TRƯỜNG ĐẠI HỌC THỦY LỢI



Báo cáo BỘ MÔN NHẬN DẠNG MẪU

Tên đề tài:

Xác định và nhận diện khuôn mặt trên ảnh sử dụng mô hình

RetinaFace và Vision Transformer

*GIÁO VIÊN HƯỚNG DẪN: PGS.TS Nguyễn Quan Hoan*

Lớp: 64TTNT1

|  |  |
| --- | --- |
| Họ và tên | MSV |
| Nguyễn Gia Bách | 2251262579 |

Hà Nội, Năm 2025

LỜI CẢM ƠN

Lời đầu tiên, để có được báo cáo đồ án như ngày hôm nay em xin gửi lời cảm ơn trân trọng nhất tới Nguyễn Quan Hoan, giảng viên bộ môn Nhận Dạng Mẫu tại Trường đại học Thuỷ lợi. Suốt thời gian qua thầy đã hỗ trợ nhiệt tình cho em, sự hỗ trợ và động viên quý báu của thầy trong cả quá trình làm bài tập lớn giúp em hoàn thiện bại tập lớn của mình một cách đầy đủ và trọn vẹn nhất. Thầy đã dành thời gian và tâm huyết để hướng dẫn em trong việc tìm hiểu, phát triển ý tưởng, cũng như giúp đỡ em vượt qua những thách thức khó khăn. Sự tận tâm và kiến thức sâu rộng của thầy đã giúp em có được những kiến thức quý giá và trải nghiệm thực tế trong lĩnh vực Trí tuệ nhân tạo.

Cuối cùng, em xin cảm ơn nhà trường đại học Thuỷ Lợi đã tạo điều kiện cho em có những trải nghiệm quý báu này. Trong suốt 4 năm đại học, nhà trường luôn trao cho em rất nhiều cơ hội để học tập, nghiên cứu và sáng tạo. Cảm ơn nhà trường đã trao cho em nhiều cơ hội để cải thiện bản than trong suốt quãng đời sinh viên vừa qua.

Một lần nữa, em xin cảm ơn thầy Nguyễn Quan Hoan, bạn bè, gia đình và nhà trường đã ủng hộ và hỗ trợ em trong hành trình này. Mặc dù đã cố gắng hoàn thành bài tập lớn trong phạm vi cho phép nhưng cũng không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm, nhận xét và đóng góp ý kiến từ thầy cô.

Em xin chân thành cảm ơn !

Hà Nội, ngày 01 tháng 11 năm 2025

Sinh viên

Nguyễn Gia Bách

**Mục lục**

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN 5](#_Toc212809110)

[1.1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc212809111)

[1.2. Mô tả bài toán chi tiết 5](#_Toc212809112)

[1.2.1. Mục tiêu chính 6](#_Toc212809113)

[1.2.2. Kết quả dự kiến 6](#_Toc212809114)

[CHƯƠNG 2. DỮ LIỆU 7](#_Toc212809115)

[2.1. NGUỒN DỮ LIỆU VÀ CÁC ĐẶC ĐIỂM KHÁI QUÁT CỦA DỮ LIỆU 7](#_Toc212809116)

[2.2. TIỀN XỬ LÝ DỮ LIỆU 8](#_Toc212809136)

[2.3. TRỰC QUAN HÓA DỮ LIỆU 12](#_Toc212809137)

[CHƯƠNG 3. CÁC MÔ HÌNH ĐƯỢC SỬ DỤNG 16](#_Toc212809139)

[3.1. MÔ HÌNH RETINAFACE 16](#_Toc212809140)

[3.1.1. Nguyên tắc hoạt động 16](#_Toc212809142)

[3.1.2. Ưu điểm và nhược điểm 19](#_Toc212809143)

[3.2. MÔ HÌNH VISION TRANSFORMER 20](#_Toc212809144)

[3.2.1. Nguyên tắc hoạt động 20](#_Toc212809145)

[3.2.2. Ưu điểm và nhược điểm 22](#_Toc212809146)

[CHƯƠNG 4. ÁP DỤNG CÁC MÔ HÌNH VÀO BÀI TOÁN 24](#_Toc212809147)

[4.1. Cấu hình môi trường 24](#_Toc212809148)

[4.2. Áp dụng mô hình vào bài toán xác định khuôn mặt 24](#_Toc212809149)

[4.2.1. Kết quả thu được 26](#_Toc212809150)

[4.3. Bài toán nhận diện khuôn mặt người nổi tiếng 27](#_Toc212809151)

[4.3.1. Kết quả thu được 30](#_Toc212809152)

[4.4. Bài toán nhận diện khuôn mặt 32](#_Toc212809154)

[4.4.1. Kết quả thu được 34](#_Toc212809155)

[CHƯƠNG 5. GIAO DIỆN DEMO 36](#_Toc212809156)

[KẾT LUẬN 38](#_Toc212809157)

[TÀI LIỆU THAM KHẢO 39](#_Toc212809158)

**DANH MỤC HÌNH ẢNH**

Hình 2.1 Biểu đồ phân lớp người nổi tiếng…………………………………………………….…14

Hình 2.2 Một số các hình ảnh cùng với nhãn của bộ dữ liệu người nổi tiếng…………………….15

Hình 3.1 Minh họa cách hoạt động của RetenaFace...................................................................... 16

Hình 3.2 Minh họa cách hoạt động của RetenaFace với ảnh cụ thể.............................................. 17

Hình 3.3 Minh họa cách hoạt động của Vision Transformer......................................................... 21

Hình 4.2 Biểu đồ thể hiện trainning và validation loss…………………………………………...30

Hình 4.3 Biểu đồ thể hiện trainning và validation accuracy……………………………………...30

Hình 4.4 Bảng báo cáo đánh giá hiệu suất mô hình………………………………………………31

Hình 4.5 Ma trận nhầm lẫn của mô hình………………………………….………………………32

Hình 4.6 Giao diện của mô hình nhận diện khuôn mặt…………….……………………………..34

Hình 5.1 Trang chủ của giao diện demo chính…………………………………………………...36

Hình 5.2 Kết quả sau khi so sánh embedding 2 ảnh……………………………………….……..37

Hình 5.3 Kết quả sau khi phân loại người nổi tiếng trên demo giao diện……………………….37

# GIỚI THIỆU TỔNG QUAN

## GIỚI THIỆU ĐỀ TÀI

Trong bối cảnh công nghệ trí tuệ nhân tạo và thị giác máy tính ngày càng phát triền hiện đại và tiên tiến hơn, các bài toán phát hiện và nhận diện khuôn mặt đóng vai trò quan trọng trong nhiều lĩnh vực khác nhau như kiểm soát an ninh, thương mai điện tử, ngân hàng tài chính hay trong các lĩnh vực y tế, giáo dục và đào tạo. Công nghệ này cho phép chúng ta có thể tự động phát hiện con người trong các hình ảnh, video cũng như có thể tự động nhận diện danh tính cá nhân dựa trên các đặc trưng của khuôn mặt.

Bước đầu của mô hình này đó chính là có thể tự động xác định vị trí khuôn mặt của các cá nhân trong khung hình. Sau đó, có thể tự động lưu trữ các đặc trưng trên khuôn mặt để có thể thực hiện so sánh và xác định danh tính dựa trên các dữ liệu khuôn mặt đã được lưu trữ. Các mô hình học sâu hiện đại (deep learning) cũng như mạng nơ-ron tích chập (Convolutional Neural Network – CNN) đã góp phần vô cùng lớn trong việc hoàn thiện và tối ưu độ chính xác, tốc độ xử lý cũng như hoàn thiện mô hình này.

## Mô tả bài toán chi tiết

Xác định và nhận diện khuôn mặt là một bài toán thuộc lĩnh vực thị giác máy tính, trong đó mục tiêu là xác định vị trí khuôn mặt bằng cách tạo các khung (bounding box) quanh khuôn mặt được tìm thấy, trích xuất các đặc trưng của khuôn mặt đó để lưu trữ và so sánh đặc trưng của khuôn mặt mới với dữ liệu đã có, cuối cùng xác định danh tính của người đó hoặc kết luận “không xác định” nếu không có khớp.

