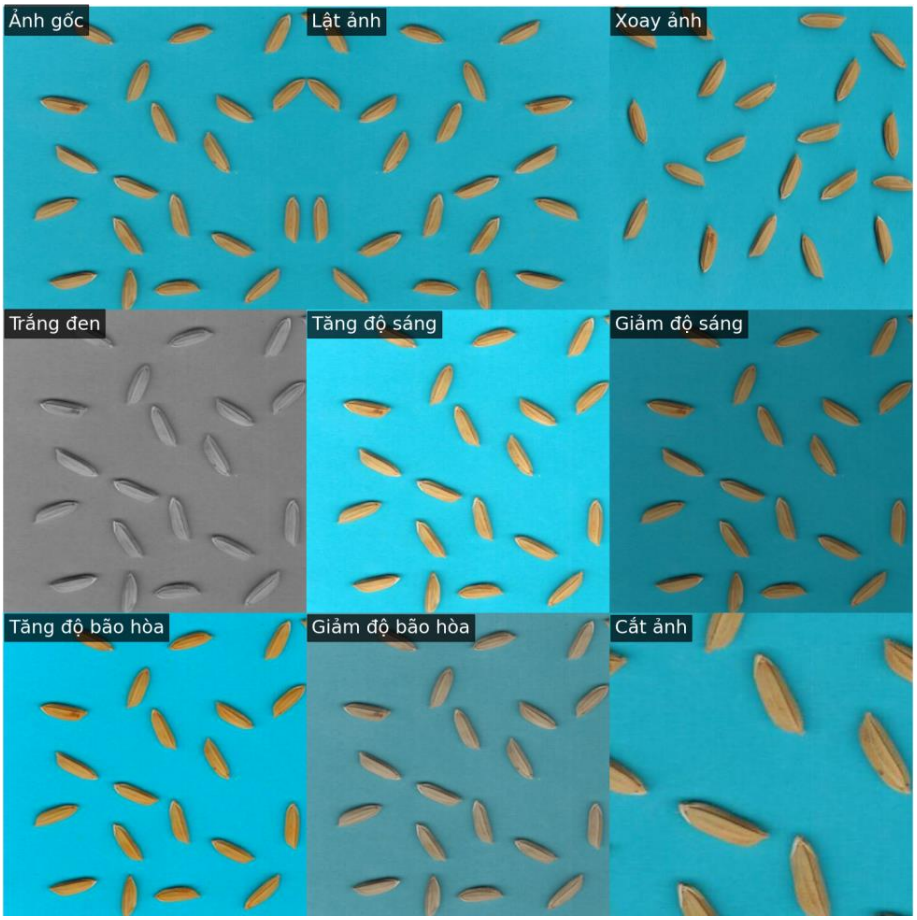
Đối với tập dữ liệu dùng cho mô hình phân loại trong bài toán hai giai đoạn, quy trình gán nhãn đơn giản hơn đáng kể. Các ảnh đã qua xử lý được tải trực tiếp lên Roboflow, sau đó chỉ cần chỉ định nhãn tương ứng cho từng ảnh. Nền tảng hỗ trợ thao tác này một cách trực quan và nhanh chóng, giúp tiết kiệm đáng kể thời gian chuẩn bị dữ liệu.

3.3.4 Tăng cường dữ liệu

Đối với bài toán có đối tượng là các vật thể nhỏ như hạt lúa giống, việc tăng cường dữ liệu đóng vai trò quan trọng trong việc nâng cao hiệu quả huấn luyện mô hình và khả năng tổng quát hóa.

Với các mô hình một giai đoạn, đã áp dụng nhiều kỹ thuật tăng cường như lật, xoay, điều chỉnh độ sáng – độ bão hòa, chuyển ảnh sang thang xám và phóng to ảnh nhằm đa dạng hóa tập dữ liệu huấn luyện và giảm thiểu hiện tượng quá khớp.

Đối với mô hình hai giai đoạn, do ảnh đầu vào là từng hạt lúa đã được tách riêng, quá trình tăng cường chỉ dừng lại ở việc thay đổi kích thước (resize) sao cho phù hợp với kiến trúc mô hình phân loại (ví dụ: 224×224 đối với ViT hoặc ResNet), thay vì áp dụng các kỹ thuật biến đổi hình ảnh phức tạp khác.



Hình 27: Tăng cường dữ liệu

3.3.5 Xuất tập dữ liệu

Sau khi hoàn tất các bước chuẩn bị và xử lý, tập dữ liệu cuối cùng được xuất ra từ nền tảng Roboflow với nhiều định dạng khác nhau, tương thích với các mô hình học sâu và ứng dụng thị giác máy tính.

Tập dữ liệu được chia thành ba phần với tỷ lệ như sau: 60% cho tập huấn luyện (train), 20% cho tập xác thực (validation) và 20% cho tập kiểm tra (test). Việc phân chia hợp lý giúp đánh giá hiệu suất mô hình một cách toàn diện.

Bảng 4: Bảng số liệu dữ liệu được chuẩn bị huấn luyện (một giai đoạn)

Tập	Số lượng mẫu			Tổng
	Train	Valid	Test	
Số lượng ảnh	936	104	103	1143
Số lượng thực thể	4015	1418	1413	6846

ANP32	305	119	116	540
F5.54	240	176	79	495
F5.62	247	184	62	493
G13.II	814	257	240	1311
G17.III	612	163	267	1042
G18.I	631	231	174	1036
G3.III	580	159	229	968
G7.III	586	129	246	961

Bảng 5: Bảng số liệu dữ liệu được chuẩn bị huấn luyện (hai giai đoạn)

Tập	Số lượng mẫu			Tổng
	Train	Valid	Test	
Số lượng ảnh	13458	1944	1934	17336
ANP32	1109	163	187	836
F5.54	945	134	130	667
F5.62	1142	159	173	818
G13.II	2443	363	363	1779
G17.III	1909	296	246	1374
G18.I	2101	306	282	1501
G3.III	2208	306	319	1568
G7.III	1601	217	234	1148

3.4 HUẤN LUYỆN MÔ HÌNH

3.4.1 Cấu hình và môi trường cài đặt

Môi trường thử nghiệm triển khai trên Google Colab với cấu hình phần cứng gồm CPU Intel Xeon, 12.7GB RAM và GPU 15GB, kết hợp hệ điều hành Windows 11 cùng các framework TensorFlow, PyTorch và Detectron2. Hệ thống sử dụng Python 3.10 cùng thư viện OpenCV, NumPy, Pandas để xử lý dữ liệu và huấn luyện mô hình.

