TRƯỜNG ĐẠI HỌC SÀI GÒN



KHOA CÔNG NGHỆ THÔNG TIN

**----------****----------**

BÁO CÁO ĐỒ ÁN CUỐI KỲ

HỌC PHẦN: NGÔN NGỮ LẬP TRÌNH PYTHON

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG QUẢN LÝ SINH VIÊN**

|  |  |  |
| --- | --- | --- |
| **MÃ HỌC PHẦN:** | **841422** | |
| **NHÓM MÔN HỌC:** | **04** | |
| **HỌC KỲ:** | **02** | |
| **NHÓM BÁO CÁO:** | **PythonGroup1** | |
| **GVHD**: | **ThS.NCS HÀ THANH DŨNG** | |
| **THÀNH VIÊN**: | **3122410149** | **Vũ Hoàng Chung**  **(Nhóm trưởng)** |
|  | **3123410083** | **Khưu Minh Đức** |
|  | **3122410180** | **Nguyễn Quốc Khánh** |
|  | **3122410186** | **Trần Đăng Khoa** |
|  |  |  |

**TP HỒ CHÍ MINH, THÁNG 5 NĂM 2023**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**



KHOA CÔNG NGHỆ THÔNG TIN

**----------****----------**

BÁO CÁO ĐỒ ÁN CUỐI KỲ

HỌC PHẦN: NGÔN NGỮ LẬP TRÌNH PYTHON

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG QUẢN LÝ SINH VIÊN

|  |  |  |
| --- | --- | --- |
| **MÃ HỌC PHẦN:** | **841422** | |
| **NHÓM MÔN HỌC:** | **04** | |
| **HỌC KỲ:** | **02** | |
| **NHÓM BÁO CÁO:** | **PythonGroup1** | |
| **GVHD**: | **ThS.NCS HÀ THANH DŨNG** | |
| **THÀNH VIÊN**: | **3122560007** | **Vũ Hoàng Chung**  **(Nhóm trưởng)** |
|  | **3123410083** | **Khưu Minh Đức** |
|  | **3122410180** | **Nguyễn Quốc Khánh** |
|  | **3122410186** | **Trần Đăng Khoa** |

**TP HỒ CHÍ MINH, THÁNG 5 NĂM 2023**

**PHÂN CÔNG VÀ ĐÁNH GIÁ**

* **Chuẩn bị cho phần thuyết trình:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Thứ tự** | **Họ tên** | **Mã số sinh viên** | **Nội dung công việc** | **Mức độ đóng góp** | **Giảng viên đánh giá** |
| 1 | Vũ Hoàng Chung | 3122560007 | Soạn lý thuyết báo cáo, chuẩn bị Powerpoint | 25% |  |
| 2 | Khưu Minh Đức | 3123410083 | Soạn lý thuyết báo cáo, chuẩn bị Powerpoint | 25% |  |
| 3 | Nguyễn Quốc Khánh | 3122410180 | Soạn lý thuyết báo cáo, chuẩn bị Powerpoint | 25% |  |
| 4 | Trần Đăng khoa | 3122410186 | Soạn lý thuyết báo cáo, chuẩn bị Powerpoint | 25% |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Thứ tự** | **Họ tên** | **Mã số sinh viên** | **Nội dung công việc** | **Mức độ đóng góp** | **Giảng viên đánh giá** |
| 1 | Vũ Hoàng Chung | 3122560007 | Nhận diện khuôn mặt và chức năng điểm danh sinh viên | 25% |  |
| 2 | Khưu Minh Đức | 3123410083 | Tạo giao diện của sinh viên, khoa, màn hình đăng nhập và chức năng | 25% |  |
| 3 | Nguyễn Quốc Khánh | 3122410180 | Thiết kế database, viết truy vấn dữ liệu | 25% |  |
| 4 | Trần Đăng khoa | 3122410186 | Tạo giao diện chính của hệ thống, kết hợp chức năng trên giao diện và hàm truy vấn đã có | 25% |  |

* **Phần kĩ thuật và thực hành đồ án:**

**NHẬN XÉT CỦA GIẢNG VIÊN**

MỤC LỤC

[LỜI MỞ ĐẦU 5](#_Toc197101567)

[Lý do chọn đề tài: 5](#_Toc197101568)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 6](#_Toc197101569)

[1. Giới thiệu về ngôn ngữ Python 6](#_Toc197101570)

[1.1 Ngôn Ngữ Python là gì? 6](#_Toc197101571)

[1.2 Lợi ích của Python 6](#_Toc197101572)

[1.3 Ứng dụng của Python 6](#_Toc197101573)

[2. Các thư viện của Python được sử dụng 7](#_Toc197101574)

[2.1 Cơ sỡ dữ liệu 7](#_Toc197101575)

[2.2 Thư viện hệ thống 7](#_Toc197101576)

[2.3 Thư viện giao diện 8](#_Toc197101577)

[2.4 Thư viện phục vụ nhận diện khuôn mặt 9](#_Toc197101578)

[3. Lý thuyết CNN 10](#_Toc197101579)

[3.1 Deep learning 10](#_Toc197101580)

[3.2 Convolutional Neural Network (CNN): 10](#_Toc197101581)

[3.3 TensorFlow 10](#_Toc197101582)

[3.4 scikit-learn 10](#_Toc197101583)

[3.5 MTCNN 10](#_Toc197101584)

[1. Phân tích bài toán cơ bản 11](#_Toc197101585)

[2. Phân tích bài toán ứng dụng AI (Nhận diện hình ảnh) 11](#_Toc197101586)

[CHƯƠNG 3: CÀI ĐẶT, DEMO CHƯƠNG TRÌNH 11](#_Toc197101587)

[1. Cài đặt 11](#_Toc197101588)

[2. Demo chương trình 13](#_Toc197101589)

[KẾT LUẬN 27](#_Toc197101590)

[1. ĐÁNH GIÁ NHỮNG NỘI DUNG ĐÃ THỰC HIỆN 27](#_Toc197101591)

[2. ĐÁNH GIÁ NHỮNG NỘI DUNG CHƯA THỰC HIỆN 27](#_Toc197101592)

[3. HƯỚNG PHÁT TRIỂN 27](#_Toc197101593)

# LỜI MỞ ĐẦU

## Lý do chọn đề tài:

Trong bối cảnh công nghệ thông tin phát triển nhanh chóng, việc số hóa quy trình quản lý và tự động hóa các nghiệp vụ trong giáo dục là một xu hướng tất yếu. Tại các trường đại học, việc quản lý thông tin sinh viên, giảng viên và các đơn vị khoa chuyên môn vẫn còn đang được thực hiện thủ công ở nhiều nơi, gây ra sự chồng chéo, kém hiệu quả và khó khăn trong việc truy xuất dữ liệu nhanh chóng. Đặc biệt, quá trình điểm danh sinh viên trong các buổi học hiện nay vẫn phụ thuộc phần lớn vào hình thức gọi tên truyền thống hoặc ký tên, vừa mất thời gian, vừa dễ xảy ra sai sót hoặc gian lận.

Từ thực tiễn đó, nhóm em lựa chọn đề tài **“Hệ thống quản lý thông tin sinh viên kết hợp điểm danh bằng nhận diện khuôn mặt”** nhằm giải quyết đồng thời hai vấn đề: quản lý dữ liệu người dùng (gồm sinh viên, giảng viên, khoa) một cách tập trung, có cấu trúc và ứng dụng công nghệ thị giác máy tính (Computer Vision) để tự động hóa quy trình điểm danh trong lớp học.

Đề tài tận dụng mô hình học sâu (Deep Learning) để huấn luyện và nhận diện khuôn mặt từ dữ liệu ảnh, tích hợp với thư viện MTCNN (Multi-task Cascaded Convolutional Networks) để phát hiện khuôn mặt chính xác và nhanh chóng. Ngoài ra, mô hình nhận diện sử dụng kiến trúc CNN (Convolutional Neural Network) huấn luyện từ tập dữ liệu khuôn mặt chuẩn hóa giúp tăng độ chính xác và ổn định khi điểm danh qua webcam.

Trong đề tài này, chúng em hy vọng tạo ra được một phần mềm giúp ích được trong việc quản lý sinh viên, cũng như học hỏi thêm về các mô hình nhận diện khuôn mặt giúp cải thiện khả năng tìm hiểu và học hỏi các công nghệ mới áp dụng trong thực tiễn.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1. Giới thiệu về ngôn ngữ Python

### 1.1 Ngôn Ngữ Python là gì?

Python là một ngôn ngữ lập trình bậc cao, được tạo ra bởi Guido van Rossum vào những năm 1980 và phát triển tiếp theo trong thập niên 1990. Python là một ngôn ngữ lập trình đơn giản, dễ học và dễ sử dụng, có thể được sử dụng để xây dựng các ứng dụng máy tính, trang web, các hệ thống đám mây, trí tuệ nhân tạo, máy học và các ứng dụng phân tích dữ liệu. Python cũng là một ngôn ngữ lập trình đa nền tảng, có thể chạy trên các hệ điều hành khác nhau như Windows, Linux và MacOS. Python có một cộng đồng lớn và đa dạng, với hàng ngàn thư viện và framework hỗ trợ cho việc phát triển ứng dụng.

