

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ



ĐỒ ÁN TỐT NGHIỆP

Ngành: Công nghệ thông tin

Họ và tên sinh viên: Hoàng Văn Vũ

Lớp, khoá: DK10-CNTT

NGÀNH: CÔNG NGHỆ THÔNG
TIN

Hoàng Văn Vũ

HẢI DƯƠNG – NĂM 2023

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ



ĐỒ ÁN TỐT NGHIỆP

Ngành: Công nghệ thông tin

TÊN ĐỀ TÀI: NGHIÊN CỨU, XÂY DỰNG ỦNG DỤNG PHÁT HIỆN
KHUÔN MẶT GIẢ

Họ và tên sinh viên: Hoàng Văn Vũ

Lớp, khoá: DK10-CNTT

Giảng viên hướng dẫn: ThS. Hoàng Thị An

HẢI DƯƠNG – NĂM 2023

LỜI CAM ĐOAN

Tôi xin cam đoan các kết quả đưa ra trong đồ án/khóa luận tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu, thực nghiệm của tôi dưới sự hướng dẫn của ThS. Hoàng Thị An, không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục các tài liệu tham khảo.

Nếu sai tôi xin chịu mọi hình thức kỷ luật theo quy định.

Hải Dương, ngày.....tháng.....năm.....

Sinh viên thực hiện

(Ký, ghi rõ họ và tên)

Hoàng Văn Vũ

Đại học Sao Đỏ

MỤC LỤC

MỞ ĐẦU	1
Chương 1. CƠ SỞ LÝ THUYẾT	3
1.1. Tổng quan về xử lý ảnh	3
1.1.1. Tổng quan.....	3
1.1.2. Các quá trình xử lý ảnh[1]	3
1.1.3. Ảnh và biểu diễn ảnh	4
1.1.4. Phạm vi ứng dụng xử lý ảnh	6
1.1.5. Các loại tệp cơ bản trong xử lý ảnh	6
1.2. Bài toán nhận dạng khuôn mặt [2].....	9
1.2.1. Tổng quan.....	9
1.2.2. Nhận dạng khuôn mặt dựa vào bản đồ cạnh[2]	10
1.2.3. Khoảng cách Hausdorff	11
1.2.4. Face Recognition.....	15
1.3. Bài toán nhận diện khuôn mặt giả	18
1.3.1. Tổng quan.....	18
1.3.2. Các phương pháp phát hiện khuôn mặt giả [14]	18
Chương 2. KỸ THUẬT TẠO VÀ PHÁT HIỆN KHUÔN MẶT GIẢ	22
2.1. Kỹ thuật tạo khuôn mặt giả.....	22
2.1.1. Mô hình Generative Adversarial Networks (GAN)[3]	22
2.1.2. Deepfake Technology	25
2.1.3. 3D Printing và Silicone Masks.....	27
2.2. Kỹ thuật phát hiện khuôn mặt giả.....	29
2.2.1. Phân tích đặc điểm vùng khuôn mặt	29
2.2.2. Công cụ quét 3D	31
2.2.3. Liveness Detection.....	32
2.2.4. Phát hiện hoán đổi khuôn mặt Deepfake dựa trên nhận dạng ngầm[7]	33
2.2.5. Phát hiện hình ảnh Deepfake bằng kỹ thuật Deep Learning [8]	36
Chương 3. XÂY DỰNG ỨNG DỤNG	39
3.1. Lựa chọn công cụ.....	39
3.2. Mô hình mạng CNN phát hiện khuôn mặt giả mạo.....	39
3.3. Cơ sở dữ liệu.....	40
3.4. Xây dựng module	42
3.4.1. Module nhận diện khuôn mặt.....	42
3.4.2. Module phát hiện khuôn mặt giả.....	46
3.5. Xây dựng website demo	51
3.6. Thủ nghiệm.....	53
KẾT LUẬN	55
TÀI LIỆU THAM KHẢO	56

DANH MỤC BẢNG CÁC CHỮ VIẾT TẮT

Kí hiệu, chữ viết tắt	Viết đầy đủ	Ý nghĩa
XLA	Xử lý ảnh	Xử lý ảnh
CNN	Convolutional neural networks	Mạng nơ-ron tích chập
GAN	Generative Adversarial Network	Mạng đối nghịch tạo sinh

Đại học Sao Đỏ

DANH MỤC HÌNH ẢNH

Hình 1.1. Sơ đồ tổng quát của hệ thống xử lý ảnh.....	4
Hình 1.2. Ảnh vector.....	5
Hình 1.3. Ảnh raster	6
Hình 1.4. Bản đồ cạnh (edge map) của khuôn mặt	11
Hình 1.5. Kết quả của bản đồ cạnh	11
Hình 1.6. Khoảng cách giữa hai đường thẳng song song	13
Hình 1.7. Các trường hợp $d_{ss} = 0$	14
Hình 1.8. Khoảng cách Hausdorff	15
Hình 1.9. Luồng xử lý của Face Recognition	16
Hình 1.10.Tiền xử lý	16
Hình 1.11.Nhận diện khuôn mặt trong điều kiện lý tưởng	17
Hình 1.12.Nhận diện khuôn mặt trong điều kiện phức tạp	17
Hình 1.13.Xây dựng một LBP (bước 1).....	19
Hình 1.14.Biểu diễn thập phân của 8 liền kề điểm đang xét	19
Hình 1.15.Giá trị LBP sau được tính toán	20
Hình 1.16.Biểu đồ bảng số lần mỗi mẫu LBP	20
Hình 2.1. Minh họa các mạng trong GAN	22
Hình 2.2. Ảnh mặt GAN sinh ra qua các năm	23
Hình 2.3. StyleGAN	24
Hình 2.4. Stargan.....	24
Hình 2.5. Age-cGAN	24
Hình 2.6. Ví dụ Deepfake	25
Hình 2.7. Khuôn mặt giả Deepfake.....	26
Hình 2.8. Dấu hiệu nhận biết DeepFake	27
Hình 2.9. In 3D.....	27
Hình 2.10.Khuôn mặt giả 3D	28
Hình 2.11.Công nghệ in 3D mặt giả	28
Hình 2.12.Sinh trắc học.....	29
Hình 2.13.Phép phân tích thành phần chính	30
Hình 2.14.Ví dụ phân tích đặc điểm vùng khuôn mặt	31
Hình 2.15.Quét 3D đặc điểm khuôn mặt	31
Hình 2.16.Quét 3D khuôn mặt	32
Hình 2.17.Phát hiện sự sống	33
Hình 2.18.Nhận dạng khuôn mặt mục tiêu tương ứng dựa trên khuôn mặt giả tạo.....	34
Hình 2.19.Phác thảo về khuôn khổ định hướng nhận dạng tiềm ẩn được đề xuất	35
Hình 2.20.Deepfake	36
Hình 2.21.Kiến trúc nhận dạng Deepfake.....	37
Hình 2.22.InceptionResNetV2. (a) độ chính xác và (b) biểu đồ tổn hao.....	38
Hình 3.1. Thư viện sử dụng trong dự án	39

Hình 3.2. Cấu trúc của dự án.....	40
Hình 3.3. Tập dữ liệu	41
Hình 3.4. Hình ảnh được quay từ điện thoại	41
Hình 3.5. Hình ảnh giả mạo được quay lại từ camera máy tính	42
Hình 3.6. Nhận diện khuôn mặt	43
Hình 3.7. Phát hiện khuôn mặt.....	45
Hình 3.8. Quá trình xử lý video	46
Hình 3.9. Dữ liệu trích xuất từ video	47
Hình 3.10.Kiến trúc LivenessNet.....	48
Hình 3.11.Quy trình huấn luyện module phân biệt khuôn mặt.....	49
Hình 3.12.Kết quả của module phân biệt khuôn mặt.....	50
Hình 3.13.Module Streamlist	51
Hình 3.14.Triển khai website demo	52
Hình 3.15.Kết quả trả về sau khi chạy ứng dụng	52
Hình 3.16.Trong môi trường thiếu sáng.....	53
Hình 3.17.Trong điều kiện tối	53

Đại học Sao Đỏ

DANH MỤC BẢNG

Bảng 1.1. So sánh các chuẩn định dạng ảnh	9
Bảng 1.2. Bảng tổng hợp thư viện và thuật toán cho mỗi phương pháp	17
Bảng 3.1. Hiệu suất và độ chính xác	53

Đại học Sao Đỏ

MỞ ĐẦU

1. Lý do chọn đề tài

Bảo mật là một trong những vấn đề quan trọng và được chú trọng hàng đầu trong thế giới công nghệ hiện nay. Với sự phát triển của công nghệ thông tin, việc sử dụng các thiết bị nhận diện khuôn mặt để bảo vệ an ninh và dữ liệu đã trở thành xu hướng phổ biến. Tuy nhiên, việc sử dụng khuôn mặt để xác thực cũng đem lại một số rủi ro, trong đó có việc sử dụng khuôn mặt giả để đánh lừa hệ thống bảo mật. Vì vậy, nghiên cứu và xây dựng ứng dụng phát hiện khuôn mặt giả trong bảo mật là một vấn đề cấp bách và được quan tâm rất nhiều trong cộng đồng khoa học và công nghệ.

Trong thời đại công nghệ số, việc sử dụng khuôn mặt để xác thực đã trở thành một xu hướng phổ biến. Tuy nhiên, việc sử dụng khuôn mặt để xác thực cũng đem lại một số rủi ro, trong đó có việc sử dụng khuôn mặt giả để đánh lừa hệ thống bảo mật. Vì vậy, nghiên cứu và xây dựng ứng dụng phát hiện khuôn mặt giả trong bảo mật là một vấn đề cấp bách và được quan tâm rất nhiều trong cộng đồng khoa học và công nghệ.

2. Mục tiêu nghiên cứu

Xây dựng ứng dụng có khả năng phát hiện được khuôn mặt thật và khuôn mặt giả.

3. Đối tượng nghiên cứu

- Kỹ thuật phát hiện khuôn mặt giả.
- Các thư viện xây dựng và triển khai mô hình học sâu.
- Các thư viện hỗ trợ xây dựng giao diện người dùng và xử lý video trực tuyến.

4. Phạm vi nghiên cứu

Phát hiện Khuôn mặt giả: Nghiên cứu tập trung vào việc phát hiện khuôn mặt giả trong các hình ảnh và video. Điều này bao gồm quá trình xử lý ảnh và sử dụng mô hình học sâu để đánh giá tính thật/giả của khuôn mặt.

Liveness Detection: Phạm vi nghiên cứu mở rộng đến việc sử dụng mô hình liveness detection để xác định xem khuôn mặt là thật hay giả. Điều này bao gồm việc sử dụng dữ liệu huấn luyện để đào tạo mô hình nhận diện tính thật/giả của khuôn mặt.

