**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

**KHOA/VIỆN: CƠ ĐIỆN CÔNG TRÌNH**

========&&&=======



**BÁO CÁO ĐỒ ÁN**

**MÔN: KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: XÂY DỰNG MÔ HÌNH VÀ ỨNG DỤNG DEMO**

**ĐỂ NHẬN DIỆN SỮA QUA HÌNH ẢNH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **:ThS. Mai Hà An** |
| **Sinh viên thực hiện** | **:Nguyễn Thị Thu Hoài** |
|  | **Phan Thu Trang** |
|  | **Nguyễn Thị Vọng** |
| **Lớp** | **:K66-HTTT** |

**Hà Nội, 2025**

**MỤC LỤC**

[CHƯƠNG 1 BÀI TOÁN NHẬN DIỆN LOẠI SỮA 1](#_Toc194239077)

[1.1. Tổng quan về nhận dạng 1](#_Toc194239078)

[1.1.1. Tính chất và sự cần thiết của bài toán nhận dạng 1](#_Toc194239079)

[1.1.2. Bản chất của quá trình nhận dạng 1](#_Toc194239080)

[1.2. Bài toán nhận diện loại sữa 2](#_Toc194239081)

[1.1.3. Khái niệm 2](#_Toc194239082)

[1.1.4. Nguyên tắc hoạt động cơ bản 2](#_Toc194239083)

[CHƯƠNG 2 NHẬN DIỆN LOẠI SỮA MÔ HÌNH HUẤN LUYỆN YOLOV5 3](#_Toc194239084)

[2.1. Mô hình huấn luyện yolov5 3](#_Toc194239085)

[2.1.1. Giới thiệu về mô hình huấn luyện yolov5 3](#_Toc194239086)

[2.1.2. Quy trình hoạt động mô hình huấn luyện yolov5 3](#_Toc194239087)

[2.1.3. Ưu và nhược điểm của mô hình huấn luyện yolov5 3](#_Toc194239088)

[2.2. Xử lý hình ảnh để nhận dạng 4](#_Toc194239089)

[2.2.1. Khái niệm về xử lý ảnh 4](#_Toc194239090)

[2.2.2. Phương pháp xử lý ảnh 4](#_Toc194239091)

[2.3. Một số thư viện phổ biến cho việc huấn luyện 5](#_Toc194239092)

[2.3.1. Thư viện cho học sâu 5](#_Toc194239093)

[2.3.2. Thư viện xử lý dữ liệu và ảnh 5](#_Toc194239094)

[2.3.3. Thư viện xử lý tệp cấu hình 6](#_Toc194239095)

[2.3.4. Thư viện vẽ đồ thị 6](#_Toc194239096)

[2.4. Biểu đồ theo dõi hiệu suất của mô hình YOLOv5 6](#_Toc194239097)

[CHƯƠNG 3 QUY TRÌNH THỰC HIỆN NHẬN DIỆN LOẠI SỮA 8](#_Toc194239098)

[3.1. Chọn đối tượng nhận diện 8](#_Toc194239099)

[3.2. Tiền xử lý dữ liệu 8](#_Toc194239100)

[3.3. Xác định vùng chứa hộp sữa và trích chọn đặc trưng từ ảnh bằng cách gán nhãn với labeling. 8](#_Toc194239101)

[3.4. Xây dựng model với mô hình huấn luyện yolov5 để phân biệt các loại sữa. 8](#_Toc194239102)

[3.5. Xây dựng ứng dụng nhận diện bằng file best.pt 12](#_Toc194239103)

[CHƯƠNG 4 KẾT LUẬN 14](#_Toc194239104)

[4.1. Khó khăn và hướng giải quyết 14](#_Toc194239105)

[4.2. Kết quả đã đạt được 14](#_Toc194239106)

[4.3. Vấn đề tồn dồn và phương hướng phát triển khai, nâng cấp 14](#_Toc194239107)

# CHƯƠNG 1 **BÀI TOÁN NHẬN DIỆN LOẠI SỮA**

## Tổng quan về nhận dạng

### Tính chất và sự cần thiết của bài toán nhận dạng

Bài toán nhận dạng là một lĩnh vực quan trọng trong trí tuệ nhân tạo và khai phá dữ liệu, giúp máy tính có khả năng phân loại và xác định các đối tượng dựa trên những đặc trưng nhất định. Một số tính chất quan trọng của bài toán nhận dạng bao gồm tính tự động, tính chính xác và tính mở rộng. Cụ thể, tính tự động cho phép hệ thống hoạt động mà không cần sự can thiệp trực tiếp của con người, giúp tăng tốc độ xử lý và giảm thiểu sai sót. Tính chính xác đảm bảo rằng kết quả nhận dạng có độ tin cậy cao, đặc biệt khi mô hình được huấn luyện với lượng dữ liệu đủ lớn. Ngoài ra, tính mở rộng giúp hệ thống có thể học từ dữ liệu mới và áp dụng vào nhiều trường hợp khác nhau, từ nhận diện hình ảnh, chữ viết tay đến giọng nói và sản phẩm.