### 1.2 Lợi ích của Python

+ *Dễ học và sử dụng:* Python có cú pháp đơn giản và dễ hiểu, giúp người mới bắt đầu học lập trình dễ dàng tiếp cận và làm quen với các khái niệm lập trình cơ bản.

+ *Đa năng:* Python có thể được sử dụng để phát triển các loại ứng dụng khác nhau như ứng dụng máy tính, web, trí tuệ nhân tạo, máy học, phân tích dữ liệu, đám mây, hệ thống nhúng và nhiều hơn nữa.

+ *Cộng đồng lớn và đa dạng:* Python có một cộng đồng phát triển lớn và đa dạng, với hàng ngàn thư viện và framework hỗ trợ cho việc hát triển ứng dụng.

+ *Tính tương thích cao:* Python là một ngôn ngữ đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và MacOS.

+ *Hiệu suất cao:* Python có thể được tối ưu hóa để đạt được hiệu suất cao, đặc biệt là khi sử dụng các thư viện và framework được viết bằng C/C++.

+ *Dễ dàng mở rộng:* Python có tính mở rộng cao, cho phép người dùng tạo ra các module và thư viện riêng để mở rộng chức năng của Python.

+ *Tương tác tốt với các ngôn ngữ khác:* Python có thể tương tác tốt với các ngôn ngữ khác như C/C++, Java, và .NET, giúp cho việc tích hợp các phần mềm và dịch vụ khác nhau trở nên dễ dàng hơn.

### 1.3 Ứng dụng của Python

Dưới đây là một số ví dụ về các ứng dụng của Python:

+ *Phát triển ứng dụng web*: Python có thể được sử dụng để phát triển các ứng dụng web, với các framework như Django, Flask và Pyramid.

+ *Trí tuệ nhân tạo và máy học*: Python là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực trí tuệ nhân tạo và máy học, với các thư viện như TensorFlow, Keras và PyTorch.

+ *Phân tích dữ liệu*: Python có các thư viện như Pandas và NumPy để xử lý và phân tích dữ liệu, và Matplotlib và Seaborn để tạo đồ thị và biểu đồ.

+ *Game development*: Python có thể được sử dụng để phát triển game với các thư viện như Pygame và PyOpenGL.

+ *Đám mây*: Python có thể được sử dụng để quản lý và triển khai các hệ thống đám mây với các framework như OpenStack và Ansible.

+ *Hệ thống nhúng*: Python có thể được sử dụng để phát triển các hệ thống nhúng, với các framework như MicroPython và CircuitPython.

+ *Ứng dụng máy tính*: Python có thể được sử dụng để phát triển các ứng dụng máy tính trên nhiều nền tảng khác nhau, với các thư viện như PyQt và wxPython.

## 2. Các thư viện của Python được sử dụng

Các thư viện được sử dụng trong đồ án:

* *mysql.connector*
* *os, sys*
* *Import cv2*
* *Import numpy*
* *Tkinter, messagebox*
* *Datetime from datetime*
* *Tensorflow*
* *slkearn*
* *mtcnn*

### 2.1 Cơ sỡ dữ liệu

#### 2.1.1 mysql.connector

Thư viện mysql.connector là driver chính thức của Python để kết nối và tương tác với cơ sở dữ liệu MySQL. Trong đồ án này, nó được sử dụng để:

+ Thiết lập kết nối tới CSDL lưu trữ thông tin sinh viên, giảng viên, khoa và bảng điểm danh.

+ Thực thi các câu lệnh SQL (INSERT, SELECT) để ghi nhận dữ liệu điểm danh khi hệ thống nhận diện khuôn mặt thành công.

+ Quản lý giao dịch (commit/rollback) đảm bảo tính toàn vẹn của dữ liệu trong việc lưu trữ tông tin sinh viên

### 2.2 Thư viện hệ thống

#### 2.2.2 os, sys

- **os**: Cung cấp các phương thức thao tác với hệ thống tệp và thư mục. Nhóm em sử dụng để:

+ Xây dựng đường dẫn động đến thư mục ảnh gốc (image/), thư mục lưu khuôn mặt đã cắt (faces/).

+ Kiểm tra sự tồn tại và tự động tạo thư mục khi cần: tránh lỗi file không tìm thấy.

* **sys**: Hỗ trợ xử lý các tham số dòng lệnh (nếu có) và mở rộng đường dẫn tìm kiếm:

**+** thêm đường dẫn thư mục cha của file hiện tại vào sys.path, giúp import module trong cùng project dễ dàng hơn, ngay cả khi thư mục đó không phải là package.

#### 2.2.3 cv2

OpenCV là một thư viện các chức năng lập trình chủ yếu dành cho thị giác máy tính thời gian thực. Trong đồ án:

+ **Đọc & ghi ảnh**: dùng cv2.imread(), cv2.imwrite() để nạp và lưu khuôn mặt cắt ra.

+ **Tiền xử lý ảnh:** chuyển đổi màu sắc (cv2.cvtColor), resize (cv2.resize) trước khi huấn luyện hay dự đoán.

+ **Hiển thị kết quả điểm danh:** vẽ khung chữ nhật, chú thích tên sinh viên lên webcam feed với cv2.rectangle và cv2.putText.

#### 2.2.4 numpy

Thư viện nền tảng cho tính toán khoa học, đặc biệt với mảng đa chiều (ndarray).

+ **Lưu trữ & xử lý dữ liệu ảnh**: chuyển đổi danh sách ảnh thành mảng np.array(X) để đưa vào mô hình.

+ **Chuẩn hóa**: chia giá trị pixel cho 255.0 để đưa về [0, 1] giúp mô hình hội tụ nhanh hơn.

+ **Lưu trữ nhãn**: dùng np.save()/np.load() đọc–ghi mảng nhãn (labels.npy).

#### 2.2.5 Datetime form datetime

Cung cấp lớp datetime để làm việc với ngày giờ:

+ Ghi nhận chính xác thời gian điểm danh (datetime.now()).

+ Lưu trữ vào CSDL để tra cứu dữ liệu điểm danh theo ngày–giờ.

### 2.3 Thư viện giao diện

#### 2.3.1 Tkinter, messagebox

Thư viện GUI tiêu chuẩn của Python:

+ **Tkinter** là một liên kết với bộ công cụ Tk GUI cho Python. Đây là giao diện Python tiêu chuẩn cho bộ công cụ GUI Tk và là GUI tiêu chuẩn trên thực tế của Python. Tkinter được bao gồm trong các bản cài đặt Python, Microsoft Windows và macOS tiêu chuẩn của Python.

+ **Messagebox** là một mô-đun trong python cung cấp một tập hợp các hộp thoại khác nhau được sử dụng để hiển thị các hộp thông báo, hiển thị lỗi hoặc cảnh báo, tiện ích để chọn tệp hoặc thay đổi màu sắc

### 2.4 Thư viện phục vụ nhận diện khuôn mặt

#### 2.4.1 tensorflow

**Framework Deep Learning của Google**, chịu trách nhiệm xây dựng, huấn luyện và lưu mô hình CNN nhận diện khuôn mặt.+ Xây dựng và huấn luyện mô hình CNN nhận diện khuôn mặt.

+ Xây dựng mô hình CNN

+ Cung cấp API Keras (tf.keras.layers, tf.keras.models.Sequential) để thiết kế các tầng Conv2D, MaxPooling, Dense, Dropout.

+ Hỗ trợ lưu (model.save()) và nạp lại mô hình (tf.keras.models.load\_model()).

#### 2.4.2 slkearn

+ **LabelEncoder** (sklearn.preprocessing): chuyển đổi nhãn người dùng (user\_id) từ chuỗi sang mã số để phù hợp với bài toán phân lớp.

+ **train\_test\_split** (sklearn.model\_selection): tách dữ liệu thành tập huấn luyện và kiểm tra, đảm bảo đánh giá đúng khả năng khái quát của mô hình.

#### 2.4.3 mtcnn

Thư viện thực thi thuật toán **M**ulti-task **C**ascaded **C**onvolutional **N**etworks, phát hiện khuôn mặt chính xác trong ảnh RGB.

+ Phát hiện nhanh và chính xác các vùng khuôn mặt trong ảnh RGB.

+ Trả về tọa độ box, từ đó cắt ảnh khuôn mặt nguyên vẹn để đưa vào mô hình nhận diện.

## 3. Lý thuyết CNN

### 3.1 Deep learning

Là một lĩnh vực của máy học (machine learning) tập trung vào việc xây dựng và huấn luyện các mạng neuron sâu (deep neural networks) để học biểu diễn dữ liệu phức tạp từ các dạng đầu vào không cấu trúc hoặc dữ liệu cấu trúc có chiều cao. Cùng với CNN, deep learning thường được sử dụng để giải quyết các bài toán như phân loại ảnh, nhận dạng văn bản, dịch máy, và nhiều ứng dụng khác.

