

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG



Công trình Nghiên cứu khoa học sinh viên năm học 2023 - 2024

HỆ THỐNG TRUY VẤN KHUÔN MẶT
TRONG TẬP DỮ LIỆU LỚN
CHO ỨNG DỤNG ANDROID VÀ FLASK BACKEND

ĐƠN VỊ


GIẢNG VIÊN HƯỚNG DẪN:

ThS. NGUYỄN THÀNH AN

SINH VIÊN/NHÓM SINH VIÊN THỰC HIỆN:

- 1. TRẦN NHỰT QUANG*
- 2. PHẠM DUY KHOA*
- 3. NGUYỄN KHẮC ANH TÀI*


TP. Hồ Chí Minh, tháng 6 năm 2024

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKH/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	---	---

TÓM TẮT CÔNG TRÌNH


Hiện nay, với sự phát triển mạnh mẽ của các nền tảng mạng xã hội, việc nhận diện các người nổi tiếng trở nên thách thức vì lượng hình ảnh lớn và đa dạng. Nhầm lẫn và sự không chính xác trong quá trình nhận diện có thể ảnh hưởng đến trải nghiệm của người dùng và gây ra sự bất tiện khi tìm kiếm thông tin về họ. Để giải quyết vấn đề này, việc sử dụng công nghệ lưu trữ và truy vấn nhanh khuôn mặt trong tập dữ liệu siêu lớn là cực kỳ cần thiết.

Nhóm sinh viên thực hiện đề tài này nhằm mục đích tạo ra giải pháp lưu trữ và truy vấn nhanh khuôn mặt của người nổi tiếng trong tập dữ liệu siêu lớn. Giúp người dùng tìm kiếm nhanh hơn nhân vật được quan tâm, cũng như các nhân vật có ngoại hình tương tự. Đánh giá sự tương đồng về khuôn mặt của bản thân với các nhân vật nổi tiếng. Đề tài có khả năng áp dụng rộng rãi trong giám sát, quản lý dữ liệu khuôn mặt trong cơ sở dữ liệu dân cư quốc gia.


	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKH/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---

MỤC LỤC


TÓM TẮT CÔNG TRÌNH.....	1
MỤC LỤC.....	2
DANH MỤC CÁC CHỮ VIẾT TẮT	5
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	6
CHƯƠNG 1 – TỔNG QUAN VỀ CÔNG TRÌNH.....	7
1.1 Giới thiệu.....	7
1.2 Mục tiêu đề tài và giới hạn	7
1.3 Ý nghĩa đề tài.....	8
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT	10
2.1 Convolutional Neural Networks.....	10
2.1.1 Ưu điểm	10
2.1.2 Nhược điểm	10
2.2 Các mô hình CNN	11
2.2.1 MobileNet	11
2.2.2 Resnet.....	11
2.2.3 EfficientNet.....	12
2.2.4 Xception.....	13
2.3 Pyspark	13
2.3.1 Giới thiệu về Pyspark	13
2.3.2 Các tính năng chính của Pyspark.....	14
2.4 LSH.....	14
2.5 Flask	15
2.5.1 Giới thiệu về Flask.....	15
2.5.2 Các tính năng của Flask.....	16
2.6 Flutter	16

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKH-CN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	--

2.6.1 Giới thiệu về Flutter.....	16
2.6.2 Ưu điểm	17
2.6.3 Nhược điểm	17
CHƯƠNG 3 – MÔ HÌNH NHẬN DẠNG	19
CHƯƠNG 4 – THÍ NGHIỆM VÀ ĐÁNH GIÁ	20
4.1 Tập dữ liệu.....	20
4.2 Thí nghiệm.....	20
4.3 Kết quả.....	21
CHƯƠNG 5 – HỆ THỐNG TRIỂN KHAI	22
5.1 Sơ đồ usecase tổng quát	22
5.2 Đặc điểm nổi bật.....	23
CHƯƠNG 6 - ỨNG DỤNG DI ĐỘNG.....	24
6.1 Chức năng.....	24
6.1.1 Đăng nhập	24
6.1.2 Đăng xuất.....	25
6.1.3 Đổi mật khẩu.....	26
6.1.4 Nhận dạng khuôn mặt.....	28
6.1.5 Xem 5 khuôn mặt người nổi tiếng giống mình nhất.....	29
6.1.6 Xem thông tin những người nổi tiếng giống mình	31
6.2 Giao diện ứng dụng	33
CHƯƠNG 7 – BACKEND SEVER	36
7.1 Database	36
7.2 Recognition Service.....	36
CHƯƠNG 8 – TỔNG KẾT.....	37
8.1 Kết quả.....	37
8.2 Ý nghĩa	37


	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKH-CN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	---	--

8.3 Thuận lợi.....	37
8.4 Khó khăn	38
8.5 Phát triển.....	39
TÀI LIỆU THAM KHẢO.....	40

	<p><u><i>Thủ tục:</i></u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	---	---

DANH MỤC CÁC CHỮ VIẾT TẮT

BERT	Bidirectional Encoder Representations from Transformers
LSH	Locality Sensitive Hashing
CNN	Convolutional Neural Networks
NLP	Natural Language Processing
NSP	Next Sentence Prediction

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---


DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

<i>Hình 1: Logo Pyspark</i>	<i>13</i>
<i>Hình 2: Logo Flask</i>	<i>15</i>
<i>Hình 3: Logo Flutter</i>	<i>17</i>
<i>Hình 4: Kết quả sau khi train.....</i>	<i>21</i>
<i>Hình 5: Sơ đồ usecase tổng quát.....</i>	<i>22</i>
<i>Hình 6: Giao diện chụp ảnh.....</i>	<i>33</i>
<i>Hình 7: Giao diện danh sách 5 người giống nhất</i>	<i>34</i>
<i>Hình 8: Giao diện hình ảnh người giống nhất.....</i>	<i>35</i>

DANH MỤC BẢNG

<i>Bảng 1: Bảng so sánh hiệu suất các mô hình CNN</i>	<i>19</i>
<i>Bảng 2: Bảng thống kê tập dữ liệu.....</i>	<i>20</i>
<i>Bảng 3: Các tác nhân usecase trong hệ thống.....</i>	<i>23</i>
<i>Bảng 4: Đặc tả usecase đăng nhập.....</i>	<i>25</i>
<i>Bảng 5: Đặc tả usecase đăng xuất.....</i>	<i>26</i>
<i>Bảng 6: Đặc tả usecase đổi mật khẩu.....</i>	<i>28</i>
<i>Bảng 7: Đặc tả usecase nhận dạng khuôn mặt</i>	<i>29</i>
<i>Bảng 8: Đặc tả usecase xem 5 khuôn mặt người nổi tiếng giống mình nhất.....</i>	<i>31</i>
<i>Bảng 9: Đặc tả usecase xem thông tin những người nổi tiếng giống mình</i>	<i>32</i>

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

CHƯƠNG 1 – TỔNG QUAN VỀ CÔNG TRÌNH

1.1 Giới thiệu


Trong thời đại số hóa ngày nay, việc quản lý và truy xuất dữ liệu khuôn mặt đang trở thành một thách thức ngày càng lớn. Đặc biệt, với việc tích lũy dữ liệu khuôn mặt từ nhiều nguồn khác nhau, việc tìm kiếm và truy xuất thông tin từ một tập dữ liệu siêu lớn trở nên phức tạp. Trong bối cảnh này, các phương pháp truy vấn nhanh dựa trên mô hình Convolutional Neural Networks (CNN), như MobileNet, và Locality Sensitive Hashing (LSH) đã thu hút sự quan tâm lớn.

