## A picture containing picture frame, rectangle, frame Description automatically generated

**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

**KHOA CƠ ĐIỆN – CÔNG TRÌNH**

---🙢🕮🙠---



**NHẬN DIỆN CÁC LOẠI QUẢ**

Môn học: Khai phá dữ liệu

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | ThS.Mai Hà An |
| Sinh viên thực hiện: | Đặng Phương Linh - 2174801040979 |
|  | Vũ Đức Anh - 2174801040609 |
|  | Lê Tuấn Đạt - 2174801040558 |
|  |  |

**HÀ NỘI, 2024**

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ phát triển như hiện nay, trí tuệ nhân tai được áp dụng rộng rãi và phát triển trong nhiều lĩnh vực. Trong ngành nghề nông nghiệp, nhiều nông dân còn đang dùng kinh nghiệm để phân biệt các loại quả gây sai sót về mặt kế quả. Điều này làm giảm chất lượng sản phẩm và gây nên sự thiếu hụt về mặt sản lượng. Việc ứng dụng AI để xây dựng một ứng dụng nhận diện loại quả có thể giúp người dùng phân biệt các loại quả một cách chính xác đồng thời cũng củng cố thêm kiến thức về loại quả đó một cách chính xác và đầy dủ nhất.

Đề tài “Nhận diện các loại quả” với mục đích phát triển một website để phân biệt các loại quả và cung cấp thông tin về loại quả đó cho người dùng. Đề tài này đối với chúng em không chỉ có mục đích là hoàn thiện sản phẩm cuối cùng mà còn là quá trình cả nhóm học hỏi và trải nghiệm trong việc huấn luyện AI có thể nhận diện. Nhóm em hy vọng rằng đề tài này sẽ cung cấp một nền tảng kiến thức để tiếp tục nghiên cứu và phát triển trong tương lai sắp tới

Với kiến thức còn hạn hẹp và thời gian có hạn, đề tài có thể sẽ có nhiều thiếu sót, mong nhận được lời đóng góp và chỉ dẫn từ thầy để đề tài của nhóm em được hoàn chỉnh hơn. Em xin chân thành cảm ơn

**MỤC LỤC**

[0](#_Toc194320119)

[1 Đặt vấn đề 3](#_Toc194320120)

[2 Thời gian thực hiện 3](#_Toc194320121)

[3 Lựa chọn công nghệ 4](#_Toc194320122)

[3.1 Deep Learning & Computer Vision 4](#_Toc194320123)

[3.1.1 YOLO (You Only Look Once) 4](#_Toc194320124)

[3.1.2 OpenCV 7](#_Toc194320125)

[3.1.3 PyTorch 8](#_Toc194320126)

[3.1.4 Roboflow 10](#_Toc194320127)

[3.2 Backend 11](#_Toc194320128)

[3.2.1 Python 11](#_Toc194320129)

[3.2.2 Flask 13](#_Toc194320130)

[3.3 Frontend 14](#_Toc194320131)

[3.3.1 HTML/CSS/ 14](#_Toc194320132)

[3.3.2 JavaScript & Fetch API 15](#_Toc194320133)

[4 Demo 15](#_Toc194320134)

[4.1 Luồng xử lý 15](#_Toc194320135)

[4.2 Quy trình phát triển 16](#_Toc194320136)

[4.2.1 Thu thập và xử lý dữ liệu 16](#_Toc194320137)

[5 Khó khăn và giải pháp 17](#_Toc194320138)

[6 Kết quả đạt được và hướng phát triển 17](#_Toc194320139)

[6.1 Kết quả đạt được 17](#_Toc194320140)

[6.2 Hướng phát triển 18](#_Toc194320141)

# Đặt vấn đề

Trong bối cảnh nền công nghiệp nông nghiệp đang ngày càng chuyển mình mạnh mẽ nhờ sự tiến bộ của khoa học và công nghệ, việc ứng dụng các phương pháp khai phá dữ liệu đã mở ra nhiều tiềm năng vô cùng lớn trong việc giải quyết các bài toán thực tế. Một trong những hướng nghiên cứu nổi bật là việc sử dụng trí tuệ nhân tạo để nhận diện và phân loại các loại quả. Đây là một loại nông sản có giá trị kinh tế và dinh dưỡng cao, thường đòi hỏi sự chính xác trong việc nhận diện để phục vụ cho các mục đích sản xuất, chế biến và xuất khẩu. Tuy nhiên, việc nhận diện quả bằng phương pháp truyền thống thường dựa vào kinh nghiệm cá nhân, dẫn đến sự chủ quan và không ổn định trong kết quả.

Đề tài “Nhận diện quả” được đề xuất nhằm khắc phục những hạn chế trên, đồng thời mang đến giải pháp hiệu quả, tiết kiệm thời gian và chi phí cho ngành nông nghiệp. Bằng việc áp dụng các công nghệ xử lý ảnh, phân tích dữ liệu và học máy, đề tài đặt mục tiêu xây dựng mô hình nhận diện chính xác dựa trên các đặc điểm đặc trưng của từng loại quả, từ hình dáng, kích thước đến màu sắc và cấu trúc bề mặt. Hệ thống nhận diện tự động không chỉ hỗ trợ nâng cao năng suất mà còn góp phần vào việc hiện đại hóa ngành nông nghiệp, đáp ứng nhu cầu ngày càng cao của thị trường trong nước và quốc tế.

