**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

——— 🙝⬩🙟 ———



**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**MÔN HỌC: PYTHON**

**Danh sách thành viên:**

1. **Nguyễn Nhật Nam**
2. **Trần Nhật Nam**
3. **Nguyễn Thành Nam**
4. **Hà Trọng Nghĩa**

**Bảng phân công và đánh giá**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ tên | Mssv | Nhiệm vụ | Điểm |
| Nguyễn Nhật Nam | 3122410248 | - Thiết kế xây dựng hệ thống bằng arduino (điều khiển đèn, cửa, quạt)  - Làm form thiết bị (thêm, xóa, sửa), hẹn giờ, danh sách chức năng có liên kết database  - Xây dựng trợ lí ảo để điều khiển các chức năng bằng giọng nói |  |
| Trần Nhật Nam |  |  |  |
| Nguyễn Thành Nam |  |  |  |
| Hà Trọng Nghĩa |  |  |  |

**LỜI MỞ ĐẦU**

Trong những năm gần đây, với sự bùng nổ của trí tuệ nhân tạo, các thiết bị IOT, các mô hình nhà ở thông minh ngày càng phổ biến và xuất hiện rộng rãi. Các mô hình smart home bên cạnh việc đem đến sự tiện nghi cho con người, nó còn cung cấp các tiện ích khác như bảo mật, an toàn,.. Vì thế nó đã và đang rất được mọi người ưa chuộng và hướng tới, là một chủ đề sẽ được khai thác rất nhiều trong tương lai. Chính vì những lí do đó, ở học kì này chúng em quyết định lựa chọn chủ đề thú vị này làm đề tài để nghiên cứu, để hiểu rõ hơn về các thiết bị IOT và các vấn đề liên quan đến máy học. Tuy mô hình vẫn đang ở quy mô rất nhỏ so với thực tế song những gì học được từ đề tài này rất hữu ích và thiết thực. Cảm ơn thầy đã tạo điều kiện cho tụi em có thể nghiên cứu một chủ đề tuy khá khó nhưng cũng không kém phần thú vị.

**CHƯƠNG 1. TỔNG QUAN VỀ CHƯƠNG TRÌNH**

**1.1 Giới thiệu chương trình:**

Python là ngôn ngữ lập trình bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ được sử dụng để tạo ra nhiều chwing trình khác nhau, không chuyên biệt cho bất kì vấn đề cụ thể nào. Với sự linh hoạt đó cùng với sự thân thiện với người mới bắt đầu, đã khiến nó trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay.

Chương trình “quản lí nhà ở thông minh” sử dụng ngôn ngữ lập trình python, giúp quản lí và nâng cao trải nghiệm người dùng với các thiết bị trong nhà. Các chức năng chính của chương trình bao gồm sử dụng giọng nói để điều khiển các thiết bị trong nhà như bật đèn, tắt đèn, đóng mở cửa, thiết đặt thời gian sử dụng thiết bị, thêm xóa sửa các thiết bị vào hệ thống, quản lí thông tin thiết bị và người dùng. Bên cạnh đó chương trình còn áp dụng công nghệ máy học để nhận diện khuôn mặt, tăng tính bảo mật cho gia đình. Các công nghệ sẽ được giới thiệu chi tiết ở những phần sau.

**1.2 Cơ sở lí thuyết:**

1.2.1 Một số khái niệm cơ bản

Python là ngôn ngữ lập trình bậc cao với các ưu điểm như câu lệnh ngắn gọn, dễ nhớ, dễ hiểu. Cấu trúc chương trình rõ ràng, dễ đọc, dễ viết hơn các ngôn ngữ lập trình khác, chạy được trên nhiều hệ điều hành khác nhau.

Python hỗ trợ chúng ta rất nhiều các thư viện để có thể làm việc với ngôn ngữ lập trình một cách dễ dàng.

1.2.2 Các thư viện sử dụng trong chương trình

Đối với IOT:

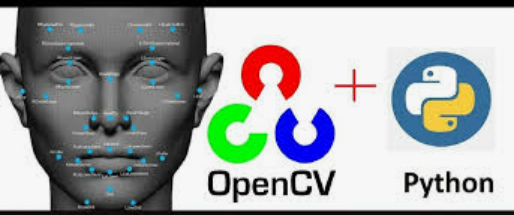
* Xuất phát từ hạn chế của IDE sử dụng cho arduino khi quản lí các thiết bị thêm vào, thư viện pyfirmata đã được lựa chọn để thay thế cho IDE truyền thống.
* PyFirmata là một thư viện Python mạnh mẽ và linh hoạt được sử dụng để giao tiếp với các bo mạch Arduino thông qua giao thức Firmata. Điều này cho phép bạn điều khiển các chức năng của bo mạch Arduino từ các chương trình Python của mình một cách dễ dàng và linh hoạt.
* Các tính năng và ưu điểm chính của PyFirmata bao gồm:

1. **Giao tiếp đơn giản**: PyFirmata cung cấp một giao diện đơn giản và dễ sử dụng để giao tiếp với bo mạch Arduino. Bằng cách sử dụng PyFirmata, bạn có thể gửi và nhận dữ liệu từ bo mạch Arduino một cách dễ dàng thông qua Python.
2. **Kiểm soát linh hoạt**: PyFirmata cho phép bạn kiểm soát các chân I/O trên bo mạch Arduino từ Python. Bạn có thể đọc giá trị từ các cảm biến kết nối với bo mạch hoặc ghi giá trị vào các chân đầu ra để điều khiển các thiết bị khác.
3. **Tương thích đa dạng:** PyFirmata hoạt động với nhiều loại bo mạch Arduino khác nhau và hỗ trợ các phiên bản Firmata khác nhau. Điều này cho phép bạn tương tác với các dòng bo mạch khác nhau mà không cần phải thay đổi mã nguồn của bạn
4. **Phát triển dự án linh hoạt**: PyFirmata là một công cụ mạnh mẽ cho việc phát triển các dự án nhúng và IoT (Internet of Things) bằng Python. Từ các dự án điều khiển robot đơn giản đến các hệ thống tự động hóa phức tạp, PyFirmata cung cấp một cơ sở linh hoạt để bạn xây dựng và thử nghiệm các ứng dụng của mình.

Đối với CNN:

**Các thư viện được dùng trong đề tài phục vụ cho CNN:**

1. Opencv-python: là một thư viện mà các nhà phát triển sử dụng để xử lý hình ảnh cho các ứng dụng thị giác máy tính. Thư viện này cung cấp nhiều hàm cho các tác vụ xử lý hình ảnh như đọc và ghi hình ảnh cùng lúc, xây dựng môi trường 3D từ môi trường 2D cũng như chụp và phân tích hình ảnh từ video.