Bảng 6: Thông số cấu hình và môi trường cài đặt

Môi trường	Google Collab
CPU	Itel Xeon
RAM	12,7 GB
GPU	15 GB
Hệ điều hành	Windows 11
Framework	Detectron2 , Pytorch
Ngôn ngữ lập trình	Python 3.10

3.4.2 Các mô hình một giai đoạn

3.4.2.1 Mô hình YOLO

Chúng tôi tiến hành thực nghiệm với các mô hình YOLOv8 và YOLOv11 cho tác vụ Object Detection. Cả hai mô hình này đều là những phiên bản hiện đại của YOLO. Các mô hình đã được thử nghiệm với các phiên bản tham số khác nhau để so sánh hiệu suất trong các tình huống khác nhau và tối ưu hóa kết quả phát hiện.

Bảng 7: Bảng cấu hình tham số huấn luyện YOLO

Mô hình	Phiên bản	Epoch	Batch Size	Learning Rate	Image Size	Optimizer	Patience
YOLOv8	n, s , m, l	100	16	0,005	640x640	Adam	30
YOLOv11							
YOLOv8							
YOLOv11							

3.4.2.2 Faster R-CNN

Bên cạnh việc triển khai và huấn luyện các mô hình YOLOv8 và YOLOv11 cho tác vụ Object Detection, nhằm thuận tiện trong việc so sánh khả năng giữa các mô hình, tôi đã tiến hành xây dựng thêm mô hình Faster R-CNN thông qua framework Detectron2.

Việc này giúp đánh giá toàn diện các phương pháp phát hiện đối tượng, từ đó đưa ra nhận xét và lựa chọn mô hình phù hợp nhất cho bài toán cụ thể.

Bảng 8: Bảng cấu hình tham số huấn luyện trên Detectron2

Mô hình	Backbone	Dataloader Workers	Batch Size	Learning Rate	Image Size	Số vòng lập iter
Faster R-CNN	ResNet50	4	8	0,0025	640x640	5000
Faster R-CNN	ResNet101	4	8	0,0025	640x640	5000

3.4.3 Các mô hình hai giai đoạn

Đối với các mô hình phân loại hai giai đoạn, quá trình huấn luyện và tinh chỉnh siêu tham số đã được thực hiện nhằm tối ưu hóa hiệu suất mô hình.

Trong quá trình này, công cụ *lr_finder* từ thư viện PyTorch đã được sử dụng để xác định tốc độ học (learning rate) phù hợp. Việc lựa chọn tốc độ học tối ưu có ý nghĩa quan trọng trong việc đẩy nhanh quá trình hội tụ và cải thiện độ chính xác của mô hình.

Bảng 9: Cấu hình tham số huấn luyện các mô hình hai giai đoạn

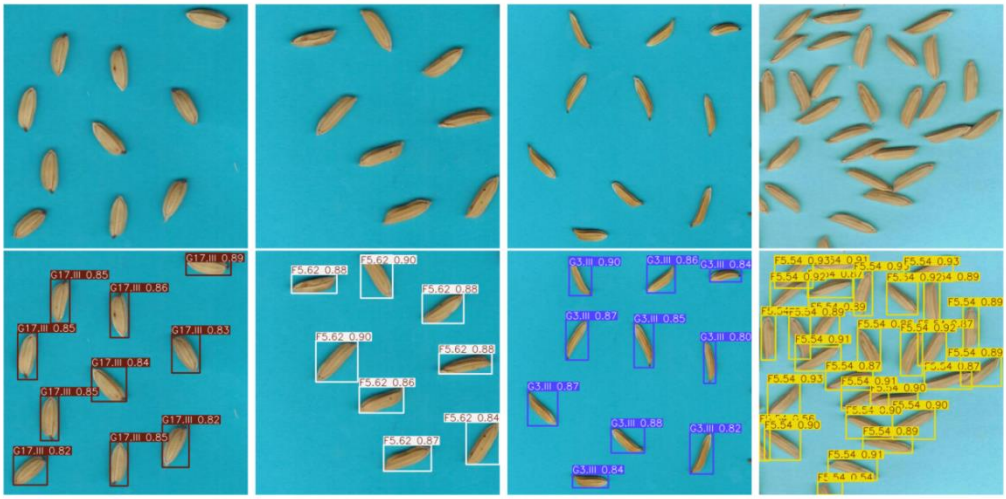
Mô hình	Phiên bản	Epochs	Learning rate	Batch size	Optimizer
ResNet	ResNet50	20	0.000117	32	Adam
EfficientNet	B3	20	0.000316	32	Adam
Inception	V3	20	0.000681	32	Adam
ViT	B16	15	0,000019	32	Adam

3.5 KẾT QUẢ HUẤN LUYỆN

3.5.1 Mô hình một giai đoạn

3.5.1.1 Mô hình YOLO

Nếu so sánh giữa hai phiên bản YOLOv8 và YOLOv11, có thể thấy rằng YOLOv11 nhìn chung thể hiện hiệu suất tốt hơn ở hầu hết các kích thước, đặc biệt là ở kích thước nano và small. Điều này cho thấy những cải tiến trong kiến trúc của YOLOv11 đã mang lại hiệu quả rõ rệt trong bài toán phát hiện đối tượng nhỏ như hạt giống lúa.