Thời gian thực hiện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Ngày thực hiện | Nội dung công việc | Người thực hiện | Kết quả | Tiến độ |
| 1 | 20/2 | Định hướng phương án và xác định đề tài | Vũ Đức Anh  Lê Tuấn Đạt  Đặng Phương Linh | Xác định được đề tài | 2% |
| 2 | 21/2 | Lập kế hoạch và phân chia công việc | Vũ Đức Anh  Lê Tuấn Đạt  Đặng Phương Linh | Xác định được công việc của từng người trong nhóm. Bắt đầu thực hiện công việc | 3% |
| 3 | 22/2-25/2 | Chụp ảnh đối tượng | Vũ Đức Anh  Lê Tuấn Đạt  Đặng Phương Linh | Hình ảnh các loại quả: Cam, dâu tây, bơ, phật thủ, lê | 10% |
| 4 | 26/2-28/2 | Tìm hiểu về những công nghệ lập trình và ngôn ngữ lập trình hiện nay đối với đề tài đã chọn | Vũ Đức Anh  Lê Tuấn Đạt | Công nghệ: YOLO, OpenCV, PyTorch, Roboflow  Ngôn ngữ lập trình: Python, Flask, HTML/CSS, JavaScript & Fetch API | 15% |
| 5 | 1/3-5/3 | Tìm hiểu về công nghệ xử lý ảnh, phân tích dữ liệu và học máy | Vũ Đức Anh  Lê Tuấn Đạt | Tìm hiểu về Google Colab | 20% |
| 6 | 6/3-8/3 | Gán nhãn | Vũ Đức Anh  Lê Tuấn Đạt | Sử dụng GoogleColab để dán nhãn các loại quả | 40% |
| 7 | 9/3-15/3 | Tải model nhận diện và train dữ liệu | Vũ Đức Anh | Tải model nhận diện và dữ liệu | 70% |
| 8 | 16/3-20/3 | Xây dựng ứng dụng web | Vũ Đức Anh | Hoàn thành giao diện web | 75% |
| 9 | 21/3-22/3 | Kiểm thử và sửa lỗi | Vũ Đức Anh  Lê Tuấn Đạt | Hoàn thiện giao diện và nhận diện được các loại quả | 80% |
| 10 | 23/3-24/3 | Thêm mô tả cho từng loại quả | Lê Tuấn Đạt | Hoàn thiện mô tả của từng loại quả | 85% |
| 11 | 25/3-28/3 | Phát triển thêm nhận diện bằng máy ảnh . | Vũ Đức Anh  Lê Tuấn Đạt | Hoàn thiện nhận diện bằng máy ảnh | 90% |
| 12 | 29/3 | Làm báo cáo và powerpoint. | Đặng Phương Linh | Hoàn thành đề tài | 100% |

# Lựa chọn công nghệ

## Deep Learning & Computer Vision

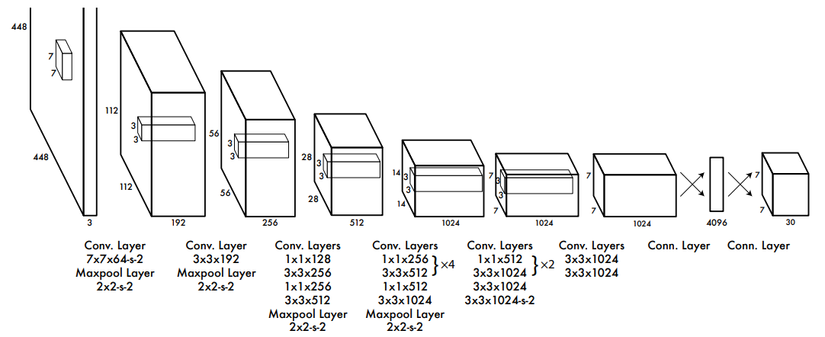
### YOLO (You Only Look Once)

#### Khái niệm

YOLO (You Only Look Once) là một thuật toán phát hiện đối tượng trong thời gian thực, nổi tiếng với tốc độ xử lý nhanh và độ chính xác cao.

#### Cách hoạt động

YOLO hoạt động dựa trên nguyên tắc xử lý toàn bộ hình ảnh đầu vào chỉ trong một lần duy nhất, từ đó đưa ra dự đoán về vị trí và lớp của các đối tượng trong ảnh.



