**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM  
KHOA CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “XỬ LÝ ẢNH”**

**Đề tài:**

**XÂY DỰNG CHƯƠNG TRÌNH NHẬN DẠNG TIỀN ĐƠN GIẢN**

**GVHD : TS.Nguyễn Hữu Tuân**

**Sinh Viên Thực Hiện : Ngô Quốc Khánh – 92950**

**Hải Phòng , tháng 05 năm 2024**

**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KHOA HỌC MÁY TÍNH**

**-----\*\*\*-----**

**Báo Cáo Bài Tập Lớn**

**Học Phần : Xử Lý Ảnh**

1. **Tên đề tài**

* Xây dựng chương trình nhận dạng tiền đơn giản .

1. **Mục Đích**

* Phát triển một chương trình máy tính có khả năng nhận dạng mệnh giá các tờ tiền của Việt Nam. Chương trình sẽ phân biệt các mệnh giá tiền khác nhau thông qua hình ảnh và camera.

1. **Công Việc Thực Hiện**

* **Thu Nhập Dữ Liệu:**
* **Thu thập ảnh**: Tập hợp các ảnh chất lượng cao của các mệnh giá tiền Việt Nam từ các nguồn như ngân hàng, internet hoặc chụp trực tiếp bằng camera.
* **Xử lý ảnh**: Đảm bảo rằng tất cả các ảnh đều được chuẩn hóa về kích thước, độ phân giải, và định dạng để phù hợp với yêu cầu của mô hình học máy.
* **Huấn Luyện Mô Hình:**
* **Xây dựng mô hình:** Sử dụng các mạng nơ-ron tích chập (CNN) để phân loại ảnh. Các kiến trúc như VGG16, ResNet, hoặc các mô hình tùy chỉnh có thể được huấn luyện để nhận dạng mệnh giá tiền.
* **Huấn luyện và tối ưu:** Huấn luyện mô hình trên tập dữ liệu đã thu thập, sử dụng các kỹ thuật như tăng cường ảnh (data augmentation) và tối ưu hóa hyperparameters để cải thiện hiệu suất.
* **Phát Triển Giao Diện Người Dùng:**
* **Thiết kế giao diện:** Phát triển giao diện người dùng thân thiện, cho phép người dùng tải ảnh và hiển thị kết quả nhận dạng mệnh giá.
* **Tích hợp camera**: Cho phép người dùng chụp ảnh tờ tiền trực tiếp qua camera tích hợp trong ứng dụng và nhận dạng mệnh giá ngay lập tức.

1. **Yêu Cầu**

* Chuẩn bị các phần mềm và thư viện cần thiết Python , opencv , Tkinter …. V.v
* Kỹ năng lập trình và kiến thức về học phần xử lý ảnh
* Bộ sưu tập các tờ tiền Việt Nam để làm Dữ Liệu

**Hải Phòng , ngày 18 tháng 05 năm 2024**

**Người Hướng Dẫn**

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 4](#_Toc167993260)

[PHẦN 1 : GIỚI THIỆU BÀI TOÁN 5](#_Toc167993261)

[1. Đặt vấn đề 5](#_Toc167993262)

[2. Một số vấn đề cần lưu ý 5](#_Toc167993263)

[Phần 2 : Cơ Sở Lý Thuyết 6](#_Toc167993264)

[1. Ngôn Ngữ Lập Trình Python 6](#_Toc167993265)

[2. Một số thư viện được sử dụng 7](#_Toc167993266)

[1.1 Tkinter 7](#_Toc167993267)

[1.2 Pillow (PIL) 7](#_Toc167993268)

[1.3 OpenCV 7](#_Toc167993269)

[1.4 PyTorch 7](#_Toc167993270)

[1.5 Ultralytics 8](#_Toc167993271)

[1.6 datetime 8](#_Toc167993272)

[2.Mô Hình YOLOv8 8](#_Toc167993273)

[2.1 YOLO 8](#_Toc167993274)

[2.2 YOLOv8 10](#_Toc167993275)

[Phần 3 : Xây Dựng Chương Trình 11](#_Toc167993276)

[1. Phân Tích Bài Toán 11](#_Toc167993277)

[2. Sơ đồ hệ thống chương trình 13](#_Toc167993278)

[3. Xử Lý Dữ Liêụ Hình Ảnh 14](#_Toc167993279)

[Bước 1 : Thu Nhập Hình Ảnh 14](#_Toc167993280)

[Bước 2 : Tổ Chức Dữ Liệu 14](#_Toc167993281)

[Bước 3 : Sử dụng Robotflow để dán nhãn 15](#_Toc167993282)

[Kết quả sau khi train 18](#_Toc167993283)

[Báo cáo Hiệu suất Mô hình YOLOv8 Nhận dạng Tiền Việt Nam 20](#_Toc167993286)

[4. Nhận dạng Tiền Qua Cam , Hình Ảnh 21](#_Toc167993287)

[Phần 4 : Kết Luận 24](#_Toc167993288)

[1. Ưu Điểm 24](#_Toc167993289)

[2. Nhược Điểm 24](#_Toc167993290)

[3. Kết Luận 25](#_Toc167993291)

# 

# DANH MỤC HÌNH ẢNH

[Hình 1 : Sơ Đồ Hệ Thống Chương Trình 13](#_Toc167993425)

[Hình 2 : Thu Nhập Dữ Liệu 14](#_Toc167993426)

[Hình 5 : Ví dụ về tải ảnh lên và gán nhãn 15](#_Toc167993428)

[Hình 4 : Class 1000 dã được dán nhãn 15](#_Toc167993429)

[Hình 6 : quá trình tải dữ liệu lên kaggle 16](#_Toc167993430)

[Hình 7 : số lần train với epoch đag chạy 18](#_Toc167993431)

[Hình 8 : Confusion\_matrix 18](#_Toc167993432)

[Hình 9 : F1-confidem Curve 19](#_Toc167993433)

[Hình 10 : Hình ảnh Nhận dạng Qua Cam 22](#_Toc167993434)

[Hình 11 : Nhận Dạng Qua Hình Ảnh 24](#_Toc167993435)

# PHẦN 1 : GIỚI THIỆU BÀI TOÁN

## Đặt vấn đề

* Chúng ta đang phát triển một chương trình có khả năng nhận dạng tiền tệ từ hình ảnh và video thu thập qua camera. Chương trình này sau có thể phát triển thành các phần mềm trong việc thanh toán và quản lý .Ứng dụng này hướng đến việc đếm tiền tự động và cải thiện quản lý tài chính, bằng cách xác định tự động mệnh giá tiền khi thanh toán tại quầy. Điều này không chỉ cải thiện hiệu quả trong quản lý tài chính mà còn giúp tăng cường tính tự động và chính xác của các giao dịch.
* Chương trình này sử dụng mô hình YOLOv8 để đảm bảo độ chính xác cao trong nhận dạng các loại tiền khác nhau. Quá trình phát triển bao gồm chuẩn bị dữ liệu kỹ lưỡng, huấn luyện mô hình, và tinh chỉnh kỹ thuật để tối ưu hóa hiệu suất, đảm bảo tính ứng dụng cao trong môi trường thực tế

