

# ỨNG DỤNG MACHINE LEARNING TRONG NHẬN DIỆN HÌNH ẢNH QUẢ DÂU TÂY

MACHINE LEARNING APPLICATION IN IDENTIFICATION STRAWBERRY PICTURE

Đỗ Bá Quang Huy<sup>1</sup>, Trần Quang Huy<sup>1</sup>,  
Đinh Văn Lực<sup>1</sup>, Trần Thủy Văn<sup>1</sup>

DOI: <https://doi.org/10.57001/huih5804.2023.058>

## TÓM TẮT

Nhận diện và xử lý hình ảnh là một phân đoạn quan trọng đang được ứng dụng nhiều ở trong các dự án như nhận diện biển số xe, nhận diện khuôn mặt,... Nhận dạng và phân loại hình ảnh dựa trên Machine learning đã phát triển nhiều ứng dụng trong những năm gần đây, chẳng hạn như phân tích hình ảnh y sinh, nhận dạng mục tiêu phương tiện, nhận dạng biểu thức và nhận dạng ký tự [9]. Trong bài báo này, thuật toán Machine learning được ứng dụng để nhận diện hình ảnh quả dâu tây, những kết quả sau đó thông qua ngôn ngữ Python được xử lý thành dữ liệu. Quá trình bắt đầu từ việc trích xuất khung hình dữ liệu gốc, qua các thuật toán hiệu chỉnh để có được hình ảnh có kích thước và tỉ lệ phù hợp. Những hình ảnh đó được đánh giá nhận diện bởi Machine learning, sau đó những hình ảnh đúng lại được nạp lại vào Machine Learning. Trong khi các kết quả chính xác cũng được xử lý thành dữ liệu bằng ngôn ngữ python để tạo cơ sở dữ liệu. Qua càng nhiều lần chạy các hình ảnh được nhận diện đúng sẽ được đưa lại vào cơ sở dữ liệu của Machine Learning để nâng cao tốc độ và tính chính xác (cao hơn, tốt hơn, thời gian ngắn hơn, hiệu quả hơn,...). Kết quả được mô phỏng trên phần mềm YOLOv7 (chỉ trong 1 lần quét) trong môi trường colab với cơ sở dữ liệu ảnh chứng minh tính hiệu quả của phương pháp đề xuất [2].

**Từ khóa:** Python, machine learning, dâu tây, nhận diện, YOLOv7.

## ABSTRACT

Image recognition and processing is an important segment that is being applied in many projects such as license plate recognition, face recognition,.... Image recognition and classification based on developed Machine learning many applications in recent years, such as biomedical image analysis, vehicle target recognition, expression recognition, and character recognition [9]. In this paper, a Machine learning algorithm is applied to recognize strawberry images, the results are then processed into data through Python. The process starts from extracting the original data frame, through the correction algorithms to get the image with the right size and ratio. Those images are evaluated by Machine learning, then the correct images are fed back into Machine Learning. While the exact results are also processed into data in python language to create the database. Through more and more runs the correctly recognized images will be fed back into the Machine Learning database to improve speed and accuracy (higher, better, shorter time, more efficient,...). The results are simulated on YOLOv7 software (just 1 scan) in colab environment with image database proving the effectiveness of the proposed method [2].