Mô hình CNN, đặc biệt là MobileNet, đã chứng minh khả năng xuất sắc trong việc trích xuất đặc trưng từ hình ảnh khuôn mặt, giúp cho quá trình nhận diện và phân loại trở nên chính xác hơn. Tuy nhiên, khi xử lý tập dữ liệu lớn, việc truy vấn trực tiếp vào các biểu diễn đặc trưng của hình ảnh có thể trở nên không hiệu quả do độ phức tạp tính toán cao.

Để giải quyết vấn đề này, kỹ thuật LSH đã được áp dụng. LSH là một phương pháp giảm chiều dữ liệu, giúp tăng tốc độ truy vấn bằng cách ánh xạ các điểm dữ liệu vào các vùng "nhảy cảm địa phương" trên không gian hash. Khi kết hợp với mô hình CNN, LSH giúp tối ưu hóa quá trình truy vấn trong tập dữ liệu khuôn mặt lớn, đồng thời giảm thiểu độ phức tạp tính toán.

Bằng cách kết hợp giữa mô hình CNN như MobileNet và kỹ thuật LSH, nhóm đã xây dựng một hệ thống truy vấn nhanh, linh hoạt và hiệu quả đối với bộ dữ liệu FaceScrub với hơn 50000 tấm ảnh của 530 người nổi tiếng trên toàn thế giới của Đại học Quốc gia Singapore (NUS). Tính linh hoạt của hệ thống cho phép áp dụng nó vào nhiều lĩnh vực khác nhau, từ giải trí đến an ninh, từ quản lý cơ sở dữ liệu ảnh cá nhân đến tổ chức sự kiện lớn.

1.2 Mục tiêu đề tài và giới hạn

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---

Ứng dụng sẽ cho phép người dùng chụp hình khuôn mặt của họ hoặc bất cứ ai và tự động cho ra kết quả khuôn mặt đó giống với khuôn mặt người nổi tiếng nào nhất bằng cách sử dụng các kỹ thuật học sâu.

Bên cạnh đó ứng dụng còn tập trung vào việc xem xét với các người nổi tiếng có mức độ tương đồng thấp hơn giữa các khuôn mặt trong tập dữ liệu.

Để xây dựng được hệ thống ứng dụng có khả năng giải quyết bài toán đặt ra, đề tài này chia làm các bước tính toán chính cần phải giải quyết:


1. Xây dựng mô hình chuyển đổi ảnh khuôn mặt.
2. Cài đặt thuật toán truy vấn nhanh LSH cho tập dữ liệu siêu lớn.
3. Xây dựng backend server vận hành dựa trên Flask.
4. Cài đặt ứng dụng minh họa trên nền tảng Android.
5. Các đề xuất cải tiến để tăng độ chính xác truy vấn.

1.3 Ý nghĩa đề tài


Đề tài này tập trung vào phát triển một ứng dụng hoàn chỉnh cho việc biểu diễn, lưu trữ và truy vấn nhanh chóng các khuôn mặt có trong tập dữ liệu siêu lớn. Mục tiêu là xây dựng một hệ thống mạnh mẽ có khả năng xử lý hàng triệu hoặc thậm chí hàng tỷ hình ảnh khuôn mặt một cách hiệu quả. Mô hình này có thể được áp dụng trong nhiều lĩnh vực khác nhau như giải trí và quản lý cơ sở dữ liệu quốc gia về dân cư.

Trong lĩnh vực giải trí, một mô hình mạnh mẽ về nhận dạng khuôn mặt có thể được sử dụng để cải thiện trải nghiệm người dùng trong ứng dụng trò chơi, ứng dụng tạo hình ảnh hoặc video, cũng như các nền tảng truyền thông xã hội. Nó cung cấp khả năng nhận dạng và tạo ra các gợi ý, bài viết, hoặc thậm chí là bộ lọc dựa trên người dùng có mặt trong hình ảnh hoặc video.

Còn trong lĩnh vực quản lý cơ sở dữ liệu quốc gia về dân cư, việc có một hệ thống mạnh mẽ về nhận dạng khuôn mặt có thể giúp chính phủ hoặc các tổ chức liên

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	---	---

quan trọng việc theo dõi dân số, quản lý hộ chiếu và giấy tờ tùy thân, và thậm chí là trong việc đảm bảo an ninh quốc gia.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	---	--

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

2.1 Convolutional Neural Networks

2.1.1 Định nghĩa

Convolutional Neural Network (CNN) là một dạng mạng nơ-ron được ứng dụng chủ yếu trong việc xử lý ảnh và thị giác máy tính. CNN được tạo ra để tự động học và nhận biết các đặc điểm không gian từ dữ liệu đầu vào, thường là các hình ảnh, bằng cách sử dụng các lớp tích chập.

Mỗi lớp tích chập sử dụng các bộ lọc để rút trích các đặc điểm từ ảnh gốc, tạo ra các bản đồ đặc điểm. Bộ lọc này là các ma trận nhỏ di chuyển trên ảnh, thực hiện phép nhân chập với các pixel tương ứng. Quá trình này giúp mạng học được các mẫu nhỏ như cạnh và góc, từ đó tạo ra các đặc điểm phức tạp hơn.

CNN cũng sử dụng các lớp pooling để giảm kích thước của các bản đồ đặc điểm và làm cho mạng ít nhạy cảm hơn với các biến thể không gian nhỏ. Các phương pháp pooling phổ biến bao gồm max pooling và average pooling, lấy giá trị lớn nhất hoặc trung bình của các khu vực nhỏ trên bản đồ đặc điểm.

Cuối cùng, CNN thường kết thúc bằng các lớp kết nối đầy đủ, nhận đầu ra từ các lớp trước và dự đoán cho ảnh đầu vào.

2.1.2 Ưu điểm


Hiệu suất cao: Xử lý ảnh với độ phân giải và độ phức tạp cao, mang lại độ chính xác cao hơn so với phương pháp truyền thống.

Tự động hóa trích xuất đặc điểm: Học các đặc điểm của ảnh tự động, giúp mô hình phân loại và nhận diện đối tượng tốt hơn.

Đa dạng ứng dụng: Áp dụng cho nhiều bài toán trong thị giác máy tính.

2.1.3 Nhược điểm

Yêu cầu dữ liệu lớn: Để đạt hiệu suất tốt, cần một lượng lớn dữ liệu huấn luyện.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	---	--

Tính toán tốn kém: Yêu cầu nhiều tài nguyên tính toán và thời gian để huấn luyện.

Không hiệu quả đối với các bài toán đơn giản: Có thể dẫn đến quá khớp dữ liệu.

Khó hiểu: Cấu trúc phức tạp, khó hiểu và giải thích.

2.2 Các mô hình CNN

2.2.1 MobileNet


MobileNet là một kiến trúc mạng neural network được Google phát triển cho việc chạy trên các thiết bị di động như điện thoại thông minh và máy tính bảng. Ra mắt vào năm 2017, MobileNet được tối ưu hóa để hoạt động nhanh chóng trên các thiết bị có tài nguyên hạn chế, với kích thước mô hình nhỏ hơn đáng kể so với các mô hình truyền thống như VGG hoặc ResNet.

