# LỜI NÓI ĐẦU

Nhận diện khuôn mặt đã trở thành một trong những lĩnh vực phát triển nhanh chóng và được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống. Công nghệ này cho phép hệ thống máy tính nhận biết, xác định danh tính hoặc xác thực con người dựa trên các đặc điểm khuôn mặt được trích xuất từ hình ảnh hoặc video. Nhờ sự phát triển của trí tuệ nhân tạo (AI), học sâu (Deep Learning) và các thuật toán thị giác máy tính (Computer Vision), hiệu suất của các hệ thống nhận diện khuôn mặt đã được cải thiện vượt bậc trong những năm gần đây. Các mô hình tiên tiến có khả năng xử lý hàng triệu khuôn mặt với độ chính xác cao, thậm chí trong các điều kiện bất lợi như ánh sáng kém, góc nhìn thay đổi, hay khuôn mặt bị che khuất một phần.

Tuy nhiên, bài toán này vẫn đối mặt với nhiều thách thức, đặc biệt là trong môi trường thực tế. Các yếu tố như biến đổi về ánh sáng, biểu cảm, trang phục (ví dụ: kính, khẩu trang), hoặc các yếu tố giả mạo (face spoofing) vẫn là trở ngại lớn. Bên cạnh đó, vấn đề bảo mật và quyền riêng tư trong việc sử dụng công nghệ nhận diện khuôn mặt cũng đang thu hút nhiều sự chú ý từ phía dư luận và giới nghiên cứu.

Nhận diện khuôn mặt đã và đang được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ các hệ thống bảo mật cho đến các ứng dụng tiêu dùng. Một số ứng dụng tiêu biểu bao gồm:

* **Bảo mật và kiểm soát truy cập**: Nhận diện khuôn mặt được sử dụng trong các hệ thống xác thực sinh trắc học, hỗ trợ mở khóa điện thoại, truy cập máy tính, hoặc vào các khu vực bảo mật cao như sân bay và cơ quan chính phủ.
* **Giám sát và an ninh**: Camera an ninh được tích hợp công nghệ nhận diện khuôn mặt giúp phát hiện và theo dõi các đối tượng khả nghi trong không gian công cộng, giúp hỗ trợ các cơ quan chức năng trong việc ngăn ngừa tội phạm.
* **Mạng xã hội và truyền thông**: Các nền tảng như Facebook, Instagram sử dụng công nghệ này để tự động gắn thẻ tên người trong ảnh và video, cải thiện trải nghiệm người dùng.
* **Thanh toán và thương mại điện tử**: Công nghệ nhận diện khuôn mặt được áp dụng để xác thực thanh toán điện tử, giúp giảm thiểu rủi ro gian lận trong giao dịch trực tuyến.
* **Y tế**: Nhận diện khuôn mặt hỗ trợ trong việc theo dõi tình trạng bệnh nhân, phát hiện các bệnh lý di truyền và thậm chí giúp quản lý bệnh nhân mất trí nhớ trong các viện dưỡng lão.

Mặc dù các phương pháp dựa trên học sâu (Deep Learning) đã đạt được những thành tựu ấn tượng trong việc cải thiện độ chính xác của nhận diện khuôn mặt, nhưng chúng đòi hỏi lượng lớn dữ liệu và tài nguyên tính toán. Điều này khiến các mô hình học sâu khó có thể áp dụng trong các hệ thống nhỏ gọn hoặc môi trường giới hạn về tài nguyên.

Trong bối cảnh này, thuật toán **Máy vector hỗ trợ** (Support Vector Machine - SVM) nổi lên như một phương pháp có thể giải quyết bài toán nhận diện khuôn mặt hiệu quả với lượng dữ liệu vừa phải và chi phí tính toán thấp hơn so với các mô hình học sâu. SVM có khả năng phân loại mạnh mẽ trong không gian đặc trưng cao, giúp phân biệt các đặc điểm khuôn mặt một cách chính xác, ngay cả trong những trường hợp khó khăn như ánh sáng kém hoặc thay đổi biểu cảm. Nhờ vào sự tổng quát hóa tốt, SVM vẫn là một phương pháp lý tưởng trong nhiều trường hợp mà các hệ thống học sâu trở nên kém hiệu quả.

Trong bối cảnh nhận diện khuôn mặt đang ngày càng trở thành một công nghệ quan trọng với nhiều ứng dụng thực tiễn, việc nghiên cứu và phát triển các thuật toán hiệu quả cho bài toán này là cần thiết. Với những ưu điểm của SVM như khả năng phân loại tốt trên dữ liệu nhỏ, chi phí tính toán thấp và tính ứng dụng cao, việc lựa chọn thuật toán SVM để giải quyết bài toán nhận diện khuôn mặt là một hướng đi hợp lý và có nhiều tiềm năng.

Hơn nữa, đề tài này cũng nhằm khai thác và so sánh tính hiệu quả của SVM với các phương pháp hiện đại khác, từ đó đóng góp vào việc tối ưu hóa và phát triển các hệ thống nhận diện khuôn mặt trong tương lai. Việc lựa chọn nghiên cứu ứng dụng SVM vào bài toán nhận diện khuôn mặt không chỉ có ý nghĩa học thuật mà còn có tiềm năng áp dụng rộng rãi trong thực tế, đặc biệt trong các hệ thống yêu cầu độ chính xác cao nhưng lại bị hạn chế về tài nguyên.

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc180858871)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 5](#_Toc180858872)

[DANH MỤC HÌNH VẼ 5](#_Toc180858873)

[DANH MỤC BẢNG BIỂU 5](#_Toc180858874)

[TÓM TẮT ĐỒ ÁN 5](#_Toc180858875)

[PHẦN MỞ ĐẦU 6](#_Toc180858876)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc180858877)

[1.1 Bài toán nhận diện khuôn mặt 7](#_Toc180858878)

[1.1.1 Tổng quan 7](#_Toc180858879)

[1.1.2 Học một lần (One-shot Learning) 8](#_Toc180858880)

[1.2 Máy Vector hỗ trợ (Support Vector Machine) 9](#_Toc180858881)

[1.2.1 Xây dựng bài toán tối ưu cho máy vector hỗ trợ 9](#_Toc180858882)

[1.2.2 Máy vector hỗ trợ đa lớp 11](#_Toc180858883)

[1.3 Mạng Nơ-ron tích chập 12](#_Toc180858884)

[1.3.1 Lớp tích chập 12](#_Toc180858885)

[1.3.2 Lớp tổng hợp 15](#_Toc180858886)

[1.3.3 Lớp kết nối đầy đủ 16](#_Toc180858887)

[1.4 Mô hình ResNet 17](#_Toc180858888)

[1.5 Kết luận chương 18](#_Toc180858889)

[CHƯƠNG 2. PHƯƠNG PHÁP ĐỀ XUẤT 19](#_Toc180858890)

