ỦY BAN NHÂN DÂN

THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**BÁO CÁO TỔNG KẾT**

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

**GVHD: Đỗ Như Tài**

**Face Recognition Using**

**Convolutional Neural Networks**

Chủ nhiệm đề tài: Lê Quang Vinh

Thành viên tham gia

1.Lê Quang Vinh

2.Vũ Thị Thanh Vân

**TP. HỒ CHÍ MINH, năm 2025****LỜI CẢM ƠN**

Để hoàn thành dự án nghiên cứu khoa học “Nghiên cứu và so sánh cơ chế xử lý dữ liệu phân tán – song song và minh họa,” chúng em đã nhận được rất nhiều sự giúp đỡ và hỗ trợ tận tình. Chúng em xin trân trọng gửi lời cảm ơn sâu sắc đến:

● Khoa Công Nghệ Thông Tin – Trường Đại Học Sài Gòn đã tạo mọi điều kiện thuận lợi để chúng em có thể thực hiện nghiên cứu này.

● Chúng em xin gửi lời tri ân đến thầy Đỗ Như Tài đã tận tình hướng dẫn, chỉ bảo trong suốt quá trình thực hiện đề tài. Sự định hướng và hỗ trợ của thầy đã giúp chúng em hoàn thành bài nghiên cứu một cách thuận lợi và hiệu quả.

● Các thành viên trong nhóm đã luôn đoàn kết, hỗ trợ lẫn nhau và nỗ lực hết mình để hoàn thành dự án với kết quả tốt nhất.

Cuối cùng, chúng em xin kính chúc các thầy cô luôn mạnh khỏe, thành công để tiếp tục dìu dắt các thế hệ học sinh, sinh viên trên con đường học tập và nghiên cứu.

**MỤC LỤC**

|  |  |
| --- | --- |
|  | **Trang** |
| **GIỚI THIỆU** | **1** |
| **TÓM TẮT** | **2** |
| **Chương 1: Tổng quan vấn đề** | **3** |
| 1. **Lý do chọn đề tài** | **3** |
| 1. **Vấn đề nghiên cứu** | **3** |
| 1. **Mục tiêu nghiên cứu** | **3** |
| 1. **Câu hỏi nghiên cứu** | **4** |
| 1. **Phạm vi nghiên cứu** | **4** |
| **Chương 2: Lược khảo tài liệu** | **4** |
| **2.1. Cơ sở lý thuyết** | **5** |
| **2.1.1. Mạng Nơ-ron Tích Chập (CNN)** | **5** |
| **2.1.2. Kiến trúc Inception V3** | **5** |
| **2.1.2.1. Giới thiệu module Inception** | **5** |
| **2.1.2.2. Node Inception V3 mở rộng** | **6** |
| **2.1.3. Quá trình xử lý và huấn luyện mô hình** | **9** |
| **2.1.3.1. Input và Tiền xử lý** | **9** |
| **2.1.3.2. Mô hình Trích Xuất Đặc Trưng** | **9** |
| **2.1.3.3. Lớp Phân Loại Mới** | **9** |
| **2.1.3.4. Huấn luyện** | **9** |
| **2.1.3.5. Lưu Mô hình** | **10** |
| **2.1.4. Luồng dữ liệu** | **10** |
| **2.1.5. Transfer Learning** | **12** |
| **2.1.5.1. Khái Niệm Transfer Learning** | **12** |
| **2.1.5.2. Tiền xử lý dữ liệu** | **13** |
| **2.1.5.3. Phân chia Dữ liệu** | **13** |
| **2.2. Điểm mạnh, điểm yếu của các nghiên cứu trước và cách nghiên cứu kế thừa/phát triển** | **13** |

|  |  |
| --- | --- |
| **2.3. Cách nghiên cứu kế thừa và phát triển** | **14** |
| **CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU** | **14** |
| **3.1. Đối tượng và mẫu nghiên cứu** | **14** |
| **3.2. Cách thu thập dữ liệu** | **15** |
| **3.3. Phân tích dữ liệu** | **15** |
| **CHƯƠNG 4: THỰC NGHIỆM VÀ THẢO LUẬN** | **16** |
| **4.1. Yêu cầu Thực nghiệm** | **16** |
| **4.1.1. Mục tiêu** | **16** |
| **4.1.2. Phương pháp** | **16** |
| **4.2. Tập Dữ liệu** | **16** |
| **4.3. Ngôn ngữ Chính** | **17** |
| **4.4. Các Thư viện** | **17** |
| **4.5. Thực nghiệm** | **17** |
| **4.5.1. Xác định lớp khởi đầu CNN để đào tạo** | **27** |
| **4.5.2. Trực quan hóa TensorBoard** | **30** |
| **4.5.3. Huấn luyện mô hình InceptionV3** | **33** |
| **4.5.4. Đánh giá hiệu suất** | **34** |
| **4.5.5. Tăng cường dữ liệu** | **34** |
| **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** | **35** |
| **TÀI LIỆU THAM KHẢO (References)** | **37** |

**BẢNG HÌNH ẢNH**

|  |  |
| --- | --- |
|  | **Trang** |
| *Hình 1: Sơ đồ luồng dữ liệu minh hoạ cách dữ liệu được xử lý.* | **6** |
| *Hình 2. Sơ đồ luồng dữ liệu của mô hình InceptionV3* | **7** |
| *Hình 3. Sơ đồ Luồng Dữ Liệu minh hoạ quá trình dữ liệu được xử lý.* | **11** |
| *Hình 4. Một số thư viện cần cài đặt.* | **18** |
| *Hình 5. Khởi tạo mô hình inception V3.* | **18** |
| *Hình 6. Khởi tạo mô hình không bao gồm lớp phân loại cuối cùng.* | **19** |
| *Hình 7. Tải và giải nén bộ dữ liệu LFW.* | **19** |
| *Hình 8. Thư mục tạo mới chứa 10 người có nhiều ảnh nhất.* | **21** |
| *Hình 9. Biểu đồ cột thể hiện tỷ lệ chiếm của mười cá nhân có nhiều ảnh nhất.* | **22** |
| *Hình 10. Ví dụ về các ảnh khuôn mặt từ lớp nhãn 0 trong tập dữ liệu LFW.* | **23** |
| *Hình 11. Độ chính xác so với độ phức tạp tính toán của các kiến trúc CNN phổ biến trên tập ImageNet.* | **27** |
| *Hình 12. Biểu đồ đường thể hiện phát triển của mô hình qua các bước huấn luyện.* | **31** |
| *Hình 13. Biểu đồ đường phản ánh sự tăng trưởng của độ chính xác kiểm định theo thời gian.* | **32** |

**GIỚI THIỆU**

Trong thế giới hiện đại, khoa học và công nghệ đã trở thành nền tảng vững chắc thúc đẩy sự phát triển của nhân loại. Để hiểu rõ hơn về các tiến bộ khoa học và khám phá mới, việc nghiên cứu và phân tích các bài báo khoa học đóng vai trò vô cùng quan trọng. Quá trình tìm kiếm và đánh giá các bài báo nghiên cứu khoa học không chỉ giúp chúng ta tiếp cận những tri thức mới, mà còn mở rộng khả năng tư duy phản biện và phân tích thông tin. Bài tiểu luận này sẽ hướng dẫn các phương pháp và kỹ thuật hiệu quả để tìm kiếm, phân loại và đánh giá các bài báo nghiên cứu khoa học, từ đó xây dựng nền tảng kiến thức vững chắc cho các nghiên cứu tương lai.

**TÓM TẮT**

Nhận diện khuôn mặt là một trong những ứng dụng quan trọng của trí tuệ nhân tạo và thị giác máy tính trong các lĩnh vực như an ninh, giám sát và xác thực danh tính. Nghiên cứu này được thực hiện nhằm đánh giá hiệu quả của mô hình học sâu trong việc nhận diện khuôn mặt người thông qua bộ dữ liệu Labeled Faces in the Wild (LFW) – một tập dữ liệu tiêu chuẩn chứa hơn 13.000 hình ảnh khuôn mặt của 5.749 cá nhân với sự đa dạng về biểu cảm, ánh sáng và góc nhìn. Phương pháp nghiên cứu bao gồm tiền xử lý ảnh (chuẩn hóa, cắt, điều chỉnh kích thước), trích xuất đặc trưng khuôn mặt bằng mô hình InceptionV3 của TensorFlow và huấn luyện mô hình nhận diện. Tinh chỉnh hình ảnh được thực hiện bằng thư viện OpenCV để phát hiện khuôn mặt, căn giữa và chuẩn hóa vùng chứa khuôn mặt nhằm giảm nhiễu và cải thiện chất lượng đặc trưng đầu vào. Kết quả ban đầu cho thấy mô hình có khả năng nhận diện chính xác các cá nhân có số lượng ảnh lớn nhất. Kết luận, mô hình InceptionV3 kết hợp với tiền xử lý bằng OpenCV có tiềm năng trong các ứng dụng thực tế, nhưng cần tiếp tục tối ưu hóa dữ liệu và kiến trúc để nâng cao hiệu quả tổng quát hóa.

**CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ**

1. **Lý do chọn đề tài**

Nhận diện khuôn mặt là một bài toán phức tạp do sự đa dạng về biểu cảm, góc nhìn, điều kiện ánh sáng và môi trường thực tế. Để đánh giá hiệu quả của các mô hình nhận diện khuôn mặt trong những điều kiện đa dạng này, LFW (Labeled Faces in the Wild) được coi là một trong những bộ dữ liệu chuẩn và phổ biến nhất. Việc nghiên cứu và phát triển mô hình trên LFW giúp kiểm tra khả năng tổng quát hóa và độ chính xác của mô hình khi áp dụng vào thực tiễn. Nhờ đó, các ứng dụng trong lĩnh vực an ninh, quản lý và nhiều lĩnh vực khác có thể được cải thiện, mang lại hiệu quả cao hơn trong việc xác thực và nhận diện khuôn mặt.