**Keywords:** Python, machine learning, strawberry, recognition, YOLOv7.

<sup>1</sup>Khoa Điện, Trường Đại học Công nghiệp Hà Nội

\*Email: dohuy9379@gmail.com

Ngày nhận bài: 22/10/2022

Ngày nhận bài sửa sau phản biện: 01/02/2023

Ngày chấp nhận đăng: 15/3/2023

## 1. GIỚI THIỆU

Machine Learning (học máy) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể [1]. Hiện nay, cùng với sự bùng nổ của chia sẻ thông tin internet và sự nâng cao trong chế tạo các máy tính có khả năng xử lý cao, machine learning đang được ứng dụng ngày càng nhiều và được cải thiện, đặc biệt có khả năng dự đoán (dự đoán giá cả vật liệu, dự toán nhu cầu sử dụng,...) và nhận diện (nhận diện chữ viết, nhận diện biển báo, nhận diện vật thể,...) [4]. Trong nông nghiệp hiện nay, Machine Learning đang được ứng dụng khá đặc trưng cho vấn đề nhận diện giống cây ngoài tự nhiên và nhận diện để phân loại quả dựa trên màu sắc. Các phương hướng hiện nay đa số nhận diện hình ảnh sẵn có dựa vào kho dữ liệu đặc tính [3]. Có những tổn thất tích lũy rất lớn do các bệnh như vậy làm giảm năng suất và làm tăng tổn thất kinh tế trong ngành nông nghiệp. Ngành nông nghiệp cần phải duy trì và phát triển từ những trở ngại như vậy để có lợi nhuận cao [5]. Trong bài báo này, chúng tôi đề xuất nhận diện riêng biệt và xác định vị trí quả dâu tây - một loại cây kinh tế cao trong ảnh dựa vào ứng dụng Machine Learning và ngôn ngữ lập trình Python. Nhằm vào đặc tính tự cải thiện của Machine Learning và độ thích hợp của ngôn ngữ lập trình python với đối tượng hình ảnh. Máy móc được sử dụng thay cho mắt người để đo lường và đánh giá [7]. Mục tiêu đối tượng là giám sát cây trồng trong phòng thí nghiệm, và cần lược bỏ việc xác định nhầm các cây cỏ, chậu không, hay cây đã chết mà khiến máy tăng công việc xử lý. Ngoài ra, phải xác định được mục tiêu để giám sát cùng với vị trí trong khung hình từ máy quay trực tiếp thời gian thực với độ chính xác cao. Nhận diện quả dâu tây và xây dựng cơ sở dữ liệu là một bước nhỏ nhưng lại không thể thiếu, đây là bước đầu và cũng là cơ sở nên cần có độ chính xác cao mà cũng không được quá chậm trễ gây ảnh hưởng các giai đoạn phía sau. Với điểm mạnh có thể tự cải thiện, Machine Learning hoàn toàn phù hợp để nhận diện và xây dựng cơ sở dữ liệu giúp hệ thống được tối ưu hóa.

## 2. TÀI LIỆU VÀ LÝ THUYẾT

### 2.1. Tập dữ liệu hình ảnh

Kho dữ liệu dùng để huấn luyện hệ thống nhận diện các quả dâu tây, và được chia thành 3 nhãn: cuống dâu tây

(peduncle strawberries), dâu tây chín (ripe strawberries), dâu tây chưa chín (unripe strawberries). Kho dữ liệu lớn lên hơn tới khoảng 1000 ảnh cho cả ba nhãn.

**2.2. Phát hiện đối tượng**

Phát hiện đối tượng là một công nghệ máy tính liên quan đến tầm nhìn máy tính và xử lý hình ảnh, liên quan đến việc phát hiện các trường hợp của các đối tượng ngữ nghĩa của một lớp nhất định (như con người, tòa nhà hoặc xe hơi) trong hình ảnh và video kỹ thuật số [1]. Các lĩnh vực được nghiên cứu kỹ lưỡng giải quyết việc phát hiện đối tượng bao gồm: phát hiện khuôn mặt và phát hiện người đi bộ. Phát hiện đối tượng đã được ứng dụng trong nhiều lĩnh vực có sử dụng máy tính, bao gồm cả truy xuất hình ảnh và giám sát video.

**2.3. Machine learning**

Machine Learning rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ [1]. Một số ứng dụng cho trên thực tế cho thấy các máy có thể "học" cách phân loại, ví dụ: thư điện tử có thể được xem là thư rác (spam) hay không và tự động xếp thư vào thư mục tương ứng. Machine Learning cũng được coi là một phần của trí tuệ nhân tạo. Thuật toán Machine Learning xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu huấn luyện, để đưa ra dự đoán hoặc quyết định mà không được lập trình rõ ràng để làm như vậy [2]. Nếu phân tích dựa trên kinh nghiệm và một số lượng đủ các ví dụ trung thực nền tảng có sẵn thì một phương pháp máy học được chỉ ra [6]. Các thuật toán học máy được sử dụng trong nhiều ứng dụng khác nhau trong đó khó hoặc không thể phát triển các thuật toán thông thường để thực hiện các nhiệm vụ cần thiết. Có nhiều nghiên cứu về tối ưu hóa toán học, đã cung cấp các phương pháp, lý thuyết và ứng dụng cho lĩnh vực Machine Learning. Với sự tiến bộ của khoa học xã hội, công nghệ trí tuệ nhân tạo cũng phát triển nhanh chóng và con người đã đạt được những tiến bộ đột phá trong nghiên cứu về học máy [8].