[2.1 Các nghiên cứu liên quan 20](#_Toc180858891)

[2.1.1 Mô hình ArcFace 20](#_Toc180858892)

[2.2 Mô hình đề xuất 22](#_Toc180858893)

[CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ 23](#_Toc180858894)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| CNN | Convolution Neural Network | Mạng Nơ-ron tích chập |

# DANH MỤC HÌNH VẼ

# DANH MỤC BẢNG BIỂU

# TÓM TẮT ĐỒ ÁN

# PHẦN MỞ ĐẦU

# CƠ SỞ LÝ THUYẾT

## Bài toán nhận diện khuôn mặt

### Tổng quan

Nhận diện khuôn mặt là một trong những lĩnh vực quan trọng của thị giác máy tính và trí tuệ nhân tạo. Bài toán này liên quan đến việc phát hiện và nhận dạng khuôn mặt của con người từ các hình ảnh hoặc video. Đây là một bài toán phức tạp, bởi mỗi người có cấu trúc khuôn mặt khác nhau, và hình ảnh khuôn mặt có thể bị ảnh hưởng bởi nhiều yếu tố như ánh sáng, góc nhìn, biểu cảm, hay sự xuất hiện của các vật cản (như kính mắt, mũ, khẩu trang, v.v.).

***Các bước chính trong bài toán nhận diện khuôn mặt:***

**Phát hiện khuôn mặt (Face Detection)**: Trước tiên, hệ thống cần phát hiện xem trong hình ảnh hoặc video có khuôn mặt nào không và xác định vị trí của chúng. Các phương pháp phổ biến sử dụng các mô hình như Viola-Jones, HOG + SVM, hoặc mạng nơ-ron tích chập (CNN).

**Tiền xử lý hình ảnh (Image Preprocessing)**: Sau khi phát hiện được khuôn mặt, hình ảnh cần được tiền xử lý như điều chỉnh độ sáng, tương phản, xoay chỉnh khuôn mặt, cắt khung, và chuẩn hóa kích thước để tạo điều kiện thuận lợi cho bước nhận dạng.

**Trích xuất đặc trưng (Feature Extraction)**: Đây là bước chuyển đổi khuôn mặt thành các đặc trưng đại diện. Những đặc trưng này là những mô tả toán học về hình dáng, kết cấu hoặc các điểm nổi bật của khuôn mặt. Một số phương pháp như LBP (Local Binary Patterns), PCA (Principal Component Analysis), hoặc các mô hình học sâu như FaceNet, DeepFace thường được sử dụng.

**So sánh và nhận diện (Face Recognition)**: Sau khi trích xuất đặc trưng, hệ thống sẽ so sánh các đặc trưng này với cơ sở dữ liệu có sẵn để xác định danh tính của người trong ảnh. Một số kỹ thuật so sánh bao gồm sử dụng khoảng cách Euclid hoặc cosine similarity.

*Nhận diện khuôn mặt được ứng dụng rộng rãi trong nhiều lĩnh vực như:*

**Bảo mật**, nhận diện khuôn mặt được sử dụng rộng rãi trong bảo mật thiết bị như điện thoại, máy tính và các hệ thống an ninh. **Giám sát**, hỗ trợ nhận diện người qua camera an ninh trong các không gian công cộng. **Mạng xã hội**, các ứng dụng như Facebook, Instagram dùng để tự động gợi ý thẻ tên bạn bè khi tải ảnh. **Thanh toán điện tử**, hỗ trợ xác thực người dùng trong các hệ thống thanh toán trực tuyến ,v.v.

Mặc dù nhận diện khuôn mặt đã có những bước tiến đáng kể, nó vẫn đối mặt với nhiều thách thức, đặc biệt là trong môi trường ánh sáng yếu, khuôn mặt bị che khuất, hoặc các tình huống giả mạo (face spoofing).

### Học một lần (One-shot Learning)

Học một lần(tiếng anh là One-shot Learning) là một lĩnh vực trong học máy và trí tuệ nhân tạo, nhằm mục tiêu xây dựng các mô hình có khả năng học hỏi và nhận diện các đối tượng mới chỉ từ một hoặc rất ít ví dụ huấn luyện. Điều này khác biệt so với các phương pháp học truyền thống yêu cầu một lượng lớn dữ liệu để đạt được hiệu suất cao. One-shot learning có nhiều ưu điểm như: *Tiết kiệm Dữ liệu*, giảm nhu cầu thu thập nhiều hình ảnh cho mỗi cá nhân, tiết kiệm thời gian và tài nguyên. *Tính Linh hoạt Cao*, cho phép hệ thống nhanh chóng thích nghi với các cá nhân mới mà không cần huấn luyện lại toàn bộ mô hình*. Hiệu quả Trong Thực Tế*, thích hợp cho các ứng dụng thực tế nơi dữ liệu hạn chế hoặc thay đổi thường xuyên.

Trong lĩnh vực nhận diện khuôn mặt, One-shot Learning mang lại nhiều lợi ích đáng kể, giúp cải thiện hiệu suất và tính linh hoạt của hệ thống nhận diện. Trong nhiều tình huống thực tế, việc thu thập nhiều hình ảnh của một cá nhân để huấn luyện mô hình nhận diện là không khả thi hoặc tốn kém. Ví dụ:

* **An ninh và Giám sát**: Các hệ thống camera an ninh cần nhận diện những cá nhân mới chỉ từ một hoặc vài hình ảnh đầu vào.
* **Hệ thống Đăng nhập Sinh trắc học**: Người dùng mới chỉ cần đăng ký với một hình ảnh duy nhất để hệ thống có thể nhận diện trong các lần đăng nhập tiếp theo.

Các Phương pháp One-shot Learning phổ biến trong Nhận diện Khuôn mặt:

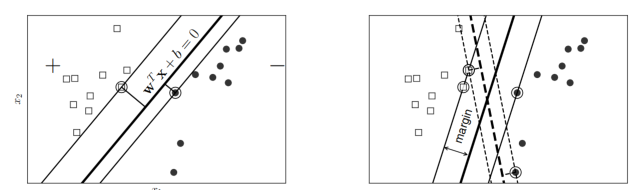
Mạng Siamese (Siamese Networks): Sử dụng hai nhánh mạng thần kinh để so sánh hai hình ảnh khuôn mặt và xác định xem chúng có thuộc cùng một cá nhân hay không. Phù hợp cho việc xác thực khuôn mặt trong thời gian thực.

Mạng Triplet (Triplet Networks): Sử dụng ba hình ảnh (anchor, positive, negative) để học các đặc trưng sao cho anchor và positive gần nhau hơn so với anchor và negative. Tăng cường khả năng phân biệt giữa các cá nhân khác nhau.