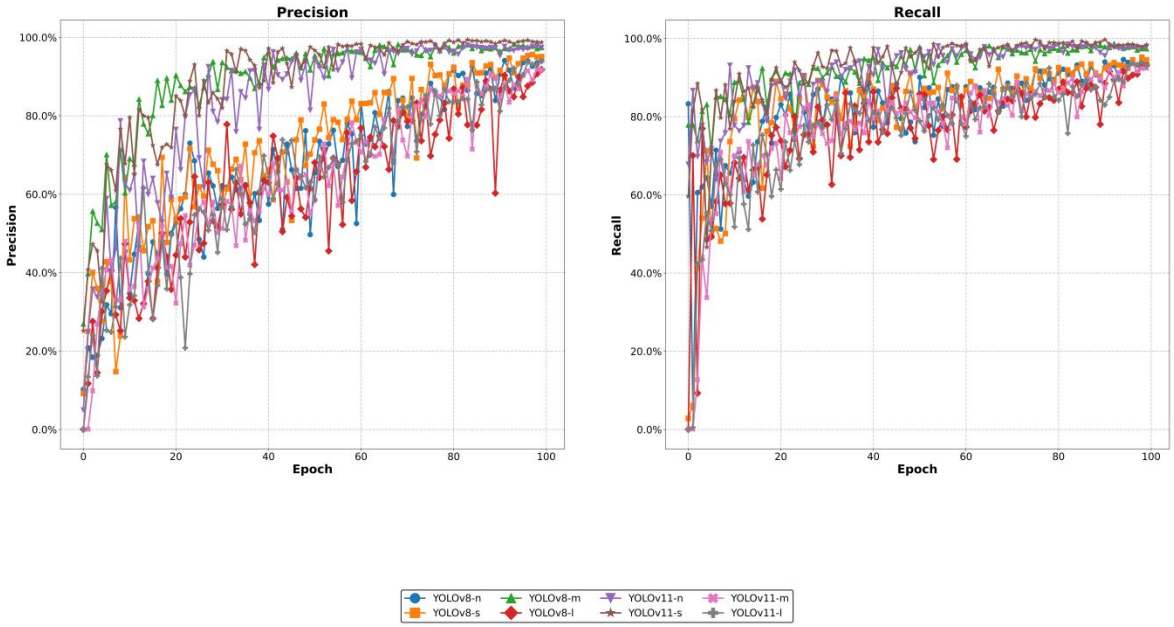


Hình 28:Kết quả dự đoán hình ảnh với mô hình YOLOv11s

Kết quả huấn luyện của các mô hình YOLO được tổng hợp thành bảng dưới. qua đó cung cấp cái nhìn toàn diện về hiệu quả của từng biến thể trong bài toán phân loại hạt giống lúa.

Bảng 10: Kết quả hiệu suất các mô hình tích hợp (YOLO)

Mô hình	**Precision**	**Recall**	**mAP$_{50}$**	**mAP$_{50-95}$**	**Thời gian (phút)**	**Thời gian Inference (giây)**	**FPS**
YOLOv8n	0,897	0,887	0,949	0,941	29	0,0143	66
YOLOv8s	0,922	0,912	0,965	0,958	34	0,0186	53
YOLOv8m	0,98	0,977	0,993	0,992	56	0,0342	29
YOLOv8l	0,876	0,871	0,924	0,919	90	0,0508	19
YOLOv11n	0,97	0,981	0,992	0,985	31	0,0152	60
YOLOv11s	**0,987**	**0,993**	**0,995**	**0,99**	**36**	**0,0176**	**58**
YOLOv11m	0,918	0,925	0,971	0,937	65	0,0382	28
YOLOv11l	0,936	0,933	0,976	0,952	86 phút	0,0413	25



Biểu đồ 2: Biểu đồ độ chính xác và độ nhạy

YOLOv11s nổi bật là mô hình hiệu quả nhất với các chỉ số precision (0,987), recall (0,993), mAP50 (0,995) và mAP50-95 (0,99) cao nhất. Đây là một kết quả đáng kinh ngạc, đặc biệt khi xét đến thời gian huấn luyện khá ngắn (36 phút) và tốc độ xử lý nhanh (58 FPS) của mô hình này. YOLOv11s thể hiện sự cân bằng lý tưởng giữa độ chính xác và hiệu suất tính toán, khiến nó trở thành lựa chọn tối ưu cho bài toán phát hiện hạt giống lúa.

YOLOv8m cũng thể hiện hiệu suất rất tốt với precision (0,98), recall (0,977), mAP50 (0,993) và mAP50-95 (0,992) chỉ kém YOLOv11s một chút, nhưng đòi hỏi thời gian huấn luyện dài hơn đáng kể (56 phút) và có tốc độ xử lý chậm hơn nhiều (29 FPS so với 58 FPS của YOLOv11s). Điều này cho thấy YOLOv8m là một mô hình mạnh về độ chính xác nhưng kém hiệu quả về mặt tài nguyên tính toán.

YOLOv11n cũng đạt hiệu suất rất tốt với precision (0,97), recall (0,981), mAP50 (0,992) và mAP50-95 (0,985), đồng thời có thời gian huấn luyện ngắn (31 phút) và tốc độ xử lý nhanh (60 FPS). Đây có thể là một lựa chọn tốt nếu cần triển khai trên thiết bị với nguồn tài nguyên hạn chế.

Đáng chú ý, YOLOv8l lại có hiệu suất thấp nhất trong tất cả các mô hình được thử nghiệm, với precision (0,876), recall (0,871), mAP50 (0,924) và mAP50-95 (0,919), đồng thời có thời gian huấn luyện lâu nhất (90 phút) và tốc độ xử lý chậm nhất (19 FPS). Kết quả này trái ngược với kỳ vọng thông thường rằng mô hình lớn hơn sẽ có hiệu suất tốt hơn, cho thấy YOLOv8l có thể bị overfit với dữ liệu huấn luyện hoặc không phù hợp với đặc thù của bài toán phát hiện hạt giống lúa.

Thông qua *Bảng 10*, tôi đã lựa chọn mô hình YOLOv11s làm đại diện cho các phiên bản YOLO trong bài toán này. Ngoài việc xây dựng ma trận hỗn độn trên tập test (xem *Bảng 4*) để đánh giá khả năng phân loại, mô hình này còn là cơ sở để so sánh với các mô hình thuộc các kiến trúc khác trong cùng phương pháp, cũng như so sánh giữa các phương pháp khác nhau.

Bảng 11: Ma trận hỗn độn (YOLOv11s)

		Lớp dự đoán							
		ANP3.2	F5.54	F5.62	G13.II	G17.III	G18.I	G3.III	G7.III
Lớp thực tế	ANP3.2	**117**	0	0	0	0	0	0	0
	F5.54	0	**161**	0	0	0	0	0	0
	F5.62	0	0	**168**	3	0	0	0	0
	G13.II	1	0	3	**250**	0	0	0	0
	G17.III	0	0	0	0	**154**	0	0	0
	G18.I	0	0	0	0	0	**215**	1	0
	G3.III	1	0	2	0	2	1	**147**	0
	G7.III	0	0	0	0	0	0	0	**122**

Ma trận hỗn độn cho thấy hiệu suất xuất sắc của mô hình YOLOv11s trong việc phân loại các lớp hạt giống lúa. Cụ thể, trong tổng số 1,347 mẫu (được tính bằng cách cộng tất cả các giá trị trong ma trận), chỉ có 14 mẫu bị phân loại sai**,** dẫn đến tỷ lệ lỗi chỉ khoảng 1,04%**,** tương đương với độ chính xác lên tới 98,96%**.**

Từ ma trận hỗn độn, tôi tiếp tục đánh giá khả năng phân loại của mô hình theo từng lớp và trình bày kết quả trong *Bảng 12*. Như ta có thể thấy, mô hình YOLOv11s thể hiện khả năng phân tích và phân loại ấn tượng, với một số lớp đạt hiệu suất gần như hoàn hảo, như F5.54 và G7.III. Cụ thể, F1-score thấp nhất của mô hình vẫn đạt 0.976, cho thấy hiệu suất vượt trội trong hầu hết các lớp. Điều này khẳng định rằng mô hình YOLOv11s là một công cụ cực kỳ hiệu quả cho việc phân loại tự động các loại hạt giống lúa.