Tom Mitchell, giáo sư nổi tiếng của Đại học Carnegie Mellon University - CMU định nghĩa cụ thể và chuẩn mực hơn như sau: "Một chương trình máy tính được xem là học cách thực thi một lớp nhiệm vụ thông qua trải nghiệm, đối với thang đo năng lực nếu như dùng năng lực mà ta đo thấy năng lực thực thi của chương trình có tiến bộ sau khi trải qua kinh nghiệm" (máy đã học). Một trong những trọng tâm khác của học máy là đạt được tính phổ quát (generalization), tính chất của chương trình có thể làm việc tốt với dữ liệu mà nó chưa gặp bao giờ (unseen data) để dần tự có khả năng phán đoán nhất định và có thể cập nhật thời gian thực với những dữ liệu mà nó gặp được.

**2.4. Yolo V7**

Yolo (You Only Look Once) là một mô hình dùng để phát hiện, nhận dạng và phân loại đối tượng với thời gian thực. Yolo được tạo ra từ việc kết hợp giữa các convolutional-layers và connected-layers. Trong đó các convolutional-layers sẽ trích xuất ra các feature của ảnh, còn full-connected-layers sẽ dự đoán ra xác suất đó và tọa

độ của đối tượng. Nhiệm vụ phát hiện đối tượng bao gồm xác định vị trí trên hình ảnh nơi có các đối tượng nhất định, cũng như phân loại các đối tượng đó. Các phương pháp trước đây cho việc này, như R-CNN và các biến thể của nó, đã sử dụng một đường truyền để thực hiện nhiệm vụ này trong nhiều bước. Điều này có thể chạy chậm và cũng khó để tối ưu hóa, bởi vì mỗi thành phần riêng lẻ phải được huấn luyện riêng. Cũng cần phải dự đoán xác suất của lớp, PR (lớp (i) | đối tượng), và được điều hòa trên ô lưới chứa một đối tượng. Sau khi thực hiện được việc mã hóa các dự đoán, phần còn lại là để dàng được thực thi.

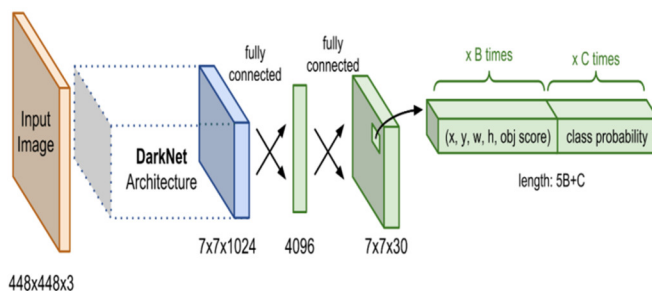
**3. XÂY DỰNG MÔ HÌNH THUẬT TOÁN NHẬN DIỆN HÌNH ẢNH**

**3.1. Huấn luyện hệ thống**

**3.1.1. Xây dựng cơ sở dữ liệu nguồn**

Tạo một hệ cơ sở dữ liệu ban đầu về vật thể muốn nhận dạng là quả dâu tây, được thực hiện gồm 4 bước:

- Bước 1: Phân tích và điều chỉnh mức độ điểm ảnh của dữ liệu đầu vào:
  - Đầu tiên, huấn luyện trước 20 lớp kết hợp đầu tiên bằng cách sử dụng bộ dữ liệu cạnh tranh lớp 1000 ImageNet, sử dụng kích thước đầu vào là 224x224.
  - Sau đó, tăng độ phân giải đầu vào lên 448x448.
  - Huấn luyện toàn bộ mạng cho khoảng 135 epochs sử dụng kích thước nhóm là 64, động lượng 0,9 và phân rã là 0,0005.
  - Đối với các vòng lặp đầu tiên, tỷ lệ học được tăng chậm từ 0,001 lên 0,01. Huấn luyện cho khoảng 75 vòng lặp và sau đó bắt đầu giảm.
  - Sử dụng phép gia tăng dữ liệu với tỷ lệ và chuyển đổi ngẫu nhiên và điều chỉnh ngẫu nhiên độ phơi sáng và bão hòa [10].