1. **Vấn đề nghiên cứu**

Làm thế nào để xây dựng mô hình nhận diện khuôn mặt có độ chính xác cao trên bộ dữ liệu LFW, vốn chứa nhiều biến thể thực tế như ánh sáng, góc nhìn và biểu cảm khác nhau?

Mô hình CNN có thể trích xuất đặc trưng hiệu quả từ ảnh LFW như thế nào để phân biệt chính xác các cá nhân khác nhau?

1. **Mục tiêu nghiên cứu**

Phát triển và đánh giá mô hình nhận diện khuôn mặt dựa trên CNN trên bộ dữ liệu LFW.

Cải thiện độ chính xác nhận diện khuôn mặt trong các điều kiện đa dạng được thể hiện trong LFW.

Tối ưu hóa mô hình để phù hợp với đặc điểm và kích thước ảnh trong LFW.

1. **Câu hỏi nghiên cứu**

Mô hình CNN có thể xử lý tốt các biến thể trong bộ dữ liệu LFW để nhận diện chính xác khuôn mặt không?

Các kỹ thuật tiền xử lý và tăng cường dữ liệu nào giúp cải thiện hiệu quả nhận diện trên LFW?

Mức độ ảnh hưởng của các kiến trúc CNN khác nhau đến kết quả nhận diện trên LFW là gì?

1. **Phạm vi nghiên cứu**

Đối tượng: Ảnh khuôn mặt trong bộ dữ liệu Labeled Faces in the Wild (LFW).

Không gian: Bộ dữ liệu LFW chứa ảnh khuôn mặt thu thập từ Internet với nhiều điều kiện thực tế khác nhau.

Thời gian: Nghiên cứu áp dụng trên dữ liệu LFW hiện có (được phát hành từ năm 2007 và được sử dụng rộng rãi đến nay).

Phạm vi kỹ thuật: Tập trung vào xây dựng và đánh giá mô hình CNN cho nhận diện khuôn mặt trên LFW, bao gồm các bước tiền xử lý, huấn luyện và đánh giá.

**CHƯƠNG 2: LƯỢC KHẢO TÀI LIỆU**

**Dựa vào bài báo khoa học “Facial Recognition Using Google’s Convolutional Neural Network” của William Koehrsen trên Medium**

Bài báo tổng hợp các nghiên cứu về việc sử dụng mạng nơ-ron tích chập (CNN), đặc biệt là kiến trúc Inception-v3 của Google, trong các tác vụ nhận diện hình ảnh và nhận diện khuôn mặt. Inception-v3 đã đạt được độ chính xác cao trên bộ dữ liệu ImageNet với 1.2 triệu ảnh thuộc 1000 lớp khác nhau, nhờ kiến trúc sâu và các module inception đặc trưng. Tuy nhiên, mô hình này chỉ được huấn luyện để phân loại các lớp trong ImageNet, không bao gồm khuôn mặt người.

Các nghiên cứu trước đây cho thấy các lớp thấp trong CNN học được các đặc trưng cơ bản như cạnh, màu sắc, trong khi các lớp cao hơn học các đặc trưng phức tạp hơn đặc thù cho từng lớp. Vì vậy, việc tận dụng mô hình đã huấn luyện sẵn (transfer learning) và chỉ huấn luyện lại các lớp trên cùng cho tác vụ mới như nhận diện khuôn mặt là một hướng tiếp cận hiệu quả, tiết kiệm tài nguyên tính toán.

Bài báo cũng tham khảo dataset Labeled Faces in the Wild (LFW) – một bộ dữ liệu chuẩn dùng trong nhận diện khuôn mặt với hơn 13,000 ảnh của 5,749 cá nhân, có nhiều biến thể về góc nhìn, ánh sáng và biểu cảm. LFW được sử dụng rộng rãi trong các nghiên cứu nhận diện khuôn mặt thực tế.

**2.1. Cơ sở lý thuyết**

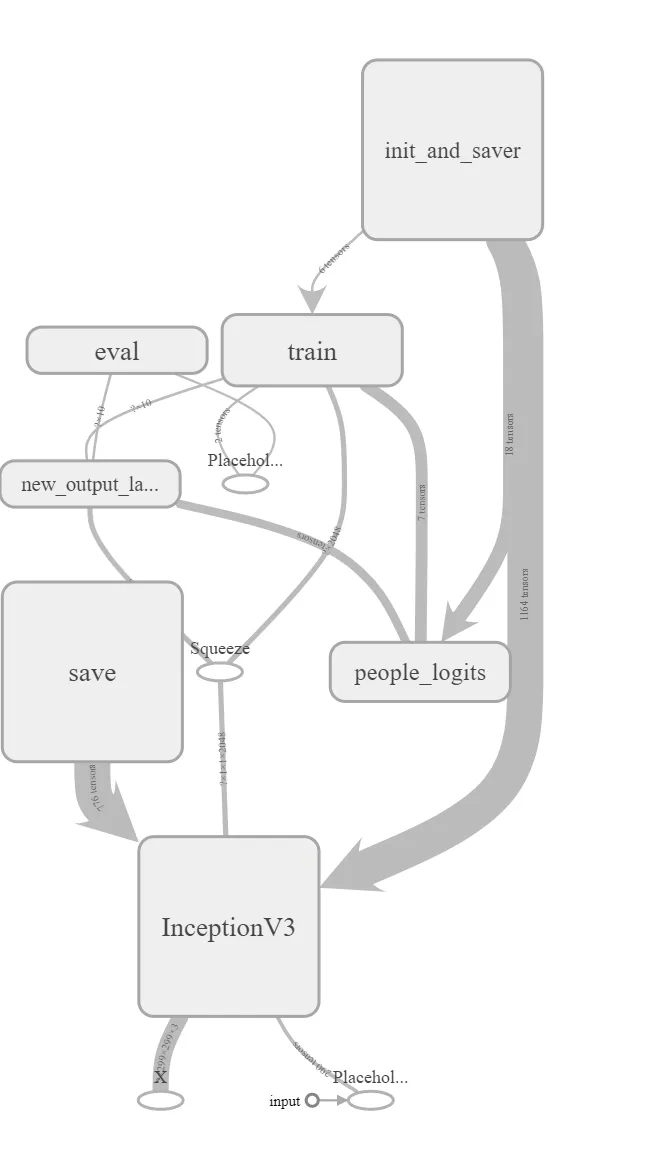
**2.1.1. Mạng Nơ-ron Tích Chập (CNN)**

CNN là mô hình học sâu hiệu quả trong nhận dạng hình ảnh, gồm nhiều lớp convolutional trích xuất đặc trưng từ ảnh, tiếp theo là các lớp fully connected để phân loại.

**2.1.2. Kiến trúc Inception V3**

**2.1.2.1. Giới thiệu module Inception**

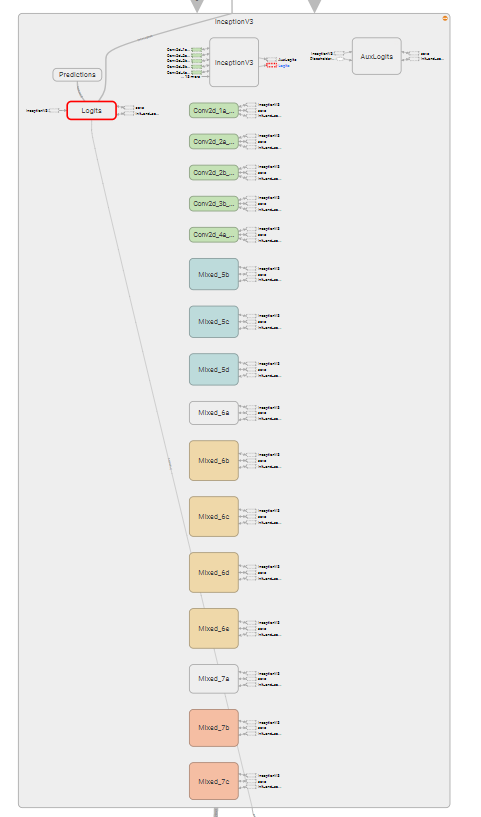
Sử dụng các module inception cho phép mạng học đa quy mô đặc trưng song song, giúp tăng hiệu quả và độ sâu mà không tăng quá nhiều tham số.



*Hình 1: Sơ đồ luồng dữ liệu minh hoạ cách dữ liệu được xử lý.*

Hình ảnh minh họa cách dữ liệu được xử lý trong mô hình nhận diện khuôn mặt dựa trên InceptionV3. Các thành phần chính bao gồm trích xuất đặc trưng, phân loại khuôn mặt, huấn luyện và đánh giá mô hình.

**2.1.2.2. Node Inception V3 mở rộng**



*Hình 2. Sơ đồ luồng dữ liệu của mô hình InceptionV3*

**Luồng dữ liệu chính**

Ảnh đầu vào được đưa vào mô hình InceptionV3.

Dữ liệu lần lượt đi qua nhiều lớp tích chập (Convolutional) và các khối Inception (Mixed), thể hiện qua các node như:

Conv2d\_1a\_3x3, Conv2d\_2a\_3x3, Conv2d\_2b\_3x3, Conv2d\_3b\_1x1, Conv2d\_4a\_3x3: Các lớp convolution đầu tiên giúp trích xuất đặc trưng cơ bản của ảnh.