Bài toán minh hoạ đi sau là bài toán phân lớp các người nổi tiếng, sử dụng mô hình xác định vị trí khuôn mặt bằng cách tạo các khung (bounding box) quanh khuôn mặt được tìm thấy và thực hiện phân loại các khuôn mặt đó thuộc về người nổi tiếng nào, sau khi phân loại sẽ hiện các nhãn bên dưới khuôn mặt đã được tìm thấy.

Đầu vào :

+ Ảnh để thực hiện lưu trữ embedding

+ Anh để so sánh với ảnh đã lưu trữ trước đó

+ Ảnh của những người nổi tiếng.

Đầu ra :

+ Ảnh đã được xác định vị trí khuôn mặt

+ Kết quả so sánh 2 ảnh có cùng là một người

+ Ảnh những người nổi tiếng đã được xác định danh tính.

### Mục tiêu chính

Tìm hiểu và phân tích các mô hình nhận dạng khuôn mặt hiện có, xây dựng được mô hình xác định và nhận dạng khuôn mặt hiệu quả, đánh giá độ hiệu quả của mô hình trên tập dữ liệu thực tế, đề xuất hướng cải tiến và ứng dụng thực tiễn.

### Kết quả dự kiến

Xây dựng được mô hình xác định và nhận diện khuôn mặt hiệu quả và chính xác, phân loại đúng các danh tính của người nổi tiếng trên ảnh dựa trên các tiêu chí đánh giá sau:

+ Độ chính xác cosin

+ Accuracy: Độ chính xác tổng thể

+ Precision: Độ chính xác theo từng lớp

+ Recall : Độ bao phủ

+ F1-score: Trung bình điều hoà của Precision và Recall

+ Confusion Matrix: Ma trận nhầm lẫn

Tạo ra một chương trình demo cho mô hình đã huấn luyện

# DỮ LIỆU

## NGUỒN DỮ LIỆU VÀ CÁC ĐẶC ĐIỂM KHÁI QUÁT CỦA DỮ LIỆU

Dữ liệu được lấy từ trang web kaggle, một trong những nền tảng hàng đầu thế giới dành cho khoa học dữ liệu và học máy. Kaggle chứa hàng trăm ngàn bộ dữ liệu công khai mà người dùng có thể sử dụng.

Link dữ liệu:

Dữ liệu để xác định ví trí khuôn mặt: https://www.kaggle.com/datasets/b2004x/wider-face-retinaface

Dữ liệu để phân lớp các người nổi tiếng: <https://www.kaggle.com/datasets/vishesh1412/celebrity-face-image-dataset>

**MÔ TẢ DỮ LIỆU**

Dữ liệu để xác định vị trí khuôn mặt :

## data/

## └── widerface/

## ├── train/

## │ ├── images/

## │ └── label.txt

## └── val/

## ├── images/

## └── wider\_val.txt

## images: folder ảnh dùng để thực hiện hiện huấn luyện và đáng giá

label.txt: file chứa vị trí khuôn mặt của tập huấn luyện

wider\_val: file chứa vị trí khuôn mặt của tập đánh giá

## Dữ liệu để phân lớp các người nổi tiếng

## data/

## └── Celebrity Faces Dataset

## ├── Angelina Jolie

## └── Brad Pitt

## ├── Denzel Washington

## └── Hugh Jackman

## ├── Jennifer Lawrence

## └── Johnny Deep

Celebrity Faces Dataset: Chứa các folder dùng để huấn luyện, đánh giá và thử nghiệm, mỗi folder là một nhãn và trong đó là các ảnh khuôn mặt của người nổi tiếng thuộc nhãn đó

## ĐẶC ĐIỂM DỮ LIỆU

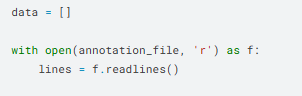
Kích thước:

Dữ liệu để xác định vị trí khuôn mặt : 1.86 GB

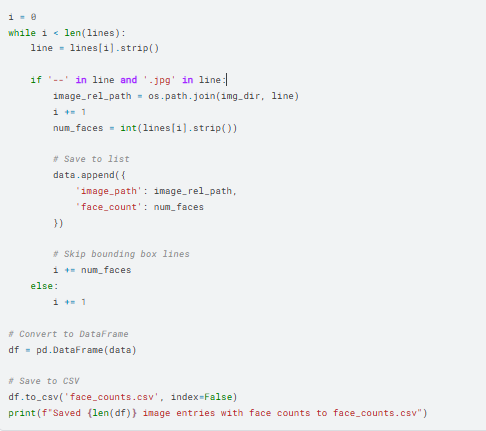
Dữ liệu để phân lớp người nổi tiếng: 55.68 MB

## TIỀN XỬ LÝ DỮ LIỆU

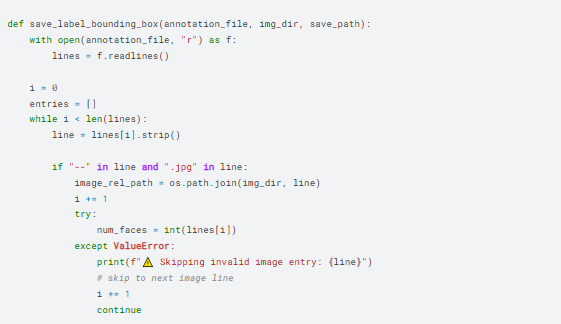
Dữ liệu để xác định vị trí khuôn mặt :

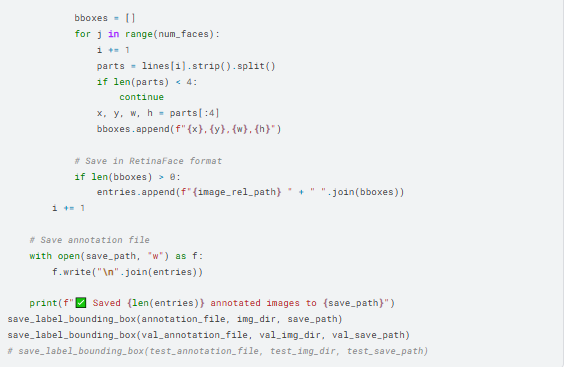


Mở annotation\_file (file chứa các nhãn dữ liệu) ở chế độ đọc, lines chứa danh sách các dòng trong nội dung file.



Đoạn mã trên có chức năng đọc file ghi chú chứa các nhãn (annotation\_file) của bộ dữ liệu khuôn mặt, nhằm trích xuất số lượng khuôn mặt mỗi ảnh và lưu thông tin này vào trong file csv. Sau khi mở file annotation và đọc toàn bộ các dòng, chương trình sẽ duyệt tuần tự từng dòng một để tìm tên ảnh (kết thúc bằng – hoặc .jpeg), sau khi tìm thấy chương trình sẽ lấy số lượng khuôn mặt của ảnh đó ở dòng tiếp theo ( i + 1 ) . Thông tin này cùng đường dẫn ảnh sẽ được lưu vào danh sách khác ( data ). Chương trình này chỉ để lấy số lượng khuôn mặt nên sẽ bỏ qua tất cả những dòng chứa vị trí khuôn mặt của ảnh (mô tả bounding box tương ứng). Sau khi xử lý toàn bộ file, dữ liệu thu thập được sẽ được chuyển dưới dạng Dataframe và lưu lại dưới file face\_count.csv. File này chứa hai cột dẫn, một là đường dẫn đến file ảnh và cột thứ hai là số lượng khuôn mặt trong file ảnh tương ứng, giúp chúng ta có thể dễ dàng phân tích và thống kê số lượng khuôn mặt trong ảnh.