Hình 1. Mô hình biểu diễn phân tích hình ảnh

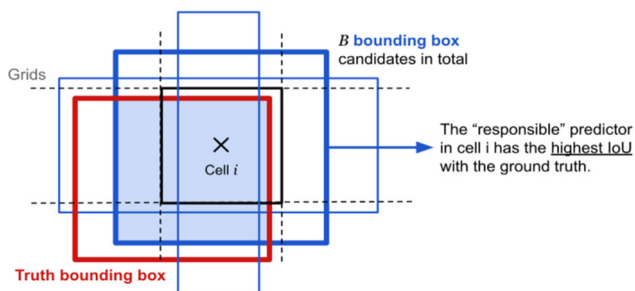
Bước 2: Quét các điểm ảnh để xác định vùng các điểm nổi bật:

Tại đây hệ thống sẽ quét lần lượt điểm ảnh sau đó phân vùng cơ bản dữ liệu đầu vào. Sau đó hệ thống so sánh mật độ điểm ảnh có độ trùng khớp rồi xác định vùng có điểm ảnh nổi bật.

Bước 3: Định vị vật thể:

Hệ thống sẽ dựa vào những hình ảnh gốc này tiến hành định vị các vật thể trong ảnh sau đó đối chiếu tìm các điểm

chung của một số vật thể nổi trội (ở đây là quả dâu tây). Trong bước này, cần dữ liệu đầu vào là tập hợp các ảnh để thực hiện việc huấn luyện.



Hình 2. Phân tích khung định vị vật thể

Bước 4: Chọn các thuộc tính mẫu từ các điểm ảnh trong các vùng đã được định vị:

Ở bước này hệ thống sẽ lưu trữ lại thuộc tính của các đặc trưng được chọn để làm tham số quy chiếu, từ đó đánh giá độ tương đồng dùng cho định vị và nhận diện vật thể. Với số lượng ảnh của cơ sở dữ liệu càng lớn và càng nhiều góc độ thì lượng tham số quy chiếu được xác định sẽ càng nhiều và càng chi tiết.

**3.1.2. Xử lý đầu vào và xác định vị trí**

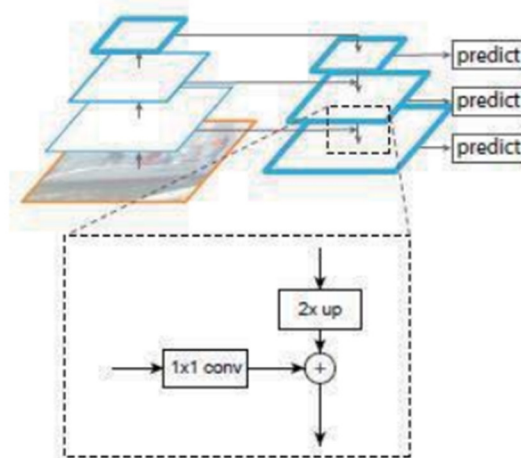
Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành 5x5 ô thường thì sẽ là 3x3, 7x7, 9x9,... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng. Đầu ra mô hình là một ma trận 3 chiều có kích thước 5x5x(5xN+M) với số lượng tham số mỗi ô là (5xN+M) với N và M lần lượt là số lượng khung và lớp mà mỗi ô biến độc lập. Khung giới hạn mỗi biến độc lập gồm 5 thành phần (x, y, w, h, prediction). Với (x, y) là tọa độ tâm của khung giới hạn; w, h lần lượt là chiều rộng và chiều cao của khung giới hạn; prediction được định nghĩa Pr(Object)xIOU(pred,truth). Tâm của khung giới hạn nằm ở ô nào thì ô đó sẽ chứa đối tượng, cho dù đối tượng có thể ở các ô khác thì cũng sẽ trả về là 0. Vì vậy, việc mà 1 ô chứa 2 hay nhiều tâm của khung giới hạn hay đối tượng thì sẽ không thể detect được, khi đó, cần phải tăng số lượng ô chia trong 1 ảnh lên.

Trình tự của các lớp giảm 1x1 và các lớp tích chập 3x3 được lấy cảm hứng từ mô hình GoogLeNet (Inception), lớp cuối cùng sử dụng chức năng kích hoạt tuyến tính. Tất cả các lớp khác sử dụng relu rò rỉ (φ(x) = x, nếu x > 0; 0,1x khác).