## Một số vấn đề cần lưu ý

* Độ Chính Xác
* Cần có dữ liệu huấn luyện đa dạng và đầy đủ, bao gồm các mệnh giá khác nhau, mẫu tiền từ nhiều góc độ và trong điều kiện ánh sáng khác nhau.
* Đa Dạng Về Mệnh Giá và Loại Tiền
* Thu thập và huấn luyện dữ liệu từ nhiều nguồn khác nhau để đảm bảo tính đa dạng.
* Độ Tin Cậy
* Sử dụng các kỹ thuật tinh chỉnh và điều chỉnh mô hình để cải thiện độ tin cậy của các dự đoán.
* Thời Gian Thực Hiện
* Tối ưu hóa mô hình và thuật toán để giảm độ trễ và tăng tốc độ xử lý

# Phần 2 : Cơ Sở Lý Thuyết

## Ngôn Ngữ Lập Trình Python

Python là một ngôn ngữ lập trình bậc cao, được tạo ra bởi Guido van Rossum và phát hành lần đầu vào năm 1991. Python nổi bật với cú pháp rõ ràng, dễ đọc và khả năng mở rộng mạnh mẽ, thường được sử dụng trong nhiều lĩnh vực như phát triển web, phân tích dữ liệu và trí tuệ nhân tạo.

Lịch sử các phiên bản Python

Python 1.0 (1991): Phiên bản đầu tiên.

Python 2.0 (2000): Giới thiệu nhiều cải tiến.

Python 3.0 (2008): Cải tiến lớn với nhiều tính năng mới.

Đặc điểm nổi bật của Python

Cú pháp rõ ràng và dễ đọc: Giúp việc viết và bảo trì mã dễ dàng hơn.

Đa nền tảng: Chạy trên nhiều hệ điều hành như Windows, macOS và Linux.

Thư viện phong phú: Hỗ trợ nhiều lĩnh vực khác nhau như NumPy, Pandas, Matplotlib cho phân tích dữ liệu và Django, Flask cho phát triển web.

Hỗ trợ lập trình hướng đối tượng và lập trình chức năng: Linh hoạt trong việc chọn phong cách lập trình.

Cộng đồng lớn: Nhiều tài liệu, khóa học và nhóm hỗ trợ có sẵn.

Ứng dụng thực tế của Python

Phát triển web: Sử dụng các framework như Django và Flask.

Khoa học dữ liệu và học máy: Sử dụng các thư viện như NumPy, Pandas, TensorFlow.

Tự động hóa và kịch bản hóa: Viết các kịch bản để tự động hóa các tác vụ.

Phát triển trò chơi: Sử dụng thư viện như Pygame.

## Một số thư viện được sử dụng

### 1.1 Tkinter

**Giới thiệu:** Thư viện chuẩn của Python cho giao diện đồ họa (GUI), phát triển bởi John Ousterhout từ năm 1991.

**Cách sử dụng:** Import và khởi tạo cửa sổ chính, thêm widget và thiết lập sự kiện.

**Công dụng:** Tạo GUI trực quan cho ứng dụng desktop.

**Chức năng chính:** Tạo cửa sổ, khung chứa, nhãn, nút bấm, ô nhập liệu, hộp thoại chọn tệp tin, combobox.

### 1.2 Pillow (PIL)

**Giới thiệu:** Thư viện xử lý ảnh mạnh mẽ, tiếp nối từ Python Imaging Library (PIL), phát triển bởi Alex Clark vào năm 2010.

**Cách sử dụng:** Mở, thay đổi kích thước, cắt, xoay, và áp dụng bộ lọc lên ảnh.

**Công dụng:** Xử lý ảnh số, tích hợp tốt với Tkinter.

**Chức năng chính:** Mở và xử lý ảnh, chuyển đổi ảnh PIL thành định dạng hiển thị trong Tkinter.

### 1.3 OpenCV

**Giới thiệu:** Thư viện mã nguồn mở mạnh mẽ cho xử lý ảnh và thị giác máy tính, phát triển bởi Intel vào năm 2000.

**Cách sử dụng:** Cài đặt qua pip, hỗ trợ nhiều định dạng ảnh và video, cung cấp hàm chuyển đổi màu sắc, làm mịn ảnh, phát hiện cạnh.

**Công dụng:** Nhận dạng đối tượng, xử lý video, ứng dụng thị giác máy tính.

**Chức năng chính:** Mở và xử lý luồng video, đọc ảnh từ tệp, chuyển đổi giữa các không gian màu, vẽ hình chữ nhật, thêm văn bản lên ảnh.

### 1.4 PyTorch

**Giới thiệu:** Framework học sâu mã nguồn mở, phát triển bởi Facebook's AI Research lab (FAIR) vào năm 2016.

**Cách sử dụng:** Cung cấp tensor và module để xây dựng mô hình neural network, hỗ trợ GPU tăng tốc tính toán.

**Công dụng:** Xây dựng và huấn luyện mô hình học sâu.

**Chức năng chính:** Làm việc với tensor, xây dựng và huấn luyện mô hình neural network.

### 1.5 Ultralytics

**Giới thiệu:** Thư viện chứa các phiên bản mới của mô hình YOLO, phát triển bởi Glenn Jocher vào năm 2020.

**Cách sử dụng:** Tải và sử dụng các mô hình YOLO để nhận dạng đối tượng từ hình ảnh và video.

**Công dụng:** Triển khai và sử dụng mô hình YOLO cho nhận dạng đối tượng thời gian thực.

**Chức năng chính:** Tải và sử dụng mô hình YOLO, huấn luyện và đánh giá mô hình.

### 1.6 datetime

Giới thiệu: Thư viện tiêu chuẩn của Python để xử lý và định dạng ngày giờ, phát triển từ phiên bản đầu tiên của Python vào năm 1991.

Cách sử dụng: Lấy thời gian hiện tại, tính toán sự khác biệt giữa các khoảng thời gian, định dạng ngày giờ thành chuỗi.

Công dụng: Xử lý các tác vụ liên quan đến thời gian.

Chức năng chính: Làm việc với đối tượng thời gian, định dạng và tính toán thời gian.

## 2.Mô Hình YOLOv8

### 2.1 YOLO

YOLO (You Only Look Once) là một mô hình học sâu nổi tiếng trong nhận dạng đối tượng, được Joseph Redmon và Ali Farhadi giới thiệu lần đầu vào năm 2015. YOLO được biết đến với tốc độ và độ chính xác cao trong việc nhận dạng đối tượng.

Các phiên bản của YOLO

YOLOv1 (2015): Phiên bản đầu tiên, giới thiệu mô hình nhận dạng đối tượng nhanh và chính xác.

YOLOv2 (2016): Cải thiện với batch normalization, anchor boxes, và dimension clusters.

YOLOv3 (2018): Nâng cao hiệu suất với backbone network hiệu quả hơn, nhiều anchor hơn và spatial pyramid pooling.

YOLOv4 (2020): Giới thiệu Mosaic data augmentation, anchor-free detection head mới, và hàm mất mát mới.