Bài toán nhận dạng ngày càng trở nên cần thiết trong nhiều lĩnh vực thực tế. Trong thương mại, hệ thống nhận dạng sản phẩm giúp tự động phân loại hàng hóa, hỗ trợ thanh toán tại siêu thị và quản lý kho hàng hiệu quả hơn. Trong công nghiệp, công nghệ nhận dạng được sử dụng để kiểm tra chất lượng sản phẩm, phát hiện lỗi trong dây chuyền sản xuất. Ngoài ra, trong thị giác máy tính, các mô hình nhận dạng hình ảnh đóng vai trò quan trọng trong giám sát an ninh, y tế (chẩn đoán bệnh từ ảnh X-quang), và giao thông (nhận diện biển số xe). Sự phát triển của trí tuệ nhân tạo và khai phá dữ liệu ngày càng thúc đẩy ứng dụng của bài toán nhận dạng, góp phần nâng cao hiệu suất làm việc, giảm chi phí nhân lực và cải thiện trải nghiệm người dùng trong nhiều lĩnh vực khác nhau.

### Bản chất của quá trình nhận dạng

Quá trình nhận dạng có thể được chia thành các bước cơ bản sau:

* Bước 1: Thu thập dữ liệu: Dữ liệu có thể là hình ảnh, video hoặc tín hiệu cảm biến.
* Bước 2: Tiền xử lý dữ liệu: Các bước như lọc nhiễu, cân bằng sáng, chuyển đổi ảnh về cùng kích thước, chuẩn hóa màu sắc,... giúp cải thiện chất lượng đầu vào cho mô hình.
* Bước 3: Trích xuất đặc trưng: Các đặc trưng quan trọng như màu sắc, hình dạng, nhãn hiệu sẽ làm đầu vào cho mô hình phân loại.
* Bước 4: Huấn luyện mô hình: Sử dụng các thuật toán như mạng nơ-ron nhân tạo (CNN), YOLO, hoặc SVM để xây dựng mô hình phân loại.
* Bước 5: Dự đoán và đánh giá: Kiểm tra hiệu suất mô hình trên tập dữ liệu kiểm tra và tối ưu hóa để đạt độ chính xác cao nhất.

## Bài toán nhận diện loại sữa

### Khái niệm

Bài toán nhận diện loại sữa là một ứng dụng trong lĩnh vực thị giác máy tính nhằm phân loại các loại sữa khác nhau dựa trên hình ảnh của bao bì sản phẩm. Mục tiêu chính là xây dựng một hệ thống có thể tự động xác định và phân loại sữa dựa trên hình dáng, màu sắc, nhãn hiệu, logo hoặc mã vạch.

Hệ thống này có thể ứng dụng trong:

* Hệ thống quản lý siêu thị: Tự động nhận diện sản phẩm để hỗ trợ thanh toán hoặc kiểm kho.
* Kiểm tra hàng hóa trong nhà máy: Giúp phân biệt các loại sữa trong dây chuyền sản xuất.
* Ứng dụng cho người tiêu dùng: Hỗ trợ quét mã sản phẩm để tìm hiểu thông tin sữa.

### Nguyên tắc hoạt động cơ bản

Hệ thống nhận diện sữa hoạt động theo các bước sau:

* Bước 1: Thu thập dữ liệu hình ảnh: Chụp ảnh các loại sữa từ nhiều góc độ, ánh sáng và điều kiện khác nhau.
* Bước 2: Tiền xử lý ảnh: Điều chỉnh độ sáng, tăng độ nét, loại bỏ nhiễu để chuẩn bị cho quá trình nhận dạng.
* Bước 3: Trích xuất đặc trưng: Các đặc trưng quan trọng như màu sắc, kích thước, nhãn hiệu, mã vạch sẽ được xác định để làm dữ liệu đầu vào cho mô hình phân loại.
* Bước 4: Huấn luyện mô hình: Sử dụng mô hình YOLOV5 là một mô hình nhận diện đối tượng mạnh mẽ dựa trên mạng nơ-ron tích chập (CNN), giúp phát hiện và phân loại các loại sữa với tốc độ và độ chính xác cao.
* Bước 5: Dự đoán và đánh giá: Kiểm tra độ chính xác của mô hình, điều chỉnh tham số và bổ sung dữ liệu để tăng hiệu suất nhận diện.

# CHƯƠNG 2 NHẬN DIỆN LOẠI SỮA MÔ HÌNH HUẤN LUYỆN YOLOV5

## Mô hình huấn luyện yolov5

### Giới thiệu về mô hình huấn luyện yolov5

YOLOv5 (You Only Look Once version 5) là một trong những mô hình phát hiện đối tượng tiên tiến, do Ultralytics phát triển. Đây là phiên bản cải tiến của YOLOv4, với tốc độ nhanh hơn, chính xác hơn và dễ triển khai hơn. YOLOv5 đặc biệt mạnh trong các ứng dụng nhận diện đối tượng theo thời gian thực như giám sát giao thông, kiểm tra chất lượng sản phẩm, robot tự động,…

### Quy trình hoạt động mô hình huấn luyện yolov5

YOLOv5 hoạt động theo nguyên lý "You Only Look Once", tức là mô hình chỉ cần một lần quét toàn bộ ảnh để phát hiện tất cả đối tượng trong đó. Quá trình hoạt động gồm các bước sau:

* *Tiền xử lý ảnh đầu vào*
* Ảnh được chia tỷ lệ về kích thước cố định (ví dụ: 640x640 pixel).
* Chuẩn hóa dữ liệu để phù hợp với mô hình.
* *Dự đoán vị trí và phân loại đối tượng*
* Ảnh được đưa qua mạng CNN để trích xuất đặc trưng.
* Mô hình chia ảnh thành mạng lưới (grid cells) và dự đoán bounding box (hộp giới hạn) cùng với nhãn lớp của đối tượng.
* Dự đoán được thực hiện trên nhiều kích thước hộp khác nhau để nhận diện các đối tượng có kích thước khác nhau.
* *Hậu xử lý kết quả*
* Non-Maximum Suppression (NMS): Loại bỏ các hộp giới hạn trùng lặp.
* Lọc theo độ tin cậy (confidence score): Chỉ giữ lại các dự đoán có độ tin cậy cao.
* Xuất kết quả gồm vị trí hộp giới hạn, nhãn lớp và độ tin cậy.