Bảng 12: Khả năng phân loại theo từng lớp của mô hình (YOLOv11s)

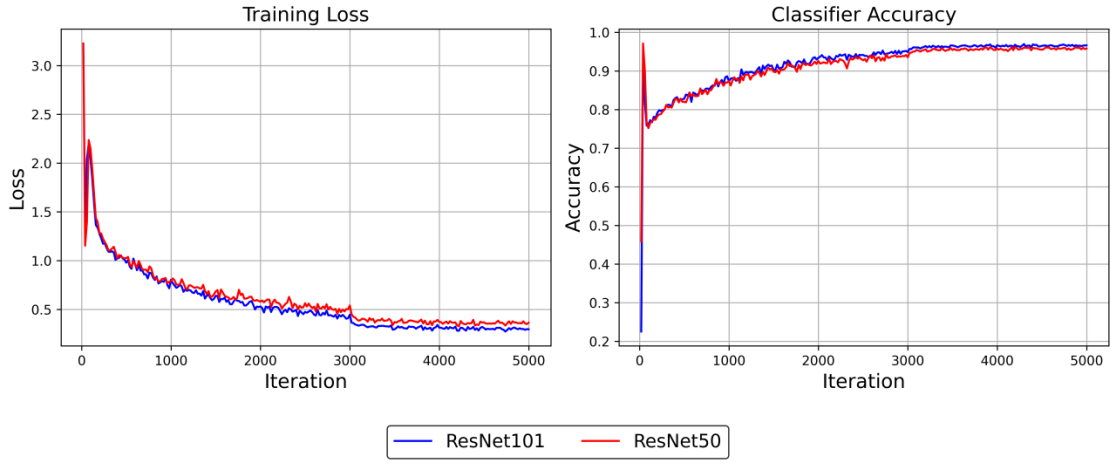
Lớp	**Precision**	**Recall**	**F1**
ANP3.2	0.991	1.000	0.996
F5.54	1.000	1.000	1.000
F5.62	0.971	0.982	0.976
G13.II	0.988	0.984	0.986
G17.III	0.987	1.000	0.993

G18.I	0.995	0.995	0.995
G3.III	0.993	0.961	0.977
G7.III	1.000	1.000	1.000

3.5.1.2 Mô hình Faster R-CNN

Bảng 13: Kết quả hiệu suất các mô hình tích hợp (Faster R-CNN)

Mô hình	Backbone	Recall	mAP_{50}	mAP_{75}	mAP_{50-95}	Thời gian (phút)	Thời gian Inference (giây)	FPS
Faster R-CNN	ResNet50 FPN	0,938	0,940	0,937	0,887	67	0,0917	10
Faster R-CNN	**ResNet101 FPN**	**0,947**	**0,948**	**0,946**	**0,904**	**83**	**0,1293**	**7**



Biểu đồ 3: Biểu đồ Loss và hiệu suất phân loại

Mô hình Faster R-CNN với backbone ResNet101 có hiệu suất nhận dạng tốt hơn so với ResNet50, thể hiện qua các chỉ số mAP. Điều này chứng tỏ rằng backbone ResNet101, với độ sâu lớn hơn, giúp mô hình nhận diện các đặc trưng phức tạp của hạt giống lúa một cách chính xác hơn ở nhiều ngưỡng IoU khác nhau.

Dựa trên đồ thị huấn luyện, ResNet101 thể hiện khả năng hội tụ nhanh hơn, với đường loss (màu xanh) giảm ổn định và nhanh chóng so với ResNet50 (màu đỏ). Đồng thời, trên đồ thị độ chính xác, ResNet101 duy trì ưu thế vượt trội, đạt độ chính xác cao hơn so với ResNet50.

Tuy nhiên, sự cải thiện về hiệu suất này đi kèm với chi phí tính toán lớn hơn. Thời gian huấn luyện của mô hình ResNet101 FPN kéo dài 83 phút, lâu

hơn so với 67 phút của ResNet50 FPN. Đặc biệt, thời gian inference của ResNet101 là 0,1293 giây, chậm hơn so với 0,0917 giây của ResNet50, dẫn đến tốc độ xử lý ảnh giảm xuống còn 7 FPS. Mặc dù ResNet101 cho kết quả nhận dạng chính xác hơn, nhưng sự giảm sút 30% về tốc độ xử lý là một yếu tố quan trọng cần được cân nhắc khi triển khai mô hình trong các ứng dụng thực tế.

Đối với kiến trúc mô hình Faster R-CNN tôi chọn mô hình với backbone ResNet101 để xây dựng ma trận hỗn độn để đánh giá khả năng phân loại của nó.

Bảng 14: Ma trận hỗn độn (Faster R-CNN)

		Lớp dự đoán							
		ANP3.2	F5.54	F5.62	G13.II	G17.III	G18.I	G3.III	G7.III
Lớp thực tế	ANP3.2	**95**	0	1	12	3	0	0	2
	F5.54	0	**71**	0	0	0	0	0	0
	F5.62	3	0	**45**	7	0	3	2	0
	G13.II	1	0	0	**209**	0	2	0	11
	G17.III	0	1	0	1	**244**	0	0	1
	G18.I	0	0	1	4	0	**138**	22	3
	G3.III	0	0	0	1	1	15	**198**	4
	G7.III	1	0	1	22	0	2	1	**199**

Mô hình Faster R-CNN với backbone ResNet101 FPN cho thấy hiệu suất phân loại tương đối ổn định, thể hiện qua việc phần lớn các giá trị trong ma trận hỗn độn tập trung trên đường chéo chính – phản ánh rằng đa số đối tượng đã được nhận diện chính xác. Tuy nhiên, so với mô hình YOLOv11s đã được phân tích trước đó, ma trận hỗn độn của Faster R-CNN xuất hiện nhiều hơn các giá trị nằm ngoài đường chéo, cho thấy số lượng các trường hợp phân loại sai có xu hướng cao hơn.

Mặc dù đã tích hợp Feature Pyramid Network (FPN) để cải thiện khả năng nhận diện các vật thể nhỏ như hạt giống lúa, mức độ cải thiện về độ chính xác của mô hình không thực sự đáng kể. Để có được cái nhìn chi tiết hơn về hiệu suất mô hình trên từng lớp cụ thể, tương tự như với YOLOv11s, tôi tiếp tục sử dụng ma trận hỗn độn làm cơ sở để xây dựng *Bảng 13*. Qua đó, có thể phân tích điểm mạnh và hạn chế của mô hình trong việc phân biệt các loại hạt giống khác nhau một cách cụ thể và định lượng hơn.