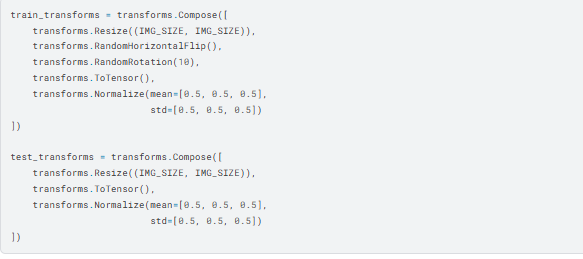




Đoạn mã trên được dùng để đọc file ghi chú chứa các nhãn (annotation) của bộ dữ liệu khuôn mặt, lấy thông tin vị trí của các khuôn mặt trong ảnh (bounding box) và lưu nó lại theo định dạng mà mô hình Retinaface có thể sử dụng. Chương trình đầu tiên duyệt tuần tự từng dòng trong file cho đến khi gặp dòng chứa tên ảnh ( kết thúc bằng “--“ hoặc “.jpeg”), ngay sau dòng chứa tên ảnh chính là dòng chứa số lượng khuôn mặt của ảnh đó, nếu giá trị không hợp lệ, chương trình sẽ ngay lập tức bỏ qua và chuyển sang ảnh tiếp theo. Khi xác định được số lượng khuôn mặt, chương trình sẽ đọc tiếp dòng chứa vị trí khuôn mặt tương ứng, dòng này chứa bốn số mô tả bounding box theo thứ tự x, y , width, height. Các thông tin hợp lệ sẽ được lưu lại dưới dạng chuỗi và cho vào danh sách tổng hợp (bboxes).

Sau khi ghi đủ thông tin bounding box cho ảnh, chương trình ghép đường dẫn ảnh và các bounding box trên cùng một dòng theo đúng format mà mô hình Retinaface yêu cầu rồi thêm vào danh sách kết quả. Cuối cùng, sau khi duyệt tất cả chương trình sẽ lưu tất cả vào file mới thông qua save\_path để tiện cho việc sử dụng hơn. Việc này được thực hiện cho cả 2 bộ dữ liệu train và validation, đảm bảo bộ dữ liệu đều được chuyển đổi sang dạng chú thích thích hợp cho mô hình.

Dữ liệu để phân lớp người nổi tiếng:



Đây là 2 bộ transform cho dữ liệu cho dữ liệu ảnh trong quá trình huấn luyện và đánh giá.

Với tập huấn luyện, chương trình sẽ chỉnh kích thước đầu vào về chuẩn 224 x 224 (IMG\_SIZE), lật ânh ngẫu nhiên theo trục ngang để tăng tính đa dạng cho dữ liệu. Xoay ảnh ngẫu nhiên 10 giúp mô hình học sự biên thiên các góc khuôn mặt. Chuyển ảnh sang dạng tensor về scale [0,1] và chuẩn hoá theo mean và std để ổn định quá trình huấn luyện.

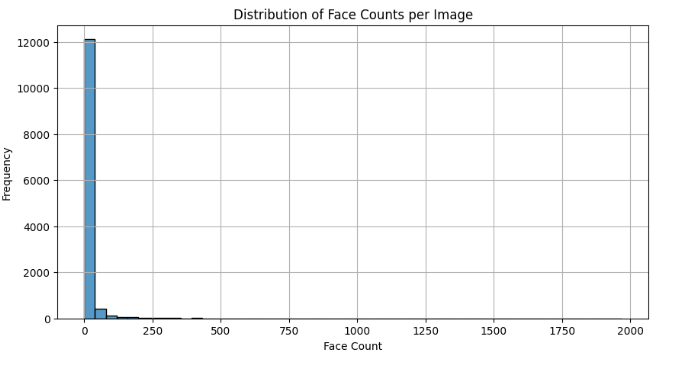
Với tập đánh giá, , chương trình sẽ chỉnh kích thước đầu vào về chuẩn 224 x 224 (IMG\_SIZE), Chuyển ảnh sang dạng tensor về scale [0,1] và chuẩn hoá theo mean và std để phù hợp với dữ liệu đầu vào của mô hình khi đánh giá.

## TRỰC QUAN HÓA DỮ LIỆU

Dữ liệu để xác định vị trí khuôn mặt :



Có tổng tất cả 12800 ảnh trong bộ dữ liệu này.



Biểu đồ thể hiện **phân bố số lượng khuôn mặt trong mỗi ảnh.** Cụ thể:

**Đa số ảnh chứa rất ít khuôn mặt** (gần như tập trung sát vùng 0–10), có một vài ảnh có số khuôn mặt rất cao và một số điểm dữ liệu nằm xa trục chính → ảnh có **hàng trăm đến hàng nghìn khuôn mặt** (có thể là ảnh đám đông lớn, hoặc dữ liệu nhiễu).

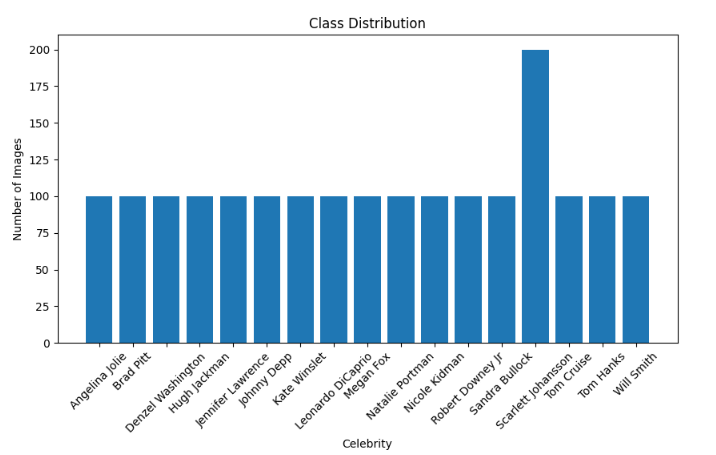
Đây là phân bố lệch mạnh sang phải với trung vị (median) rất thấp, giá trị cực trị (outlier) rất cao rất phù hợp với các dataset nhận diện khuôn mặt nơi phần lớn ảnh là chân dung / ít người

Một số các hình ảnh với nhãn trong tập dữ liệu:





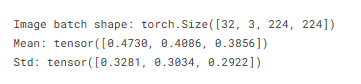
## Dữ liệu để phân lớp các người nổi tiếng:



*Hình 2.1 Biểu đồ phân lớp người nổi tiếng*

Đa số các người nổi tiếng đều có số lượng ảnh để phân lớp giống nhau (100 ảnh) -> bộ dữ liệu khá cân bằng. Lớp Sandra Bullock có đến 200 ảnh thuộc lớp đó gây mất cân bằng cục bộ.

Về tổng quan, đây là dữ liệu gần cân bằng phù hợp với các mô hình phân loại khuôn mặt.



Kích thước mỗi ảnh là 224 x 224 với batch size chia dữ liệu là 32 ảnh mỗi batch. **Mean gần giống ImageNet**, nhưng **Std lớn hơn**, cho thấy dataset đa dạng ánh sáng, màu sắc (bình thường với ảnh người thật).

Có tổng tất cả 1800 ảnh trong bộ dữ liệu này.

Một số các ảnh với nhãn ở trong tập dữ liệu:



*Hình 2.2 Một số các hình ảnh cùng với nhãn của bộ dữ liệu người nổi tiếng*

# CÁC MÔ HÌNH ĐƯỢC SỬ DỤNG

## MÔ HÌNH RETINAFACE

### **Đây là một mô hình học sâu thị giác máy tính được dùng cho việc xác định chính xác vị trí của các khuôn mặt trong ảnh và được dựa trên nghiên cứu của bài báo** [[1905.00641] RetinaFace: Single-stage Dense Face Localisation in the Wild](https://arxiv.org/abs/1905.00641). Mô hình này cho phép phát hiện chính xác các điểm mốc trên khuôn mặt và các bounding box cho khuôn mặt trên ảnh. Đây là một mô hình có sẵn trên github với rất nhiều mô hình xương sống để có thể lựa chọn phù hợp như: mobilenet1\_0.25, mobilenetv1\_0.50, mobilenetv1, mobilenetv2, resnet18, resnet34 và resnet50. Trong việc áp dụng bài toán này, ta lựa chọn mô hình resnet34 làm xương sống chính cho mô hình này.