### Ưu và nhược điểm của mô hình huấn luyện yolov5

* *Ưu điểm*
* Tốc độ cao: YOLOv5 có thể xử lý hàng chục đến hàng trăm FPS (frames per second), giúp phát hiện đối tượng theo thời gian thực.
* Độ chính xác tốt: Cải thiện so với YOLOv4 nhờ tối ưu kiến trúc backbone, head và loss function.
* Dễ sử dụng và triển khai: YOLOv5 viết bằng PyTorch, hỗ trợ xuất sang ONNX, TorchScript để tích hợp vào ứng dụng.
* Tùy chỉnh linh hoạt: Hỗ trợ các mô hình từ nhỏ (YOLOv5s) đến lớn (YOLOv5x), phù hợp cho nhiều ứng dụng.
* Hiệu quả trên GPU và CPU: Dễ triển khai trên các thiết bị từ máy tính cá nhân đến hệ thống AI edge như NVIDIA Jetson.
* *Nhược điểm*
* Không phải là phiên bản chính thức của YOLO: YOLOv5 không do tác giả gốc Joseph Redmon phát triển, mà là bản mở rộng của Ultralytics.
* Cần nhiều dữ liệu huấn luyện: Để đạt độ chính xác cao, mô hình cần dữ liệu lớn và đa dạng.
* Khả năng nhận diện vật thể nhỏ hạn chế: Do cơ chế phân vùng ảnh, YOLO có thể gặp khó khăn trong việc phát hiện các vật thể rất nhỏ.
* Mô hình lớn có thể yêu cầu phần cứng mạnh: YOLOv5x yêu cầu GPU mạnh để chạy nhanh.

## Xử lý hình ảnh để nhận dạng

### Khái niệm về xử lý ảnh

Xử lý ảnh là một lĩnh vực trong trí tuệ nhân tạo và thị giác máy tính, tập trung vào việc cải thiện, phân tích và trích xuất thông tin từ hình ảnh. Bao gồm các kỹ thuật biến đổi hình ảnh để phục vụ cho các mục đích khác nhau như nhận diện vật thể, phân loại và nâng cao chất lượng ảnh.

### Phương pháp xử lý ảnh

* Tiền xử lý ảnh: Chuẩn hóa dữ liệu, thay đổi kích thước, chuyển đổi định dạng.
* Lọc ảnh: Làm mịn, giảm nhiễu hoặc làm sắc nét ảnh bằng các bộ lọc như Gaussian, Median.
* Phát hiện cạnh: Sử dụng các phương pháp như Sobel, Canny để xác định ranh giới vật thể.
* Tăng cường ảnh: Xoay, lật, thay đổi độ sáng, độ tương phản để làm phong phú tập dữ liệu.
* Phân đoạn ảnh: Chia ảnh thành các vùng khác nhau để phân tích chi tiết hơn.

## Một số thư viện phổ biến cho việc huấn luyện

### Thư viện cho học sâu

* *torch*
* Là một thư viện mã nguồn mở được phát triển bởi Facebook AI Research (FAIR), hỗ trợ học sâu và tính toán tensor hiệu suất cao.
* Sử dụng nền tảng autograd giúp dễ dàng xây dựng và huấn luyện mô hình mạng nơ-ron nhân tạo (ANN).
* Hỗ trợ GPU thông qua CUDA giúp tăng tốc quá trình huấn luyện.
* Có nhiều module quan trọng như torch.nn (xây dựng mô hình), torch.optim (tối ưu hóa), torch.utils.data (quản lý dữ liệu).
* *torchvision*
* Là một thư viện bổ sung cho torch, chuyên về xử lý dữ liệu hình ảnh.
* Cung cấp các bộ dữ liệu phổ biến như CIFAR-10, ImageNet, MNIST để tiện lợi trong huấn luyện mô hình.
* Hỗ trợ các phép biến đổi ảnh (transforms), giúp tiền xử lý dữ liệu dễ dàng.
* Cung cấp các mô hình đã được huấn luyện sẵn như ResNet, VGG, MobileNet.

### Thư viện xử lý dữ liệu và ảnh

* *numpy*
* Là thư viện mạnh mẽ hỗ trợ tính toán khoa học và xử lý mảng dữ liệu nhiều chiều (ndarray).
* Cung cấp nhiều phép toán đại số tuyến tính, thống kê, nội suy, FFT (biến đổi Fourier nhanh).
* Hoạt động nhanh hơn so với danh sách thông thường của Python do được tối ưu hóa bằng C.
* Được sử dụng rộng rãi trong học sâu, thị giác máy tính, xử lý tín hiệu.
* *opencv-python-headless*
* Là phiên bản nhẹ của OpenCV, phù hợp với môi trường máy chủ không có giao diện đồ họa.
* Cung cấp các công cụ xử lý ảnh mạnh mẽ như phát hiện cạnh (Canny), lọc ảnh, nhận diện khuôn mặt, theo dõi đối tượng.
* Hỗ trợ làm việc với video, camera, trích xuất đặc trưng từ ảnh.
* Kết hợp tốt với torchvision để tiền xử lý dữ liệu trong thị giác máy tính.

