BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**THAM GIA XÉT GIẢI THƯỞNG**

**XÂY DỰNG ỨNG DỤNG THẺ SINH VIÊN ĐIỆN TỬ**

**Mã số đề tài : UTEHY.S.2022.83**

Thuộc nhóm ngành khoa học:

**Hưng Yên, 2022**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**THAM GIA XÉT GIẢI THƯỞNG**

**XÂY DỰNG ỨNG DỤNG THẺ SINH VIÊN ĐIỆN TỬ**

**Mã số đề tài : UTEHY.S.2022.83**

Thuộc nhóm ngành khoa học:

Sinh viên thực hiện: CẤN CÔNG CƯỜNG Nam, Nữ: Nam

Dân tộc: Kinh

Lớp, khoa: 101195, Công nghệ thông tin

Năm thứ/Số năm đào tạo: 3/4

Ngành học: Lập trình di động.

Người hướng dẫn: Ths. NGUYỄN HOÀNG ĐIỆP

**Hưng Yên, 10/04/2022**

MỤC LỤC

[DANH MỤC BẢNG BIỂU 5](#_Toc103271590)

[DANH SÁCH HÌNH VẼ 6](#_Toc103271591)

[DANH MỤC NHỮNG TỪ VIẾT TẮT 7](#_Toc103271592)

[CHƯƠNG 1. MỞ ĐẦU 12](#_Toc103271596)

[1.1. Lý do chọn đề tài 12](#_Toc103271597)

[1.2. Mục tiêu của đề tài 12](#_Toc103271598)

[1.3. Nội dung thực hiện đề tài 12](#_Toc103271606)

[1.4. Phương pháp tiếp cận đề tài 13](#_Toc103271607)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT THỰC HIỆN ĐỀ TÀI 14](#_Toc103271608)

[2.1. Quy trình phát triển phần mềm di động 14](#_Toc103271609)

[2.2. Phương pháp phân tích thiết kế hướng đối tượng 15](#_Toc103271612)

[2.3. Thư viện hỗ trợ kết nối và trao đổi dữ liệu 16](#_Toc103271615)

[2.4 Giới thiệu - Pytorch 17](#_Toc103271619)

[2.5. Nền tảng toán học của Neural Networks 20](#_Toc103271624)

[2.6. Alignment 23](#_Toc103271628)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG PHẦN MỀM 45](#_Toc103271637)

[3.1. Xác định yêu cầu phần mềm 45](#_Toc103271638)

[3.1.1. Khảo sát yêu cầu 45](#_Toc103271639)

[3.1.2. Phân tích yêu cầu phần mềm 46](#_Toc103271640)

[3.1.2. Biểu đồ lớp 48](#_Toc103271641)

[3.1.3. Biểu đồ tuần tự 52](#_Toc103271642)

[3.1.4. Các yêu cầu phi chức năng 52](#_Toc103271643)

[CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÁNH GIÁ 54](#_Toc103271644)

[4.1. Các kết quả đã đạt được 54](#_Toc103271645)

[4.1.1. Về các chức năng phần mềm 54](#_Toc103271646)

[4.1.2. Về tính ứng dụng của phần mềm vào thực tế 56](#_Toc103271647)

[4.2. Đánh giá các kết quả đã đạt được 57](#_Toc103271648)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 58](#_Toc103271649)

[5.1. Kết luận về đề tài 58](#_Toc103271650)

[5.1.1. Về ưu điểm đã đạt được 58](#_Toc103271651)

[5.1.2. Về một số hạn chế của phần mềm 58](#_Toc103271652)

[5.2. Một số kiến nghị về phần mềm 59](#_Toc103271653)

[5.3. Hướng phát triển của đề tài 59](#_Toc103271654)

[TÀI LIỆU THAM KHẢO 60](#_Toc103271655)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 Bảng phân tích các yêu cầu phần mềm 30](#_Toc103272259)

[Bảng 3.2 Cấu trúc bảng lớp 34](#_Toc103272260)

[Bảng 3.3 Cấu trúc bẳng tin tức 34](#_Toc103272261)

[Bảng 3.4 Cấu trúc bảng sinh viên 34](#_Toc103272262)

[Bảng 3.5 Cấu trúc bảng giáo viên 35](#_Toc103272263)

[Bảng 3.6 Cấu trúc bảng điểm thi 35](#_Toc103272264)

[Bảng 3.7 Cấu trúc bảng học phần 35](#_Toc103272265)

[Bảng 3.8 Cấu trúc bảng lịch học 35](#_Toc103272266)

[Bảng 3.9 Cấu trúc bảng lịch thi 36](#_Toc103272267)

# DANH SÁCH HÌNH VẼ

[Hình 3.1. Sơ đồ use-case mức độ tổng quát của hệ thống 32](#_Toc103272302)

[Hình 3.2. Biểu đồ lớp 33](#_Toc103272303)

[Hình 3.3. Biểu đồ tuần tự cập nhật hồ sơ cá nhân 37](#_Toc103272304)

[Hình 4.1. Giao diện khởi động của phần mềm 39](#_Toc103272305)

[Hình 4.2. Giao diện đăng nhập 39](#_Toc103272306)

[Hình 4.3. Giao diện trang chủ 40](#_Toc103272307)

[Hình 4.4. Giao diện xem thành tích cá nhân 41](#_Toc103272308)

[Hình 4.5. Giao diện xem lịch học, lịch thi 42](#_Toc103272309)

[Hình 4.6. Giao diện tài khoản 43](#_Toc103272310)

DANH MỤC NHỮNG TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Cụm từ tiếng anh | Diễn giải |
| 1 | IT | Information Technology | Công nghệ Thông tin |
| 2 | DB | DataBase | Cơ sở dữ liệu |
| 3 | SQL | Structured Query Language | Ngôn ngữ truy vấn cấu trúc |
| 4 | UTEHY | University of Technology and Education Hung Yen | Trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên |

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

# THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

**1. Thông tin chung:**

* Tên đề tài: Xây dựng ứng dụng thẻ sinh viên điện tử TSVID.
* Sinh viên thực hiện: Cấn Công Cường, Đỗ Thành Tôn, Vũ Xuân Thiện, Đỗ Quang Hào, Nguyễn Văn Quang.
* Lớp: 101195
* Khoa: Công nghệ thông tin.
* Năm thứ: 3
* Số năm đào tạo: 4
* Người hướng dẫn: Ths. Nguyễn Hoàng Điệp.

**2. Mục tiêu đề tài:**

* Tạo được ứng dụng giúp các sinh viên trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên có thể dễ dàng cập nhật thông tin về bản thân và nhà trường một cách nhanh chóng.
* Giúp các bạn học sinh có thể sử dụng thẻ sinh viên online thay cho thẻ cứng.
* Tạo được ứng dụng dành riêng cho sinh viên trường ĐHSPKT Hưng Yên.
* -Có thể xem lịch học, lịch thi, bảng điểm của mỗi người.

**3. Tính mới và sáng tạo:**

* Áp dụng công nghệ Firebase vào dự án thực tế.
* Sinh viên UTEHY dễ dàng cập nhật thông tin chỉ với một chiếc điện thoại thông minh.
* Là ứng dụng dành riêng cho sinh viên UTEHY.
* Quản lý thông tin cá nhân, lịch học, gửi thông báo, … dễ dàng hơn với ứng dụng này.

**4. Kết quả nghiên cứu:**

* Hoàn thành được các chức năng cơ bản của phân hệ quản trị viên hệ thống, phân hệ sinh viên
* Phân hệ quản trị viên hệ thống
* Quản lý tài khoản chung: thêm tài khoản sinh viên, cấp quyền người dùng.
* Quản lý điểm: thêm môn học, chỉnh sửa thông tin môn học.
* Quản lý lịch học: thêm lịch học, chỉnh sửa thông tin lớp học.
* Gửi thông báo: gửi thông báo tất cả các lớp, gửi thông báo tới 1 lớp.
* Phân hệ sinh viên
* Xem bảng tin trường.
* Xem hoạt động trường.
* Truy cập cổng thông tin sinh viên.
* Xem thông báo lớp.
* Xem điểm.
* Xem lịch học
* Xem lịch thi
* Cập nhật thông tin tài khoản.
* Ứng dụng demo được vào các lớp học thực tế và đã tạo được một số ý nghĩa tích cực nhất định.
* Ứng dụng được cập nhật dữ liệu nhanh chóng thông qua lưu trữ dữ liệu bằng Google Firebase.

**5. Đóng góp về mặt kinh tế - xã hội,** **giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:**

Phần mềm được ứng dụng trong môi trường Trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên với những tác động tích cực trong việc cập nhật thông tin giữa các sinh viên trong trường, việc theo dõi thời khóa biểu và lịch hoạt động luôn được cập nhật một cách dễ dàng và nhanh chóng.

Các sinh viên giờ đây có thể dung thẻ điện tử để thay thế cho thẻ cứng mỗi khi quên mang đi.

**6.** **Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài.**

Ngày 10 tháng 04 năm 2022

|  |  |
| --- | --- |
|  | **Sinh viên chịu trách nhiệm chính**  **thực hiện đề tài**  **CẤN CÔNG CƯỜNG** |

**Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài:**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Ngày 10 tháng 04 năm 2022

|  |  |
| --- | --- |
| **Xác nhận của trường đại học** | **Người hướng dẫn**  **NGUYỄN HOÀNG ĐIỆP** |

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

# THÔNG TIN VỀ SINH VIÊN

# CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI

**I. SƠ LƯỢC VỀ SINH VIÊN:**

Ảnh 4x6

Họ và tên: CẤN CÔNG CƯỜNG

Sinh ngày: 20 tháng 07 năm 2001

Nơi sinh: xã Phụng Thương, huyện Phúc Thọ, thành phố Hà Nội.

Lớp: 101195 Khóa: 2019-2023

Khoa: Công nghệ thông tin

Địa chỉ liên hệ: xã Phụng Thương, huyện Phúc Thọ, thành phố Hà Nội.

Điện thoại: +8438 632 8935

Email: congcuong108@gmail.com

**II. QUÁ TRÌNH HỌC TẬP**

***\* Năm thứ 1:***

Ngành học: Lập trình di động. Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: Giỏi

Sơ lược thành tích:

***\* Năm thứ 2:***

Ngành học: Lập trình di động Khoa: Công nghệ thông tin

Kết quả xếp loại học tập: Giỏi

Sơ lược thành tích:

Ngày tháng năm

|  |  |
| --- | --- |
| **Xác nhận của trường đại học** | **Sinh viên chịu trách nhiệm chính**  **thực hiện đề tài**  **CẤN CÔNG CƯỜNG** |

CHƯƠNG 1. MỞ ĐẦU

# Lý do chọn đề tài

Ngày nay, cùng với sự phát triển về mọi mặt của xã hội, ngành công nghệ thông tin đã đóng góp một phần không thể thiếu trong cuộc sống của mỗi con người.

Nền khoa học máy tính ngày nay đang giữ một vai trò chủ đạo trong hầu hết các lĩnh vực của xã hội.

Chính vì vậy, nhóm đã chọn đề tài “Xây dựng ứng dụng thẻ sinh viên điện tử TSVID” nhằm giúp các sinh viên trong trường dễ dàng quản lý các thông tin liên quan đến bản thân và có thẻ dung thẻ điện tử thay cho thẻ cứng khi các bạn quên không mang theo.

# Mục tiêu của đề tài

## Mục tiêu tổng quát

Ứng dụng hỗ trợ thông tin cho sinh viên trường ĐHSPKT Hưng Yên được tạo ra nhằm mục đích giúp các sinh viên trong trường có sử dụng thẻ điện tử thay thế cho thẻ cứng.

## Phạm vi nghiên cứu

1. **Phạm vi không gian:**

Ứng dụng được nghiên cứ trong phạm vi không gian trường Đại học Sư Phạm Kỹ Thuật Hưng Yên.

1. **Phạm vi thời gian:**

Ứng dụng được nghiên cứu được thu thập thông tin và số liệu vào tháng 4 và tháng 3 năm 2022.

Với các mục đích đã nêu ở trên, ứng dụng này mong muốn đạt được các ý nghĩa trong thực tiễn giúp đơn giản hóa việc liên lạc qua lại giữa nhà trường và sinh viên.

# Nội dung thực hiện đề tài

- Tổng quan về đề tài nghiên cứu

- Trình bày cơ sở lý thuyết xây dựng ứng dụng.

- Phân tích và thực hiện thiết kế hệ thống.

- Triền khai ứng dụng.

# Phương pháp tiếp cận đề tài

Ứng dụng được lên ý tưởng và xây dựng dựa theo các ứng dụng đã có trên CH Play. Tất nhiên, ứng dụng nào cũng có ưu và nhược điểm và ứng dụng này cũng không phải ngoại lệ.

* Về ưu điểm, ứng dụng có thể giúp sinh viên trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên có thể dễ dàng gửi thông tin và cập nhật thông tin.
* Về nhược điểm, UTEHY app là một ứng dụng được lập trình cá nhân, chưa được sự tham gia của nhiều người nên còn khá đơn giản và chưa bảo mật tối ưu.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT THỰC HIỆN ĐỀ TÀI

# 2.1. Quy trình phát triển phần mềm di động

## 2.1.1. Giới thiệu quy trình phát triển phần mềm

Cũng như các ngành sản xuất khác, quy trình là một trong những yếu tố đầu tiên và cực kỳ quan trọng đem lại thành công cho các nhà phát triển phần mềm, nó giúp cho mọi thành viên trong dự án từ người cũ đến người mới, trong hay ngoài công ty đều có thể xử lý đồng bộ công việc tương ứng với trị trí của tôi thông qua cách thức chung của công ty. Có thể nói, quy trình phát triển phần mềm có tính chất quyết định để tạo ra 1 sản phẩm có chi phí thấp và năng suất cao.

