

TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

---

# Phần Mềm Nhận Diện Hình Ảnh

---

GVHD: Từ Lăng Phiêu  
SV: Nguyễn Hữu Huân - 3120410189  
Nguyễn Minh Hùng - 3120410193

TP. HỒ CHÍ MINH, THÁNG 5/2024

## Mục lục

<b>CHƯƠNG 1. PHẦN GIỚI THIỆU . . . . .</b>	<b>2</b>
1.1 Giới thiệu về phần mềm Nhận diện hình ảnh, mục tiêu và phạm vi của phần mềm . . . . .	3
1.2 Tầm quan trọng của phần mềm Nhận diện hình ảnh . . . . .	4
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT . . . . .</b>	<b>5</b>
2.1 Mô tả các thư viện, công nghệ được sử dụng trong phần mềm Nhận diện hình ảnh . . . . .	6
2.2 Ưu điểm và nhược điểm của các thư viện, công nghệ . . . . .	10
<b>CHƯƠNG 3. THIẾT KẾ PHẦN MỀM . . . . .</b>	<b>11</b>
3.1 Mô tả chi tiết thiết kế phần mềm . . . . .	11
3.2 Ý nghĩa của các bảng dữ liệu và các trường trong bảng . . . . .	11
3.3 Trình bày cấu trúc . . . . .	11
<b>CHƯƠNG 4. VỀ DỰ ÁN PHẦN MỀM NHẬN DẠNG HÌNH ẢNH . . . . .</b>	<b>12</b>
4.1 Trình bày các tính năng được thiết kế và hình ảnh minh họa . . . . .	12
4.2 Mô tả cách thức cài đặt ứng dụng, môi trường chạy phần mềm . . . . .	19
<b>Kết luận . . . . .</b>	<b>20</b>
<b>Tài liệu . . . . .</b>	<b>21</b>

## CHƯƠNG 1. PHẦN GIỚI THIỆU

### 1.1 Giới thiệu về phần mềm Nhận diện hình ảnh, mục tiêu và phạm vi của phần mềm

#### Giới thiệu:

Phần mềm nhận diện hình ảnh là một phần mềm được thiết kế để tự động nhận diện, phân tích và hiểu các hình ảnh và video. Phần mềm này sử dụng các thuật toán máy học và trí tuệ nhân tạo để xác định các đối tượng, biểu cảm khuôn mặt, văn bản, định vị đối tượng, hay nhận dạng hoạt động trong hình ảnh và video.

Phần mềm nhận diện hình ảnh có nhiều ứng dụng trong các lĩnh vực khác nhau như xử lý hình ảnh y tế, an ninh, giao thông, tự động hóa công nghiệp, nhận dạng khuôn mặt, và nhiều ứng dụng khác. Điều này giúp tăng cường hiệu suất và đem lại giá trị đối với các tổ chức và cá nhân trong việc quản lý, xử lý và hiểu biết từ dữ liệu hình ảnh và video.

#### Mục tiêu xây dựng:

- **Mục tiêu xây dựng:** Mục tiêu xây dựng phần mềm nhận diện hình ảnh có thể đa dạng tùy thuộc vào ứng dụng cụ thể và nhu cầu của tổ chức hoặc cá nhân. Một trong những mục tiêu quan trọng nhất là tăng cường độ chính xác và độ tin cậy của hệ thống nhận diện hình ảnh. Điều này đảm bảo rằng hệ thống có khả năng nhận diện và phân loại hình ảnh một cách chính xác và hiệu quả.
- **Giao diện người dùng thân thiện:** Giao diện người dùng của phần mềm nhận diện ảnh được xây dựng đơn giản, dễ sử dụng, dễ nhìn, dễ hiểu và cực kì thân thiện với người dùng. Giao diện bao gồm: Tiêu đề, Input nhập tên định dạng file ảnh cần nhận dạng, hình ảnh được nhận diện.
- **Trải nghiệm mượt mà và cho ra hình ảnh chuẩn xác:** Một trong những ưu tiên hàng đầu của dự án là đảm bảo rằng người dùng có được trải nghiệm độ mượt mà, không bị chậm trễ quá lâu, cùng với kết quả là các hình ảnh phải được

nhận diện chính xác nhất có thể.