One-shot Learning đóng vai trò quan trọng trong việc nâng cao khả năng nhận diện khuôn mặt, đặc biệt trong các ứng dụng đòi hỏi tính linh hoạt và hiệu quả với dữ liệu hạn chế. Bằng cách áp dụng các phương pháp học tiên tiến, One-shot Learning không chỉ giúp giải quyết các bài toán thực tiễn mà còn mở ra nhiều cơ hội mới trong việc phát triển các hệ thống nhận diện thông minh và linh hoạt hơn.

## Máy Vector hỗ trợ (Support Vector Machine)

Máy vector hỗ trợ là một trong những thuật toán phân loại phổ biến và hiệu quả. Với bài toán phân loại nhị phân, giả sử có hai lớp dữ liệu được mô tả bởi các vector đặc trưng trong không gian nhiều chiều và bộ dữ liệu hai lớp này là tách biệt tuyến tính hay tồn tại một siêu phẳng có thể phân chia chính xác hai lớp đó.

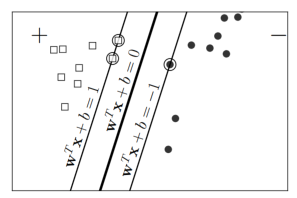


Hình 1.1. Ý tưởng của SVM [1]

Gọi khoảng cách nhỏ nhất từ một điểm thuộc một lớp tới đường phân chia là lề (tiếng anh là margin), như hình **1.1**, cần tìm đường phân chia sao cho lề của hai lớp là như nhau đối với đường phân chia đó. Độ rộng của lề càng lớn thì khả năng xảy ra phân loại lỗi càng thấp. Bài toán tối ưu SVM chính là bài toán đi tìm đường phân chia sao cho lề rộng nhất.

### Xây dựng bài toán tối ưu cho máy vector hỗ trợ

Giả sử dữ liệu trong tập huấn liệu là các cặp giá trị, gồm vector đặc trưng và nhãn, với nhãn là giá trị +1 hoặc -1 và là tổng số điểm dữ liệu. Không mất tính tổng quát, giả sử các điểm vuông có nhãn là +1 và điểm tròn có nhãn là -1, và siêu phẳng là mặt phân chia hai lớp như Hình **1.2**.



Hình 1.2. Bài toán SVM [1]

Với cặp điểm dữ liệu , khoảng cách từ điểm dữ liệu tới mặt phân cách được là , với giả sử là cùng dấu với phía của . Với mặt phân chia này lề được tính là khoảng cách gần nhất từ một điểm (trong cả hai lớp) tới mặt phân chia

|  |  |
| --- | --- |
|  | (1.1) |

***Bài toán tối ưu của SVM là đi tìm giá trị w và b sao cho lề đạt giá trị lớn nhất:***

|  |  |
| --- | --- |
|  | (1.2) |

Thay đổi giá trị trọng các vector w và b với hằng số k dương bất kỳ sẽ nhận thấy mặt phân chia không thay đổi, hay khoảng cách từ từng điểm đến mặt phân chia không đổi. Giả sử với những điểm nằm gần mặt phân chia nhất, như những điểm được khoanh tròn trong hình **1.2**. Vậy với mọi điểm n sẽ có .

Bài toán SVM có thể được đưa về dạng

|  |  |
| --- | --- |
| *Thỏa mãn*  *hay* | (1.3) |

Trong bài toán này, hàm mục tiêu là một chuẩn có dạng toàn phương. Các hàm bất phương trình rang buộc là affine. Nên bài toán là một bài toán quy hoạch toàn phương, hơn nữa hàm mục tiêu là lồi chặt vậy nên nghiệm của SVM là duy nhất.

***Xác định lớp cho một điểm dữ liệu mới***

Sau khi đã tìm được mặt phân chia , nhãn của một điểm bất kỳ sẽ được xác định bằng

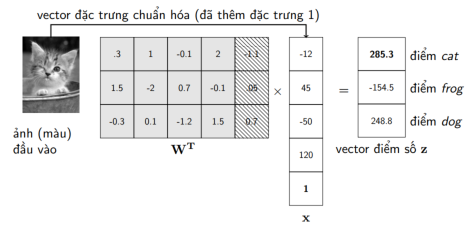
|  |  |
| --- | --- |
|  | (1.4) |

### Máy vector hỗ trợ đa lớp

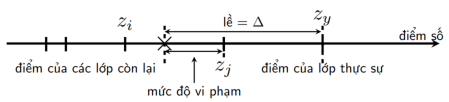
Mô hình SVM đa lớp (tiếng anh là multi-class SVM) sẽ tìm giá trị trọng số W và vector điều chỉnh b sao cho thành phần cao nhất của vector điểm số nằm tại vị trí tương ứng với nhãn của dữ liệu đầu vào.

***Xây dựng hàm mất mát***

Trong SVM đa lớp, nhãn của một điểm dữ liệu mới được xác định bởi thành phần có giá trị cao nhất trong vector điểm số , như hình 1.3. Hàm mất mát của bài toán SVM đa lớp được xây dựng theo cơ chế lề an toàn. Cụ thể, SVM đa lớp ép thành phần ứng với nhãn của vector điểm số lớn hơn các phần tử khác một đại lượng như trong hình 1.4. Đại lượng được gọi là vùng an toàn.



Hình 1.3. Ví dụ về cách tính vector điểm số [1]



Hình 1.4. Mô tả mất mát bản lề tổng quan [1]

Để mô tả mức vi phạm, giả sử điểm dữ liệu x được xét có nhãn y và vector điểm số là , vậy điểm của nhãn thực là , và điểm số của các nhãn khác là . Như hình 1.4, các điểm nằm trong vùng an toàn như sẽ có giá trị của hàm mất mát bằng 0. Với các điểm số không nằm trong vùng an toàn và vượt quá vùng an toàn một giá trị ký hiệu là , mất mát sẽ được tình bằng khoảng cách từ điểm đó tới

Giá trị mất mát của một điểm số sẽ được tính bằng công thức

|  |  |
| --- | --- |
|  | (1.5) |

Trong đó là cột thứ j của ma trận trọng số W. Với một điểm dữ liệu

bất kỳ có nhãn tương ứng là , giá trị mất mất sẽ được tính như sau:

Với là vector điểm số tương ứng với . Mất mát trên toàn bộ tập dữ liệu được định nghĩa là

|  |  |
| --- | --- |
|  | (1.6) |

## Mạng Nơ-ron tích chập

### Lớp tích chập

Trong các mạng nơ rơn tích chập (tiếng Anh là Convolution Neural Network), lớp tích chập là một phép toán tuyến tính liên quan đến phép nhân một tập hợp trọng số với đầu vào.