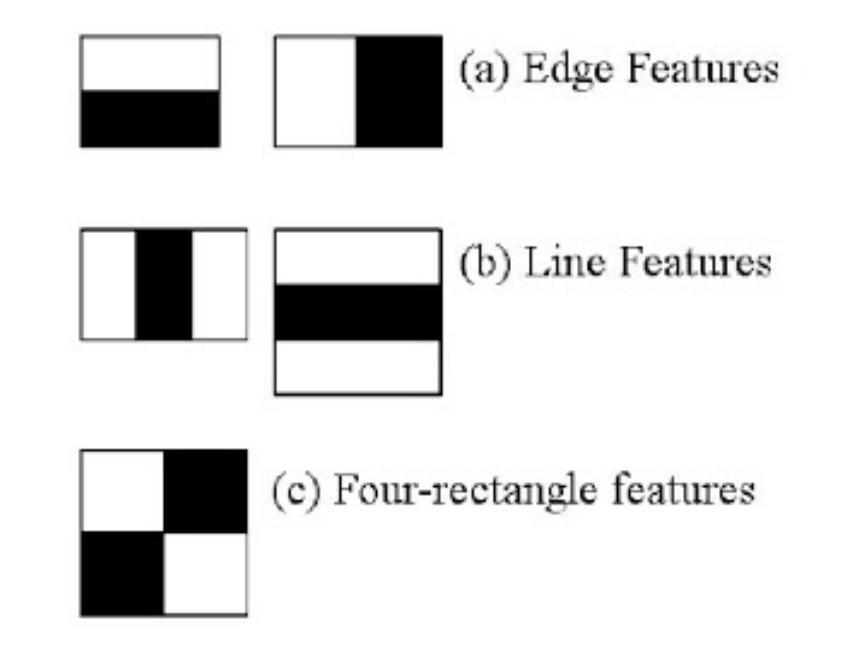


Ngoài ra trong đề tài còn sử dụng thêm một phương pháp phát hiện đối tượng đó là Haar Cascade được xây dựng bởi opencv:

Haar Cascade là một thuật toán được tạo ra dựa trên những tính năng đó để phát hiện đối tượng (có thể là khuôn mặt, mắt, tay, đồ vật,…) được đề xuất vào năm 2001 bởi Paul Viola và Michael Jones trong bài báo của họ với khẳng định “Phát hiện đối tượng một cách nhanh chóng bằng cách sử dụng tầng (Cascade) tăng cường các tính năng đơn giản”.

Là một lớp model có thể giúp chúng ta nhận diện khuôn mặt (Haar Cascade face detection) Haar Cascade sử dụng các tầng Haar và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (Cascade) và tạp thành một cỗ máy nhận diện khuôn mặt hoàn chỉnh.

Trong bài báo của tác giả Paul Viola và Michael Jones đã trình bày một phương pháp mới và nhanh hơn để xử lý hình ảnh và phát hiện khuôn mặt bằng cách sử dụng các đặc điểm hình chữ nhật như hình dưới đây. Các đặc điểm hình chữ nhật tương tự như nhân được sử dụng để phát hiện các đặc điểm khác nhau của khuôn mặt như mắt và các nốt như trong hình minh họa.



1. Keras: là một thư viện mã nguồn mở được sử dụng rộng rãi trong lĩnh vực deep learning (học sâu) và mạng nơ-ron. Nó được thiết kế để giúp các nhà phát triển xây dựng và thử nghiệm các mô hình học sâu một cách dễ dàng và linh hoạt.

Mục tiêu chính của Keras là cung cấp một API đơn giản, trực quan và dễ sử dụng, giúp người dùng tập trung vào việc xây dựng mô hình mà không cần lo lắng về chi tiết kỹ thuật phức tạp.

Một trong những điểm mạnh của Keras là khả năng tích hợp với các framework học sâu khác như TensorFlow và Theano, cho phép người dùng tận dụng sức mạnh của cả hai thế giới. Đây chính là điều giúp Keras trở thành một lựa chọn hàng đầu cho các dự án học sâu từ phân loại ảnh, dự đoán chuỗi thời gian, đến các ứng dụng trong lĩnh vực ngôn ngữ tự nhiên và nhiều lĩnh vực khác.

Các thư viện bổ sung:

1. Speech recognition:

Speech recognition, hay nhận dạng giọng nói, là quá trình tự động chuyển đổi giọng nói thành văn bản hoặc các lệnh điều khiển. Công nghệ này cho phép máy tính hiểu và xử lý các câu nói từ con người. Các ứng dụng của speech recognition rất đa dạng và phong phú, từ việc điều khiển các thiết bị điện tử thông qua lệnh giọng nói đến việc biến các bài thuyết trình thành văn bản viết.

Các hệ thống speech recognition thường bao gồm các bước chính sau:

Ghi âm: Dữ liệu âm thanh được ghi lại từ người nói bằng micro hoặc các thiết bị ghi âm khác.

Tiền xử lý: Dữ liệu âm thanh thường cần được xử lý trước khi được đưa vào bước nhận dạng để loại bỏ nhiễu và cải thiện chất lượng.

Nhận dạng: Trong bước này, một hệ thống máy tính phân tích và xác định các đặc điểm của giọng nói và chuyển đổi nó thành văn bản hoặc lệnh điều khiển.

Xử lý kết quả: Kết quả từ bước nhận dạng thường được xử lý thêm để cải thiện độ chính xác và thực thi các hành động phù hợp.

1. Tkinter:

Tkinter là thư viện chính để xây dựng giao diện cho chương trình, sử dụng những hàm có sẵn của Tkinter và thỏa sức sáng tạo.

Figma là ứng dụng hỗ trợ xây dưng giao diện kéo thả nhanh chóng.

PIL được sử dụng để xử lý ảnh trong giao diện .

Tkinter Designer là ứng dụng hỗ trợ convert từ file giao diện Figma trở thành file giao diện dạng python,ứng dụng hỗ trợ tách các thành phần trong Figma thành dạng ảnh và các Widget cơ bản trong Tkinter

## Giới thiệu sơ bộ về Tkinter

Tkinter là một thư viện Python được sử dụng để tạo giao diện người dùng đồ họa (GUI). Nó là một phần của thư viện tiêu chuẩn Python và được cài đặt sẵn trên hầu hết các hệ điều hành. Tkinter cung cấp một tập hợp các widget cơ bản như nút, nhãn, trường nhập, khung, thanh cuộn, menu, v.v., cho phép bạn xây dựng các ứng dụng GUI đa dạng.

**Ưu điểm của Tkinter:**

* **Dễ sử dụng:** Tkinter có cú pháp đơn giản và dễ học, phù hợp cho cả người mới bắt đầu lập trình.
* **Miễn phí và mã nguồn mở:** Tkinter là một phần của thư viện tiêu chuẩn Python, vì vậy nó hoàn toàn miễn phí và mã nguồn mở.
* **Đa nền tảng:** Tkinter có thể được sử dụng trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux.
* **Phổ biến:** Tkinter là một thư viện GUI phổ biến với cộng đồng người dùng Python lớn, do đó bạn có thể dễ dàng tìm kiếm tài liệu và trợ giúp khi cần thiết.

**Nhược điểm của Tkinter:**

* **Giao diện cơ bản:** Tkinter cung cấp các widget cơ bản, giao diện có thể không được bóng bẩy hoặc hiện đại như các thư viện GUI khác.
* **Yêu cầu kiến thức về Tk:** Để sử dụng Tkinter hiệu quả, bạn cần có kiến thức cơ bản về Tk, hệ thống GUI được sử dụng bởi Tkinter.
* **Hiệu suất:** Hiệu suất của Tkinter có thể không tốt bằng các thư viện GUI khác, đặc biệt là khi xử lý các ứng dụng GUI phức tạp.

## Giới thiệu sơ bộ về Figma

**Figma** là một ứng dụng thiết kế giao diện người dùng (UI) và tạo nguyên mẫu (prototype) hoạt động chủ yếu trên nền tảng web. Nó cung cấp cho người dùng một bộ công cụ mạnh mẽ để tạo ra các giao diện đẹp mắt, trực quan và tương tác cho các trang web, ứng dụng di động và các sản phẩm kỹ thuật số khác.