Mixed\_5b, Mixed\_5c, Mixed\_5d, Mixed\_6a, Mixed\_6b, Mixed\_6c, Mixed\_6d, Mixed\_6e, Mixed\_7a, Mixed\_7b, Mixed\_7c: Các khối Inception, là tổ hợp nhiều lớp convolution với kích thước kernel khác nhau, giúp mô hình học được đặc trưng đa chiều và phức tạp hơn.

**Các lớp đầu ra**

**Logits**: Sau khi đi qua toàn bộ các lớp convolution và khối Inception, đầu ra được gom lại và đưa vào lớp "Logits". Đây là lớp tuyến tính cuối cùng, tạo ra một vector điểm số (logits) cho từng lớp (tức là từng đối tượng hoặc cá nhân cần nhận diện).

**Predictions**: Từ logits, mô hình tính toán xác suất dự đoán cuối cùng cho mỗi lớp thông qua hàm softmax, cho ra kết quả phân loại.

**Aux\_Logits:** Một nhánh phụ (auxiliary logits) được sử dụng để hỗ trợ quá trình huấn luyện, giúp giảm hiện tượng mất mát gradient ở các lớp sâu.

**Ý nghĩa các khối màu**

Các khối màu xanh lá (Conv2d) là các lớp convolution đầu tiên.

Các khối màu xanh dương, vàng, cam (Mixed) là các khối Inception ở các tầng khác nhau, mỗi khối này thực hiện nhiều phép tích chập song song với các kích thước kernel khác nhau.

Các khối màu đỏ (Logits) và xám (Predictions, Aux\_Logits) là các lớp đầu ra chính và phụ.

**2.1.3. Quá trình xử lý và huấn luyện mô hình**

**2.1.3.1. Input và Tiền xử lý**

**Input**: Dữ liệu đầu vào là ảnh khuôn mặt, được nạp vào hệ thống qua một node "input" (Placeholder).

Ảnh này sau đó được chuyển vào mô hình InceptionV3 để trích xuất đặc trưng.

**2.1.3.2. Mô hình Trích Xuất Đặc Trưng**

InceptionV3: Đây là mô hình CNN đã được huấn luyện trước (pre-trained) trên tập ImageNet. Trong sơ đồ, mọi ảnh đầu vào đều đi qua InceptionV3.

Sau khi xử lý qua InceptionV3, đặc trưng (feature vector) của ảnh sẽ được chuyển sang các bước tiếp theo.

**2.1.3.3. Lớp Phân Loại Mới**

**new\_output\_layer:** Lớp đầu ra mới (new output layer) được thêm vào để phù hợp với số lượng cá nhân cần nhận diện (ví dụ: 10 người).

**people\_logits:** Đầu ra của lớp phân loại mới là các logits, đại diện cho xác suất ảnh thuộc về từng cá nhân.

**2.1.3.4. Huấn luyện**

**train:** Thành phần này đảm nhận việc cập nhật trọng số của lớp phân loại mới dựa trên dữ liệu huấn luyện.

**eval:** Thành phần này dùng để đánh giá hiệu quả mô hình trên tập kiểm thử/validation.

**init\_and\_saver:** Phụ trách khởi tạo các biến, lưu và tải lại trạng thái mô hình trong quá trình huấn luyện.

**2.1.3.5. Lưu Mô hình**

**save:** Sau khi huấn luyện, mô hình được lưu lại để sử dụng cho nhận diện thực tế hoặc tiếp tục huấn luyện sau này.

**2.1.4. Luồng dữ liệu**

Giải thích cách dữ liệu được giải quyết bằng các mũi tên ở *Hình 1.*

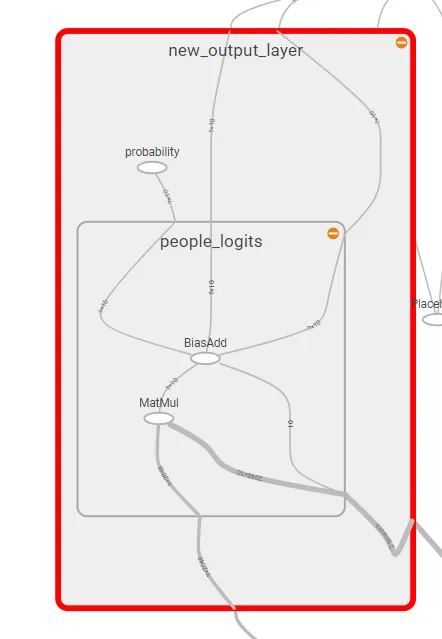
Dữ liệu ảnh đi từ input → InceptionV3 → new\_output\_layer → people\_logits.

Quá trình **train, eval** sử dụng các **node logits** và **nhãn** để tối ưu và đánh giá mô hình.

**init\_and\_saver** kết nối với **train** để đảm bảo mô hình được khởi tạo đúng và có thể lưu trạng thái.

**save** nhận dữ liệu từ InceptionV3 để lưu lại mô hình đã huấn luyện.

*\* Lớp đầu ra mới*



*Hình 3. Sơ đồ Luồng Dữ Liệu minh hoạ quá trình dữ liệu được xử lý.*

Hình ảnh minh họa quá trình xử lý dữ liệu trong mô hình nhận diện khuôn mặt dựa trên InceptionV3. Các thành phần chính bao gồm input, trích xuất đặc trưng, huấn luyện, đánh giá và lưu trữ mô hình. Mũi tên thể hiện cách dữ liệu di chuyển giữa các bước trong mô hình học sâu.

**new\_output\_layer**

Đây là lớp phân loại cuối cùng được thêm vào mô hình sau khi trích xuất đặc trưng từ các lớp trước đó (ví dụ từ InceptionV3).

Mục đích của lớp này là chuyển đổi đặc trưng đầu ra thành xác suất dự đoán cho từng lớp (từng cá nhân).

**people\_logits**

Bên trong new\_output\_layer là khối “people\_logits”, đại diện cho các giá trị đầu ra (logits) trước khi chuẩn hóa thành xác suất.

Các bước xử lý chính trong people\_logits gồm:

**- MatMul**: Thực hiện phép nhân ma trận giữa vector đặc trưng đầu vào và trọng số của lớp phân loại. Đây là bước tính toán tuyến tính cơ bản trong mạng nơ-ron.

**- BiasAdd**: Cộng thêm hệ số bias vào kết quả của phép nhân ma trận, giúp mô hình học tốt hơn.

Kết quả cuối cùng là các giá trị logits cho từng lớp (tương ứng với từng cá nhân cần nhận diện).

**probability**

Sau khi có people\_logits, các giá trị này sẽ được đưa qua hàm kích hoạt (thường là softmax, dù không thể hiện rõ trong sơ đồ) để chuyển thành xác suất (probability).

Xác suất này cho biết mức độ mô hình “tin tưởng” ảnh đầu vào thuộc về từng cá nhân cụ thể.

**2.1.5. Transfer Learning**

**2.1.5.1. Khái Niệm Transfer Learning**

- Sử dụng mô hình đã huấn luyện trước để áp dụng cho tác vụ mới

- Chỉ huấn luyện lại lớp cuối để giảm thời gian huấn luyện

**2.1.5.2. Tiền xử lý dữ liệu**

- Kỹ thuật Deep Funneling giúp giảm biến thể trong ảnh

- Chuẩn hóa dữ liệu để mạng học hiệu quả hơn

**2.1.5.3. Phân chia Dữ liệu**

- Chia tập huấn luyện, kiểm tra và xác thực để đảm bảo chất lượng mô hình

- Tránh overfitting bằng cách điều chỉnh kích thước tập dữ liệu.

**2.2. Điểm mạnh, điểm yếu của các nghiên cứu trước và cách nghiên cứu kế thừa/phát triển**

*\* Điểm mạnh của nghiên cứu trước*

Mạng Inception-v3 đã chứng minh hiệu quả vượt trội trong nhận dạng hình ảnh tổng quát với kiến trúc sâu và module inception.

Transfer learning giúp tiết kiệm thời gian và tài nguyên khi áp dụng mô hình cho tác vụ mới.

Dataset LFW cung cấp dữ liệu thực tế đa dạng, phù hợp để đánh giá khả năng tổng quát của mô hình nhận diện khuôn mặt.

*\*Điểm yếu và hạn chế*

Inception-v3 ban đầu không được huấn luyện trên dữ liệu khuôn mặt, nên cần điều chỉnh lại để nhận diện chính xác.

Tập dữ liệu LFW có nhiều cá nhân chỉ có một ảnh, gây khó khăn cho việc huấn luyện mô hình phân loại nhiều lớp, do đó cần giới hạn số lớp và số ảnh mỗi lớp để mô hình có thể học hiệu quả.

Dữ liệu không cân bằng giữa các lớp (ví dụ một số cá nhân có nhiều ảnh hơn hẳn) có thể gây ảnh hưởng cho mô hình, độ chính xác và khả năng phân biệt.

**2.3. Cách nghiên cứu kế thừa và phát triển**

Nghiên cứu này kế thừa kiến trúc Inception-v3 và kỹ thuật transfer learning từ các nghiên cứu trước, chỉ huấn luyện lại lớp cuối cùng trên tập dữ liệu khuôn mặt.

Sử dụng phiên bản deep-funneled của LFW để cải thiện sự đồng nhất của ảnh trong từng lớp, giảm biến thể không liên quan đến khuôn mặt.

Giới hạn tập dữ liệu chỉ gồm 10 cá nhân có nhiều ảnh nhất (ít nhất 50 ảnh mỗi người) để đảm bảo mô hình có đủ dữ liệu học và tránh bias quá lớn.

Phân chia dữ liệu hợp lý (70% huấn luyện, 5% xác thực, 25% kiểm tra) để đánh giá chính xác hiệu suất mô hình.