Các mạng neuron sâu (deep neural networks) bao gồm nhiều lớp neuron kết nối với nhau thành các cấu trúc phức tạp. Một số loại mạng neuron sâu phổ biến bao gồm:

**+ Mạng nơ-ron Lan truyền tiến (Feedforward Neural Networks):** Đây là một trong những loại mạng neuron đơn giản nhất, trong đó dữ liệu chảy qua từ lớp đầu vào đến lớp đầu ra mà không có kết nối phản hồi (feedback).

**+ Mạng Nơ-ron Hồi quy (Recurrent Neural Networks - RNNs):** Loại mạng này có khả năng xử lý dữ liệu có cấu trúc dạng chuỗi, như ngôn ngữ tự nhiên hoặc dữ liệu chuỗi thời gian. Các lớp trong RNNs có kết nối phản hồi, cho phép chúng nhớ thông tin từ quá khứ.

**+ Mạng Nơ-ron Hồi quy Dài-ngắn (Long Short-Term Memory - LSTM Networks):** Một dạng đặc biệt của RNNs, LSTM có khả năng học và lưu trữ thông tin trong thời gian dài, giúp giải quyết vấn đề triệt tiêu biến mất hoặc bùng nổ đạo hàm trong quá trình huấn luyện.

**+ Mạng Nơ-ron Con đỏ (Convolutional Neural Networks - CNNs)**: Loại mạng này được thiết kế đặc biệt cho việc xử lý và phân tích dữ liệu không gian như ảnh và video, thông qua việc sử dụng các lớp convolutional và pooling.

**+ Mạng Nơ-ron Gần sâu (Deep Belief Networks - DBNs):** Loại mạng này thường sử dụng trong các ứng dụng như phân loại ảnh và nhận dạng âm thanh.

### 3.2 Convolutional Neural Network (CNN):

+ CNN là một dạng của mạng nơ-ron nhân tạo (neural network) được thiết kế đặc biệt cho việc xử lý dữ liệu không gian như hình ảnh.

+ Cấu trúc của CNN bao gồm các lớp chính như lớp convolution (tích chập), lớp pooling (tổng hợp), và các lớp kết nối đầy đủ (fully connected layers).

+ Lớp convolution giúp mô hình học được các đặc trưng cục bộ từ dữ liệu đầu vào bằng cách áp dụng các bộ lọc (kernels) qua ảnh.

+ Lớp pooling giúp giảm kích thước của đầu vào và giữ lại các đặc trưng quan trọng.

Các lớp kết nối đầy đủ được sử dụng để kết nối các đặc trưng đã học được với các lớp đầu ra để thực hiện phân loại.

**+** *Lớp Convolutional (Convolutional Layer):*

* Lớp này chịu trách nhiệm xử lý đầu vào bằng cách áp dụng các bộ lọc (kernels) qua ảnh đầu vào để tạo ra các đặc trưng cục bộ.
* Các bộ lọc này giúp phát hiện các đặc trưng như cạnh, góc, hoặc các hình dạng cụ thể trong ảnh.
* Kết quả của lớp convolution thường được gọi là các feature map.

**+** *Lớp Tổng hợp (Pooling Layer):*

* Lớp này giúp giảm kích thước của feature map, đồng thời giữ lại các thông tin quan trọng.
* Các phương pháp phổ biến trong pooling là max pooling và average pooling, trong đó max pooling giữ lại giá trị lớn nhất trong mỗi vùng và bỏ qua các giá trị khác

.

**+** *Lớp Kích hoạt (Activation Layer):*

* Lớp này thường được đặt sau lớp convolution và pooling để tạo ra các biểu diễn phi tuyến tính từ đầu vào.
* Hàm kích hoạt phổ biến nhất là ReLU (Rectified Linear Activation), nhưng cũng có thể sử dụng các hàm khác như tanh hoặc sigmoid.
* Lớp Kết nối Đầy đủ (Fully Connected Layer):
* Lớp này thường được sử dụng ở cuối mạng để kết nối các đặc trưng đã học được với các lớp đầu ra để thực hiện phân loại.
* Các lớp fully connected được sử dụng để học các mối quan hệ phức tạp giữa các đặc trưng và lớp đầu ra.

### 3.3 TensorFlow

TensorFlow là một framework mã nguồn mở do Google phát triển, chuyên hỗ trợ xây dựng và huấn luyện các mô hình Deep Learning ở quy mô lớn.

* **Đồ thị tính toán (Computational Graph):** TensorFlow biểu diễn mọi phép tính dưới dạng đồ thị: các nút (nodes) là phép toán, các cạnh (edges) là dữ liệu (tensor) chảy qua. Cơ chế này cho phép tối ưu hóa, phân phối tính toán và tận dụng GPU hiệu quả.
* **Eager Execution vs Graph Mode:**

+ *Graph Mode* (TF 1.x): Xây dựng đồ thị trước, sau đó “chạy” cả đồ thị.

+ *Eager Mode* (TF 2.x, mặc định): Thực thi tuần tự như Python thông thường, giúp debug và phát triển nhanh.

* **API Keras tích hợp:**

**+** tf.keras.Sequential và tf.keras.Model cho phép định nghĩa mô hình tuần tự hoặc tùy biến.

**+** Các lớp như Conv2D, MaxPooling2D, Dense, Dropout đều nằm trong tf.keras.layers.

**+** Các hàm hỗ trợ compile (model.compile(optimizer, loss, metrics)), huấn luyện (model.fit(...)) và lưu/nạp mô hình (model.save(), tf.keras.models.load\_model()).

* **Tensor và biến (Variables):** TensorFlow sử dụng tf.Tensor để biểu diễn mảng nhiều chiều, và tf.Variable để lưu trữ trọng số có thể cập nhật trong quá trình huấn luyện.

### 3.4 scikit-learn

Scikit-learn là một thư viện học máy phổ biến và mạnh mẽ, hỗ trợ nhiều loại thuật toán cũng như các công cụ để trực quan hóa Machine Learning, tiền xử lý, điều chỉnh, lựa chọn và đánh giá mô hình.

Scikit-learn cung cấp các mô-đun để xây dựng, điều chỉnh và đánh giá mô hình ML:

* *Preprocessing* đề cập đến các công cụ Scikit-learn hữu ích trong việc trích xuất và chuẩn hóa features trong quá trình phân tích dữ liệu.
* *Classification* đề cập đến một bộ công cụ xác định danh mục liên kết với dữ liệu trong mô hình học máy. Ví dụ: những công cụ này được sử dụng để phân loại email hợp lệ hoặc thư rác. Về cơ bản, Classification xác định đối tượng thuộc về loại nào.
* *Regression* (Phân tích hồi quy) đề cập đến việc tạo ra một mô hình học máy để tìm hiểu mối quan hệ giữa dữ liệu đầu vào và đầu ra, chẳng hạn như hành vi hoặc giá cổ phiếu. Hồi quy dự đoán một thuộc tính có giá trị liên tục được liên kết với một đối tượng.
* Các công cụ *Clustering* trong Scikit-learn tự động nhóm dữ liệu có đặc điểm tương tự thành các bộ (sets), chẳng hạn như dữ liệu khách hàng được sắp xếp theo bộ dựa trên vị trí thực tế.
* *Dimensionality Reduction* làm giảm số lượng biến ngẫu nhiên để phân tích. Ví dụ: để tăng hiệu quả trực quan hóa, dữ liệu bên ngoài có thể bị loại bỏ.
* *Model selection* đề cập đến các thuật toán và khả năng cung cấp các công cụ so sánh, xác thực và chọn tham số tối ưu để sử dụng trong các dự án học máy khoa học dữ liệu.
* *Pipeline*đề cập đến các tiện ích để xây dựng quy trình làm việc mô hình.
* *Visualization* cho học máy cho phép vẽ đồ thị và điều chỉnh trực quan nhanh chóng.

### 3.5 MTCNN

MTCNN (Multi-task Cascaded Convolutional Networks) là một kiến trúc Deep Learning chuyên để **phát hiện khuôn mặt** trong ảnh, kết hợp ba mạng con (P-Net, R-Net, O-Net) chạy tuần tự.

* **Ba bước chính**

1. **Proposal Network (P-Net)**: Quét ảnh ở nhiều kích thước, sinh ra các đề xuất vùng chứa mặt (candidate windows).
2. **Refine Network (R-Net)**: Loại bỏ hầu hết vùng không phải mặt, tinh chỉnh tọa độ hộp (bounding box).
3. **Output Network (O-Net)**: Chốt lại kết quả, xác định chính xác khuôn mặt và điểm landmark (mắt, mũi, miệng).

* **Multi-task Learning**
* Cùng lúc tối ưu hai nhiệm vụ: phát hiện hộp (face detection) và định vị landmark (facial landmark).
* Giúp mô hình phát hiện chính xác hơn so với các phương pháp truyền thống.

**CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH NHẬN DIỆN KHUÔN MẶT ĐIỂM DANH SINH VIÊN**

## 1. Phân tích bài toán cơ bản

Chương trình được xây dựng nhằm tự động hóa quá trình điểm danh sinh viên thông qua **nhận diện khuôn mặt bằng camera máy tính (webcam)**. Hệ thống bao gồm các tính năng chính:

* **Quét khuôn mặt qua webcam:** sử dụng thư viện OpenCV để mở camera và đọc từng khung hình. Dữ liệu khuôn mặt được trích xuất qua mô hình MTCNN.
* **Nhận diện khuôn mặt:** khuôn mặt được cắt từ ảnh và đưa qua mô hình học sâu CNN đã huấn luyện trước để xác định danh tính sinh viên.
* **Lưu kết quả vào cơ sở dữ liệu:** khi sinh viên được nhận diện thành công, hệ thống tự động lưu thông tin điểm danh (tên, thời gian check-in, trạng thái) vào MySQL thông qua DAO (Data Access Object).
* **Xử lý dữ liệu khuôn mặt ban đầu:** Chương trình hỗ trợ duyệt qua thư mục ảnh gốc, tự động phát hiện và trích xuất khuôn mặt từ từng ảnh, rồi lưu chúng phục vụ cho huấn luyện mô hình.

Ngoài các chức năng chính, chương trình còn đảm bảo:

* Giao diện sử dụng đơn giản, thân thiện với người dùng qua Tkinter.
* Hệ thống được tổ chức dễ bảo trì, tách biệt rõ ràng giữa phần xử lý AI, lưu trữ dữ liệu và giao diện người dùng.

.

## 2. Phân tích bài toán ứng dụng AI (Nhận diện hình ảnh)

**2.1 Mô hình nhận diện khuôn mặt**

Hệ thống sử dụng kết hợp hai thành phần chính:

* **MTCNN (Multi-task Cascaded Convolutional Networks):** để phát hiện vị trí khuôn mặt trong ảnh đầu vào.
* **CNN (Convolutional Neural Network):** để huấn luyện và nhận diện khuôn mặt đã được cắt ra.

**2.2 Các bước xây dựng hệ thống**

**Bước 1: Chuẩn bị dữ liệu**

* Dữ liệu đầu vào là các ảnh khuôn mặt của sinh viên, được đặt theo từng thư mục tương ứng với tên người dùng (định dạng: ten\_maSV).
* Sử dụng MTCNN để phát hiện khuôn mặt, cắt và lưu lại dưới định dạng chuẩn 160x160.
* Gắn nhãn (label) là tên người dùng để phục vụ huấn luyện.

**Bước 2: Huấn luyện mô hình CNN**

* Mỗi ảnh khuôn mặt được chuẩn hóa (resize và chia giá trị pixel về khoảng [0, 1]).
* Mô hình CNN gồm 3 lớp tích chập (Conv2D), kết hợp với MaxPooling, Flatten và Dense.
* Hàm kích hoạt relu, tầng đầu ra sử dụng softmax tương ứng với số lớp (số sinh viên).
* Huấn luyện với Adam optimizer và sparse\_categorical\_crossentropy.

**Bước 3: Lưu mô hình**

* Sau khi huấn luyện xong, mô hình được lưu lại dưới dạng .h5 để sử dụng sau này.
* Các nhãn (labels) cũng được lưu vào file labels.npy.

**2.3 Nhận diện từ webcam và điểm danh**

* Camera đọc khung hình liên tục và chuyển sang ảnh RGB.
* Sử dụng MTCNN để phát hiện khuôn mặt từ ảnh.
* Cắt và resize khuôn mặt về 160x160, chuẩn hóa và đưa vào mô hình CNN đã huấn luyện.
* Mô hình dự đoán ra tên người dùng và độ tin cậy.
* Nếu độ tin cậy cao và tên chưa được nhận diện trong phiên hiện tại → lưu điểm danh vào cơ sở dữ liệu MySQL.

**2.4 Cấu trúc lưu điểm danh**

Thông tin lưu vào DB bao gồm:

* id: mã điểm danh (sinh tự động bằng uuid)
* student\_id: mã sinh viên nhận diện được
* checkin\_time: thời gian điểm danh
* scheduledetail\_id: ID của lịch học
* status: trạng thái điểm danh (1: có mặt)

# CHƯƠNG 3: CÀI ĐẶT, DEMO CHƯƠNG TRÌNH

## 1. Cài đặt

Môi trường thực hiện train và các bước thực hiện:

Môi trường: PyCharm, Visual Studio Code

* Trong quá trình xây dựng hệ thống quản lý và điểm danh sinh viên bằng khuôn mặt, nhóm đã sử dụng nhiều thư viện của Python để phục vụ cho việc xử lý giao diện, xử lý ảnh, kết nối cơ sở dữ liệu, huấn luyện mô hình học sâu, và nhận diện khuôn mặt. Dưới đây là hướng dẫn cài đặt các thư viện cần thiết:

Cài đặt thư viện bằng pip:

pip install mysql-connector-python

pip install opencv-python

pip install numpy

pip install mtcnn

pip install tensorflow

pip install scikit-learn

* Đối với các thư viện tiêu chuẩn (không cần cài đặt):
  + os, sys (quản lý hệ thống, đường dẫn tệp)
  + datetime (làm việc với ngày giờ)
  + tkinter, tkinter.messagebox (thư viện GUI mặc định của Python)

Các bước train:

* Chuẩn bị thư viện và dữ liệu:
* **Thư viện**
  + Sử dụng os để duyệt cấu trúc thư mục của tập ảnh đã cắt (faces/).
  + Dùng OpenCV (cv2) để đọc, resize và chuẩn hóa pixel từ [0–255] về [0–1].
  + NumPy (np) lưu trữ mảng ảnh và mảng nhãn.
  + TensorFlow/Keras xây dựng, huấn luyện và lưu mô hình CNN.
  + scikit-learn cung cấp LabelEncoder để mã hóa nhãn chuỗi thành số và train\_test\_split để chia tập dữ liệu.
* **Dữ liệu**
  + Tập dữ liệu đầu vào gồm thư mục faces/, bên trong có một thư mục con cho mỗi user\_id chứa ảnh khuôn mặt đã phát hiện và cắt ra. Nên để tối thiểu 20 ảnh, ảnh càng nhiều độ chuẩn sát sẽ càng cao.
  + Ảnh được chuẩn hóa về kích thước 160×160 và giá trị pixel chia 255.
* Tiền xử lý và phân tách nhãn:
  + **Load ảnh**: Lặp qua từng thư mục người dùng, đọc lần lượt các ảnh và resize.
  + **Chuẩn hóa**: Chuyển đổi ảnh sang dạng float, chia 255 để đưa về phân phối [0–1].
  + **Mã hóa nhãn**:
* Dùng LabelEncoder để chuyển user\_id (chuỗi) thành chỉ số số nguyên.
* Lưu mảng lớp (classes\_) ra file labels.npy để sử dụng khi dự đoán.
* Chia tập huấn luyện – kiểm tra:

+ Chuyển đổi data và labels từ danh sách thành mảng numpy bằng cách sử dụng np.array().

* Chia dữ liệu thành tập huấn luyện và tập kiểm tra:

+ Chia tập thành tỉ lệ 80% cho huấn luyện - 20% để kiểm tra.

+ Dữ liệu được chia ngẫu nhiên nhưng cố định nhờ random\_state=42, đảm bảo kết quả lặp lại được.

* Xây dựng mô hình:

+ Tạo một đối tượng mô hình Sequential bằng cách sử dụng model = Sequential().

+ Thêm các lớp Conv2D, MaxPool2D, Dropout, Flatten, và Dense vào mô hình theo kiến trúc đã được định nghĩa.

+ Lớp cuối cùng sử dụng softmax activation để đưa ra dự đoán cho 43 lớp.