**3.1.3. Mạng lưới nhận diện**

Sau 32 lần lấy mẫu, ta lấy mẫu ngược với kích thước bước từ 2 để gấp đôi kích thước của bảng đặc điểm kết quả, trở thành 16 lần mẫu lấy xuống. Tiếp tục lấy mẫu ngược với bước nhảy là 2 ta về kích thước 8 lần lấy mẫu. Từ đó ta có thể sử dụng chiều sâu để nhận diện. Việc phát hiện các mục tiêu ở các quy mô khác nhau là một thách thức, đặc biệt là với các mục tiêu nhỏ. Kim tự tháp mạng đặc tính

(FPN) là một tính năng trích xuất được thiết kế để cải thiện độ chính xác và tốc độ [8]. Nó thay thế bộ trích xuất đặc điểm trong bộ dò và tạo ra các kim tự tháp đồ thị đối tượng chất lượng cao hơn.



Hình 3. Cấu trúc mạng lưới kết nối.

Cấu trúc mạng lưới kết nối gồm 2 tuyến (tuyến trên, tuyến dưới) và kết nối ngang. Quá trình dưới và trên thực chất là quá trình chuyển tiếp của mạng. Trích xuất các đặc điểm đầu ra của lớp cuối cùng của mỗi giai đoạn để tạo thành kim tự tháp đặc tính. Quá trình từ trên dưới được thực hiện bằng cách lấy mẫu lên. Trong khi kết nối ngang là để hợp nhất các kết quả của lấy mẫu lên và bảng đặc điểm có cùng kích thước được tạo từ dưới lên.

Chức năng chính của nhân chập 1 \* 1 là giảm số lượng nhân chập mà không làm thay đổi kích thước của bảng đặc điểm. Điều này cho phép có thêm thông tin về đặc tính đối tượng có giá trị từ lớp lấy mẫu lên và các đặc điểm từ bảng đặc điểm trước đó. Các đặc tính hàng đầu hợp nhất với đặc tính lấy mẫu lên và lấy mẫu xuống, và mỗi lớp được dự đoán độc lập.

**3.1.4. Hàm tổn thất (loss function)**

Phần 1:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \tag{1}$$

Phương trình (1) tính toán tổn thất liên quan đến vị trí khung giới hạn biến độc lập (x,y). Ở đây λ<sub>coord</sub> là một hằng số. Hàm tính toán một tổng trên mỗi bộ khung giới hạn (j = 0... B) của mỗi ô lưới (i = 0 ... S^2). Trong đó,  $\mathbb{1}_{ij}^{obj}$  được định nghĩa là 1, nếu một đối tượng có mặt trong ô lưới i và khung giới hạn biến độc lập j<sup>th</sup> là "chịu trách nhiệm" cho biến độc lập đó; ngược lại,  $\mathbb{1}_{ij}^{obj} = 0$ .

Yolo dự đoán nhiều khung giới hạn trên mỗi ô lưới. Vào thời điểm khởi tạo, chỉ có một khung giới hạn biến độc lập phải chịu trách nhiệm cho từng đối tượng. Cho 1 biến độc lập "chịu trách nhiệm" cho việc dự đoán một đối tượng dựa trên dự đoán nào có IOU hiện tại cao nhất với độ chính xác cơ sở. Các thuật ngữ khác trong phương trình trở nên dễ hiểu: (x, y) là vị trí khung giới hạn dự đoán và (x̂, ŷ) là vị trí thực tế từ dữ liệu huấn luyện.

Phần 2:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \quad (2)$$

Phương trình (2) là tổn thất liên quan đến chiều rộng / chiều cao khung dự đoán. Tương tự như phương trình (1), ngoại trừ căn bậc hai. Số liệu lỗi đã phản ánh rằng độ lệch nhỏ trong các khung lớn hơn và có vấn đề ít hơn các khung nhỏ, trong đó, căn bậc hai của chiều rộng và chiều cao của khung bị giới hạn thay vì chiều rộng và chiều cao trực tiếp.

Phần 3:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (3)$$

Tính toán tổn thất liên quan đến điểm tin cậy cho mỗi khung giới hạn biến độc lập thể hiện trong (3). C là độ tin cậy và  $\hat{C}$  là sự tương giao qua kết hợp giữa khung giới hạn biến độc lập với độ chính xác cơ sở. Giá trị  $\mathbb{1}^{obj}$  bằng 1 nếu có đối tượng trong ô, và bằng 0 nếu không. Riêng đối với  $\mathbb{1}^{noobj}$  thì ngược lại.