Lớp tích chập bao gồm một tập các bộ lọc có thể huấn luyện, các bộ lọc có kích thước tương đối nhỏ về mặt không gian (theo chiều rộng và chiều cao) nhưng được mở rộng theo toàn chiều sâu của đầu vào. Trong quá trình forward, các bộ lọc sẽ được trượt theo chiều ngang và chiều cao của đầu vào và phép tính tích vô hướng được sử dụng để đưa ra giá trị giữa các thành phần của bộ lọc với đầu vào tại mỗi vị trí. Khi bộ lọc được dịch qua toàn bộ chiều ngang và chiều cao của đầu vào sẽ tạo ra một bản đồ đặc trưng (tiếng anh là activation map) hai chiều là đáp ứng của bộ lọc tại mỗi điểm. Một cách trực quan, mô hình sẽ học cách kích hoạt các bộ lọc khi mà nó thấy các biểu diễn đặc trưng như cạnh của một đối tượng hay một vệt màu sắc nào đó tại lớp đầu tiên, hay là các biểu diễn phức tạp hơn như mặt người hay bánh xe tại các lớp sâu hơn của mô hình. Mỗi bộ lọc trong một lớp tích chập sẽ tạo ra một bản đồ đặc trưng hai chiều, sau đó chúng được xếp lại với nhau theo chiều sâu và đưa ra đầu ra.

A diagram of a machine

Description automatically generated

Hình 1.5. Ví dụ về cách tính tích chập [2]

***Kết nối cục bộ:***

Khi phải xử lý một đầu vào nhiều chiều như ảnh, việc kết nối các nơ ron tới toàn bộ các nơ ron ở tầng phía trước là không thực tế và tốn tài nguyên, thay vào đó mỗi nơ ron sẽ chỉ kết nối tới một vùng cục bộ của đầu vào. Phạm vi không gian của kết nối này là một siêu tham số và được gọi là vùng tiếp nhận (tên tiếng Anh là receptive field) (kích thước của vùng này tương đương với kích thước của bộ lọc). Chiều sâu của vùng tiếp nhận luôn bằng với chiều sâu của đầu vào.

***Sự sắp xếp về mặt không gian:***

Sẽ có ba tham số kiểm soát kích thước của đầu ra đó là: chiều sâu, bước nhảy (tiếng Anh là stride) và bù không (tiếng Anh là zero padding).

* Chiều sâu của đầu ra là một siêu tham số, nó tương ứng với số lượng bộ lọc được sử dụng, mỗi bộ lọc sẽ học biểu diễn một phần khác nhau của đầu vào.
* Bước nhảy được định nghĩa khi dịch bộ lọc. Khi bước nhảy bằng một, bộ lọc sẽ được dịch một pixel mỗi lần. Bước nhảy lớn hơn, bộ lọc sẽ được dịch một khoảng xa hơn mỗi lần và đưa ra đầu ra có kích thước nhỏ hơn.
* Đôi khi sẽ vô cùng thuận tiện khi đệm giá trị không vào biên của đầu vào. Kích thước của zero-padding là một siêu tham số giúp kiểm soát được kích thước không gian của đầu ra.

Kích thước của đầu ra khi đi qua lớp tích chập là một hàm của kích thước đầu vào (W), kích thước vùng tiếp nhận (F), bước nhảy được sử dụng (S), và số lượng giá trị không (P) được thêm tại biên. Công thức cho kích thước của đầu ra được tính như sau:

***Chia sẻ tham số:***

Cơ chế chia sẻ tham số (tiếng Anh là Parameter Sharing) được sử dụng để kiểu soát số lượng tham số của mô hình. Giả sử đầu vào có kích thước là 227x227x3, một bộ lọc với kích thước 11x11x3 với bước nhảy bằng 4, và đầu ra của lớp tích chập này có chiều sâu mong muốn là 96. Qua bộ lọc đầu tiên sẽ đưa ra đầu ra có kích thước là 55x55x96 sẽ bao gồm 290400 nơ ron và mỗi nơ ron này sẽ được kết nối với một bộ lọc 11x11x3 có tổng số trọng số là 363 cùng với 1 bias. Nếu các lớp tích chập có số tham số khác nhau thì lớp tích chập đầu tiên sẽ bao gồm 290400\*364 = 105705600 tham số. Đây là một con số rất lớn.

Số tham số này có thể được giảm một cách đáng kể bằng việc giả sử: Nếu một bộ lọc có thể xác định một đặc trưng hữu ích tại một vị trí thì nó cũng có thể tìm được đặc trưng đó tại một vị trí khác. Một cách nói khác, nếu gọi một lát cắt hai chiều theo chiều sâu là một “depth slice” (ví dụ một volume có kích thước là 55x55x96 sẽ bao gồm 96 depth slice với kích thước là 55x55), mỗi depth slice sẽ bị ràng buộc sử dụng bộ lọc có cùng giá trị trọng số và bias.

A grid of squares with different colors

Description automatically generated

Hình 1.6. Ví dụ về các bộ lọc được học bởi mô hình của Krizhevsky [3]

Chú ý rằng đôi khi việc việc giả sử các bộ lọc có tham số được chia sẻ với nhau là không hợp lý. Đặc biệt khi đầu vào của các mạng tích chập có cấu trúc đặc biệt, ví dụ một đặc trưng cụ thể nào đó chỉ nên được học ở một phía của ảnh chứ không phải là ở phía còn lại, có thể mong đợi rằng các đặc điểm cụ thể về mắt hay về tóc khác nhau nên được học ở các vị trí không gian khác nhau.

### Lớp tổng hợp

Tác dụng của lớp tổng hợp (tiếng Anh là Pooling layer) là giảm kích thước không gian của biểu diễn đặc trưng đầu vào giúp giảm số lượng tham số và số lượng tính toán của mô hình, và cũng là một phương pháp để kiểm soát vấn đề quá khớp. Lớp pooling hoạt động độc lập trên từng depth slice của đầu vào và thay đổi kích thước không gian của nó sử dụng một biển thức tính toán xác định, chiều sâu của đầu vào sẽ không bị ảnh hưởng bởi lớp pooling. Một lớp pooling bao gồm:

* Đầu vào với kích thước W1 x H1 x D1
* Yêu cầu 2 tham số: Kích thước bộ lọc F, bước nhảy S
* Đầu ra với kích thước W2 x H2 x D2:
  + W2 = (W1-F)/S + 1
  + H2 = (H2-F)/S + 1
  + D2 = D1
* Lớp pooling không sử dụng zero padding
* Lớp pooling không làm tăng thêm số lượng tham số của mô hình vì nó là một hàm cố định

Các hàm thực hiện cho lớp pooling bao gồm: max pooling, average pooling và có thể là L2-pooling.

A diagram of a single depth slice

Description automatically generated

Hình 1.7. Ví dụ về cách hoạt động của Max Pooling Layer [3]

Gần đây những mô hình học sâu thường không sử dụng lớp Pooling, kiến trúc của các mô hình bao gồm những lớp tích chập. Để giảm kích thước của biển diễn đầu vào, lớp tích chập với bước nhảy lớn được sử dụng. Mặc dù lớp pooling đã được chứng minh là quan trọng cho các mô hình sinh (tiếng Anh là generative model) ví dụ như VAEs hay GANs, nhưng những kiến trúc của các mô hình học sâu trong tương lai sẽ ít hoặc không còn sử dụng lớp pooling.