Giao diện người dùng và xử lý video trực tuyến: Sử dụng Streamlit và WebRTC để xây dựng giao diện người dùng và thực hiện xử lý video trực tuyến từ webcam hoặc camera khác.

5. Phương pháp nghiên cứu

Phương pháp nghiên cứu tài liệu: Nghiên cứu các tạp chí khoa học, đồ án và tài liệu chuyên ngành.

Phương pháp thực nghiệm: Thiết kế, xây dựng, chạy thử nghiệm ứng dụng và đánh giá kết quả.

6. Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học:

Nâng cao kiến thức về bảo mật khuôn mặt: Đóng góp vào lĩnh vực nghiên cứu về bảo mật khuôn mặt, đặc biệt là trong việc phát hiện khuôn mặt giả.

Tích hợp công nghệ học sâu và thị giác máy tính: Áp dụng các phương pháp và kỹ thuật mới nhất trong lĩnh vực học sâu và thị giác máy tính.

Ý nghĩa thực tiễn:

Ứng dụng trong bảo mật và an ninh: Cung cấp một giải pháp thực tế cho việc phát hiện khuôn mặt giả, có thể tích hợp vào các hệ thống bảo mật và an ninh.

Bảo vệ thông tin cá nhân: Đóng góp vào việc bảo vệ thông tin cá nhân của người dùng khi sử dụng các hệ thống xác thực khuôn mặt.

7. Kết cấu đề tài

Đề tài được chia thành 3 chương:

Chương 1. Cơ sở lý thuyết: Trình bày các kiến thức tổng quan về xử lý ảnh, bài toán nhận dạng khuôn mặt và bài toán nhận dạng khuôn mặt giả.

Chương 2. Kỹ thuật tạo và phát hiện khuôn mặt giả: Trình bày các phương pháp và kỹ thuật hiện đại trong việc tạo và phát hiện khuôn mặt giả. Đi sâu vào các thuật toán và công nghệ được sử dụng, cũng như các thách thức và hạn chế của chúng.

Chương 3. Xây dựng ứng dụng: Tập trung vào việc áp dụng các kiến thức và kỹ thuật đã được trình bày trong hai chương trước để xây dựng một ứng dụng thực tế. Chương này sẽ bao gồm thiết kế ứng dụng, lựa chọn công nghệ, triển khai và kiểm thử ứng dụng.

Chương 1. CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về xử lý ảnh

1.1.1. Tổng quan

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hóa các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

Biến đổi ảnh làm tăng chất lượng ảnh.

Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn ảnh,...

Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

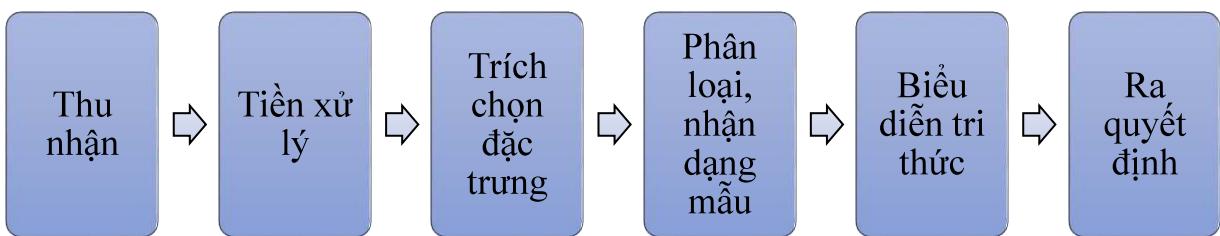
1.1.2. Các quá trình xử lý ảnh[1]

Một hệ thống xử lý ảnh thường bao gồm các thành phần chính như thiết bị phần cứng (máy ảnh) để chụp hình và lưu trữ dữ liệu, các công cụ phần mềm phục vụ xử lý và giải quyết yêu cầu của chức năng hệ thống đề ra. Trong lĩnh vực khoa học máy tính, hệ thống xử lý ảnh là đối tượng nghiên cứu liên quan đến kỹ thuật thị giác máy tính (computer vision), là quá trình biến đổi từ một ảnh ban đầu được thu nhận từ thiết bị sang một không gian mới sao cho làm nổi bật đặc tính dữ liệu, thuận lợi cho quá trình xử lý thông tin và nâng cao độ chính xác. Một hệ thống xử lý ảnh thường bao gồm một số thành phần chính sau:

Thu nhận ảnh là việc hình ảnh về thế giới thực được thu nhận và chuyển qua tín hiệu ảnh rời rạc thông qua máy ảnh kỹ thuật số hoặc các thiết bị thu hình khác.

Tiền xử lý là bước xử lý trên ảnh đầu vào nhằm khử nhiễu, làm nổi bật một số tính chất của ảnh nhằm nâng cao chất lượng các bước xử lý sau.

Trích chọn đặc trưng là quá trình biến đổi dữ liệu ảnh đầu vào thành các tập đặc trưng. Các đặc trưng thường có tính phân biệt cao của mẫu đầu vào giúp cho việc phân biệt mẫu dữ liệu ảnh dễ dàng hơn nhằm nâng cao chất lượng phân loại mẫu sao với xử lý dữ liệu thô trên giá trị pixel ảnh. Việc trích chọn đặc trưng cũng có thể làm giảm kích thước thể hiện thông tin trong ảnh khi dữ liệu về đặc trưng ảnh có tính phân biệt cao.



Hình 1.1. Sơ đồ tổng quát của hệ thống xử lý ảnh

Phân loại, nhận dạng mẫu là quá trình xử lý dữ liệu bằng các kỹ thuật, phương pháp phân tích đặc trưng để phân loại mẫu về các nhóm có một số tính chất chung. Các phương pháp phân loại, nhận dạng mẫu thường liên quan đến kỹ thuật học máy, bao gồm cả học có giám sát và học không có giám sát.

Biểu diễn tri thức là bước thể hiện mức cao của biểu diễn tri thức, các mẫu dữ liệu sau khi phân loại, nhận dạng được biểu diễn dưới dạng tri thức giúp hệ thống có khả năng “hiểu biết” ngữ nghĩa của nó theo từng kiểu ứng dụng khác nhau trong hệ thống trí tuệ nhân tạo và hệ thống thông minh.

Ra quyết định là bước cuối cùng của một hệ thống trong lĩnh vực thông minh. Các mẫu được biểu diễn dưới dạng tri thức và được suy luận ngữ nghĩa để đưa ra các quyết định thực hiện một nhiệm vụ nào đó. Ví dụ trong hệ thống robot di chuyển tự động, khi phát hiện chướng ngại vật, robot sẽ tự động ra quyết định tìm kiếm đường đi mới và di chuyển theo đường đi khả thi.

1.1.3. Ảnh và biểu diễn ảnh

Ảnh trong thực tế là một ảnh liên tục cả về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính thì cần thiết phải tiến hành số hóa ảnh. Quá trình số hóa biến đổi các tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng tử hóa các thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được hai điểm liền kề nhau. Các điểm như vậy được gọi là các pixel (Picture Element) hay các phần tử ảnh hoặc điểm ảnh. Ở đây cần phân biệt khái niệm pixel hay đè cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm các điểm nhỏ, gọi là pixel. Mỗi pixel gồm một tập tọa độ (x, y) và màu. Như vậy mỗi ảnh là tập hợp các điểm ảnh. Khi được số hóa nó thường được biểu diễn bởi mảng 2 chiều $I(n,p)$: n là dòng và p là cột. Về mặt toán học có thể xem ảnh là một hàm hai biến $f(x,y)$ với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột còn y là các hàng). Giá trị của hàm ảnh $f(x,y)$ được hạn chế trong phạm vi của các số nguyên dương

$$0 \leq f(x, y) \leq f_{max}$$

Với ảnh đen trắng mức xám của ảnh có thể được biểu diễn bởi một số như sau:

$$f = k \int c(\lambda) S(\lambda) d\lambda$$

Trong đó SBW là đặc tính phổ của cảm biến được sử dụng và k là hệ số tỷ lệ xích. Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng nên

SBW được chọn giống như là hiệu suất sáng tương đối. Vì f biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn.

$$0 \leq f \leq f_{max}$$

Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster.

Trong tổ chức lưu trữ và xử lý hình ảnh có hai dạng cấu trúc dữ liệu cơ bản là dạng ảnh bitmap (hay còn gọi là raster) và dạng ảnh vector.

Ảnh vector

Xét về mặt cấu trúc tổ chức, ảnh vector được tạo nên từ những yếu tố chính của hình học như điểm rời rạc, các đường thẳng, đường cong, đa giác và các vùng tương ứng với các đối tượng. Trên cơ sở đó vector được tạo thành dựa trên những biểu thức toán học (hoặc xấp xỉ), các vector này đi qua các điểm chính với mỗi điểm có một tọa độ x, y nhất định trên hệ trục tọa độ. Nhờ vậy, các điểm ảnh chi tiết trên đối tượng khi phóng sẽ được nội suy dựa vào những điểm chính và biểu thức toán học để tính giá trị điểm ảnh giữa các điểm chính.



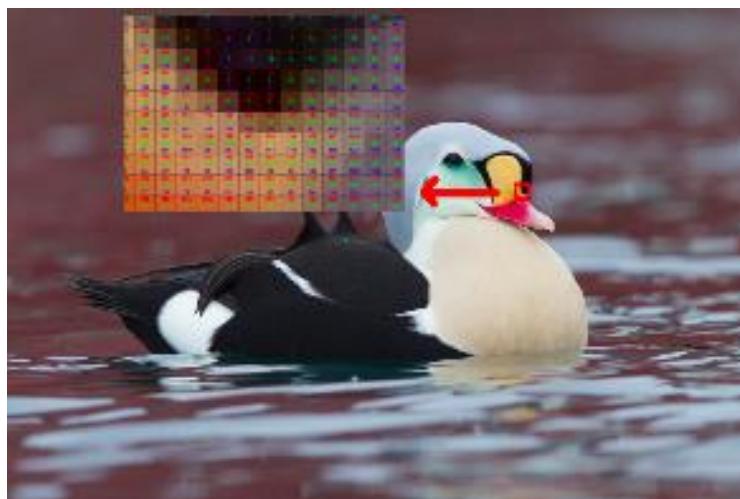
Hình 1.2. Ảnh vector

Ưu điểm của ảnh vector là khi phóng to hoặc thu nhỏ ảnh không bị vỡ, đường biên giữa các vùng không bị hiện tượng răng cưa. Kích thước ảnh vector thường nhỏ, chứa đối tượng đơn giản. Ảnh vector được dùng nhiều trong các trường hợp thiết kế logo, icon avatar, ảnh nghệ thuật vector.