YOLOv5 (2020): Tích hợp tối ưu hóa siêu tham số, theo dõi thí nghiệm và xuất khẩu tự động sang các định dạng phổ biến.

YOLOv6 (2022): Được mở mã nguồn bởi Meituan, sử dụng trong các robot giao hàng tự động.

YOLOv7 (2022): Thêm các nhiệm vụ bổ sung như ước tính tư thế trên bộ dữ liệu COCO keypoints.

YOLOv8 (2023): Phiên bản mới nhất của Ultralytics, hỗ trợ nhận dạng, phân đoạn, ước tính tư thế, theo dõi và phân loại.

YOLOv9 (2024): Giới thiệu Programmable Gradient Information (PGI) và Generalized Efficient Layer Aggregation Network (GELAN).

**Ưu nhược điểm của YOLO**

Ưu điểm:

Tốc độ cao: Nhận dạng đối tượng trong thời gian thực.

Độ chính xác cao: Phù hợp cho nhiều ứng dụng.

Dễ triển khai: Có nhiều tài liệu và cộng đồng hỗ trợ.

Nhận dạng đa đối tượng: Nhận dạng nhiều loại đối tượng trong một ảnh.

Nhược điểm:

Yêu cầu tài nguyên cao: Cần phần cứng mạnh.

Hiệu suất trên đối tượng nhỏ: Khó khăn với đối tượng nhỏ.

Yêu cầu dữ liệu huấn luyện lớn: Cần lượng dữ liệu lớn.

Ứng dụng của YOLO trong nhận dạng tiền tệ

Hiệu suất vượt trội: Nhận dạng các mệnh giá tiền tệ nhanh chóng và chính xác, quan trọng trong máy đếm tiền, máy ATM, và kiểm tra tiền tệ trong ngân hàng.

Ứng dụng đa dạng: Nhận dạng nhiều loại tiền tệ khác nhau, giúp giảm gian lận và sai sót trong kiểm tra và xác thực tiền tệ.

Dễ triển khai: Hỗ trợ từ thư viện Ultralytics và cộng đồng người dùng, dễ dàng triển khai và tinh chỉnh mô hình.

### 2.2 YOLOv8

Giới thiệu về YOLOv8

YOLOv8 là phiên bản mới nhất của mô hình YOLO, cải thiện đáng kể về tốc độ và độ chính xác, phù hợp cho nhiều ứng dụng nhận dạng đối tượng thời gian thực.

Phân loại YOLOv8

YOLOv8n (Nano): Mô hình nhỏ nhất, tiết kiệm tài nguyên.

YOLOv8s (Small): Cân bằng giữa tốc độ và độ chính xác.

YOLOv8m (Medium):Cải thiện độ chính xác với chi phí tăng tài nguyên.

YOLOv8l (Large): Ưu tiên độ chính xác cao.

YOLOv8x(Extra Large):Tối đa hóa độ chính xác,yêu cầu tài nguyên cao.

Ưu nhược điểm của YOLOv8

Ưu điểm:

Tốc độ cao: Xử lý ảnh nhanh, nhận dạng thời gian thực.

Độ chính xác cao: Sử dụng phương pháp tiên tiến trong học sâu.

Dễ triển khai: Công cụ và tài liệu chi tiết từ Ultralytics.

Nhận dạng đa đối tượng: Nhận dạng nhiều loại đối tượng trong một ảnh.

Cộng đồng hỗ trợ mạnh mẽ: Nhiều tài liệu và ví dụ có sẵn.

Nhược điểm:

Yêu cầu tài nguyên cao: Cần phần cứng mạnh.

Phức tạp trong tùy chỉnh: Đòi hỏi kiến thức học sâu.

Hiệu suất trên đối tượng nhỏ: Có thể gặp khó khăn.

Yêu cầu dữ liệu huấn luyện lớn: Cần dữ liệu chất lượng.

Cách sử dụng YOLOv8

YOLOv8 có thể được tải và sử dụng qua thư viện Ultralytics.

Lý do nên chọn YOLOv8

Tốc độ cao: Xử lý ảnh nhanh, phù hợp cho nhận dạng thời gian thực.

Độ chính xác cao: Giảm thiểu lỗi, cải thiện hiệu quả hệ thống.

Dễ dàng triển khai:Thư viện Ultralytics,tài liệu chi tiết, hỗ trợ cộng đồng.

Nhận dạng đa đối tượng: Tối ưu hóa kiểm tra và quản lý phức tạp.

Tiết kiệm tài nguyên: Phiên bản như YOLOv8n và YOLOv8s phù hợp với thiết bị hạn chế phần cứng.

Hỗ trợ mạnh mẽ: Cộng đồng lớn, nhiều tài liệu và ví dụ.

Ứng dụng YOLOv8 trong nhận dạng tiền tệ

Hiệu suất vượt trội: Nhận dạng mệnh giá tiền nhanh chóng và chính xác.

Giảm gian lận: Giảm thiểu sai sót trong kiểm tra và xác thực tiền tệ.

Tăng cường tự động hóa: Hỗ trợ tự động hóa quy trình kiểm tra tiền tệ.

# Phần 3 : Xây Dựng Chương Trình

## Phân Tích Bài Toán

**1. Thu thập Dữ liệu**

Từ Mạng: Tải ảnh tiền tệ từ các nguồn trực tuyến.

Chụp Trực Tiếp: Sử dụng camera để chụp ảnh các mệnh giá khác nhau.

**2. Dán nhãn Dữ liệu**

Roboflow: Tải lên và dán nhãn các ảnh với mệnh giá: 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000.

**3. Huấn luyện Mô hình**

Chuẩn bị Dữ liệu: Tải tập dữ liệu đã dán nhãn từ Roboflow.

Thiết kế Mô hình: Sử dụng YOLOv8 để nhận dạng tiền tệ.

Huấn luyện trên Kaggle: Tải dữ liệu lên Kaggle, huấn luyện mô hình, và lưu lại mô hình (best.pt).

**4. Sử dụng Mô hình**

Nhận dạng Từ Hình Ảnh: Tải ảnh lên và nhận dạng mệnh giá tiền tệ.

Nhận dạng Từ Camera: Sử dụng camera để nhận dạng tiền tệ trực tiếp.

**5. Hiển thị Kết quả**

Hiển thị ảnh gốc và ảnh đã xử lý với kết quả nhận dạng trên giao diện người dùng.

**6. Cải thiện và Bảo trì Mô hình**

Thu thập Thêm Dữ liệu: Liên tục cập nhật thêm ảnh từ mạng và chụp trực tiếp.

Huấn luyện Lại: Định kỳ huấn luyện lại mô hình với dữ liệu mới.

## Sơ đồ hệ thống chương trình



Hình 1 : Sơ Đồ Hệ Thống Chương Trình

**Quy trình Nhận dạng Tiền tệ**

**1. Bắt đầu**

Khởi động chương trình.

**2. Nhận dạng bằng Hình ảnh hay Camera**

Hình Ảnh: Nhận ảnh đầu vào từ người dùng.

Camera: Nhận trực tiếp hình ảnh liên tục từ webcam.

**3. Nhận Dạng bằng Mô Hình Đã Đào Tạo (YOLOv8)**

Sử dụng mô hình YOLOv8 để nhận dạng các mệnh giá tiền tệ.