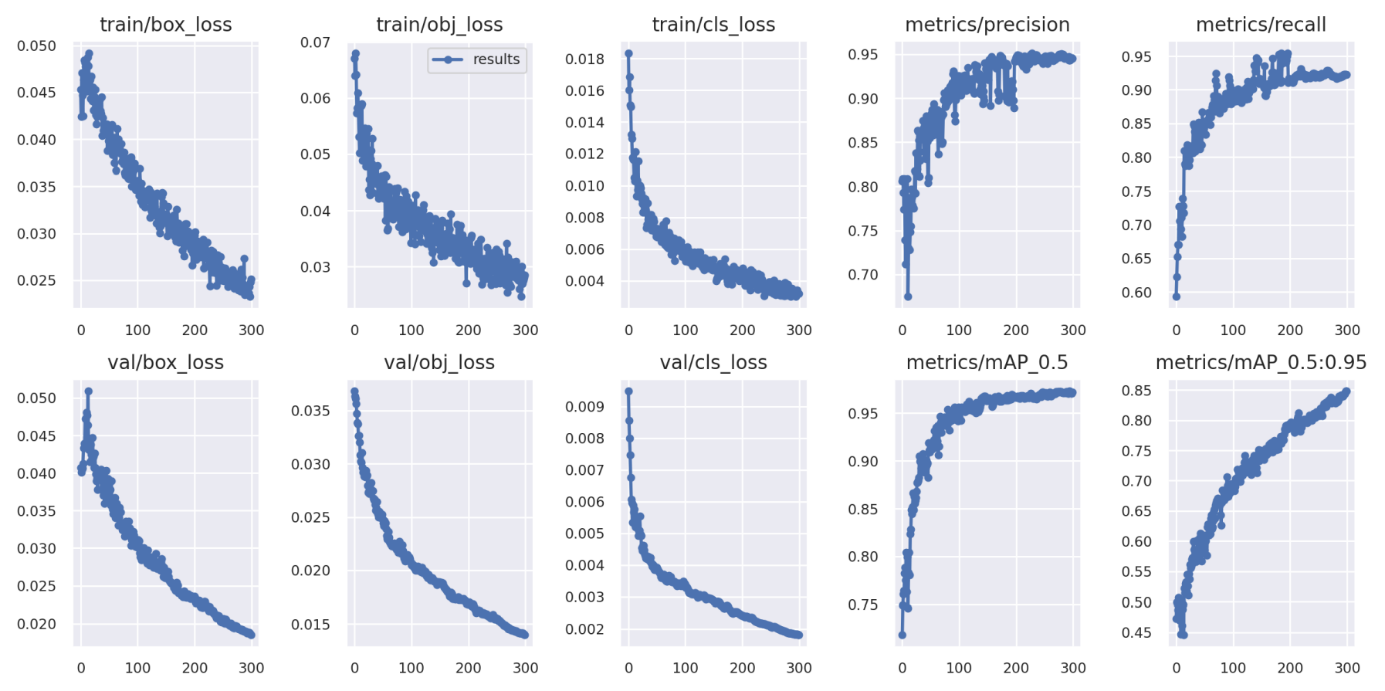
### Thư viện xử lý tệp cấu hình

* *pyyaml*
* Hỗ trợ đọc và ghi tệp cấu hình YAML, định dạng thường được sử dụng trong các ứng dụng AI/ML.
* Cung cấp các công cụ dễ sử dụng để chuyển đổi YAML sang Python dictionary và ngược lại.
* Được sử dụng trong việc thiết lập tham số mô hình, lưu trữ siêu tham số trong học máy.

### Thư viện vẽ đồ thị

* *matplotlib*
* Là thư viện giúp trực quan hóa dữ liệu dưới dạng biểu đồ và đồ thị.
* Hỗ trợ nhiều loại biểu đồ như đường, cột, tán xạ, histogram, heatmap.
* Có thể kết hợp với numpy để vẽ dữ liệu một cách dễ dàng.

## Biểu đồ theo dõi hiệu suất của mô hình YOLOv5



* *Các thông số về tổn thất (Losses)*
* train/box\_loss: Đo lường lỗi (loss) khi xác định hộp bao quanh (bounding box) trong quá trình huấn luyện. Con số này càng thấp chứng tỏ mô hình dự đoán vị trí của các hộp bao quanh ngày càng chính xác.
* train/obj\_loss: Đo lường lỗi khi dự đoán đối tượng trong quá trình huấn luyện. Lỗi này giảm dần cho thấy mô hình ngày càng tốt hơn trong việc nhận diện có đối tượng hay không trong một vùng cụ thể.
* train/cls\_loss: Lỗi phân loại trong quá trình huấn luyện, nghĩa là lỗi khi mô hình dự đoán nhãn của đối tượng. Con số nhỏ đi thể hiện mô hình ngày càng phân loại chính xác.
* val/box\_loss, val/obj\_loss, val/cls\_loss: Tương ứng với các lỗi đã mô tả ở trên nhưng được đo lường trên dữ liệu kiểm định (validation data).
* *Các thông số đánh giá hiệu suất (Metrics)*
* metrics/precision: Precision (độ chính xác) đo lường tỷ lệ số dự đoán đúng (True Positives) trên tổng số dự đoán (bao gồm cả sai).
* metrics/recall: Recall (độ nhạy) đo lường tỷ lệ số dự đoán đúng (True Positives) trên tổng số thực tế cần phát hiện.
* metrics/mAP\_0.5: Mean Average Precision (mAP) ở ngưỡng IoU là 0.5. Đây là thước đo phổ biến để đánh giá mô hình trong bài toán phát hiện đối tượng, với giá trị càng cao chứng tỏ mô hình càng tốt.
* metrics/mAP\_0.5:0.95: Tương tự mAP, nhưng tính trung bình trên nhiều ngưỡng IoU (từ 0.5 đến 0.95). Đây là chỉ số toàn diện hơn để đo lường hiệu suất của mô hình.
* Kết luận: Losses càng thấp mô hình càng tốt vì chúng thể hiện mức độ lỗi giảm khi mô hình học. Metrics càng cao càng tốt vì chúng thể hiện mô hình ngày càng chính xác và hiệu quả.