### Lớp kết nối đầy đủ

Lớp kết nối đầy đủ (tiếng Anh là Fully connected layer) khi nó kết nối tới tất các các nơ-ron từ lớp ngay phía trước. Giá trị từng nơ-ron của lớp được tính bằng phép nhân ma trận cộng với một giá trị bias.

***Chuyển đổi lớp kết nối đầy đủ thành lớp tích chập:***

Sự khác biệt duy nhất giữa lớp tích chập và lớp kết nối đầy đủ là các nơ-ron của lớp tích chập được kết nối tới một vùng cục bộ ở đầu vào, và rất nhiều nơ-ron ở cùng một lớp tích chập có chung giá trị tham số. Nhưng phương pháp tính toán ở cả hai lớp đều là phép nhân vô hướng, do đó hoàn toàn có thể chuyển đổi giữa hai lớp:

* Với bất cứ lớp tích chập nào, sẽ có một lớp kết nối đầy đủ có thể thực hiện chức năng tương tự. Ma trận trọng số với kích thước lớn với giá trị bằng 0 tại hầu hết các vị trí ngoại trừ tại các khối nhất định trong đó trọng số trong nhiều khối có giá trị bằng nhau.
* Ngược lại, các lớp kết nối đầy đủ có thể được chuyển thành lớp tích chập. Ví dụ, một lớp kết nối đầy đủ với K=4096 node với đầu vào với kích thước 7x7x512 có thể được biểu diễn tương ứng bằng một lớp tích chập với F=7, P=0, S=1, K=4096. Kích thước bộ lọc được thiết kế bằng với kích thước của đầu vào và do đó đầu ra sẽ có kích thước 1x1x4096, trùng với đầu ra của lớp kết nối đầy đủ.

Trong hai cách chuyển đổi, việc chuyển đổi lớp kết nối đầy đủ thành một lớp tích chập rất hiệu quả trong thực tế. Ví dụ, một mô hình mạng nơ ron tích chập nhận đầu vào là một ảnh với kích thước 224x224x3, qua một chuỗi các lớp tích chập và lớp pooling để đưa ra một bản đồ đặc trưng với kích thước 7x7x512. Sau đó, sử dụng hai lớp kết nối đầy đủ với kích thước 4096 và cuối cùng là một lớp kết nối đầy đủ với 1000 nơ-ron để tính toán xác suất của các lớp. Các lớp kết nối đầy đủ có thể được chuyển thành các lớp tích chập như sau:

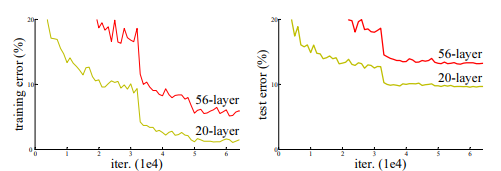
* Thay thế lớp kết nối đầy đủ đầu tiên nhìn vào ma trận đặc trưng với kích thước 7x7x512 bằng một lớp tích chập với kích thước bộ lọc F=7, đưa ra đầu ra với kích thước [1x1x4096].
* Thay thể lớp kết nối đầy đủ thứ 2 bằng lớp tích chập với kích thước bộ lọc F=1, cho ra đầu ra [1x1x4096].
* Thay thế lớp kết nối đầy đủ cuối cùng với bộ lọc có kích thước F=1, đưa ra đầu ra với kích thước [1x1x1000].

Trong thực tế, mỗi chuyển đối liên quan đến thao tác (ví dụ reshape) với các ma trận trọng số W của mỗi lớp kết nối đầy đủ biến đổi thành bộ lọc của lớp tích chập.

## Mô hình ResNet

ResNet (Residual Network) được giới thiệu đến công chúng vào năm 2015 và thậm chí đã giành được vị trí thứ 1 trong cuộc thi ILSVRC 2015 với tỉ lệ lỗi top 5 chỉ 3.57%. Không những thế nó còn đứng vị trí đầu tiên trong cuộc thi ILSVRC and COCO 2015 với ImageNet Detection, ImageNet localization, Coco detection và Coco segmentation. Hiện tại thì có rất nhiều biến thể của kiến trúc ResNet với số lớp khác nhau như ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152,...Với tên là ResNet theo sau là một số chỉ kiến trúc ResNet với số lớp nhất định.

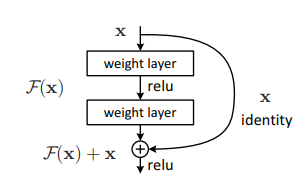
Các mô hình học sâu ngày càng trở nên sâu hơn và khó hơn trong quá trình huấn luyện. Với những mô hình với số lượng lớp (tiếng anh là layers) nhiều, khi mô hình dần đạt trạng thái bão hòa một vấn đề xuất hiện đó là độ chính xác của mô hình bão hòa sau đó suy giảm một cách nhanh chóng. Theo thực nghiệm, vấn đề suy giảm về độ chính xác không phải do quá khớp mà do việc thêm các lớp vào mô hình học sâu dẫn đến hiện tượng tắt đạo hàm (tiếng anh là Gradient Vanishing) là việc đạo hàm trở nên quá nhỏ sau quá trình lan truyền đạo hàm ngược (tiếng anh là Gradient Back-Propagation) làm cho các lớp thấp trong mô hình sẽ không còn được cập nhập trọng số, điều này gây ra vấn đề lỗi trong quá trình huấn luyện (tiếng anh là training error) tăng cao, như hình 1.6.



Hình 1.8. Giá trị lỗi trong quá trình huấn luyện và kiểm thử trên tập dữ liệu CIFAR-10 với mô hình học sâu 20 lớp và 56 lớp [4]

***Kiến trúc mạng ResNet***

Để giải quyết vấn đề này, mô hình học sâu dư (tiếng anh là deep residual network) hay ResNet đã được giới thiệu. ResNet sử dụng một khối kiến trúc kết nối tắt xuyên suốt mô hình



Hình 1.9. Khối Residual [4]

Khối Residual, như hình 1.9 sẽ bổ sung đầu vào X vào đầu ra của lớp, hay chính là phép cộng như trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm giá trị của X.

## Hàm mất mát CrossEntropy

Trong các bài toán phân loại hàm sigmoid thường được sử dụng cho phân loại nhị phân và hàm softmax cho phân loại đa nhãn để có thể tính toán xác xuất của một mẫu thuộc về lớp đối tượng nào. Hàm mất mát được sử dụng trong cả hai trường hợp phân loại nhị phân và đa nhãn là hàm mất mát cross entropy.