Hạn chế của ảnh vector là hình ảnh hiển thị thường không “tự nhiên”, có sự chuyển màu và không sắc nét với ảnh ngoại cảnh, phân phối màu phức tạp.

Ảnh raster

Ảnh raster hay ảnh bitmap tổ chức biểu diễn theo cấu trúc lưới các điểm màu thể hiện các pixel, nó được tạo ra bởi các điểm ảnh rời rạc, chứa giá trị mỗi màu nhất định. Như vậy, ở độ phân giải chuẩn, ảnh raster nguyên gốc thể hiện hình ảnh đối tượng giống với tự nhiên hơn ảnh vector do không phải tính dựa vào các biểu thức toán học để tính ra các điểm ảnh giữa trên cơ sở các điểm chính. Hầu hết các ảnh được lưu trữ theo các định dạng thông thường đều theo dạng cấu trúc raster và các biến thể nén như GIF, JPEG và PNG. Ảnh dạng raster thường có kích thước lớn hơn ảnh vector.



Hình 1.3. Ảnh raster

Khác với ảnh vector, ảnh raster khi phóng to thường bị hiện tượng răng cưa, đối tượng không sắc nét. Nếu ảnh gốc kích thước nhỏ, khi phóng quá to so với ban đầu đối tượng thường không giữ lại được diện mạo, bị mờ.

1.1.4. Phạm vi ứng dụng xử lý ảnh

Ngày nay, với sự hỗ trợ của các hệ thống tính toán lớn, các thuật toán tiên tiến ra đời cho phép máy tính có thể hiểu biết và quyết định tốt hơn con người trong một số lĩnh vực nhất định. Ví dụ hệ thống xử lý ảnh bằng mạng nơron nhân tạo học sâu có thể nhận dạng, phân loại các kiểu đối tượng khác nhau tốt hơn và nhanh hơn con người. Xử lý ảnh có rất nhiều ứng dụng trong hầu hết các lĩnh vực của đời sống xã hội dân sự, an ninh quốc phòng, hàng không vũ trụ như:

Lĩnh vực quân sự, an ninh, quốc phòng: Tự động nhận dạng, phát hiện tội phạm, theo vết và truy tìm thủ phạm thông qua hình ảnh phát hiện hiện trường phạm tội và các vấn đề hỗ trợ dò tìm tội phạm qua hệ thống giám sát an ninh toàn cầu, quốc gia.

Trong lĩnh vực y tế: Phân tích hình ảnh, chẩn đoán bệnh qua các loại hình ảnh tia Gamma, X-quang, scan PET/CT (cắt lớp phản xạ), ảnh cực tím và đặc biệt với sự thành công của kỹ thuật học sâu đã giúp cho các chẩn đoán hình ảnh y học đạt kết quả cao.

Trong lĩnh vực viễn thám, vũ trụ: Thám hiểm vũ trụ, do thám, phân tích và phát hiện vật thể trong vũ trụ.

Trong lĩnh vực giao thông, dân sự: Các hệ thống khôi phục ảnh, chỉnh sửa, điều chỉnh độ phân giải, xử lý màu sắc, mã hóa và truyền tin, nhận dạng và phân loại hành động trong các hệ thống giám sát an ninh; hệ thống xe không người lái, giám sát sản xuất công nghiệp, robot phục vụ dân sự, giám sát bãi xe thông minh, kiểm soát – điều khiển giao thông thông minh.

1.1.5. Các loại tệp cơ bản trong xử lý ảnh

Ảnh thu được sau quá trình số hoá có nhiều loại khác nhau phụ thuộc vào kỹ thuật số hoá ảnh và các ảnh thu nhận được có thể lưu trữ trên tệp để dùng cho việc xử lý các bước tiếp theo. Sau đây là một số loại tệp cơ bản và thông dụng nhất hiện nay

Ngày nay có rất nhiều kiểu định dạng ảnh khác nhau, một số loại định dạng được dùng phổ biến như JPG, PNG, GIF, TIFF và BMP. Ứng với mỗi định dạng ảnh cụ thể sẽ có các thuộc tính khác nhau, phương pháp mã hóa, lưu trữ khác nhau và được tạo ra để sử dụng vào những mục đích khác nhau.

a. Định dạng ảnh TIFF

Định dạng TIFF (Tagged Image File Format) được nghiên cứu và giới thiệu vào năm 1986 bởi công ty Aldus Corp. – là một định dạng file ảnh chất lượng cao và được sử dụng nhiều trong các ứng dụng thu nhận ảnh từ máy quét (scan). Chuẩn định dạng TIFF là một trong những tiêu chuẩn quan trọng, được sử dụng nhiều trong ngành công nghiệp in ấn và xuất bản. File ảnh dạng TIFF thường có kích thước lớn hơn nhiều so với các file ảnh nén theo chuẩn JPEG. Định dạng TIFF lưu trữ dữ liệu hình ảnh nén hoặc không nén và có thể sử dụng các kỹ thuật nén không mất dữ liệu hoặc mất thông tin. Khác với định dạng JPEG, định dạng TIFF có thể có độ sâu màu từ 8 bits/channel đến 16 bits/channel và có thể có nhiều lớp ảnh được lưu trữ đồng thời trong cùng file ảnh TIFF. Định dạng TIFF thường có các kiểu nén là LZW, ZIP và JPGE.

Đặc điểm của ảnh theo định dạng TIFF là thường không bị mất dữ liệu hình ảnh khi lưu trữ ra thiết bị nhớ và đọc lại để xử lý trong máy tính, thường được sử dụng để biểu diễn ảnh có màu sắc phức tạp. Ảnh định dạng TIFF sử dụng trong các trường hợp đòi hỏi chất lượng cao như in ấn, phân tích mẫu.

b. Định dạng ảnh GIF

Định dạng GIF (graphics interchange format) được phát triển từ năm 1987, thường được dùng trong biểu diễn và truyền hình ảnh trong môi trường web. Ảnh định dạng GIF thường biểu diễn hình ảnh thành các frame để tạo ảnh chuyển động. Với mục đích tạo ra định dạng trao đổi hình ảnh nên các file ảnh theo định dạng GIF thường có kích thước nhỏ, chất lượng hình ảnh vừa phải, đáp ứng được môi trường mạng. Khác với JPEG, GIF sử dụng thuật toán nén ít mất thông tin (lossless) mà không làm giảm chất lượng hình ảnh sau khi nén. Trong kỹ thuật nén ảnh theo chuẩn GIF, dữ liệu lưu bằng cách sử dụng màu chỉ mục (index), mỗi hình ảnh có thể bao gồm 256 màu.

Một trong những ưu điểm của GIF là nén theo chuẩn lossless nên ảnh thường không bị mất dữ liệu khi nén, hình ảnh dạng GIF được tự động nhận biết trên hầu hết các trình duyệt web. Vì chuẩn GIF lưu trữ dữ liệu theo bảng chỉ mục nên nó thường được dùng để tạo các khung nhìn khác nhau tạo nên hiệu ứng chuyển động, vì hình ảnh giữa các frame có mức độ tương tự cao nên sẽ tiết kiệm được không gian nhớ so với video thông thường. Ảnh GIF sử dụng tốt đối với các trường hợp biểu diễn hình ảnh đơn giản như những bản vẽ chỉ có nét, bảng màu sắc và những minh họa đơn giản, tạo những hình ảnh động, hình ảnh web không có quá nhiều màu sắc, những ảnh avatar có kích thước nhỏ. Hình mô phỏng về hình ảnh chuyển động của hai con lắc minh họa thí nghiệm của Newton được tạo thành từ các ảnh đơn lẻ. Phần lớn các đối tượng đều không thay đổi, chỉ có hai quả cầu ở hi bên ngoài cùng chuyển động luân phiên nhau. Các ảnh này được nén theo chuẩn GIF cho ảnh chất lượng cao trong khi dung lượng file ảnh không

tăng nhiều so với kích thước của một ảnh đơn lẻ vì phần lớn dữ liệu đều giống nhau, chỉ một vài chi tiết nhỏ thay đổi, do vậy bảng chỉ mục nhỏ chỉ cần tham chiếu đến các frame.

c. Định dạng ảnh JPG

Định dạng JPG được đề xuất năm 1992 trong công bố của tác giả Haines. Định dạng JPG được gắn liền với chuẩn nén ảnh JPEG (Joint Photographic Experts Group) và lưu trữ trong máy tính theo file JPG. Định dạng JPG gắn liền với thuật toán nén mật thông tin (lossy), tức là khi nén dữ liệu để lưu trữ, thông tin sẽ bị mất trong quá trình nén và giải nén. Do đó, chất lượng hình ảnh sẽ bị giảm so với ảnh ban đầu. Tuy nhiên, với phương pháp nén mật thông tin thì kích thước file lưu trữ của ảnh cũng giảm đáng kể. Phương pháp nén JPEG thường được dùng để nén ảnh số có mất mát thông tin. Các file ảnh dùng nén theo chuẩn JPEG thường có tên mở rộng là *.jpg, *.jpeg, *.jfif, *.jpe.

Thông thường, định dạng JPG dùng 24 bit để biểu diễn màu với mỗi kênh màu chiếm 8 bit (1 byte). Như vậy, ảnh JPG 24 bit có thể biểu diễn được hơn 16,7 triệu màu khác nhau. Dung lượng lưu trữ file ảnh nhỏ hơn rất nhiều so với ảnh không nén (dạng bitmap). Các ảnh sử dụng phương pháp nén JPEG tương thích với hầu hết các trình duyệt web hiện nay. Ảnh JPG sử dụng tốt và hiệu quả đối với các loại ảnh tĩnh, ảnh có màu sắc phức tạp, ảnh đa mức xám, ảnh ngoại cảnh và ảnh chân dung.

d. Định dạng ảnh BMP

BMP là loại định dạng bitmap, được phát triển vào năm 1994. BMP là loại định dạng và lưu trữ file ảnh đồ họa lưới (raster) được sử dụng để lưu trữ ảnh số dạng thô. File ảnh dạng BMP thường có kích thước lớn và dữ liệu không nén do vậy cũng không mất thông tin trong quá trình lưu file và đọc ảnh từ file. Dữ liệu ảnh BMP độc lập với các thiết bị hiển thị như Graphics adapter, đặc biệt trên các ứng dụng chạy trong môi trường Microsoft Windows và hệ điều hành OS/2.