Điểm đặc biệt của MobileNet là việc sử dụng kỹ thuật depthwise separable convolution để giảm số lượng tham số cần học và tăng tốc độ tính toán. Thay vì sử dụng các lớp convolution truyền thống, MobileNet chia ra thành hai bước riêng biệt: convolution theo chiều sâu (depthwise convolution) và convolution thông thường (pointwise convolution), giúp tiết kiệm tài nguyên và tăng hiệu suất.

MobileNet đã được áp dụng trong nhiều lĩnh vực ứng dụng như xử lý hình ảnh, nhận diện khuôn mặt, phân loại đối tượng, và nhận dạng từ vựng trong các ứng dụng dịch thuật. Đối với các ứng dụng di động, MobileNet đóng vai trò quan trọng trong việc cung cấp hiệu suất cao và trải nghiệm người dùng mượt mà trên các thiết bị có tài nguyên hạn chế.

2.2.2 Resnet

ResNet, hay Residual Network, là một kiến trúc mạng neural network sâu được giới thiệu bởi nhóm nghiên cứu tại Microsoft vào năm 2015. Mục tiêu của ResNet là khắc phục vấn đề "vanishing gradient" phổ biến trong các mạng neural network sâu khi

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	--

độ sâu của mạng tăng, dẫn đến việc gradient truyền ngược giảm dần và gây khó khăn trong quá trình huấn luyện.

ResNet áp dụng kỹ thuật residual learning, cho phép mạng học cách học được sự khác biệt giữa các đầu vào và đầu ra của một hàm số. Điều này giúp gradient truyền ngược được truyền lại một cách hiệu quả hơn từ các lớp trước đó, giúp mô hình học được các đặc trưng phức tạp.


Kiến trúc của ResNet được chia thành các "block", mỗi block có đầu vào và đầu ra giống nhau, với kết nối "skip connection" đưa đầu vào trực tiếp đến đầu ra của block. Điều này giúp mô hình học được các đặc trưng nhỏ và phức tạp hơn, đồng thời giảm thiểu ảnh hưởng của độ sâu mạng lên gradient truyền ngược.

ResNet đã đạt được kết quả ấn tượng trên nhiều bài toán phân loại hình ảnh trên các bộ dữ liệu như ImageNet và CIFAR-10. Nó đã được áp dụng trong nhiều ứng dụng thực tiễn như nhận diện khuôn mặt, phân loại đối tượng trong hình ảnh y tế và phân tích nội dung video.

2.2.3 EfficientNet

EfficientNet là một kiến trúc mạng neural network được Google phát triển với mục tiêu mang lại hiệu suất cao và độ chính xác tốt trong việc phân loại hình ảnh, đồng thời đảm bảo kích thước mô hình nhỏ gọn và tốc độ tính toán nhanh. Công bố vào năm 2019, kiến trúc này đã nhanh chóng thu hút sự chú ý trong cộng đồng machine learning.

EfficientNet sử dụng kỹ thuật scaling để tăng cường hiệu suất của mô hình. Thay vì tăng kích thước mô hình một cách tùy tiện, kỹ thuật này tăng đồng đều và cân bằng độ rộng (width), độ sâu (depth) và độ phân giải (resolution) của mạng. Bằng cách kết hợp ba yếu tố này, EfficientNet đạt được sự cân bằng tốt giữa độ chính xác và kích thước mô hình.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

EfficientNet đã đạt được kết quả ấn tượng trên các bộ dữ liệu phân loại hình ảnh quan trọng như ImageNet và CIFAR-10. Các phiên bản của EfficientNet đã được phát hành dưới dạng mã nguồn mở, giúp các nhà nghiên cứu và nhà phát triển áp dụng kiến trúc này vào các ứng dụng thực tế, từ phân loại hình ảnh đến các ứng dụng trong lĩnh vực nhận dạng và xử lý hình ảnh.

2.2.4 Xception

Xception là một kiến trúc mạng neural network được phát triển dựa trên mạng Inception và được giới thiệu bởi nhóm nghiên cứu tại Google vào năm 2017. Tên gọi "Xception" kết hợp giữa "eXtreme" và "Inception".

Mục tiêu của Xception là tạo ra một kiến trúc mạng neural network có khả năng học sâu và độ chính xác tốt hơn so với các kiến trúc truyền thống, đồng thời vẫn giữ được tính linh hoạt trong việc thích ứng với các tác vụ khác nhau. Kỹ thuật chính trong Xception là depthwise separable convolution, tương tự như trong MobileNet. Tuy nhiên, Xception sử dụng cấu trúc của mạng Inception để tăng cường tính linh hoạt của mô hình.

Xception đã đạt được kết quả tốt trên nhiều nhiệm vụ phân loại ảnh và nhận diện vật thể trong các bộ dữ liệu như ImageNet và COCO. Nó đã được sử dụng trong nhiều ứng dụng thực tế như nhận diện khuôn mặt, phân loại đối tượng trong hình ảnh y tế và phân tích nội dung video.


2.3 Pyspark

2.3.1 Giới thiệu về Pyspark

PySpark là một giao diện lập trình ứng dụng (API) Python cho Apache Spark, một công cụ mạnh mẽ giúp thực hiện các tác vụ xử lý big data và thời gian thực trong một môi trường phân tán, sử dụng ngôn ngữ lập trình Python.



Hình 1: Logo Pyspark

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---

2.3.2 Các tính năng chính của Pyspark

Xử Lý Thời Gian Thực: PySpark chuyên về xử lý dữ liệu trong bộ nhớ và có thể tính toán trên dữ liệu lớn một cách thời gian thực. Nhờ đó, nó giảm thiểu độ trễ và có thể xử lý dữ liệu từ nhiều nguồn khác nhau theo thời gian thực bằng Spark Streaming.

Hỗ Trợ Nhiều Ngôn Ngữ: PySpark hỗ trợ nhiều ngôn ngữ lập trình như Scala, Java, Python và R. Điều này khiến nó trở thành framework lý tưởng để xử lý big data với sự tương tác linh hoạt. Spark dùng máy chủ RPC để cho phép các ngôn ngữ khác tiếp cận API của nó.


Đồng Nhất của Dữ Liệu và Caching: PySpark có tính năng caching mạnh mẽ và bảo đảm tính nhất quán của dữ liệu trên ổ đĩa. Tuy nhiên, cần phải dùng caching một cách cẩn trọng để tránh các vấn đề về tính nhất quán khi dữ liệu bị thay đổi bất ngờ hoặc có sự cố về điện.

Xử Lý Nhanh Chóng: PySpark giúp xử lý dữ liệu nhanh chóng, với hiệu suất cao gấp 100 lần trong bộ nhớ và gấp 10 lần trên ổ đĩa so với các phương pháp cũ. Điều này giúp cải thiện hiệu suất xử lý và rút ngắn thời gian chờ cho các tác vụ tính toán.

Hiệu Quả Với RDD: Ngôn ngữ lập trình Python linh hoạt giúp việc làm việc với RDD dễ dàng hơn. RDD (Resilient Distributed Dataset) là công cụ để thực hiện các thao tác dữ liệu nhanh chóng và hiệu quả trong PySpark. RDD có thể lưu trữ dữ liệu trong bộ nhớ và tái sử dụng, làm cho nó trở thành công cụ mạnh mẽ để xử lý và phân tích big data.