Biểu diễn toán học của hàm cross entropy rời rạc là:

|  |  |
| --- | --- |
|  | (1.7) |

Với n là số lượng thùng (tiếng Anh là bin) phân phối rời rạc có thể, là xác suất để thuộc vào thùng thứ i của phân phối và là xác suất để thuộc vào thùng thứ i của phân phối .

Khi được sử dụng làm hàm mất mát trong các bài toán phân loại thường là nhãn thật của một mẫu còn là giá trị xác suất mà mẫu được phân loại vào đúng nhãn i được tính bởi mô hình học sâu.

Trong trường hợp phân loại nhị phân, giá trị hàm mất mát tính trên một mẫu là:

|  |  |
| --- | --- |
|  | (1.8) |

với p là xác suất được tính bởi mô hình học sâu và là nhãn thật của mẫu.

Một cách đơn giản hơn

|  |  |
| --- | --- |
|  | (1.9) |

Cross-Entropy Loss càng nhỏ khi mô hình dự đoán gần đúng với nhãn thực tế. Khi mô hình dự đoán đúng (xác suất của lớp đúng tiến tới 1), giá trị mất mát sẽ tiến gần đến 0. Ngược lại, khi mô hình dự đoán sai, giá trị của hàm mất mát sẽ tăng cao. Vì vậy, Cross-Entropy Loss giúp điều chỉnh mô hình để cải thiện xác suất dự đoán đúng.

## Hàm mất mát Hingle

Hàm mất mát Hinge, hay Hinge Loss, là một hàm mất mát thường được sử dụng trong các bài toán phân loại nhị phân, đặc biệt là trong các thuật toán máy vector hỗ trợ (Support Vector Machine - SVM). Hinge Loss đóng vai trò quan trọng trong việc đo lường mức độ sai lệch giữa các dự đoán của mô hình và các nhãn thực tế, từ đó giúp mô hình tối ưu hóa các tham số để giảm thiểu sai số.

Hàm mất mát Hinge được định nghĩa dựa trên việc tối đa hóa "khoảng cách biên" (margin) giữa các điểm dữ liệu của các lớp khác nhau, nhằm đảm bảo rằng các điểm dữ liệu được phân loại đúng và cách biệt một khoảng nhất định so với đường phân chia. Cụ thể, với một mẫu đầu vào (x,y) có nhãn y là +1 hoặc −1, hàm mất mát Hinge có dạng:

|  |  |
| --- | --- |
|  | (1.10) |

Trong đó:

là là kết quả dự đoán của mô hình cho mẫu x.

đại diện cho tích giữa nhãn thực tế y và dự đoán của mô hình . Nếu , thì mô hình được coi là phân loại đúng với một "khoảng cách an toàn" từ ranh giới quyết định.

Hàm mất mát Hinge được thiết kế để đạt giá trị bằng 0 khi dự đoán của mô hình cách biệt với ranh giới quyết định một khoảng ít nhất là 1. Điều này có nghĩa là mô hình không chỉ cần dự đoán đúng mà còn phải phân loại đủ cách biệt để tối đa hóa độ chính xác của việc phân loại. Khi, hàm mất mát Hinge sẽ tăng dần theo độ lệch của dự đoán, yêu cầu mô hình điều chỉnh để giảm thiểu sai số này.

Hàm mất mát Hinge là cốt lõi trong các mô hình SVM, giúp tối ưu hóa khoảng cách biên giữa các lớp và tăng cường khả năng phân loại chính xác. Ngoài ra, Hinge Loss cũng được điều chỉnh và áp dụng trong các biến thể khác như **Soft-margin SVM**, **Multi-class SVM**, và thậm chí trong một số ứng dụng học sâu với mục tiêu phân loại nhị phân mạnh mẽ hơn.

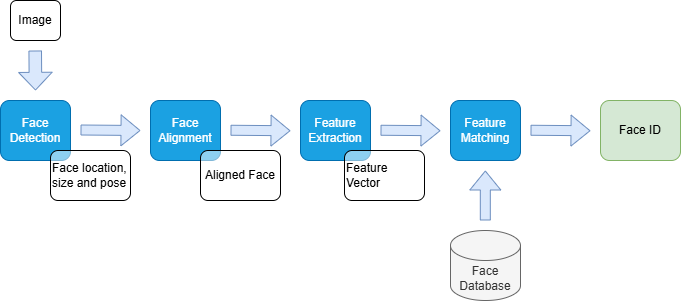
Nhờ những đặc điểm và khả năng tối ưu hóa mạnh mẽ, Hinge Loss đóng vai trò quan trọng trong các hệ thống phân loại nhị phân hiện đại, đặc biệt trong các lĩnh vực như nhận diện đối tượng, nhận diện khuôn mặt, và phân loại văn bản.

## Kết luận chương

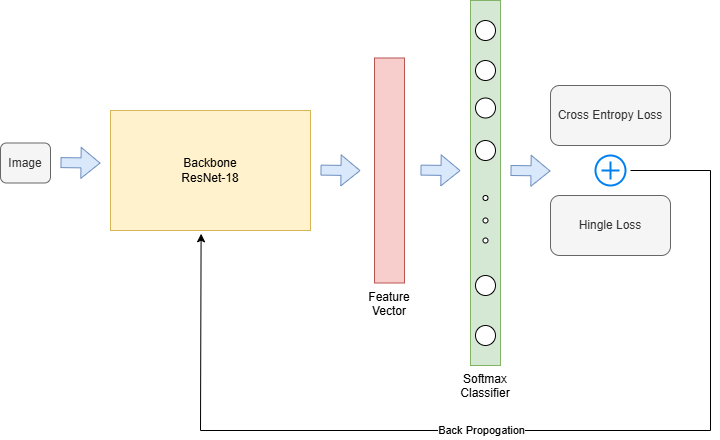
Chương 1 đã trình bày khái quát về Bài toán nhận diện khuôn mặt, tổng quan kiến thức về máy vector hỗ trợ, tóm tắt các kiến thức về mạng nơ-ron tích chập và một mô hình cụ thể là ResNet và lý thuyết về các hàm mất mát CrossEntropy và Hingle.

# PHƯƠNG PHÁP ĐỀ XUẤT

Mô hình đề xuất là một mạng nơ-ron tích chập được mở rộng từ mô hình ArcFace, mô hình sẽ trích chọn véc tơ đặc trưng của các ảnh khuôn mặt sau đó so sánh sự tương đồng giữa các véc tơ đặc trưng đó. Hàm mất mát Hingle được thêm vào cùng với những hàm mất mát được sử dụng trong mô hình ArcFace tại quá trình huấn luyện với mục đích giảm thời gian huấn luyến cũng như tăng mức độ phân tách giữa các nhãn.