Định dạng BMP có ưu điểm là không làm mất thông tin của ảnh đang xử lý, nên nó phù hợp cho việc in ấn, chỉnh sửa hình ảnh. Mặt khác, vì ảnh không nén nên file ảnh BMP được đọc dễ dàng bằng các chương trình phần mềm dùng chung với những thuật toán đơn giản. Tuy nhiên, ảnh không hỗ trợ nén cũng ảnh hưởng không tốt cho việc lưu trữ vì dung lượng file thường lớn hơn các loại định dạng khác.

e. Định dạng ảnh PNG

PNG (Portable Network Graphics) được đề xuất năm 1996 là một định dạng file đồ họa dạng raster. PNG hỗ trợ nén dữ liệu không bị mất thông tin (lossless – ít mất thông tin). Định dạng PNG được xem là một dạng cải tiến và thay thế cho GIF trong môi trường ảnh vector và được sử dụng nhiều trên internet. Chuẩn định dạng PNG thường sử dụng hai dạng khác nhau là PNG-8 và PNG-24. Trong trường hợp ảnh có màu sắc phức tạp, không phân bố theo dạng vector thì PNG có dung lượng lớn hơn ảnh JPEG.

Ưu điểm của định dạng PNG là hình ảnh các đối tượng không bị cạnh răng cưa khi phóng to ảnh, điểm ảnh được biểu diễn dạng vector. Ảnh định dạng PNG được nén theo chuẩn không mất thông tin do vậy khi giải nén ảnh vẫn giữ nguyên được chất lượng ban đầu trước khi nén.

Ảnh dạng PNG thích hợp với các loại hình ảnh chứa đối tượng phân phối màu đơn giản, tuân theo quy luật như văn bản, các loại hình vẽ. Với các loại hình ảnh mà nền trong suốt hoặc có thể được thiết lập giữa mờ đặc lưu trữ theo định dạng PNG cho ảnh chất lượng cao với kích thước file nhỏ. Bên cạnh đó, nó cũng được dùng trong quá trình chỉnh sửa hình ảnh nhằm không làm mất thông tin của ảnh đang xử lý. Ngoài ra, định dạng PNG sử dụng tốt cho các hình ảnh web/blog, những mảng màu phẳng, thiết kế logo, hình ảnh có nền trong suốt hoặc bán trong suốt.

f. So sánh các chuẩn định dạng

Bảng 1.1. So sánh các chuẩn định dạng ảnh

Định dạng	Nén không mất thông tin	Cấu trúc lưu trữ	Chỉ mục màu	Hỗ trợ ảnh trong suốt	Nhiều trang	Ảnh động	Quản lý màu
BMP	x	Raster	x	x			x
GIF	x	Raster	x	x	x	x	
JPEG		Raster					x
PNG	x	Raster	x	x			x
TIFF	x	Cả 2	x	x	x		x

Không có một định dạng ảnh phổ biến nào có thể dùng tốt nhất cho tất cả các trường hợp. Mỗi loại định dạng ảnh có ưu và nhược điểm riêng, người dùng có thể lựa chọn định dạng phù hợp với bài toán của mình:

Định dạng JPEG: Thích hợp với các hình ảnh trên Web/Blog.

Định dạng GIF: Thích hợp với hình ảnh web, hình ảnh động và clip ART.

Định dạng PNG: Thích hợp với hình ảnh web, thiết kế Logo & Line ART.

Định dạng TIFF: Thích hợp cho việc in ấn.

Định dạng BMP: Thích hợp cho việc in ấn.

1.2. Bài toán nhận dạng khuôn mặt [2]

1.2.1. Tổng quan

Nhận dạng khuôn mặt là một trong những vấn đề quan trọng trong hướng nghiên cứu về nhận dạng của ngành thị giác máy tính. Do tính giống nhau của khuôn mặt nên việc trích ra các đặc trưng của khuôn mặt dùng cho nhận dạng là rất khó. Trong các đặc trưng của khuôn mặt dùng để nhận dạng thì đặc trưng về cạnh là một đặc trưng chỉ mới được nghiên cứu và phát triển trong những năm gần đây. Phần này trình bày một hướng nghiên cứu nhận dạng khuôn mặt dựa trên bản đồ cạnh (edge map) của khuôn mặt. Việc tính toán sự trùng khớp sẽ dựa trên khoảng cách Hausdorff. Các mô phỏng sẽ so sánh sự chính xác của việc nhận dạng khuôn mặt dựa vào bản đồ cạnh, với phương pháp rất phổ biến của nhận dạng khuôn mặt là Eigenface. Các kết quả cũng chỉ ra rằng việc nhận dạng khuôn mặt dựa vào bản đồ cạnh cho kết quả nhận dạng chính xác cao hơn phương pháp Eigenface trong hầu hết các so sánh.

Tự động nhận dạng khuôn mặt là một hướng nghiên cứu thú vị đã thu hút được rất đông các nhà nghiên cứu trong khoảng hơn 20 năm qua. Từ khi bắt đầu ra đời đến

nay, hướng nghiên cứu về nhận dạng khuôn mặt đã thu hút được rất nhiều nhà nghiên cứu trên toàn thế giới. Chính vì là một hướng nghiên cứu thu hút nên đã có rất nhiều phương pháp khác nhau về nhận dạng khuôn mặt đã được đề xuất. Các nghiên cứu về nhận dạng khuôn mặt có thể được chia thành các nhóm chính sau: Eigenface, mạng nơ-ron, mô hình Markov ẩn, nhận dạng dựa vào các đặc trưng hình học (geometrical feature matching) và nhận dạng mẫu (template matching). Trong khi các hướng nghiên cứu về mạng nơ-ron và template matching cho tỷ lệ nhận dạng chính xác cao, nhưng yêu cầu phải có nhiều ảnh làm cơ sở dữ liệu cho cùng một đối tượng, thì sẽ không thích hợp với các ứng dụng mà chỉ có một ảnh cho một đối tượng để nhận dạng. Trong khi đó, các nghiên cứu dựa vào mô hình Markov ẩn và đặc trưng hình học thì tỷ lệ chính xác lại phụ thuộc rất nhiều vào việc chọn các thông số huấn luyện cũng như thời gian nhận dạng khá lớn. Eigenface tuy là một hướng nghiên cứu đã lâu nhưng lại đơn giản, cho kết quả nhận dạng chính xác tương đối và dễ dàng áp dụng cho các ứng dụng đòi hỏi chỉ có một ảnh làm cơ sở dữ liệu cho một đối tượng. Trong vấn đề nhận dạng, cạnh cũng là một đặc trưng rất hay được sử dụng. Tuy nhiên trong nhận dạng khuôn mặt, đặc trưng cạnh của khuôn mặt vẫn không được sử dụng khi nghiên cứu. Takács [4] là người đầu tiên sử dụng đặc trưng cạnh của khuôn mặt trong việc nhận dạng khuôn mặt. Tuy nhiên các nghiên cứu của Takács và sau này chỉ dựa vào các điểm trên cạnh của khuôn mặt nên không cho thông tin chính xác cao về khuôn mặt. Y. Gao và K. H. Leung [5] đã đưa ra một phương pháp nhận dạng khuôn mặt dựa trên các đường trong bản đồ cạnh của khuôn mặt (Line Edge Map – LEM). Phương pháp này cho tỷ lệ nhận dạng chính xác rất cao, và cũng cho thấy sự bền vững của việc nhận dạng khuôn mặt trong các điều kiện khác nhau về ánh sáng, cũng như việc thay đổi cảm xúc trên khuôn mặt.

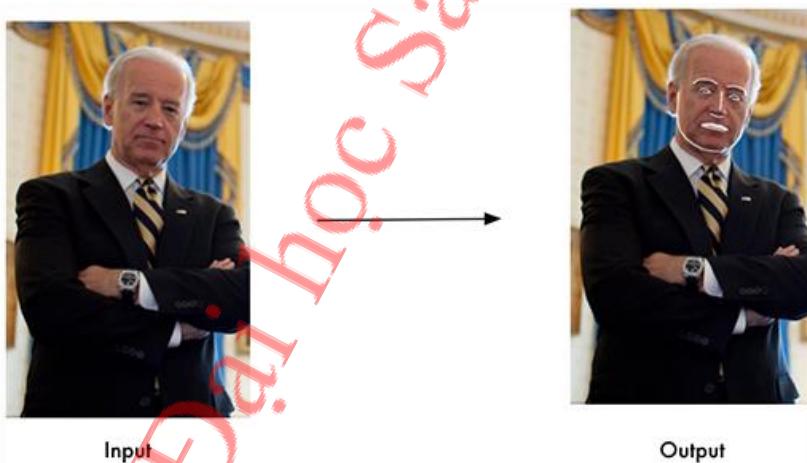
1.2.2. Nhận dạng khuôn mặt dựa vào bản đồ cạnh[2]

Trong vấn đề xử lý ảnh, cạnh (edge) được định nghĩa là sự thay đổi độ sáng đột ngột giữa các pixel, là kết quả của sự thay đổi về cấu trúc hình học của các thành phần trong vật thể. Vì thế, cạnh cũng là một đặc trưng quan trọng và được dùng để nhận dạng. Có rất nhiều phương pháp tìm cạnh trong một bức ảnh đã được đề xuất. Mỗi phương pháp tìm cạnh sẽ có ưu và nhược điểm khác nhau. Heath [6] đã chỉ ra rằng không có một phương pháp tìm cạnh nào là ưu điểm tuyệt đối và rất khó để có thể tìm được một phương pháp tìm cạnh tốt nhất cho một bức ảnh bất kỳ. Để tạo ra các bản đồ cạnh cho một bức ảnh khuôn mặt, Badu [7] sử dụng phương pháp tìm cạnh kết hợp với phương pháp làm mỏng cạnh để tạo ra các cạnh có độ dày 1 pixel của bức ảnh. Tuy nhiên số pixel của cạnh vẫn còn rất lớn, dẫn tới việc tăng khối lượng tính toán không cần thiết. Áp dụng thuật toán Dynamic-two-Strip Dyn2S [8] trên bản đồ cạnh này ta sẽ tìm được các điểm trội. Các điểm trội này có độ uốn cong lớn nhất. Thuật toán Dyn2S [8] được mô tả: Tại mỗi điểm trên đường cong, sử dụng hai dải chữ nhật ở bên trái và bên phải của điểm đó. Các điểm còn lại trên đường cong nếu nằm trong hai dải đó được xem xấp xỉ như trên cùng một đường thẳng với điểm đang xét. Nếu ta đặt tỉ số của chiều dài và chiều rộng của mỗi dải là E , góc tạo bởi giữa hai dải là θ thì tại mỗi điểm, ta sẽ có một

chỉ số merit được tính bởi $W = E_{left} \cdot S_{Eright}$, trong đó $S = |180\theta - \theta|$, E_{left} và E_{right} là tỉ số chiều dài và rộng của hai dải bên trái và phải của điểm chúng ta đang xét. Chiều dài của các dải là như nhau. Các dải sẽ được thay đổi chiều rộng trong một khoảng giới hạn và thay đổi góc quay để làm sao có được nhiều điểm nằm trong dải nhất. Tại tất cả các điểm trên cạnh, ta đều làm tương tự. Sau đó, các điểm trội trên cạnh sẽ được chọn như sau: i) những điểm có chỉ số W nhỏ hơn hai điểm bên cạnh sẽ được bỏ đi; ii) với những điểm còn lại, những điểm nào có thể được xấp xỉ là một đường thẳng với điểm đang xét thì ta bỏ đi, chỉ giữ lại điểm cuối của hai dải bên trái và bên phải. Như vậy, sau khi kết hợp giữa thuật toán tìm cạnh và Dyn2S lên một bức ảnh khuôn mặt, ta sẽ được bản đồ cạnh như hình 1.4