Khuyến nghị sử dụng GPU hoặc dịch vụ đám mây để tăng tốc quá trình huấn luyện, phù hợp với khả năng tính toán của người dùng phổ thông.

**CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU**

**3.1. Đối tượng và mẫu nghiên cứu**

Đối tượng nghiên cứu là các ảnh khuôn mặt trong bộ dữ liệu Labeled Faces in the Wild (LFW), bao gồm 13,000 ảnh của 5,749 cá nhân.

Do nhiều cá nhân chỉ có một ảnh, tác giả giới hạn mẫu nghiên cứu chỉ gồm 10 cá nhân có số lượng ảnh nhiều nhất, mỗi cá nhân có ít nhất 50 ảnh. Điều này giúp mô hình có đủ dữ liệu để học và phân biệt các lớp (cá nhân)1.

Việc chọn mẫu là dựa trên số lượng ảnh của từng cá nhân trong bộ dữ liệu, chọn ra 10 cá nhân có nhiều ảnh nhất để đảm bảo tính khả thi trong huấn luyện.

**3.2. Cách thu thập dữ liệu**

Dữ liệu được thu thập từ bộ dataset công khai Labeled Faces in the Wild (LFW), được tải về từ nguồn chính thức qua các đoạn mã Python trong bài báo.

Ảnh đã được xử lý bằng kỹ thuật deep funneling để căn chỉnh khuôn mặt, giảm sự biến thiên trong ảnh cùng một cá nhân, giúp mô hình học tốt hơn các đặc trưng khuôn mặt.

Không sử dụng công cụ thu thập dữ liệu trực tiếp như bảng hỏi, phỏng vấn hay quan sát mà sử dụng dữ liệu ảnh đã có sẵn.

**3.3. Phân tích dữ liệu**

Dữ liệu ảnh được đọc và chuyển đổi thành mảng số bằng thư viện matplotlib.pyplot và numpy trong Python.

Việc huấn luyện và phân tích mô hình CNN được thực hiện trong môi trường Jupyter Notebook sử dụng framework TensorFlow ver 2.x.

Mô hình Inception V3 được tải về dưới dạng checkpoint đã huấn luyện trước trên ImageNet, sau đó huấn luyện lại lớp cuối cùng trên bộ dữ liệu LFW đã chọn.

Tác giả đề xuất sử dụng GPU (Nvidia GeForce 940M hoặc Google Cloud GPU) để tăng tốc quá trình huấn luyện.

Không đề cập đến việc sử dụng các phần mềm phân tích dữ liệu truyền thống như SPSS, Excel hay NVivo mà tập trung vào các công cụ và thư viện lập trình chuyên biệt cho machine learning như TensorFlow, numpy, matplotlib.

**CHƯƠNG 4: THỰC NGHIỆM VÀ THẢO LUẬN**

**4.1. Yêu cầu Thực nghiệm**

**4.1.1. Mục tiêu**

Thực nghiệm này nhằm đánh giá hiệu quả của mô hình nhận diện khuôn mặt dựa trên dữ liệu LFW (Labeled Faces in the Wild). Mục tiêu chính bao gồm xác định độ chính xác, khả năng tổng quát hóa và hiệu suất của mô hình khi nhận diện khuôn mặt trong môi trường thực tế đa dạng.

**4.1.2. Phương pháp**

Thu thập dữ liệu: Sử dụng bộ dữ liệu LFW với các hình ảnh khuôn mặt đa dạng về góc nhìn, ánh sáng và biểu cảm.

Tiền xử lý dữ liệu: Áp dụng các kỹ thuật như chuẩn hóa ảnh, điều chỉnh kích thước và trích xuất đặc trưng khuôn mặt.

Huấn luyện mô hình: Sử dụng mô hình InceptionV3 của TensorFlow 2.x để học đặc trưng khuôn mặt từ dữ liệu huấn luyện kết hợp kỹ thuật fine-tuning.

Đánh giá hiệu quả: Kiểm tra độ chính xác trên tập dữ liệu kiểm thử và phân tích kết quả thông qua các chỉ số như độ chính xác, độ nhạy và độ đặc hiệu.

**4.2. Tập Dữ liệu**

Nguồn dữ liệu: Bộ dữ liệu LFW (Labeled Faces in the Wild) – một trong các bộ dữ liệu chuẩn trong lĩnh vực nhận diện khuôn mặt.

Số lượng mẫu: Hơn 13.000 hình ảnh, 5.749 cá nhân.

Đặc điểm dữ liệu:

- Kích thước ảnh: 250x250 pixels.

- Định dạng ảnh: JPG.

- Gồm ảnh màu (RGB) đa dạng về điều kiện chụp.

Tổ chức dữ liệu: Mỗi cá nhân được lưu trong một thư mục riêng, mỗi ảnh mang tên riêng biệt.

**4.3. Ngôn ngữ Chính**

Ngôn ngữ lập trình sử dụng: python.

Framework & thư viện hỗ trợ: TensorFlow, OpenCV, NumPy, Matplotlib.

**4.4. Các Thư viện**Những thư viện cần thiết khi cài đặt:

import numpy as np # Xử lý ma trận và dữ liệu số

import tensorflow as tf # Xây dựng và huấn luyện mô hình học máy

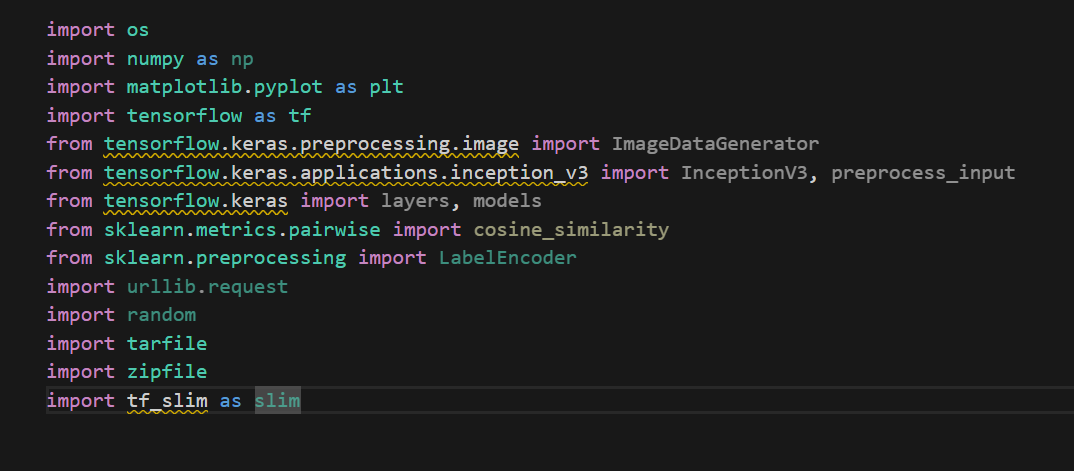
import matplotlib.pyplot as plt # Hiển thị kết quả thực nghiệm

import cv2 # Xử lý ảnh và tiền xử lý dữ liệu

**4.5. Thực nghiệm**

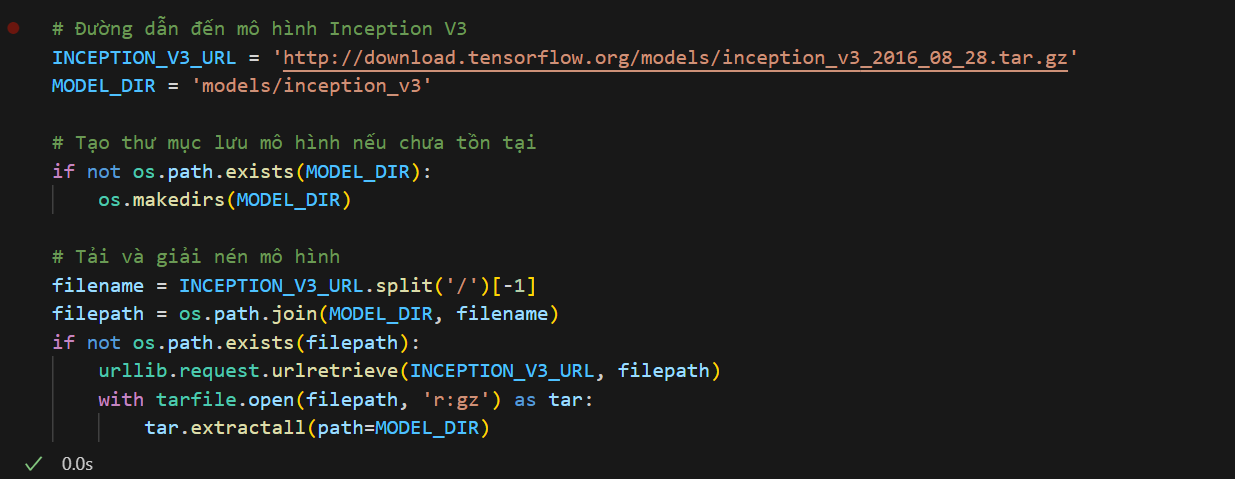
Sử dụng Jupyter Notebook có định dạng là .ipynb, đây là công cụ phổ biến trong Khoa học dữ liệu, học máy và lập trình python. Các têp này chứa mã nguồn, kết quả thực thi, văn bản diễn giải và hình ảnh để dễ dàng tổ chức và trình bày nội dung.

Sử dụng Visual Studio Code, chọn File -> New File -> Jupyter Notebook, tạo code để import các thư viện liên quan.