Bảng 15: Khả năng phân loại theo từng lớp của mô hình (Faster R-CNN)

Lớp	Precision	Recall	F1
ANP3.2	0.950	0.841	0.891
F5.54	1.000	1.000	1.000
F5.62	0.978	0.750	0.849
G13.II	0.817	0.937	0.872
G17.III	0.984	0.988	0.986
G18.I	0.862	0.821	0.841
G3.III	0.888	0.904	0.896
G7.III	0.904	0.880	0.892

3.5.2 Mô hình hai giai đoạn

Sau khi hoàn tất quá trình huấn luyện trên các mô hình đã lựa chọn, kết quả thu được được tổng hợp và trình bày trong bảng dưới đây. Trên cơ sở các chỉ số đánh giá này, tôi sẽ tiến hành phân tích chi tiết hiệu suất của từng mô hình nhằm lựa chọn kiến trúc phù hợp nhất. Đồng thời, kết quả này cũng là cơ sở quan trọng để thực hiện so sánh giữa các mô hình thuộc phương pháp hai giai đoạn với các mô hình của phương pháp một giai đoạn như YOLO, từ đó đánh giá toàn diện hiệu quả của từng hướng tiếp cận trong thực tiễn ứng dụng

Bảng 16: Bảng kết quả mô hình phát hiện và phân loại riêng biệt

Mô hình YOLO phát hiện hạt giống lúa			
Độ chính xác (precision)	0,994		
Độ nhạy (recall)	0,998		
Độ chính xác trung bình (mAP)	0,995		
Các mô hình phân loại hạt giống lúa			
Mô hình	Độ chính xác (Accuracy)	Thời gian huấn luyện (phút)	Thời gian Inference (giây)
ViT	**0,95**	**180**	**0,0117**
ResNet50	0,925	56	0,0101
EfficientNetB3	0,90	140	0,0193
InceptionV3	0,95	70	0,0173

Mô hình YOLO cho thấy hiệu suất vượt trội trong nhiệm vụ phát hiện hạt giống lúa, với độ chính xác đạt 99,4%, độ nhạy (recall) đạt 99,8% và độ chính xác trung bình (precision) đạt 99,5%. Những chỉ số này tiệm cận ngưỡng tối ưu, cho thấy mô hình có khả năng phát hiện chính xác vị trí của các hạt giống lúa với độ tin cậy rất cao. Đặc biệt, giá trị recall cao cho thấy mô hình gần như không bỏ sót các đối tượng cần phát hiện.

Ở giai đoạn phân loại, hai mô hình ViT (Vision Transformer) và InceptionV3 đạt độ chính xác cao nhất, đều ở mức 95%. Tuy nhiên, ViT yêu cầu thời gian huấn luyện dài nhất (180 phút) so với chỉ 70 phút của InceptionV3. Điều này phản ánh kiến trúc phức tạp và nhu cầu tính toán cao của ViT, vốn được thiết kế dựa trên cơ chế self-attention. Mặc dù vậy, ViT vẫn thể hiện thời gian suy luận (inference) tương đối nhanh, chỉ 0,0117 giây gần tương đương với ResNet50 (0,0101 giây), mặc dù độ chính xác của ResNet50 thấp hơn đáng kể.

Ngược lại, EfficientNetB3 cho thấy hiệu suất kém hơn trong bài toán này, với độ chính xác chỉ 90% và thời gian inference dài nhất (0,0193 giây), dù được thiết kế nhằm tối ưu hóa cả độ chính xác và hiệu quả tính toán. Điều này cho thấy kiến trúc này không phù hợp với đặc điểm của tập dữ liệu hạt giống lúa trong thí nghiệm này.

Từ các kết quả thu được, tôi lựa chọn ViT làm đại diện cho các mô hình phân loại trong phương pháp hai giai đoạn, nhờ hiệu suất phân loại cao và khả năng suy luận nhanh. Tiếp theo, tôi xây dựng ma trận hỗn độn dựa trên dữ liệu từ *Bảng 5* nhằm phân tích chi tiết khả năng phân loại của từng loại hạt giống lúa mà mô hình ViT đạt được.

Bảng 17: Ma trận hỗn độn (mô hình ViT)

		Lớp dự đoán							
		ANP3.2	F5.54	F5.62	G13.II	G17.III	G18.I	G3.III	G7.III
Lớp thực tế	ANP3.2	**179**	0	2	1	1	1	2	1
	F5.54	0	**128**	1	0	0	1	0	0
	F5.62	1	1	**160**	0	2	7	2	0
	G13.II	2	0	0	**354**	1	0	1	5
	G17.III	1	0	1	0	**242**	1	0	1
	G18.I	0	0	1	0	1	**273**	6	1
	G3.III	0	0	1	0	0	6	**312**	0
	G7.III	0	0	0	2	2	20	9	**201**