**Phần mềm mang lại nhiều lợi ích cho người dùng, bao gồm:**

- **Tăng cường hiệu suất làm việc:** Phần mềm nhận diện hình ảnh có thể tự động hóa các nhiệm vụ phức tạp như phân loại, đánh giá và xử lý hình ảnh. Điều này giúp tăng cường hiệu suất làm việc và giảm bớt thời gian và công sức cần thiết cho các quy trình thủ công.
- **Nâng cao chất lượng dịch vụ:** Trong lĩnh vực y tế, phần mềm nhận diện hình ảnh có thể hỗ trợ các bác sĩ trong việc chẩn đoán bệnh và lập kế hoạch điều trị bằng cách phân tích và hiểu biết từ hình ảnh y tế như cắt lớp MRI hay siêu âm.
- **Thao tác đơn giản:** Giao diện người dùng trực quan giúp người dùng dễ dàng sử dụng tính năng của phần mềm mà không cần hướng dẫn. Việc thiết kế giao diện dễ hiểu giúp người dùng thuận tiện trong việc mong muốn.
- **Tăng cường an ninh và giám sát:** Trong lĩnh vực an ninh, phần mềm nhận diện hình ảnh có thể được sử dụng để phát hiện và theo dõi các hoạt động đáng ngờ, nhận diện khuôn mặt, hoặc đếm lượng người đi lại trong một khu vực.
- **Cải thiện trải nghiệm khách hàng:** Trong các ngành như bán lẻ và dịch vụ khách hàng, phần mềm nhận diện hình ảnh có thể được sử dụng để phát hiện và nhận diện khách hàng, cung cấp trải nghiệm cá nhân hóa và tăng cường sự tiện lợi.

## 1.2 Tầm quan trọng của phần mềm Nhận diện hình ảnh

### Tầm quan trọng của phần mềm:

- **Tăng cường tính tự động hóa:** Phần mềm nhận diện hình ảnh giúp tăng cường tính tự động hóa trong nhiều lĩnh vực, từ công nghiệp đến y tế và giao thông. Việc tự động hóa quy trình nhận diện và phân tích hình ảnh giúp tiết kiệm thời gian và công sức của con người.
- **Cải thiện hiệu suất làm việc:** Sử dụng phần mềm nhận diện hình ảnh giúp cải thiện hiệu suất làm việc bằng cách nhanh chóng và chính xác nhận diện các đối tượng trong hình ảnh, từ việc phát hiện lỗi sản xuất đến phân tích dữ liệu y tế.
- **Hỗ trợ y tế và khoa học:** Trong lĩnh vực y tế và nghiên cứu khoa học, phần mềm nhận diện hình ảnh giúp trong việc chẩn đoán bệnh, nghiên cứu dược phẩm và phát hiện các biến đổi trong hình ảnh y tế.
- **Nâng cao trải nghiệm người dùng:** Trong các ứng dụng tiêu dùng, phần mềm nhận diện hình ảnh cung cấp trải nghiệm người dùng cá nhân hóa và thuận tiện, từ việc nhận diện sản phẩm để mua hàng trực tuyến đến việc tạo ảnh chân dung từ ảnh selfie.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 Mô tả các thư viện, công nghệ được sử dụng trong phần mềm Nhận diện hình ảnh

- **Numpy:** là thư viện mạnh mẽ và linh hoạt cho việc làm việc với dữ liệu đa chiều và thực hiện các phép toán số học trong Python, là công cụ không thể thiếu trong nghiên cứu khoa học, tính toán số, và xử lý dữ liệu - Mảng đa chiều, Toán số học, Truy cập và chỉnh sửa dữ liệu, Thống kê, Truy vấn dữ liệu.
- **Pandas:** là thư viện mã nguồn mở trong Python, được thiết kế để cung cấp các công cụ và cấu trúc dữ liệu mạnh mẽ cho việc xử lý và phân tích dữ liệu dạng bảng, giống như các bảng dữ liệu trong các cơ sở dữ liệu hoặc các bảng tính như Microsoft Excel - DataFrame và Series, Xử lý dữ liệu đa chiều, Đọc và ghi dữ liệu, Tích hợp với Numpy, Matplotlib và Seaborn.
- **Matplotlib:** là thư viện mã nguồn mở trong Python, được sử dụng để tạo ra các biểu đồ và đồ thị đồ họa có chất lượng cao. Đây là một công cụ mạnh mẽ và linh hoạt cho việc hiển thị dữ liệu một cách trực quan và dễ hiểu - Tạo biểu đồ và đồ thị, Tùy chỉnh đa dạng, Hỗ trợ nhiều định dạng, Tích hợp linh hoạt, Có nhiều giao diện người dùng.
- **Tensorflow:** là một thư viện mã nguồn mở phát triển bởi Google, được sử dụng rộng rãi trong lĩnh vực học máy và trí tuệ nhân tạo. Nó cung cấp một nền tảng mạnh mẽ cho việc xây dựng và triển khai các mô hình học máy, từ mạng nơ-ron nhân tạo cơ bản đến các mạng nơ-ron sâu phức tạp - Học máy và trí tuệ nhân tạo, Mạng Nơ-ron nhân tạo, Tích hợp hệ thống.
- **Streamlit:** là thư viện mã nguồn mở trong Python, được sử dụng để xây dựng ứng dụng web dựa trên dữ liệu và mô hình học máy một cách nhanh chóng và dễ dàng. Với Streamlit, người dùng có thể tạo ra các ứng dụng web tương tác mà không cần phải có kiến thức vững về phát triển web.
- **Keras:** là thư viện mã nguồn mở mạnh mẽ và phổ biến trong lĩnh vực học sâu (deep learning) của Python. Được phát triển bởi François Chollet, Keras được thiết kế với mục tiêu giúp việc xây dựng và thử nghiệm các mô hình học sâu trở

nên đơn giản và dễ tiếp cận.