**4. Hiển Thị Kết Quả Nhận Dạng**

Hiển thị kết quả nhận dạng lên giao diện người dùng, bao gồm mệnh giá tiền tệ và độ tin cậy.

**5. Kết Thúc**

Hoàn thành quá trình nhận dạng và hiển thị kết quả.

## Xử Lý Dữ Liêụ Hình Ảnh

### Bước 1 : Thu Nhập Hình Ảnh

* **Thu Nhập Hình ảnh trên Internet**

Sử dụng các công cụ tìm kiếm hình ảnh như Google Images để tìm kiếm và tải xuống các hình ảnh của các tờ tiền với các mệnh giá khác nhau.

* **THU NHẬP HÌNH ẢNH TRỰC TIẾP**

Sử dụng điện thoại hoặc máy ảnh để chụp ảnh các tờ tiền. Đảm bảo rằng hình ảnh được chụp từ các góc độ và điều kiện ánh sáng khác nhau để có bộ dữ liệu đa dạng hơn.

Sau là hình ảnh về những ảnh đã được dùng để làm dữ liệu cho chương trình

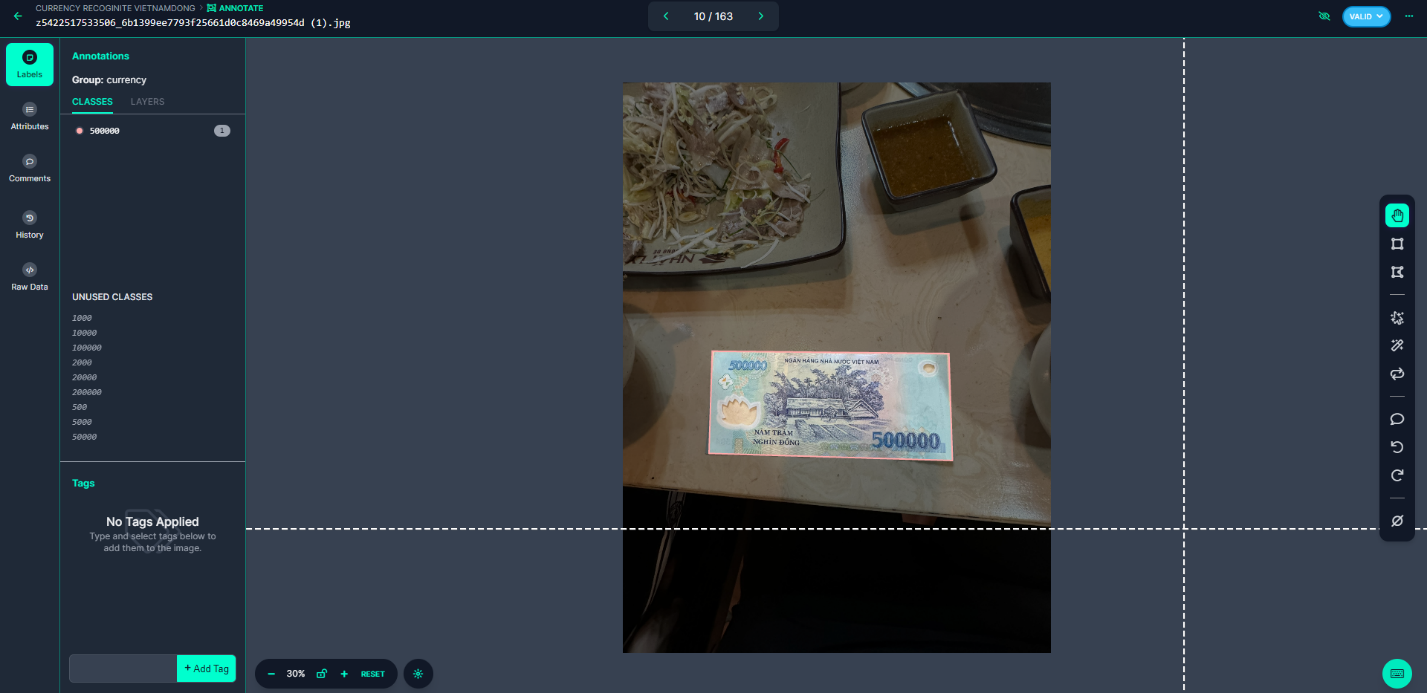
### Bước 2 : Tổ Chức Dữ Liệu

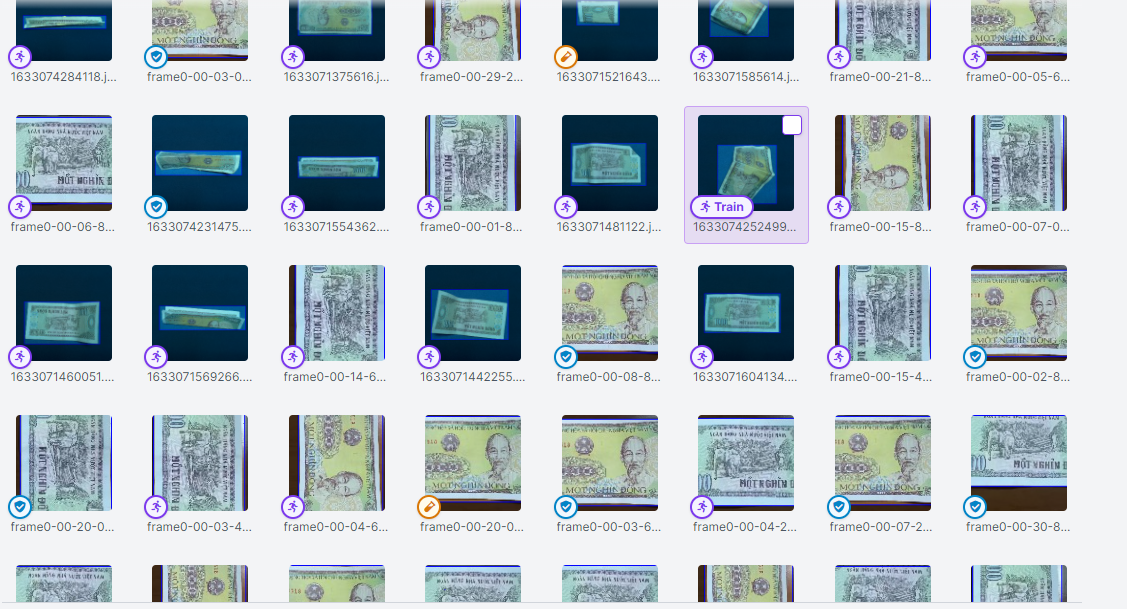
Hình 2 : Thu Nhập Dữ Liệu

Xây Dựng các class là 1000 , 2000 , 5000 , 10000 , 20000 , 50000, 100000,200000,500000