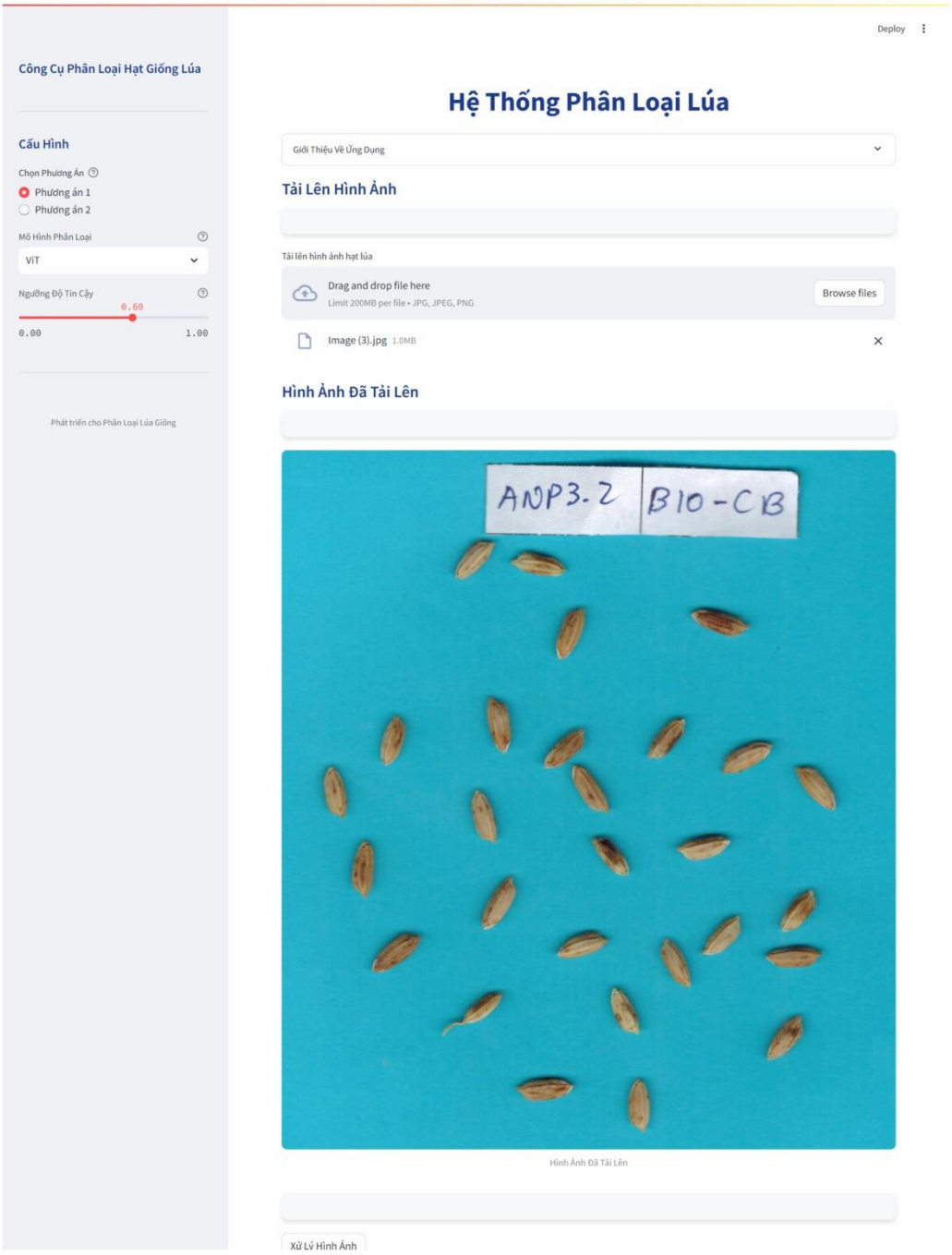
Bảng 18: Khả năng phân loại theo từng lớp của mô hình (ViT)

Lớp	Precision	Recall	F1
ANP3.2	0.978	0.957	0.967
F5.54	0.992	0.985	0.988
F5.62	0.976	0.930	0.953
G13.II	0.992	0.975	0.984
G17.III	0.988	0.988	0.988
G18.I	0.886	0.968	0.925
G3.III	0.966	0.978	0.972
G7.III	0.962	0.859	0.908

Qua *Bảng 18*, mô hình ViT thể hiện hiệu suất rất tốt với chỉ số F1 trung bình khoảng 0.961, cho thấy tiềm năng ứng dụng cao trong các tác vụ phân loại thực tế. Tuy nhiên, vẫn có thể cải thiện thêm hiệu suất ở các lớp G7.III và G18.I để đạt được kết quả đồng đều hơn giữa các lớp.

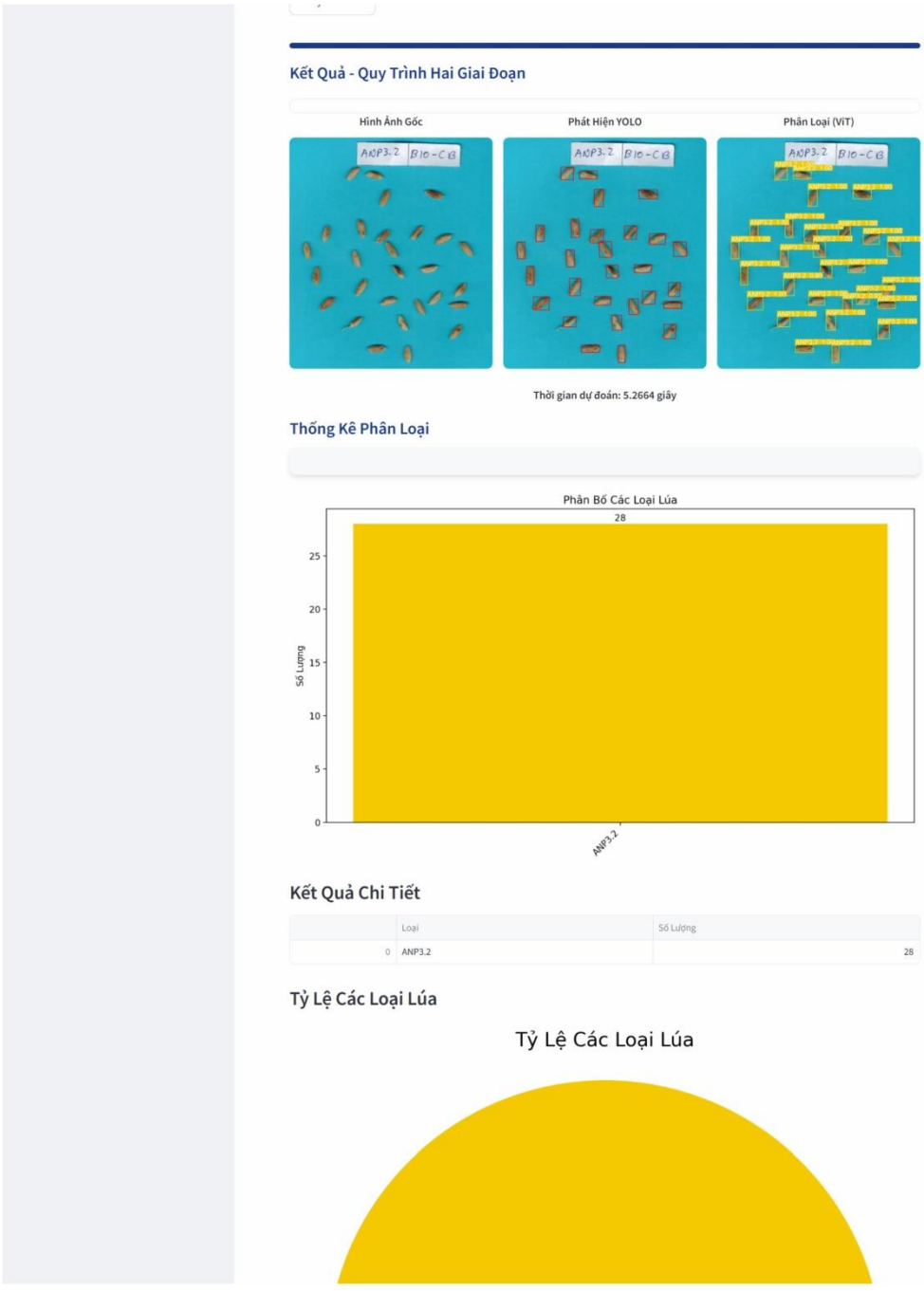
3.5.3 Website phân loại hạt lúa giống

Bên cạnh quá trình huấn luyện và đánh giá các mô hình học sâu theo từng phương pháp, tôi cũng đã xây dựng một hệ thống phân loại hạt giống lúa với giao diện trực quan trên nền tảng Website, sử dụng công cụ Streamlit nhằm hỗ trợ quá trình xây dựng.