Hình 1.4. Bản đồ cạnh (edge map) của khuôn mặt



Hình 1.5. Kết quả của bản đồ cạnh

1.2.3. Khoảng cách Hausdorff

Khoảng cách Hausdorff là khoảng cách được dùng để tính cho khoảng cách giữa hai tập hợp điểm, nhưng không cần xét đến sự tương ứng điểm – điểm giữa hai tập hợp như các khoảng cách khác. Huttenlocher [9] đã ứng dụng khoảng cách Hausdorff để so sánh sự giống nhau giữa các bức ảnh. Cho tập hợp các điểm trên bản đồ cạnh của một bức ảnh trong cơ sở dữ liệu là $C = \{c_1, c_2, \dots, c_p\}$ và của một bức ảnh cần nhận dạng là $N = \{n_1, n_2, \dots, n_m\}$. Khoảng cách Hausdorff của hai tập điểm được định nghĩa:

$$H(C, N) = \max(h(C, N), h(N, C))$$

trong đó:

$$h(C, N) = \max_{c_i \in C} \min_{n_i \in N} \|c_i - n_i\|$$

và $\|ci - nj\|$ là khoảng cách Euclid giữa hai điểm ci và nj . Khoảng cách $h(C, N)$ được gọi là khoảng cách Hausdorff từ ảnh C đến ảnh N . Như vậy, với cách định nghĩa trên thì khoảng cách Hausdorff giữa hai bức ảnh $H(C, N)$ chính là chỉ số dùng để đo sự khác nhau giữa hai bức ảnh. Tuy nhiên với cách định nghĩa trên, khoảng cách Hausdorff rất nhạy với các điểm đặc biệt trong bức ảnh. Ví dụ, nếu vì một lý do nào đó mà trong bản đồ cạnh có vài điểm, hoặc thậm chí một điểm nằm cách biệt ra ngoài, khác với những điểm khác; thì với cách định nghĩa của khoảng cách Hausdorff như phương trình (1) và (2) thì khoảng cách này sẽ do điểm đặc biệt đó quyết định. Như vậy, hai vật thể hoặc khuôn mặt có hình dáng rất khác nhau, nhưng nếu có một vài hoặc một điểm đặc biệt thì sẽ trở nên rất giống nhau, nếu chỉ xét khoảng cách Hausdorff như trên. Dubuisson và Jain [12] đã chỉ ra một phương pháp cải tiến của khoảng cách Hausdorff (Modified Hausdorff Distance – MHD) để loại bỏ nhược điểm của phương pháp tính khoảng cách Hausdorff trực tiếp như phương trình (2). Phương pháp MHD này đã được Takács [4] ứng dụng vào việc tính khoảng cách giữa các bản đồ cạnh của khuôn mặt trong việc nhận dạng khuôn mặt. Tuy nhiên, bản đồ cạnh của khuôn mặt được Takács sử dụng để tính toán chỉ là cạnh của bức ảnh khi được áp dụng phương pháp tìm cạnh kết hợp với phương pháp làm mỏng cạnh, chứ chưa sử dụng thuật toán Dyn2S để tìm các điểm trội như đã nói ở trên. Khoảng cách MHD được định nghĩa như sau:

$$h_{MHD}(C, N) = \frac{1}{P} \sum_{c_i \in C} \min_{n_i \in N} \|c_i - n_i\|$$

Trong đó P là số điểm trong C . Với cách định nghĩa này sẽ làm giảm sự tác động của các điểm đột biến đến khoảng cách giữa hai bức ảnh. Gao [10] đã đề xuất một phương pháp khác cũng dựa trên việc tính toán khoảng cách Hausdorff để nhận dạng ảnh. Trong phương pháp này, Gao đã sử dụng thuật toán Dyn2S để tìm các điểm trội trên cạnh khuôn mặt. Chính việc dùng thuật toán Dyn2S đã làm giảm số lượng điểm trên cạnh rất nhiều, dẫn đến việc làm giảm khối lượng tính toán. Trong phương pháp của mình, Gao đã định nghĩa một khoảng cách Hausdorff từ ảnh C đến ảnh N như sau:

$$h_{MHD}(C, N) = \frac{1}{\sum_{c_i \in C} W_{c_i n_j}} \sum_{c_i \in C} W_{c_i n_j} \min_{n_j \in N} \|c_i - n_j\|$$

Với $W_{cinj} = 1/2 (Wci + Wnj)$ là trung bình của chỉ số merit tại hai điểm ci và nj trong thuật toán Dyn2S. Như vậy, khoảng cách Hausdorff của hai bức ảnh sẽ là:

$$H_{MMHD}(C, N) = \max(h_{MMHD}(C, N), h_{MMHD}(N, C))$$

Dựa trên kết quả các mô phỏng trong [10] của Gao được thực hiện trên cơ sở dữ liệu AR và Bern, các kết quả cho thấy rằng tỷ lệ chính xác của phương pháp MHD của Takács và MMHD của Gao là tương đương nhau. Như vậy, với đề xuất phương pháp tính khoảng cách Hausdorff như tại phương trình (4) và (5) thì không làm tăng thêm tính chính xác của thuật toán so với phương trình (3). Vậy, đóng góp của Gao trong bài báo này thực chất là việc áp dụng thuật toán Dyn2S vào bản đồ cạnh để làm giảm bớt đi số

điểm cần tính toán, dẫn đến làm giảm khối lượng tính toán mà không làm giảm đi độ chính xác. Việc áp dụng thuật toán Dyn2S làm giảm đến 80% số điểm cần tính toán trên bản đồ cạnh. Vì tỷ lệ chính xác của MHD và MMHD như nhau nên trong các mô phỏng của bài báo này, em sẽ sử dụng thuật toán MHD, nhưng trên bản đồ cạnh khuôn mặt đã được áp dụng thuật toán Dyn2S như hình 1.14.

Một nhược điểm của thuật toán MHD và MMHD là việc coi tất cả các điểm trên bản đồ cạnh là các điểm độc lập, vì vậy, hai điểm cạnh nhau cũng giống như hai điểm nằm xa nhau và không có thông tin về sự liên hệ giữa các điểm với nhau. Gao và Leung [5] cho rằng các điểm trên cùng một cạnh được nối với nhau bằng một đường thẳng. Như vậy, thay vì dùng khoảng cách Hausdorff giữa hai tập hợp điểm, thì Gao đã đưa ra cách tính khoảng cách Hausdorff giữa hai tập hợp đường, được gọi là Line Segment Hausdorff Distance (LHD). Khác với các phương pháp khác sử dụng cách tính khoảng cách giữa tập hợp các đường, giống như MHD và MMHD, phương pháp LHD không phụ thuộc vào sự tương ứng giữa các đường – đường ~~trong~~ ảnh cơ sở dữ liệu và ảnh cần nhận dạng. Chính vì ưu điểm này mà phương pháp LHD vẫn có thể được dùng ngay cả khi một bức ảnh bị mất một số cạnh do lỗi trong quá trình phân đoạn ảnh. Cho tập hợp các đường trên bản đồ cạnh của một bức ảnh trong cơ sở dữ liệu là $Cl = \{c_1^l, c_2^l, \dots, c_p^l\}$ và t của một bức ảnh cần nhận dạng là $Nl = \{n_1^l, n_2^l, \dots, n_m^l\}$. Khoảng cách giữa 2 đường bất kỳ được định nghĩa như sau:

$$d(c_i^l, n_j^l) = \sqrt{d_0^2(c_i^l, n_j^l) + d_{ss}^2(c_i^l, n_j^l) + d_{vg}^2(c_i^l, n_j^l)}$$

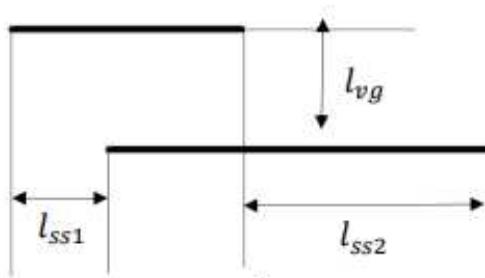
Trong đó: $d\theta(c_i^l, n_j^l)$, $dss(c_i^l, n_j^l)$, $dvg(c_i^l, n_j^l)$ lần lượt là khoảng cách góc, khoảng cách song song và khoảng cách vuông góc của hai đường c_i^l và n_j^l . Trong đó, khoảng cách góc được định nghĩa là:

$$d_0(c_i^l, n_j^l) = \theta^2(c_i^l, n_j^l)/W$$

Với $\theta(c_i^l, n_j^l)$ là góc giao nhau nhỏ nhất giữa hai đường c_i^l và n_j^l . Hệ số W là một trọng số tùy chọn và được xác định trong quá trình huấn luyện tập ảnh. Để tính khoảng cách song song và vuông góc giữa hai đường thẳng, chúng ta sẽ xoay đường thẳng có độ dài ngắn hơn để song song với đường thẳng dài hơn. Khi hai đường thẳng song song như Hình 2, khoảng cách song song và vuông góc sẽ được tính như sau:

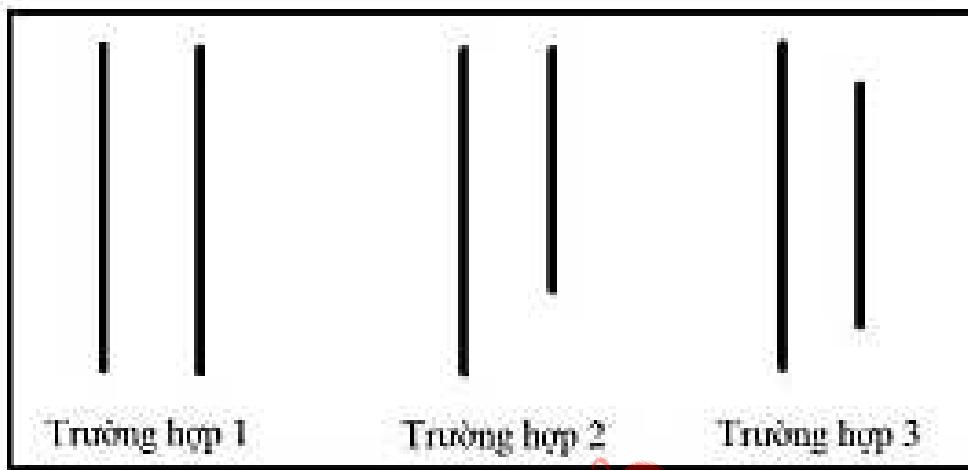
$$d_{ss}(c_i^l, n_j^l) = \min(l_{ss1}, l_{ss2})$$

$$d_{vg}(c_i^l, n_j^l) = l_{vg}$$



Hình 1.6. Khoảng cách giữa hai đường thẳng song song

Khoảng cách song song sẽ được tính là khoảng cách nhỏ nhất giữa điểm ngoài cùng bên trái và ngoài cùng bên phải của hai đường thẳng. Khoảng cách song song sẽ bằng 0 khi hai đường thẳng thuộc một trong ba trường hợp được mô tả trong hình 1.7



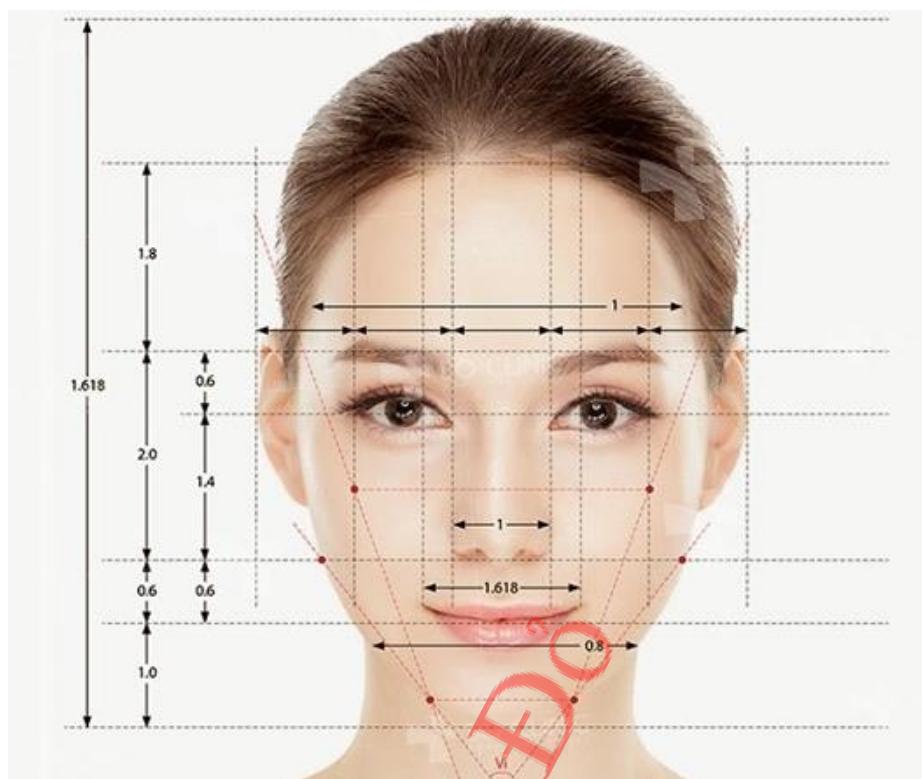
Hình 1.7. Các trường hợp $d_{ss} = 0$

$$H_{pLHD}(C^1, N^1 = \max(h(C^1, N^1), h(N^1, C^1))$$

$$h(C^1, N^1) = \frac{1}{\sum_{c_i^1 \in C^1} \sum_{c_j^1 \in C^1}} \sum_{c_i^1 \in C^1} \min_{n_j^1 \in N^1} d(c_i^1, n_j^1)$$

Tuy nhiên, với khoảng cách pLHD được tính như phương trình (11) thì có một điểm yếu. Giả sử chúng ta có $N l$ là LEM của khuôn mặt cần nhận dạng, $nj l$ là một đường trong LEM đó; $Cg l$ và $Ck l$ lần lượt là LEM của khuôn mặt nhận dạng đúng của $N l$ và khuôn mặt khác với $N l$ trong cơ sở dữ liệu. Nếu vì một lý do nào đó mà đường tương ứng với $nj l$ trong $Cg l$ là $c_{gj} l$ bị mất đi, khi đó, đường $nj l$ trong $N l$ sẽ có khoảng cách gần nhất tới một đường khác trong $Cg l$, giả sử là đường $c_{gi} l$. Khi đó khoảng cách $d(c_{gi} l, nj l)$ có thể sẽ lớn hơn rất nhiều so với $d(c_{kj} l, nj l)$, với $c_{kj} l$ là đường trong $Ck l$ có khoảng cách ngắn nhất tới $nj l$. Điều này dẫn tới việc $N l$ sẽ có khoảng cách tới $Ck l$ gần hơn so với $Cg l$ và dẫn tới nhận dạng sai.

Để khắc phục điều này, Gao đưa thêm vào một thông số nữa vào khoảng cách Hausdorff, đó là tỉ số tin cậy. Nếu một đường thẳng $ci l$ trong LEM $C l$ có khoảng cách gần nhất tới đường thẳng $nj l$ trong LEM $N l$, và hai đường nào có khoảng cách góc nhỏ hơn một lượng Kg , và khoảng cách giữa hai trung điểm của hai đường thẳng nhỏ hơn một lượng Kvt , thì khi đó đường $nj l$ được xem là tin cậy với đường $ci l$. Khi đó, tỉ số tin cậy của một bức ảnh được định nghĩa là tỉ số giữa tổng số đường tin cậy Dtc và tổng số đường trong LEM $Dtotal$ của bức ảnh.



Hình 1.8. Khoảng cách Hausdorff

1.2.4. Face Recognition

Face Recognition là bài toán nhận dạng và xác thực người dựa vào khuôn mặt của họ. Đối với con người thì đó là một nhiệm vụ rất đơn giản, thậm chí là ở trong những điều kiện môi trường khác nhau, tuổi tác thay đổi, đội mũ, đeo kính, ... Tuy nhiên, đối với máy tính thì nó vẫn còn là một thử thách khó khăn trong vài thập kỷ qua cho đến tận ngày nay. Face Recognition có thể chia thành 3 bài toán nhỏ:

- Face Authentication: Hạn chế quyền truy cập của một người đến một nguồn tài nguyên nào đó.
- Face Verification: Xác nhận một người phù hợp với ID của họ.
- Face Identification: Gán chính xác tên của người.

Tất cả những bài toán này đều được giải quyết trong cả 3 trường hợp:

- Người trong ảnh
- Người trong file video
- Người thực (stream real-time từ camera)

Luồng xử lý của bài toán Face Recognition

Bài toán Face Recognition bắt buộc phải bao gồm tối thiểu 3 bước sau:

Bước 1: Face Detection - Xác định vị trí của khuôn mặt trong ảnh (hoặc video frame). Vùng này sẽ được đánh dấu bằng một hình chữ nhật bao quanh.

Bước 2: Face Extraction (Face Embedding) - Trích xuất đặc trưng của khuôn mặt thành một vector đặc trưng trong không gian nhiều chiều (thường là 128 chiều).

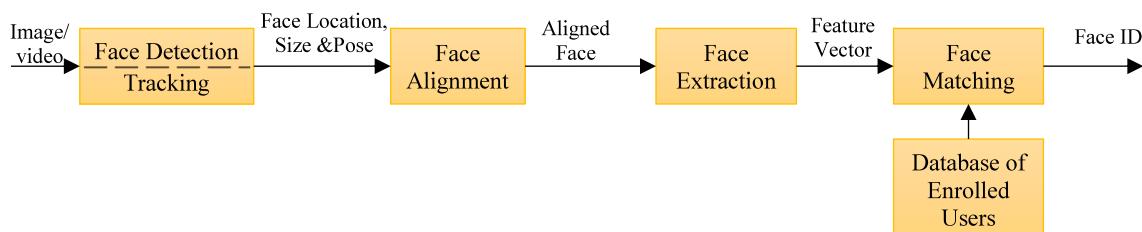
Bước 3: Face Classification (Face Authentication - Face Verification - Face Identification).

Ngoài 3 bước trên, trong thực tế chúng ta thường bổ sung thêm một số bước để tăng độ chính xác nhận diện:

Image Preprocessing: Xử lý giảm nhiễu, giảm mờ, giảm kích thước, chuyển sang ảnh xám, chuẩn hóa, ...

Face Alignment: Nếu ảnh khuôn mặt bị nghiêng thì căn chỉnh lại cho ngay ngắn.

- Kết hợp nhiều phương pháp khác nhau tại bước 3.