*Cách hoạt động của YOLO*

* *Bước 1: Chia lưới hình ảnh*
* YOLO chia hình ảnh đầu vào thành một lưới các ô có kích thước SxS.
* Mỗi ô trong lưới này chịu trách nhiệm dự đoán các hộp giới hạn và xác suất lớp đối tượng nếu tâm của đối tượng nằm trong ô đó.
* *Bước 2: Dự đoán hộp giới hạn và lớp đối tượng*
* Mỗi ô lưới dự đoán B hộp giới hạn, mỗi hộp giới hạn được mô tả bằng 5 thuộc tính: x, y, w, h, confidence.
* Mỗi ô lưới dự đoán chính xác xác suất lớp đối tượng.
* *Bước 3: Mạng nơ-ron tích chập (CNN)*
* YOLO sử dụng một mạng nơ-ron tích chập (CNN) duy nhất để trích xuất các đặc trưng từ hình ảnh đầu vào và đưa ra các dự đoán.
* Mạng CNN này có nhiệm vụ phân tích toàn bộ hình ảnh một cách đồng thời, cho phép YOLO xử lý nhanh chóng.
* *Bước 4: Lọc và loại bỏ các hộp giới hạn dư thừa*
* Sau khi mạng nơ-ron đưa ra các dự đoán, YOLO sử dụng kỹ thuật NMS (Non-Maximum Suppression) để lọc và loại bỏ các hộp giới hạn dư thừa.
* *Bước 5: Kết quả*
* YOLO đưa ra kết quả là các hộp giới hạn xung quanh các đối tượng được phát hiện, cùng với xác suất lớp tương ứng.

#### Ưu điểm

* *Tốc độ xử lý nhanh*.
* YOLO được thiết kế để xử lý hình ảnh trong thời gian thực, đây là ưu điểm lớn nhất của nó.
* Phù hợp cho các ứng dụng đòi hỏi phản ứng nhanh chóng, ở đây là nhận diện.
* *Hiệu quả cao:*
* Đạt được độ chính xác cao trong việc nhận diện đối tượng.
* Có khả năng nhận diện các đối tượng với nhiều kích thước và hình dạng khác nhau.
* *Tính linh hoạt*
* Có thể áp dụng trong nhiều lĩnh vực nhận diện khác nhau, từ nhận diện khuôn mặt, nhận diện biển số xe, đến nhận diện vật thể trong dây chuyền sản xuất.
* YOLO cũng có thể hoạt động trên nhiều loại phần cứng, từ GPU mạnh mẽ đến các thiết bị nhúng.

#### Nhược điểm

* *Khả năng nhận diện đối tượng nhỏ:*
* Gặp khó khăn trong việc nhận diện các đối tượng nhỏ, đặc biệt là khi chúng xuất hiện dày đặc.
* Hạn chế trong các ứng dụng nhận diện đối tượng ở khoảng cách xa hoặc trong các cảnh có nhiều chi tiết nhỏ.
* *Độ chính xác tương đối*
* Độ chính xác thấp hơn so với các thuật toán nhận diện hai giai đoạn trong một số trường hợp nhất định.
* *Khả năng xử lý các đối tượng có hình dạng bất thường*
* Đôi khi gặp khó khăn trong việc nhận diện các vật thể có hình dạng quá khác biệt so với những hình ảnh được huấn luyện.
* *Yêu cầu tài nguyên tính toán*
* Để đạt được hiệu suất tối ưu, YOLO đòi hỏi tài nguyên tính toán đáng kể, đặc biệt là khi xử lý video độ phân giải cao.
* Hạn chế triển khai YOLO trên các thiết bị có tài nguyên hạn chế.

#### Phiên bản YOLOv8

Đây là phiên bản mới nhất của dòng mô hình YOLO, mang đến nhiều cải tiến so với các phiên bản trước.

* *Kiến trúc mới và cải tiền*
* Sử dụng các kiến trúc xương sống và cổ hiện đại, giúp trích xuất các đặc trưng từ hình ảnh hiệu quả hơn.
* YOLOv8 loại bỏ nhu cầu về điểm neo, giúp đơn giản hóa quá trình huấn luyện và cải thiện độ chính xác.
* Đạt được sự cân bằng tốt nhất giữa độ chính xác và tốc độ, phù hợp cho nhiều ứng dụng thời gian thực.
* *Tính linh hoạt và khả năng mở rộng*
* Cung cấp nhiều mô hình được đào tạo sẵn, giúp người dùng dễ dàng lựa chọn mô hình phù hợp với nhu cầu.
* Hỗ trợ phát hiện đối tượng, phân đoạn hình ảnh và phân loại.

#### So sánh với công nghệ khác

|  |  |  |
| --- | --- | --- |
| Nội dung | YOLOv8 | TensorFlow |
| Mục đích sử dụng | Phát hiện và định vị các quả trong hình ảnh, xác định vị trí và phân loại chúng | Xây dựng các mô hình phân loại ảnh để phân biệt các loại quả khác nhau  Kết hợp với các mô hình phát hiện đối tượng để tạo ra một hệ thống nhận diện quả toàn diện |
| Ưu điểm | Đạt độ chính xác cao trong việc nhận diện các đối tượng, kể cả các quả có kích thước nhỏ hoặc bị che khuất  Cung cấp giao diện đơn giản và dễ sử dụng, dễ dàng huấn luyện và triển khai mô hình | Xây dựng mô hình phức tạp và tùy chính cho nhiều tác vụ khác nhau  Có thể triển khai trên nhiều nền tảng và thiết bị khác nhau |
| Nhược điểm | Không linh hoạt như TensorFlow trong việc xử lý các tác vụ khác ngoài phát hiện đối tượng  Có thể gặp khó khăn trong việc phân loại các loại quả có hình dạng và màu sắc tương tự nếu dữ liệu huấn luyện không đủ tốt | Tốc độ xử lý chậm hơn  Yêu cầu kiến thức về học máy và học sâu |

### OpenCV

#### Khái niệm

Là một thư viện phần mềm mã nguồn mở mạnh mẽ và phổ biến trong lĩnh vực thị giác máy tính, học máy và xử lý hình ảnh.