2.4 LSH

Trong Apache Spark, LSH (Locality Sensitive Hashing) là một kỹ thuật được sử dụng để tìm các điểm dữ liệu "gần nhau" trong không gian đa chiều một cách hiệu quả. Điều này hữu ích trong các tác vụ như phân cụm dữ liệu, tìm kiếm đối tượng gần nhất (nearest neighbor search) và phát hiện ngoại lệ.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

Trong PySpark, thư viện `pyspark.ml.feature` cung cấp một triển khai của LSH cho việc xử lý dữ liệu lớn. Cụ thể, PySpark cung cấp các lớp sau liên quan đến LSH:

- `BucketedRandomProjectionLSH`: Sử dụng kỹ thuật LSH với phương pháp Bucketed Random Projection. Nó chia không gian dữ liệu thành các nhóm (buckets) sử dụng các hàm băm (hash functions) ngẫu nhiên.
- `MinHashLSH`: Triển khai kỹ thuật MinHash LSH, một phương pháp LSH sử dụng các hàm MinHash để tìm các tập hợp gần nhau.

Cả hai lớp này đều cho phép bạn tìm các cặp điểm dữ liệu gần nhau hoặc các nhóm điểm gần nhau trong một tập dữ liệu lớn. Các cặp hoặc nhóm này có thể được sử dụng để thực hiện các tác vụ phân cụm, tìm kiếm đối tượng gần nhất, hoặc phát hiện ngoại lệ.

Để sử dụng LSH trong PySpark, bạn cần tạo một pipeline gồm các bước xử lý dữ liệu và bước LSH tương ứng. Sau đó, bạn có thể sử dụng pipeline này để fit và transform dữ liệu của mình.

2.5 Flask


2.5.1 Giới thiệu về Flask

Flask là loại framework web phổ biến được viết bằng trình lập ngôn ngữ Python. Công nghệ thường được sử dụng để xây dựng trang web từ những ứng dụng đơn giản đến những hệ thống phức tạp hơn.

Flask được thiết kế để hoạt động và mở rộng một cách, đồng thời nó cũng cung cấp các công cụ và thư viện cần thiết để phát triển ứng dụng web hiệu quả. Flask cũng có cộng đồng sáng tạo và hỗ trợ mạnh mẽ từ cộng đồng Python.



Hình 2: Logo Flask

	<u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKHHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
---	--	---

2.5.2 Các tính năng của Flask

Nhẹ và dễ sử dụng: Công nghệ có cấu trúc nhẹ nhàng và mã nguồn dễ đọc, giúp người phát triển dễ dàng tiếp cận và tùy chỉnh theo nhu cầu cụ thể của họ.

Định tuyến linh hoạt: Flask cung cấp cơ chế hoạt động định tuyến, cho phép người phát triển xác định các mẫu URL và phân bổ chúng cho các hàm xử lý tương ứng. Điều này giúp quản lý và xử lý yêu cầu HTTP một cách hiệu quả.

Công cụ mẫu: Flask tích hợp Jinja2, đây là một loại trình biên dịch mẫu mạnh mẽ cho phép tạo ra các giao diện người dùng.

Được mở rộng rộng rãi: Mặc dù mang đặc điểm rút gọn nhưng Flask vẫn có khả năng mở rộng mạnh mẽ thông qua việc sử dụng các tiện ích và thư viện của cộng đồng. Người dùng có thể phân tích các tính năng như xác thực, đăng nhập, điều hướng, cơ sở dữ liệu tương tác và nhiều tính năng khác.

Máy chủ phát triển tích hợp: Flask cung cấp máy chủ phát triển hợp đồng, giúp người phát triển dễ dàng kiểm tra và phát triển ứng dụng mà không cần cấu hình bổ sung.


Gửi yêu cầu RESTful: Flask hỗ trợ xây dựng API và các ứng dụng RESTful theo cách hoạt động và hiệu quả.

Cộng đồng lớn và tích cực: Số lượng người dùng Flask rất đông và luôn nhận được hỗ trợ mạnh mẽ từ cộng đồng Python, điều này giúp người phát triển tìm kiếm thông tin và tài liệu một cách dễ dàng.

2.6 Flutter

2.6.1 Giới thiệu về Flutter

Flutter là mobile UI framework của Google để tạo ra các giao diện chất lượng cao trên iOS và Android trong khoảng thời gian ngắn. Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức. Flutter hoàn toàn miễn phí và cũng là mã nguồn mở.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---



Hình 3: Logo Flutter

2.6.2 Ưu điểm

Đa nền tảng: Flutter cho phép phát triển ứng dụng cho cả iOS và Android từ một mã nguồn duy nhất. Điều này giúp giảm thời gian và chi phí phát triển.

Giao diện người dùng đẹp: Flutter sử dụng thư viện widget tùy chỉnh để tạo ra các giao diện người dùng đẹp mắt và mượt mà, giúp ứng dụng của bạn có trải nghiệm người dùng tốt.


Tốc độ phát triển: Hot reload là một tính năng mạnh mẽ của Flutter, cho phép bạn thấy ngay lập tức các thay đổi trong ứng dụng của mình mà không cần phải rebuild toàn bộ dự án.

Hiệu suất cao: Flutter sử dụng rendering engine riêng của nó (Skia) để vẽ giao diện người dùng, giúp cải thiện hiệu suất ứng dụng.

Cộng đồng phát triển mạnh mẽ: Flutter có một cộng đồng lớn và nhiệt tình, cung cấp nhiều tài liệu học tập, công cụ và gói thư viện hữu ích.

2.6.3 Nhược điểm

Kích thước ứng dụng lớn: Ứng dụng Flutter có thể có kích thước lớn hơn so với các ứng dụng được phát triển bằng các công nghệ khác, do việc đi kèm với các thư viện và tính năng không cần thiết.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

Hạn chế của Dart: Mặc dù Dart là một ngôn ngữ mạnh mẽ, nhưng do không phổ biến như các ngôn ngữ khác như JavaScript, việc tìm kiếm tài liệu và hỗ trợ có thể hơi khó khăn đôi khi.

Tính tương thích ngược: Đôi khi, các thư viện bên thứ ba không tương thích tốt với Flutter hoặc có thể không được hỗ trợ hoàn toàn.

Có thể gặp phải vấn đề tương thích thiết bị: Mặc dù Flutter cố gắng hỗ trợ nhiều thiết bị, nhưng có thể gặp phải vấn đề tương thích với một số thiết bị cũ hoặc đặc biệt.

Học công cụ mới: Nếu bạn không quen thuộc với Dart hoặc cách làm việc của Flutter, có thể mất thời gian để học và thích nghi với công nghệ này.