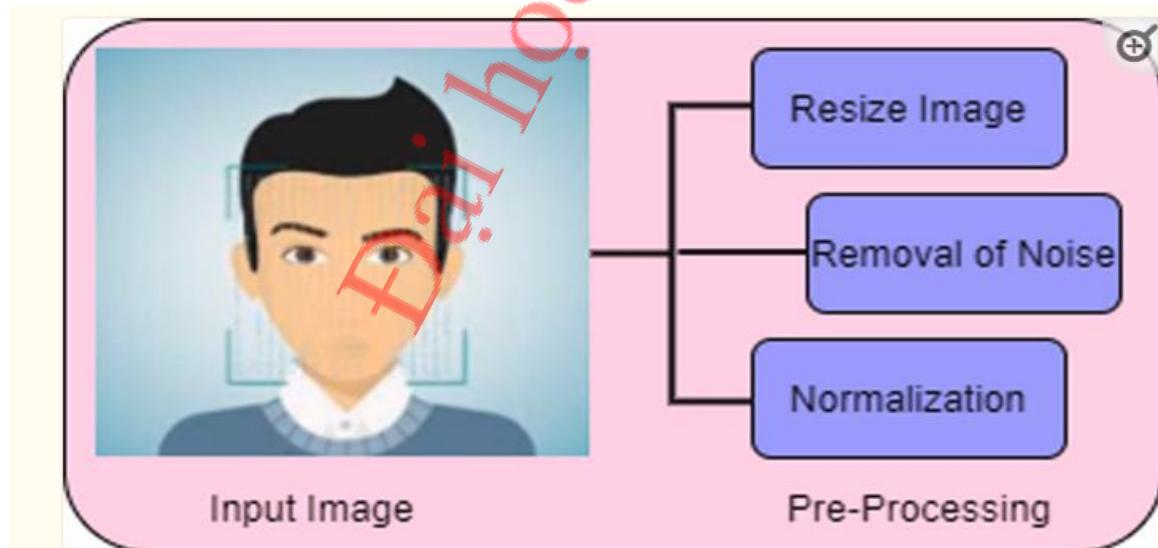


Hình 1.9. Luồng xử lý của Face Recognition

Face Detection

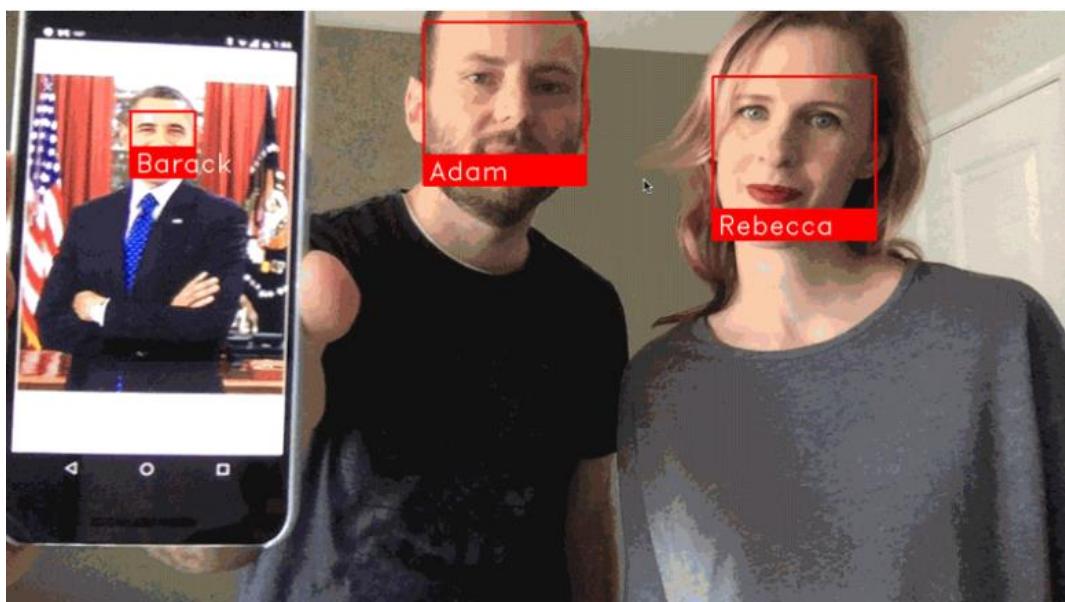
Face Detection là bước đầu tiên trong bài toán Face Recognition, có vai trò rất lớn trong việc nâng cao độ chính xác của toàn bộ hệ thống. Đầu vào của nó là một bức ảnh có chứa mặt người, đầu ra của nó sẽ là các tọa độ của vùng chứa khuôn mặt, thường thể hiện bằng một hình chữ nhật bao quanh khuôn mặt đó. Có 2 phương pháp tiếp cận để giải quyết vấn đề:

Feature-based: Sử dụng các bộ lọc thủ công (hand-crafted filters) để tìm kiếm và định vị vị trí khuôn mặt trong ảnh. Phương pháp này rất nhanh và hiệu quả trong điều kiện gần lý tưởng, nhưng không hiệu quả trong điều kiện phức tạp hơn.



Hình 1.10. Tiết xú xử lý

Điều kiện gần lý tưởng: Trường hợp đủ ánh sáng, các khuôn mặt được nhìn chính diện, rõ các phần trên khuôn mặt. Việc nhận dạng khuôn mặt có độ chính xác cao.



Hình 1.11. Nhận diện khuôn mặt trong điều kiện lý tưởng

Trong điều kiện phức tạp hơn: Khi khuôn mặt có phụ kiện như kính, khẩu trang, hoặc khuôn mặt bị che khuất một phần khi tay, điện thoại, đồ che mặt.



Hình 1.12. Nhận diện khuôn mặt trong điều kiện phức tạp

Image-based: Sử dụng các thuật toán deep learning để học và tự động định vị vị trí khuôn mặt dựa trên toàn bộ bức ảnh. Ưu điểm của phương pháp này là độ chính xác cao hơn so với phương pháp Feature-based, nhưng tốc độ thực hiện thì lại chậm hơn. Tùy theo điều kiện cụ thể của từng bài toán mà ta chọn phương pháp phù hợp.

Bảng 1.2. Bảng tổng hợp thư viện và thuật toán cho mỗi phương pháp

Phương pháp	Model/Thuật toán	Thư viện/Open Source
Feature - based	Haarcascade_frontalface_default.xml	OpenCV
	HOG	dlib/face_recognition
Image-based	CNN	dlib/face_recognition
	MTCNN	

Phương pháp	Model/Thuật toán	Thư viện/Open Source
	SSD,Resnet: Res10_300x300_ssd_iter_140000.caffemodel Deploy.proto.txt	Caffe, OpenCV

Phương pháp Image-based có sử dụng các thuật toán deep learning nên độ chính xác cao hơn so với phương pháp Feature-based nhưng đáp ứng thời gian thực thì không bằng khi chạy trên máy có cấu hình thấp.

1.3. Bài toán nhận diện khuôn mặt giả

1.3.1. Tổng quan

Hệ thống nhận dạng khuôn mặt đang trở nên phổ biến hơn bao giờ hết. Từ nhận dạng khuôn mặt trên iPhone/điện thoại thông minh đến nhận dạng khuôn mặt để giám sát hàng loạt ở Trung Quốc, các hệ thống nhận dạng khuôn mặt đang được sử dụng ở khắp mọi nơi.

Tuy nhiên, các hệ thống nhận dạng khuôn mặt dễ bị đánh lừa bởi các khuôn mặt "giả mạo" và "không thật".

Hệ thống nhận dạng khuôn mặt có thể bị phá vỡ chỉ bằng cách giữ ảnh của một người (cho dù được in, trên điện thoại thông minh, v.v.) vào camera nhận dạng khuôn mặt.

Để làm cho hệ thống nhận dạng khuôn mặt an toàn hơn, cần có ứng dụng có khả năng phát hiện khuôn mặt giả/không thật và thuật ngữ “phát hiện sự sống” được sử dụng để chỉ các thuật toán như vậy.

1.3.2. Các phương pháp phát hiện khuôn mặt giả [14]

Phân tích kết cấu, bao gồm tính toán các mẫu nhị phân cục bộ (LBP) trên các vùng khuôn mặt và sử dụng SVM để phân loại khuôn mặt là thật hoặc giả mạo.

Phân tích tần số, chẳng hạn như kiểm tra miền Fourier của khuôn mặt.

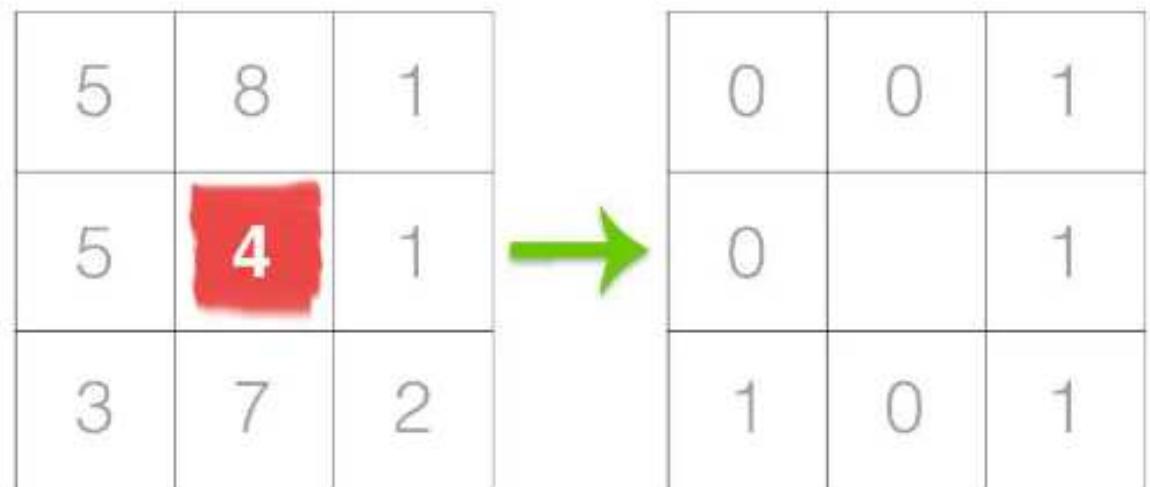
Phân tích lấy nét thay đổi, chẳng hạn như kiểm tra sự thay đổi của các giá trị pixel giữa hai khung hình liên tiếp.

Các thuật toán dựa trên heuristic, bao gồm chuyển động mắt, chuyển động môi và phát hiện chớp mắt. Bộ thuật toán này cố gắng theo dõi chuyển động của mắt và chớp mắt để đảm bảo người dùng không cầm ảnh của người khác (vì ảnh sẽ không chớp mắt hoặc di chuyển môi).

Các thuật toán luồng quang học, cụ thể là kiểm tra sự khác biệt và tính chất của luồng quang được tạo ra từ các vật thể 3D và mặt phẳng 2D.

Hình dạng khuôn mặt 3D, tương tự như những gì được sử dụng trên hệ thống nhận dạng khuôn mặt iPhone của Apple, cho phép hệ thống nhận dạng khuôn mặt phân biệt giữa khuôn mặt thật và bản in/ảnh/hình ảnh của người khác.

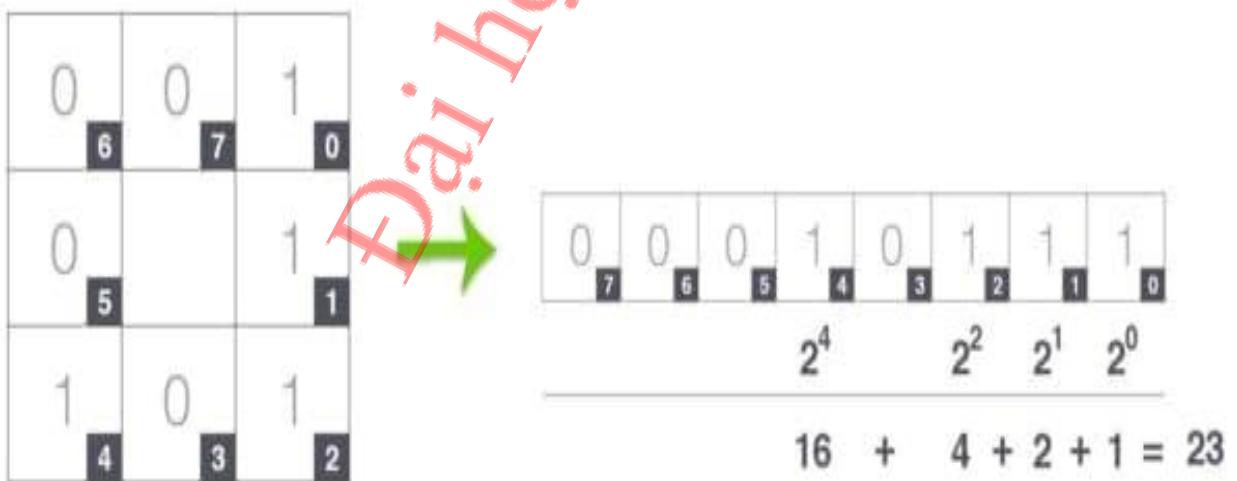
Sự kết hợp của những điều trên, cho phép một kỹ sư hệ thống nhận dạng khuôn mặt chọn và chọn các mô hình phát hiện sự sống phù hợp với ứng dụng cụ thể của họ.