#### Tính năng chính

* *Xử lý ảnh cơ bản*
* Đọc, ghi và hiển thị ảnh.
* Các thao tác xử lý ảnh như lọc, làm mờ, chuyển đổi màu sắc, biến đổi hình học.
* Phân tích histogram, xử lý độ sáng và độ tương phản.
* *Phân tích và nhận dạng đối tượng*
* Phát hiện các đối tượng.
* Nhận dạng đối tượng bằng thuật toán học máy.
* *Học máy*
* Cung cấp các thuật toán học máy cơ bản .
* Tích hợp với các thư viện học máy khác.

#### Ưu, nhược điểm

* *Ưu điểm*
* Cung cấp một loạt các thuật toán và công cụ để xử lý ảnh.
* Tối ưu hóa để xử lý ảnh một cách hiệu quả, cho phép xây dựng các hệ thống nhận diện hoạt động trong thời gian thực.
* Có thể tích hợp với các thưu viện học máy khác, cho phép kết hợp các phương pháp học sâu để cải thiện độ chính xác .
* *Nhược điểm*
* Độ chính xác phụ thuộc vào điều kiện sánh sáng và môi trường.
* Khó khăn trong việc nhận diện các loại quả có hình dạng và màu sắc tương tự.
* Hiệu xuất bị hạn chế trong các mô hình tài nguyên thấp.

### PyTorch

#### Khái niệm

Là một thư viện học máy mã nguồn mở dựa trên Torch, được phát triển bởi Facebook AI Research. Nó được sử dụng cho các ứng dụng như xử lý ngôn ngữ tự nhiên và thị giác máy tính.

#### Đặc điểm

* *Đồ thị tính toán động*

Khác với các thư viện khác sử dụng đồ thị tính toán tĩnh, Pytorch cho phép thay đổi đồ thị tính toán trong quá trình chạy. Mang lại sự linh hoạt cao.

* *Tensor*

Là đơn vị dữ liệu cơ bản trong Pytorch tương tự như mảng NumPy. Tuy nhiên, tensor có thể được tăng tốc bằng GPU, giúp tăng tốc đáng kể quá trình tính toán.

* *Autograd*

Có hệ thống tính toán đạo hàm tự động mạnh mẽ, giúp dễ dàng tính toán gradient cho các mô hình mạng nơ-ron.

* *Module*

Cung cấp các moduel được xây dựng sẵn để dễ dàng tạo và quản lý các lớp mạng nơ-ron.

* *Tối ưu hóa*

Chứa nhiều thuật toán tối ưu hóa khác nhau, hỗ trợ tìm ra các tham số tốt nhất cho mô hình.

#### Ưu điểm

* *Tính linh hoạt và dễ gỡ lỗi*
* Sử dụng đồ thị tính toán động, cho phép thay đổi mô hình trong quá trình chạy.
* Hữu ích trong việc thử nghiệm các kiến trúc mạng nơ-ron khác nhau và gỡ lỗi.
* Giao diện trực quan và cú pháp gần gũi cới Python giúp quá trình phát triển và gỡ lỗi trở nên dễ dàng hơn.
* *Hiệu suất cao*
* Tận dụng khả năng tăng tốc của GPU, giúp tăng tốc quá trình huấn luyện và suy luận của mô hình.
* Khả năng tối ưu hóa bộ nhớ và linh hoạt giúp PyTorch hoạt động hiệu quả trên các thiết bị có tài nguyên hạn chế.
* *Phù hợp cho nghiên cứu và phát triển*

Thiết kế linh hoạt, phù hợp với việc thử nghiệm các mô hình và các ý tưởng nghiên cứu.

#### Nhược điểm

* *Triển khai mô hình*

Khó khăn trong việc đưa mô hình nhận diện vào các ứng dụng thực tế.

* *Khả năng mở rộng*

Hạn chế trong việc triển khai và tối ưu các mô hình trên quy mô lớn.

### Roboflow

#### Khái niệm

Là một nền tảng trực tuyến giúp đơn giản hóa quá trình xây dựng các mô hình thị giác máy tính. Cung cấp một bộ công cụ toàn diện để thu thập, chú thích, tiền xử lý và quản lý tập dữ liệu hình ảnh, giúp người dùng dễ dàng tạo ra các mô hình nhận diện đối tượng chính xác.

#### Tính năng chính

* *Thu thập dữ liệu*

Cho phép người dùng tải lên hình ảnh và video từ nhiều nguồn khác nhau, bao gồm máy tính, thiết nọ di động và camera IP.

* *Chú thích dữ liệu*
* Cung cấp các công cụ chú thích trực quan để gán nhãn đối tượng trong hình ảnh.
* Hỗ trợ nhiều loại chú thích khác nhau, gồm hộp giới hạn, đa giác, phân đoạn.
* *Tiền xử lý dữ liệu*

Cung cấp các công cụ tiền xử lý mạnh mẽ để tăng cường và biến đổi dữ liệu hình ảnh.

Cải thiện độ chính xác của mô hình và giảm thiểu sự thiên vị.