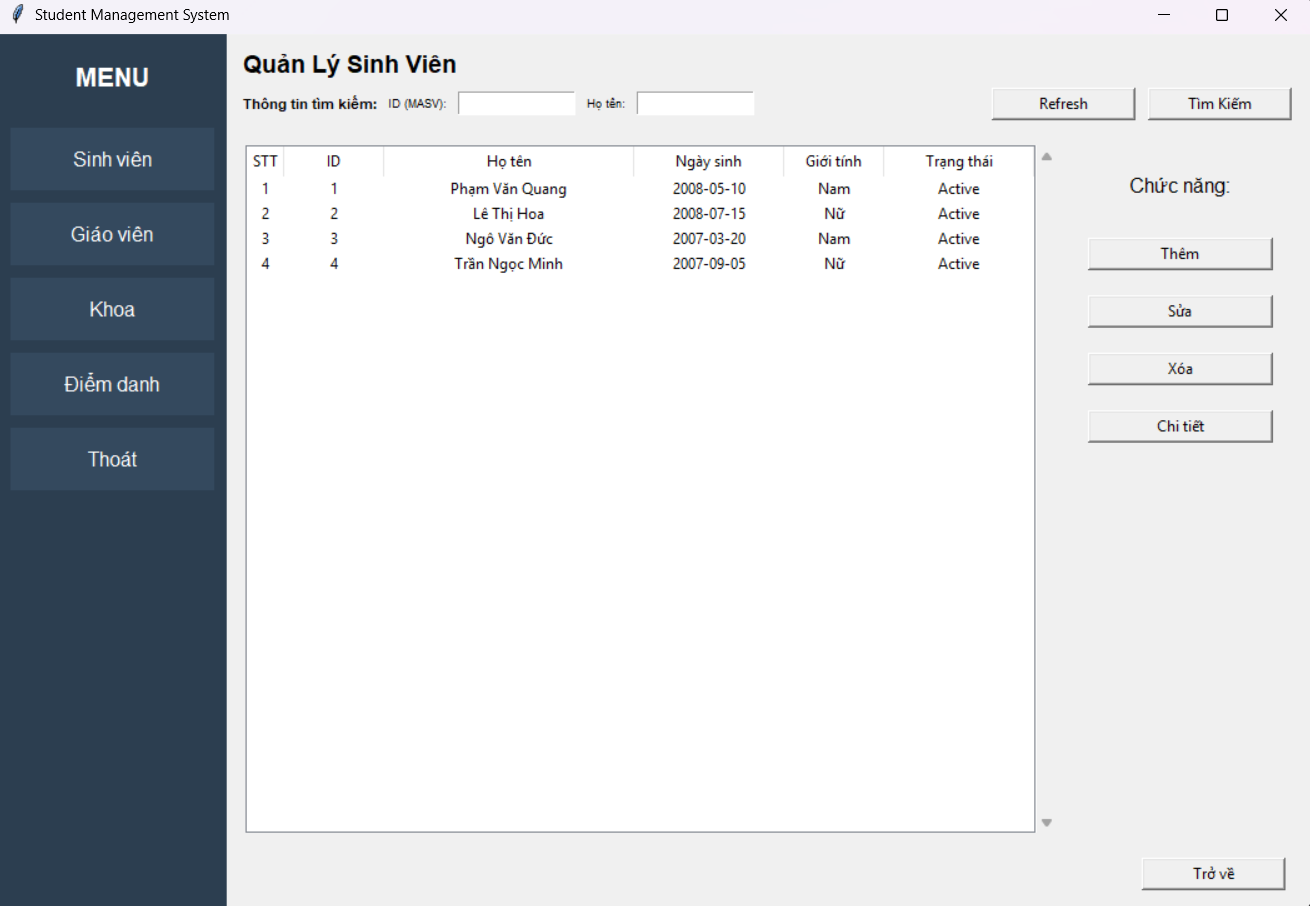
* **Định nghĩa kiến trúc CNN**

Kiến trúc tuần tự (Sequential) bao gồm:

* **Chuỗi tầng tích chập (Conv2D) và pooling (MaxPooling2D)**
  + Khai thác đặc trưng cục bộ từ ảnh (các cạnh, góc, vùng có độ tương phản cao).
  + Giảm dần kích thước không gian đồng thời tăng độ sâu kênh.
* **Tầng phẳng hóa (Flatten)**
  + Biến đổi tensor 3 chiều thành vector 1 chiều để nhập vào lớp kết nối đầy đủ.
* **Các lớp dày đặc (Dense) kèm Dropout**
  + Dense 256 neuron với kích hoạt ReLU học mối quan hệ phi tuyến tính giữa đặc trưng.
  + Dropout 0.5 ngăn chặn over-fitting.
  + Lớp đầu ra dùng softmax với số nút bằng số người dùng (số lớp).
* **Biên dịch mô hình**
* **Hàm mất mát**: sparse\_categorical\_crossentropy phù hợp với nhãn nguyên.
* **Bộ tối ưu**: Adam giúp hội tụ nhanh trên các mạng sâu.
* **Metric**: accuracy để theo dõi độ chính xác phân lớp.
* **Huấn luyện mô hình**
* **Epochs**: Mô hình lặp qua toàn bộ tập dữ liệu 20 lần.
* **Batch size**: Mỗi lần cập nhật trọng số dùng 16 ảnh.
* **Validation**: Sau mỗi epoch, đánh giá trên tập kiểm tra để theo dõi việc generalization (độ khái quát).
* **Lưu mô hình sau huấn luyện**
* Mô hình hoàn chỉnh được lưu dưới dạng file **face\_recognition\_model.h5**.
* File nhãn labels.npy và file mô hình này sẽ được nạp lại trong bước triển khai nhận diện (file webcam.py).

## 2. Demo chương trình

- Sinh viên:



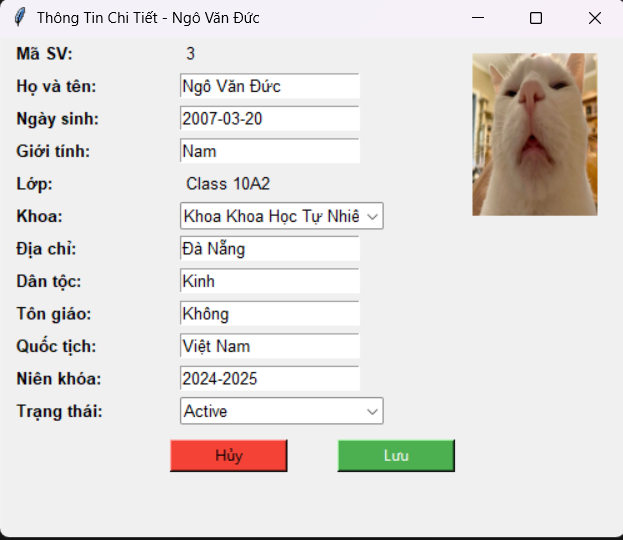
**Hình 1. Quản lý sinh viên**

Hiển thị danh sách sinh viên với những thông tin cơ bản như mã sinh viên, họ tên, ngày sinh, giới tính, trạng thái của sinh viên đó.