Hình 2.1. Pipeline của hệ thống nhận diện khuôn mặt



Hình 2.2. Pha huấn luyện của mô hinh đề xuất

Trong quá trình thực hiện nhận diện khuôn mặt như hình 2.1, hệ thống sẽ nhận một ảnh đầu vào chứa khuôn mặt, sau đó sẽ thực hiện các bước phát hiện khuôn mặt, căn chỉnh lại hướng khuôn mặt, sau đó đưa vào một mô hình học sâu để trích chọn vec tơ đặc trưng khuôn mặt, vec tơ đặc trưng khuôn mặt này sẽ được so sánh với những vec tơ đặc trưng có sẵn trong cơ sở dữ liệu để đưa ra nhãn cho khuôn mặt trong ảnh đầu vào.

Đồ án này sẽ tập trung vào bước trích chọn đặc trưng khuôn mặt bằng mô hình học sâu có sử dụng SVM.

## Các nghiên cứu liên quan

### Mô hình ArcFace

Các mô hình tích chập học sâu (tiếng anh là Deep Convolution Neural Network) đã trở thành lựa chọn phổ biến trong việc trích xuất các đặc trưng khuôn mặt và đã chứng tỏ ưu thế vượt trội trong nhiệm vụ này. Tuy nhiên, việc quan trọng tiếp theo là sử dụng các vector đặc trưng để phân loại hiệu quả, nhằm cải thiện độ chính xác của hệ thống nhận diện khuôn mặt. Có hai hướng tiếp cận chính trong việc này: một số người ưu tiên xây dựng thêm một mô hình phân loại, chẳng hạn sử dụng hàm **softmax**, trong khi một số khác lựa chọn huấn luyện trực tiếp với các vector đặc trưng thông qua các hàm mất mát, ví dụ như **triplet loss**. Cả hai phương pháp này đều mang lại kết quả tốt, nhưng vẫn tồn tại một số hạn chế đáng chú ý.

Việc sử dụng **softmax** làm tăng kích thước ma trận biến đổi tuyến tính tỉ lệ thuận với số lượng danh tính cần phân loại, và phương pháp này thường chỉ hoạt động hiệu quả trong các bài toán phân loại kín, nơi số lượng lớp (tiếng anh là class) là cố định. Điều này không thực tế khi số lượng khuôn mặt cần nhận diện liên tục thay đổi. Trong khi đó, **triplet loss** khắc phục được vấn đề này nhưng lại có nhược điểm khác. Vì **triplet loss** dựa trên việc so sánh ba mẫu (anchor, positive, negative) cùng lúc, nên khi số lượng dữ liệu tăng, số lượng bộ ba cần xử lý cũng tăng theo cấp số nhân, khiến cho thời gian huấn luyện trở nên tốn kém. Hơn nữa, việc áp dụng **semi-hard sample training**, được coi là tối ưu cho triplet loss, cũng gặp khó khăn trong quá trình huấn luyện hiệu quả.

Chính vì những hạn chế này mà các nhà nghiên cứu đã đề xuất một hướng đi mới cho nhận diện khuôn mặt, với việc giới thiệu hàm mất mát **Additive Angular Margin Loss**, nhằm khắc phục những vấn đề trên.

#### Hàm mất mát Softmax

Hàm mất mát thường xuyên được sử dụng trong các bài toán phân loại đa lớp là hàm mất mát Softmax được thể hiện bởi công thức

|  |  |
| --- | --- |
|  | (2.1) |

Với là véc tơ đặc trưng của mẫu thứ i, với nhãn tương ứng là . là ma trận trọng số và là vecto thiên khiến (tiếng anh là bias) và N là số lượng nhãn. Hàm này thường được sử dụng cho các bài toán nhận diện khuôn mặt. Tuy nhiên, điểm yếu của hàm này đó là không có khả năng tối ưu hóa rõ ràng các véc tơ embedding chứa các đặc điểm của khuôn mặt để tăng thêm sự tương đồng giữa những khuôn mặt trong một lớp và tăng thêm sự đa dạng giữa các khuôn mặt giữa các lớp, dẫn đến tạo ra một khoảng cách cho những khuôn mặt với nhiều biến thể (do khác biệt tuổi tác hay do khác biệt về dáng khuôn mặt).

#### Hàm mất mát Additive Angular Margin

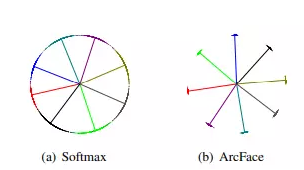
Để đơn giản hóa, véc tơ thiên khiến được cố định bằng 0. Biến đổi với là góc giữa 2 véc tơ và . Sau đó giá trị và được chuẩn hóa về 1 bằng phương pháp chuẩn hóa và được chia lại tỷ lệ về giá trị s. Bước chuẩn hóa weights và véc tơ đặc điểm này tạo nên sự dự đoán thuần túy dựa trên góc giữa véc tơ đặc điểm và weight. Véc tơ đặc điểm đã được học sau đó được phân bổ trên một siêu cầu (tiếng anh là hypersphere) với bán kính là *s*.

|  |  |
| --- | --- |
|  | (2.2) |

Vì các véc tơ đặc điểm được phân bổ xung quanh mỗi đặc điểm trung tâm theo một siêu cầu, một hình phạt biên góc cộng (additive angular margin penalty) *m* sẽ được thêm vào giữa ​ và  ​​​ để đồng thời tăng cường tính nhỏ gọn trong nội bộ lớp và sự khác biệt giữa các lớp. Và phương pháp này được đặt tên là *ArcFace*.

|  |  |
| --- | --- |
|  | (2.3) |

Ví dụ, với 8 khuôn mặt được chọn và mỗi khuôn mặt sẽ bao gồm 1500 ảnh để huấn luyện cho một mô hình học sâu sử dụng hàm mất mát Softmax và Arcface



Hình 2.3. Ví dụ đơn giản cho việc sử dụng 2 hàm mất mát để phân loại 8 nhãn [5]

Theo như hình, ta có thể thấy việc sử dụng softmax với hàm mất mát entropy chéo giúp chúng ta phân tách các véc tơ đặc điểm tuy nhiên lại tạo nên 1 sự mơ hồ rõ ràng trên đường ranh giới quyết định (decision boudaries). Trong khi đó, sử dụng hàm mất mát ArcFace lại có sự phân chia rõ rệt giữa các lớp gần nhau.

Mô hình ArcFace đạt được kết quả nhận diện với độ chính xác cao. Và một ưu điểm hơn cả khi so sánh với triplet loss của Facenet là hàm mất mát ArcFace dễ triến khai trên code hơn, có thể tùy chỉnh hàm để kết hợp cùng với rất nhiều các mô hình CNN hiện có để tạo ra một mô hình nhận diện khuôn mặt phù hợp nhất với yêu cầu cá nhân. Do đó đồ án này lựa chọn phát triển mô hình ArcFace để kết hợp với hàm mất mát Hinge cho đồ án “”.