## 2.2 Ưu điểm và nhược điểm của các thư viện, công nghệ

### Numpy:

#### *Ưu điểm:*

- **Hiệu suất cao:** NumPy được viết bằng C và Fortran, cho phép thực hiện các phép toán số học và mảng nhanh chóng và hiệu quả.
- **Mảng đa chiều:** NumPy cung cấp đối tượng mảng đa chiều (ndarray), cho phép lưu trữ và xử lý dữ liệu nhiều chiều một cách hiệu quả.
- **Tích hợp linh hoạt:** NumPy tích hợp tốt với nhiều thư viện khác trong hệ sinh thái Python như Pandas, Matplotlib, và SciPy, giúp tạo ra một hệ sinh thái mạnh mẽ cho tính toán số học và khoa học dữ liệu.
- **Tính năng mạnh mẽ:** NumPy cung cấp nhiều tính năng mạnh mẽ như truy cập và cắt lát mảng, phân tích thống kê, tích hợp với C/C++ và Fortran, và nhiều tính năng khác.

#### *Nhược điểm:*

- **Khó sử dụng với người mới:** NumPy có ngưỡng học khá cao đối với người mới bắt đầu với tính toán và lập trình, đặc biệt là đối với những người không có kinh nghiệm với lập trình hàm.
- **Thiếu tính tương tác:** NumPy không cung cấp các tính năng tương tác như đồ họa hoặc giao diện người dùng, điều này làm cho việc trực quan hóa dữ liệu hoặc tương tác với người dùng không phải lúc nào cũng thuận tiện.
- **Yêu cầu kiến thức vững về lập trình:** Để sử dụng NumPy hiệu quả, người dùng cần có kiến thức vững về lập trình và tính toán, đặc biệt là trong lĩnh vực tính toán số học và xử lý mảng.

## Pandas:

### *Ưu điểm:*

- **Xử lý dữ liệu dạng bảng:** Pandas cung cấp các cấu trúc dữ liệu như DataFrame và Series, cho phép xử lý dữ liệu dạng bảng một cách linh hoạt và mạnh mẽ.
- **Tính toán thống kê và phân tích dữ liệu:** Pandas cung cấp nhiều hàm và phương thức để thực hiện tính toán thống kê và phân tích dữ liệu, từ tổng hợp dữ liệu, phân tích chuỗi thời gian, đến tạo đồ thị và biểu đồ.
- **Tích hợp linh hoạt:** Pandas tích hợp tốt với các thư viện và công cụ khác trong hệ sinh thái Python như NumPy, Matplotlib, và Scikit-learn, giúp xử lý dữ liệu và thực hiện phân tích dữ liệu một cách thuận tiện.
- **Dễ sử dụng:** Pandas cung cấp cú pháp rõ ràng và dễ hiểu, giúp người dùng tạo ra và xử lý dữ liệu một cách nhanh chóng và dễ dàng.

### *Nhược điểm:*

- **Hiệu suất:** Pandas không phải là công cụ tối ưu nhất cho các tập dữ liệu lớn, đặc biệt là khi xử lý các phép toán phức tạp trên các tập dữ liệu có kích thước lớn.
- **Ngưỡng học cao:** Pandas có một ngưỡng học khá cao đối với người mới bắt đầu, đặc biệt là đối với những người không có kinh nghiệm với lập trình và xử lý dữ liệu.
- **Tương tác với người dùng:** Pandas không cung cấp các tính năng tương tác như đồ họa hoặc giao diện người dùng, điều này làm cho việc trực quan hóa dữ liệu hoặc tương tác với người dùng không phải lúc nào cũng thuận tiện.



## Matplotlib:

### *Ưu điểm:*

- **Đa dạng và linh hoạt:** Matplotlib cung cấp một loạt các loại biểu đồ và đồ thị, bao gồm biểu đồ đường, biểu đồ cột, biểu đồ hình tròn, biểu đồ scatter, biểu đồ histogram, vv. Điều này cho phép người dùng tạo ra các biểu đồ phù hợp với nhu cầu của họ.
- **Tùy chỉnh đa dạng:** Matplotlib cho phép tùy chỉnh các đặc điểm của biểu đồ, bao gồm màu sắc, kiểu đường, kích thước, nhãn, chú thích, vv. Người dùng có thể điều chỉnh mọi chi tiết của biểu đồ để phản ánh ý nghĩa của dữ liệu một cách chính xác.
- **Hỗ trợ nhiều định dạng đầu ra:** Matplotlib hỗ trợ xuất biểu đồ và đồ thị ra nhiều định dạng khác nhau như hình ảnh PNG, PDF, SVG, và các định dạng vectơ khác. Điều này giúp người dùng tích hợp biểu đồ vào các tài liệu và báo cáo một cách linh hoạt.