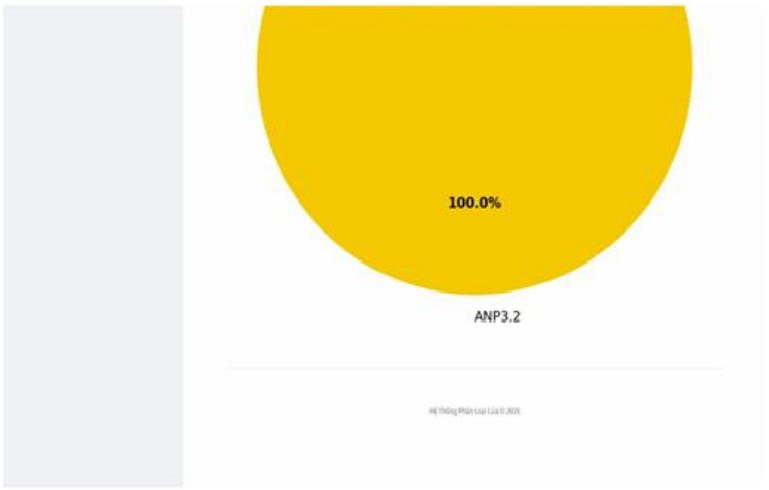


Hình 29: Giao diện website (1)

Với giao diện được xây dựng trên nền tảng Streamlit, người dùng có thể dễ dàng lựa chọn phương pháp phân loại (một giai đoạn hoặc hai giai đoạn) cùng với mô hình tương ứng để thực hiện nhận diện hình ảnh hạt giống lúa. Kết quả phân loại được hiển thị một cách trực quan, bao gồm tên loại hạt giống được nhận diện, phân loại, thời gian dự đoán, cùng với các biểu đồ thống kê như: số lượng và tỉ lệ các loại lúa giống được phát hiện, biểu đồ thể hiện sự phân bố độ tin cậy của mô hình đối với từng lớp.



Hình 30: Giao diện website (2)



Hình 31: Giao diện website (3)

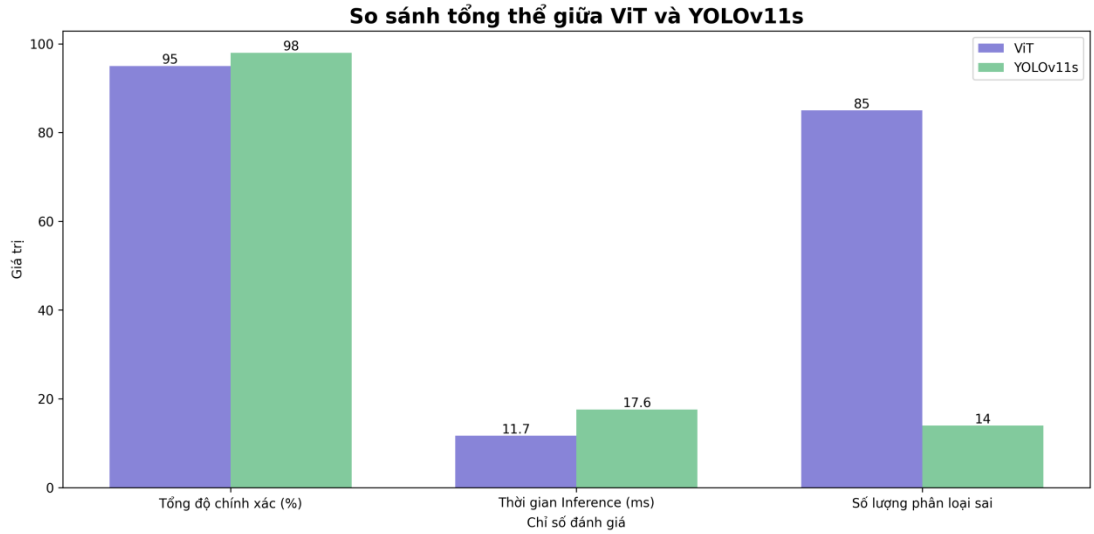
3.6 KẾT LUẬN

3.6.1 Kết quả thu nhận

Đề tài đã chứng minh tính khả thi của hệ thống phân loại hạt giống lúa dựa trên các hướng tiếp cận đề xuất.

Các mô hình phân loại riêng biệt dù có sự khác nhau ít nhiều về hiệu suất nhưng nhìn chung tất cả điều đạt con số 90% về hiệu suất, nó là một con sô đủ lớn khi tiến hành xây dựng các mô hình phân loại. Ngoài ra các mô hình tích hợp như YOLO hay Faster R-CNN cũng cho ra các kết quả rất tốt khi tích hợp quá trình phát hiện và phân loại, mặc dù điều đó sẽ gây ảnh hưởng ít nhiều về hiệu suất phân loại nhưng các mô hình trên điều cho thấy các kết quả hết sức hài lòng đặc biệt là mô hình YOLOv11s cho thấy một hiệu suất gần như là hoàn hảo với 98% độ chính xác phân loại.

Để tiến hành so sánh phương pháp nào hiệu quả hơn trong bài toán "Ứng dụng mô hình học sâu trong phân loại hạt giống lúa" tôi sẽ lựa chọn hai mô hình tiêu biểu nhất của hai phương pháp là: ViT và YOLOv11s với mục tiêu là so sánh hiệu quả khả năng phân loại và thời gian inference của từng mô hình theo cấu hình phần cứng như *Bảng 6*.