CHƯƠNG 3 – MÔ HÌNH NHẬN DẠNG

CNN	Parameters	Train acc	Val acc	Test acc	Train loss	Val loss	Test loss
<i>ResNet50</i>	30,339,390	0.9882	0.6840	0.6135	4.6277e-04	0.0047	0.00478
<i>Efficient Net-B7</i>	71,897,941	0.9802	0.6671	0.5926	3.1257e-04	0.0065	0.00536
<i>Xception Net</i>	27,613,158	0.9767	0.6482	0.5693	2.3394e-04	0.0072	0.00685
<i>MobileNet</i>	<u>7,522,898</u>	<u>0.9931</u>	<u>0.7979</u>	<u>0.7710</u>	<u>2.8447e-04</u>	<u>0.0038</u>	<u>0.00366</u>

Bảng 1: Bảng so sánh hiệu suất các mô hình CNN

CHƯƠNG 4 – THÍ NGHIỆM VÀ ĐÁNH GIÁ

4.1 Tập dữ liệu

Đề tài sử dụng tập dữ liệu FaceScrub từ Đại học Quốc gia Singapore (NUS) với hơn 100.000 tấm ảnh của 530 người nổi tiếng trên thế giới

Thông kê tập dữ liệu :

Tập dữ liệu	Số lượng
Train	27600
Val	6899
Test	8625

Bảng 2: Bảng thống kê tập dữ liệu

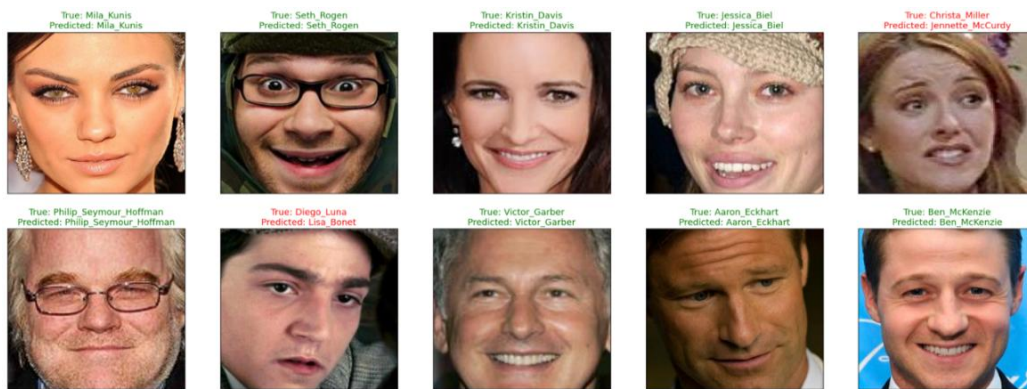
4.2 Thí nghiệm

Khi tiến hành training Neural Network thường cần hình ảnh có cùng kích thước những hình ảnh ban đầu sẽ có các kích thước khác nhau, từ 153×258 đến 819×1137 . Vì thế cần thay đổi kích thước tất cả các hình ảnh có kích thước cố định 128×128 .

Sau khi train mô hình trên ResNet50, MobileNet, XceptionNet, Efficient Net- B0, Efficient Net-B7 thu được kết quả accuracy và loss trên từng tập val, train, test.

Kết quả thu được MobileNet cho kết quả tốt nhất vì thế chọn làm feature extraction, áp dụng để chạy trên ứng dụng.

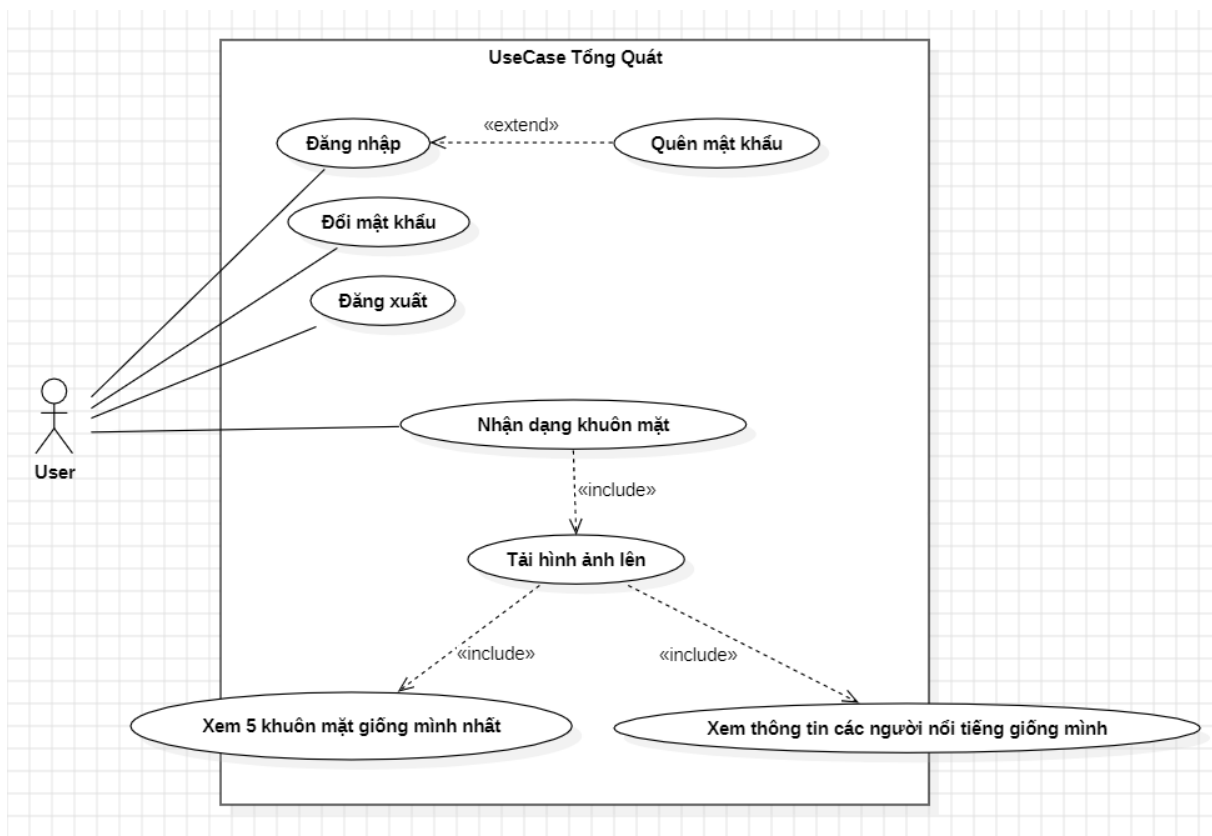
4.3 Kết quả




Hình 4: Kết quả sau khi train

CHƯƠNG 5 – HỆ THỐNG TRIỂN KHAI

5.1 Sơ đồ usecase tổng quát



Hình 5: Sơ đồ usecase tổng quát

	<u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKH-CN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
---	--	---

STT	Tác nhân	Mô tả
1	Người dùng	<ul style="list-style-type: none"> - Cung cấp và chỉnh sửa thông tin như họ tên, ngày tháng năm sinh, email, số điện thoại,... khi đăng ký. - Đăng nhập vào hệ thống - Đổi mật khẩu - Cá nhân sử dụng hệ thống để thực hiện các tác vụ như xem 5 khuôn mặt giống nhau nhất, và xem thông tin về những người nổi tiếng giống mình.


Bảng 3: Các tác nhân usecase trong hệ thống

5.2 Đặc điểm nổi bật

Đề tài này tập trung vào việc cung cấp ứng dụng cho phép người dùng xem xét khuôn mặt của mình giống với người nổi tiếng nào trên thế giới.

Ngoài ra còn có thể áp dụng trên một số dữ liệu khác như: dữ liệu hình ảnh khuôn mặt trong hệ thống thông tin quốc gia từ đó có thể truy vấn nhanh khuôn mặt của một người nào đó trong quốc gia đó dễ dàng và nhanh chóng.