### Bước 3 : Sử dụng Robotflow để dán nhãn

* Tải Hình Ảnh lên
* Dán nhãn



Hình 5 : Ví dụ về tải ảnh lên và gán nhãn

Hình 4 : Class 1000 dã được dán nhãn

Huấn Luyện Mô Hình

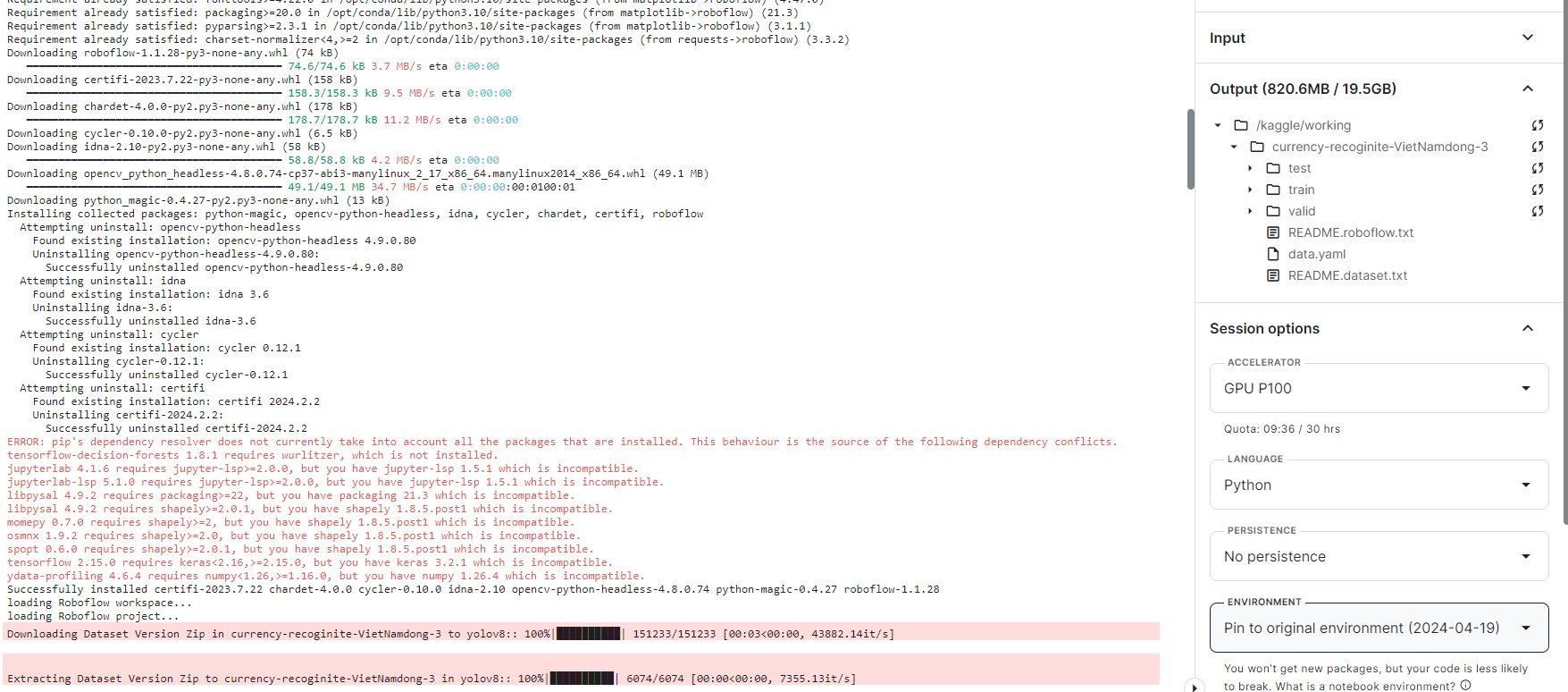
Sử Dụng Kaggle Để Train

Đoạn code này dùng để thêm thự viện yolov8

|  |
| --- |
| # Pip install method (recommended)  !pip install ultralytics==8.0.196  from IPython import display  display.clear\_output()  import ultralytics  ultralytics.checks() |
| from ultralytics import YOLO  from IPython.display import display, Image |

Đoạn code tiếp theo để tải dữ liệu từ robotflow sang kaggle

|  |
| --- |
| !pip install roboflow  from roboflow import Roboflow  rf = Roboflow(api\_key="9FUlIHtGhPMZw0yYnzD5")  project = rf.workspace("khanh-ngo").project("currency-recoginite-vietnamdong")  version = project.version(3)  dataset = version.download("yolov8") |



Hình 6 : quá trình tải dữ liệu lên kaggle

Code để train là

|  |
| --- |
| !yolo task=detect mode=train model=yolov8m.pt data=/kaggle/working/currency-recoginite-VietNamdong-3/data.yaml epochs=160 imgsz=640 device=0 |

!yolo: Gọi lệnh YOLO.

task=detect: Thực hiện nhiệm vụ phát hiện đối tượng.

mode=train: Chế độ huấn luyện mô hình.

model=yolov8m.pt: Sử dụng mô hình YOLOv8 phiên bản trung bình.

data=/kaggle/working/currency-recoginite-VietNamdong-3/data.yaml: Đường dẫn đến tệp cấu hình dữ liệu.

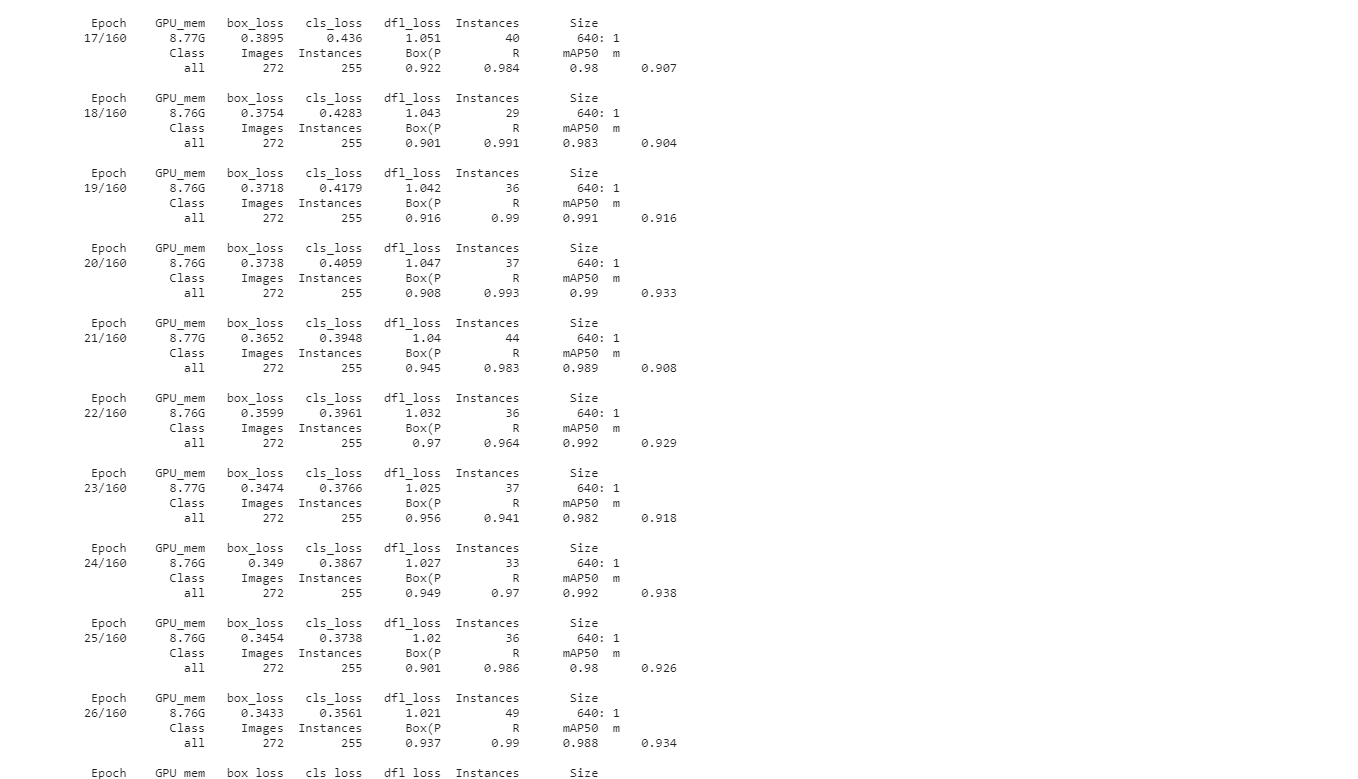
epochs=160: Huấn luyện mô hình trong 160 epoch.Huấn luyện trong 160 epoch có nghĩa là mô hình sẽ đi qua toàn bộ tập dữ liệu 160 lần.