# CHƯƠNG 3 QUY TRÌNH THỰC HIỆN NHẬN DIỆN LOẠI SỮA

## Chọn đối tượng nhận diện

* Lựa chọn 5 loại sữa: Đà lạt milk, Milo, Fami, Vinamilk óc chó, Vinamilk sữa tươi.

## Tiền xử lý dữ liệu

* Thu thập dữ liệu: Chụp ảnh cho 5 đối tượng, mỗi đối tượng 50 ảnh.
* Chuẩn hoa dữ liệu: Đưa tất cả các ảnh sau khi thu thập về kích thước 640x640.

## Xác định vùng chứa hộp sữa và trích chọn đặc trưng từ ảnh bằng cách gán nhãn với labelimg.

***Bước 1:*** Tải phần mềm labelimg và thực hiện gãn nhãn cho 5 đối tượng.

***Bước 2:*** Sau khi gán nhãn thi được file .txt chứa thông tin về tọa độ các đối tượng trong ảnh.

***Bước 3:*** Sắp xếp lại thư mục như sau: Ảnh cho vào 1 thư mục có tên là images, tệp được gán nhãn cho vào 1 thư mục có tên là labels.

***Bước 4:*** Cuối cùng, nén lại và tải lên google drive.

## Xây dựng model với mô hình huấn luyện yolov5 để phân biệt các loại sữa.

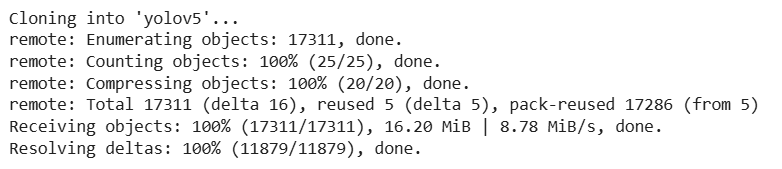
***Bước 1:*** Truy cập vào google colab và tạo mới một bản ghi. Thực hiện thay đổi “Change runtimr type: T4 GPU” tốc độ xử lý sẽ nhanh hơn là CPU, sau đó nhấn connect để kết nối.

***Bước 2:*** Kết nối thành công, bắt đầu thực hiện viết lệnh cho việc huấn luyện:

* Cài đặt yolov5: Sao chép repository yolov5 từ github về mục làm việc trên google colab. Khi hoàn thành, yolov5 sẽ được tải về trong thư mục /conten/yolov5

*!git clone https://github.com/ultralytics/yolov5 # clone*

Kết quả hiển thị sẽ như sau:



* Cài đặt thư viện cần thiết: Cài đặt các thư viện Python cần thiết từ file requirement.txt (nằm trong thư viện yolov5). Một số thư viện phổ biến được cài bao gồm: torch, torchvision, numpy, opencv-python-headless, PyYAML, matplotlib.

*!pip install -r /content/yolov5/requirements.txt*

* Kết nối với google drive: Để truy cập và xử lý dữ liệu trong Drive. Thư mục /content/gdrive sẽ là nơi lưu trữ dữ liệu và kết quả.

*from google.colab import drive*

*drive.mount('/content/gdrive')*

* Sao chép dữ liệu từ google drive: coco128.rar là tệp dữ liệu được nén (bao gồm ảnh và nhãn) dùng để huấn luyện mô hình.

*!cp /content/gdrive/MyDrive/coco128.rar /content*

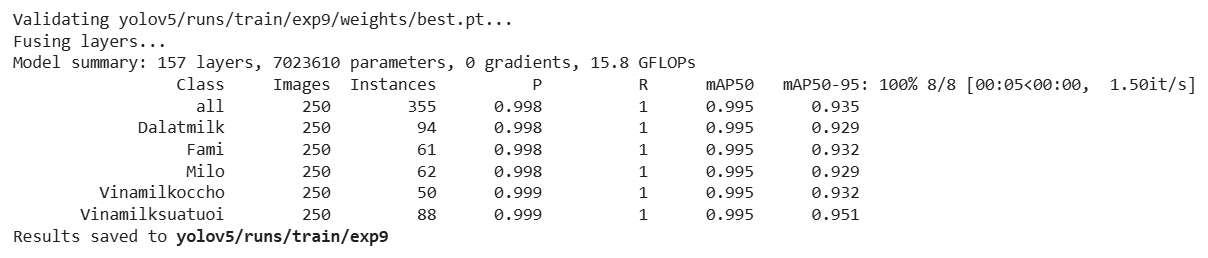
* Giải nén dữ liệu: Giải nén file .rar, sau khi giải nén, các tệp ảnh và nhãn sẽ nằm trong thư mục /content/coco128

*!unrar x /content/coco128.rar*

* Huấn luyện mô hình trên tệp dữ liệu với các tham số có ý nghĩa như sau:
  + train.py: File thực thi lệnh huấn luyện.
  + --img 640: Kích thước ảnh đưa vào mô hình (640x640)
  + --batch 16: Kích thước batch (số lượng ảnh xử lý mỗi lần)
  + --epochs 50: Số vòng lặp huấn luyện
  + –data coco128.yaml: File cấu hình dữ liệu (đường dẫn đến file .yaml chứa thông tin nhãn, số lớp, vị trí tệp huấn luyện.
  + –weights yolov5.pt: Trọng số ban đầu của yolov5 loại small.