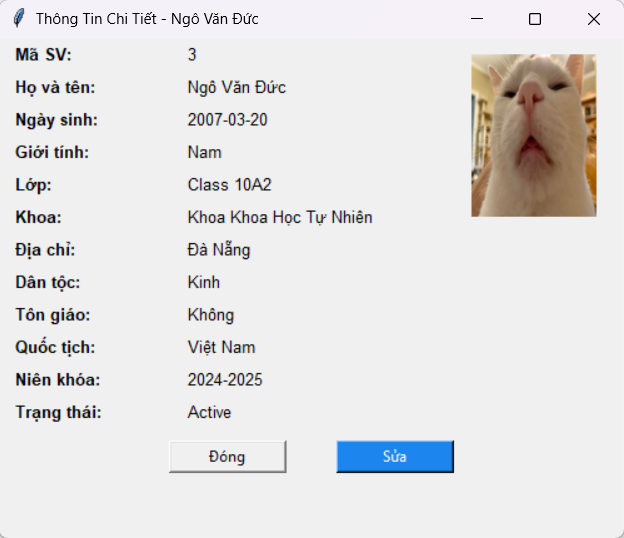
**Hình 2: Thêm sinh viên**

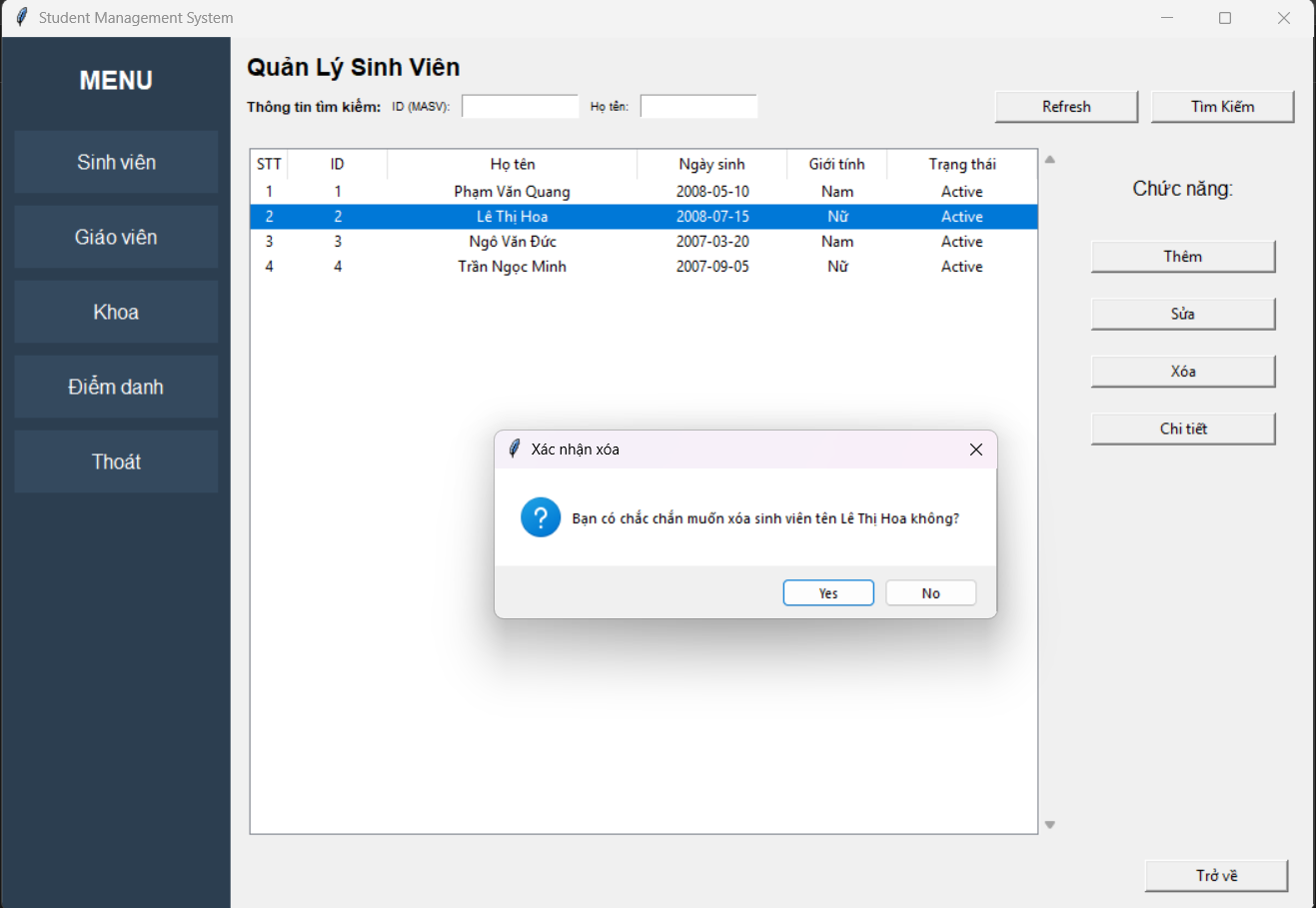
Hiển thị các thông tin sinh viên cần để thêm vào danh sách sinh viên bao gồm các thông tin cơ bản của sinh viên.



**Hình 3: Sửa sinh viên**

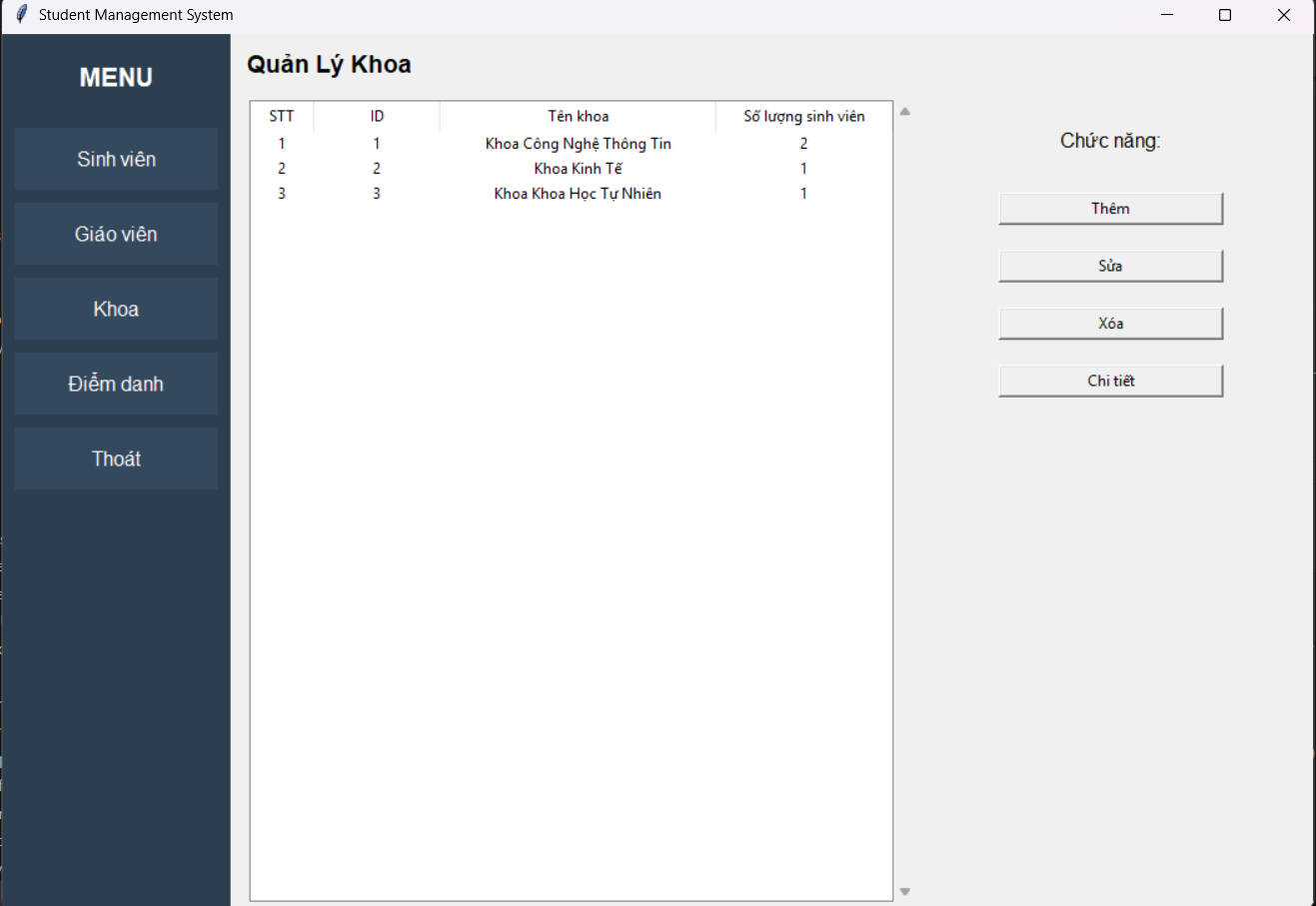
Khi ấn vào một dòng trong bảng bất kỳ và ấn sửa. Thì đây là các thông tin của sinh viên cần sửa



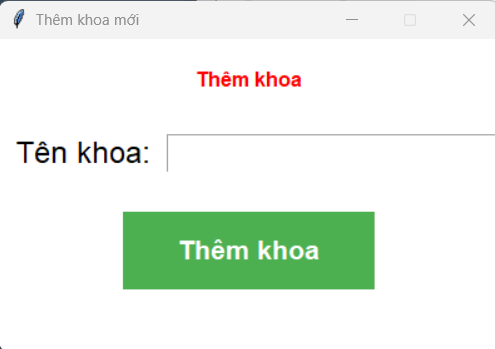
**Hình 4: Xem chi tiết sinh viên. **

**Hình 5: Xóa sinh viên.**

* Khoa:



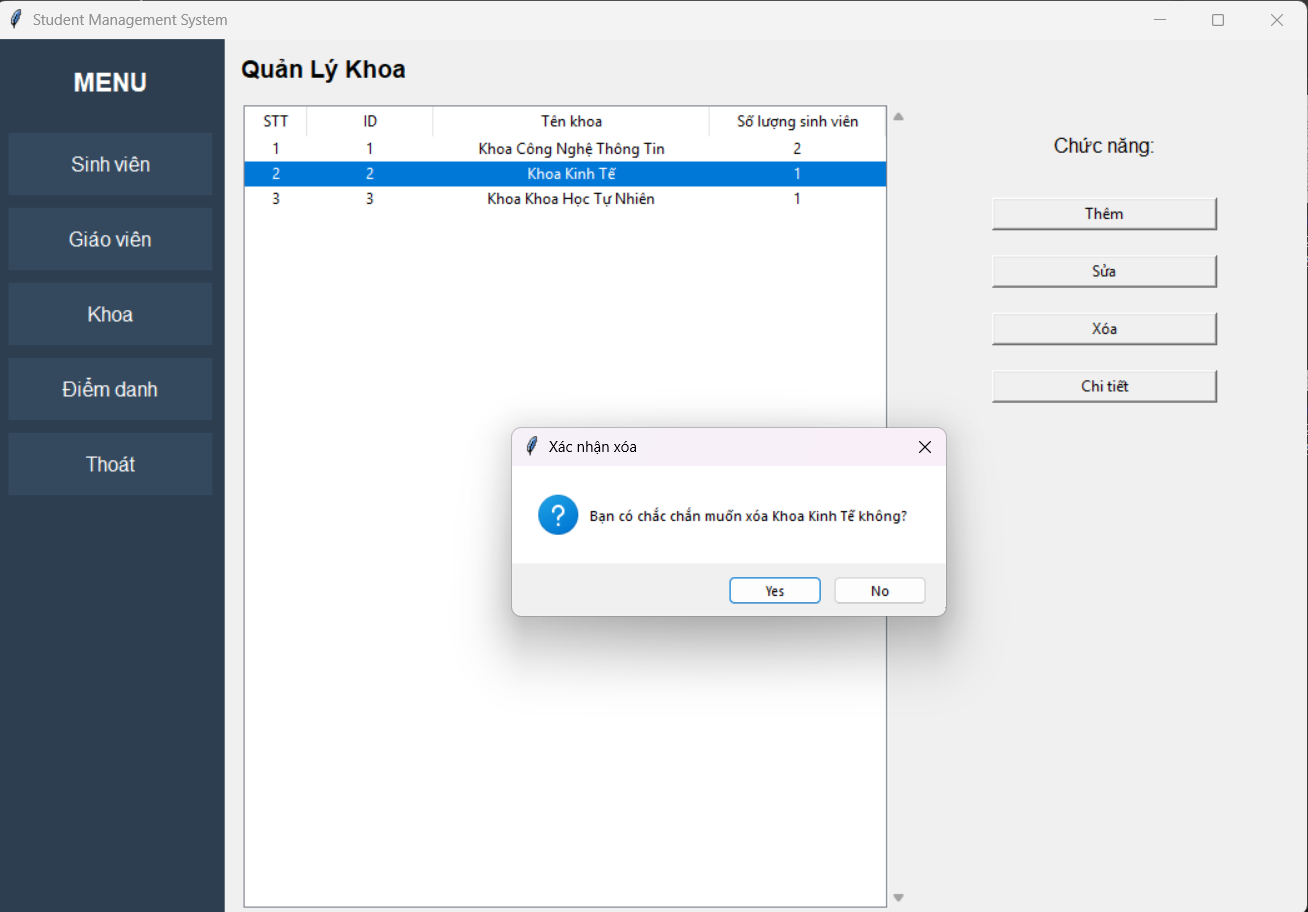
**Hình 6: Danh sách khoa**

****

**Hình 7: Thêm khoa**

****

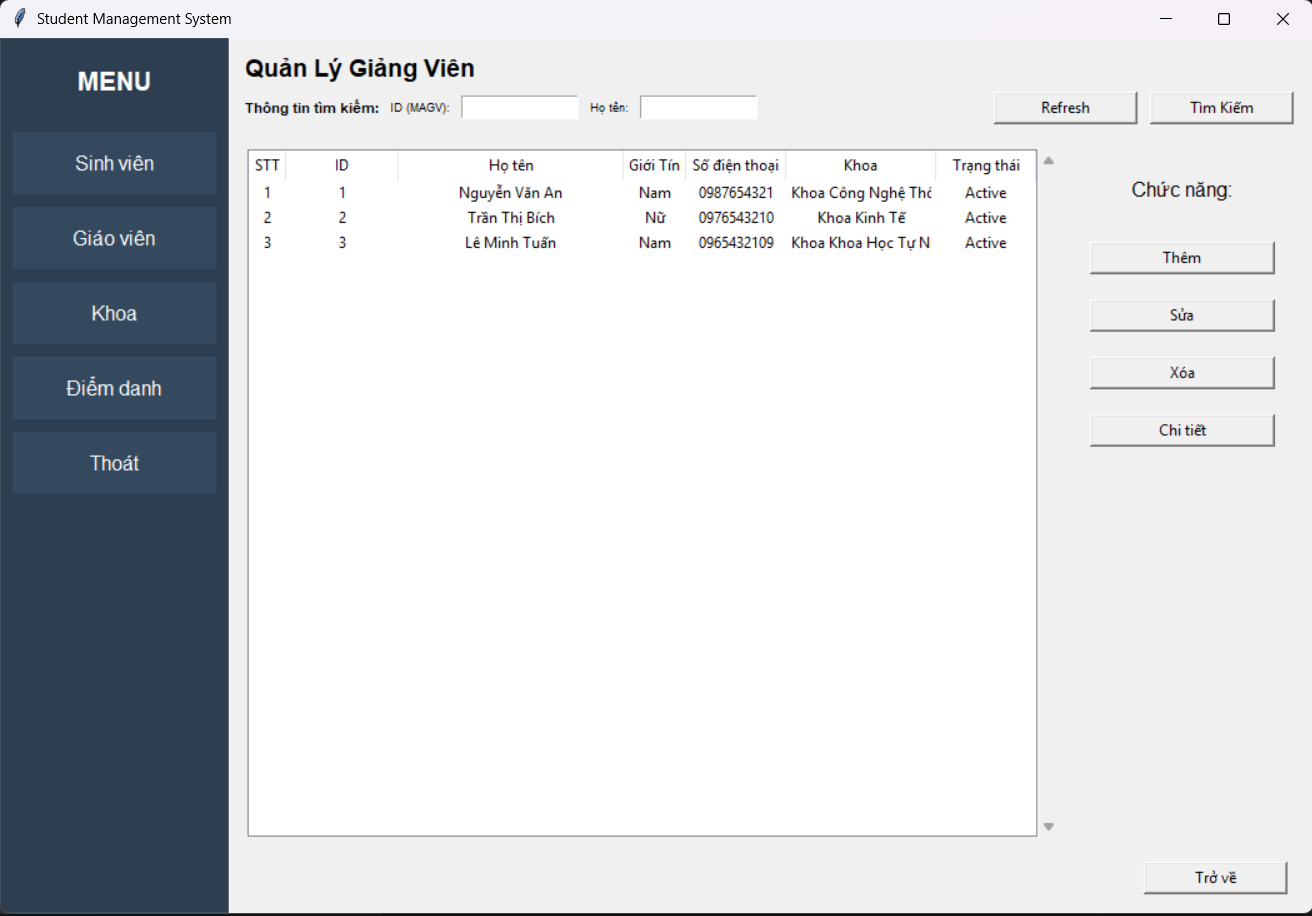
**Hình 8: Cập nhật thông tin khoa**

****

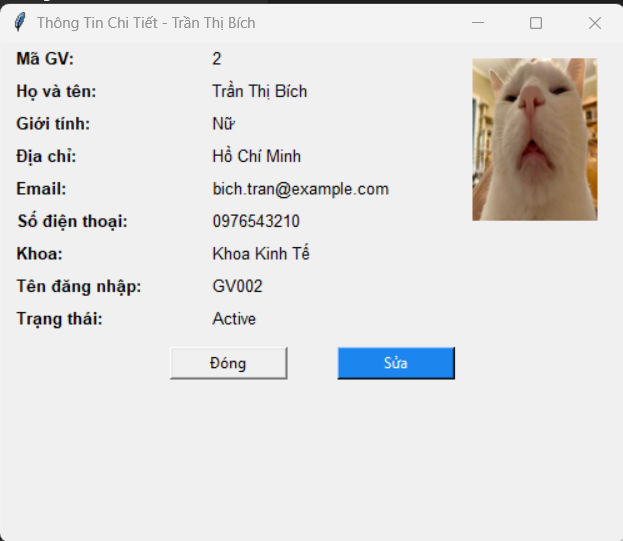
**Hình 9: Xóa khoa**

Việc xóa khoa sẽ phụ thuộc vào số lượng sinh viên. Chỉ được xóa khi khoa đó không có sinh viên.