**Điểm nổi bật của Figma:**

* **Dễ sử dụng:** Figma có giao diện trực quan và dễ sử dụng, phù hợp cho cả người mới bắt đầu và nhà thiết kế chuyên nghiệp.
* **Miễn phí:** Figma có phiên bản miễn phí với đầy đủ các tính năng cơ bản để đáp ứng nhu cầu thiết kế của hầu hết người dùng.
* **Hợp tác:** Figma hỗ trợ cộng tác thời gian thực, cho phép nhiều người cùng làm việc trên một dự án thiết kế cùng lúc.
* **Plugin phong phú:** Figma có cộng đồng nhà phát triển plugin mạnh mẽ, cung cấp cho bạn nhiều plugin bổ sung để mở rộng chức năng và nâng cao hiệu quả công việc.
* **Tạo nguyên mẫu:** Figma cho phép bạn tạo ra các nguyên mẫu tương tác, giúp bạn dễ dàng kiểm tra và thu thập phản hồi về thiết kế của mình trước khi phát triển.
* **Hỗ trợ đa nền tảng:** Figma hoạt động trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS, Linux và Chrome OS.

**Lợi ích sử dụng Figma:**

* **Tăng tốc độ thiết kế:** Figma giúp bạn thiết kế giao diện nhanh hơn và hiệu quả hơn so với các công cụ truyền thống.
* **Cải thiện giao tiếp:** Figma giúp bạn dễ dàng chia sẻ và thảo luận về thiết kế của mình với đồng nghiệp và khách hàng.
* **Thu thập phản hồi nhanh chóng:** Figma giúp bạn thu thập phản hồi về thiết kế của mình từ người dùng cuối một cách nhanh chóng và hiệu quả.
* **Tạo ra các giao diện đẹp mắt và tương tác:** Figma cung cấp cho bạn đầy đủ các công cụ để tạo ra các giao diện đẹp mắt, trực quan và tương tác.

**Ví dụ sử dụng Figma:**

* **Thiết kế giao diện web:** Figma được sử dụng rộng rãi để thiết kế giao diện web, bao gồm trang chủ, trang sản phẩm, trang thanh toán, v.v.
* **Thiết kế ứng dụng di động:** Figma cũng là một công cụ phổ biến để thiết kế giao diện ứng dụng di động cho iOS và Android.
* **Thiết kế giao diện người dùng (UI) cho các sản phẩm kỹ thuật số khác:** Figma có thể được sử dụng để thiết kế giao diện người dùng cho nhiều loại sản phẩm kỹ thuật số khác nhau, chẳng hạn như bảng điều khiển, phần mềm, giao diện trò chơi điện tử, v.v.
* **Tóm lại, Figma là một ứng dụng thiết kế giao diện người dùng và tạo nguyên mẫu mạnh mẽ, dễ sử dụng và miễn phí. Nó là một công cụ tuyệt vời cho cả người mới bắt đầu và nhà thiết kế chuyên nghiệp.**

## - Giới thiệu về Tkinter Designer

Tkinter Designer là một công cụ hỗ trợ thiết kế giao diện người dùng (GUI) cho Tkinter, một thư viện Python được sử dụng để tạo GUI. Nó giúp đơn giản hóa quy trình phát triển GUI bằng cách cho phép bạn thiết kế giao diện trực quan và sau đó tự động tạo mã Tkinter tương ứng.

**Tính năng chính của Tkinter Designer:**

* **Thiết kế giao diện trực quan:** Sử dụng giao diện kéo thả để tạo bố cục GUI, thêm các widget, điều chỉnh thuộc tính và xem trước giao diện theo thời gian thực.
* **Tạo mã Tkinter tự động:** Dựa trên thiết kế của bạn, Tkinter Designer tạo mã Tkinter tương ứng, giúp tiết kiệm thời gian và công sức viết mã thủ công.
* **Hỗ trợ nhiều widget:** Tkinter Designer hỗ trợ đầy đủ các widget Tkinter phổ biến, bao gồm nút, nhãn, trường nhập, khung, thanh cuộn, menu, v.v.
* **Tùy chỉnh giao diện:** Thay đổi màu sắc, phông chữ, kích thước và các thuộc tính khác của widget để tạo giao diện theo ý muốn.
* **Lưu và chia sẻ thiết kế:** Lưu thiết kế của bạn thành tệp .tdesigner để chỉnh sửa sau hoặc chia sẻ với người khác.

**Lợi ích sử dụng Tkinter Designer:**

* **Tăng tốc độ phát triển GUI:** Tkinter Designer giúp bạn tạo GUI nhanh hơn và dễ dàng hơn so với viết mã Tkinter thủ công.
* **Giảm thiểu lỗi:** Tự động hóa việc tạo mã giúp giảm thiểu lỗi so với viết mã thủ công.
* **Thiết kế giao diện trực quan:** Giao diện kéo thả cho phép bạn dễ dàng hình dung và thiết kế giao diện GUI.
* **Tạo giao diện nhất quán:** Tkinter Designer giúp đảm bảo giao diện GUI của bạn nhất quán và chuyên nghiệp.
* **Tóm lại, Tkinter Designer là một công cụ mạnh mẽ và dễ sử dụng giúp đơn giản hóa quy trình phát triển GUI Tkinter, mang lại lợi ích cho nhiều đối tượng người dùng.**

## - Giới thiệu về Tkinter Designer

Tkinter Designer là một công cụ hỗ trợ thiết kế giao diện người dùng (GUI) cho Tkinter, một thư viện Python được sử dụng để tạo GUI. Nó giúp đơn giản hóa quy trình phát triển GUI bằng cách cho phép bạn thiết kế giao diện trực quan và sau đó tự động tạo mã Tkinter tương ứng.

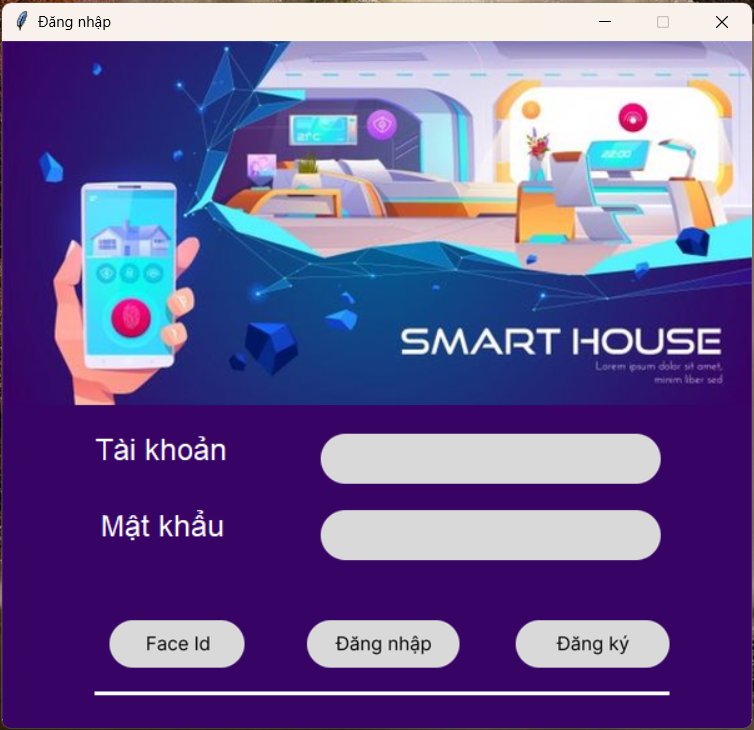
**Tính năng chính của Tkinter Designer:**