Giao diện web và ứng dụng được thiết kế đơn giản và thân thiện với người dùng, giúp mọi người dễ dàng tương tác và sử dụng. Người dùng có thể chụp và tải lên hình ảnh của mình, nhận được gợi ý nhanh chóng.


	<u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKH-CN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
--	--	---

CHƯƠNG 6 - ỨNG DỤNG DI ĐỘNG

6.1 Chức năng

6.1.1 Đăng nhập

Mã use case	UC01	
Use Case	Đăng nhập	
Ngữ cảnh	Truy cập vào ứng dụng khi đã đăng ký tài khoản thành công hoặc có tài khoản trong hệ thống cơ sở dữ liệu.	
Mô tả	Tác nhân đăng nhập vào hệ thống để có thể sử dụng được các chức năng của ứng dụng.	
Tác nhân	Người dùng	
Sự kiện kích hoạt	Đăng nhập vào hệ thống.	
Điều kiện tiên quyết	Người dùng có tài khoản trên hệ thống và truy cập theo quyền được cấp theo vai trò của tài khoản.	
Kết quả	Đăng nhập thành công	
Lưu đồ sự kiện	Actor	System
	1. Người dùng nhấn chọn mục đăng nhập trên hệ thống.	1.1. Hệ thống hiển thị giao diện đăng nhập để người dùng điền thông tin username (số điện

	<u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
--	--	--

		thoại hoặc email) và password
	2. Người dùng điền thông tin tài khoản của mình username (số điện thoại hoặc email) và password.	
	3. Người dùng chọn nút “Đăng nhập”.	3.1. Hệ thống kiểm tra thông tin, ghi nhận hoạt động 3.2. Cho phép truy cập hệ thống.
Ngoại lệ	<ul style="list-style-type: none"> - Người dùng nhập sai username hoặc password => hệ thống không cho truy cập và báo lỗi. - Người dùng hủy ý định đăng nhập, tắt giao diện đăng nhập. 	

Bảng 4: Đặc tả usecase đăng nhập

6.1.2 Đăng xuất

Mã use case	UC02
Use Case	Đăng xuất
Ngữ cảnh	Khi đã truy cập thành công vào hệ thống.
Mô tả	Đăng xuất tài khoản ra khỏi hệ thống, mất quyền chức năng sử dụng hệ thống.
Tác nhân	Người dùng


Sự kiện kích hoạt	Đăng xuất khỏi hệ thống.	
Điều kiện tiên quyết	Người dùng đã và đang truy cập hệ thống.	
Kết quả	Đăng xuất thành công	
Luồng sự kiện	Actor	System
	1. Người dùng nhấn chọn mục “đăng xuất”.	1.1. Hệ thống thoát khỏi tài khoản người dùng.
Ngoại lệ	Không.	

Bảng 5: Đặc tả usecase đăng xuất

6.1.3 Đổi mật khẩu

Mã use case	UC03
Use Case	Đổi mật khẩu
Ngữ cảnh	Khi người dùng đã đăng nhập vào hệ thống và muốn thay đổi mật khẩu.
Mô tả	Người dùng khi quên mật khẩu hoặc muốn thay đổi mật khẩu mới.
Tác nhân	Người dùng ứng dụng
Sự kiện kích hoạt	Thay đổi mật khẩu.
Điều kiện tiên quyết	Người dùng đã có tài khoản truy cập hệ thống trước đó.

Kết quả	Thay đổi mật khẩu thành công.	
Luồng sự kiện	Actor	System
	1. Người dùng chọn mục “Thay đổi mật khẩu” hoặc “Quên mật khẩu”.	1.1. Hệ thống hiển thị thị form thay đổi mật khẩu hoặc form lấy lại mật khẩu.
	2. Người dùng điền thông tin.	
	3. Người dùng chọn “Xác nhận”.	3.1. Hệ thống kiểm tra thông tin tài khoản. 3.1.1. Nếu thay đổi mật khẩu khi đăng nhập thì xác nhận mật khẩu cũ. 3.1.2. Nếu quên mật khẩu cần xác minh người dùng bằng mã qua số điện thoại hoặc email.

 <p>ĐẠI HỌC TÔN ĐỨC THẮNG TON DUC THANG UNIVERSITY</p>	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	--

		3.2. Hệ thống cập nhật mật khẩu thành công.
Ngoại lệ	<ul style="list-style-type: none"> - Mật khẩu mới trùng mật khẩu cũ. - Mật khẩu không đạt yêu cầu tiêu chuẩn. - Nhập mật khẩu cũ không chính xác. - Mật khẩu mới và phân nhập lại mật khẩu mới không trùng nhau. - Mã xác minh email, số điện thoại không chính xác. 	

Bảng 6: Đặc tả usecase đổi mật khẩu

6.1.4 Nhận dạng khuôn mặt


Mã use case	UC04
Use Case	Nhận dạng khuôn mặt
Ngữ cảnh	Truy cập vào ứng dụng và sử dụng chức năng nhận dạng khuôn mặt sau khi đã đăng nhập thành công.
Mô tả	Tác nhân tải hình ảnh lên hệ thống để nhận dạng khuôn mặt và tìm ra những khuôn mặt tương tự.
Tác nhân	Người dùng

Sự kiện kích hoạt	Tải hình ảnh khuôn mặt lên hệ thống	
Điều kiện tiên quyết	Người dùng đã đăng nhập vào hệ thống..	
Kết quả	Hệ thống trả về kết quả nhận dạng khuôn mặt.	
Luồng sự kiện	Actor	System
	1. Người dùng chọn chức năng "Nhận dạng khuôn mặt".	1.1. Hệ thống hiển thị giao diện tải hình ảnh lên.
	2. Người dùng tải hình ảnh khuôn mặt lên.	2.1. Hệ thống nhận hình ảnh và bắt đầu quá trình nhận dạng.
	3. Người dùng chọn nút “Nhận dạng”.	3.1. Hệ thống xử lý hình ảnh và so sánh với cơ sở dữ liệu. 3.2. Hệ thống trả về kết quả nhận dạng khuôn mặt.
Ngoại lệ	Hệ thống không nhận dạng được khuôn mặt trong hình ảnh tải lên.	


Bảng 7: Đặc tả usecase nhận dạng khuôn mặt

6.1.5 Xem 5 khuôn mặt người nổi tiếng giống mình nhất

Mã use case	UC05
Use Case	Xem 5 khuôn mặt người nổi tiếng giống mình nhất
Ngữ cảnh	Sau khi sử dụng chức năng nhận dạng khuôn mặt, người dùng muốn xem danh sách 5 khuôn mặt người nổi tiếng giống mình nhất.

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

Mô tả	Hệ thống hiển thị danh sách 5 khuôn mặt người nổi tiếng giống khuôn mặt của người dùng nhất.
Tác nhân	Người dùng


	Thủ tục: NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
---	---	--

Sự kiện kích hoạt	Kết quả nhận dạng khuôn mặt thành công.	
Điều kiện tiên quyết	Người dùng đã sử dụng chức năng nhận dạng khuôn mặt.	
Kết quả	Hệ thống hiển thị danh sách 5 khuôn mặt người nổi tiếng giống người dùng nhất.	
Luồng sự kiện	Actor	System
	1. Người dùng chọn "Xem 5 khuôn mặt người nổi tiếng giống mình nhất".	1.1. Hệ thống hiển thị danh sách 5 khuôn mặt người nổi tiếng giống người dùng nhất.
Ngoại lệ	Không có kết quả nào phù hợp.	