Tham số  $\lambda_{noobj}$  xuất hiện ở đây và trong phần đầu tiên được dùng cho các phần có trọng số khác nhau của các hàm tổn thất. Điều này là cần thiết để tăng sự ổn định mô hình. Bù lỗi cao nhất có thể cho phép cho các dự đoán tọa độ ( $\lambda_{coord}$ ) và thấp nhất cho các dự đoán tin cậy khi không có đối tượng ( $\lambda_{noobj}$ ). Để tránh dữ liệu nhận diện sai lệch quá lớn so với đầu vào và thực nghiệm thực tế, cần thiết lập  $\lambda_{coord} = 5$  và  $\lambda_{noobj} = 0,5$ .

Phần 4:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (4)$$

Để phân loại, (4) khá tương tự như một lỗi tổng bình phương bình thường, ngoại trừ số hạng  $\mathbb{1}^{obj}$ . Số hạng này được sử dụng, vì vậy không bù lỗi phân loại khi không có đối tượng nào trên ô (do đó xác suất lớp có điều kiện).

$\mathbb{1}_i^{obj}$ : Một hàm chỉ báo về việc liệu ô I có chứa một đối tượng hay không.

$\mathbb{1}_{ij}^{obj}$ : Nó cho biết hộp giới hạn thứ J của ô I "chịu trách nhiệm" cho dự đoán đối tượng.

$C_{ij}$ : Điểm tin cậy của CELL I, PR (chứa một đối tượng) \* iou (pre, sự thật).

$\hat{C}_{ij}$ : Điểm tin cậy dự đoán.

C: Tập hợp tất cả các lớp.

$p_i(c)$ : Xác suất có điều kiện của liệu tế bào I có chứa một đối tượng của lớp CEC hay không.

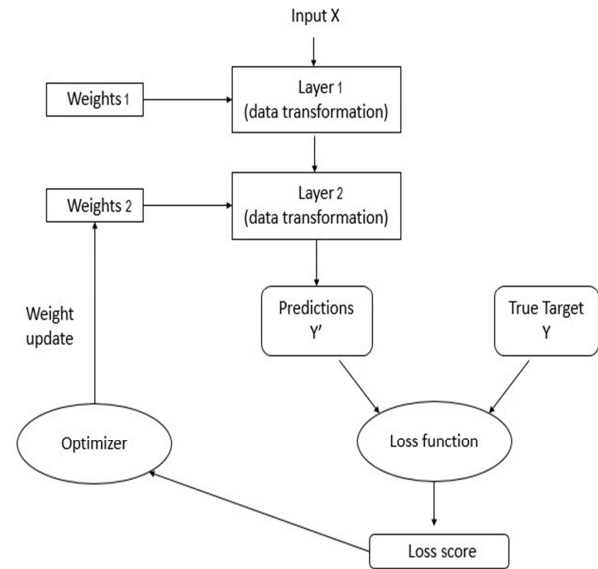
$\hat{p}_i(c)$ : Xác suất lớp có điều kiện dự đoán.

### 3.2. Lưu đồ thuật toán nhận diện hình ảnh

- Layer 1 (data transformation): Lớp chuyển đổi dữ liệu khởi tạo.

- Layer 2 (data transformation): Lớp chuyển đổi dữ liệu sau tinh chỉnh trọng số.

- Prediction Y': Kết quả dự đoán thu được.
- True Target Y: Mô hình mẫu từ kết quả huấn luyện.
- Weights 1: Trọng số khởi tạo ngẫu nhiên.
- Weights 2: Trọng số đã tinh chỉnh.
- Optimizer: Chương trình tối ưu hóa trọng số.
- Loss function: Hàm tổn thất.
- Loss score: Kết quả tổn thất.