### Nguyên tắc hoạt động

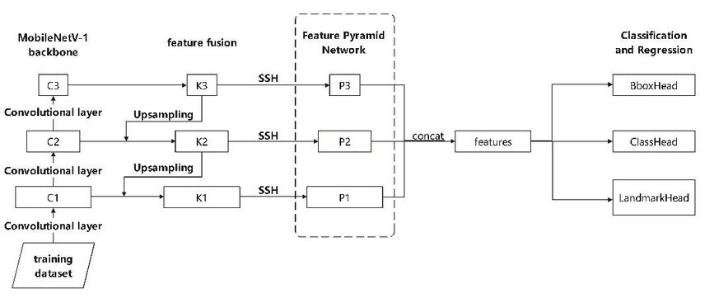
**Các thành phần cơ bản của RetinaFace:**

Backbone trích xuất đặc trưng : dùng các mạng học sâu để trích xuất đặc trưng của ảnh (feature extraction) để tạo ra các feature map với mức độ trừu tượng khác nhau

Feature Pỷamid network: mạng kim tự tháp đặc trưng giúp mô hình xử lý khuôn mặt ở các kích thước khác nhau (nhỏ -> lớn) giúp có thể phát hiện khuôn mặt ở các điều kiện phức tạp (nhỏ, nghiêng, dọc).

Anchor Base + Context Module : dùng để tăng receptive field mà **không giảm** kích thước (không pooling).

**Quá trình huấn luyện và cho ra outputs:**



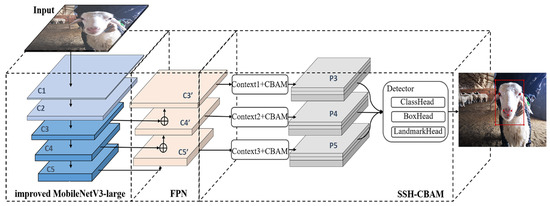
*Hình 3.1 Minh hoạ cách hoạt động của RetinaFace*

Với ví dụ pipline trên, chúng ta lựa chọn MobileNetV-1 làm backbone để trích xuất đặc trưng của ảnh đầu vào. Các dữ liệu huấn luyện sẽ đi qua các lớp tích chấp của mạng MobileNetV-1 và tạo ra ba tầng đặc trưng chính C1, C2 và C3. Mỗi tầng sẽ biểu diễn thông tin ở mức độ khác nhau : C1 sẽ chứa đặc trưng chi tiết (low-level), C2 sẽ chứa đặc trưng trung gian và C3 sẽ chứa đặc trưng cấp cao nhất (high-level).

Sau đó, mô hình bắt đầu quá trình hợp nhất đặc trưng (feature fusion) để kết hợp thông tin từ các tầng này. Các đầu ra C1, C2 và C3 được xử lý qua các lớp tích chấp k1, k2 và k3 sau đó được upsampling (phóng đại cái feature ở các tầng sâu hơn ) để có thể kết hợp lại (fuse) nhằm tạo ra đặc trưng thống nhất giúp mô hình có thể phát hiện khuôn mặt ở nhiều kích thước khác nhau.

Tiếp theo, các khối đặc trưng này sẽ đi qua các khối SSH (Single Stage headless). Mỗi khối SSH này có nhiệm vụ giúp mở rộng vùng nhìn (reception field), tăng cường thông tin ngữ cảnh mà không làm thay đổi kích thước của feature map, giúp mô hình có thể nhận diện cả những khuôn mặt nhỏ hay những khuôn mặt khó có thể phát hiện, bị che khuất. Đầu ra ba khối SSH là P1, P2 và P3 tạo thành FPN (feature pyramid networks) giúp mô hình có thể xử lý khuôn mặt ở các tỉ lệ, kích thước khác nhau, đảm bảo mô hình hoạt động tốt trên nhiều độ phân giải khuôn mặt.

Cuối cùng, các đặc trưng P1, P2 và P3 được nối (concatenate) lại và đưa ra đầu ra gồm 3 nhánh phân loại và hồi quy. BboxHead dùng để dự đoán bounding box quanh khuôn mặt được xác định, ClassHead chứa xác suất khuôn mặt trong vùng đó và LandmarkHead dự đoán đặc trưng trên khuôn mặt như mắt, mũi và miệng.



*Hình 3.2 Minh hoạ cách hoạt động của RetinaFace với hình ảnh cụ thể*

Biểu đồ trên thể hiện quá trình đi qua mô hình để xác định khuôn mặt của dê.

Mô hình này có 3 nhánh dự đoán, do đó Loss Function sẽ gồm 3 phần:

Classification Loss – Phân loại mặt / không mặt:

Sử dụng **Focal Loss** (giống RetinaNet) để xử lý class imbalance

p : xác suất dự đoán

: nhãn ground-truth (0 hoặc 1)

Bounding Box Regression Loss:

Dùng **Smooth L1 Loss** để dự đoán bounding box:

t : giá trị dự đoán bbox

: nhãn ground truth bbox

Landmark Regression Loss:

Dự đoán facial landmarks (mắt, mũi, miệng):

l : giá trị landmarks dự đoán

: nhãn ground-truth của landmarks

Tổng loss function:

RetinaFace được tối ưu hoá bằng **SGD (**Stochastic Gradient Descent) **với momentum, weight decay, warm-up và giảm learning rate theo từng mốc (step decay).**

### Ưu điểm và nhược điểm

**Ưu điểm:**

Độ chính xác cao :

RetinaFace có khả năng phát hiện khuôn mặt với **độ chính xác vượt trội**, đặc biệt là với các khuôn mặt nhỏ, nghiêng, hoặc bị che khuất. Nhờ việc sử dụng **Feature Pyramid Network (FPN)** và **SSH block**, mô hình có thể nắm bắt được thông tin ngữ cảnh ở nhiều tỉ lệ khác nhau.

Phát hiện khuôn mặt đa tỉ lệ:

Nhờ cơ chế feature fusion và upsampling, giúp mô hình có thể phát hiện tốt các khuôn mặt có kích thước khác nhau trong ảnh.

Phát hiện LandMarks:

Mô hình không chỉ phát hiện vị trí các khuôn mặt trong ảnh mà còn có thể xác định các đặc trưng trên khuôn mặt như mắt, mũi và miệng.

Hiệu năng tốt trên các mô hình nhẹ:

Hiệu năng của mô hình phụ thuộc phần lớn vào backbone mà ta lựa chọn. Với các mô hình nhẹ như MobilenetV1 có số lượng tham số nhỏ và giúp tăng khả năng chạy.

Khả năng tổng quát hoá tốt:

Mô hình chạy ổn định trên nhiều điều kiện góc sáng và biểu cảm khác nhau.

**Khuyết điểm:**

Chi phí cao nếu dùng backbone lớn:

Như đã nói hiệu năng mô hình phụ thuộc lớn vào backbone ta lựa chọn để trích xuất đặc trưng. Các mô hình như Resnet50 hay Resnet152 có thể khiến mô hình nặng và tốn tài nguyên.

Huấn luyện phức tạp:

Đây là một mô hình có quá trình huấn luyện phức tạp vì nó không chỉ xác định khuôn mặt mà còn các đặc trưng trên khuôn mặt đó.

Kích thước mô hình lớn:

Dù có thể chọn các mô hình backbone nhẹ hơn như mobilenetV1 nhưng Retinaface vẫn cho ra mô hình tương đối lớn hơn các mô hình khác.

## MÔ HÌNH VISION TRANSFORMER

Vision Transformer (ViT) là một mô hình học sâu tiên tiến được sử dụng trong lĩnh vực thị giác máy tính, mô hình này được cải tiến từ mô hình Transformer vô cùng nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên, để có thể áp dụng với dữ liệu hình ảnh. Không giống với mạng tích chập (CNN) vốn dựa vào các lớp tích chập để nắm bắt các đặc trưng, Vision Transformer sử dụng cơ chế self-attention để mô hình hoá mối quan hệ toàn cục trên các mảng ảnh. Mô hình này đã thể hiện hiệu suất vô cùng tốt trong nhiều tác vụ thị giác máy tính như phân loại hình ảnh, phát hiện đối tượng và phân cụm.