imgsz=640: Kích thước hình ảnh là 640x640 pixel.

device=0: Sử dụng GPU đầu tiên để huấn luyện.

đây là data.yaml

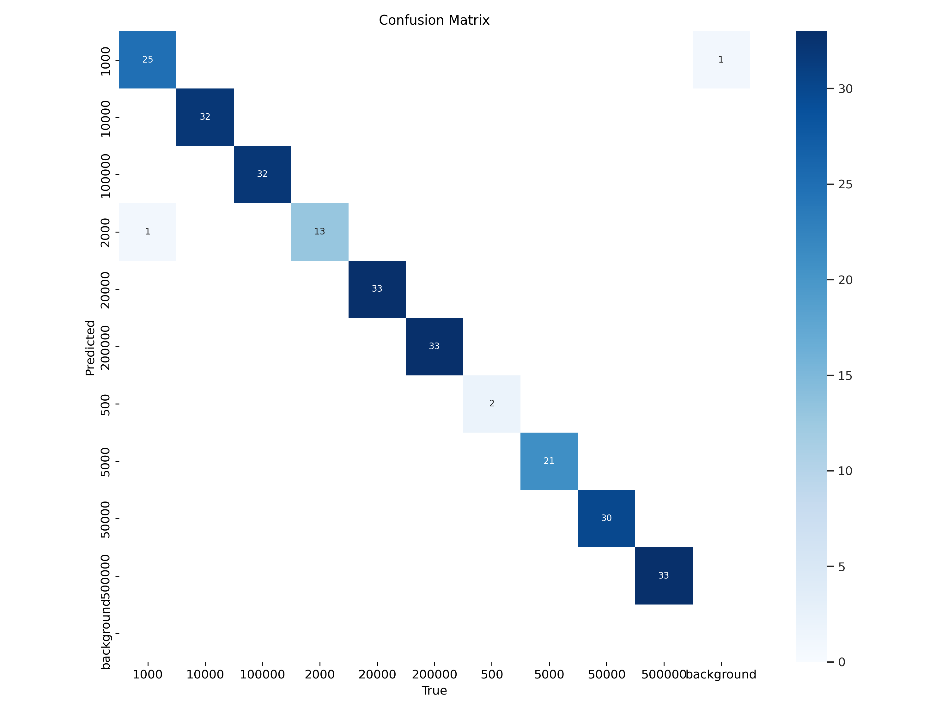
|  |
| --- |
| train: ../train/images  val: ../valid/images  test: ../test/images  nc: 10  names: ['1000', '10000', '100000', '2000', '20000', '200000', '500', '5000', '50000', '500000']  roboflow:  workspace: khanh-ngo  project: currency-recoginite-vietnamdong  version: 3  license: CC BY 4.0  url: https://universe.roboflow.com/khanh-ngo/currency-recoginite-vietnamdong/dataset/3 |



Hình 7 : số lần train với epoch đag chạy

## Kết quả sau khi train

### Confusion\_matrix

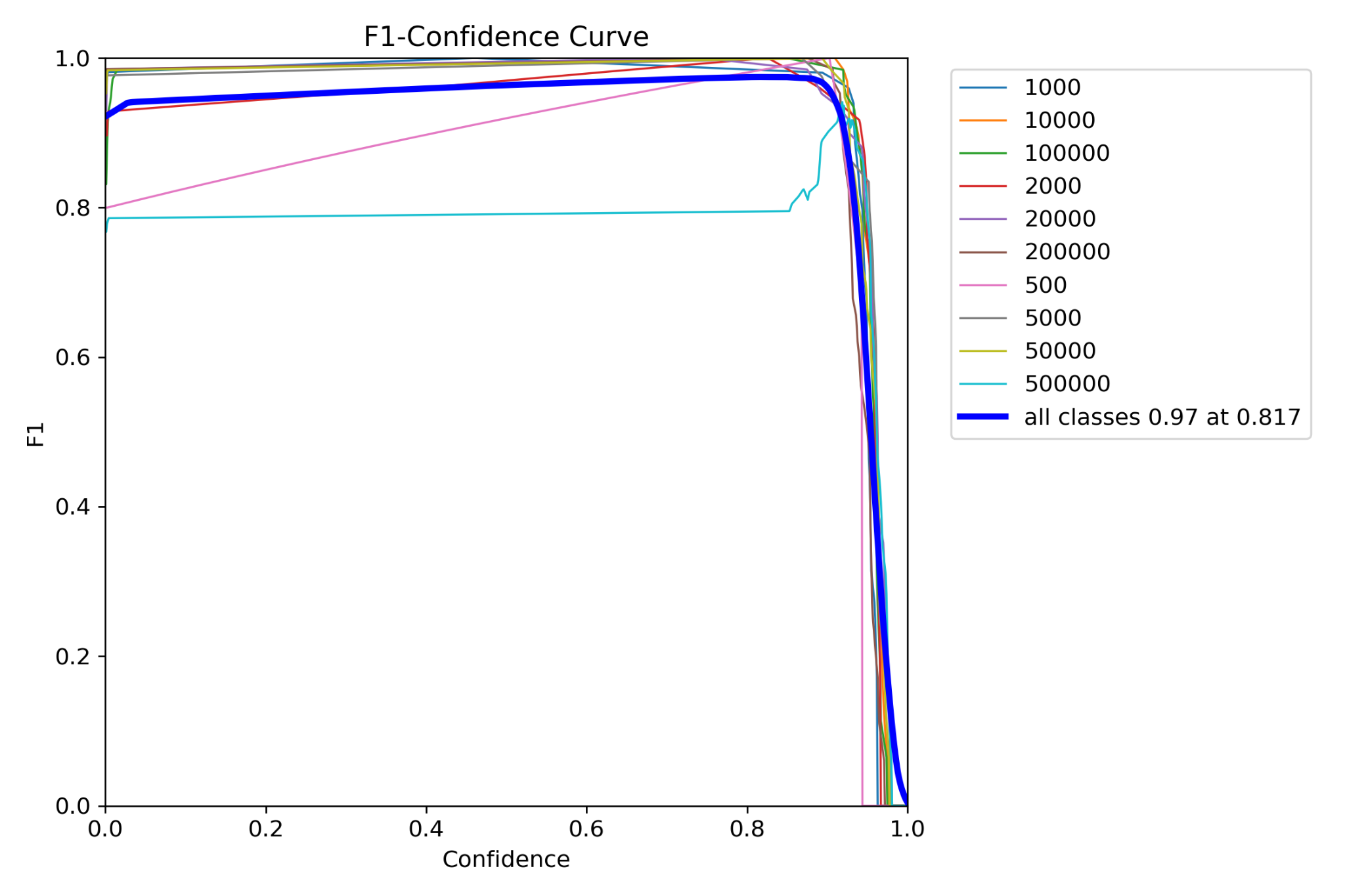
Thu được Confusion\_matrix chứa ma trận nhầm lẫn của mô hình trên tập kiểm tra. Ma trận này cho biết số lượng các trường hợp mà mô hình dự đoán đúng hoặc sai lớp của các đối tượng.