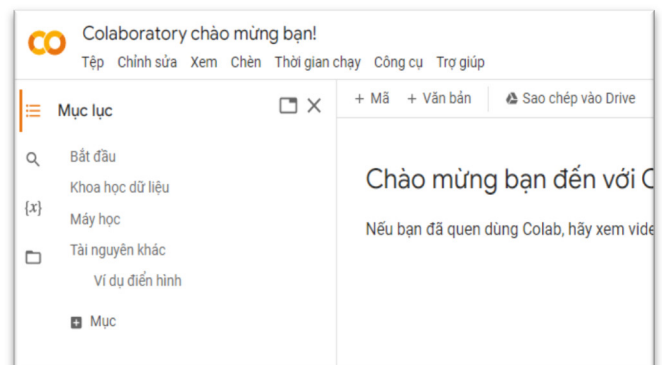


Hình 4. Lưu đồ thuật toán nhận diện hình ảnh

## 4. KẾT QUẢ THỰC NGHIỆM VÀ PHÂN TÍCH

### 4.1. Môi trường thử nghiệm

Colab google: laboratory, hay "colab" là sản phẩm của Google Research. Colab cho phép bất kỳ ai viết và thực hiện mã Python tùy ý thông qua trình duyệt và đặc biệt phù hợp với học máy, phân tích dữ liệu và giáo dục. Về mặt kỹ thuật, Colab là một dịch vụ Notebook Jupyter được lưu trữ mà không cần thiết lập để sử dụng, đồng thời cung cấp quyền truy cập miễn phí cho các tài nguyên điện toán bao gồm GPU.

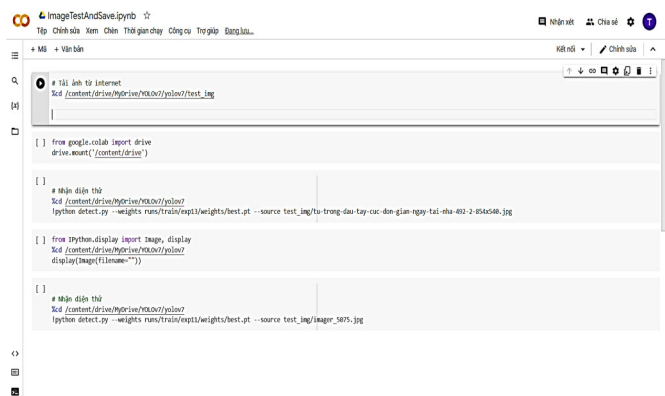


Hình 5. Màn hình chào của COLAB

### 4.2. Phần mềm YOLOv7

Hình ảnh về phần mềm như hình 6.





Hình 6. Màn hình khởi động của phần mềm YOLO

### 4.3. Đối tượng

Một số hình ảnh thu thập được như trong hình 7, 8.



Hình 7. Hình chùm dâu tây



Hình 8. Hình ảnh về dâu tây tại vườn

### 4.4. Các bước tiến hành

Bước 1: Công cụ tự động download 1 ảnh chứa vật thể và lưu vào đường dẫn: drive/MyDrive/YOLOv7/yolov7/test\_img



Hình 9. Màn hình thông báo lấy thành công ảnh

Bước 2: Lấy ảnh chứa vật thể cần nhận diện ở file test\_img để đưa vào xử lý nhận diện khoảng vùng đối tượng và tính toán xác suất. Kết quả được lưu vào đường dẫn runs/detect. Cùng với đó là đưa ra thời gian xử lý bức ảnh.



Hình 10. Màn thông báo khoảng vùng đối tượng và tính toán tỉ lệ tin cậy thành công

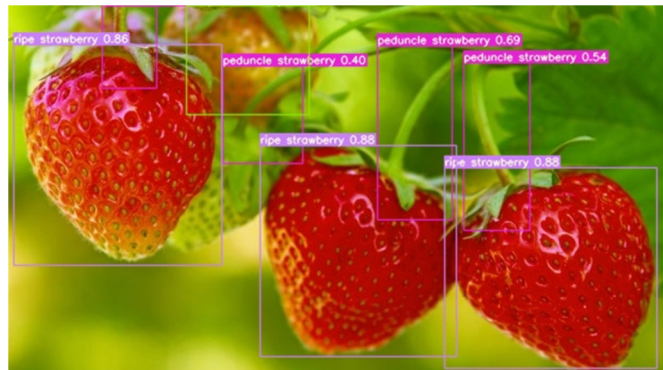
Bước 3: Hiển thị kết quả lên màn hình từ file có đường dẫn runs/detect/exp12



Hình 11. Thông báo hiển thị kết quả thành công

### 4.5. Kết quả

Xác định vị trí tốt, kể cả với thời gian thực, tỉ lệ phát hiện đúng cao như thể hiện trong các hình 12, 13 và bảng 1.