* **Thiết kế giao diện trực quan:** Sử dụng giao diện kéo thả để tạo bố cục GUI, thêm các widget, điều chỉnh thuộc tính và xem trước giao diện theo thời gian thực.
* **Tạo mã Tkinter tự động:** Dựa trên thiết kế của bạn, Tkinter Designer tạo mã Tkinter tương ứng, giúp tiết kiệm thời gian và công sức viết mã thủ công.
* **Hỗ trợ nhiều widget:** Tkinter Designer hỗ trợ đầy đủ các widget Tkinter phổ biến, bao gồm nút, nhãn, trường nhập, khung, thanh cuộn, menu, v.v.
* **Tùy chỉnh giao diện:** Thay đổi màu sắc, phông chữ, kích thước và các thuộc tính khác của widget để tạo giao diện theo ý muốn.
* **Lưu và chia sẻ thiết kế:** Lưu thiết kế của bạn thành tệp .tdesigner để chỉnh sửa sau hoặc chia sẻ với người khác.

**Lợi ích sử dụng Tkinter Designer:**

* **Tăng tốc độ phát triển GUI:** Tkinter Designer giúp bạn tạo GUI nhanh hơn và dễ dàng hơn so với viết mã Tkinter thủ công.
* **Giảm thiểu lỗi:** Tự động hóa việc tạo mã giúp giảm thiểu lỗi so với viết mã thủ công.
* **Thiết kế giao diện trực quan:** Giao diện kéo thả cho phép bạn dễ dàng hình dung và thiết kế giao diện GUI.
* **Tạo giao diện nhất quán:** Tkinter Designer giúp đảm bảo giao diện GUI của bạn nhất quán và chuyên nghiệp.
* **Tóm lại, Tkinter Designer là một công cụ mạnh mẽ và dễ sử dụng giúp đơn giản hóa quy trình phát triển GUI Tkinter, mang lại lợi ích cho nhiều đối tượng người dùng.**

**Giao diện sử dụng Tkinter Designer:**



## Giới thiệu sơ bộ về PIL

**PIL (Python Imaging Library)** là một thư viện mã nguồn mở phổ biến cho xử lý ảnh trong Python. Nó cung cấp các chức năng để mở, xem, chỉnh sửa, lưu và thao tác với các định dạng hình ảnh khác nhau như JPEG, PNG, TIFF, GIF, v.v. PIL là một thư viện nhẹ và dễ sử dụng, phù hợp cho cả người mới bắt đầu và lập trình viên Python nâng cao.

**Điểm nổi bật của PIL:**

* **Dễ sử dụng:** PIL có cú pháp đơn giản và dễ học, giúp bạn dễ dàng bắt đầu với xử lý ảnh trong Python.
* **Miễn phí và mã nguồn mở:** PIL là một thư viện miễn phí và mã nguồn mở, bạn có thể sử dụng và sửa đổi nó mà không cần bất kỳ hạn chế nào.
* **Hỗ trợ nhiều định dạng hình ảnh:** PIL hỗ trợ nhiều định dạng hình ảnh phổ biến, giúp bạn dễ dàng làm việc với nhiều loại hình ảnh khác nhau.
* **Cung cấp nhiều chức năng:** PIL cung cấp nhiều chức năng để xử lý ảnh, bao gồm:
  + Mở và lưu hình ảnh
  + Chuyển đổi định dạng hình ảnh
  + Thay đổi kích thước hình ảnh
  + Cắt, xoay và thay đổi hình dạng hình ảnh
  + Áp dụng các bộ lọc và hiệu ứng cho hình ảnh
  + Vẽ hình ảnh và đồ họa
  + V.v.

**Tóm lại, PIL là một thư viện xử lý ảnh mạnh mẽ và dễ sử dụng trong Python. Nó là một công cụ tuyệt vời cho cả người mới bắt đầu và lập trình viên Python nâng cao.**

**1.2.2 CNN**

**- Giới thiệu về CNN:**

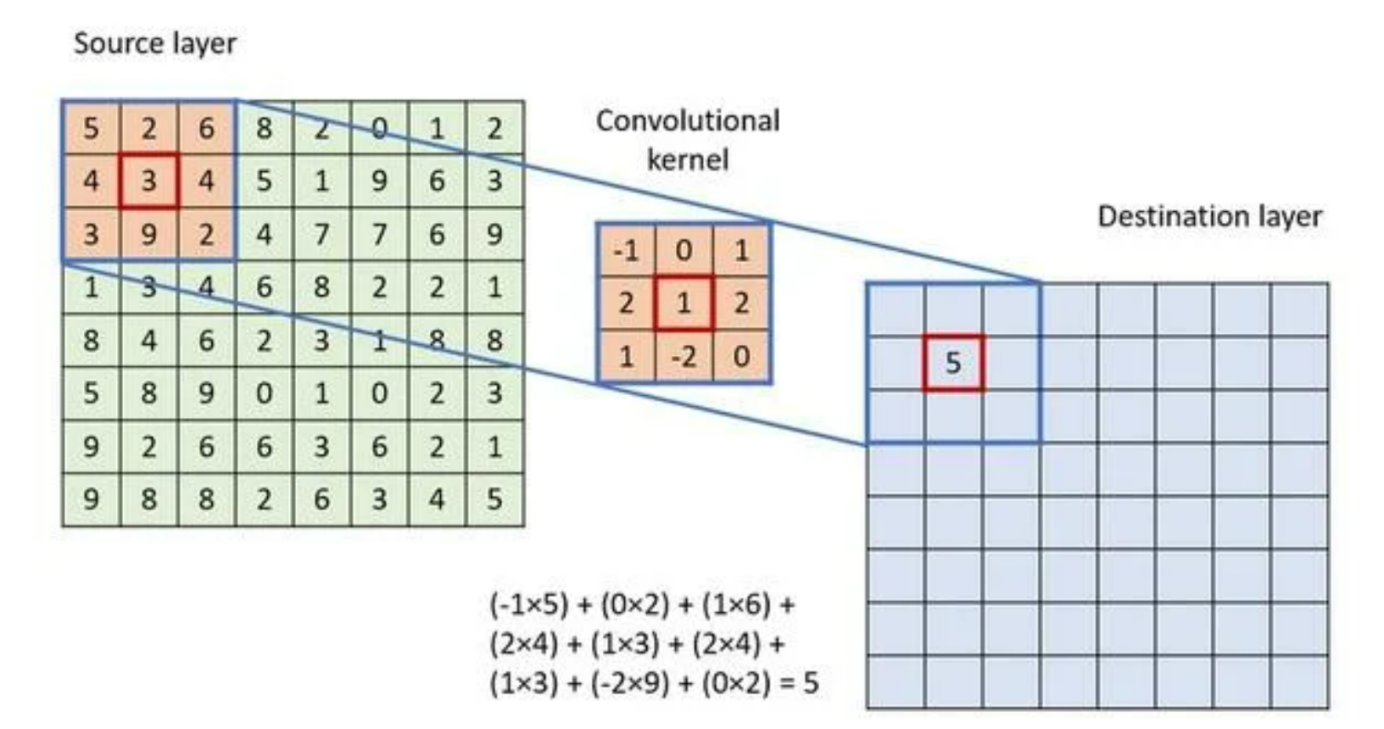
+ **Convolutional Neural Network**(CNN),là một loại mạng nơ-ron tích tụ được sử dụng phổ biến trong lĩnh vực xử lý hình ảnh. [**Thuật toán CNN**](https://lanit.com.vn/thuat-toan-cnn.html) là một phần quan trọng của Deep Learning – tập hợp các thuật toán nhằm xây dựng mô hình dữ liệu trừu tượng thông qua việc sử dụng nhiều lớp xử lý cấu trúc phức tạp.