Quy trình phát triển phần mềm là một cấu trúc bao gồm tập hợp các thao tác và các kết quả tương quan sử dụng trong việc phát triển để sản xuất ra một sản phẩm phần mềm.

## 2.1.2. Các bước phát triển một phần mềm di động

1. **Giải pháp, yêu cầu**

* Nhiệm vụ: Thực hiện khảo sát chi tiết yêu cầu của khách hàng để từ đó tổng hợp vào tài liệu giải pháp. Tài liệu này phải mô tả đầy đủ các yêu cầu về chức năng, phi chức năng và giao diện.
* Kết quả: Tài liệu đặc tả yêu cầu.

1. **Thiết kế**

* Nhiệm vụ: Thực hiện thiết kế và tổng hợp vào tài liệu thiết kế.
* Kết quả: Tài liệu thiết kế tổng thể, thiết kế module, thiết kế CSDL.

1. **Lập trình**

* Nhiệm vụ: Lập trình viên thực hiện lập trình dựa trên tài liệu Giải pháp và Thiết kế đã được phê duyệt.
* Kết quả: Source Code.

1. **Kiểm thử**

* Nhiệm vụ: Tester tạo kịch bản kiểm thử (testcase) theo tài liệu đặc tả yêu cầu, thực hiện kiểm thử và cập nhật kết quả vào kịch bản kiểm thử, log lỗi trên các tool quản lý lỗi.
* Kết quả: Testcase, lỗi trên hệ thống quản lý lỗi.

1. **Triển khai**

* Nhiệm vụ: Triển khai sản phẩm cho khách hàng.
* Kết quả: Biên bản triển khai với khách hàng.

# 2.2. Phương pháp phân tích thiết kế hướng đối tượng

## 2.2.1. Tổng quan về ngôn ngữ UML

1. **Ngôn ngữ UML là gì?**

UML (Unified Modeling Language) là ngôn ngữ dành riêng cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm. UML tạo cơ hội để viết thiết kế hệ thống, bao gồn những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống. Cụ thể, nó hữu dụng cho những ngôn ngữ khai báo, giản đồ cơ sở dữ liệu, thành phần phần mềm có khả năng tái sử dụng.

UML được phát triển bởi Rational Rose và một số nhóm cộng tác, nó nhanh chóng trở thành một trong những ngôn ngữ chuẩn để xây dựng hệ thống phần mềm hướng đối tượng (Object-Oriented). Đây là ngôn ngữ kế vị xứng đáng cho những ngôn ngữ mô hình hóa như Booch, OOSE/Jacobson, OMT và một số các phương thức khác.

1. **Mục tiêu của ngôn ngữ UML**

* Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
* Cung cấp khả năng mở rộng và chuyên môn hóa để mở rộng những khái niệm cốt lõi.
* Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
* Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hóa.
* Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
* Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern and component.

## 2.2.2. Các bước phân tích thiết kế hướng đối tượng

Thiết kế hướng đối tượng (Object Oriented Design - OOD) là giai đoạn tổ chức chương trình thành các tập hợp đối tượng cộng tác, mỗi đối tượng trong đó là thực thể của một lớp. Các lớp là thành viên của một cây cấu trúc với mối quan hệ kế thừa.

Mục đích của giai đoạn OOD là tạo thiết kế dựa trên kết quả của giai đoạn OOA, dựa trên những quy định phi chức năng, những yêu cầu về môi trường, những yêu cầu về khả năng thực thi, … OOD tập trung vào việc cải thiện kết quả của OOA, tối ưu hoá giải pháp đã được cung cấp trong khi vẫn đảm bảo thoả mãn tất cả các yêu cầu đã được xác lập.

Trong giai đoạn OOD, nhà thiết kế định nghĩa các chức năng, thủ tục (operations), thuộc tính (attributes) cũng như mối quan hệ của một hay nhiều lớp (class) và quyết định chúng cần phải được điều chỉnh sao cho phù hợp với môi trường phát triển. Đây cũng là giai đoạn để thiết kế ngân hàng dữ liệu và áp dụng các kỹ thuật tiêu chuẩn hoá.

Về cuối giai đoạn OOD, nhà thiết kế đưa ra một loạt các biểu đồ (diagram) khác nhau. Các biểu đồ này có thể được chia thành hai nhóm chính là tĩnh và động. Các biểu đồ tĩnh biểu thị các lớp và đối tượng, trong khi biểu đồ động biểu thị tương tác giữa các lớp và phương thức hoạt động chính xác của chún. Các lớp đó sau này có thể được nhóm thành các gói (packages) tức là các đơn vị thành phần nhỏ hơn của ứng dụng.

# 2.3. Thư viện hỗ trợ kết nối và trao đổi dữ liệu

## 2.3.1. Giới thiệu về dịch vụ Google Firebase

**Firebase** là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn. Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

## 2.3.2. Ưu điểm của Firebase

* **Triển khai ứng dụng nhanh chóng**

Bởi không phải quan tâm đến phần backend cùng các API tốt, hỗ trợ đa nền tảng, Firebase tiết kiệm rất nhiều thời gian quản lý và đồng bộ tất cả dữ liệu cho người dùng. Song song đó, nó còn cung cấp hosting và hỗ trợ xác thực thông tin khách hàng khiến việc triển khai ứng dụng nhanh chóng hơn

* **Bảo mật**

Hoạt động trên nền tảng đám mây cloud, sử dụng kết nối thông qua giao thức bảo mật SSL và cho phép phân quyền người dùng cơ sở dữ liệu bằng Javascript, các đặc điểm này của Firebase giúp nâng cao độ bảo mật cho các ứng dụng.

* **Ổn định**

Hầu hết các ứng dụng trên nền tảng Firebase luôn hoạt động ổn định vì chúng được viết dựa trên nền tảng cloud cung cấp bởi Google. Hơn nữa, nhờ Firebase mà việc nâng cấp hay bảo trì Server cũng diễn ra khá đơn giản mà không cần phải dừng lại.

## 2.3.4. Nhược điểm của Firebase

Firebase chính là phần Realtime Database, đúng hơn chỉ là phần Database. Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức đôi chút tới khá nhiều.

# 2.4. Giới thiệu - Pytorch

PyTorch được định nghĩa là một thư viện machine learning mã nguồn mở cho Python. Nó được sử dụng cho các ứng dụng như xử lý ngôn ngữ tự nhiên. Ban đầu nó được phát triển bởi nhóm nghiên cứu trí tuệ nhân tạo của Facebook và phần mềm Uber’s Pyro để lập trình xác suất.

PyTorch được Hugh Perkins phát triển như một trình bao bọc Python cho LusJIT dựa trên khuôn khổ Torch. Có hai biến thể PyTorch.

PyTorch thiết kế lại và triển khai Torch bằng Python trong khi chia sẻ cùng các thư viện C lõi cho code phụ trợ. Các nhà phát triển PyTorch đã điều chỉnh mã back-end này để chạy Python một cách hiệu quả. Họ cũng giữ nguyên khả năng tăng tốc phần cứng dựa trên GPU cũng như các tính năng mở rộng đã tạo nên Torch dựa trên Lua-based.

## 2.4.1. Features (Tính năng):

Các tính năng chính của PyTorch

**Giao diện thân thiện** - PyTorch cung cấp API dễ sử dụng; do đó nó được coi là rất đơn giản để vận hành và chạy trên Python. Việc thực thi mã trong khuôn khổ này khá dễ dàng.

**Sử dụng Python** - Thư viện này được coi là Pythonic tích hợp trơn tru với ngăn xếp khoa học dữ liệu Python. Do đó, nó có thể tận dụng tất cả các dịch vụ và chức năng được cung cấp bởi môi trường Python.

**Đồ thị tính toán** - PyTorch cung cấp một nền tảng tuyệt vời cung cấp đồ thị tính toán động. Do đó người dùng có thể thay đổi chúng trong thời gian chạy. Điều này rất hữu ích khi nhà phát triển không biết cần bao nhiêu bộ nhớ để tạo mô hình mạng nơron.

PyTorch được biết đến với ba cấp độ trừu tượng như được đưa ra dưới đây

* Tensor − Mảng n-chiều bắt buộc chạy trên GPU.
* Variable −Nút trong đồ thị tính toán. Lưu trữ dữ liệu và gradient.
* Module −Lớp mạng nơ-ron sẽ lưu trữ trạng thái hoặc trọng số có thể học được.

## 2.4.2. Ưu điểm của PyTorch:

* Dễ dàng để gỡ lỗi và hiểu mã.
* Gồm rất nhiều hàm mất mát(loss function).
* Có thể coi đây là phần mở rộng NumPy cho GPU.
* Cho phép xây dựng các mạng có cấu trúc phụ thuộc vào chính tính toán.

## 2.4.3. Cài đặt - Pytorch

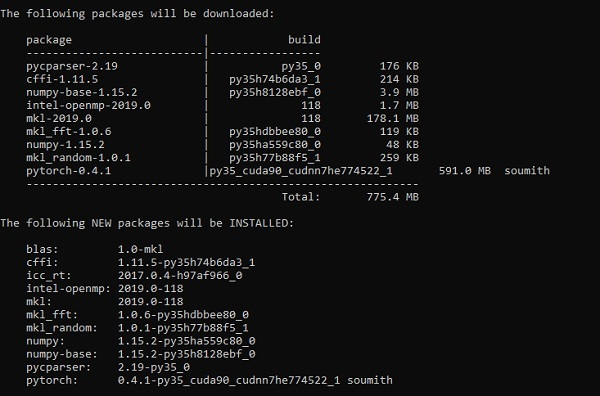
PyTorch là một framework sâu phổ biến. Trong hướng dẫn này, tôi sẽ sử dụng windows 10 để cài đặt và chạy pytorch. Các bước để thiết lập môi trường như sau:

**Bước 1:**

Liên kết sau bao gồm danh sách các gói trong đó có các gói phù hợp cho PyTorch.

<https://drive.google.com/drive/folders/0B-X0-FlSGfCYdTNldW02UGl4MXM>

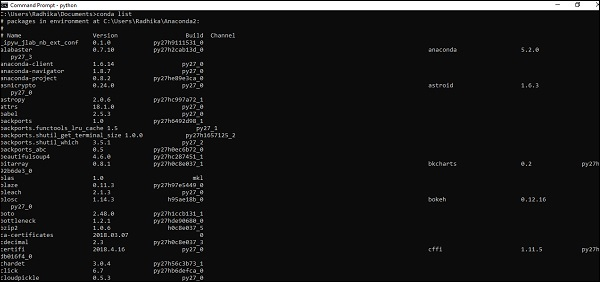
Bạn chọn bản tương ứng, sau đó tải xuống và cài đặt, kết quả như sau:



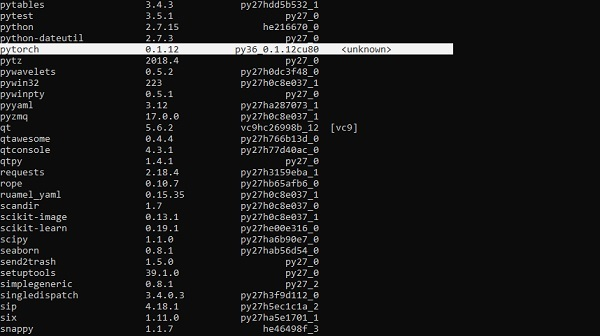
**Bước 2 :**

Liên quan đến việc xác minh cài đặt framework PyTorch bằng framework Anaconda.

conda list



“Conda list” hiển thị danh sách các framework đã được cài đặt.

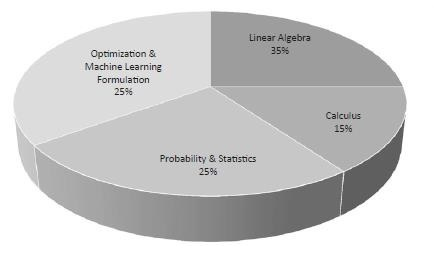


Phần được đánh dấu cho thấy PyTorch đã được cài đặt thành công trong máy.

# 2.5. Nền tảng toán học của Neural Networks

Toán học quan trọng trong bất kỳ thuật toán học máy nào và bao gồm các khái niệm cốt lõi khác nhau của toán học để có được thuật toán phù hợp được thiết kế theo một cách cụ thể.

Tầm quan trọng của các chủ đề toán học đối với học máy và khoa học dữ liệu như sau:



Bây giờ, ta sẽ tập trung vào các khái niệm toán học chính của học máy, điều quan trọng theo quan điểm Xử lý ngôn ngữ tự nhiên.

## 2.5.1. Vectors

Vectơ được coi là mảng số liên tục hoặc rời rạc và không gian bao gồm các vectơ được gọi là không gian vectơ. Kích thước không gian của vectơ có thể hữu hạn hoặc vô hạn nhưng người ta đã quan sát thấy rằng các vấn đề về học máy và khoa học dữ liệu giải quyết các vectơ có độ dài cố định.