### Nguyên tắc hoạt động

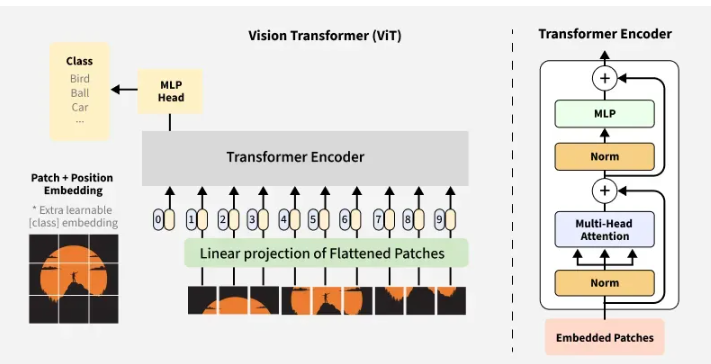
**Các thành phần cơ bản của Vision Transformer:**

Image Patching và Embedding: Image Patching (chia ảnh thành các patch) thay vì dùng các lớp tích chập như CNN để trích xuất feature map, ViT coi ảnh như một chuỗi (sequence) – tương tự như các token trong NLP. Sau đó các chuỗi này sẽ đi qua patch embedding để flatten các patch rồi chiếu tuyến tính (linear embedding) để biến thành chuỗi vector embedding đưa vào transformer.

**Positional Encoding:** giúp mô hình biết được vị trí của từng patch (token) trong ảnh, bù lại việc Transformer không có khái niệm thứ tự không gian.

**Transformer Encoder:** là khối xử lý chuỗi token bằng cơ chế self-attention và feed-forward network để trích xuất quan hệ và đặc trưng ngữ cảnh giữa các token.

**Classification Head (MLP Head):** là mạng tuyến tính (thường 1–2 lớp fully connected) dùng đầu ra của token [CLS] để dự đoán nhãn cuối cùng cho toàn bộ ảnh.



*Hình 3.2.1 Minh hoạ cách hoạt động của Vision Transformer*

Biểu đồ trên mô tả kiển trúc của mô hình Vision Transformer, các mà mô hình được áp dụng cho ảnh.

Ảnh → Patch Embedding + Position Encoding → Transformer Encoder → MLP Head → Dự đoán lớp ảnh.

Đầu tiên ảnh sẽ được chia thành các patch nhỏ (16x16). Mỗi patch sẽ được làm phẳng (flatten) và đưa qua một lớn chiếu tuyến tính (Flattened Patches) để biến thành các vector embedding có kích thuóc cố định. Các vector Embedding này sẽ được cộng thêm Positional Embedding (chứa thông tin vị trí của từng patch trong ảnh) và thêm một token đặc biệt (CLS) dùng để tổng hợp thông tin toàn ảnh trước khi vào Transformer Encoder.

Chuỗi embedding này sẽ được đưa vào Transformer Encoder, gồm nhiều lớp cấu trúc lặp lại gồm Multi-head Self-attention, Normalization và Feed-Forward MLP. Các lớp này giúp mô hình có thể học được mối quan hệ giữa các patch trong không gian ảnh, tương tự như cách Transformer học giữa các từ trong câu trong xử lý ngôn ngữ tự nhiên.

Multi-head Attention là cơ chế quan trọng nhất trong Vision Transformer và là cơ chế khiến nó đặc biệt hơn cả các mô hình khác khi nó cho phép mô hình có thể học được mối quan hệ giữa các patch trong không gian ảnh. Trong này, mỗi patch ảnh sẽ được xem như là một token, mục tiêu ở đây là để token có thể “thấy” và “hiểu” mối quan hệ với tất cả các token khác. Ở đây, mỗi patch sẽ được biến đổi thành 3 vector khác nhau: Query (Q), Key(K) và Value (V) thông qua phép chiếu tuyến tính (linear projection). Các vector này giúp mô hình xác định mối liên hệ giữa các patch. Cụ thể, Query của patch này sẽ được so sánh với Key của các patch khác để tính độ tương đồng (attetion score), biểu thị mức độ quan hệ giữa patch này và patch còn lại. Sau đó , điểm số này sẽ được chuẩn hoá bằng softmax thành trọng số chú ý rồi cộng (weight sum) với các vector value còn lại, tạo ra biểu diễn mới (các thông tin từ patch liên quan cộng lại) cho patch đó

Quá trình lại được xảy ra song song trên nhiều (head), mỗi head tập trung học kiểu quan hệ khác nhau trên các patch (màu sắc, cấu trúc , hình dạng) . Cuối cùng tất cả các head được nối lại rồi chiếu tuyến tính thành đầu ra cuối cùng. Nhờ cơ chế này khiến Vision Transformer trở nên đặc biệt hơn hẳn khi có thể học toàn cục hình ảnh thay vì chỉ học cục bộ như các mạng tích chập như CNN, giúp Vison Transformer có thể hiểu ngữ cảnh rộng và sâu hơn.

Sau khi qua tất cả các lớp encoder, **token [CLS]** (đại diện cho toàn bộ ảnh) được đưa vào **MLP Head (Classification Head)** — một mạng tuyến tính nhỏ dùng để **dự đoán lớp của ảnh**, như “bird”, “ball”, hoặc “car”.

Vision Transformer cho bài toán phân loại sử dụng hàm mất mát Cross-Entropy Loss:

C: Số lớp

: Nhãn thực tế (0 hoặc 1)

: Nhãn thực tế (0 hoặc 1)

Mô hình này sử dụng AdamW làm hàm tối ưu khi hàm này có thể giúp Vision Transformer học hiệu quả và ổn định hơn bằng cách điều chỉnh trọng số theo hướng tối ưu trong khi giảm overfitting bằng weight decay.

### Ưu điểm và nhược điểm

Ưu điểm :

Hiểu được mối quan hệ toàn cục:

Với cơ chế Multi-head Attetion có thể nắm bắt mối liên kết các vùng xa trong ảnh, điều mà CNN không làm được.

Linh hoạt và mở rộng tốt:

Cấu trúc Transformer dễ mở rộng (scaleable) và có thể áp dụng vào nhiều tác vụ thị giác khác nhau như classification, detection,…

Tận dụng pretrain tốt:

Khi được huấn luyện trên tập dữ liệu lớn (imageNet hoặc JFT 300M), Vit đạt khả năng vượt trội.

Dễ kết hợp với các mô hình ngôn ngữ:

Vì mô hình gốc được sử dụng cho xử lý ngôn ngữ tự nhiên nên ViT dễ tích hợp với các mô hình NLP.

**Khuyết điểm:**

Cần dữ liệu huấn luyện lớn: ViT khó học hiệu quả nếu dữ liệu ít

Tốn nhiều tài nguyên tính toán:

Cơ chế self-attention có độ phức tạp O() với số patch N, nên Vit tốn nhiều RAM và GPU hơn CNN.

Hội tụ chậm hơn:

Cần nhiều epoch và kỹ thuật huấn luyện (như warm-up, data augmentation) để đạt kết quả tốt.

Thiếu tính cục bộ:

Không tự động tập trung vào chi tiết nhỏ trừ khi được bổ sung cơ chế như Hybrid CNN-ViT*.*

Khó diễn giải trực quan hơn CNN:

Attention map giúp quan sát phần mô hình chú ý, nhưng vẫn khó hiểu hơn các feature map của CNN

# ÁP DỤNG CÁC MÔ HÌNH VÀO BÀI TOÁN

## Cấu hình môi trường

Sử dụng môi trường Kaggle notebooks để có thể thực hiện huấn luyện.

Cấu hình môi trường Kaggle notebooks:

GPU : NVIDIA T4

RAM: 29GB

CPU: Thường 4 nhân (cores) trong môi trường GPU

## Áp dụng mô hình vào bài toán xác định khuôn mặt

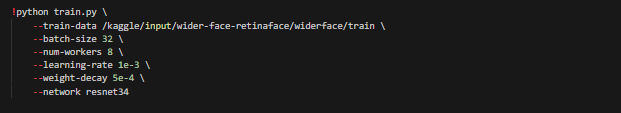
Với bài toán này, chúng ta sử dụng github repo có sẵn mô hình RetinaFace để thực hiện huấn luyện trên tập dữ liệu Wider-Face để có thể nhận diện khuôn mặt



Sau khi clone về máy, cài các thư viện học máy tương ứng



Sau khi cài tất cả các thư viện yêu cầu, ta bắt đầu thực hiện huấn luyện với bộ dữ liệu đã tiền xử lý với câu lệnh

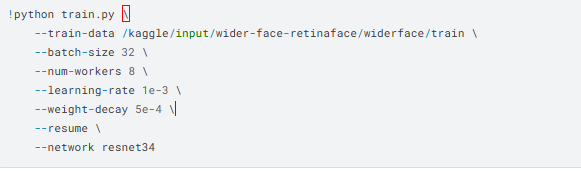


--train-data: đường dẫn đến nơi chứa tập data để train

--batch-size: với tập dữ liệu 1.82GB sử dụng batch-size 32 ảnh mỗi batch, nếu dùng 64 batch sẽ bị tràn

--learning rate, weitgt-decay sẽ dùng với các giá trị khởi tạo mặc định : 1e-3 và 5e-4.