### *Nhược điểm:*

- **Cú pháp phức tạp:** Mặc dù Matplotlib mạnh mẽ, nhưng cú pháp của nó có thể phức tạp đối với người mới bắt đầu. Việc tùy chỉnh và điều chỉnh các đặc điểm của biểu đồ có thể đòi hỏi một lượng lớn mã và kiến thức về Matplotlib.
- **Thiếu tính tương tác:** Matplotlib không cung cấp các tính năng tương tác như đồ họa hoặc giao diện người dùng, làm cho việc tạo ra các biểu đồ tương tác hoặc phản hồi đến sự tương tác của người dùng trở nên khó khăn.
- **Khả năng mở rộng:** Trong một số trường hợp, Matplotlib có thể không linh hoạt đối với việc tạo ra các biểu đồ phức tạp hoặc đòi hỏi tính linh hoạt cao trong việc điều chỉnh các đặc điểm của biểu đồ.

## Keras:

### *Ưu điểm:*

- **API dễ hiểu:** Keras cung cấp một API thân thiện và dễ hiểu, giúp giảm bớt độ phức tạp trong việc xây dựng các mô hình học sâu.
- **Modular có thể mở rộng:** Các thành phần trong Keras được thiết kế dưới dạng các mô-đun độc lập, cho phép người dùng dễ dàng tạo, sửa đổi, và mở rộng các mô hình.
- **Tích hợp với Tensorflow:** Kể từ TensorFlow 2.0, Keras đã trở thành API mặc định cho việc xây dựng các mô hình học sâu trong TensorFlow, giúp tích hợp liền mạch giữa hai thư viện.
- **Hỗ trợ tính toán phân tán:** Keras kế thừa khả năng tính toán phân tán mạnh mẽ của TensorFlow, cho phép huấn luyện mô hình trên nhiều GPU và TPU.
- **Backend linh hoạt:** Keras hỗ trợ nhiều backend khác nhau như TensorFlow, Theano, và CNTK, mặc dù TensorFlow là backend chính được khuyến nghị.
- **Các mô hình Pre-trainer:** Keras cung cấp nhiều mô hình đã được huấn luyện trước trên các bộ dữ liệu lớn, giúp tiết kiệm thời gian và tài nguyên khi xây dựng các ứng dụng thực tế.

### *Nhược điểm:*

- **Hiệu suất kém hơn so với một số thư viện khác:** Mặc dù Keras dễ sử dụng, nhưng nó có thể không tối ưu về mặt hiệu suất so với việc sử dụng trực tiếp các API cấp thấp của TensorFlow hoặc các thư viện học sâu khác. Điều này có thể ảnh hưởng đến tốc độ huấn luyện và dự đoán trên các tập dữ liệu lớn và mô hình phức tạp.
- **Hạn chế trong một số tình huống phức tạp:** Keras, với API đơn giản và trực quan, có thể không cung cấp đủ tính linh hoạt cho các nhà nghiên cứu cần thực hiện các thí nghiệm phức tạp hoặc các mô hình tùy chỉnh cao.
- **Phụ thuộc vào TensorFlow:** Sự tích hợp chặt chẽ với TensorFlow có thể là một nhược điểm nếu bạn cần chuyển đổi hoặc sử dụng một framework học sâu



khác. Mặc dù Keras hỗ trợ nhiều backend khác, nhưng sự tích hợp tốt nhất vẫn là với TensorFlow.

## CHƯƠNG 3. THIẾT KẾ PHẦN MỀM

### 3.1 Mô tả chi tiết thiết kế phần mềm

- **Cấu trúc hệ thống:** phần mềm được thiết kế theo mô hình client-server, trong đó máy khách (client) giao tiếp với máy chủ (server).
- **Kiến trúc hệ thống:** phần mềm Streamlit để tạo giao diện người dùng đồ họa.
- **Giao diện người dùng:** sử dụng Streamlit để thiết kế giao diện người dùng thân thiện.
- **Cơ sở dữ liệu:** là các file ảnh được phân loại rõ ràng (có thể sưu tầm) để AI của phần mềm có thể được training và đưa ra nhiều dữ liệu chính xác.
- **Các thuật toán và mô hình được sử dụng:** Không có thuật toán hoặc mô hình cụ thể được sử dụng trong ứng dụng này.