*!python /content/yolov5/train.py --img 640 --batch 16 --epochs 50 --data coco128.yaml --weights yolov5s.pt*

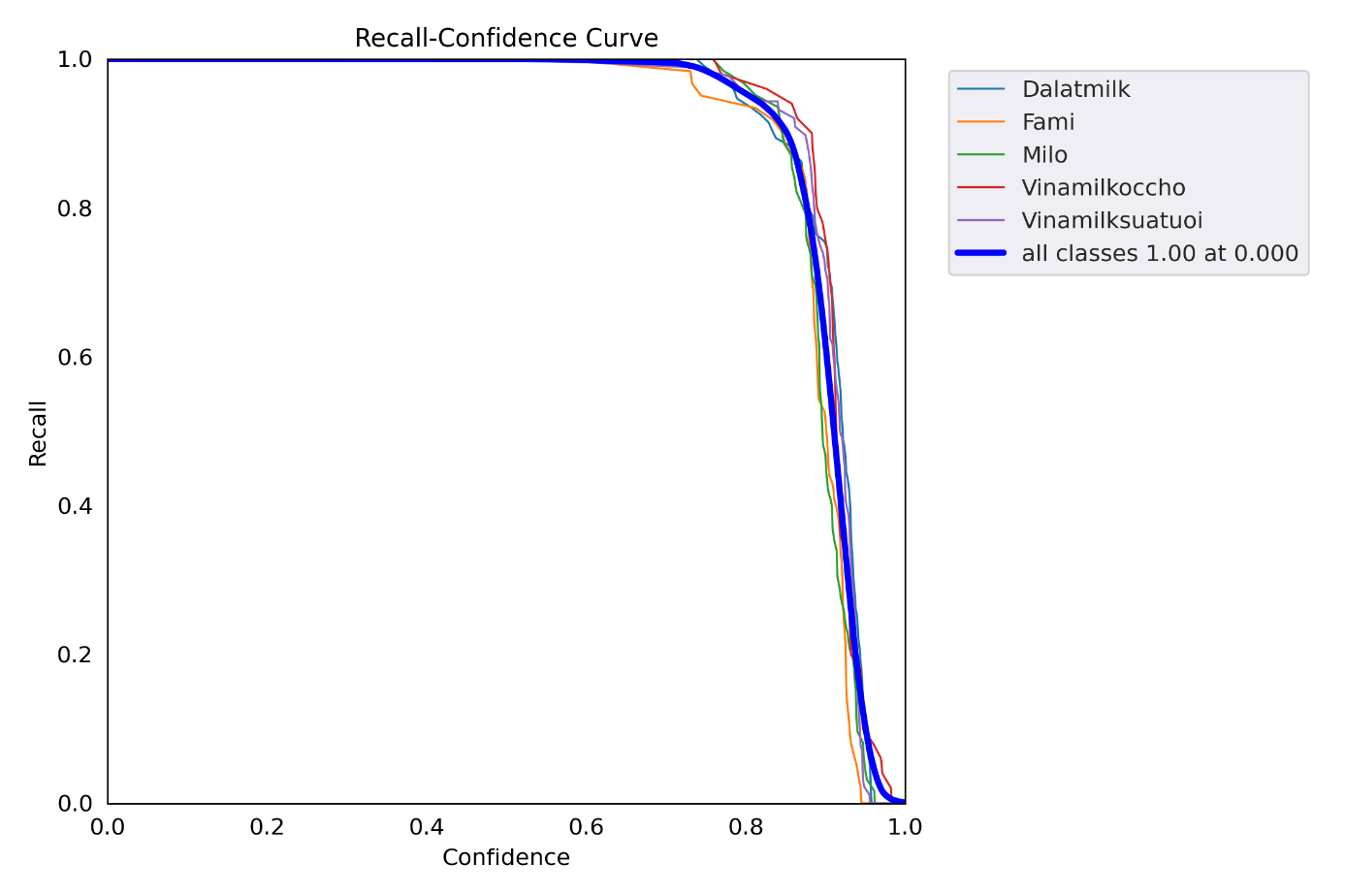
Kết quả huấn luyện xong sẽ được thông báo ở cuối như ảnh dưới đây:



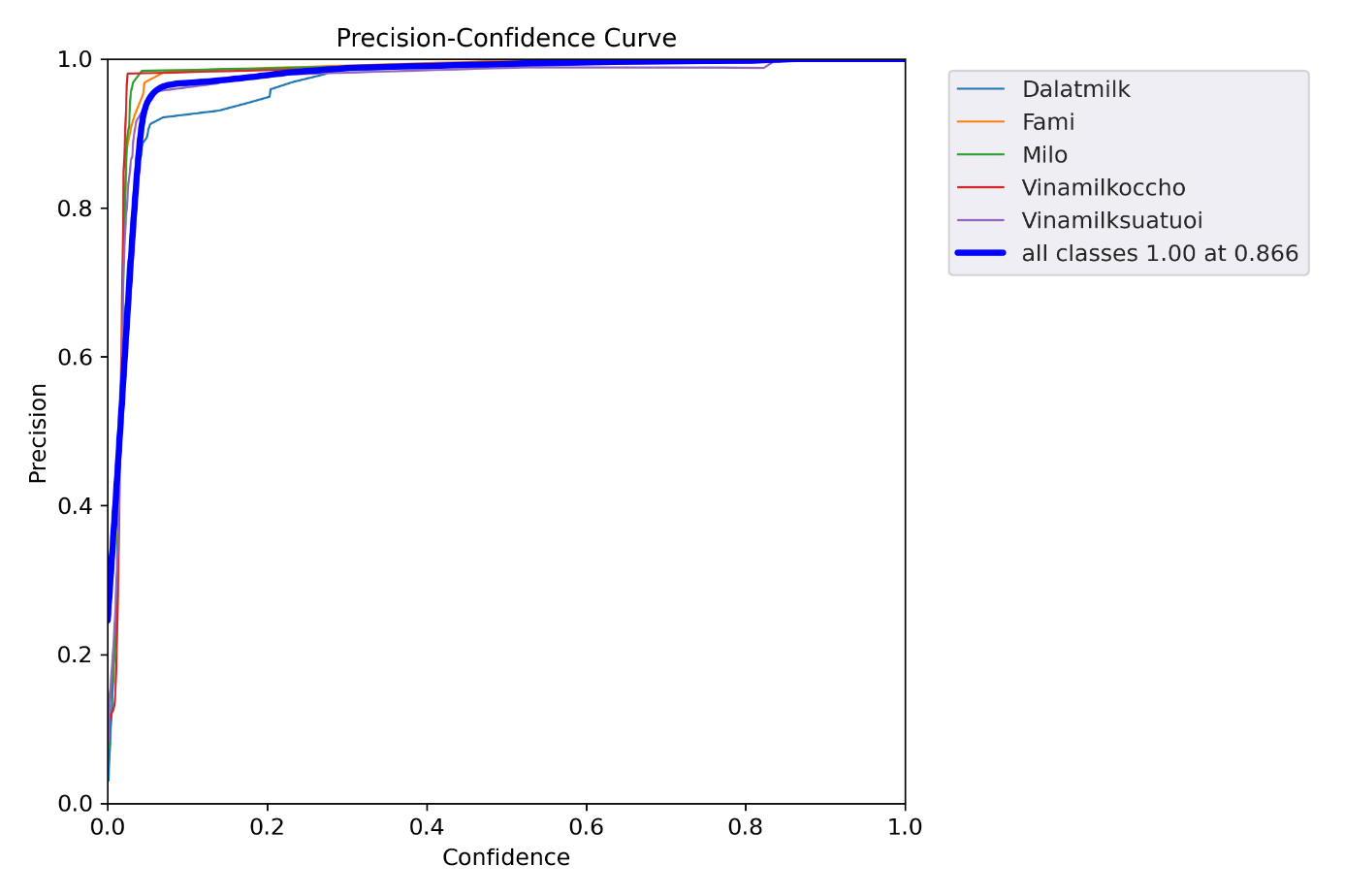
***Bước 3:*** Kiểm tra kết quả sau khi train.

Sau khi huấn luyện với 50 steps thì thu được các biểu đồ sau:

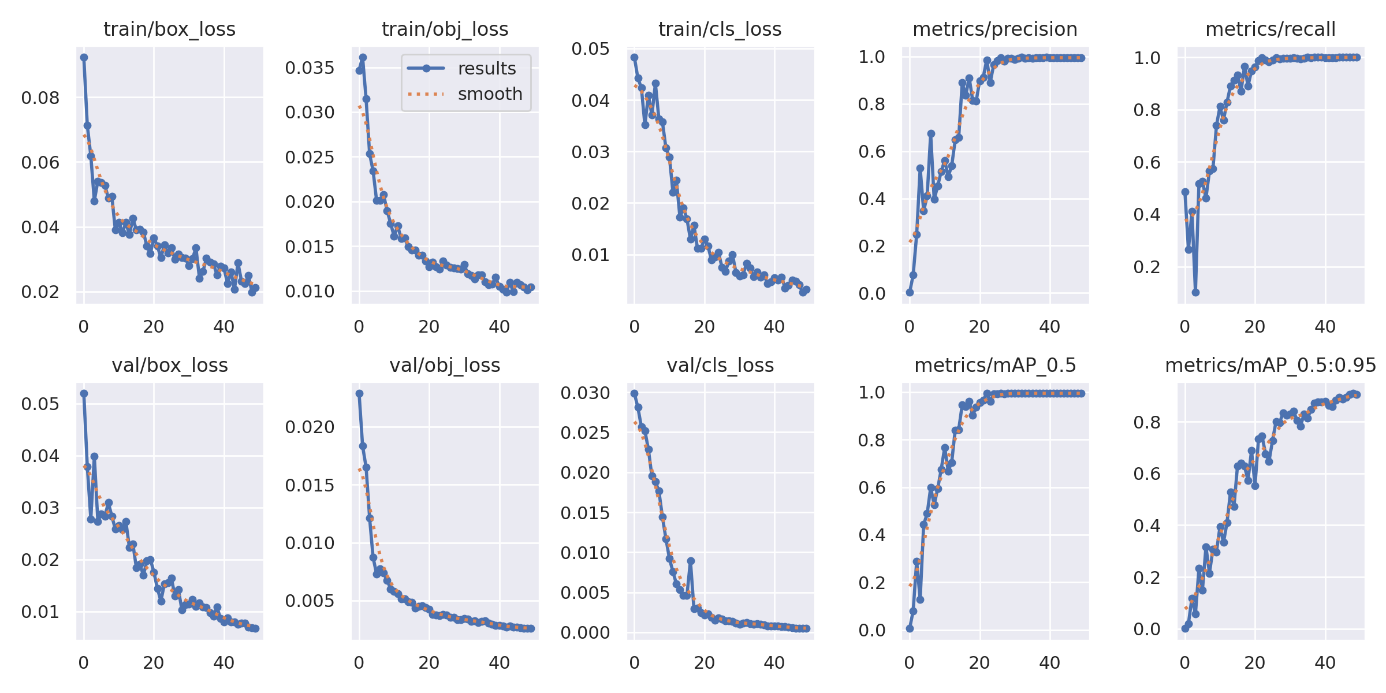
* Biểu đồ Recall Curve – Đường cong Recall cho biết tỷ lệ đối tượng thực sự có trong ảnh mà mô hình phát hiện đúng. Nếu giá trị Recall cao thì mô hình phát hiện được hầu hết đối tượng trong ảnh.