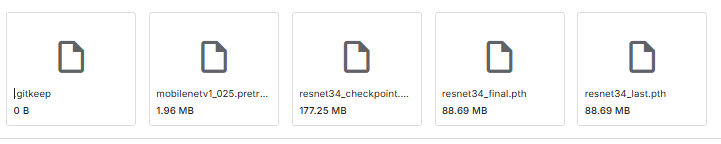
-- networks : sử dụng mô hình resnet34 dể trích xuất đặc trưng làm backbone



--resume: để tiếp tục huấn luyện từ checkpoint đã có.

Trên đây là câu lệnh tiếp tục huânts luyện từ checkpoint đã có trong folder weights.

Sau khi huấn luyện ta sẽ được mô hình :

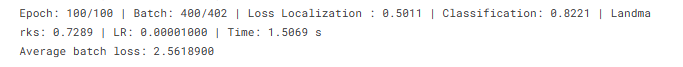


Tao có resnet34\_checkpoint.ckpt giữ checkpoint huấn luyện của mô hình để khi mô hình bị ngắt do lỗi thì có thể tiếp tục huấn luyện

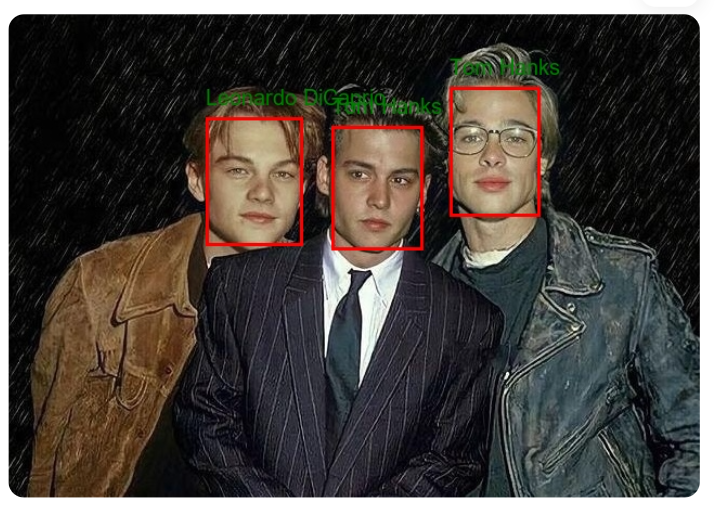
Resnet34\_last.pth giữ trọng số sau mỗi epoch huấn luyện

Resnet34\_final: Giữ trọng số cuối cùng sau khi mô hình huấn luyện.

### Kết quả thu được



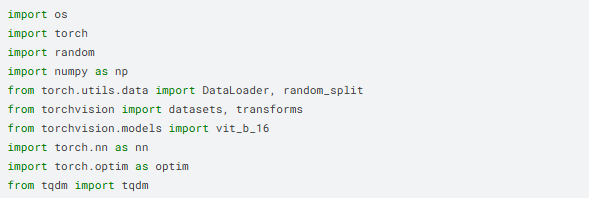
Mô hình hoàn tất quá trình huấn luyện 100 epoch; tổng loss đang hội tụ quanh 2.56, trong đó localization hoạt động tốt (0.50) nhưng classification và landmarks còn tương đối cao.



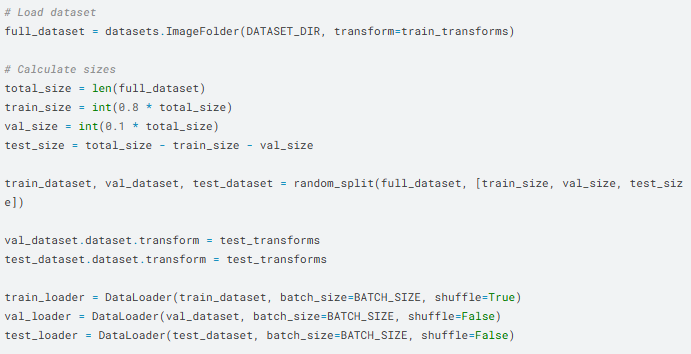
*Hình 4.1 Hình ảnh đầu ra của mô hình xác định khuôn mặt*

## Bài toán nhận diện khuôn mặt người nổi tiếng

Quá trình huấn luyện:



Import các thư viện sẽ được sử dụng trong quá trình huấn luyện và kiểm thử.

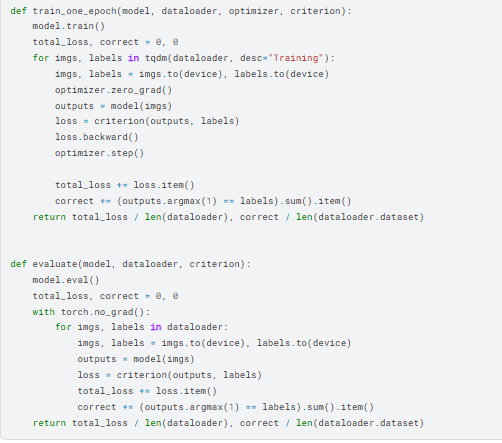


Sau khi load full\_dataset từ folder dữ liệu tổng. Chia dataset này ra các train\_loader, val\_loader, và test\_loader với train\_loader chiếm 80% dữ liệu, test\_loader và val\_loader chiếm mỗi cái 10% dữ liệu.



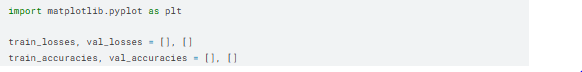


Load mô hình Vision Transformer với trọng số “IMAGENET1K\_V1”, sau đó thay đầu ra cuối lớp bằng số lớp hiện có trong tập dữ liệu. Sau đó khai báo hàm loss fuction CrossEntropyLoss cùng với hàm tối ưu AdamW để sử dụng cho việc huấn luyện



Ở đầu mỗi epoch chuyển model sang chế độ huânts luyện. Trong hàm huấn luyện, load từng ảnh và nhãn vào GPU, sau đó reset toàn bộ giá trị gradient của các tham số mô hình về 0 trước khi tính gradient sau mỗi lần huấn luyện. Rồi mô hình tính toán đầu ra và sai số lan truyền ngược để tính gradient, sau đó cập nhật trọng số. Cuối cùng, hàm tính tổng loss và số lượng dự đoán đúng để đánh giá độ chính xác trung bình của epoch.

Hàm evaluate dùng để kiểm tra hiệu suất của mô hình trên tập validation hoặc test. Mô hình được đặt ở chế độ đánh giá (model.eval()), tắt tính năng tính gradient với torch.no\_grad() để tiết kiệm bộ nhớ và tăng tốc. Tương tự như huấn luyện, hàm duyệt qua toàn bộ dữ liệu, tính loss và đếm số dự đoán chính xác nhưng **không cập nhật trọng số**. Kết quả trả về là giá trị loss trung bình và độ chính xác của mô hình trên toàn bộ tập dữ liệu đánh giá.





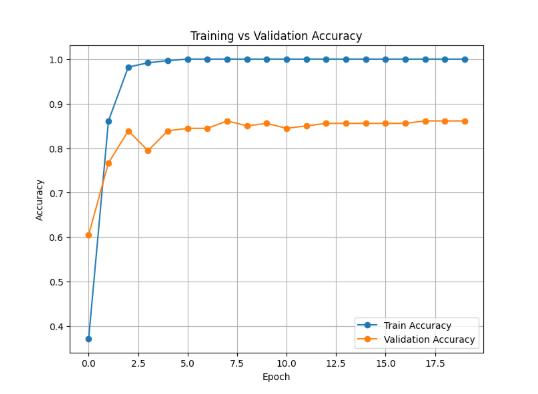
Bắt đầu thực hiện huấn luyện. Gọi hàm train\_one\_epoch để huấn luyện sau mỗi epoch. Sau đó gọi hàm đánh giá model với hàm evaluate để đánh giá tập validation sau mỗi epoch huất luyện với các criterion = CrossEntropyLosss. Sau đó lưu các kết quả đánh giá vào các danh sách train\_losses, val\_losses, train\_acciracies, val\_accuracies để đánh giá.