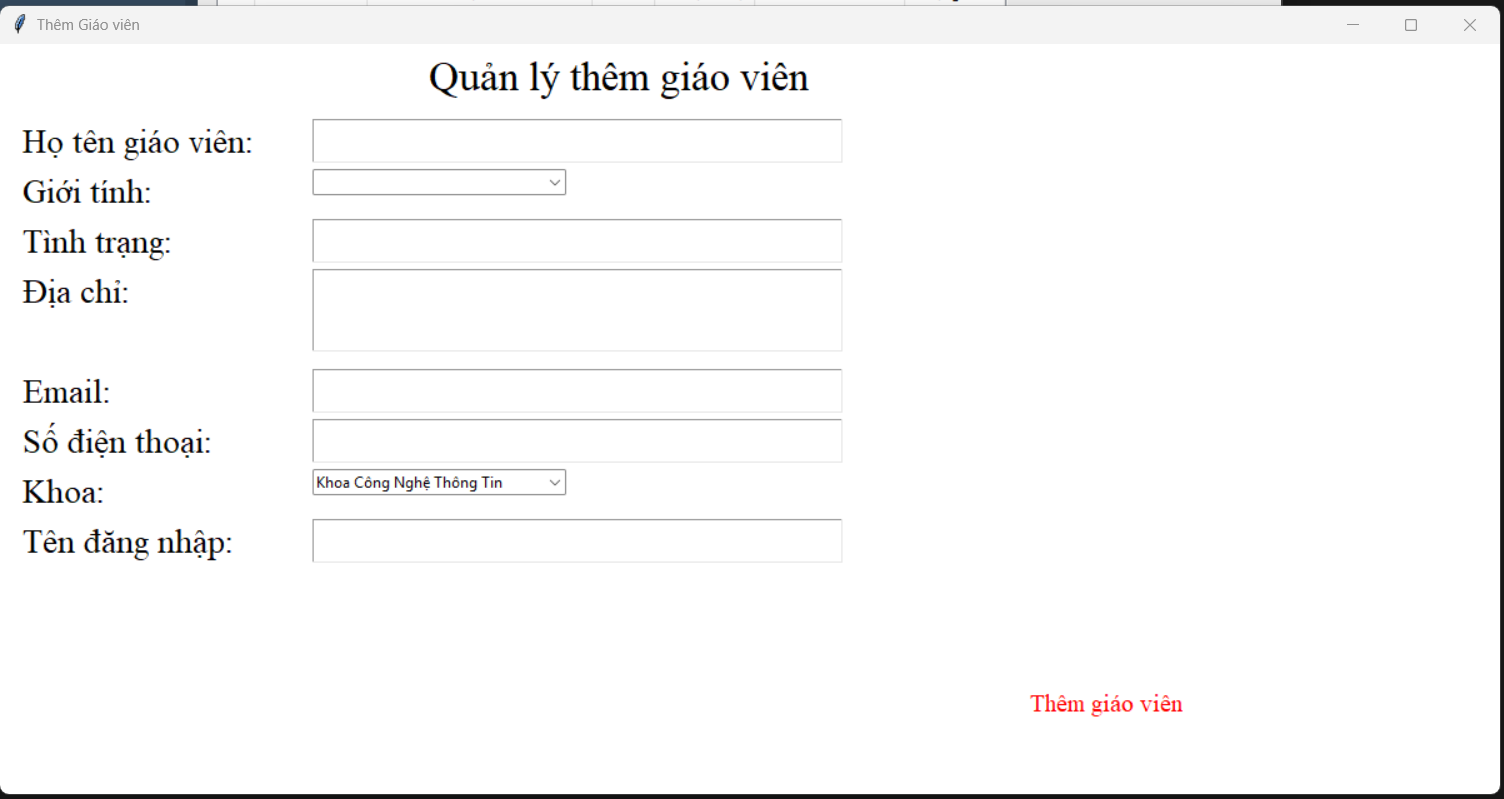
* Giáo viên:



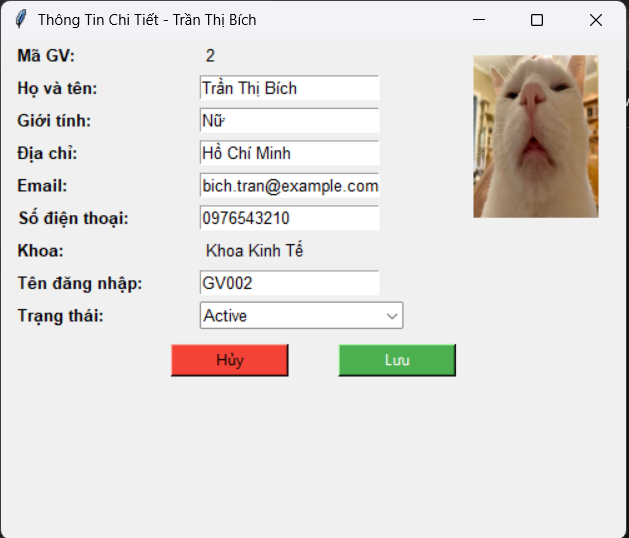
**Hình 10: Danh sách giảng viên**

****

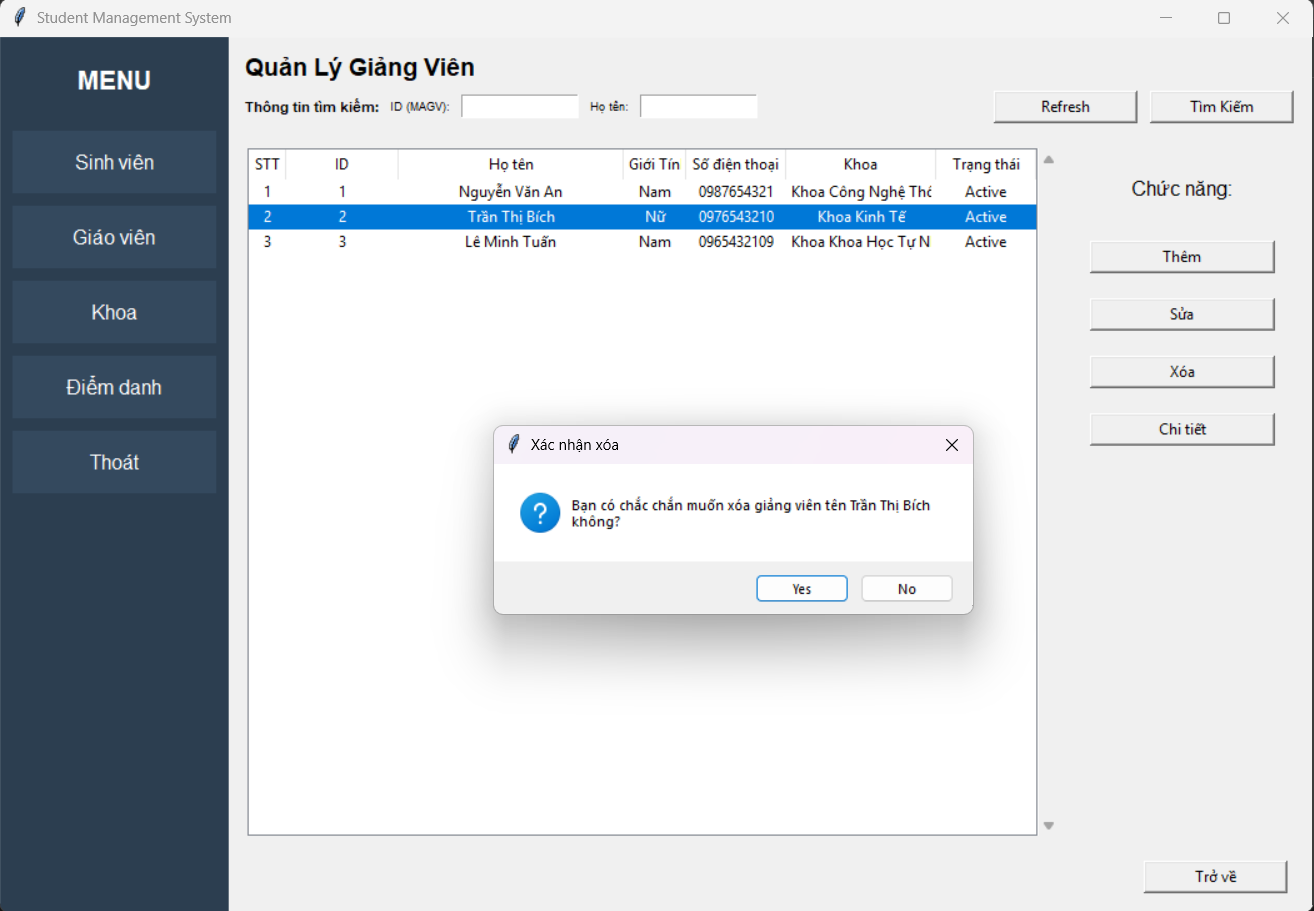
**Hình 11: Xem chi tiết giảng viên**

****

**Hình 12: Thêm mới giảng viên**

****

**Hình 13: Sửa giảng viên**

****

**Hình 14: Xóa thông tin giảng viên**

# KẾT LUẬN

## ĐÁNH GIÁ NHỮNG NỘI DUNG ĐÃ THỰC HIỆN

**-**Thiết kế hệ thống bằng Tkintercó giao diện ưa nhìn, dễ dàng sử dụng, trực quan

**-** Nhận diện được các khuôn mặt có tỉ lệ tương đối

**-** Thực hiện được các chức năng cơ bảng trong hệ thống và nắm được cơ bản mô hình gui bus dao.

**-** Thao tác được với CSDL

## ĐÁNH GIÁ NHỮNG NỘI DUNG CHƯA THỰC HIỆN

**-** Một số tính năng còn hạn chế, quy mô hệ thống nhỏ

**-** Giao diện hệ thống đơn giản thành ra còn sơ sài

**-** Bị hạn chế trong việc khắt phục tình trạng dùng ảnh, video để nhận diện

- Hệ thống chưa tối ưu về tốc độ, và hiệu năng. Có một số khung bị đóng ngây lập tức khi mở trên một số máy

## HƯỚNG PHÁT TRIỂN

* + Hoàn thiện chương trình:

+ Cải thiện giao diện hệ thống.

+ Cải thiện chức năng điểm danh chống gian lận.

+ Mở rộng thêm những tính năng hỗ trợ cho người dùng thuận tiện trong việc quản lý số lượng lớn hơn như: các module quản lý điểm, lịch học, thông tin đăng ký môn, yêu cầu nhận xét của sinh viên,…

+ Tìm hiểu để kết nối và tối ưu cơ sở dữ liệu lớn.

+ Linh hoạt trong việc train khuôn mặt và nhận diện.