## Mô hình đề xuất

Mô hình đề xuất được giới thiệu như hình 2.2 bao gồm một mô hình trích chọn đặc trưng ở đây được sử dụng là mô hình Resnet-18 (mục) và một đầu ra phân loại.

Mô hình đầu tiên sẽ được huấn luyện để phục vụ mục đích phân loại. Sau quá trình huấn luyện đầu ra phân loại sẽ được loại bỏ chỉ giữ lại backbone trích chọn đặc trưng. Các ảnh đi qua backbone đưa ra các vec tơ đặc trưng, sử dụng các vec tơ đặc trưng này để so sánh mức độ tương đồng giữa các ảnh từ đó đưa ra nhãn chính xác của ảnh khuôn mặt đầu vào.

***Các hàm mất mát sử dụng***

Trong đồ án lần này, các hàm mất mát CrossEntropy và Hingle sẽ lần lượt được sử dụng để đánh giá hiệu quả với bài toán nhận diện khuôn mặt. Sau đó, đánh giá mô hình sử dụng cả hai hàm mất mát trên.

## Kết luận chương

# THỰC NGHIỆM VÀ KẾT QUẢ

## Cở sở dữ liệu

### Bộ dữ liệu CasiaWebFace

Casia-WebFace là một bộ dữ liệu khuôn mặt quy mô lớn được phát triển bởi Viện Tự động hóa Trung Quốc thuộc Học viện Khoa học Trung Quốc (CASIA). Bộ dữ liệu này được thiết kế dành riêng cho các bài toán về nhận diện khuôn mặt, với mục đích cung cấp một nguồn dữ liệu chất lượng, đa dạng và phù hợp cho việc huấn luyện các mô hình học sâu, đặc biệt là mạng nơ-ron tích chập sâu (Deep Convolutional Neural Networks - DCNN). Casia-WebFace đã trở thành một nguồn dữ liệu phổ biến trong nghiên cứu và phát triển các hệ thống nhận diện khuôn mặt hiện đại.

Bộ dữ liệu có kích thước tầm 4.1G, bao gồm 494,414 hình khuôn mặt của 10,575 người thật được thu thập trên web và đã gán nhãn đầy đủ.

Các hình ảnh trong bộ dữ liệu này được thu thập từ internet, đảm bảo sự đa dạng về độ tuổi, giới tính, sắc tộc và các yếu tố khác như biểu cảm khuôn mặt, góc chụp, điều kiện ánh sáng và nền phông.



Hình 3.1 Một vài ví dụ trong tập dữ liệu CasiaWebFace

Với số lượng lớn hình ảnh và danh tính, Casia-WebFace cung cấp môi trường phong phú để mô hình học được các đặc trưng đa dạng từ nhiều kiểu khuôn mặt. Bộ dữ liệu hỗ trợ các bài toán phức tạp về nhận diện khuôn mặt trong môi trường thực tế với các điều kiện biến đổi. Casia-WebFace đã được trích dẫn và sử dụng rộng rãi trong cộng đồng nghiên cứu về nhận diện khuôn mặt, trở thành một bộ dữ liệu chuẩn trong các nghiên cứu học sâu.

Hạn chế của bộ dữ liệu ở chất lượng hình ảnh cũng là do được thu thập từ internet, một số hình ảnh có thể bị ảnh hưởng bởi chất lượng thấp hoặc nhiễu, điều này có thể làm giảm hiệu suất của các mô hình. Casia-WebFace cũng tập trung nhiều vào khuôn mặt của những người nổi tiếng, do đó không hoàn toàn phản ánh tính ngẫu nhiên và đa dạng của khuôn mặt trong dân số chung.

### Bộ dữ liệu Labelled Faces in the Wild (LFW)

Labelled Faces in the Wild (LFW) là một bộ dữ liệu hình ảnh khuôn mặt có chú thích nổi tiếng trong lĩnh vực nhận diện khuôn mặt, được giới thiệu vào năm 2007 bởi Đại học Massachusetts Amherst. LFW là một trong những bộ dữ liệu đầu tiên được thiết kế đặc biệt cho các bài toán nhận diện khuôn mặt trong điều kiện thực tế, nơi có nhiều yếu tố biến đổi như góc chụp, điều kiện ánh sáng, biểu cảm khuôn mặt và môi trường nền khác nhau. Bộ dữ liệu này đã trở thành một tiêu chuẩn trong việc đánh giá và so sánh các mô hình nhận diện khuôn mặt, từ đó thúc đẩy các tiến bộ trong lĩnh vực này.



Hình 3.2. Ví dụ về các ảnh trong bộ dữ liệu LFW

Bộ dữ liệu LFW chứa 13,233 ảnh khuôn mặt. Bao gồm hình ảnh của 5,749 cá nhân khác nhau, trong đó có nhiều hình ảnh của cùng một người từ các thời điểm khác nhau. Bộ dữ liệu có độ biến thiên cao về góc chụp, biểu cảm, độ phân giải và điều kiện ánh sáng, giúp các mô hình học được cách xử lý trong các tình huống thực tế.

Mặc dù LFW bao gồm một số lượng lớn danh tính, sự phân bổ sắc tộc trong bộ dữ liệu không cân đối, gây ảnh hưởng đến tính tổng quát của mô hình. Một số hình ảnh có chất lượng thấp hoặc bị mờ, điều này có thể ảnh hưởng đến hiệu suất của các mô hình nhận diện khuôn mặt. So với các bộ dữ liệu hiện đại với hàng triệu ảnh, quy mô của LFW có thể hạn chế khả năng tổng quát hóa cho các mô hình học sâu lớn hơn.

## Quá trình thực nghiệm

Toàn bộ quá trình huấn luyện và thử nghiệm trong đồ án lần này đều được thực hiện trên nền tảng Google Colab với cấu hình như sau:

Bảng 3.6. Bảng cấu hình Google Colab

|  |  |
| --- | --- |
| CPU | AMD EPYC 7B12 @ 2.25GHz |
| RAM | 13Gb DDR4 |
| GPU | NVIDIA T4 16GB GDDR6 PCIe 3.0 |
| OS | Ubuntu 22.04.2 LTS 64bit |

### Huấn luyện mô hình

Mô hình lần lượt được huấn luyện sử dụng hàm mất mát Hingle, sau đó là huấn luyện mô hình sử dụng kết hợp hàm mất mát Hingle và hàm mất mát CrossEntropy. Mô hình được huấn luyện trên tập dữ liệu CasiaWebFace, sau đó được đánh giá trên tập dữ liệu LFW.

Bảng 3.7. Bảng thông số huấn luyện trên bộ dữ liệu CasiaWebFace

|  |  |
| --- | --- |
| Bộ dữ liệu sử dụng | CasiaWebFace |
| Epochs | 25 |
| Kích thước ảnh đầu vào | 128x128 |
| Batch size | 16 |
| lr | 1e-1 |
| lr\_scheduler (giảm 10) | 10, 20 |
| Optimizer | SGD |