+ CNN hoạt động dựa trên dữ liệu ảnh và được sử dụng rộng rãi trong các ứng dụng như nhận dạng hình ảnh và khuôn mặt. Thuật toán CNN hiện nay đã được tích hợp vào các nền tảng mạng phổ biến như Facebook và Google,… Khác với các mạng nơ-ron thông thường, thuật toán CNN nhận đầu vào dưới dạng một mảng hai chiều và xử lý trực tiếp trên hình ảnh, giúp nó trích xuất các đặc trưng quan trọng từ dữ liệu hình ảnh một cách hiệu quả.

**- Các lớp cơ bản của CNN:**

+ Convolutional layer:

Tích chập là lớp đầu tiên để lấy ra các đặc điểm từ hình ảnh ban đầu. Nó giữ lại các mối quan hệ giữa các pixel bằng cách xem xét các tính năng của hình ảnh thông qua việc sử dụng các ô vuông nhỏ từ dữ liệu gốc. Đây là một phép tính với hai thành phần đầu vào, là ma trận hình ảnh ban đầu và một bộ lọc hoặc hạt nhân.



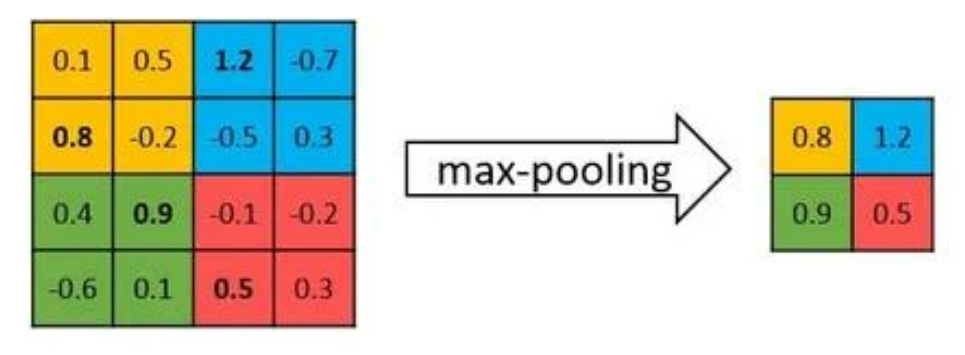
+ ReLU layer:

Lớp ReLU này là hàm kích hoạt trong mạng CNN, được gọi là activation function. Nó có tác dụng mô phỏng những nơ ron có tỷ lệ truyền xung qua axon. Các hàm activation khác như Leaky, Sigmoid, Leaky, Maxout,.. tuy nhiên hiện nay, hàm ReLU được sử dụng phổ biến và thông dụng nhất.

Hàm này được sử dụng cho những yêu cầu huấn luyện mạng nơ ron với những ưu điểm nổi bật điển hình là hỗ trợ tính toán nhanh hơn. Trong quá trình dùng hàm ReLU, bạn cần chú ý đến việc tùy chỉnh những learning rate và dead unit. Những lớp ReLU được dùng sau khi filter map được tính và áp dụng ReLU lên các giá trị của filter map.

+ Pooling layer:

Khi ma trận ảnh đầu vào có kích thước quá lớn, các lớp Pooling layer sẽ được đặt vào giữa những lớp Convolutional để làm giảm những parameters. Hiện, hai loại lớp Pooling được sử dụng phổ biến là Max pooling và Average.



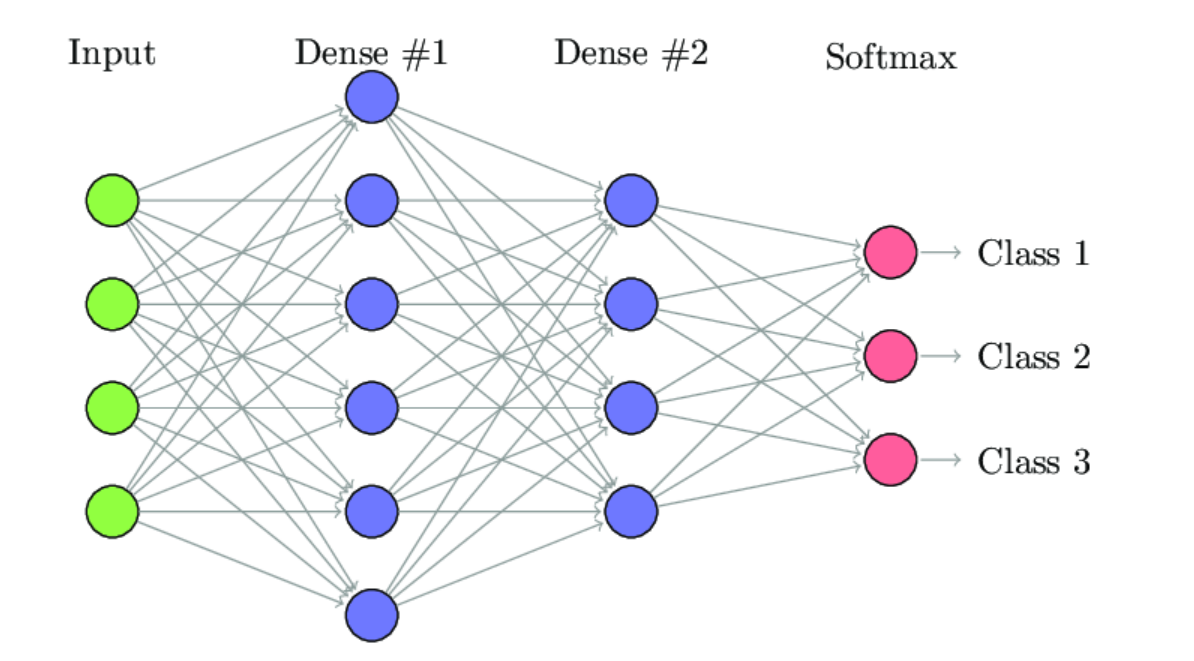
+ Fully connected layer:

Đây là lớp có nhiệm vụ đưa ra kết quả sau khi hai lớp Convolutional và Pooling đã nhận được ảnh truyền. Khi này, ta sẽ thu được một model đọc được thông tin của ảnh. Để có thể liên kế chúng cũng như cho nhiều đầu ra hơn ta sẽ sử dụng Fully connected layer.

**- Cấu trúc của mạng CNN:**

+ Mạng CNN là một trong các lớp Convolution chồng lên nhau. Nó sử dụng các hàm kích hoạt phi tuyến (như ReLU và tanh) để tạo ra thông tin trừu tượng và kích hoạt trọng số trong node. Các lớp này cung cấp tính bất biến và tính kết hợp, nhằm giúp mạng hiểu được đối tượng theo nhiều góc độ khác nhau.