### 3.2 Ý nghĩa của các bảng dữ liệu và các trường trong bảng.

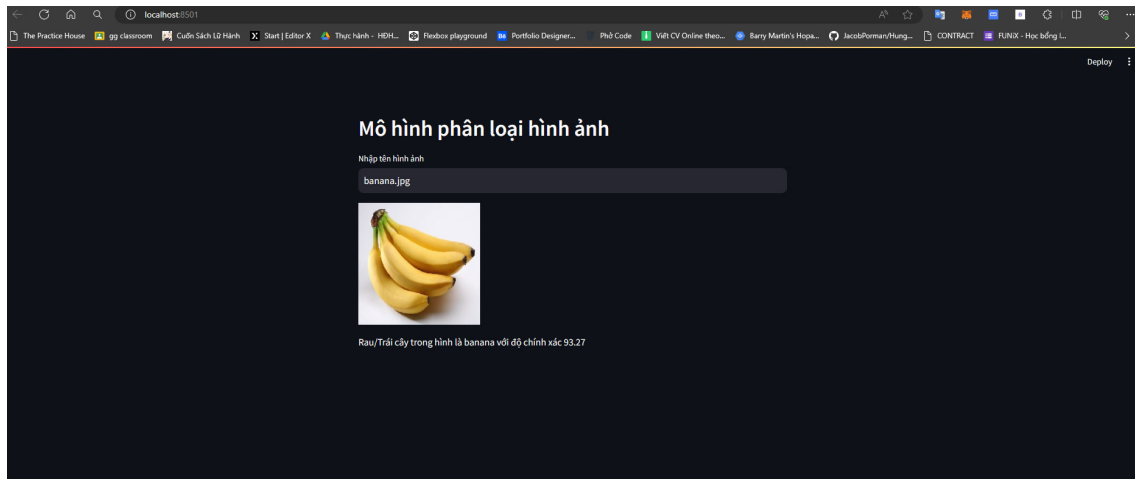
- Không có bảng dữ liệu hoặc trường dữ liệu nào được sử dụng trong ứng dụng này do không sử dụng cơ sở dữ liệu.

### 3.3 Trình bày cấu trúc

- **Mã nguồn:** Không có cơ sở dữ liệu được sử dụng trong ứng dụng này
- **Mô hình ứng dụng:** Mô hình của phần mềm bao gồm một giao diện người dùng được tạo bằng streamlit.
- **Tính năng được xây dựng:** Nhận diện các loại hình ảnh (được train cho AI).

## CHƯƠNG 4. VỀ DỰ ÁN PHẦN MỀM NHẬN DẠNG HÌNH ẢNH

### 4.1 Trình bày các tính năng được thiết kế và hình ảnh minh họa



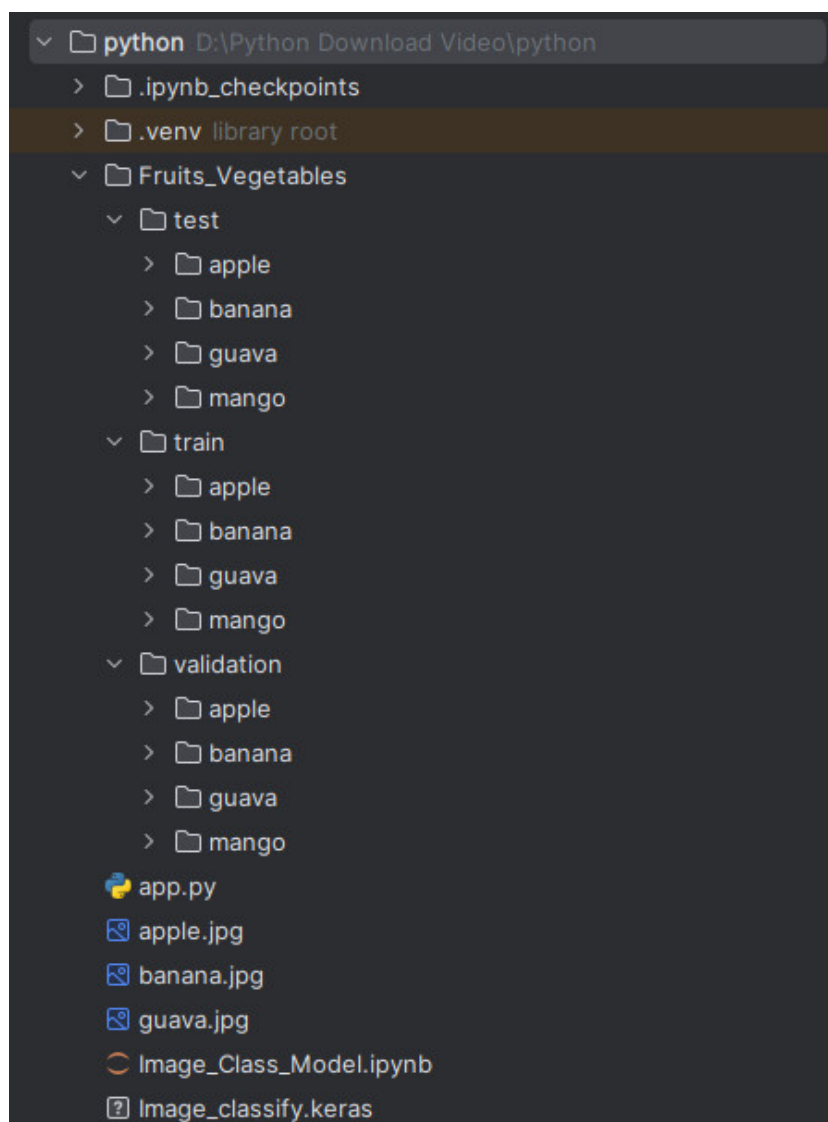
#### Giao diện chính

Gồm:

- + Tiêu đề phần mềm
- + Ô nhập file hình ảnh muốn nhận dạng
- + Hình ảnh được chọn
- + Dòng trạng thái nhận dạng hình ảnh của AI
- + Nút Deploy của Streamlit

Chức năng:

- + Nhận dạng hình ảnh
- + Deploy lên Streamlit



### Cấu trúc phần mềm

Gồm :

- + Các thư viện (.venv)
- + Folder chứa hình ảnh (Bản test, train và validation)
- + File app.py chứa source tạo giao diện
- + Các hình ảnh ví dụ
- + File xử lý Image Model

```
[223] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf

from tensorflow import keras
from keras import layers

[224] data_train_path = 'Fruits_Vegetables/train'
data_train_test = 'Fruits_Vegetables/test'
data_train_val = 'Fruits_Vegetables/validation'

[225] img_width = 180
img_height = 180

[226] data_train = tf.keras.utils.image_dataset_from_directory(
    data_train_path,
    shuffle=True,
    image_size = (img_width, img_height),
    batch_size = 32,
    validation_split = False
)

Found 4 files belonging to 4 classes.

[249] data_cat = data_train.class_names
```

Xử lý nhận diện

```
[263] data_cat

['apple', 'banana', 'guava', 'mango']

[250] data_val = tf.keras.utils.image_dataset_from_directory(
    data_train_val,
    shuffle=True,
    image_size = (img_width, img_height),
    batch_size = 32,
    validation_split = False
)

Found 4 files belonging to 4 classes.

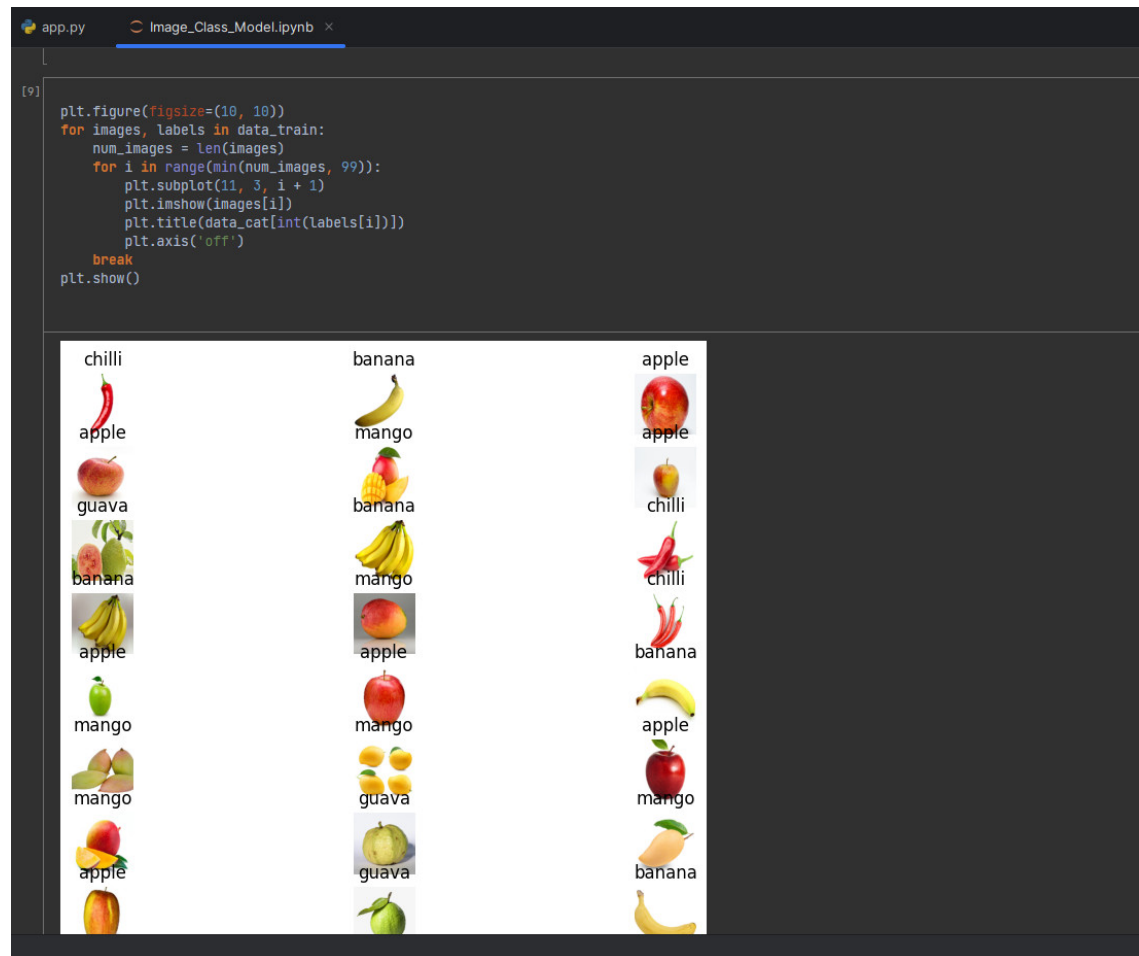
[251] data_val = tf.keras.utils.image_dataset_from_directory(
    data_train_test,
    shuffle=True,
    image_size = (img_width, img_height),
    batch_size = 32,
    validation_split = False
)

Found 4 files belonging to 4 classes.

[252] plt.figure(figsize=(10, 10))
for images, labels in data_train.take(1):
    num_images = min(9, len(images))
    for i in range(num_images):
        plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype('uint8'))
        plt.title(data_cat[labels[i].numpy()])
        plt.axis('off')
```