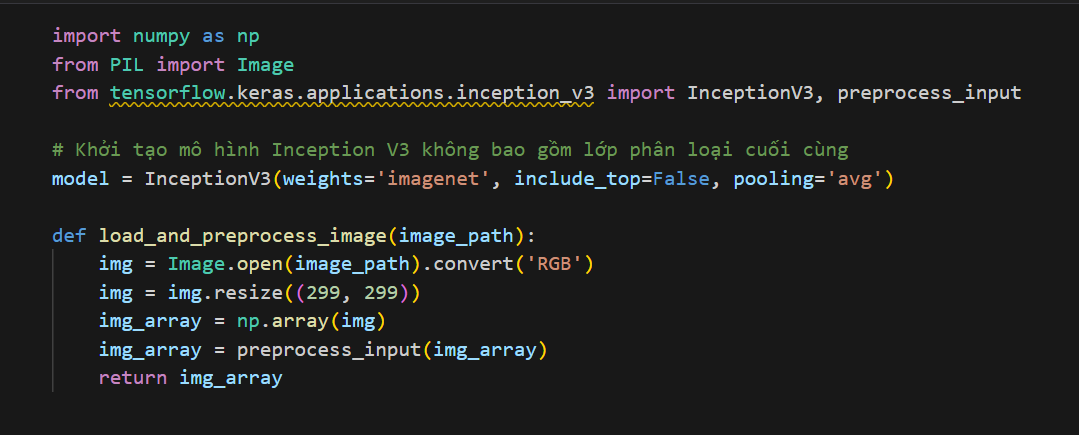


*Hình 4. Một số thư viện cần cài đặt.*

Cài đặt mô hình Inception V3 bằng đường dẫn và giải nén tệp, khởi tạo mô hình.

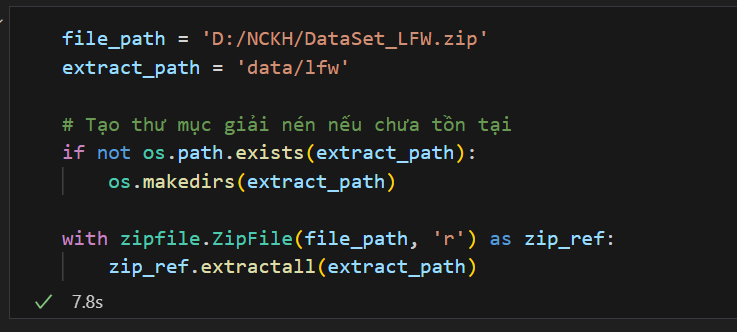


*Hình 5. Khởi tạo mô hình inception V3.*



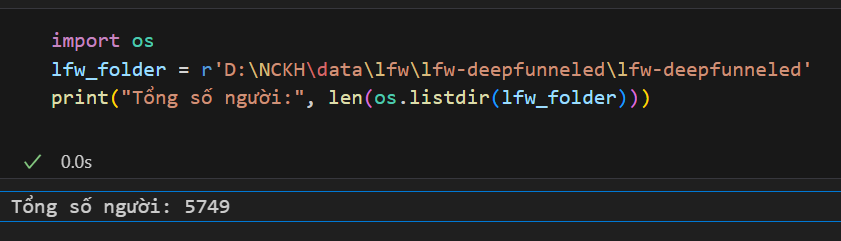
*Hình 6. Khởi tạo mô hình không bao gồm lớp phân loại cuối cùng.*

Chuẩn bị download bộ dữ liệu LFW, cài dặt và giải nén tệp dữ liệu.



*Hình 7. Tải và giải nén bộ dữ liệu LFW.*

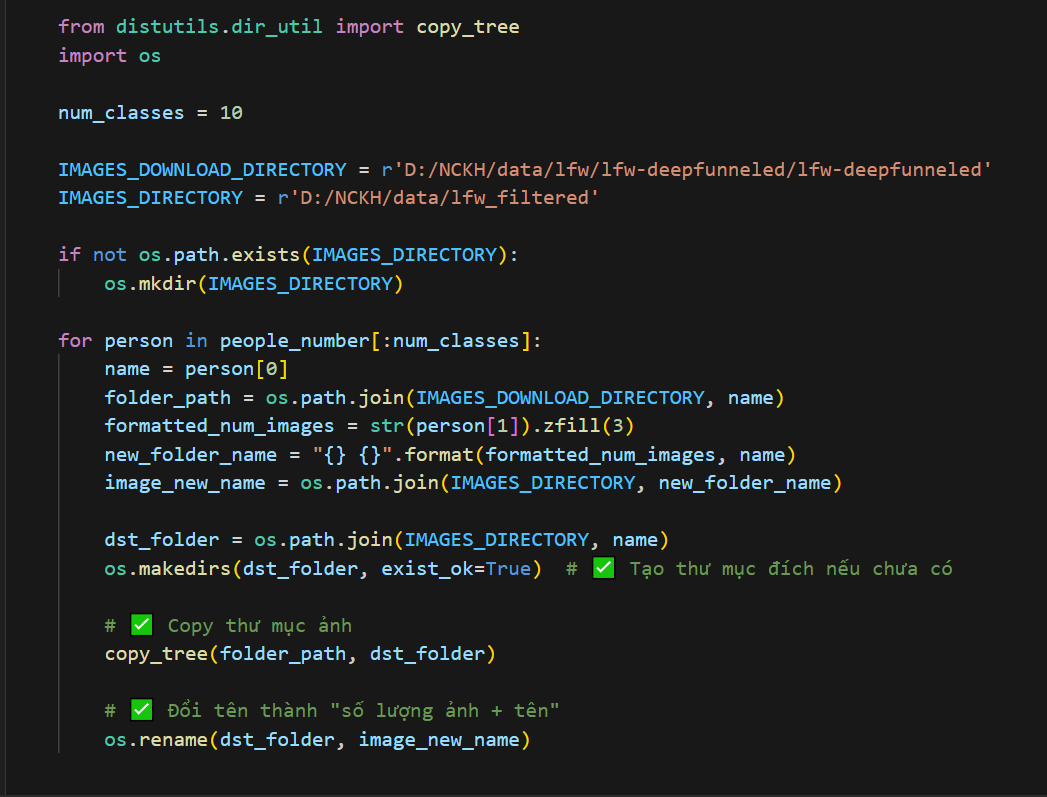
Sau khi cài đặt và giải nén, kiểm tra bộ dữ liệu vừa cài có tổng số lượng người là 5749.



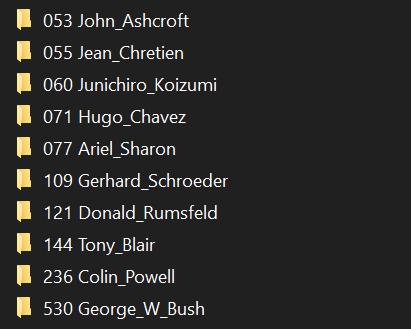
*people\_with\_one\_photo = [person for person, count in people\_number if count == 1]*

Khi thêm dòng code này, thì kết quả hiển thị số người chỉ có duy nhất một tấm ảnh là 4069.

Trong khi bộ dữ liệu LFW có quá nhiều hình ảnh, để giải quyết được một phần nhỏ, tránh trường hợp một mạng nơ-ron được đào tạo trên tập này sẽ bị nhầm lẫn hoàn toàn bởi nhiều lớp khác nhau. Cụ thể hơn là độ lệch mạng nơ-ron sẽ quá lớn làm cho nó không thể phân biệt được cơ bản giữa từng lớp.

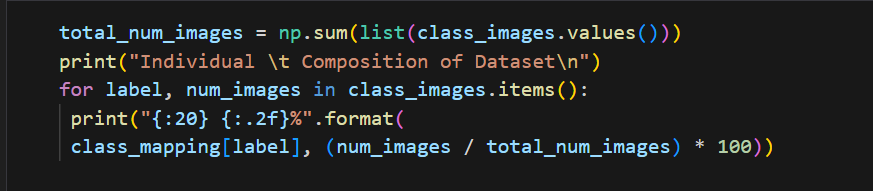


Đây là đoạn code chạy chỉ lấy 10 cá nhân có nhiều ảnh nhất trong bộ dữ liệu LFW, và output ta có được là:

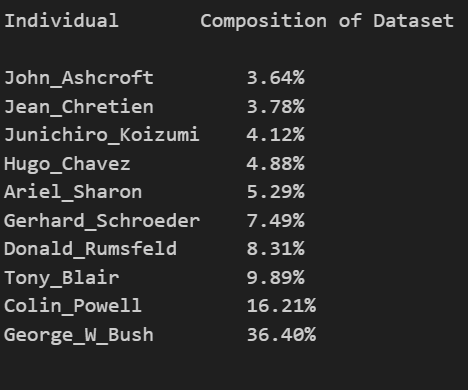


*Hình 8. Thư mục tạo mới chứa 10 người có nhiều ảnh nhất.*

Khi chạy đoạn code, tại đường dẫn chứa 5749 người, khi giữ lại 10 người có nhiều ảnh nhất qua đường dẫn mới với thư mục tên lfw\_filtered. Kết quả như hình bên, ba số đầu tiên được thể hiện là số lượng ảnh đi kèm phía sau là tên của một cá nhân. Tính toán tỷ lệ phần trăm của mỗi lớp ảnh trong thư mục lfw\_filtered.