Hình 8 : Confusion\_matrix

### **Đường cong F1-Confidence**

Mô tả: Đường cong này cho thấy mối quan hệ giữa điểm F1 và độ tin cậy cho các lớp khác nhau.

Phân tích: - Đường màu xanh đậm đại diện cho hiệu suất tổng thể của tất cả các lớp, với điểm F1 là 0.97 ở mức độ tin cậy 0.817. - Các lớp riêng lẻ được biểu diễn bằng các đường mỏng hơn, mỗi đường cho thấy sự biến đổi nhẹ trong hiệu suất. - Hầu hết các lớp đều có điểm F1 cao (>0.8) ở mức độ tin cậy cao (>0.8), cho thấy hiệu suất tốt.



Hình 9 : F1-confidem Curve

### Báo cáo Hiệu suất Mô hình YOLOv8 Nhận dạng Tiền Việt Nam

1. Các Tổn Thất (Loss)

Box Loss: khoảng 0.1

Đo lường sự khác biệt giữa các hộp dự đoán và các hộp thực tế.

Objectness Loss: không cụ thể

Đánh giá sự chắc chắn về việc có một đối tượng trong hộp dự đoán.

Class Loss: khoảng 0.5

Đo lường sự khác biệt giữa các nhãn dự đoán và các nhãn thực tế.

DFL Loss (Distribution Focal Loss): khoảng 0.8

Đánh giá sự khác biệt về phân phối của nhãn.

2. Chỉ số Đánh giá

Precision: 1.00 tại độ tin cậy 0.956

Tỷ lệ các dự đoán đúng trên tổng số các dự đoán được đưa ra. Precision cao nghĩa là ít dương tính giả.

Recall: 1.00 tại độ tin cậy 0.000

Tỷ lệ các dự đoán đúng trên tổng số các đối tượng thực sự có trong hình ảnh. Recall cao nghĩa là ít âm tính giả.

F1 Score: 0.97 tại độ tin cậy 0.817

Trung bình điều hòa của Precision và Recall, cung cấp một đánh giá tổng thể về mô hình.

mAP (mean Average Precision): 0.993

Đánh giá tổng thể về độ chính xác của mô hình dựa trên cả precision và recall qua nhiều ngưỡng IOU. mAP cao cho thấy mô hình tốt.

IOU (Intersection Over Union): Cao

Đo lường mức độ trùng lặp giữa hộp dự đoán và hộp thực tế. IOU cao cho thấy các dự đoán chính xác về vị trí và kích thước của đối tượng.

3. Ma trận Nhầm lẫn (Confusion Matrix)

Độ chính xác (Accuracy): > 95% cho hầu hết các loại tiền.

Nhầm lẫn nhỏ: 2000 (7.14%), 500 (8.7%)

Ý nghĩa: Mô hình phân loại chính xác phần lớn các loại tiền.

4. Phân phối Nhãn và Bounding Box

Số lượng mẫu: Phân bố đồng đều giữa các loại tiền.

Ý nghĩa: Dữ liệu phân bố tốt giúp mô hình học hiệu quả.

5. Đường Cong Precision-Recall

mAP@0.5 IOU: 0.993

Ý nghĩa: Mô hình hoạt động rất tốt với độ chính xác cao.

6. Tổn thất Huấn luyện và Xác thực

Tổn thất (loss): Giảm dần qua các epoch

Ý nghĩa: Mô hình học hiệu quả, không overfitting.

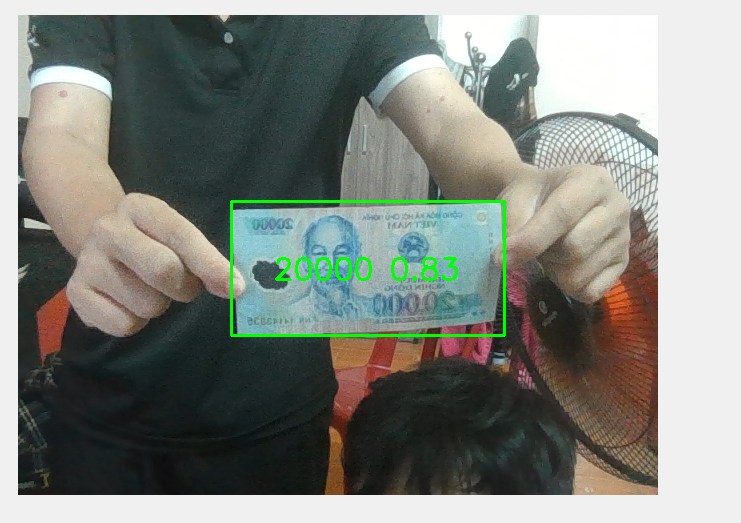
Kết luận

Mô hình YOLOv8 đạt hiệu suất cao với mAP@0.5 IOU 0.993 và F1 score 0.97, phân loại chính xác hầu hết các loại tiền với ít nhầm lẫn. Tổn thất huấn luyện giảm dần, chứng tỏ quá trình huấn luyện hiệu quả.