Xử lý nhận diện





Xử lý nhận diện



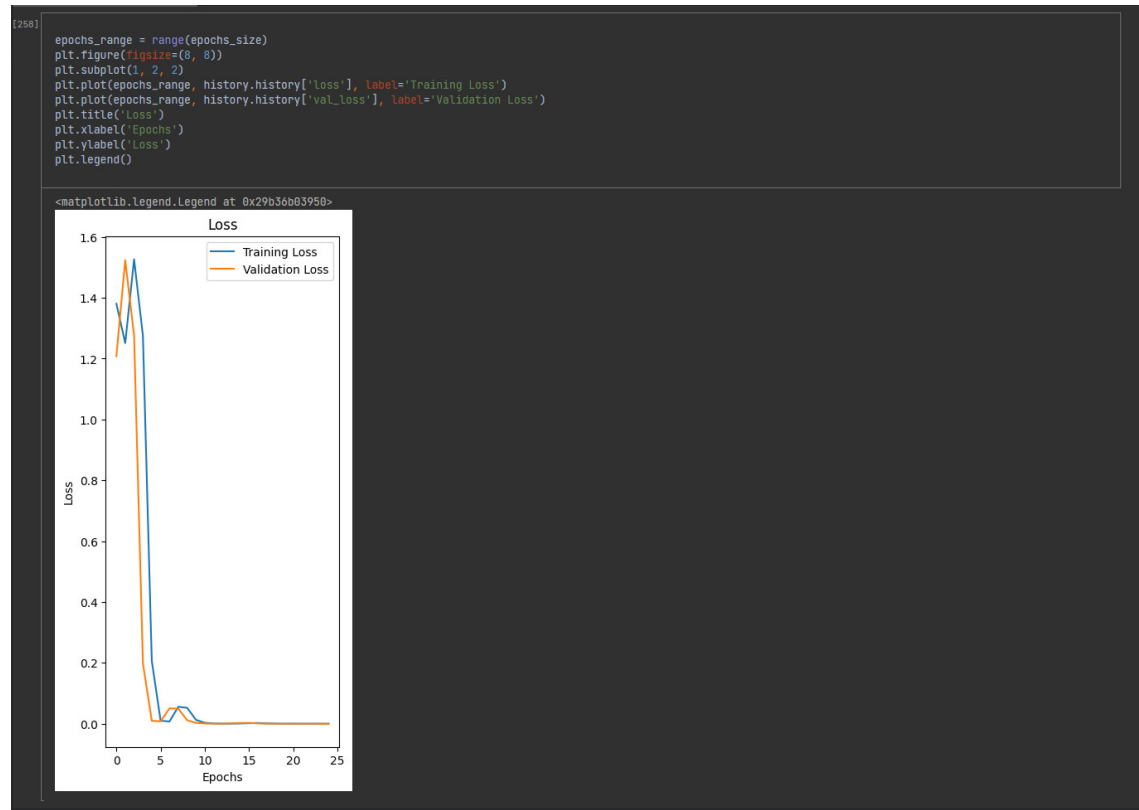
```
[255] model = Sequential([
    layers.Rescaling(1./255),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dropout(0.2),
    layers.Dense(128),
    layers.Dense(len(data_cat))
])

[256] model.compile(optimizer='adam', loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])

[257] epochs_size = 25
history = model.fit(data_train, validation_data=data_val, epochs=epochs_size, batch_size=32, verbose=1)

Epoch 1/25
1/1 ----- 1s 1s/step - accuracy: 0.2500 - loss: 1.3802 - val_accuracy: 0.5000 - val_loss: 1.2083
Epoch 2/25
1/1 ----- 0s 111ms/step - accuracy: 0.5000 - loss: 1.2509 - val_accuracy: 0.7500 - val_loss: 1.5233
Epoch 3/25
1/1 ----- 0s 102ms/step - accuracy: 0.7500 - loss: 1.5262 - val_accuracy: 0.7500 - val_loss: 1.2755
Epoch 4/25
1/1 ----- 0s 107ms/step - accuracy: 0.7500 - loss: 1.2752 - val_accuracy: 1.0000 - val_loss: 0.1981
Epoch 5/25
1/1 ----- 0s 101ms/step - accuracy: 1.0000 - loss: 0.2070 - val_accuracy: 1.0000 - val_loss: 0.0098
Epoch 6/25
1/1 ----- 0s 104ms/step - accuracy: 1.0000 - loss: 0.0100 - val_accuracy: 1.0000 - val_loss: 0.0079
Epoch 7/25
1/1 ----- 0s 115ms/step - accuracy: 1.0000 - loss: 0.0072 - val_accuracy: 1.0000 - val_loss: 0.0503
Epoch 8/25
1/1 ----- 0s 116ms/step - accuracy: 1.0000 - loss: 0.0556 - val_accuracy: 1.0000 - val_loss: 0.0494
Epoch 9/25
1/1 ----- 0s 119ms/step - accuracy: 1.0000 - loss: 0.0523 - val_accuracy: 1.0000 - val_loss: 0.0116
Epoch 10/25
1/1 ----- 0s 103ms/step - accuracy: 1.0000 - loss: 0.0126 - val_accuracy: 1.0000 - val_loss: 0.0027
Epoch 11/25
1/1 ----- 0s 118ms/step - accuracy: 1.0000 - loss: 0.0029 - val_accuracy: 1.0000 - val_loss: 7.1193e-04
Epoch 12/25
1/1 ----- 0s 116ms/step - accuracy: 1.0000 - loss: 7.4722e-04 - val_accuracy: 1.0000 - val_loss: 2.4528e-04
Epoch 13/25
1/1 ----- 0s 117ms/step - accuracy: 1.0000 - loss: 1.6365e-04 - val_accuracy: 1.0000 - val_loss: 3.3173e-04
```