Biểu diễn vector được hiển thị như sau:

temp = torch.FloatTensor([23,24,24.5,26,27.2,23.0])

temp.size()

Output - torch.Size([6])

Trong học máy, xử lý dữ liệu đa chiều. Vì vậy, vectơ trở nên rất quan trọng và được coi là đặc điểm đầu vào cho bất kỳ câu lệnh bài toán dự đoán nào.

## 2.5.2. Scalars

Vô hướng được gọi là không có thứ nguyên chỉ chứa một giá trị. Khi nói đến PyTorch, nó không bao gồm một tensor đặc biệt với kích thước bằng không; do đó khai báo sẽ được thực hiện như sau:

x = torch.rand(10)

x.size()

Output - torch.Size([10])

## 2.5.3. Matrices (Ma trận)

Hầu hết dữ liệu có cấu trúc thường được biểu diễn dưới dạng bảng hoặc một ma trận cụ thể. Ta sẽ sử dụng một tập dữ liệu có tên là Giá Nhà ở Boston, có sẵn trong thư viện máy học Python scikit-learning.

boston\_tensor = torch.from\_numpy(boston.data)

boston\_tensor.size()

Output: torch.Size([506, 13])

boston\_tensor[:2]

Output:

Columns 0 to 7

0.0063 18.0000 2.3100 0.0000 0.5380 6.5750 65.2000 4.0900

0.0273 0.0000 7.0700 0.0000 0.4690 6.4210 78.9000 4.9671

Columns 8 to 12

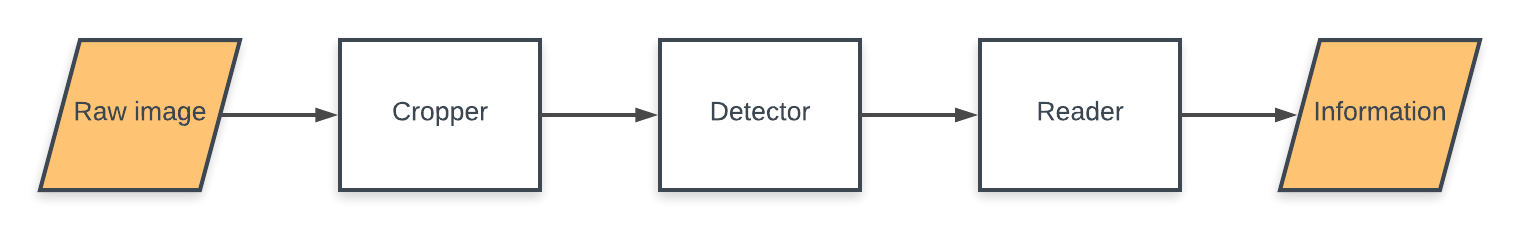
1.0000 296.0000 15.3000 396.9000 4.9800

2.0000 242.0000 17.8000 396.9000 9.1400

Alignment ảnh thẻ sinh viên với PyTorch

Bài toán nhận diện thẻ sinh viên

Đây là một bài toán không mới và cũng có khá nhiều bạn coi đây như một dự án để thực hành kĩ năng xử lý với các mô hình Deep Learning. Tôi khuyên các bạn nên thực hành bài toán này vì nó sẽ giúp cho các bạn có điều kiện làm quen với nhiều loại mô hình khác nhau trong Deep Learning như Object Detection, Instance Segmentation, Optical Character Recognition. Pipeline cơ bản của nó có thể giải quyết như flow sau đây:



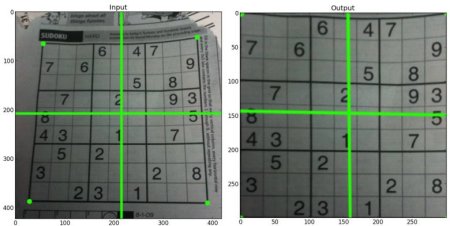
* **Cropper** hay còn gọi là **alignment ảnh** nhận đầu vào từ ảnh raw data. Crop vùng chứa thẻ sinh viên trong ảnh. Sử dụng Geometric Transform để xoay ảnh cho đúng chiều
* **Detector** được sử dụng để detect các thành phần trong ảnh như họ tên, ngày tháng năm sinh ....
* **Reader** là một module OCR để đọc nội dung chữ viết từ các thành phần đã được crop.

Cuối cùng tổng hợp và sắp xếp lại ta được một bài toán nhận diện thẻ sinh viên hoàn chỉnh. OK giờ chúng ta đi vào phần chính sẽ được giải quyết trong ngày hôm nay nhé

## 2.5.4. Alignment

Là một kĩ thuật phổ biến trong xử lý ảnh Nó chính là quá trình biến đổi các bộ dữ liệu khác nhau về cùng một hệ tọa độ. Các hình ảnh được chụp từ điện thoại, từ cảm biến được chụp bởi các góc độ khác nhau. Có nhiều phương pháp để thực hiện điều này như sử dụng Feature Based hay Template Matching. Tuy nhiên mục đích cuối cùng vẫn là làm sao để thu được một bức ảnh ngay ngắn trong hệ tọa độ để dễ dàng xử lý nhất. Ở đây với bài toán thẻ sinh viên bạn sẽ cần như trong hình sau:

Vậy làm thế nào để thực hiện được điều này, chúng ta có thể nghĩ ngay đến **Geometric Transformations of Images** trong xử lý ảnh mà cụ thể đó là **Perspective Transformation** - mapping 4 tọa độ ở ảnh gốc vào 4 tọa độ ở ảnh đích. Chúng ta có thể thấy ảnh ví dụ kinh điển như sau:



Trong đó bàn cờ đã được **trải phẳng** trong hệ tọa độ mới giúp chúng ta dễ xử lý hơn. Vậy nếu đã chọn hướng giải quyết là **Perspective Transformation**

# 2.6. VietOCR - Nhận Dạng Tiếng Việt Sử Dụng Mô Hình Transformer và AttentionOCR

## 2.6.1. Giới Thiệu



Trong bài này, tôi trình bày 2 mô hình OCR cho bài toán nhận dạng chữ tiếng việt: AttentionOCR và TransformerOCR. AttentionOCR sử dụng kiến trúc attention seq2seq đã được sử dụng khá nhiều trong các bài toán NLP và cả OCR, còn TransformerOCR sử dụng kiến trúc của Transformer đã đạt được nhiều tiến bộ vượt bậc cho cộng đồng NLP. Một câu hỏi mà tôi cũng khá quan tâm là Liệu TransformerOCR có mang lại kết quả vượt bậc như những gì các bạn đã nhìn thấy trong các bài toán NLP hay không ?  
Đồng thời, tôi cũng cung cấp một [thư viện](https://github.com/pbcquoc/vietocr) mới cho bài toán OCR, thư viện hướng tới kết quả chính xác, nhanh chóng, dễ huấn luyện, dễ dự đoán cho cả các bạn chưa có nhiều kinh nghiệm cũng có thể sử dụng được trong các bài toán liên quan đến số hóa.

## 2.6.2. Mô Hình

Trong phần này tôi sẽ trình bày cách kết hợp mô hình CNN và mô hình Language Model (Seq2Seq và Transformer) để tạo thành một mô hình giúp các bạn giải quyết bài toán OCR. Chi tiết từng bước mô hình hoạt động, các bạn nên tham khảo những bài giới thiệu kiến trúc seq2seq của cộng đồng NLP vì mô hình này được ứng dụng khá nhiều và nổi tiếng.

Ngoài ra, tôi cũng so sánh hạn chế của mô hình OCR cổ điển sử dụng CTCLoss với 2 mô hình kể trên từ đó giúp các bạn lựa chọn mô hình phù hợp trong các vấn đề thực tế.

## 2.6.3. CNN Của Mô Hình OCR

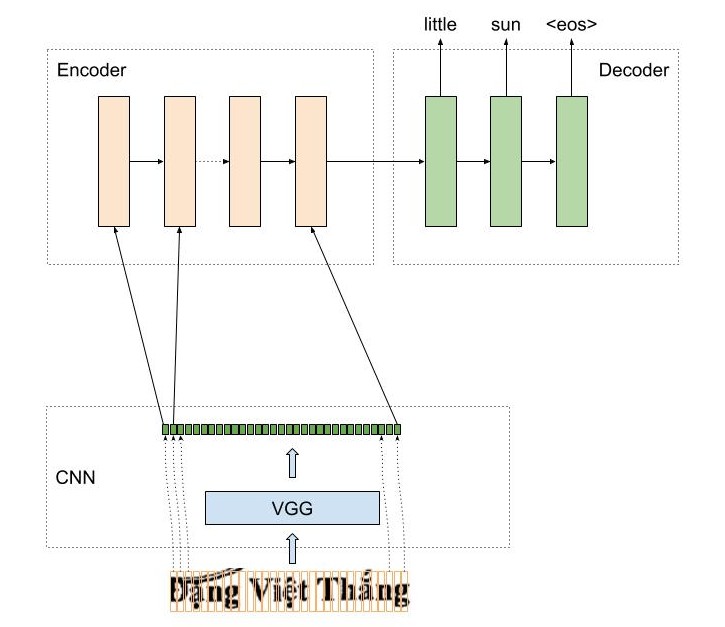
Mô hình CNN dùng trong bài toán OCR nhận đầu vào là một ảnh, thông thường có kích thước với chiều dài lớn hơn nhiều so với chiều rộng, do đó việc điều chỉnh tham số stride size của tầng pooling là cực kì quan trọng. Tôi thường chọn kích thước stride size của các lớp pooling cuối cùng là wxh=2x1 trong mô hình OCR. Không thay đổi stride size phù hợp với kích thước ảnh thì sẽ dẫn đến kết quả nhận dạng của mô hình sẽ tệ.

Đối với mô hình VGG, việc thay đổi pooling size khá dễ do kiến trúc đơn giản, tuy nhiên đối với mô hình phức tạp khác như resnet việc điều chỉnh tham số pooling size hơi phức tạp do một ảnh bị downsampling không chỉ bởi tầng pooling mà còn tại các tầng convolution khác.

Trong pytorch, đối với mô hình VGG, các bạn chỉ đơn giản là thay thế stride size của tầng pooling.

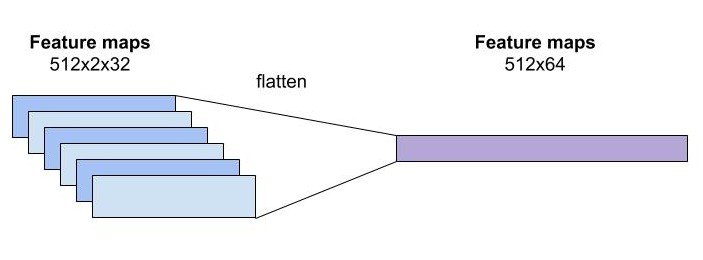
cnn.features[i] = torch.nn.AvgPool2d(kernel\_size=ks[pool\_idx], stride=ss[pool\_idx], padding=0)

## 2.6.4. AttentionOCR



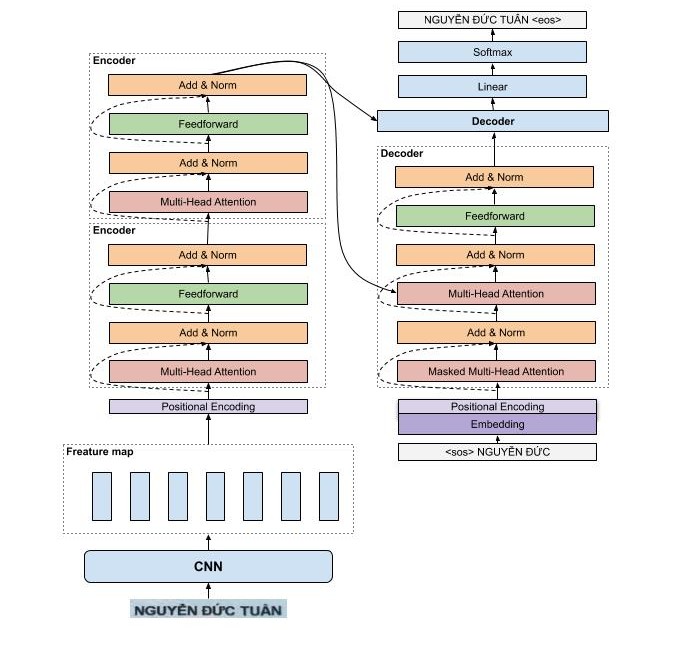
AttentionOCR là sự kết hợp giữa mô hình CNN và mô hình Attention Seq2Seq. Cách hoạt động của mô hình này tương tự như kiến trúc của mô hình seq2seq trong bài toán dịch máy. Với bài toán dịch máy từ tiếng việt sang anh, chúng ta cần encode một chuỗi tiếng việt thành một vector đặc trưng, còn trong mô hình AttentionOCR, thì dữ liệu đầu vào này là một ảnh.

Một ảnh qua mô hình CNN, sẽ cho một feature maps có kích thước channelxheightxwidth, feature maps này sẽ trở thành đầu vào cho mô hình LSTM, tuy nhiên, mô hình LSTM chỉ nhận chỉ nhận đầu vào có kích thước là hiddenxtime\_step. Một cách đơn giản và hợp lý là 2 chiều cuối cùng heightxwidth của feature maps sẽ được duổi thẳng. Feature maps lúc này sẽ có kích thước phù hợp với yêu cầu của mô hình LSTM.