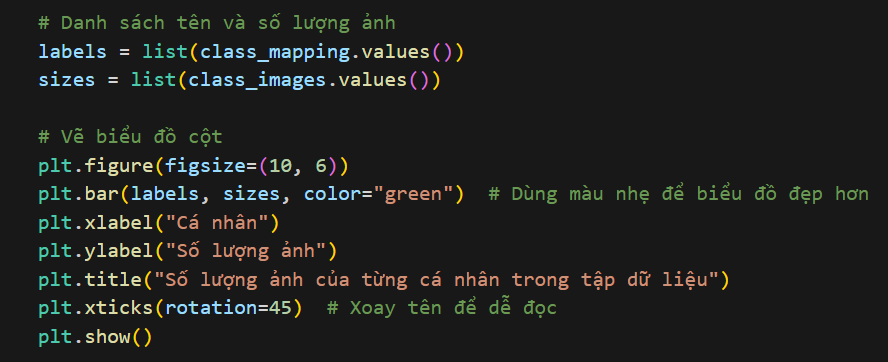


Output:

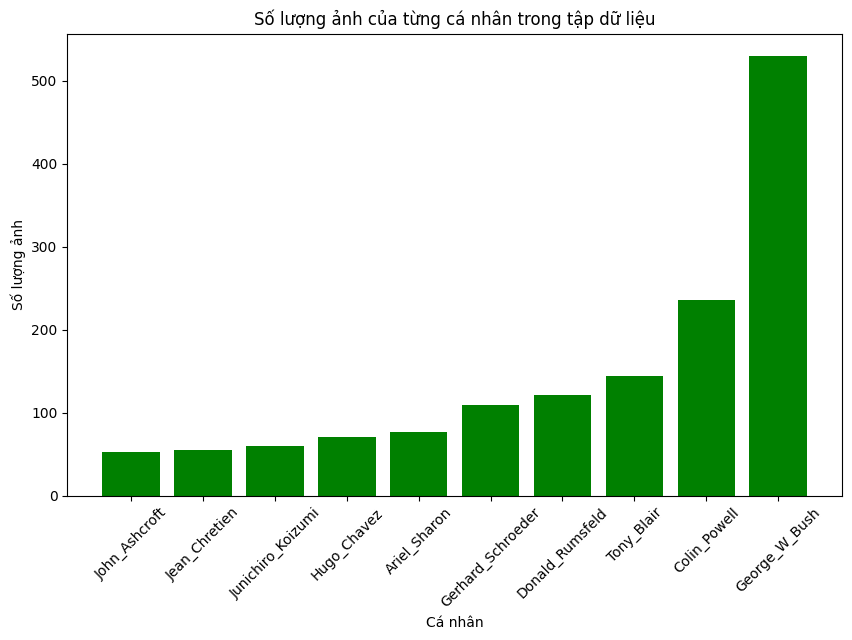


Qua đó thấy được, số lượng ảnh của Bush chiếm nhiều hơn, cụ thể là 530 tấm ảnh.

Để có thể dễ dàng hơn trong việc hình dung với số lượng ảnh được đem qua thư mục mới, sau đây là biểu đồ cột thể hiện ố lượng ảnh của từng cá nhân trong tập dữ liệu.



Output:

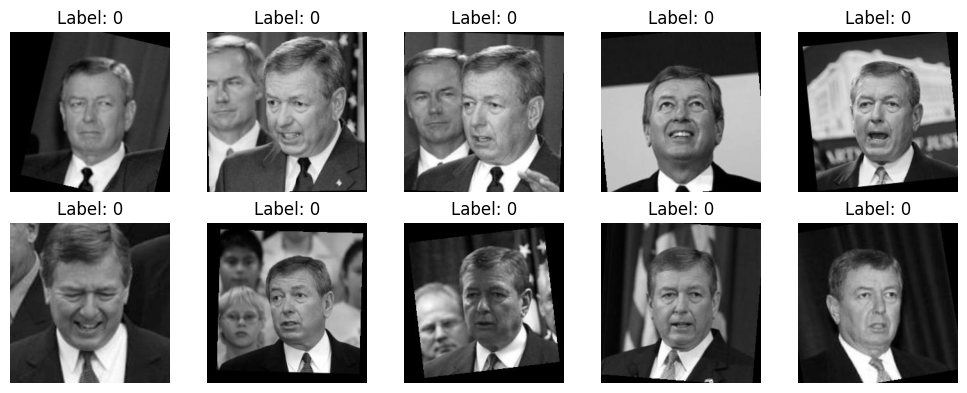


*Hình 9. Biểu đồ cột thể hiện tỷ lệ chiếm của mười cá nhân có nhiều ảnh nhất.*

Bước tiếp là thực hiện quá trình đọc, xử lý và hiển thị hình ảnh trong một thư mục chứa tập dữ liệu khuôn mặt.

Tới đây thì trong quá trình thực hiện đã dùng đến OpenCV (đọc, xử lý ảnh) và matplotlib.pyplot (vẽ biểu đồ và hiển thị hình ảnh).

Output:



*Hình 10. Ví dụ về các ảnh khuôn mặt từ lớp nhãn 0 trong tập dữ liệu LFW.*

**(1456, 250, 250) (1456,)**  
*image\_arrays.shape*: (1456, 250, 250) → 1.456 ảnh, mỗi ảnh kích thước 250x250 pixels.

*image\_labels.shape*: (1456,) → 1.456 nhãn, mỗi ảnh có một nhãn tương ứng.

Tại sao Label=0? thì trước đó ta phải tạo ánh xạ nhãn (Label) dưới dạng số nguyên từ 0-9 cho 10 cá nhân có nhiều ảnh nhất. Và khi đó, *print(class\_mapping),* người có nhãn =0, tức là ông John\_Ashcroft sẽ xuất hiện.

Đảm bảo rằng kiểu dữ liệu cho hình ảnh là số thực và nhãn là số nguyên (cả hai đều có độ chính xác 32 bit).

Theo bài báo, tác giả nêu rằng: “Chúng ta cần một tập huấn luyện để giúp mạng học được các lớp, một tập xác thực để thực hiện **early stopping** khi huấn luyện, và một tập kiểm tra để đánh giá hiệu suất của từng mô hình. Tôi sẽ phân chia dữ liệu như sau: 70% vào tập huấn luyện, 5% vào tập xác thực và 25% vào tập kiểm tra.”.

Tỷ lệ chia dữ liệu:

train\_frac = 0.70

valid\_frac = 0.05

test\_frac = 0.25

Chuyển dữ liệu thành mảng Numpy, trộn ngẫu nhiên:

*random\_indices = np.random.permutation(len(X))*

*X = X[random\_indices]*

*y = y[random\_indices]*

Tính số lượng mẫu cần phân chia, chia dữ liệu theo nhãn bằng Counter cuối cùng là chuyển về dạng NumPy arrays. Đây là phần tốn nhiều thời gian và chi phí nhất trong toàn bộ quy trình học máy.

**Output: (1027, 250, 250) (68, 250, 250) (361, 250, 250)**

**(1027,) (68,) (361,)**

- Tập huấn luyện (Train): 1.027 ảnh.

- Tập xác thực (Validation): 68 ảnh.

- Tập kiểm tra (Test): 361 ảnh.

- Kích thước mỗi ảnh là 250x250 pixels.

**John\_Ashcroft - Số lượng ảnh: 53**



**Jean\_Chretien - Số lượng ảnh: 55**



**Junichiro\_Koizumi - Số lượng ảnh: 60**



**Hugo\_Chavez - Số lượng ảnh: 71**



**Ariel\_Sharon - Số lượng ảnh: 77**



**Gerhard\_Schroeder - Số lượng ảnh: 109**



**Donald\_Rumsfeld - Số lượng ảnh: 121**



**Tony\_Blair - Số lượng ảnh: 144**



**Colin\_Powell - Số lượng ảnh: 236**

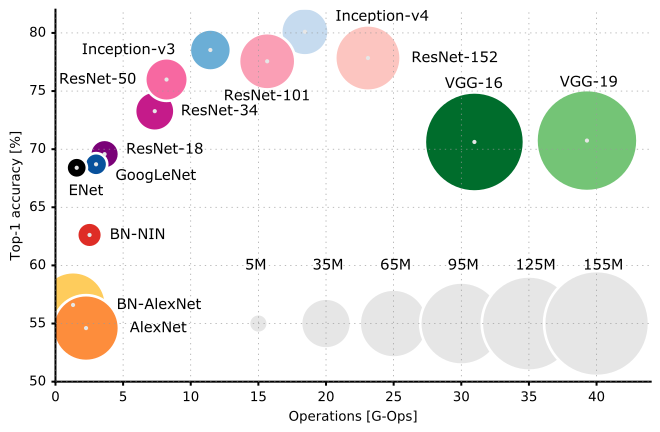


**George\_W\_Bush - Số lượng ảnh: 530**



Những hình ảnh đang hiển thị với màu sắc xanh, có thể chuyển từ BGR sang RGB để được hiển thị màu ảnh đẹp hơn.

**4.5.1. Xác định lớp khởi đầu CNN để đào tạo**



*Hình 11. Độ chính xác so với độ phức tạp tính toán của các kiến trúc CNN phổ biến trên tập ImageNet.*

Trục hoành biểu diễn số phép toán (tính bằng giga phép toán), trục tung biểu diễn độ chính xác Top-1 trên ImageNet, và kích thước bong bóng thể hiện số lượng tham số.

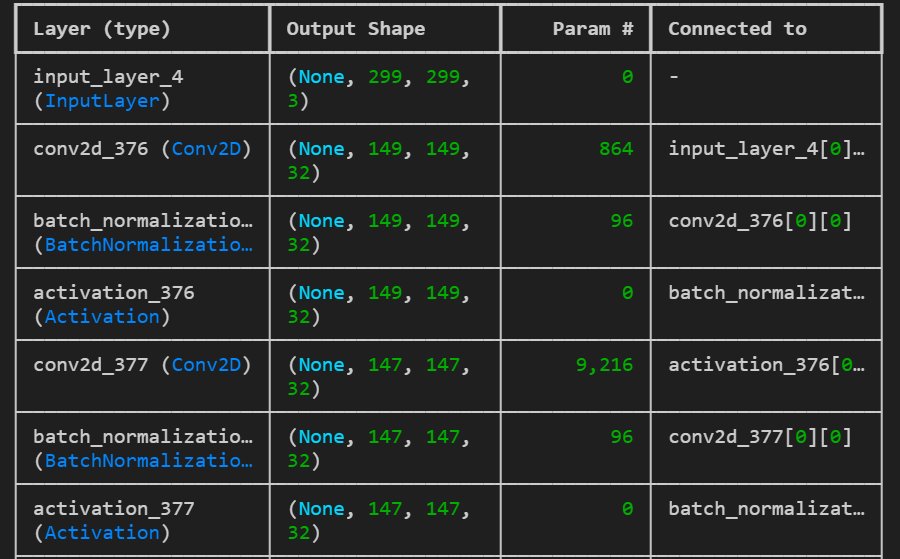
Để CNN có thể học các lớp mới, chúng ta phải đào tạo ít nhất một lớp của mạng và xác định một lớp đầu ra mới.