## Xử lý nhận diện



Xử lý nhận diện

## 4.2 Mô tả cách thức cài đặt ứng dụng, môi trường chạy phần mềm

### Hướng dẫn cài đặt:

Bước 1 : Tải Source Code tại **Github**.

Bước 2 : Cài đặt môi trường Python theo **hướng dẫn** (Visual Studio Code hoặc Pycharm).

Bước 3 : Cài đặt các thư viện cần thiết :

- Numpy: `pip install numpy`
- Pandas: `pip install pandas`
- Matplotlib: `pip install matplotlib`
- Tensorflow : `pip install tensorflow`
- Streamlit: `pip install streamlit`
- Streamlit: `pip install keras`

Bước 4 : Chạy file `streamlit run app.py`

Bước 5 : Sử dụng ứng dụng

## KẾT LUẬN

Nhận dạng hình ảnh đã trở thành một phần quan trọng trong cuộc sống hàng ngày và các lĩnh vực công nghiệp khác nhau. Từ việc nhận dạng khuôn mặt trên các thiết bị di động đến ứng dụng trong y tế, an ninh, và tự động hóa công nghiệp, phần mềm nhận dạng hình ảnh đang mang lại những tiện ích và tiến bộ đáng kể.

Mặc dù đã đạt được những tiến bộ đáng kể, nhưng vẫn còn những thách thức cần phải vượt qua. Đối với những ứng dụng nhận diện hình ảnh trong thực tế, việc cải thiện độ chính xác và tốc độ xử lý là mục tiêu quan trọng. Cùng với đó, việc bảo vệ sự riêng tư và đảm bảo rằng công nghệ này được sử dụng một cách công bằng và đạo đức cũng là một vấn đề cần quan tâm.

Tuy nhiên, với sự tiến bộ của trí tuệ nhân tạo, học sâu, và các phương pháp máy học khác, chúng ta có hy vọng rằng phần mềm nhận dạng hình ảnh sẽ tiếp tục phát triển và mang lại những lợi ích to lớn cho xã hội. Đồng thời, việc nghiên cứu và phát triển công nghệ này cũng cần được thực hiện một cách có trách nhiệm, đảm bảo rằng nó sẽ góp phần vào việc giải quyết các vấn đề thực tế một cách tích cực và bền vững.



## Tài liệu

- [1] AIR Introduction “link: <https://vi.shaip.com/blog/what-is-ai-image-recognition-and-how-does-it-work/>”, lần truy cập cuối: 15/05/2024.
- [2] AI Model “link: <https://blogchiasekienthuc.com/thu-thuat-hay/tao-mot-model-ai-nhan-dang-hinh-anh.html/>”, lần truy cập cuối: 15/05/2024.