Feature maps của mô hình CNN sau khi được flatten thì được truyền vào làm input của mô hình LSTM, tại mỗi thời điểm, mô hình LSTM cần dự đoán từ tiếp theo trong ảnh là gì.

## 2.6.5. TransformerOCR



## 2.6.6. Huấn Luyện Mô Hình

Huấn luyện mô hình AttenionOCR hay TransformerOCR hoàn toàn giống với luyện mô hình seq2seq, chúng đều sử dụng cross-entropy loss để tối ưu thay vì sử dụng CTCLoss như mô hình CRNN, tức là tại mỗi thời điểm mô hình dự đoán một từ sau đó so sánh với nhãn để tính loss và cập nhật lại trọng số của mô hình.

## 2.6.7. Hạn Chế Của Mô Hình Sử dụng CTCLoss

Đối với mô hình CRNN sử dụng CTCloss để làm hàm mục tiêu, số lượng kí tự đối đa có thể dự đoán bằng với widthxheight của feature maps. Do đó, các bạn cần phải cẩn thận điều chỉnh kiến trúc mô hình để có thể dự đoán được số kí tự phù hợp với từng bộ dataset. Đối với mô hình AttentionOCR hoặc TransformerOCR, các bạn không gặp vấn đề này, làm cho các bạn có thể dễ dàng sử dụng lại pretrained model cho các loại dữ liệu khác nhau.

Ngoài ra, AttentionOCR hoặc TransformerOCR đều có kiến trúc của mô hình dịch này seq2seq, do đó các thủ thuật của mô hình này đều có thể ứng dụng cho mô hình của chúng ta.

Thư Viện VietOCR

Thư viện VietOCR được tôi xây dựng với mục đích hỗ trợ các bạn có thể sử dụng để giải quyết các bài toán liên quan đến OCR trong công nghiệp. Thư viện cung cấp cả 2 kiến trúc AtentionOCR và TransformerOCR. Tuy kiến trúc TransformerOCR hoạt động khá tốt trong NLP, nhưng theo tôi nhận xét thì độ chính không có sự cải thiện đáng kể so với AttentionOCR mà thời gian dự đoán lại chậm hơn khá nhiều.

# 2.7. Mô hình Facenet trong face recognition

## 2.7.1. Hệ thống nhận diện khuôn mặt

1. **Các ứng dụng phổ biến**

Vào năm 2012, tôi vẫn còn nhớ facebook bắt dầu có chức năng nhận diện khuôn mặt. Khi đó tôi còn là một sinh viên chưa biết đến khái niệm trí tuệ nhân tạo. Tôi cực kì ấn tượng với khả năng kì diệu này. Tôi nghĩ chắc facebook phải thuê hàng ngàn người ngồi gán nhãn cho từng bức ảnh mỗi khi người dùng update lên. Hơi ngây thơ chút, nhưng đó là tất cả những gì tôi có thể biết cách đây 8 năm.

Vào năm 2018, apple bắt đầu tích hợp chức năng nhận diện khuôn mặt trong các dòng sản phẩm iphone X. Sau đó, một xu hướng các smart phone ứng dụng nhận diện khuôn mặt ra đời.

Một số ngân hàng ở Autralia bắt đầu ứng dụng xác thực khuôn mặt trong các giao dịch ATM.

Tại Trung Quốc, hệ thống nhận diện khuôn mặt được triển khai trên toàn quốc giúp chấm điểm công dân và đồng thời xác minh nhiều tội phạm lẩn trốn.

Còn tại cơ quan tôi, hệ thống nhận diện khuôn mặt được nghiên cứu in-house áp dụng để chấm công cho nhân viên.

Tôi nghĩ rằng nhận diện khuôn mặt có tính ứng dụng cao. Rất nhiều công ty, doanh nghiệp và quốc gia đang cần.

Tuy nhiên khá kì lạ là nhiều nơi chưa áp dụng nhận diện khuôn mặt vào cơ quan của tôi. Việc xây dựng một hệ thống nhận diện khuôn mặt là không hề khó. Hãy có niềm tin rằng bạn có thể tự xây dựng một hệ thống nhận diện khuôn mặt cho cơ quan của tôi sau bài viết này.

Trước tiên tôi sẽ viết về lý thuyết của thuật toán. Phần thực hành tôi sẽ giới thiệu ở một bài khác.

Đối với các bạn muốn hiểu sâu về thuật toán có thể bắt đầu đọc bài này.

Đối với các bạn muốn thấy trước hiệu quả. Hãy chờ bài thực hành sau bài này.

1. **Các loại hệ thống xác thực**

Hầu hết các hệ thống xác thực sẽ dựa trên thông tin sinh trắc để định danh một người. Các thông tin sinh trắc là những thông tin duy nhất với mỗi người. Đó có thể là vân tay, ADN, vân mắt, khuôn mặt (không phẫu thuật thẩm mỹ, hãy thả lỏng khuôn mặt nhất có thể, không chu môi, nháy mắt, tạo kiểu cute các thứ), hình dạng, dáng đi,….

Theo đó, có vô số các phương pháp xác thực khác nhau, tùy vào việc sử dụng thông tin sinh trắc nào. Ở Việt Nam thì chắc phổ biến nhất là xác thực bằng vân tay.

Trước đây khi computer vision chưa phát triển thì các hệ thống xác thực bằng vân tay, bằng vân mắt được áp dụng khá phổ biến. Vài năm gần đây, xác thực bằng khuôn mặt mới được áp dụng rộng rãi. Xét về độ chính xác thì các hệ thống xác thực vân tay, vân mắt đáng tin cậy hơn so với nhận diện khuôn mặt vì dữ liệu không thay đổi, trong khi khuôn mặt một người có thể thay đổi qua thời gian và chịu sự co dãn.

Tuy nhiên hệ thống nhận diện khuôn mặt được ưa chuộng vì quá trình xác thực nhanh, có thể xác thực từ xa, không cần phải tiếp xúc như vân tay, vân mắt. Bạn được phép nhìn người khác nhưng muốn động chạm thì cần phải được sự xin phép của họ.

1. **Các phương pháp xác thực khuôn mặt**

* Phương pháp truyền thống

Các phương pháp truyền thống: Xác thực khuôn mặt bằng cách trích xuất ra một land mark cho face. Land mark như là một bản đồ xác định các vị trí cố định trên khuôn mặt của một người như mắt, mũi, miệng, lông mày,….



Như vậy thay land mark face đã loại bỏ những phần thông tin không cần thiết và giữ lại những thông tin chính. Khi đó mỗi khuôn mặt sẽ được nén thành một véc tơ n chiều. Thông thường là 68 chiều.

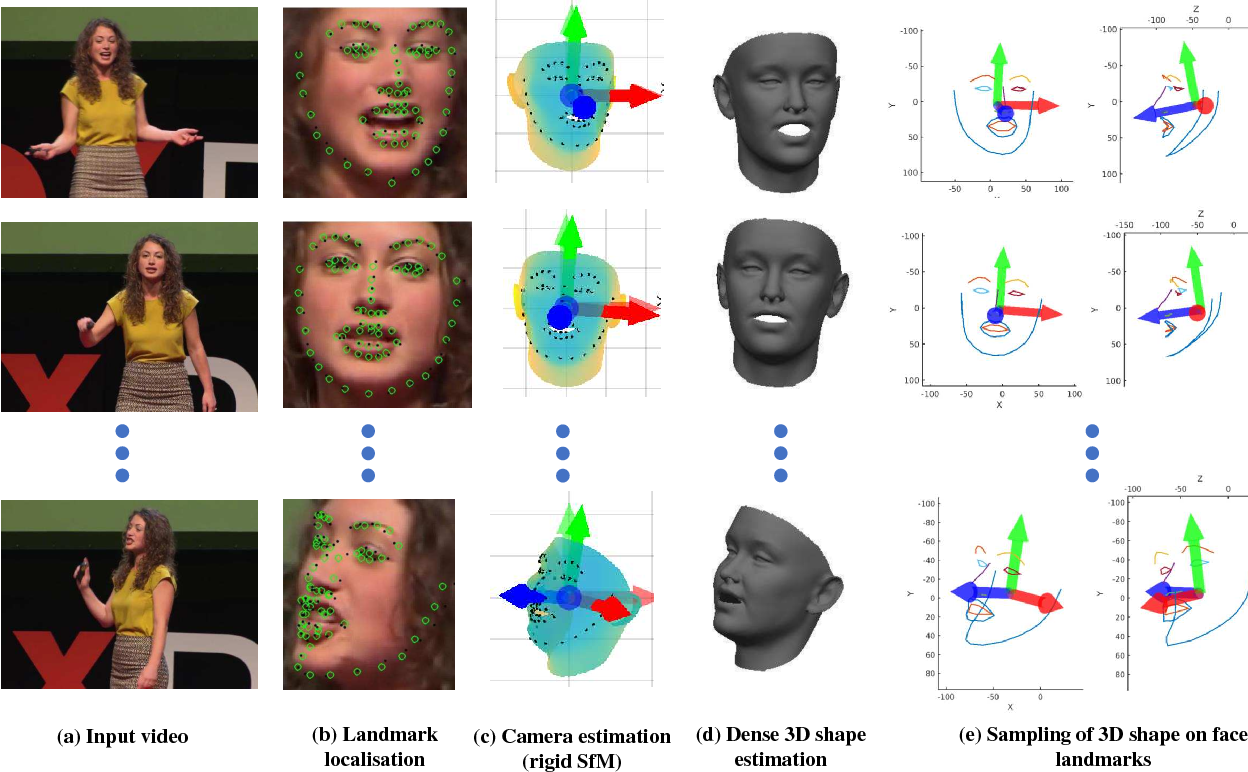
Sử dụng đầu vào là land mark face, áp dụng các thuật toán cổ điển như SVM, k-NN, Naive Bayes, Random Forest, MLP, … để phân loại khuôn mặt cho một người.

* Nhận diện 3D

Kĩ thuật nhận diện 3D sẽ sử dụng không gian 3 chiều để biểu diễn khuôn mặt. Từ thông tin này có thể xác định các đặc trưng khác nhau trên bề mặt khuôn mặt như các đường countour của mắt, mũi, cằm.

Một lợi thế của nhận diện khuôn mặt 3D là không bị ảnh hưởng bởi những thay đổi về ánh sáng như các phương pháp 2D. Dữ liệu 3D đã cải thiện đáng kể độ chính xác của nhận dạng khuôn mặt.

Để tạo ra một ảnh 3D, một cụm ba camera được áp dụng. Mỗi camera sẽ hướng vào một góc khác nhau. Tất cả các camera này phối hợp cùng nhau trong việc theo dõi khuôn mặt của một người trong thời gian thực và có thể nhận diện chúng.



Nhận diện khuôn mặt của iphone là nhận diện 3D. Bạn sẽ phải quay tròn khuôn mặt của tôi khi xác thực nó để thuật toán học các góc độ khác nhau.

* Các phương pháp nhận diện khác

Ngoài ra còn có một số phương pháp nhận diện khuôn mặt như nhận diện cảm biến da và phương pháp kết hợp.

Phương pháp kết hợp có thể sử dụng nhiều thông tin từ đồng thời land mark face, nhận diện 3D, nhận diện cảm biến da và mang lại độ chính xác cao hơn vì nó nhận diện tốt được trong các trường hợp khuôn mặt có các biểu cảm như cau mày, chớp mắt, co dãn khi cười, nói và ít nhạy cảm với chiếu sáng.

## 2.7.2. Các bài toán khác nhau về face

Có nhiều lớp bài toán khác nhau liên quan đến dữ liệu face. Một trong số 4 bài toán phổ biến nhất dựa trên nhu cầu thực tế cần áp dụng đó là:

Nhận diện khuôn mặt (face identification): Đây là bài toán match one-many. Bài toán này sẽ trả lời cho câu hỏi “người này là ai?” bằng cách nhận input là ảnh khuôn mặt và output là nhãn tên người trong ảnh. Tác vụ này thường được áp dụng trong các hệ thống chấm công, hệ thống giám sát công dân, hệ thống cammera thông minh tại các đô thị.

Xác thực khuôn mặt (face verification): Đây là bài toán match one-one. Bài toán này trả lời cho câu hỏi “có phải 2 ảnh đầu vào là cùng một người không?” Kết quả output sẽ là yes hoặc no. Bài toán thường được dùng trong các hệ thống bảo mật. Xác thực khuôn mặt trên điện thoại của bạn là một bài toán như vậy.

Do có mối quan hệ khá gần nên face recognition là tên gọi chung cho cả hai thuật toán face identification và face verification.

Tìm kiếm khuôn mặt đại diện (face clustering): Chắc hẳn bạn đã từng đọc hoặc xem video về người có [khuôn mặt đặc trưng nhất thế giới](https://www.youtube.com/watch?v=xLaDbe7P2RU). Đơn giản là ta chỉ cần tính ra trung bình của các ảnh khuôn mặt để thu được centroid image. Tính similarity giữa centroid với toàn bộ khuôn mặt còn lại để thu được khuôn mặt đặc trưng nhất similarity nhất với centroid. Tương tự như vậy bạn cũng có thể tìm ra khuôn mặt đặc trưng nhất của nam, nữ các quốc gia.