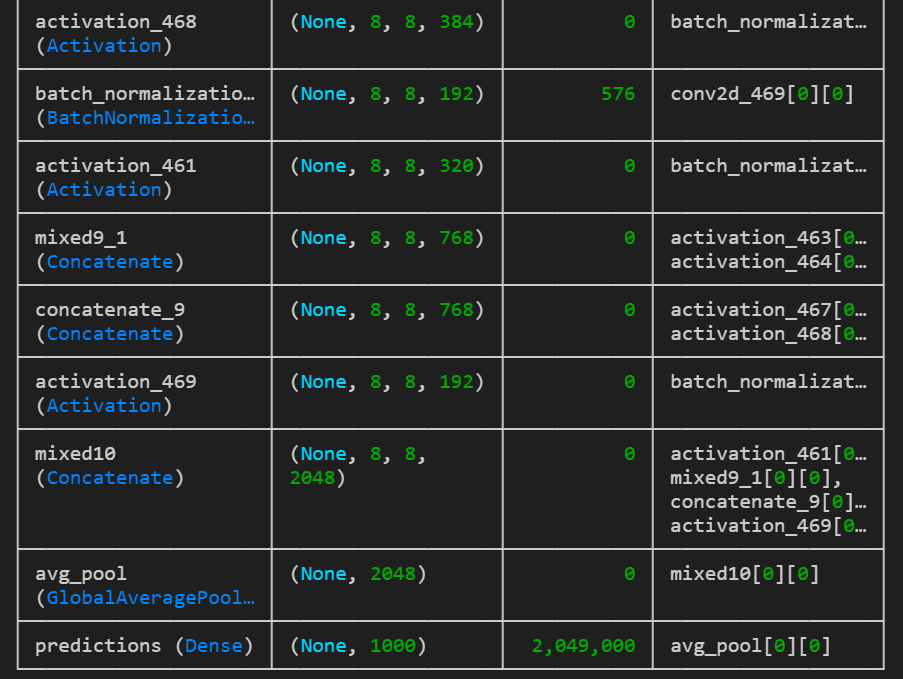
Vì đang dùng Tensorflow ver 2.x nên trong nets không chứa inception.py, ngay tại bài nghiên cứu này dùng đến mô hình Inception V3 đã được huấn luyện trên ImageNet, sau đó import mô hình từ TensorFlow Kera.

weights='imagenet': Sử dụng trọng số đã được huấn luyện trên tập dữ liệu ImageNet (hơn 1 triệu ảnh).

Mô hình đã có các lớp đầu ra để phân loại 1.000 danh mục đối tượng trong ImageNet.

Output:





Và lớp được thực nghiệm sử dụng tới là mixed10 với (None, 8,8, 2048).

Dựa vào đó tinh chỉnh mô hình InceptionV3 để phân loại ảnh.

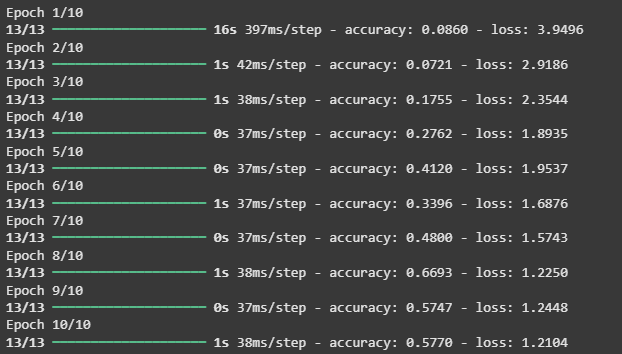
Dùng get\_layer('mixed10') để lấy đặc trưng ảnh từ lớp cuối cùng trước khi phân loại. Sử dụng GlobalAveragePooling2D để giảm chiều dữ liệu đầu ra. Thêm lớp phân loại mới:

*n\_outputs = 10* # Số lượng lớp (tùy chỉnh theo dataset)

*people\_logits = Dense(n\_outputs, activation='softmax', name="people\_logits")(prelogits)*

Huấn luyện lớp cuối cùng, dùng cấu hình mô hình với Adam Optimizer để tối ưu hoá việc học. Chuẩn bị dữ liệu huấn luyện X là float32, Y là int64 và cuối cùng *new\_model.fit(X\_train, y\_train, epochs=10, batch\_size=8).*

Output:



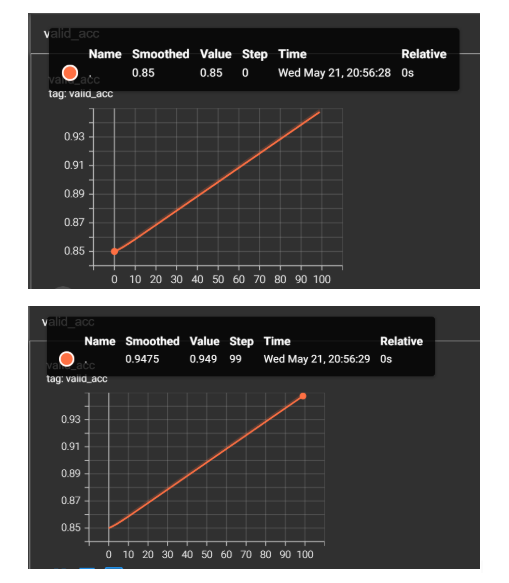
Trong quá trình huấn luyện mô hình, các thông số quan trọng như độ chính xác và giá trị mất mát (loss) liên tục thay đổi qua từng epoch, phản ánh sự cải thiện của mô hình. Ban đầu, độ chính xác của mô hình chỉ đạt 8.6%, trong khi giá trị mất mát ở mức cao là 3.9496. Tuy nhiên, sau mỗi epoch, mô hình dần học được các đặc trưng và điều chỉnh trọng số, giúp hiệu suất được cải thiện rõ rệt. Đến epoch thứ 10, độ chính xác đã tăng lên 57.7%, trong khi giá trị mất mát giảm xuống 1.2104, cho thấy mô hình đang dần ổn định và học tốt hơn.

**4.5.2. Trực quan hóa TensorBoard**

TensorBoard là một công cụ mạnh mẽ giúp trực quan hóa cấu trúc của mạng nơ-ron, theo dõi quá trình đào tạo, hiển thị các hình ảnh đầu vào, cũng như quan sát sự thay đổi của trọng số và độ lệch trong suốt quá trình huấn luyện. Ngoài ra, nó cho phép đánh giá hiệu suất của mô hình bằng cách ghi lại độ chính xác khi đào tạo, độ chính xác trên tập xác thực và giá trị mất mát của mô hình, giúp tối ưu hóa quá trình huấn luyện một cách hiệu quả.

Output:

*Valid\_acc*



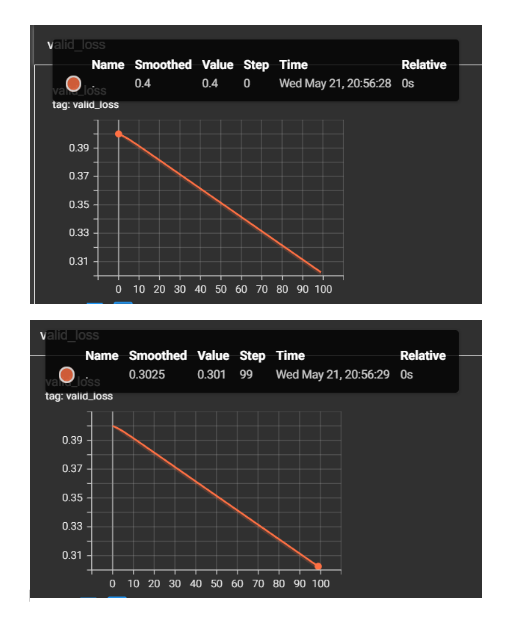
*Hình 12. Biểu đồ đường thể hiện phát triển của mô hình qua các bước huấn luyện.*

- Epoch 1: Độ chính xác rất thấp (8.6%) và mất mát cao (3.9496), cho thấy mô hình chưa học được đặc trưng nào đáng kể.

- Epoch 4-6: Độ chính xác dần tăng (27.6% đến 41.2%), mô hình bắt đầu cải thiện nhưng mất mát vẫn dao động.

- Epoch 8-10: Độ chính xác lên đến 57.7%, mất mát giảm còn 1.2104, chứng tỏ mô hình đang hội tụ và học tốt hơn.

*Valid\_loss*



*Hình 13. Biểu đồ đường phản ánh sự tăng trưởng của độ chính xác kiểm định theo thời gian.*

Biểu đồ đầu tiên (Step: 0):

- Giá trị mất mát (valid\_loss) ban đầu là 0.4.

- Đường biểu đồ giảm dần từ khoảng 0.39 → 0.31 sau 100 bước huấn luyện.