Bảng 8: Đặc tả usecase xem 5 khuôn mặt người nổi tiếng giống mình nhất


6.1.6 Xem thông tin những người nổi tiếng giống mình

Mã use case	UC06
Use Case	Xem thông tin những người nổi tiếng giống mình nhất
Ngữ cảnh	Sau khi xem danh sách những khuôn mặt giống mình, người dùng muốn xem thông tin chi tiết về những người nổi tiếng này.
Mô tả	Hệ thống cung cấp thông tin hình ảnh về những người nổi tiếng có khuôn mặt giống người dùng.
Tác nhân	Người dùng

	<u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN	Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:
---	--	--

Sự kiện kích hoạt	Người dùng chọn xem thông tin chi tiết của một khuôn mặt người nổi tiếng từ danh sách.	
Điều kiện tiên quyết	Người dùng đã xem danh sách những khuôn mặt người nổi tiếng giống mình.	
Kết quả	Hệ thống hiển thị thông tin hình ảnh về người nổi tiếng.	
Luồng sự kiện	Actor	System
	1. Người dùng chọn một khuôn mặt người nổi tiếng từ danh sách.	1.1. Hệ thống hiển thị thông tin chi tiết về người nổi tiếng đó.
Ngoại lệ	Không có kết quả nào phù hợp.	


Bảng 9: Đặc tả usecase xem thông tin những người nổi tiếng giống mình

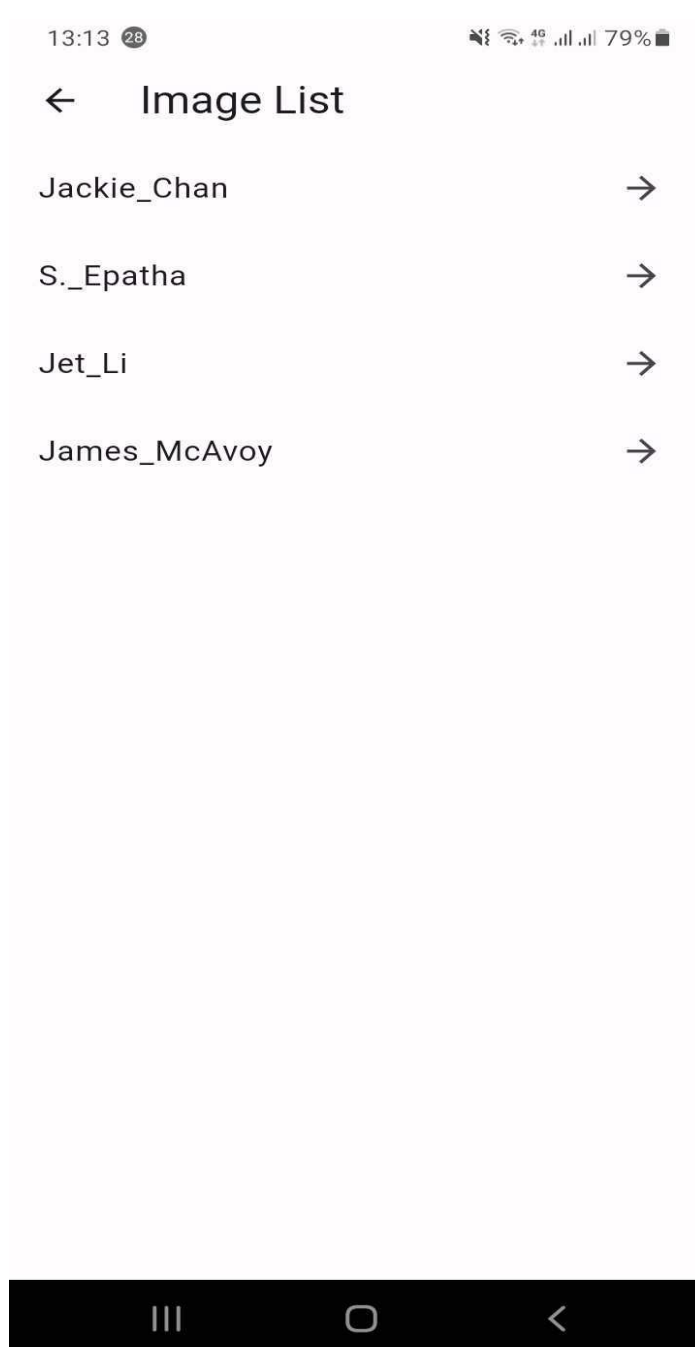
 <p>ĐẠI HỌC TÔN ĐỨC THẮNG TON DUC THANG UNIVERSITY</p>	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	---	---

6.2 Giao diện ứng dụng




Hình 6: Giao diện chụp ảnh

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---



Hình 7: Giao diện danh sách 5 người giống nhất

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---


13:14 28

79%

← Image Detail



Hình 8: Giao diện hình ảnh người giống nhất

	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
--	--	---

CHƯƠNG 7 – BACKEND SEVER

7.1 Database

7.2 Recognition Service

Khi một ảnh được gửi đến server, service Python sẽ thực hiện việc nhận dạng ảnh đó bằng cách sử dụng Flask và các thư viện như OpenCV và TensorFlow.

Khi Flask nhận được yêu cầu nhận dạng ảnh, nó sẽ sử dụng thư viện OpenCV để tiền xử lý ảnh. Tiền xử lý ảnh bao gồm việc xử lý định dạng ảnh ở đây OpenCV sẽ nhận diện khuôn mặt từ ảnh được chụp hoặc được tải lên và cắt khuôn mặt từ một tấm ảnh lớn sau đó hay đổi kích thước và chuyển đổi ảnh sang định dạng phù hợp cho việc nhận dạng. Sau khi tiền xử lý ảnh hoàn tất, ảnh được đưa vào model TensorFlow để thực hiện việc nhận dạng.

Model TensorFlow được huấn luyện trước để có thể nhận dạng các đối tượng được xác định trước. Trong trường hợp này, model TensorFlow được huấn luyện để nhận dạng các người nổi tiếng trên thế giới.

Khi model TensorFlow nhận dạng được đối tượng trên ảnh, kết quả nhận dạng sẽ được trả về cho LSH. LSH sẽ sử dụng kết quả nhận dạng vào cơ sở dữ liệu và trả về kết quả này cho client.

Một lợi ích của việc sử dụng Flask để xử lý các yêu cầu nhận dạng là hệ thống của nhóm có thể mở rộng dễ dàng. Có thể tăng hiệu suất của hệ thống bằng cách thêm nhiều service Python chạy song song và phân phối việc xử lý ảnh giữa các service Python.

Để đảm bảo an toàn và bảo mật cho hệ thống đã thiết lập một số biện pháp bảo vệ như xác thực người dùng, mã hóa dữ liệu và chống tấn công từ chối dịch vụ (DDoS).

Ngoài ra kiểm tra tính ổn định và hiệu suất của hệ thống bằng cách thực hiện các bài kiểm tra và thử nghiệm. Kết quả cho thấy hệ thống có thể xử lý hàng nghìn yêu cầu nhận dạng một cách nhanh chóng và đáng tin cậy.

CHƯƠNG 8 – TỔNG KẾT

8.1 Kết quả

Đề tài "Truy vấn nhanh khuôn mặt với LSH trên ứng dụng Flutter và Flask backend" có thể mang lại kết quả tốt trong việc truy vấn nhanh khuôn mặt trong một tập dữ liệu siêu lớn, tiết kiệm thời gian và tài nguyên, mở rộng khả năng nhận diện.