Hình 12. Kết quả cho 3 chùm dâu tây cạnh nhau

Mô hình nhận diện hình ảnh với độ chính xác lên tới 88% với những vật thể rõ nét. Với những hình ảnh nhiễu

vật thể mô hình vẫn cho khả năng nhận diện và độ chính xác khá cao lên đến 75%.



Hình 13. Kết quả cho ra với nhận diện chùm dâu tây

Bảng 1. Tỷ lệ % của vật thể được nhận diện

TT	Đặc điểm hình ảnh	Tỷ lệ nhận dạng quả xanh	Tỷ lệ nhận dạng quả chín	Tỷ lệ nhận dạng cuống
1	Dạng chùm	45% - 60%	40% - 75%	45% - 64%
2	Dạng đơn	60% - 80%	70% - 88%	40% - 69%
3	Tại vườn	48% - 62%	40% - 70%	42% - 60%

### 5. KẾT LUẬN

Bài báo đề xuất được phương pháp ứng dụng thuật toán Machine Learning để nhận dạng hình ảnh quả dâu tây. Bộ dữ liệu được xử lý bằng việc đưa hình ảnh thành dạng ma trận bằng ngôn ngữ Python. Thông qua Machine Learning và thuật toán tính toán điều chỉnh để đưa ra độ tin cậy đã giúp nhận diện được với hình ảnh theo thời gian thực, nhận diện nhanh và có độ chính xác cao đối với đối tượng được thêm vào giúp cho tương thích với hình ảnh giám sát trong thực tế để có thể giám sát đối tượng với tần suất cao và tự động trích xuất ảnh thêm vào kho dữ liệu. Bên cạnh đó có những vấn đề cần giải quyết thêm về độ chính vì vẫn chưa nhận diện được chi tiết vật thể khi có những vật thể tương tự trong khung hình. Để khắc phục vấn đề này trong tương lai, nhóm nghiên cứu sẽ gia tăng cơ sở dữ liệu và thêm cơ sở dữ liệu của các loại quả tương tự để khắc phục vấn đề này.

[4]. Jiheng Liu, Zwmin Zhou, Xinwu Zeng, 2019. *Research on Multi-Static Active Sonar Target Recognition Based on Machine Learning*. IEEE.

[5]. N Gobalakrishnan, K Pradeep, C J Raman, L Javid Ali, M P Gopinath, 2020. *A Systematic Review on Image Processing and Machine Learning Techniques for Detecting Plant Diseases*. India: IEEE.

[6]. G. Becker, 2007. *Combining Rule-based and Machine Learning Approaches for Shape*. IEEE.

[7]. Pengfei Liu, Weifeng Zhang, Jingfeng Qiu, Qiaoyi Hu, Kejing He, 2019. *Reagent Color Recognition Model for Android Platform Based on OPENCV and Machine Learning*. IEEE.

[8]. Y. Zhang, 2020 *Research on Artificial Intelligence Machine Learning Character Recognitin Based on Online Machine Learning Method*. IEEE.

[9]. Beibei Zhu, Lei Yang, Xiaoyu Wu, Tianchu Guo, 2015. *Automatic Recognition of Books Based on Machine Learning*. IEEE.

[10]. Harishwaran HariHaran, Andreas Koschan, Bisma Abidi, Andrei Gribok, Mongi Abidi, 2006. *Fusion of Visible and Infrared Images Using Empirical Mode Decomposition to Improve Face Recognition*. IEEE.

### AUTHORS INFORMATION

**Do Ba Quang Huy, Tran Quang Huy, Dinh Van Luc, Tran Thuy Van**  
Faculty of Electrical Engineering, Hanoi University of Industry

### TÀI LIỆU THAM KHẢO

[1]. Panos Louridas, Christof Ebert, 2016. *Machine Learning*. IEEE Software.

[2]. Htet Aung, Alexander V. Bobkov, Nyan Lin Tun, 2021. *Face Detection in Real Time Live Video Using Yolo Algorithm Based on Vgg16 Convolutional Neural Network*. IEEE.

[3]. Stamatia Dasiopoulou, Vasileios Mezaris, Ioannis Kompatsiaris, Vasileios-Kyriakos Papastathis, 2005. *Knowledge-Assisted Semantic Video Object Detection*. IEEE.