Tìm kiếm khuôn mặt tương đương (face similarity): Chắc các bạn đã từng nghịch qua một số ứng dụng của facebook như bạn giống diễn viên điện ảnh nào nhất. Thuật toán này khá đơn giản, chỉ cần đo lường ảnh mà bạn upload lên với các ảnh diễn viên sẵn có và chọn ra một cái gần nhất. Nhưng đừng tin là tôi là họ nhé. Có thể đây chỉ là 1 hình ảnh hai số phận.

## 2.7.3. Các thuật toán nhận diện khuôn mặt

1. **One-shot learning**

One-shot learning là thuật toán học có giám sát mà mỗi một người chỉ cần 1 vài, rất ít hoặc thậm chí chỉ 1 bức ảnh duy nhất (để khỏi tạo ra nhiều biến thể).

Từ đầu vào là bức ảnh của một người, chúng ta sử dụng một kiến trúc thuật toán CNN đơn giản để dự báo người đó là ai.

Tuy nhiên nhược điểm của phương pháp này là chúng ta phải huấn luyện lại thuật toán thường xuyên khi xuất hiện thêm một người mới vì shape của output thay đổi tăng lên 1. Rõ ràng là không tốt đối với các hệ thống nhận diện khuôn mặt của một công ty vì số lượng người luôn biến động theo thời gian.

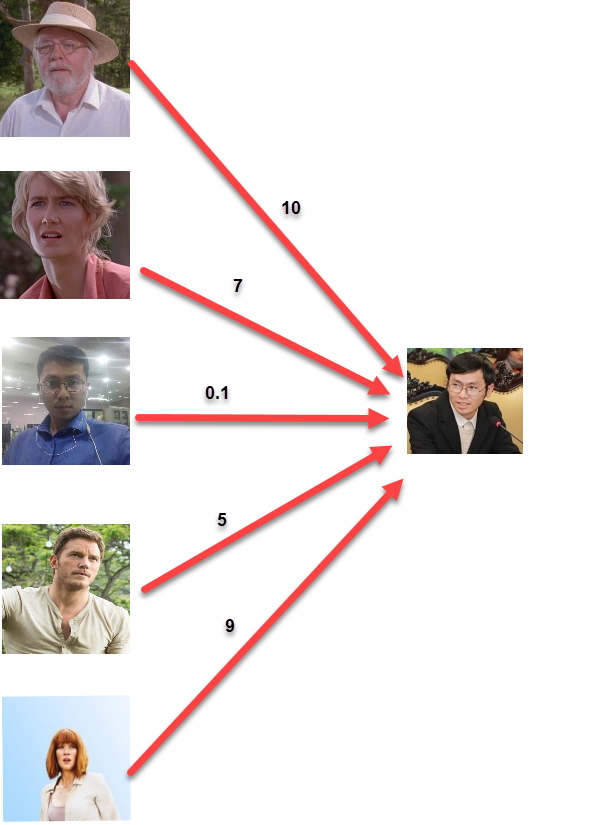
Để khắc phục được vấn đề này, chúng ta sử dụng phương pháp learning similarity.

1. **Learning similarity**

Phương pháp này dựa trên một phép đo khoảng cách giữa 2 bức ảnh, thông thường là các norm chuẩn l1 hoặc l2 sao cho nếu 2 bức ảnh thuộc cùng một người thì khoảng cách là nhỏ nhất và nếu không thuộc thì khoảng cách sẽ lớn hơn.

{d(img1,img2)≤τ→samed(img1,img2)>τ→different

Để cụ thể hơn, tôi minh họa qua hình bên dưới:



**Hình 1:** Phương pháp learning similarity. Thay vì dự báo một phân phối xác suất để tìm ra nhãn phù hợp nhất với ảnh đầu vào. Thuật toán sẽ so sánh khoảng cách giữa ảnh đầu vào (bên phải) với toàn bộ các ảnh còn lại (bên trái). Ta cần chọn một ngưỡng threshold để quyết định ảnh là giống hoặc khác. Giả sử ngưỡng threshold là 0.5. Trong các bức ảnh bên trái thì bức ảnh ở giữa có khoảng cách với ảnh bên phải nhỏ hơn 0.5. Do đó nó được dự báo cùng một người với ảnh bên phải.

Learning similarity có thể trả ra nhiều hơn một ảnh là cùng loại với ảnh đầu vào tùy theo ngưỡng threshold.

Ngoài ra phương pháp này không bị phụ thuộc vào số lượng classes. Do đó không cần phải huấn luyện lại khi xuất hiện class mới.

Điểm mấu chốt là cần xây dựng được một model encoding đủ tốt để chiếu các bức ảnh lên một không gian eucledean n chiều. Sau đó sử dụng khoảng cách để quyết định nhãn của chúng.

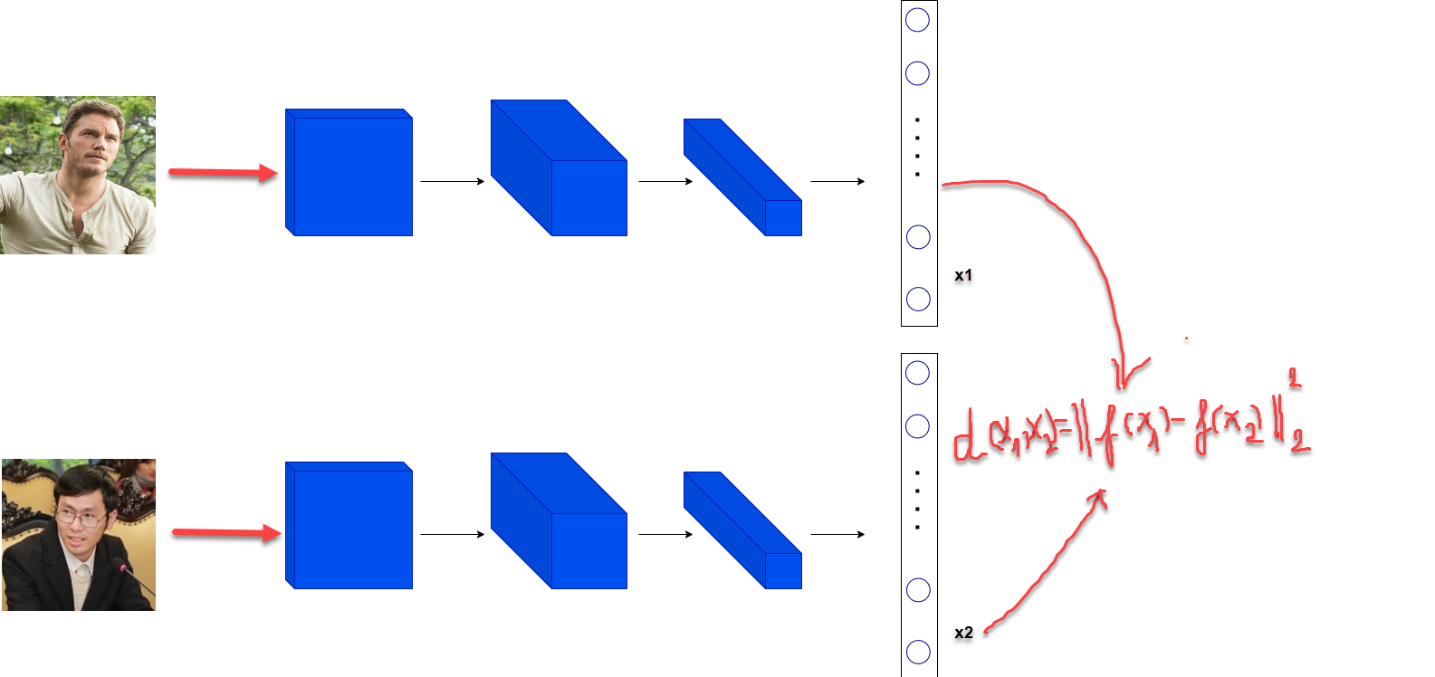
Như vậy learning similarity có ưu điểm hơn so với one-shot learning khi không phải huấn luyện lại model khi mà vẫn tìm ra được ảnh tương đồng.

Vậy làm thế nào để học được biểu diễn của ảnh trong không gian euledean n chiều? Kiến trúc siam network sẽ giúp chúng ta thực hiện điều này một cách dễ dàng.

1. **Siam network**

Những kiến trúc mạng mà khi bạn đưa vào 2 bức ảnh và mô hình sẽ trả lời chúng thuộc về cùng 1 người hay không được gọi chung là Siam network. Siam network được giới thiệu đầu tiên bởi [DeepFace: Closing the Gap to Human-Level - Yaniv Taigman. elt](https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf).

Kiến trúc của Siam network dựa trên base network là một Convolutional neural network **đã được loại bỏ output layer** có tác dụng encoding ảnh thành véc tơ embedding. Đầu vào của mạng siam network là 2 bức ảnh bất kì được lựa chọn ngẫu nhiên từ dữ liệu ảnh. Output của Siam network là 2 véc tơ tương ứng với biểu diễn của 2 ảnh input. Sau đó chúng ta đưa 2 véc tơ vào hàm loss function để đo lường sự khác biệt giữa chúng. Thông thường hàm loss function là một hàm norm chuẩn bậc 2.



**Hình 2:** Từ mô hình Convolutional neural network, mô hình trả ra 2 véc tơ encoding là x1 và x2 biểu diễn cho lần lượt ảnh 1 và 2. x1 và x2 có cùng số chiều. Hàm f(x) có tác dụng tương tự như một phép biến đổi qua layer fully connected trong mạng neural network để tạo tính phi tuyến và giảm chiều dữ liệu về các kích thước nhỏ. Thông thường là 128 đối đối với các mô hình pretrain.

Khi x1,x2 là cùng 1 người:

||f(x1)−f(x2)||22

phải là một giá trị nhỏ.

Khi x1,x2 là 2 người khác nhau:

||f(x1)−f(x2)||22

phải là một giá trị lớn.

Khi sử dụng siam network chúng ta sẽ không cần phải lo lắng về vấn đề output shape thay đổi vì base network đã được loại bỏ layer cuối.

Mục tiêu chính của siam network đó là tìm ra biểu diễn của ảnh trong không gian n chiều nên không nhất thiết phải lựa chọn hàm loss function là hàm binary cross entropy như các bài toán phân loại nhị phân. Trên thực tế lựa chọn loss function là binary cross entropy vẫn tìm ra được biểu diễn tốt cho ảnh. Ngoài ra còn một số các biến thể khác của loss function như sử dụng khoảng cách norm chuẩn l2 đã chuẩn hóa theo phân phối χ2 dạng:

||f(x1)−f(x2)||22||f(x1)+f(x2)||2

hoặc sử dụng norm chuẩn l1.

Việc lựa chọn hàm loss function sẽ có tác động rất lớn tới độ chính xác của biểu diễn ảnh. Ngoài các hàm loss function nêu trên, còn một hàm loss function có hiệu quả rất lớn đối với việc biểu diễn véc tơ cho face. Đó là hàm gì? Chúng ta sẽ tìm hiểu ở thuật toán facenet.

## 2.7.4. Thuật toán facenet

Facenet chính là một dạng siam network có tác dụng biểu diễn các bức ảnh trong một không gian eucledean n chiều (thường là 128) sao cho khoảng cách giữa các véc tơ embedding càng nhỏ, mức độ tương đồng giữa chúng càng lớn.

1. **Khái quát thuật toán**

Hầu hết các thuật toán nhận diện khuôn mặt trước facenet đều tìm cách biểu diễn khuôn mặt bằng một véc tơ embedding thông qua một layer bottle neck có tác dụng giảm chiều dữ liệu.

Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều embedding tương đối lớn (thường >= 1000) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để tăng tốc độ tính toán.

Hàm loss function chỉ đo lường khoảng cách giữa 2 bức ảnh. Như vậy trong một đầu vào huấn luyện chỉ học được **một trong hai** khả năng là sự giống nhau nếu chúng cùng 1 class hoặc sự khác nhau nếu chúng khác class mà không học được cùng lúc sự giống nhau và khác nhau trên cùng một lượt huấn luyện.