+ Pooling layer giúp làm cho mạng không nhạy cảm đối với việc dịch chuyển, co giãn và quay hình ảnh. Còn tính kết hợp cục bộ là kết quả của các lớp Convolution liên kết với nhau. Mỗi lớp tiếp theo dựa trên kết quả của lớp Convolution trước đó và đảm bảo kết nối cục bộ hiệu quả nhất.

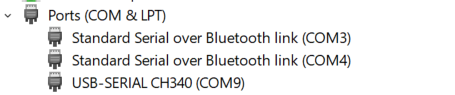


+ Trong quá trình huấn luyện, thuật toán CNN sẽ tự động học cách nhận biết các đặc điểm thông qua việc sử dụng các bộ lọc. Quá trình này tương tự việc bộ não con người nhận diện các đối tượng trong thế giới thực.

**Chương 2 Xây dựng chương trình**

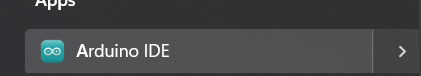
**2.1 Arduino**

**-** Đầu tiên ta phải xác định được máy tính kết nối với arduino thông qua cổng nào:

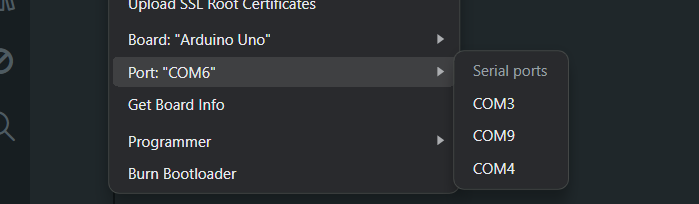


Như ví dụ ở đây arduino được kết nối thông qua cổng COM9

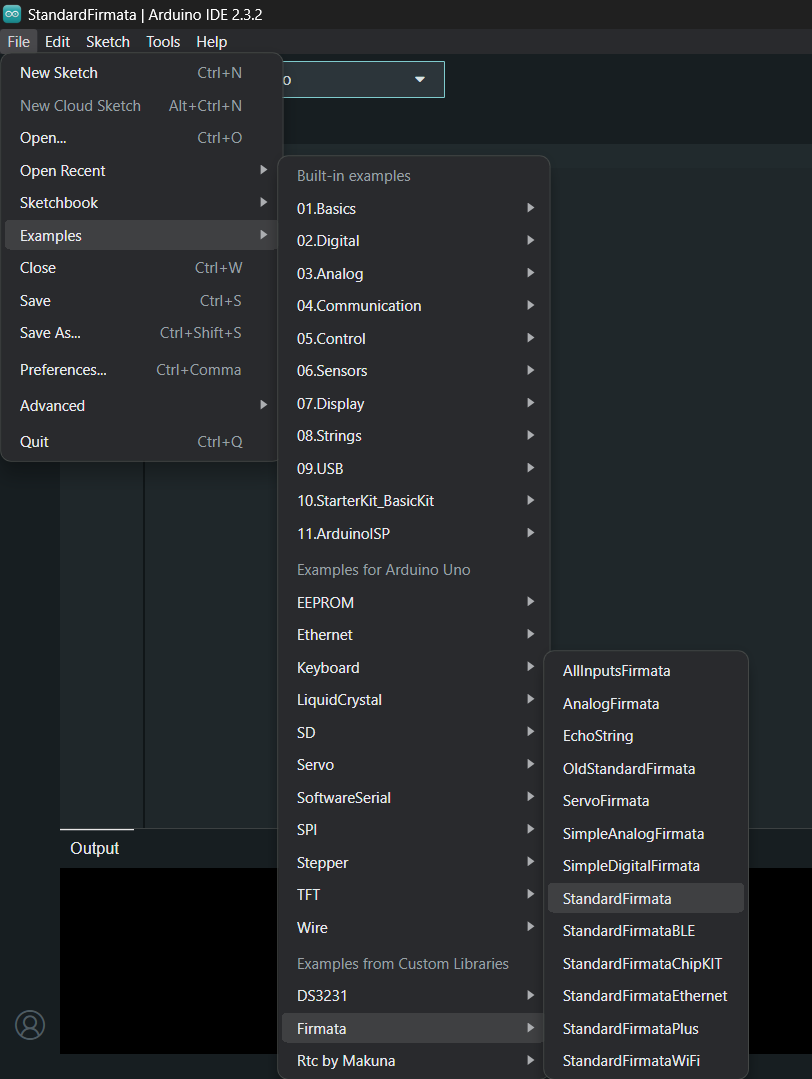
Sau khi đã xác định được cổng kết nối, sử dụng phần mềm IDE của arduino để tiếp tục thiết đặt



Điều chỉnh port trong arduino IDE cho phù hợp với cổng kết nối:



Sau khi hoàn thành việc thiết lặp, tạo một sketch mới, sau đó vào File > example >Frimata > standraFirmata



Cuối cùng nạp đoạn code vừa mới tạo ra cho arduino. Lưu ý: phải tải thư viện firmata từ ngoài rồi import vào Arduino IDE.

Đối với python:

Import thư viện pyfrimata để có thể giao tiếp với arudino.



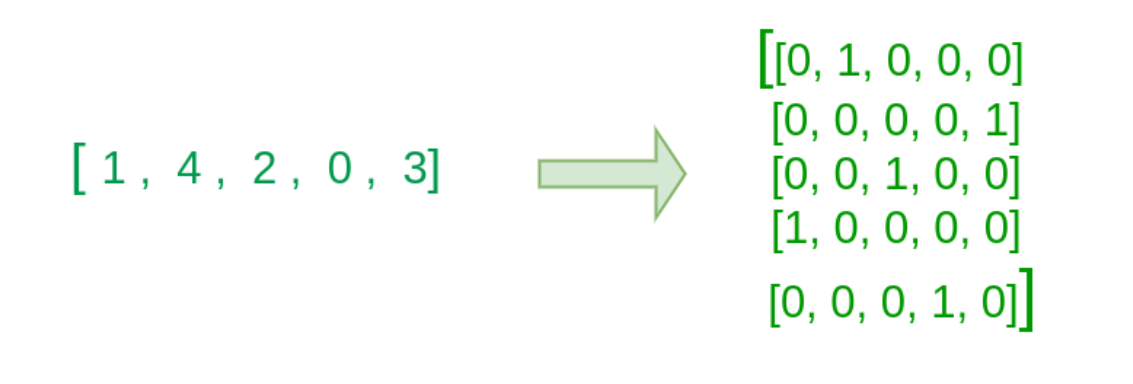
**2.2 Nhận diện khuôn mặt**

**- Đầu tiên ta sẽ thu thập và tiền xử lý dữ liệu:**

+ Ta sẽ sử dụng nguồn dữ liệu có sẵn là ảnh khuôn mặt hoặc dùng camera/webcam để chụp lại những tấm ảnh có khuôn mặt ta trong đó