Chẩn đoán đáng tin cậy và chính xác: Mô hình MobileNet đã được huấn luyện trên một tập dữ liệu lớn và đa dạng về khuôn mặt của hơn 530 người trên thế giới. Khi được áp dụng vào ứng dụng truy vấn, mô hình có khả năng phân loại hình ảnh và đưa ra kết quả nhận diện ban đầu với độ chính xác cao.

Tiết kiệm thời gian và tài nguyên: Việc sử dụng mô hình MobileNet trong ứng dụng di động cho phép người dùng chụp ảnh khuôn mặt hoặc tải ảnh và nhận được kết quả ban đầu ngay lập tức. Điều này giúp tiết kiệm thời gian và tài nguyên so với việc tìm kiếm một cách thủ công.

8.2 Ý nghĩa

Đề tài "Truy vấn nhanh khuôn mặt với LSH trên ứng Flutter và Flask backend" mang lại nhiều ý nghĩa quan trọng và tiềm năng trong lĩnh vực nhận diện khuôn mặt.

Ý nghĩa khoa học:

- Truy vấn nhanh khuôn mặt: Giải quyết vấn đề tìm kiếm nhanh thông tin về người nổi tiếng bằng thuật toán LSH.
- Đánh giá tương đồng khuôn mặt: So sánh đặc điểm khuôn mặt giữa người dùng và các nhân vật nổi tiếng.

Ý nghĩa thực tiễn

- Tìm kiếm nhanh thông tin về người nổi tiếng: Giải quyết vấn đề truy vấn nhanh khuôn mặt, giúp người dùng tìm kiếm thông tin về người nổi tiếng một cách nhanh chóng và chính xác.
- Ứng dụng trong giám sát và quản lý khuôn mặt: Áp dụng vào hệ thống giám sát và quản lý dữ liệu khuôn mặt trong cơ sở dữ liệu dân cư.

8.3 Thuận lợi

Mô hình MobileNet được xây dựng dựa trên công nghệ trí tuệ nhân tạo và học máy. Điều này mang lại khả năng phân loại và nhận diện chính xác cao, giúp cung cấp những kết quả đáng tin cậy cho người dùng.


Mô hình MobileNet có thể được cải thiện và mở rộng để đáp ứng các yêu cầu và trường hợp nhận diện đa dạng. Chúng ta có thể cung cấp dữ liệu mới và huấn luyện mô hình để nâng cao khả năng nhận diện và tương tác với ứng dụng.

Chương trình với cấu trúc kết hợp hai hệ thống chạy song song giúp tăng khả năng mở rộng và xử lý được yêu cầu của nhiều người dùng đồng thời.

8.4 Khó khăn

Trong quá trình thực hiện đề tài "Truy vấn nhanh khuôn mặt với LSH trên ứng Flutter và Flask backend", có một số khó khăn phát sinh:

- *Thu thập dữ liệu:* Việc xây dựng một tập dữ liệu đủ lớn và đa dạng để huấn luyện mô hình truy vấn nhanh khuôn mặt có thể là vấn đề lớn vì đây là dữ liệu cá nhân. Thu thập hình ảnh từ các nguồn đáng tin cậy và đảm bảo tính đa dạng của ảnh của mỗi người là rất quan trọng để đảm bảo mô hình có khả năng nhận diện đúng và chính xác.
- *Đánh giá độ chính xác:* Để đảm bảo hiệu suất chẩn đoán tốt của mô hình, cần tiến hành đánh giá độ chính xác và độ tin cậy của nó. Tuy nhiên, đánh giá này đòi hỏi sự so sánh với các kết quả nhận diện của chuyên gia, và việc thu thập dữ liệu chẩn đoán từ các chuyên gia có thể gặp khó khăn và tốn kém về thời gian và nguồn lực.
- *Sự đa dạng và phức tạp của các khuôn mặt của mỗi người:* Bệnh ngoại da có rất nhiều loại và có thể có những biến thể khác nhau. Khuôn mặt có rất nhiều loại và có nhiều trường hợp đặc biệt (các khuôn mặt tương đối giống nhau). Việc xây dựng một mô hình nhận diện chung cho tất cả mọi người có thể gặp khó khăn. Điều này yêu cầu nỗ lực lớn để đảm bảo mô hình có khả năng nhận diện và phân loại đúng khuôn mặt của mỗi người trong dữ liệu.
- *Giới hạn của mô hình di động:* Mô hình MobileNet được thiết kế để đạt được cân bằng giữa hiệu suất và tài nguyên tính toán. Tuy nhiên, đây là


	<p><u>Thủ tục:</u> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---

một mô hình có độ phức tạp thấp hơn so với các kiến trúc mạng nơ-ron sâu khác. Điều này có thể đặt ra giới hạn về khả năng nhận diện chính xác và khả năng phân loại của mô hình.

- *Hệ thống*: chương trình chạy song song hai Framework nên khá phức tạp và phải xử lý vấn đề cấu hình hệ thống.

8.5 Phát triển

Trong tương lai ứng dụng sẽ phát triển nhiều hơn cũng như thu thập nhiều data hình ảnh của nhiều người, đồng thời thêm nhiều lớp để hỗ trợ việc nhận diện tốt hơn. Bên cạnh đó, việc cải thiện tốc độ xử lý cũng rất quan trọng, nhằm giúp ứng dụng nhận diện 1 cách chính xác và nhanh chóng hơn. Ngoài ra còn có một số tính năng khác như: hỗ trợ trên nền tảng web, cải thiện giao diện ,...Giao diện sẽ được triển khai hoàn thiện và đẹp hơn nhằm đảm bảo vấn đề trải nghiệm người dùng, cũng như quản lý và phát triển hệ thống.

	<p><i>Thủ tục:</i> NGHIÊN CỨU KHOA HỌC SINH VIÊN</p>	<p>Mã số: TT/P.QLPTKHCN/13/BM18 Ban hành lần: 04 Ngày hiệu lực:</p>
---	--	---

TÀI LIỆU THAM KHẢO

- [1] Rama Chellappa, Pawan Sinha, and P Jonathon Phillips. Face recognition by computers and humans. Computer, 2010.
- [2] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In Computer Vision. IEEE, 2009.
- [3] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. Image Processing, 2010.
- [4] Greeshma, P.; Palguna Rao, K. Efficient Retrieval of Face Image from Large Scale Database Using Sparse Coding and Reranking. Int. J. Sci. Res. 2014, 3, 275–277.
- [5] Nithya, K.; Rajamani, V. Triplet Label Based Image Retrieval Using Deep Learning in Large Database. Comput. Syst. Sci. Eng. 2023, 44, 2655–2666. [[CrossRef](#)]
- [6] Taherkhani, F.; Talreja, V.; Valenti, M.C.; Nasrabadi, N.M. Error-corrected margin-based deep cross-modal hashing for facial image retrieval. IEEE Trans. Biometrics Behav. Identity Sci. 2020, 2, 279–293. [[CrossRef](#)]
- [7] Khanam, Z.; Ahsan, M.N. Implementation of the pHash algorithm for face recognition in a secured remote online examination system. Int. J. Adv. Sci. Res. Eng. 2018, 4, 1–5. [[CrossRef](#)]