* *Quản lý tệp dữ liệu*
* Cho phép người dùng quản lý tập dữ liệu của họ một cách hiệu quả.
* Cung cấp các công cụ để phân tích dữ liệu, theo dõi tiến độ và xuất dữ liệu ở nhiều định dạng khác nhau.
* *Tích hợp mô hình*
* Tích hợp với nhiều thư viện và khung công tác học sâu phổ biến bao gồm YOLO và PyTorch.
* Dễ dàng huấn luyện và triển khai các mô hình.

#### Ưu điểm

* *Dễ sử dụng*

Có giao diện người dùng trực quan và dễ sử dụng, ngay cả đối với người dùng không có kinh nghiệm về thị giác máy tính.

* *Tiết kiệm thời gian:*

Tự động hóa nhiều bước trong quy trình xây dựng mô hình, giúp người dùng tiết kiệm thời gian và công sức.

* *Tăng độ chính xác của mô hình*

Cung cấp các công cụ mạnh mẽ để tăng cường và biến đổi dữ liệu, giúp cải thiện độ chính xác của mô hình.

* *Tích hợp tốt với YOLO*
* Hỗ trợ chuyển đổi dữ liệu hình ảnh sang các định dạng mà các mô hình YOLO yêu cầu, giúp dễ dàng huấn luyện mô hình YOLO trên dữ liệu đã được chuẩn bị.
* Cung cấp các hướng dẫn và công cụ để tích hợp các mô hình YOL với dữ liệu của bạn.
* Có thể tự động hóa nhiều bước trong quy trình làm việc của YOLO, bao gồm việc gắn nhãn dữ liệu, chuyển đổi dữ liệu và tối ưu hóa quá trình huấn luyện.

#### Nhược điểm

* *Phụ thuộc vào kết nối internet*

Cần kết nối internet ổn định để sử dụng.

* *Giới hạn tùy chỉnh*

Có những giới hạn về khả năng tùy chỉnh so với việc xử lý dữ liệu thủ công.

## Backend

### Python

Là một ngôn ngữ lập trình đa năng, mạnh mẽ và dễ học, được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả thị gáic máy tính avf nhận diện đối tượng.

#### Tính năng chính

* *Xử lý ảnh*
* Python có nhiều thư viện mạnh mẽ để xử lý ảnh chẳng hạn như OpenCV.
* Các thư ciện cung cấp công cụ để đọc, ghi, hiển thị, biến đổi và phân tích ảnh, giúp trích xuất đặc trưng cần thiết cho việc nhận diện quả.
* *Xây dựng và huấn luyện mô hình học máy*

Là ngôn ngữ phổ biến nhất trong lĩnh vực học máy, với các thư viện như PyTorch, TensorFlow.

Cung cấp các công cụ về thuật toán để xây dựng và huấn luyện các mô hình học máy, bao gồm cả các mô hình nhận diện đối tượng như YOLO.

* *Tự động hóa quá trình nhận diên*

Có thể được sử dụng để tự động hóa quá trình nhận diện ảnh, từ việc thu thập dữ liệu ảnh đến việc tiền xử lý duẽ liệu, huấn luyện mô hình và triển khai hệ thống nhận diện.

* *Giao diện người dùng*

Có các thư viện để xây dựng giao diện người dùng cho ứng dụng nhận diện quả.

#### Ưu điểm

* *Dễ học và sử dụng*

Có cú pháp rõ ràng, dễ học và dễ hiểu, giúp giảm thời gian phát triển và bảo trì mã nguồn.

* *Thư viện phong phú*
* Cung cấp nhiều thư viện mạnh mẽ cho lập trình backend, chẳng hạn như Flask.
* Đơn giản hóa việc xây dựng các API, xử lý dữ liệu và tương tác với cơ sở dữ liệu.
* Cung cấp các thư viện chuyên biệt cho thị giác máy tính và học máy như OpenCV, PyTorch hữu ích cho việc tích hợp các chức năng nhận diện quả vào backend.
* *Khả năng mở rộng và linh hoạt*
* Có thể được sử dụng để xây dựng các ưunsg dụng backend từ nhỏ đến lớn, phù hợp với nhiều quy mô dự án.
* Có khả năng tích hợp tốt với các công nghệ khác, chẳng hẹn như cơ sở dữ liệu, dịch vụ đám mây và các ngôn ngữ lập trình khác ..

#### Nhược điểm

* Chậm hơn so với các ngôn ngữ khác.
* Một số vấn đề về hiệu suất trong các ứng dụng yêu cầu quản lý bộ nhớ chặt chẽ.
* Hạn chế khả năng xử lý đa luồng trong một số trường hợp.