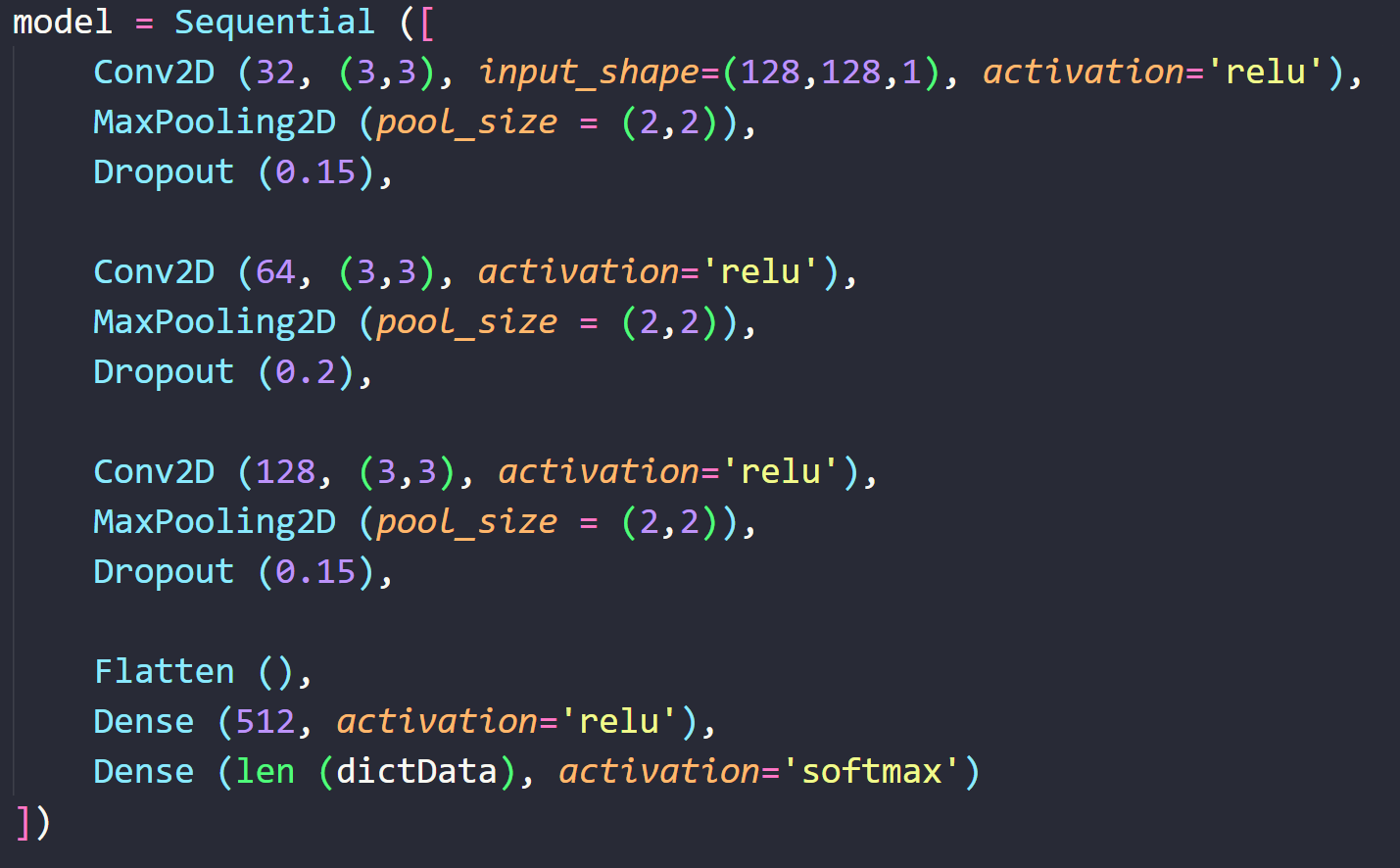
+ Với mỗi khuôn mặt ta sẽ gán nhãn cho tương ứng cho nó (có thể là tên của người đó hoặc id)

+ Với mỗi cái nhãn đó ta sẽ mã hóa nó về dạng one hot encoding

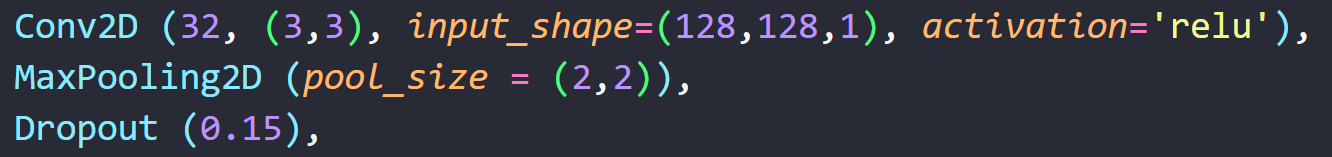


+ Chuẩn hóa kích thước của các hình ảnh về một kích thước tiêu chuẩn (ví dụ: 128 x 128 pixel) để làm đầu vào cho mô hình CNN

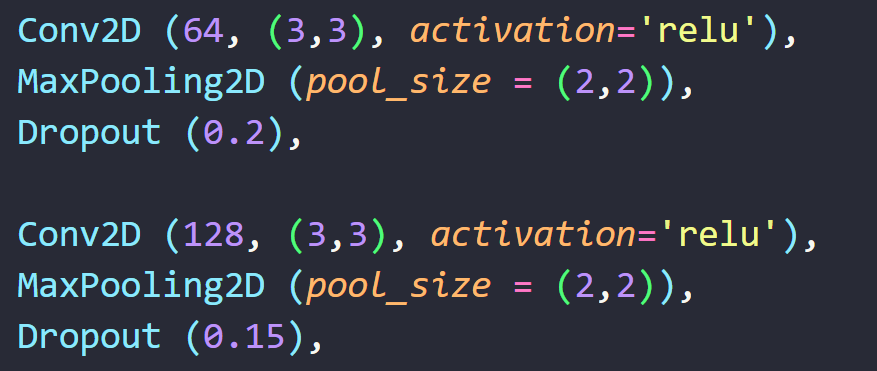
**- Xây dựng mô hình CNN sử dụng thư viện keras:**



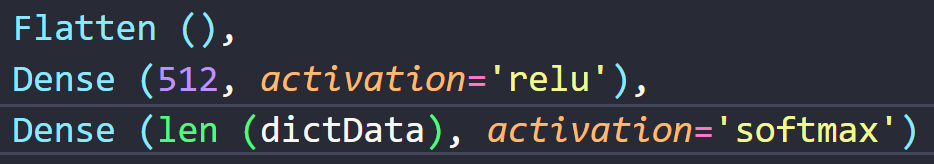
Đầu tiên ta sẽ khởi tạo một mô hình tuần tự Sequential () để xử lý dữ liệu đầu vào



Đoạn này sẽ khởi tạo lớp tích chập đầu tiên với 32 bộ lọc kernel có kích thước 3x3, đầu vào của hình có kích thước là 128x128 với kênh màu là xám. Hàm kích hoạt là ReLU. Sau đó là tới pooling layer, lớp này sẽ giảm kích thước đầu ra của lớp tích chập bằng cách lấy giá trị lớn nhất trong khung cửa sổ 2x2. Sau đó là ta dùng Dropout để tạo ra các mô hình con ngẫu nhiên từ mô hình gốc giúp tránh overfitting



Tiếp theo là 1 lớp tích chập Conv2D với 64 bộ lọc kernel, hàm kích hoạt là ReLU, theo sau vẫn là 1 lớp pooling và Dropout. Tiếp nối ở sau là 1 lớp tích chập Con2D với 128 bộ lọc kernel và thực hiện giống trên