Facenet đã giải quyết cả 2 vấn đề trên bằng các hiệu chỉnh nhỏ nhưng mang lại hiệu quả lớn:

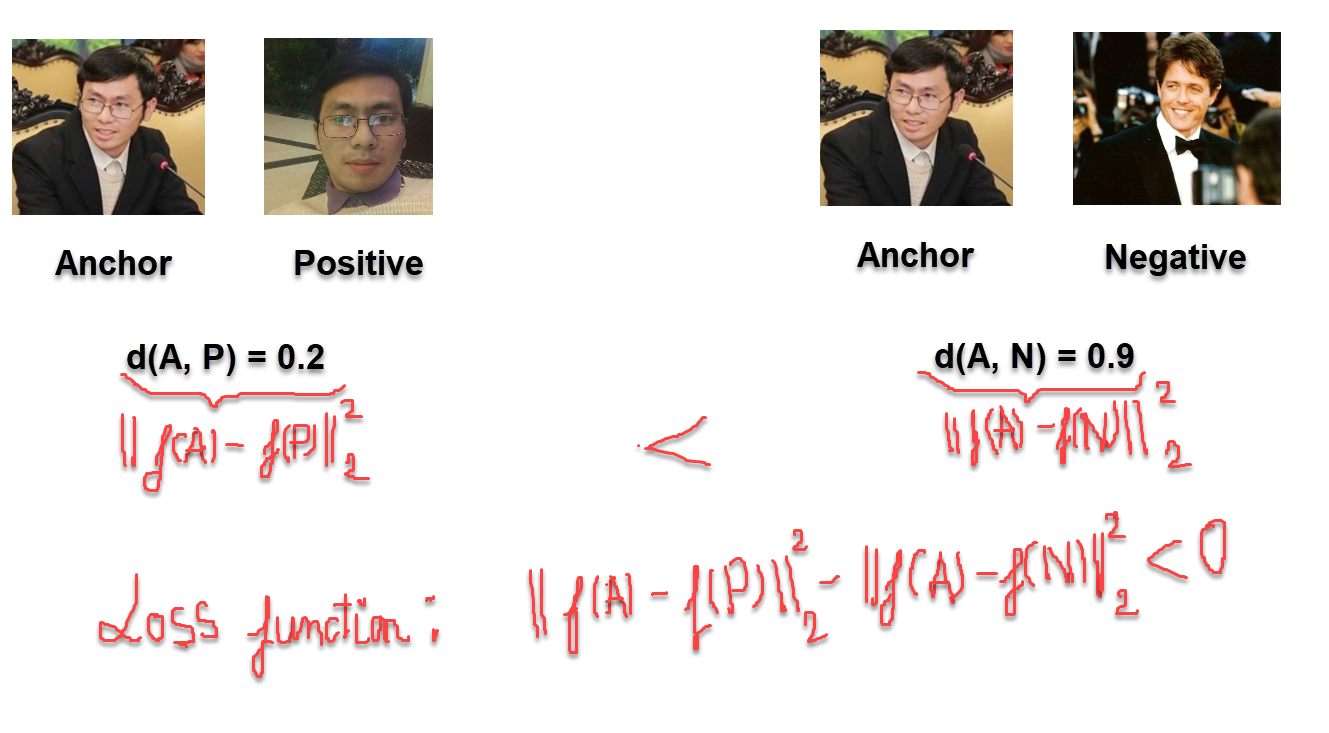
Base network áp dụng một mạng convolutional neural network và giảm chiều dữ liệu xuống chỉ còn 128 chiều. Do đó quá trình suy diễn và dự báo nhanh hơn và đồng thời độ chính xác vẫn được đảm bảo.

Sử dụng loss function là hàm triplot loss có khả năng học được **đồng thời** sự giống nhau giữa 2 bức ảnh cùng nhóm và phân biệt các bức ảnh không cùng nhóm. Do đó hiệu quả hơn rất nhiều so với các phương pháp trước đây.

1. **Triple loss**

Trong facenet, quá trình encoding của mạng convolutional neural network đã giúp ta mã hóa bức ảnh về 128 chiều. Sau đó những véc tơ này sẽ làm đầu vào cho hàm loss function đánh giá khoảng cách giữa các véc tơ.

Để áp dụng triple loss, chúng ta cần lấy ra 3 bức ảnh trong đó có một bức ảnh là anchor. Chắc bạn còn nhớ khái niệm về anchor box đã được trình bày tại [Bài 25 - YOLO You Only Look Once](https://phamdinhkhanh.github.io/2020/03/09/DarknetAlgorithm.html#5-anchor-box) chứ. Anchor image cũng có tác dụng gần như vậy. Trong 3 ảnh thì ảnh anchor được cố định trước. Chúng ta sẽ lựa chọn 2 ảnh còn lại sao cho một ảnh là negative (của một người khác với anchor) và một ảnh là positive (cùng một người với anchor).



Kí hiệu ảnh Anchor, Positive, Negative lần lượt là A,P,N.

Mục tiêu của hàm loss function là **tối thiểu hóa khoảng cách giữa 2 ảnh khi chúng là negative**và**tối đa hóa khoảng cách khi chúng là positive**. Như vậy chúng ta cần lựa chọn các bộ 3 ảnh sao cho:

Ảnh Anchor và Positive khác nhau nhất: cần lựa chọn để khoảng cách d(A,P) lớn. Điều này cũng tương tự như bạn lựa chọn một ảnh của tôi hồi nhỏ so với hiện tại để thuật toán học khó hơn. Nhưng nếu nhận biết được thì nó sẽ thông minh hơn.

Ảnh Anchor và Negative giống nhau nhất: cần lựa chọn để khoảng cách d(A,N) nhỏ. Điều này tương tự như việc thuật toán phân biệt được ảnh của một người anh em giống bạn với bạn.

Triplot loss function luôn lấy 3 bức ảnh làm input và trong mọi trường hợp ta kì vọng:

(1)d(A,P)<d(A,N)

Để làm cho khoảng cách giữa vế trái và vế phải lớn hơn, chúng ta sẽ cộng thêm vào vế trái một hệ số α không âm rất nhỏ. Khi đó (1) trở thành:

d(A,P)+α≤d(A,N)→||f(A)−f(P)||22+α≤||f(A)−f(N)||22→||f(A)−f(P)||22−||f(A)−f(N)||22+α≤0

Như vậy hàm loss function sẽ là:

L(A,P,N)=∑i=0n||f(Ai)−f(Pi)||22−||f(Ai)−f(Ni)||22+α

Trong đó n là số lượng các bộ 3 hình ảnh được đưa vào huấn luyện.

Sẽ không ảnh hưởng gì nếu ta nhận diện đúng ảnh Negative và Positive là cùng cặp hay khác cặp với Anchor. Mục tiêu của chúng ta là giảm thiểu các trường hợp hợp mô hình nhận diện sai ảnh Negative thành Postive nhất có thể. Do đó để loại bỏ ảnh hưởng của các trường hợp nhận diện đúng Negative và Positive lên hàm loss function. Ta sẽ điều chỉnh giá trị đóng góp của nó vào hàm loss function về 0.

Tức là nếu:

||f(A)−f(P)||22−||f(A)−f(N)||22+α≤0

sẽ được điều chỉnh về 0. Khi đó hàm loss function trở thành:

L(A,P,N)=∑i=0nmax(||f(Ai)−f(Pi)||22−||f(Ai)−f(Ni)||22+α,0)

Như vậy khi áp dụng Triple loss vào các mô hình convolutional neural network chúng ta có thể tạo ra các biểu diễn véc tơ tốt nhất cho mỗi một bức ảnh. Những biểu diễn véc tơ này sẽ phân biệt tốt các ảnh Negative rất giống ảnh Positive. Và đồng thời các bức ảnh thuộc cùng một label sẽ trở nên gần nhau hơn trong không gian chiếu euclidean.

Một chú ý quan trọng khi huấn luyện mô hình siam network với triplot function đó là chúng ta luôn phải xác định trước cặp ảnh (A,P) thuộc về cùng một người. Ảnh N sẽ được lựa chọn ngẫu nhiên từ các bức ảnh thuộc các nhãn còn lại. Do đó cần thu thập ít nhất 2 bức ảnh/1 người để có thể chuẩn bị được dữ liệu huấn luyện.

1. **Lựa chọn triple images input**

Nếu lựa chọn triple input một cách ngẫu nhiên có thể ảnh khiến cho bất đẳng thức (1) dễ dàng xảy ra vì trong các ảnh ngẫu nhiên, khả năng giống nhau giữa 2 ảnh là rất khó. Hầu hết các trường hợp đều thỏa mãn bất đẳng thức (1) và không gây ảnh hưởng đến giá trị của loss function do giá trị của chúng được set về 0. Như vậy việc học những bức ảnh Negative quá khác biệt với Anchor sẽ không có nhiều ý nghĩa.

Để mô hình khó học hơn và đồng thời cũng giúp mô hình phân biệt chuẩn xác hơn mức độ giống và khác nhau giữa các khuôn mặt, chúng ta cần lựa chọn các input theo bộ 3 khó học (hard triplets).

Ý tưởng là chúng ta cần tìm ra bộ ba (A,N,P) sao cho (1) là gần đạt được đẳng thức (xảy ra dấu =) nhất. Tức là d(A,P) lớn nhất và d(A,N) nhỏ nhất. Hay nói cách khác với mỗi Anchor A cần xác định:

**Hard Positive:** Bức ảnh Positive có khoảng cách xa nhất với Anchor tương ứng với nghiệm:

argmaxPi(d(Ai,Pi))

**Hard Negative:** Bức ảnh Negative có khoảng cách gần nhất với Anchor tương ứng với nghiệm:

argminNj(d(Ai,Nj))

Với i,j là nhãn của người trong ảnh.

Việc tính toán các trường hợp Hard Positive và Hard Negative có thể được thực hiện offline và lưu vào checkpoint hoặc có thể tính toán online trên mỗi mini-batch.

Chiến lược lựa chọn Triple images sẽ có ảnh hưởng rất lớn tới chất lượng của mô hình Facenet. Nếu lựa chọn Triplet images tốt, Facenet sẽ hội tụ nhanh hơn và đồng thời kết quả dự báo chuẩn xác hơn. Lựa chọn ngẫu nhiên dễ dẫn tới thuật toán không hội tụ.

## 2.7.5. Các pretrain model Facenet

Hầu hết chúng ta khi xây dựng một thuật toán nhận diện khuôn mặt sẽ không cần phải train lại mô hình facenet mà tận dụng lại các mô hình pretrain sẵn có. Bạn sẽ không cần phải tốn thời gian và công sức nếu không có đủ tài nguyên và dữ liệu. Đó cũng là lý do tôi cho rằng việc xây dựng mô hình nhận diện khuôn mặt ở thời điểm hiện tại rất dễ dàng.

Những mô hình pretrain được huấn luyện trên các dữ liệu lên tới hàng triệu ảnh. Do đó có khả năng mã hóa rất tốt các bức ảnh trên không gian 128 chiều. Việc còn lại của chúng ta là sử dụng lại mô hình, tính toán embedding véc tơ và huấn luyện embedding véc tơ bằng một classifier đơn giản để phân loại classes.

1. **Một số bộ dữ liệu public về face**

[CASIA-WebFace](http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html): Bộ dữ liệu bao gồm gần 500k ảnh được thu thập từ khoảng 10k người.

[VGGFace2](https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/): Bộ dữ liệu gồm khoảng 3 triệu ảnh được thu thập từ gần 9k người.

Trên là 2 bộ dữ liệu về face phổ biến nhất, được sử dụng nhiều trong các bài báo và nghiên cứu về face recognition.

Các bộ dữ liệu từ Trung Quốc cũng khá tốt. Những dữ liệu này được Baidu public để hỗ trợ các nhà nghiên cứu: [baidu dataset](https://ai.baidu.com/broad)

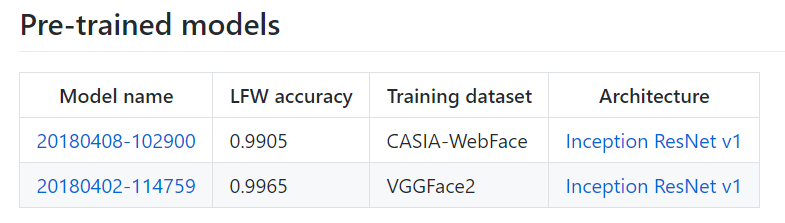
Ngoài ra bạn có thể tìm kiếm các bộ dữ liệu từ google, facebook, flickr và các phòng lab đã được liệt kê tại [top 15 free image datasets for facial recognition](https://lionbridge.ai/datasets/5-million-faces-top-15-free-image-datasets-for-facial-recognition/).

Hiện tại dữ liệu face cho người Việt thì tôi mới chỉ thấy trong cuộc thi [nhận diện người nổi tiếng - aivivn](https://www.aivivn.com/contests/2). Bộ dữ liệu này đã được các bạn AI engineer bên Sun\* đóng góp cho cuộc thi miễn phí. Tôi rất ghi nhận sự đóng góp của các bạn đối với cộng đồng AI việt. Tôi hi vọng tương lai sẽ có những bộ dữ liệu về face lớn hơn nữa cho người Việt. Tài nguyên ảnh trên mạng khá là nhiều, đặc biệt là dữ liệu về face. Một vài tip web crawler có lẽ sẽ hữu ích.

1. **Một số pretrain model phổ biến**

Các pretrain model cho facenet khá nhiều:

Bạn có thể sử dụng pretrain model từ [facenet repo - davidsandberg](https://github.com/davidsandberg/facenet)



Kiến trúc mà tác giả sử dụng là Inception ResNetv1 trên 2 bộ dữ liệu là CASIA-WebFace và VGGFace2

Ngoài ra còn rất nhiều các pretrain model khác nằm rải rác trên các nguồn khác nhau.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG PHẦN MỀM

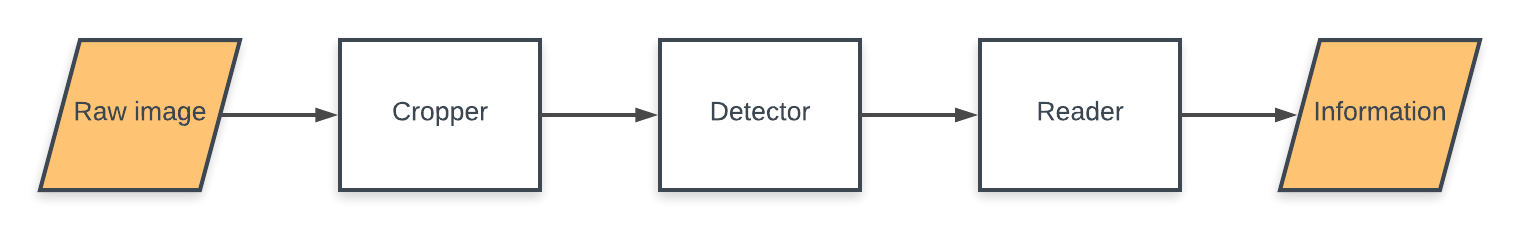
# 3.1. Xác định yêu cầu phần mềm

## 3.1.1. Khảo sát yêu cầu

Phần mềm thẻ sinh viên điện tử trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên yêu cầu các chức năng cho từng phân hệ người dùng: admin, sinh viên. Trong đó, quyền admin có quyền cao nhất, cho phép thêm sửa xóa thông tin tài khoản sinh viên, … Quyền của sinh viên được phép chỉnh sửa hồ sơ cá nhân, xem kết quả học tập, xem lịch học, xem lịch thi, xem tin tức và đăng ký học lại trên ứng dụng. Tất cả các người dùng đều phải thực hiện đăng nhập trước khi sử dụng các chức năng của phần mềm.