#### So sánh ngôn ngữ khác

|  |  |  |
| --- | --- | --- |
| So sánh | Python | Java |
| Hiệu suất | Có tốc độ thực thi chậm hơn  Phù hợp với các ứng dụng yêu cầu tốc độ phát triển nhanh và dễ bảo trì | Có hiệu suất cao hơn Python trong các tác vụ tính toán nặng  Tối ưu hóa hiệu suất và bộ nhớ hiệu quả  Phù hợp với các ứng dụng yêu cầu tốc độ xử lý nhanh và độ ổn định cao |
| Khả năng mở rộng | Có khả năng mở rộng nhưng gặp một số hạn chế so với java trong các ứng dụng quy mô lớn  Với các thư viện và framework giúp Python vẫn có thể xây dựng các ứng dụng backend có khả năng mở rộng | Có khả năng mở rộng tốt, phù hợp với các ứng dụng lớn và phức tạp  Dễ dàng mở rộng khi cần thiết  Có thể xử lý lượng lớn người dùng và dữ liệu một cách hiệu quả |
| Dễ sử dụng | Có cú pháo đơn giản và dễ đọc, giảm thời gian phát triển và bảo trì mã nguồn  Phù hợp với người mới bắt đầu và các dự án yêu cầu tốc độ phát triển nhanh | Có cú pháp phức tạp hơn Python, đòi hỏi người lập trình cần phải có kiến thức sâu về lập trình  Có tính ổn định cao và dễ bảo trì trong các dự án lớn |

### Flask

#### Khái niệm

Là một microframework web được viết bằng Python. Được thiết kế để đơn giản và dễ sử dụng, cho phép các nhà phát triển xây dựng các ứng dụng web một cách nhanh chóng và linh hoạt.

#### Ưu điểm

* *Tính linh hoạt và đơn giản*
* Cung cấp các công cụ cốt lõi cho phép tự do lựa chọn các thư viện và tiện ích mở rộng khác. Dễ dàng tùy chỉnh ứng dụng theo nhu cầu cụ thể của đề tài nhận diện.
* Cú pháp đơn giản và dễ hiểu giúp người dùng nhanh chóng xây dựng và triển khai ứng dụng .
* *Tích hợp tốt với các thư viện Python*
* Tích hợp tốt với các thư viện Python phổ biến trong lĩnh vực thị giác máy tính và học máy, chẳng hạn nhưu OpenCV…
* Dễ dàng tích hợp các chức năng nhận diện vào ứng dụng Flask.
* *Khả năng xây dựng API*
* Phù hợp để xây dựung các API RESful cho phép các ứng dụng khác nhau tương tác với hệ thống nhận diên.
* Dễ dàng chia sẻ và sử dụng lại các chức năng nhận diện trong nhiều ứng dụng khác nhau.

#### Nhược điểm

* *Khả năng mở rộng*
* Gặp khó khăn trong việc xây dựng các ứng dụng web quy mô lớn và phức tạp
* Thiếu các tính năng tích hợp sẵn.
* Không cung cấp nhiều tính năng tích hợp sẵn như ORM hoẵ hệ thống quản lý người dùng.
* Đòi hỏi phải tự xây dựng hoặc sử dụng các thư viện bên ngoài để thêm các tính năng khác.
* *Hiệu suất*

Trong các tác vụ nhận diện thời gian thữ, với lượng truy xuất lớn, Flask có thể gặp vấn đề về hiệu năng.

#### So sánh

|  |  |  |
| --- | --- | --- |
| So sánh | Flask | Django |
| Kiến trúc và quy mô ứng dụng | Phù hợp cho các ứng dụng nhỏ và trung bình hoặc các ứng dụng yêu cầu sự linh hoạt cao  Thích hợp cho việc xây dựng cáic API đơn giản và các ứng dụng web nhỏ gọn | Phù hợp cho các ứng dụng lớn avf phức tạp, haowjc các ứng dụng yêu cầu nhiều tính năng tích hợp sẵn  Thích hợp cho việc xây dựng các ứng dụng web phức tạp với nhiều chức năng và cơ sở dữ liệu |
| Tính linh hoạt và khả năng tùy chỉnh | Cung cấp sự linh họa cao, cho phép tự do lựa chọn các thư viện và công cụ phù hợp với nhu cầu | Thích hợp cho các yêu cầu sự tùy chỉnh cao và khả năng tích hợp với các thư viện bên ngoài  Thích hợp cho các dự án yêu cầu sự nhất quán và tuân thủ các quy ước |
| Hiệu suất | Có thể hiệu suất tốt hơn tỏng các ứng dụng nhỏ và đơn giản nhờ sự nhẹ nhàng và linh hoạt | Hiệu suất tốt hơn trong các ứng dụng lớn và phứuc tạp nhờ kiến trúc và các tính năng tối ưu hóa |

## Frontend

### HTML/CSS/

HTML và CSS đóng vai trò xây dựng giao diện người dùng cho ứng dụng. Cụ thể:

* *Hiển thị kết quả nhận diện*
* HTML được sử dụng để tạo cấu trúc cho trang web, bao gồm các phần tử.
* Hình ảnh quả đã được nhận diện
* Tên loại quả được nhận diện..
* CSS được sử dụng để định dạng và trình bày các phần tử này, giúp giao diện trở nên trực quan và dễ hiểu.
* *Tương tác với người dùng*
* HTML sử dụng để tạo form cho phép người dùng.
* Tải lên hình ảnh quả cần nhận diện.
* CSS sử dụng để tạo các nút bấm, thanh trượt và các thành phần tương tác khác, giúp người dùng dễ dàng thao tác với ứng dụng.
* *Thiết kế giao diện đẹp mắt và thân thiện*
* CSS được sử dụng để tạo các kiểu dáng, màu sắc và bố cục cho trang web, giúp giao diện trở nên đẹp mắt và thu hút người dùng.
* CSS cũng được sử dụng để tạo các hiệu ứng động giúp giao diện trở nên sống động và tương tác.