Biểu đồ 4: Biểu đồ so sánh hiệu suất của các phương pháp

Khi so sánh hai hướng tiếp cận thông qua các đại diện, có thể thấy rõ rằng hướng tiếp cận một giai đoạn vượt trội hơn hướng tiếp cận còn lại trên tất cả các chỉ số. Về độ chính xác, hướng tiếp cận một cao hơn 3% so với hướng tiếp cận hai. Ngoài ra thời gian inference cũng như khả năng phân loại của mô hình tích hợp cao hơn và ít sai sót hơn so với mô hình riêng biệt. Con số 14 so với 85 trong số lượng phân loại sai còn cho thấy được mô hình tích hợp học tốt hơn, trích xuất đặc trưng ổn định hơn giúp nâng cao độ chính xác và giảm thiểu sai sót trong quá trình phân loại.

Tuy nhiên, cũng cần nhấn mạnh rằng mô hình ViT, mặc dù thể hiện hiệu suất kém hơn trong thử nghiệm này, vẫn cho thấy tiềm năng mạnh mẽ. Cấu trúc Transformer – nền tảng của ViT – vốn đã chứng minh tính ưu việt trong xử lý ngôn ngữ tự nhiên, đặc biệt trong các mô hình tiên tiến như BERT hay GPT. Việc áp dụng Transformer vào thị giác máy tính, một lĩnh vực vốn chịu ảnh hưởng lớn từ mạng nơ-ron tích chập, mở ra hướng phát triển đầy triển vọng. Điều này cho thấy Transformer không chỉ giới hạn trong xử lý ngôn ngữ tự nhiên mà còn đang dần khẳng định vị thế trong các bài toán thị giác máy tính, hứa hẹn sẽ tiếp tục phát triển và có thể thay thế mạng nơ-ron tích chập trong tương lai.

Tóm lại đề tài "Ứng dụng mô hình học sâu trong phân loại hạt giống lúa" là đề tài mang tính thực tiễn cao và thông qua kết quả nghiên cứu cho thấy phương pháp hai giai đoạn mang lại hiệu suất tốt hơn. Đề tài không chỉ góp phần khẳng định tiềm năng ứng dụng của mô hình học sâu trong lĩnh vực nông nghiệp thông minh mà còn mở ra định hướng phát triển các hệ thống phân loại tự động, hỗ trợ quá trình chọn lọc, kiểm định giống lúa một cách hiệu quả, nhanh chóng và chính xác. Qua kết quả đạt được, đề tài hứa hẹn sẽ là

nền tảng cho các nghiên cứu mở rộng sau này, chẳng hạn như phân loại sâu bệnh trên cây trồng, dự đoán năng suất mùa vụ, hoặc tích hợp với hệ thống IoT nhằm tối ưu hóa chuỗi cung ứng nông nghiệp theo hướng tự động hóa và thông minh hóa.

3.6.2 Ưu điểm

Được kế thừa các thuật toán tiên tiến từ Ultralytics, giúp huấn luyện hiệu quả, nhanh chóng.

Các mô hình mang lại độ chính xác cao trong phân loại hạt lúa.

Sử dụng đa dạng kiến trúc mô hình giúp đánh giá bài toán khách quan và toàn diện hơn.

Phương pháp tiếp cận linh hoạt, phù hợp cho cả mục tiêu tốc độ lẫn độ chính xác.

Detectron2 hỗ trợ nhiều kiến trúc mạnh như Faster R-CNN, Mask R-CNN, RetinaNet,...

Dễ dàng mở rộng sang Segmentation hay sử dụng Oriented Bounding Box để có thể tăng độ chính xác hơn.

3.6.3 Nhược điểm

Dữ liệu còn hạn chế về số lượng mẫu khiến các framework mạnh như Detectron2 không phát huy được tối đa hiệu quả.

Dữ liệu chủ yếu là các vật thể nhỏ có thể khiến một số mô hình hạn chế về kích thước nhỏ như Faster R-CNN gặp khó khăn.

Chưa phát triển để tích hợp vào app mobile hay các thiết bị để ứng dụng vào thực tiễn.

3.6.4 Hướng phát triển tương lai

Xây dựng tiêu chuẩn đo lường hạt giống dựa trên thông số sinh trưởng và màu sắc kết hợp AI nhằm giảm tính chủ quan khi gán nhãn.

Xây dựng hệ thống đo lường 3D để phân tích và phân loại hạt giống lúa, để thay thế phương pháp ảnh 2D truyền thống.

Thu thập và bổ sung thêm dữ liệu về các loại hạt lúa, bổ sung thêm các hạt mầm bệnh có biểu hiện trên hạt để ước tính tỷ lệ và phân loại lúa nhiễm bệnh.

Kết hợp nhiều bài toán như phân loại theo giống lúa, đồng thời phân loại chúng theo kích thước (dài, ngắn, vừa) hay các phổ của màu sắc để có thể xây dựng tài liệu thống kê về hạt lúa giống đó.

Tích hợp lên ứng dụng mobile và web giúp dự đoán giống lúa linh hoạt, tiện dụng hơn.

TÀI LIỆU THAM KHẢO

[1] "Tiêu chuẩn: TCVN 13381-1:2023 - Giống cây nông nghiệp – Khảo nghiệm giá trị canh tác và giá trị sử dụng – Phần 1: Giống lúa." Accessed: Mar. 24, 2025. [Online]. Available: https://tieuchuan.vsqi.gov.vn/tieuchuan/view?sohieu=TCVN%201338 1-1:2023

[2] G. Wang *et al.*, "Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, Jul. 2018, doi: 10.1109/TMI.2018.2791721.

[3] K. Teoh, R. Ismail, S. Naziri, R. Hussin, M. Isa, and M. Basir, "Face Recognition and Identification using Deep Learning Approach," *J. Phys.: Conf. Ser.*, vol. 1755, no. 1, p. 012006, Feb. 2021, doi: 10.1088/1742-6596/1755/1/012006.

[4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *Int J Comput Vis*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[5] Q. Zhou and C. Yu, "Object Detection Made Simpler by Eliminating Heuristic NMS," *IEEE Transactions on Multimedia*, vol. 25, pp. 9254–9262, 2023, doi: 10.1109/TMM.2023.3248966.

[6] K. Kim and H. S. Lee, "Probabilistic Anchor Assignment with IoU Prediction for Object Detection," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 355–371. doi: 10.1007/978-3-030-58595-2_22.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[8] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," Oct. 23, 2024, *arXiv*: arXiv:2410.17725. doi: 10.48550/arXiv.2410.17725.

[9] A. V. Sai Abhishek and S. Kotni, "Detectron2 Object Detection & Manipulating Images using Cartoonization," *International Journal of Engineering and Technical Research*, vol. 10, Nov. 2022.

[10] "detectron2/README.md at main · facebookresearch/detectron2," GitHub. Accessed: Mar. 30, 2025. [Online]. Available:

https://github.com/facebookresearch/detectron2/blob/main/README.md