## Nhận dạng Tiền Qua Cam , Hình Ảnh

* 1. Nhận dạng Qua Cam

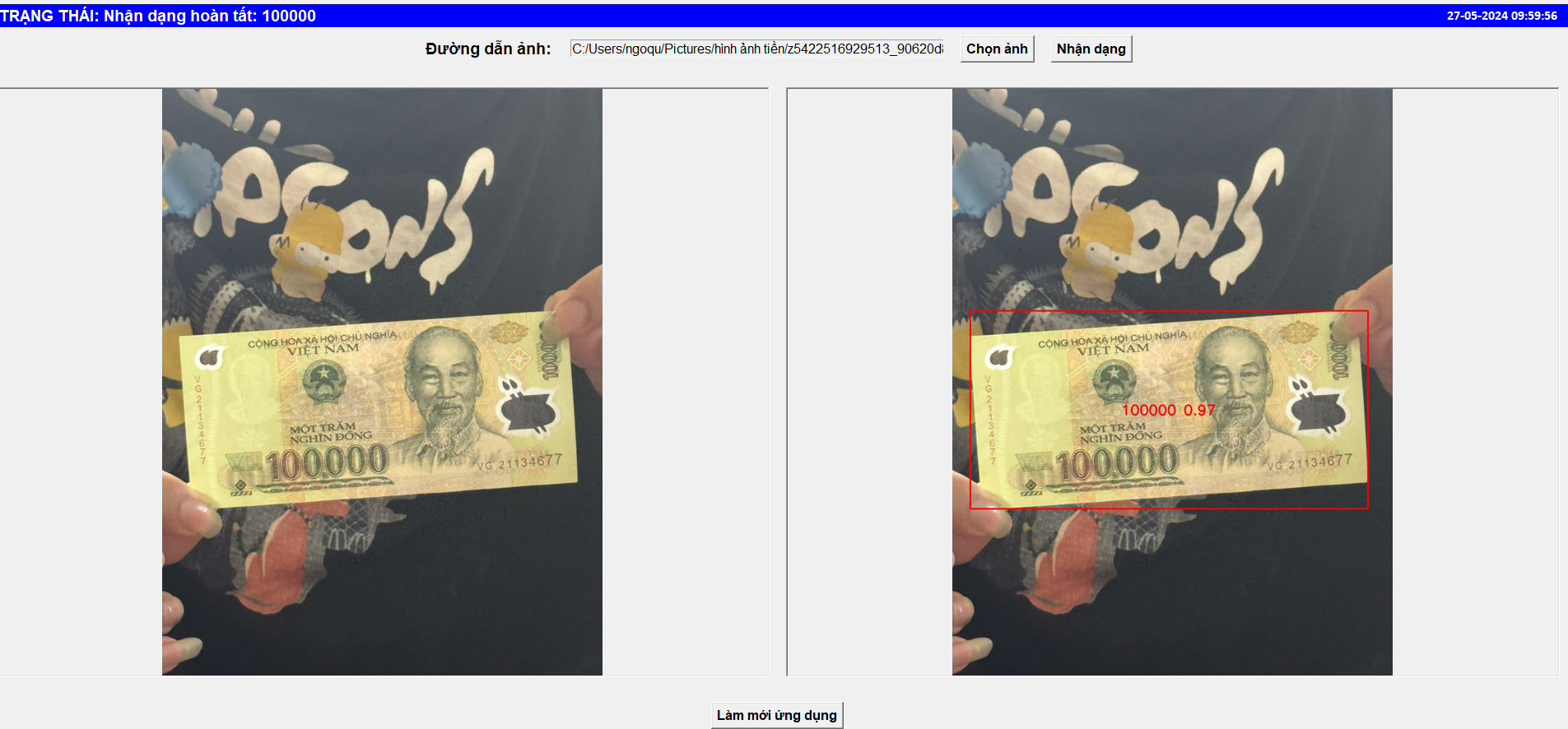
|  |
| --- |
| # Hàm nhận dạng tiền qua camera  def recognize\_from\_camera():  cap = cv2.VideoCapture(0)  while True:  ret, frame = cap.read()  if not ret:  break  results = recognize\_currency(frame)  frame\_with\_results = draw\_results(frame, results)  cv2.imshow('Nhận dạng Tiền Qua Camera', frame\_with\_results)  if cv2.waitKey(1) & 0xFF == ord('q'):  break  cap.release()  cv2.destroyAllWindows() |



Hình 10 : Hình ảnh Nhận dạng Qua Cam

* 1. Nhận dạng Qua Hình Ảnh

|  |
| --- |
| # Hàm để mở ảnh  def open\_image():  image\_path = filedialog.askopenfilename()  if image\_path:  image = cv2.imread(image\_path)  show\_image(image, img\_label\_left)  results = recognize\_currency(image)  show\_image\_with\_results(image, results)  # Hàm để hiển thị ảnh gốc  def show\_image(image, label):  image\_rgb = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)  pil\_image = Image.fromarray(image\_rgb)  tk\_image = ImageTk.PhotoImage(pil\_image)  label.config(image=tk\_image)  label.image = tk\_image  # Hàm để nhận dạng tiền tệ trong ảnh  def recognize\_currency(image):  results = model(image)  return results  # Hàm để vẽ kết quả lên ảnh  def draw\_results(image, results):  for result in results:  for box in result.boxes:  x1, y1, x2, y2 = map(int, box.xyxy[0])  label = f'{model.names[int(box.cls[0])]} {box.conf[0]:.2f}'  cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)  cv2.putText(image, label, (x1, y1 - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (36, 255, 12), 2)  return image  # Hàm để hiển thị ảnh với kết quả  def show\_image\_with\_results(image, results):  image\_with\_results = draw\_results(image.copy(), results)  show\_image(image\_with\_results, img\_label\_right) |

**

Hình 11 : Nhận Dạng Qua Hình Ảnh

# Phần 4 : Kết Luận

## Ưu Điểm

Tự động hóa dán nhãn : Sử dụng Roboflow để dán nhãn và tiền xử lý dữ liệu, tiết kiệm thời gian.

Mô hình mạnh mẽ:YOLOv8 đảm bảo độ chính xác cao trong việc nhận dạng đối tượng.

Dễ tích hợp:Giao diện Tkinter dễ sử dụng và tương tác.

Linh hoạt:Hỗ trợ nhận dạng từ hình ảnh tĩnh và camera trực tiếp.

Huấn luyện thuận tiện:Sử dụng Kaggle để huấn luyện mô hình dễ dàng.

## Nhược Điểm

Yêu cầu phần cứng cao: Cần GPU mạnh để đạt hiệu suất tốt.

Phụ thuộc vào dữ liệu: Chất lượng mô hình phụ thuộc vào dữ liệu huấn luyện.

Khả năng mở rộng hạn chế:Chỉ hỗ trợ nhận dạng các mệnh giá tiền Việt Nam.

Cần kết nối internet: Yêu cầu internet để tải dữ liệu và huấn luyện mô hình.

## Kết Luận

Ứng dụng nhận dạng tiền tệ sử dụng Roboflow và YOLOv8 mang lại nhiều ưu điểm vượt trội như độ chính xác cao, tích hợp dễ dàng và linh hoạt trong việc nhận dạng từ hình ảnh tĩnh và camera. Tuy nhiên, nó cũng có một số nhược điểm như yêu cầu phần cứng cao, phụ thuộc vào chất lượng dữ liệu và khả năng mở rộng hạn chế. Với các ưu điểm nổi bật, ứng dụng này rất hữu ích cho việc tự động hóa quy trình nhận dạng tiền tệ, đặc biệt là trong các môi trường yêu cầu nhận dạng nhanh chóng và chính xác. Việc cải thiện chất lượng dữ liệu và tối ưu hóa mô hình sẽ giúp nâng cao hiệu suất và độ chính xác của ứng dụng trong tương lai.

|  |
| --- |
| KẾT THÚC |