### JavaScript & Fetch API

#### JavaScript

* *Xử lý tương tác người dùng*
* Được sử dụng để xử lý các sự kiện tương tác của người dùng, khi tải lên hình ảnh cần nhận diện nhấn vào “nút nhận diện trái cây”.
* Sử dụng để cập nhật giao diện người dùng dựa trên kết quả nhận diện, hiển thị tên loại quả và thông tin chi tiết của loại quả đó.
* *Xử lý dữ liệu từ API*
* Sử dụng để xử lý dữ iệu nhận được từ API nhận diện quả.
* Phân tích dữ liệu JSON và hiển thị thông tin cần thiết trên giao diện người dùng.

#### Fetch API

* *Giao tiếp với API nhận diện*
* Sử dụng để gửi các yêu cần HTTP đến API nhận diện quả.
* Cho phép hình ảnh quả cần nhận diện và nhận kết quả từ API.
* *Xử lý dữ liệu từ API*
* Cho phép nhận dữ liệu từ API ở định dạng JSON.
* JavaScrpit có thể sử dụng Fatch API để gửi và nhận diện một cách linh hoạt và hiệu quả.
* *Cập nhật giao diện người dùng*
* Cập nhật giao diện người dùng dựa trên dữ liệu nhận được từ API.
* Hiển thị kết quả nhận diện một cách tự động cập nhật.

# Demo

## Luồng xử lý

|  |  |  |
| --- | --- | --- |
| **STT** | **Hoạt động** | **Mô tả** |
| 1 | Upload ảnh từ giao diện | Người dùng chọn ảnh trên máy tính hoặc chụp ảnh mới. |
| 2 | Gửi request đến Flask server | Ảnh được gửi đến backend qua Fetch API dưới dạng multipart/form-data. |
| 3 | Xử lý ảnh qua model YOLO | Model YOLO tải ảnh, resize về kích thước phù hợp, thực hiện nhận diện.  Xuất kết quả là tọa độ bounding box và nhãn dự đoán. |
| 4 | Trả về kết quả nhận diện | Flask server trả về JSON về phía client |
| 5 | Hiển thị kết quả và thông tin | Frontend vẽ bounding box lên ảnh, hiển thị nhãn và độ chính xác. |

## Quy trình phát triển

### Thu thập và xử lý dữ liệu

* *Thu thập ảnh trái cây:*
* Cần thu thập nhiều ảnh từ các góc chụp, điều kiện ánh sáng khác nhau.
* Dữ liệu có thể lấy từ nguồn có sẵn hoặc tự chụp.
* *Gán nhãn bằng Roboflow:*
* Gán nhãn thủ công từng ảnh hoặc dùng tool hỗ trợ tự động.
* Xuất dữ liệu về định dạng YOLO.
* *Data Augmentation:*
* Sử dụng các phương pháp như xoay, lật, thay đổi độ sáng để tăng dữ liệu.
* Huấn luyện model.
* *Sử dụng Google Colab:*

Chạy trên GPU miễn phí, tránh hạn chế về tài nguyên máy cá nhân.

* *Train YOLO model:*
* Chạy lệnh train.py để huấn luyện với dữ liệu đã gán nhãn.
* Điều chỉnh số epoch, batch size, optimizer để tối ưu kết quả.
* *Export file best.pt:*
* Sau khi đạt kết quả tốt, lưu lại mô hình để dùng trong Flask API.
* Phát triển ứng dụng.
* *Xây dựng Flask server:*

Viết API xử lý ảnh đầu vào và chạy mô hình nhận diện.

* *Thiết kế giao diện web:*

Xây dựng UI trực quan giúp người dùng dễ dàng upload ảnh và xem kết quả.