### Kết quả thu được



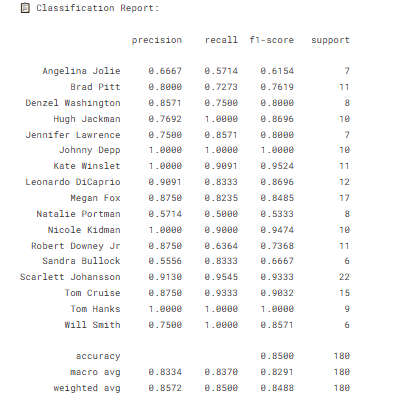
*Hình 4.2 Biểu đồ thể hiện trainning và validation loss*

Có thể thấy train loss và validation loss giảm theo thời gian cho thấy mô hình học có hiệu quả. Đặc biệt đến epoch thứ 5 thì mô hình bắt đầu có dấu hiệu hội tụ, và từ epoch thứ 11 thì mô hình bắt đầu hội tụ và không có cải thiện đáng kể thêm nữa.



*Hình 4.3 Biểu đồ thể hiện trainning và validation accuracy*

Tương tự Trainning accuracy và validation accuracy tăng dần sau mỗi epoch cho thấy mô hình học rất tốt trên tập dữ liệu này, mô hình ở đây có thể thấy bắt đầu hội tụ ở epoch thứ 10, với tranning accuracy xấp xỉ ở mức 0.98 và validation Accuracy ở mức 0.85.



*Hình 4.4 Bảng báo cáo đánh giá hiệu suất mô hình*

Bảng trên báo cáo đánh giá hiệu suất của mô hình phân loại theo từng lớp (từng người nổi tiếng) trên tập test. Mỗi hàng hiển thị các chỉ số **precision**, **recall**, và **f1-score**, cùng với **support** là số mẫu thực tế của lớp đó trong tập kiểm tra.

**Precision** đo mức độ chính xác trong các dự đoán dương tính (bao nhiêu dự đoán đúng trên tổng số dự đoán).

**Recall** cho biết mô hình phát hiện được bao nhiêu mẫu đúng trong tất cả mẫu thực tế của lớp đó.

**F1-score** là trung bình điều hòa giữa precision và recall, phản ánh cân bằng giữa hai đại lượng.

**Accuracy (85.8%)** là tỷ lệ dự đoán đúng trên toàn bộ tập dữ liệu. cho thấy mô hình dự đoán khá tốt với độ chính xác 85.8%

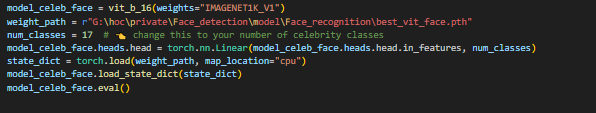
Kết quả này cho thấy mô hình hoạt động tốt, với độ chính xác và F1-score tổng thể đều trên **84%,** thể hiện khả năng phân loại khá ổn định giữa các lớp khác nhau.

### 

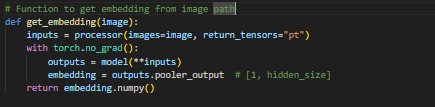
### *Hình 4.5 Ma trận nhầm lẫn của mô hình*

Ma trận nhầm lẫn cho thấy đa số các ảnh đều được phân loại đúng với ảnh chỉ trừ một số nhãn như Natalie Portman, Robert Downey jr, Megan Fox là sai.

## Bài toán nhận diện khuôn mặt



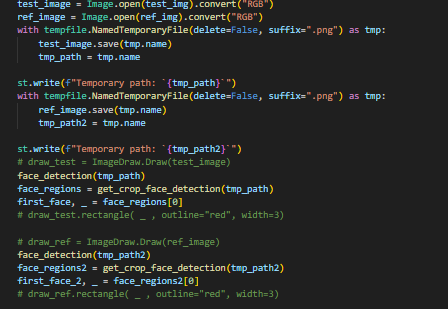
Khởi tạo mô hình dùng để embedding các ảnh sau đó dùng các embedding để so sánh 2 ảnh nhằm xác định danh tính 2 ảnh có phải cùng một người hay không.

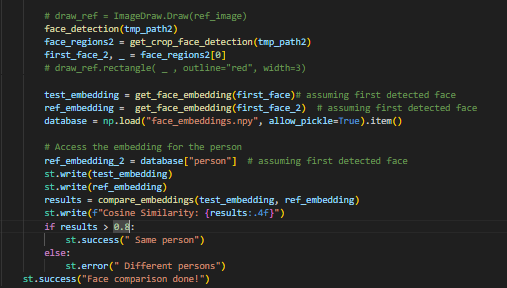


Hàm get\_face\_embedding dùng để lấy embedding của các khuôn mặt trong ảnh



Sử dụng hàm cosin để tính sự tương đồng giữa 2 embedding từ đó có thể đưa ra so sánh.

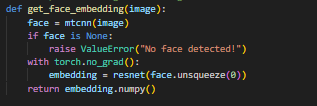




Chương trình sẽ lấy ảnh từ giao diện streamlit, sau đó sẽ đi qua mô hình để xác định các khuôn mặt trong ảnh, sau khi có được vùng khuôn mặt ta sẽ đưa vùng đó vào các Vision Transformer để lấy embedding. Sau khi lấy embedding ta sẽ lưu embedding lại và dùng nó trong tương lai khi so sánh với một ảnh khác.

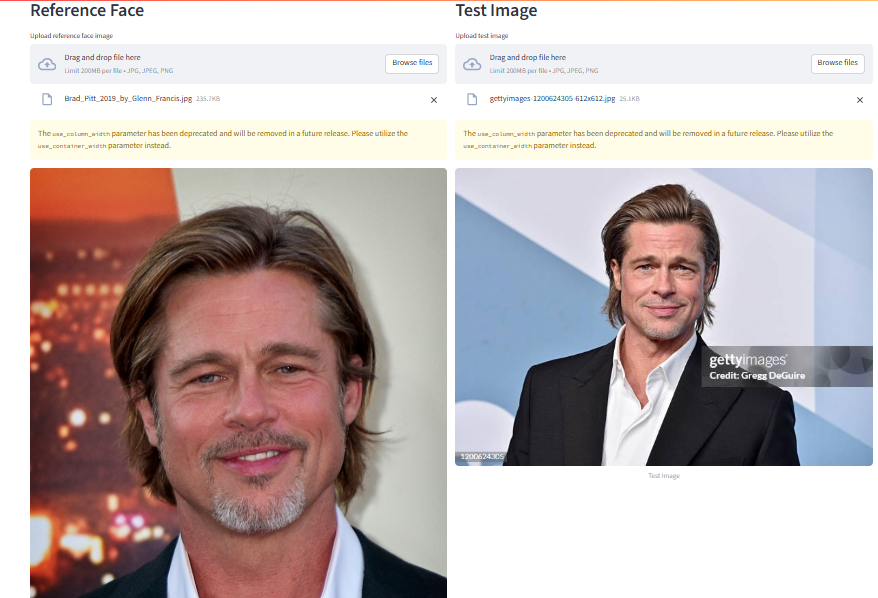
Threshhold để xác định danh tính sẽ là 0.8, tránh các trường hợp 2 khuôn mặt quá giống nhau nhưng không phải cùng một người.