* Biểu đồ Precision-Recall Curve – Đường cong Precision-Recall thể hiện mối quan hệ giữa độ chính xác và độ bao phủ. Nếu đường cong gần góc trên bên phải thì mô hình cân bằng tốt giữa độ chính xác và độ bao phủ.



* Biểu đồ cho biết quá trình huấn luyện mô hình YOLOv5.

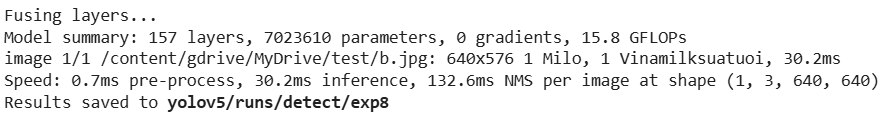


Nếu huấn luyện tiếp ta có thể dùng file best.pt trong thư mục exp/weights/ và lặp lại câu lệnh train như ở trên. Ở lần train này tôi thấy biểu đồ metrics/Map)0.5:0.95 thấy giá trị cuối cùng trên trục Y đạt >= 0.9 thì mô hình đã có thể dừng lại và download file best.pt.

* Dùng mô hình để kiểm tra nhận diện đối tượng: Thay đổi ‘yolov5s.pt’ trong file detect.py của thư mục yolov5 thành đường dẫn file best.pt, sau đó viết câu lệnh sau:

*!python /content/yolov5/detect.py --source /content/gdrive/MyDrive/test/b.jpg*

Sau khi chạy lệnh sẽ hiển thị kết quả, có thể xem hình ảnh kết quả ở thư mục exp.





## Xây dựng ứng dụng nhận diện bằng file best.pt

Sử dụng WinForms để thiết kế giao diện người dùng, sử dụng Flask (Python) để tải mô hình YoLov5 và thực hiện nhận diện ảnh.

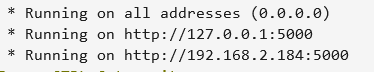
***Bước 1****:* Tải python 3.11, dùng visual studio, mở Terminal để tải YoLov5 từ GitHub và cài đặt các thư viện cần thiết.

git clone https://github.com/ultralytics/yolov5

pip install -r yolov5/requirements.txt

***Bước 2:*** Xây dựng API Flask để load mô hình YoLov5 từ tệp best.pt rồi xử lý ảnh và thực hiện nhận diện trả về kết quả.

Nhập câu lệnh: python app.py để chạy API Flask tại terminal. Chạy thành công sẽ hiển thị như ảnh.



***Bước 3:*** Xây dựng giao diện WinForms bằng C#

- Kéo thả để tạo giao diện đơn giản gồm các thành phần: Button (Nút tải ảnh và nhận diện), PictureBox (Hiển thị ảnh).

- Nội dung Form.cs sẽ thực hiện mở máy từ máy tính và gửi ảnh đến Flask API rồi trả về kết quả vẽ bounding box lên ảnh.



# CHƯƠNG 4 KẾT LUẬN

## Khó khăn và hướng giải quyết

|  |  |
| --- | --- |
| **Khó khăn** | **Hướng giải quyết** |
| Sau khi gán nhãn ảnh, băn khoăn giữa dùng TensorFlow và YoLov5. | Chọn YoLov5 các thức thực hiện đơn giản và dễ hiểu hơn. |
| Không biết train bao nhiêu lần là hợp lí. | Thực hiện train thử với 7 epochs, 50 epochs, 100 epochs riêng biệt.  Sau khi xem xét kết quả của từng exp thì lựa chọn 50 epochs huấn luyện. |
| Tải mô hình bị lỗi trên Windows. | Dùng pathlib.PosixPath để tương thích trên Windows. |

## Kết quả đã đạt được

- Huấn luyện được mô hình YoLov5 dựa trên tập dữ liệu về 5 đối tượng.

- Kết quả nhận diện có thể trả về thông tin: tọa độ của khung bao quanh sữa, độ chính xác của dự đoán, tên của loại sữa được nhận diện.

- Xây dựng được API Flask chạy ổn định trên http://127.0.0.1:5000/predict.

- Tích hợp Flask API với giao diện WinForm (C#) để xây dựng ứng dụng.

## Vấn đề tồn dồn và phương hướng phát triển khai, nâng cấp

|  |  |
| --- | --- |
| **Tồn đọng** | **Giải pháp** |
| Độ chính xác chưa cao | Tăng số lượng ảnh huấn luyện, điều chỉnh tham số mô hình |
| Giao tiếp giữa Flask và C# bị chậm | Tối ưu Flask API, tối ưu truyền dữ liệu giữa C# và Flask,.. |