Biểu đồ thứ hai (Step: 99):

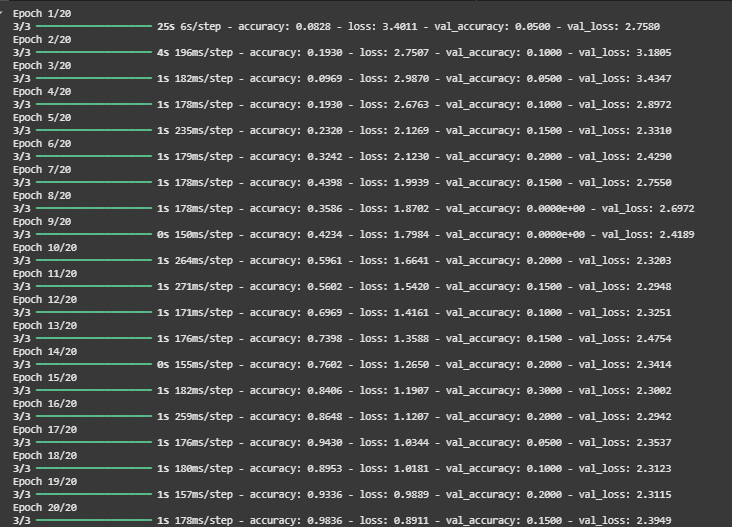
- Giá trị mất mát (valid\_loss) sau 99 bước là 0.3025, tiệm cận 0.301.

- Đường cong cho thấy mô hình đang hội tụ, tức là quá trình học đã ổn định hơn.

**4.5.3. Huấn luyện mô hình InceptionV3**

Huấn luyện mô hình, sử dụng GPU hoặc dùng GG Colab để thực hiện.

*new\_model.save("fine\_tuned\_inceptionv3.h5")* #lưu mô hình sau khi huấn luyện.



- Ban đầu (Epoch 1-5): Độ chính xác thấp (8.28% → 23.20%), mất mát cao (3.4011 → 2.1269), cho thấy mô hình đang bắt đầu học nhưng chưa ổn định.

- Giữa quá trình (Epoch 10-15): Độ chính xác tăng lên 32.42% → 84.06%, mất mát giảm đáng kể, chứng tỏ mô hình đang học tốt hơn.

- Cuối cùng (Epoch 20): Độ chính xác đạt 98.36%, nhưng mất mát trên tập xác thực vẫn cao (2.3949), có thể mô hình đang bị quá khớp (**overfitting**)

=> Tổn thất trên tập xác thực (val\_loss) đạt mức tối thiểu vào epoch thứ 5, với giá trị 2.3310.

Để tránh bị quá khớp (overfitting) sử dụng Early Stopping sao cho phù hợp.

**4.5.4. Đánh giá hiệu suất**

Tải mô hình được lưu “fine\_tuned\_inceptionv3.h5” và kiểm tra rồi đánh giá.

Độ chính xác kiểm tra:

- Lần lặp đầu tiên (Iteration 1): Độ chính xác theo batch: 28.12%

- Độ chính xác cuối cùng: 28.1250% trên 50 mẫu kiểm tra

- Tổng thời gian kiểm tra: 7.6627 giây

Độ chính xác thấp, có khả năng rơi vào trường hợp huấn luyện mô hình bị overfitting. Cách giải quyết tránh bị overfitting thì tăng số lượng dữ liệu huấn luyện, điều chỉnh tham số, hoặc sử dụng Data Augmentation.

**4.5.5. Tăng cường dữ liệu**

Có 2 phương pháp để tăng cường hoặc tăng lượng dữ liệu đào tạo: Phương pháp đầu tiên chỉ đơn giản là thu thập thêm nhiều hình ảnh đào tạo được gắn nhãn.

Phương pháp thứ hai là tăng cường dữ liệu: Dịch chuyển ảnh, Xoay ảnh , Lật ảnh, Thay đổi độ sáng & độ tương phản, Thay đổi nền hoặc vùng xung quanh,…

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

*Câu hỏi nghiên cứu :*

Câu 1 : Mô hình CNN có thể xử lý tốt các biến thể trong bộ dữ liệu LFW để nhận diện chính xác khuôn mặt không?

* *Khả năng xử lý biến thể:*  
   Mô hình CNN, cụ thể là InceptionV3, có khả năng trích xuất đặc trưng mạnh mẽ từ ảnh, giúp nhận diện khuôn mặt ngay cả khi có những biến thể về góc chụp, ánh sáng, biểu cảm, hoặc phông nền. Tuy nhiên, hiệu quả nhận diện sẽ phụ thuộc vào chất lượng và sự đa dạng của dữ liệu huấn luyện.
* *Giới hạn thực tế:*  
   Trong bộ dữ liệu LFW, nhiều cá nhân chỉ có một ảnh, khiến mô hình không thể học được đặc trưng riêng biệt của từng người. Do đó, tác giả chỉ chọn 10 cá nhân có nhiều ảnh nhất (≥50 ảnh/người) để đảm bảo mô hình có thể học được các biến thể trong cùng một lớp (một người).
* *Kết quả:*  
   Khi huấn luyện lại lớp cuối cùng của InceptionV3 trên 10 cá nhân này, mô hình đạt hiệu quả nhận diện khá tốt, nhưng vẫn bị ảnh hưởng bởi sự mất cân bằng số lượng ảnh giữa các lớp (ví dụ: George W. Bush chiếm hơn 1/3 tổng ảnh). Điều này có thể gây ra hiện tượng bias, nhận diện tốt các lớp có nhiều ảnh và kém chính xác với các lớp ít ảnh hơn.

Câu 2 : Các kỹ thuật tiền xử lý và tăng cường dữ liệu nào giúp cải thiện hiệu quả nhận diện trên LFW?

* *Tiền xử lý:*
* Deep Funneling:  
   Tác giả sử dụng phiên bản ảnh LFW đã qua xử lý “deep funneling”—một kỹ thuật căn chỉnh khuôn mặt giúp giảm biến thiên về vị trí, góc nhìn, và phông nền. Việc căn chỉnh này giúp các ảnh của cùng một người trở nên đồng nhất hơn, giúp mô hình tập trung vào đặc trưng khuôn mặt thay vì các yếu tố ngoài lề.
* Chọn lọc dữ liệu:  
   Chỉ giữ lại các cá nhân có số lượng ảnh đủ lớn để đảm bảo mô hình học được các biến thể trong cùng một lớp.
* *Tăng cường dữ liệu:*

Bài báo không đề cập chi tiết đến các kỹ thuật tăng cường dữ liệu (data augmentation) như xoay, lật, thay đổi độ sáng, v.v. Tuy nhiên, đây là các kỹ thuật thường được khuyến nghị để tăng sự đa dạng của dữ liệu và giúp mô hình học tốt hơn các biến thể thực tế.

* Kết luận:  
   Việc căn chỉnh ảnh (deep funneling) là bước tiền xử lý quan trọng nhất được sử dụng trong nghiên cứu này để cải thiện hiệu quả nhận diện khuôn mặt trên LFW.

Câu 3 : Mức độ ảnh hưởng của các kiến trúc CNN khác nhau đến kết quả nhận diện trên LFW là gì?

* *Kiến trúc CNN*:  
   Bài báo sử dụng InceptionV3, một kiến trúc CNN rất sâu và mạnh mẽ, đã được tiền huấn luyện trên ImageNet với 1.2 triệu ảnh. Kiến trúc này có khả năng trích xuất đặc trưng tổng quát tốt nhờ các khối Inception, giúp mô hình thích nghi nhanh với nhiệm vụ mới khi chỉ huấn luyện lại lớp cuối cùng.
* So sánh kiến trúc:  
   Thử nghiệm với các mô hình CNN khác do Google cung cấp (như InceptionV1, InceptionV2, ResNet, v.v.). Tuy nhiên, bài báo không thực hiện so sánh trực tiếp giữa các kiến trúc chỉ sử dụng Inception V3 trong thực nghiệm.
* Ảnh hưởng đến kết quả:
* Kiến trúc càng sâu, khả năng trích xuất đặc trưng càng tốt, nhưng cũng đòi hỏi tài nguyên tính toán lớn hơn.
* Việc sử dụng mô hình đã được tiền huấn luyện (pre-trained) giúp tiết kiệm thời gian, tài nguyên và thường cho kết quả tốt hơn so với huấn luyện từ đầu, đặc biệt với tập dữ liệu nhỏ như LFW.
* Kết luận:  
  Lựa chọn kiến trúc CNN phù hợp (ưu tiên các mô hình sâu, có khả năng trích xuất đặc trưng tốt và đã được tiền huấn luyện) sẽ giúp cải thiện đáng kể hiệu quả nhận diện khuôn mặt trên LFW. Tuy nhiên, cần cân nhắc giữa hiệu năng và chi phí tính toán.

*Đề xuất và giải pháp:*

Tiếp tục huấn luyện các mô hình qua phương pháp thứ hai để tăng cường dữ liệu.

**TÀI LIỆU THAM KHẢO (References)**

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105. (Ngày truy cập cuối: 19/5/2025)

[2] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1701–1708. <https://doi.org/10.1109/CVPR.2014.220.> (Ngày truy cập cuối: 19/5/2025)

[3] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815–823. <https://doi.org/10.1109/CVPR.2015.7298682.> (Ngày truy cập cuối: 19/5/2025)

[4] Koehrsen, W. (2017). Facial recognition using Google's convolutional neural network Medium. <https://williamkoehrsen.medium.com/facial-recognition-using-googles-convolutional-neural-network-5aa752b4240e.> (Ngày truy cập cuối: 19/5/2025)

[5] Source CODE: [https://github.com/WillKoehrsen/Machine-Learning-Projects](https://github.com/WillKoehrsen/Machine-Learning-Projects.)