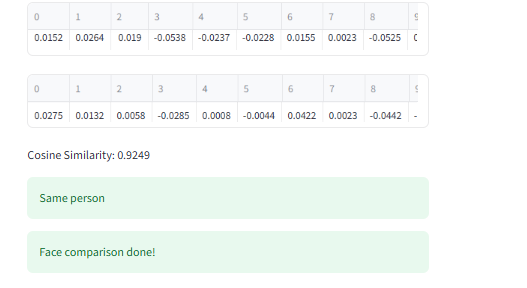


Với mô hình Vision Transformer để embedding khuôn mặt chưa có tính hiệu quả vì mô hình pretrain chưa phải là mô hình tập trung vào khuôn mặt. Một giải pháp đó chính là sử dụng mô hình InceptionResnetV1 từ thư viện MTCNN chuyên để embedding khuôn mặt và nhận dạng danh tính trùng nhau để đạt hiệu quả cao hơn.

### Kết quả thu được



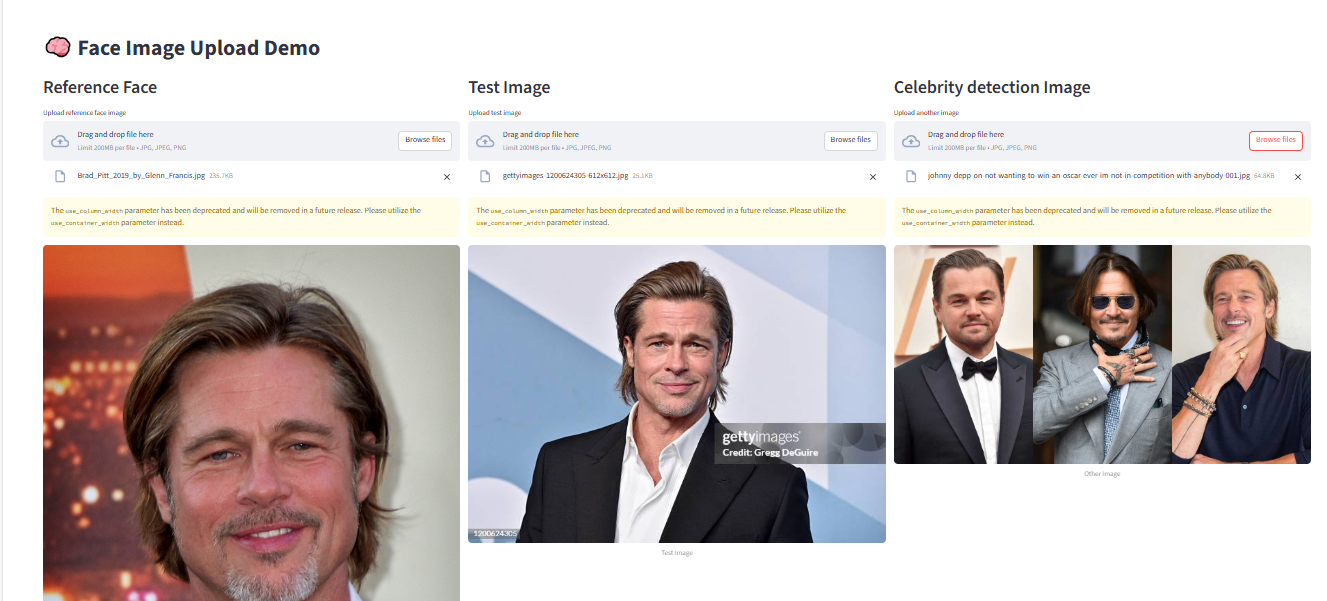
*Hình 4.6 Giao diện của mô hình nhận diện khuôn mặt*



*Hình 4.7 Kết quả của mô hình nhận diện khuôn mặt*

Sau khi embedding 2 ảnh chúng ta sẽ có những embedding như trên, sử dụng cosin để so sánh 2 embedding cho thấy cả 2 ảnh đều cùng là một người.

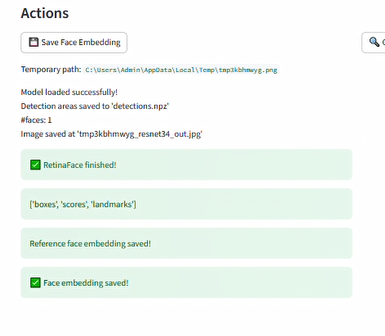
# GIAO DIỆN DEMO



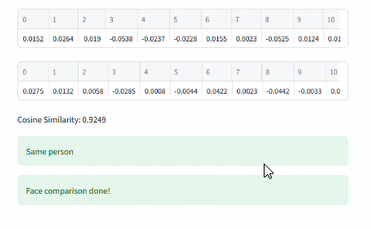
*Hình 5.1 Trang chủ của giao diện demo chính*

Demo được triển khai sử dụng streamlit – frameworks giúp tạo ra ứng dụng web tương tác cho các dự án machine learning, deep learning và AI

3 ô để truyền các ảnh vào, với reference Face để lưu embedding, Test image để so sánh danh tính với ảnh reference và cuối cùng là celebrity detection image để xác định danh tính người nổi tiếng.

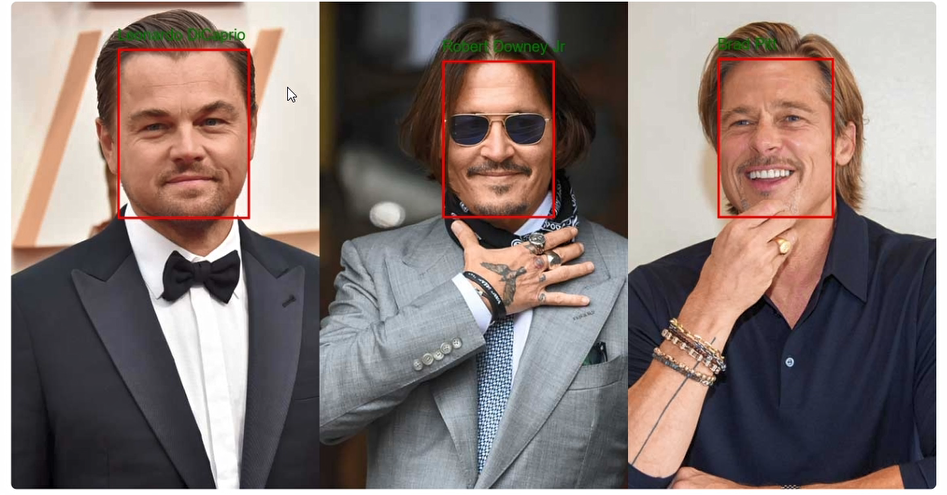


Kết quả sau khi thu được embedding của ảnh reference



*Hình 5.2 Kết quả sau khi so sánh embedding 2 ảnh*

Kết quả sau khi so sánh 2 ảnh references Face và Test image cho ta được cùng một người



*Hình 5.*3 *Kết quả sau khi phân loại người nổi tiếng trên demo giao diện*

.

# KẾT LUẬN

Sau quá trình huấn luyện và đánh giá mô hình, bài toán **xác định và nhận diện khuôn mặt** đã đạt được kết quả khả quan với độ chính xác và F1-score cao, cho thấy mô hình có khả năng xác định và phân biệt tốt giữa các khuôn mặt khác nhau. Việc áp dụng các kỹ thuật học sâu (như CNN hoặc Vision Transformer) cùng với tối ưu hóa hợp lý (AdamW, SDG) giúp mô hình học được đặc trưng khuôn mặt một cách hiệu quả.

Bài toán xác định và nhận dạng khuôn mặt có ý nghĩa vô cùng quan trọng và ứng dụng rộng dãi trong thực tế, trong an ninh giám sát, mở khoá thiết bị, xác thực danh tính, chấm công tự động hay các hệ thống quản lý thông minh. Tuy vậy, để thực sự có thể làm chủ bài toán này, vẫn có các yếu tố khác cần phải xem xét như điều kiện ánh sáng, góc chụp, biểu cảm khuôn mặt, và quyền riêng tư cá nhân. Do đó cần phải tiếp tục tối ưu và cải thiện để có thể áp dụng tốt công nghệ này vào trong đời sống

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | https://www.geeksforgeeks.org/deep-learning/vision-transformer-vit-architecture/ |
| [2]  [3] | https://www.geeksforgeeks.org/machine-learning/introduction-convolution-neural-network/  [1905.00641] RetinaFace: Single-stage Dense Face Localisation in the Wild: https://arxiv.org/abs/1905.00641 |