Bài toán nhận diện thẻ sinh viên

Đây là một bài toán không mới và cũng có khá nhiều bạn coi đây như một dự án để thực hành kĩ năng xử lý với các mô hình Deep Learning. Tôi khuyên các bạn nên thực hành bài toán này vì nó sẽ giúp cho các bạn có điều kiện làm quen với nhiều loại mô hình khác nhau trong Deep Learning như Object Detection, Instance Segmentation, Optical Character Recognition. Pipeline cơ bản của nó có thể giải quyết như flow sau đây:



* **Cropper** hay còn gọi là **alignment ảnh** nhận đầu vào từ ảnh raw data. Crop vùng chứa thẻ sinh viên trong ảnh. Sử dụng Geometric Transform để xoay ảnh cho đúng chiều
* **Detector** được sử dụng để detect các thành phần trong ảnh như họ tên, ngày tháng năm sinh ....
* **Reader** là một module OCR để đọc nội dung chữ viết từ các thành phần đã được crop.

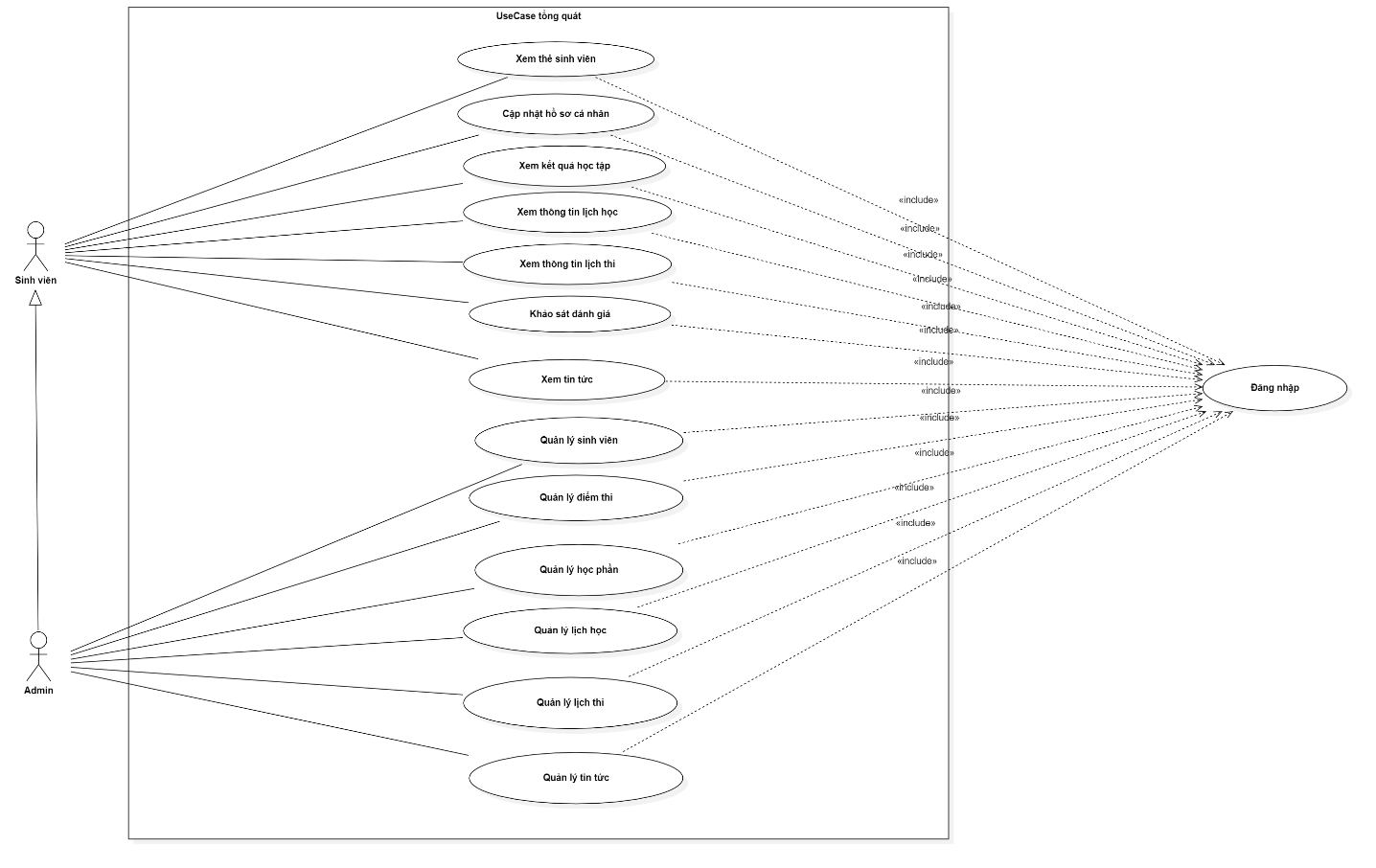
Cuối cùng tổng hợp và sắp xếp lại ta được một bài toán nhận diện thẻ sinh viên hoàn chỉnh. OK giờ chúng ta đi vào phần chính sẽ được giải quyết trong ngày hôm nay nhé

## 3.1.2. Phân tích yêu cầu phần mềm

Dựa vào những yêu cầu đã nêu tại phần trên, các chức năng của phần mềm sẽ được phân tích và nêu rõ trong bảng dưới đây:

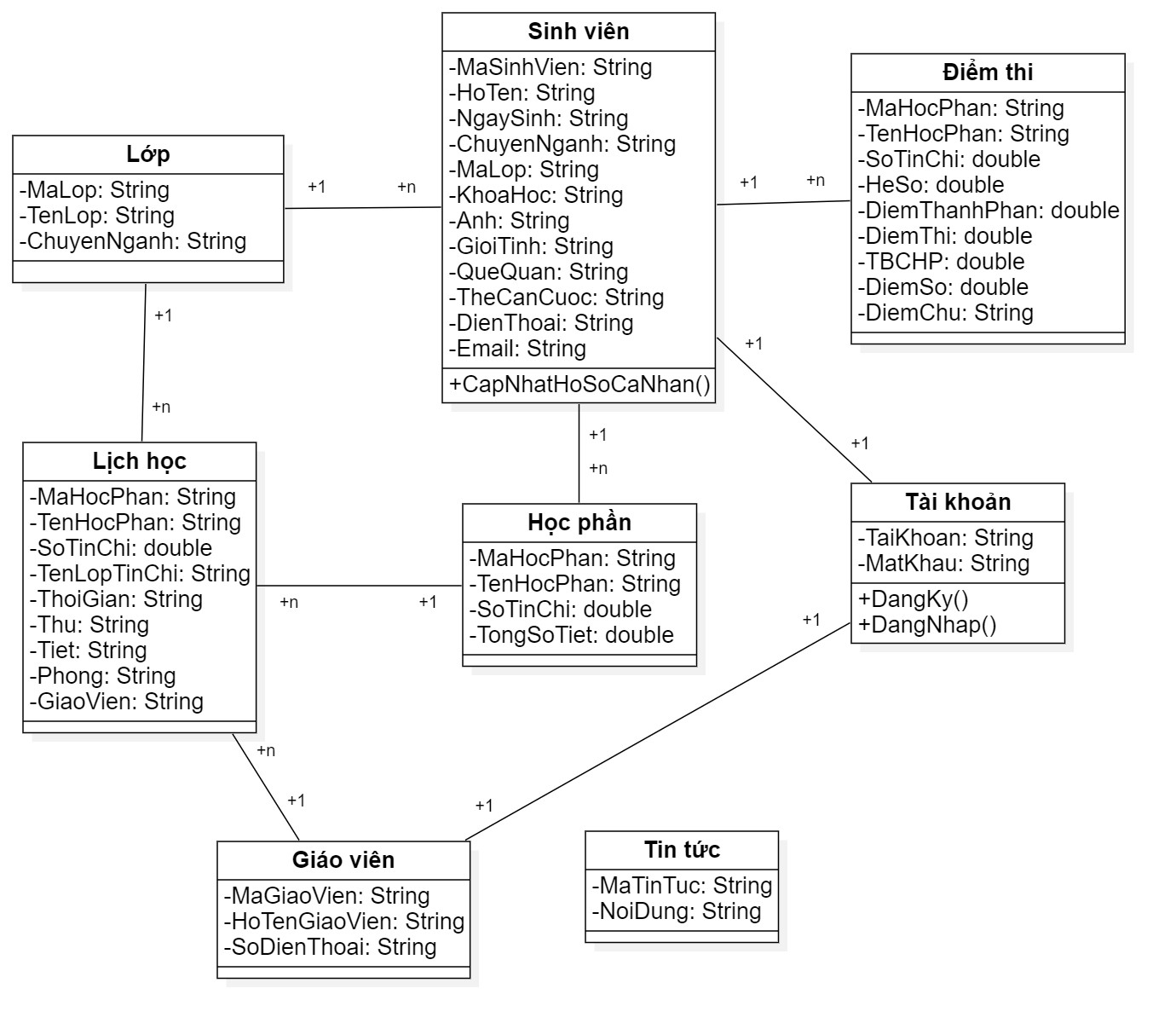
Bảng 3.1 Bảng phân tích các yêu cầu phần mềm

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| **I** | **Các yêu cầu chức năng nghiệp vụ** | Là các chức năng của phần mềm tương ứng với các công việc trong thế giới thực. |
| 1 | Tin tức nhà trường | Hiển thị thông tin tin tức của trường ĐHSPKT Hưng Yên như địa chỉ trang web http://www.utehy.edu.vn/#/, phụ huynh có thể đọc và hành động xem các bài viết khác tại trang web này. |
| 2 | Truy cập cổng thông tin sinh viên | Hiển thị thông tin trang web Cổng thông tin sinh viên trường ĐHSPKT Hưng Yên tại địa chỉ “http://qldaotao.utehy.edu.vn:81/” |
| 3 | Hồ sơ cá nhân (Sinh viên) | Hiển thị thông tin sinh viên, cho phép sinh viên cập nhật thông tin |
| 4 | Kết quả học tập (Sinh viên) | Hiển thị kết quả học tập của sinh viên |
| 5 | Thông tin lịch học (Sinh viên) | Hiển thị thông tin lịch học |
| 6 | Thông tin lịch thi (Sinh viên) | Hiển thị thông tin lịch thi |
| 7 | Đăng ký học lại/cải thiện (Sinh viên) | Cho phép sinh viên thực hiện đăng ký học lại |
| 8 | Quản lý sinh viên (Admin) | Cho phép quản lý thông tin của sinh viên, cập nhật thông tin sinh viên |
| 9 | Quản lý điểm thi (Admin) | Cho phép quản lý điểm thi của sinh viên |
| 10 | Quản lý học phần (Admin) | Cho phép quản lý học phần |
| 11 | Quản lý lịch học (Admin) | Cho phép quản lý lịch học, cập nhật thay đổi lịch học |
| 12 | Quản lý lịch thi (Admin) | Cho phép quản lý lịch học, cập nhật thay đổi lịch thi |
| 13 | Quản lý tin tức (Admin) | Cho phép quản lý tin tức |



Hình 3.1. Sơ đồ use-case mức độ tổng quát của hệ thống

## 3.1.2. Biểu đồ lớp



Hình 3.2. Biểu đồ lớp

* + Mô tả cấu trúc bảng:
* Bảng lớp:

Bảng 3.2 Cấu trúc bảng lớp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaLop | NVARCHAR(10) | PRIAMR KEY | Mã lớp |
| 2 | TenLop | NTEXT |  | Tên lớp |
| 3 | ChuyenNganh | NTEXT |  | Chuyên ngành |

* Bảng tin tức:

Bảng 3.3 Cấu trúc bẳng tin tức

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaTinTuc | NVARCHAR(10) | PRIAMR KEY | Mã tin tức |
| 2 | NoiDung | NTEXT |  | Nội dung tin tức |

* Bảng sinh viên:

Bảng 3.4 Cấu trúc bảng sinh viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaSinhVien | NVARCHAR(10) | PRIAMR KEY | Mã tin tức |
| 2 | HoTen | NTEXT | NOT NULL | Nội dung tin tức |
| 3 | NgaySinh | NTEXT | NOT NULL | Ngày sinh |
| 4 | MaLop | NTEXT | FOREIGN KEY | Mã lớp |
| 5 | KhoaHoc | NTEXT | NOT NULL | Khóa học |
| 6 | Anh | NTEXT | NOT NULL | Ảnh sinh viên |
| 7 | GioiTinh | NTEXT |  | Giới tính |
| 8 | QueQuan | NTEXT |  | Quê quán |
| 9 | TheCanCuoc | NTEXT |  | Thẻ căn cước |
| 10 | DienThoai | NTEXT |  | Số điện thoại của sinh viên |
| 11 | Email | NTEXT |  | Email |

* Bảng giáo viên:

Bảng 3.5 Cấu trúc bảng giáo viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaGiaoVien | NVARCHAR(10) | PRIAMR KEY | Mã giáo viên |
| 2 | HoTenGiaoVien | NTEXT |  | Họ tên giáo viên |
| 3 | SoDienThoai | NTEXT |  | Số điện thoại của giáo viên |

* Bảng điểm thi:

Bảng 3.6 Cấu trúc bảng điểm thi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaHocPhan | NTEXT | FOREIGN KEY | Mã tin tức |
| 2 | TenHocPhan | NTEXT |  | Nội dung tin tức |
| 3 | SoTinChi | DOUBLE |  | Số tín chỉ |
| 4 | HeSo | DOUBLE |  | Hệ số |
| 5 | DiemThanhPhan | DOUBLE |  | Điểm thành phần |
| 6 | DiemThi | DOUBLE |  | Điểm thi |
| 7 | TBCHP | DOUBLE |  | Trung bình cộng học phần |
| 8 | DiemSo | DOUBLE |  | Điểm bằng số |
| 9 | DiemChu | NTEXT |  | Điểm bằng chữ |

* Bảng học phần:

Bảng 3.7 Cấu trúc bảng học phần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaHocPhan | NVARCHAR(10) | PRIAMR KEY | Mã học phần |
| 2 | TenHocPhan | NTEXT |  | Tên học phần |
| 3 | SoTinChi | DOUBLE |  | Số tín chỉ |
| 4 | TongSoTiet | DOUBLE |  | Tổng số tiết |

* Bảng lịch học:

Bảng 3.8 Cấu trúc bảng lịch học

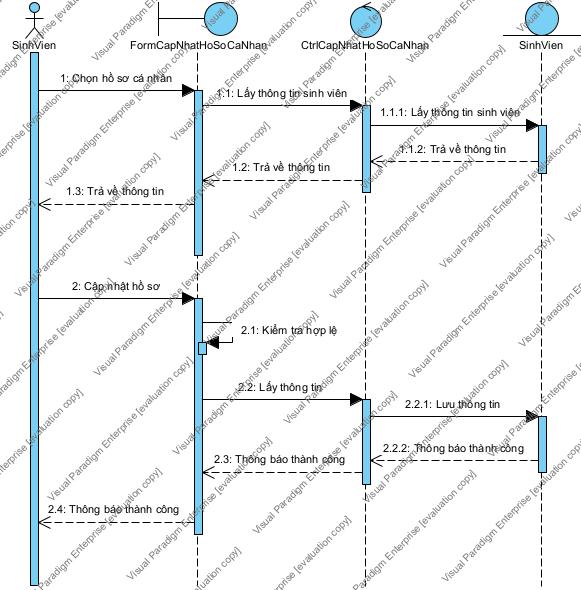
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaHocPhan | NTEXT | FOREIGN KEY | Mã học phần |
| 2 | TenHocPhan | NTEXT |  | Tên học phần |
| 3 | SoTinChi | DOUBLE |  | Số tín chỉ |
| 4 | TenLopTinChi | NTEXT |  | Tên lớp tín chỉ |
| 5 | ThoiGian | NTEXT |  | Thời gian |
| 6 | Thu | NTEXT |  | Thứ |
| 7 | Tiet | NTEXT |  | Tiết |
| 8 | Phong | NTEXT |  | Phòng |
| 9 | GiaoVien | NTEXT | FOREIGN KEY | Giáo viên giảng dạy |

* Bảng lịch thi:

Bảng 3.9 Cấu trúc bảng lịch thi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaHocPhan | NTEXT | FOREIGN KEY | Mã học phần |
| 2 | TenHocPhan | NTEXT |  | Tên học phần |
| 3 | SoTinChi | DOUBLE |  | Số tín chỉ |
| 4 | NgayThi | NTEXT |  | Ngày thi |
| 5 | GioThi | NTEXT |  | Giờ thi |
| 6 | LanThi | INTEGER |  | Lần thi |
| 7 | DotThi | INTEGER |  | Đợt thi |
| 8 | PhongThi | NTEXT |  | Phòng thi |
| 9 | HinhThuc | NTEXT |  | Hình thức |
| 10 | CanBoCoiThi | NTEXT | FOREIGN KEY | Cán bộ coi thi |

## 3.1.3. Biểu đồ tuần tự



Hình 3.3. Biểu đồ tuần tự cập nhật hồ sơ cá nhân

## 3.1.4. Các yêu cầu phi chức năng

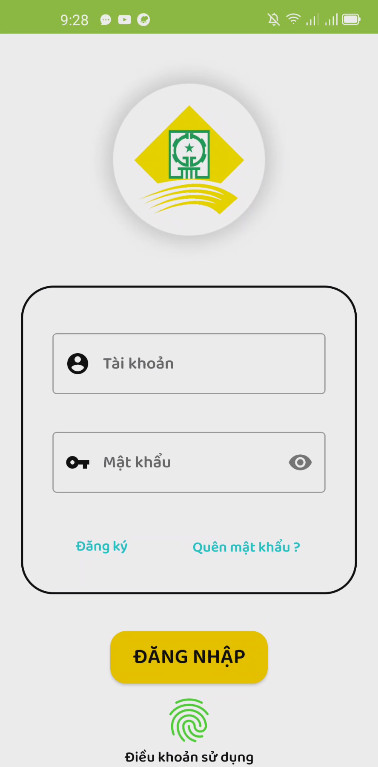
* Phần mềm yêu cầu một số yêu cầu phi chức năng như sau:
* Phần mềm yêu cầu người dùng đăng nhập trước khi sử dụng chức năng.
* Phần mềm yêu cầu có giao diện dễ dùng, không cần quá bắt mắt nhưng đẩy đủ các chức năng đã nêu trên.
* Phần mềm yêu cầu kết nối Internet để có thể sử dụng.
* Phần mềm yêu cầu lưu trữ dữ liệu trên mạng, các thông tin phải được cập nhật realtime.

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÁNH GIÁ

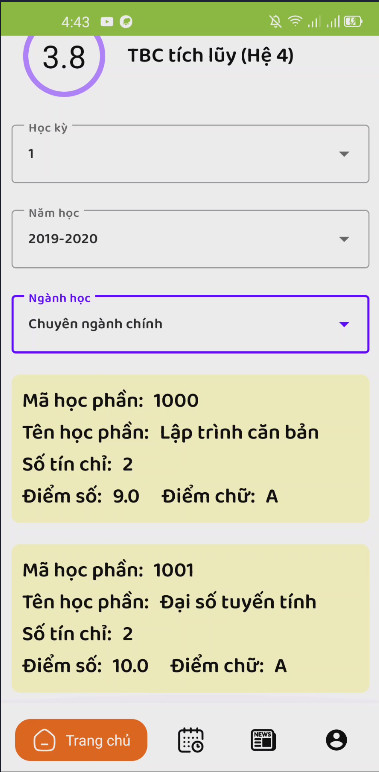
## 4.1. Các kết quả đã đạt được

Sau thời gian nghiên cứu và thực hiện xây dựng hệ thống, cho tới nay phần mềm đã đạt được một số kết quả dưới đây.

## 4.1.1. Về các chức năng phần mềm

Hình 4.1. Giao diện khởi động của phần mềm Hình 4.2. Giao diện đăng nhập

Hình 4.3. Giao diện trang chủ Hình 4.4. Giao diện xem thành tích cá nhân

Hình 4.5. Giao diện xem lịch học, lịch thi Hình 4.6. Giao diện tài khoản

## 

## 4.1.2. Về tính ứng dụng của phần mềm vào thực tế

Với các chức năng cơ bản đã nêu ở trên, phần mềm đã ứng dụng được một phần tích cực trong việc tạo môi trường liên lạc giữa các sinh viên trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên.

* Các sinh viên có thể dễ dàng cập nhật thông tin chỉ với một chiếc điện thoại thông minh.
* Các sinh viên dễ dàng xem được thời khóa biểu ngày hôm nay qua màn hình chính của phần mềm.
* Thông tin được cập nhật nhanh chóng và an toàn thông qua Firebase Realtime DB.

# 4.2. Đánh giá các kết quả đã đạt được

Phần mềm hỗ trợ thông tin cho sinh viên trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên đã đạt được một số kết quả tích cực như hỗ trợ việc cập nhật thông tin của các sinh viên trở lên nhanh chóng và thuận tiện hơn.

Tuy nhiên, phần mềm vẫn còn một số hạn chế nhất định. Thứ nhất, do chưa thể kết nối tới dữ liệu của nhà trường nên cơ sở dữ liệu của phần mềm còn hạn chế, cho nên phần mềm chỉ là mô hình ý tưởng được xây dựng nhằm hỗ trợ các sinh viên trong việc liên lạc và cập nhật thông tin. Thứ hai, do ứng dụng cập nhật thông tin dựa trên cơ sở dữ liệu được lưu trên nền tảng Firebase, vì vậy khi muốn sử dụng phần mềm người dùng cần kết nối Internet để có thể truy cập các chức năng của hệ thống.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# 5.1. Kết luận về đề tài

## 5.1.1. Về ưu điểm đã đạt được

* Các sinh viên có thể dễ dàng cập nhật thông tin chỉ với một chiếc điện thoại thông minh.
* Các sinh viên dễ dàng xem được thời khóa biểu ngày hôm nay qua màn hình chính của phần mềm.
* Các sinh viên của thẻ xem được thẻ sinh viên điện tử của tôi.
* Thông tin được cập nhật nhanh chóng và an toàn thông qua Firebase Realtime DB.
* Có thể dễ dàng ứng dụng trong môi trường Trường Đại Học Sư Phạm Kĩ Thuật Hưng Yên.
* Giảm tải thời gian làm việc trong việc liên lạc và cập nhật thông tin thông qua một số chức năng như gửi thông báo, xem thời khóa biểu, xem lịch hoạt động.

## 5.1.2. Về một số hạn chế của phần mềm

* Phần mềm vẫn chưa thực sự tối ưu trong việc bảo mật tài khoản của người dùng.
* Cơ sở dữ liệu còn hạn chế, chưa khai thác được hết tất cả dữ liệu của người dùng.
* Phần mềm chỉ bao gồm các chức năng cơ bản để hỗ trợ thông tin, chưa có thực sự nhiều chức năng.

# 5.2. Một số kiến nghị về phần mềm

Do phần mềm được xây dựng dựa trên lượng kiến thức được học trên trường và tự tìm tòi học hỏi nên phần mềm chưa được thực sự tối ưu, vì vậy nhóm có một số kiến nghị như sau :

* Mong muốn nhà trường cho phép được kết nối tới API dữ liệu của trường để có thể tối ưu nhất về mặt cơ sở dữ liệu cho hệ thống.
* Nếu ứng dụng được đưa vào sử dụng trong nhà trường thì cần rất nhiều không gian lưu trữ trên nền tảng Firebase, vì vậy nguồn kinh phí để mở rộng gói lưu trữ là cần thiết.
* Mong muốn được học hỏi thêm về kĩ năng thiết kế giao diện và lập trình của các thầy, cô trong khoa Công nghệ thông tin để ngày một hoàn thiện phần mềm hơn.

# 5.3. Hướng phát triển của đề tài

* Tiếp tục phát triển thêm các chức năng để hoàn thiện ứng dụng.
* Củng cố bảo mật tài khoản người dùng.
* Thiết kế giao diện bắt mắt hơn.
* Tăng tốc độ xử lý của các chức năng trong phần mềm.

TÀI LIỆU THAM KHẢO

1. *Giáo trính môn Chuyên đề 3: lập trình dịch vụ mạng trên Android,* Bộ môn CNPM, Trường ĐHSPKTHY.
2. *Giáo trính môn Chuyên đề 1: lập trình Android cơ bản,* Bộ môn CNPM, Trường ĐHSPKTHY.
3. *Giáo trính môn Chuyên đề 2: lập trình đa phương tiện trên Android,* Bộ môn CNPM, Trường ĐHSPKTHY.
4. *Giáo trính môn lập trình hướng đối tượng với Java,* Bộ môn CNPM, Trường ĐHSPKTHY.
5. *Giáo trính môn Cơ sở dữ liệu*, Bộ môn CNPM, Trường ĐHSPKTHY.
6. Ed Burnette, *Hello, Android Introducing Google’s Mobile Development Platform*, The Pragmatic Bookshelf, 2010.
7. Marziah Karch, *Android for Work Productivity for Professional*, Press, 2010.
8. J.F. DiMarzio, *Android A Programmer’s Guide*, The McGraw-Hill Companies, 2008.
9. Chris Haseman, *Android Essentials*, Apress, 2008.
10. Sams, *Android Application Development*, Lauren Darcey and Shane Conder, 2010.
11. stackoverflow.com, <https://stackoverflow.com>