Ta sẽ dùng Flatten () để làm phẳng đầu ra của lớp tích chập thành 1 vector một chiều. Dense(512, activation='relu') là lớp Fully connected với 512 nút ẩn được dùng hàm kích hoạt là ReLU. Dense(len(dictData), activation='softmax') là lớp fully connected cuối cùng với số nút bằng với số lượng đầu ra mong muốn, sử dụng hàm kích hoạt softmax vì đây là bài toán phân loại

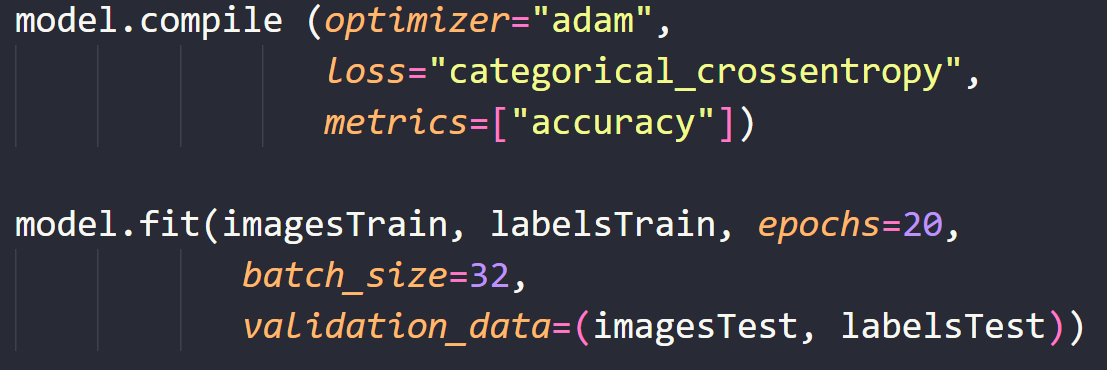
Mô hình này sử dụng ba lớp tích chập với kích thước bộ lọc tăng dần (32, 64, 128) để trích xuất các đặc trưng từ hình ảnh đầu vào. Các lớp pooling được sử dụng để giảm kích thước đầu ra và tăng khả năng chịu lỗi. Các lớp dropout được sử dụng để giảm overfitting. Cuối cùng, các lớp fully connected được sử dụng để phân loại đầu ra thành các lớp danh tính khuôn mặt.

**- Huấn luyện mô hình:**

+ Nạp dữ liệu huấn luyện: Nạp tập dữ liệu huấn luyện vào mô hình CNN.

+ Quá trình huấn luyện: Trong mỗi vòng lặp huấn luyện (epoch), mô hình sẽ học từ dữ liệu huấn luyện bằng cách tính toán đầu ra dự đoán, so sánh với nhãn thực tế, và cập nhật trọng số của mạng để giảm thiểu hàm mất mát.

+ Theo dõi hiệu suất: Trong quá trình huấn luyện, theo dõi độ chính xác trên tập huấn luyện và tập xác minh để đánh giá hiệu suất của mô hình và phát hiện quá khớp (overfitting) hoặc thiếu khớp (underfitting).



**- Đánh giá mô hình:**

+ Đánh giá trên tập kiểm tra: Sau khi hoàn thành quá trình huấn luyện, đánh giá hiệu suất của mô hình trên tập kiểm tra, đây là dữ liệu mà mô hình chưa từng gặp trước đó.

+ Phân tích kết quả: Phân tích các trường hợp mà mô hình dự đoán sai và xác định các điểm yếu của mô hình.

**- Cải thiện mô hình:**

+ Tăng cường dữ liệu (Data augmentation): Áp dụng các kỹ thuật như xoay ảnh, lật ảnh, thay đổi độ sáng, v.v. để tăng kích thước và sự đa dạng của tập dữ liệu huấn luyện.

**2.3 Tkinter**

-Đầu tiên ta import thư viện Tkinter

-Tiếp theo ta tạo ra cửa sổ chính cho ứng dụng bằng lớp có sẵn của Tkinter là tk.Tk(),chỉnh sửa các thuộc tính của cửa sổ chính như kích thước (geometry),tiêu đề(tilte)…

-Sau đó ta thêm các widget có sẵn trong Tkinter và thay đổi màu sắc , kích thước,vị trí tùy theo sở thích để tạo nên 1 giao diện đẹp,phù hợp với nhu cầu người dùng.

**2.4 Tkinter Designer**

**-Các bước thực hiện :**

Đầu tiên ta sẽ tạo giao diện cơ bản từ ứng dụng figma bằng cách kéo thả

Tạo ra các Retangle số lượng tùy theo mục đích sử dụng

Tạo các trường text

Xét các kích thước cho các thành phần

Căn chỉnh vị trí sao cho cân đối, hợp lý

Chỉnh sửa các thuộc tính cơ bản của các thành phần con như background,forceground,border,căn trái ,phải,giữa …

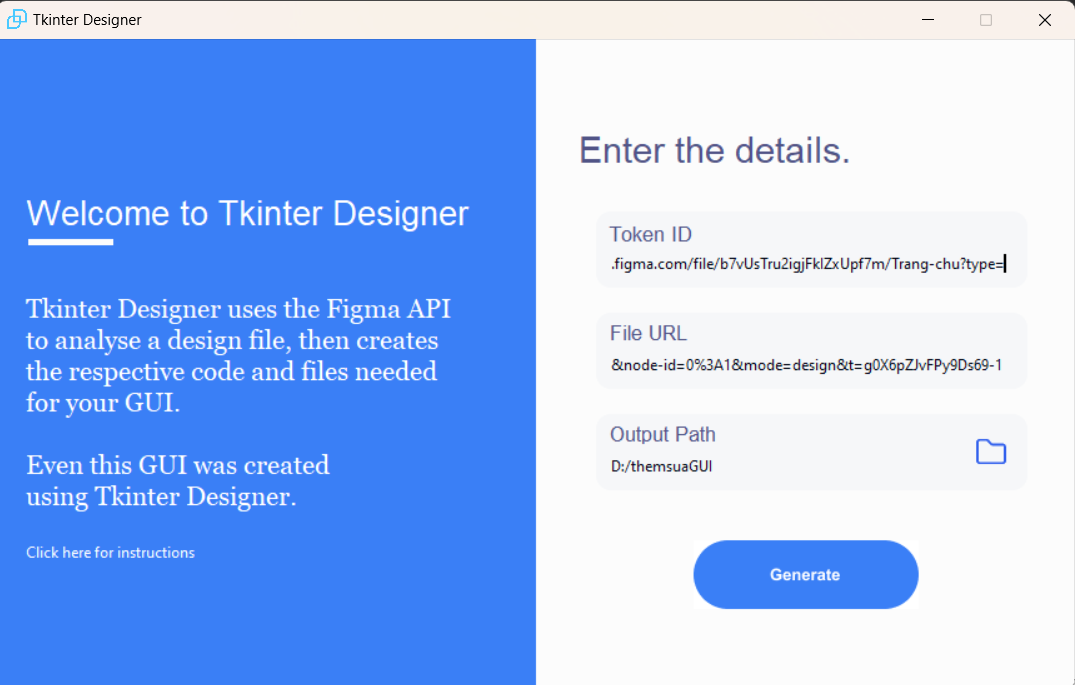
Đổi tên các Retangle về thành các Element tương ứng

Ảnh có chứa văn bản, ảnh chụp màn hình, số, hàng

Mô tả được tạo tự động

Lấy Token ID và link của file figma gắn vào Tkinter Designer

Chọn đường dẫn



Tiến hành chuyển đổi