* *Tích hợp model vào ứng dụng:*

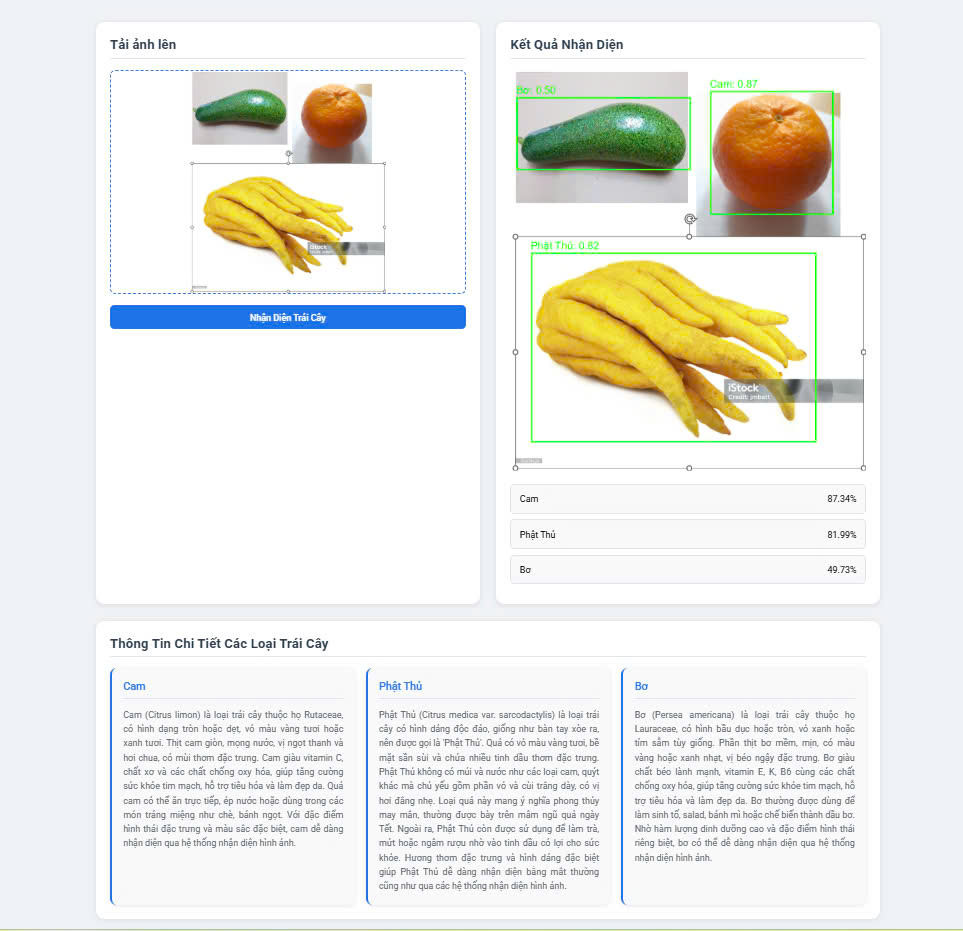
Tích hợp YOLO vào Flask, nhận request từ frontend và xử lý ảnh theo thời gian

# Khó khăn và giải pháp

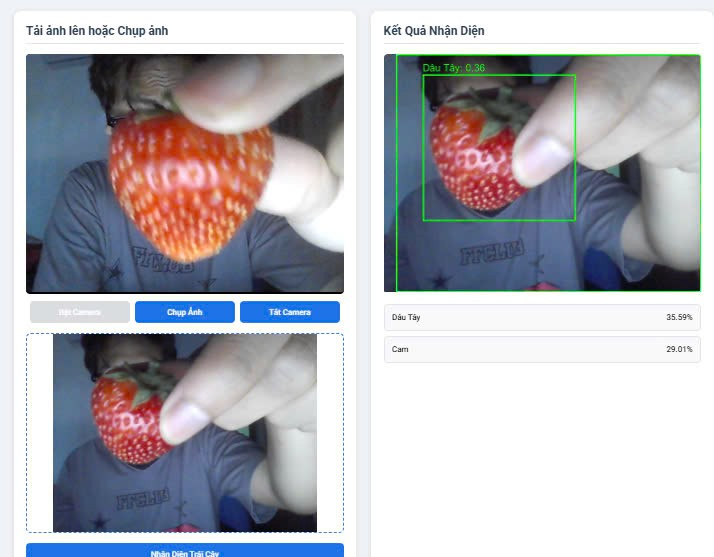
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nội dung | Khó khăn | Giải pháp |
| 1 | Thu thập dữ liệu | Cần nhiều ảnh từ nhiếu góc chụp, điều kiện khác nhau | Dùng augmentation để tăng lượng dữ liệu, lấy từ nhiều nguồn khác nhau. |
| 2 | Huấn luyện model | Giới hạn tài nguyên máy tính, training mất thời gian | Dùng Google Colab với GPU miễn phí, giảm số epoch nếu cần |
| 3 | Tích hợp hệ thống | Xử lý ảnh real-time yêu cầu tốc độ cao | Thử nghiệm thêm nhiều loại công nghệ như Faster R-CNN, Tensoflow để gia tăng độ chính xác  Giảm kích thước ảnh trước khi đưa vào model |

# Kết quả đạt được và hướng phát triển

## Kết quả đạt được



*Giao diện nhận diện quả bằng ảnh*



*Giao diện nhận diện quả bằng máy ảnh*

* Model nhận diện chính xác các loại trái cây: Độ chính xác cao, hoạt động tốt với ảnh đầu vào.
* Giao diện thân thiện, dễ sử dụng: Người dùng có thể dễ dàng tải ảnh, dùng camera và xem kết quả.
* Hiển thị thông tin chi tiết về trái cây: Hiển thị tên trái cây, độ tin cậy và bounding box trên ảnh.

## Hướng phát triển

* Mở rộng số lượng trái cây nhận diện: Thêm nhiều loại quả khác nhau vào dataset để tăng khả năng ứng dụng thực tế.
* Cải thiện độ chính xác của model: Huấn luyện với nhiều dữ liệu hơn, dùng model mạnh hơn hoặc fine-tune YOLO.
* Nâng cấp tính năng nhận diện qua camera: Dùng OpenCV để xử lý luồng video real-time. Nhận diện và hiển thị kết quả ngay khi camera bật.
* Phát triển ứng dụng mobile: Dùng Flutter/Kivy để tạo ứng dụng mobile có tính năng nhận diện trái cây.
* Triển khai YOLO trên edge device hoặc sử dụng API cloud.