Hình 1.13. Xây dựng một LBP (bước 1)

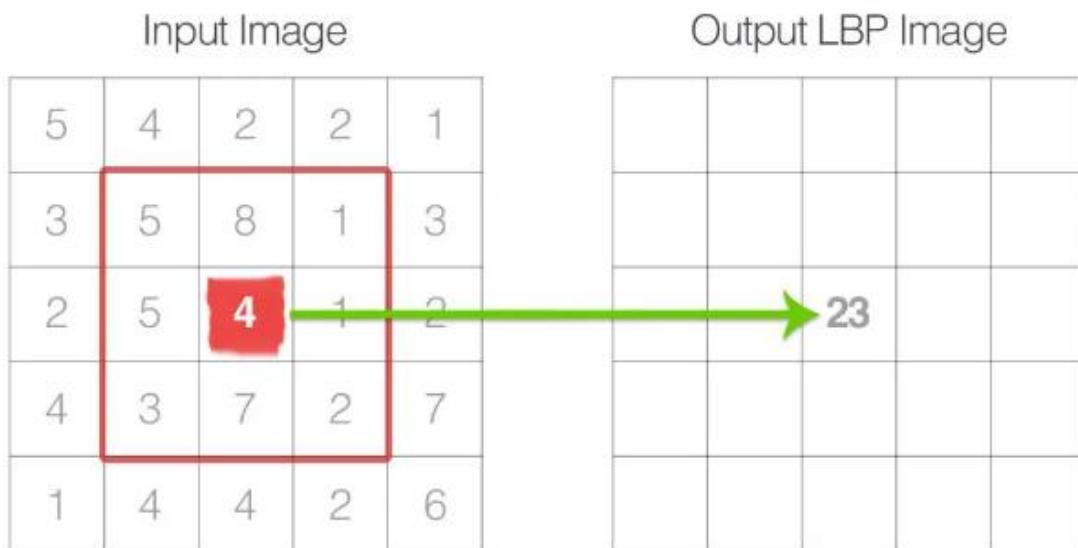
Hình 1.13 là minh họa bước 1 thực hiện xây dựng một LBP là sử dụng 8 láng giềng của điểm đang xét và đặt ngưỡng cho nó để xây dựng một tập hợp 8 chữ số nhị phân. Trong hình 1.13, ta lấy pixel trung tâm (được đánh số màu đỏ) và đặt ngưỡng cho vùng lân cận là 8 pixel. Nếu cường độ của pixel trung tâm lớn hơn hoặc bằng pixel lân cận thì đặt giá trị thành 1; nếu không, đặc nó thành 0. Với 8 pixel xung quanh, ta có tổng cộng $2^8=256$ tổ hợp mã LBP có thể có.

Ta cần tính giá trị LBP cho pixel trung tâm. Ta bắt đầu từ bất kỳ pixel lân cận nào và làm việc theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ, nhưng thứ tự phải giữ nhất quán cho tất cả các pixel trong hình ảnh và tất cả hình ảnh trong tập dữ liệu. Với một vùng lân cận 3x3, ta có 8 lân cận, kết quả của thử nghiệm nhị phân này được lưu trữ trong một mảng 8 bit, sau đó ta chuyển đổi sang thập phân:



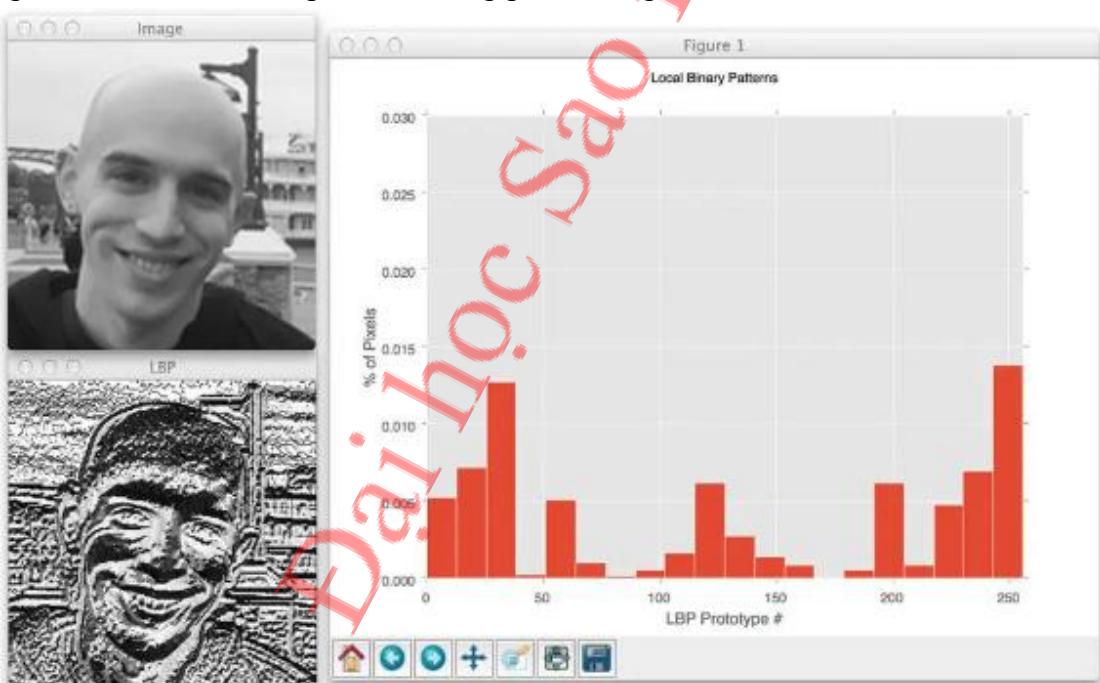
Hình 1.14. Biểu diễn thập phân của 8 liền kề điểm đang xét

Trong ví dụ này, ta bắt đầu ở điểm trên cùng bên phải và làm việc theo chiều kim đồng hồ để tích lũy chuỗi nhị phân khi ta tiếp tục. Sau đó ta có thể chuyển đổi chuỗi nhị phân này thành chuỗi thập phân có giá trị 23. Giá trị này được lưu trữ trong mảng LBP 2D đầu ra, sau đó lưu trữ trong ảnh đầu ra như hình 1.15.



Hình 1.15. Giá trị LBP sau được tính toán

Quá trình ngưỡng, tích lũy chuỗi nhị phân và lưu trữ giá trị thập phân đầu ra trong mảng LBP sau đó được lặp lại cho từng pixel trong ảnh đầu vào.

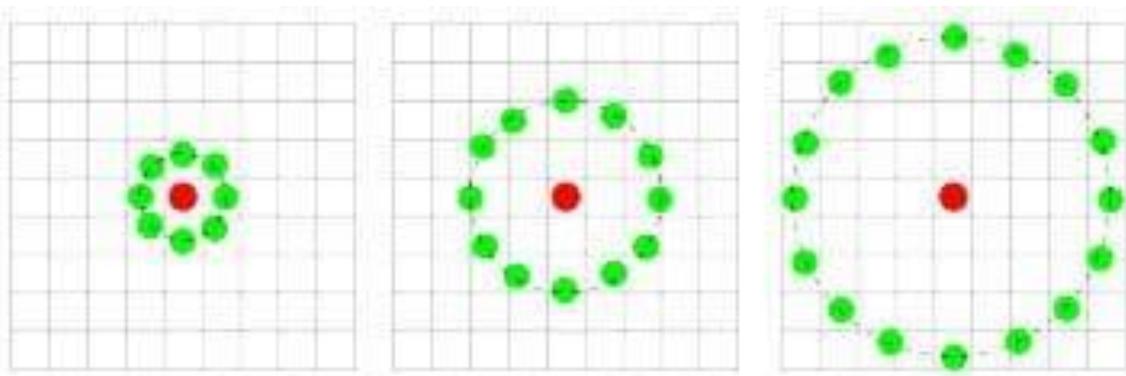


Hình 1.16. Biểu đồ bảng số lần mẫu LBP

Lợi ích chính của triển khai LBP ban đầu này là có thể chụp được các chi tiết cự mịn trong ảnh. Việc chụp được chi tiết ở quy mô nhỏ sẽ không chụp được chi tiết ở các tỷ lệ khác nhau. Để xử lý vấn đề này, Ojala cùng cộng sự đã đề xuất một phần mở rộng cho việc triển khai LBP ban đầu để xử lý các kích thước vùng lân cận thay đổi: Sử dụng hai tham số:

- Số điểm p trong một lân cận đối xứng tròn cần xem xét (loại bỏ phục thuộc vào lân cận vuông).
- Bán kính của đường tròn r cho phép tính các tỷ lệ khác nhau.

Hình 1.17 minh họa các tham số này.



Hình 1.17. Minh họa lân cận với p và r khác nhau để xây dựng LBP

Tính đồng nhất của LBP: Một LBP được coi là đồng nhất nếu có nhiều nhất hai lần chuyển tiếp 0-1 hoặc 1-0. Ví dụ, mẫu 00001000 (2 lần chuyển tiếp) và 10000000 (1 lần chuyển tiếp) đều được coi là các mẫu đồng nhất vì chúng chứa tối đa hai lần chuyển tiếp 0-1 và 1-0. Mẫu 01010010 không được coi là một mẫu đồng nhất vì nó có sáu lần chuyển đổi 0-1 hoặc 1-0.

Số lượng mẫu đồng nhất trong LBP phụ thuộc vào số điểm p . Khi giá trị của p tăng lên thì số chiều của biểu đồ cũng tăng theo. Với số điểm p trong LBP sẽ có $p+1$ mẫu đồng nhất. Do đó, chiều cuối cùng của biểu đồ là $p+2$, trong đó mục được thêm vào sẽ lập bảng tất cả các mẫu không đồng nhất.

Mẫu LBP đồng nhất bổ sung thêm một mức độ xoay và bất biến thang độ xám, do đó chúng được sử dụng khi trích xuất các vector đặc trưng LBP từ ảnh.

Việc triển khai LBP có thể sử dụng gói scikit-image và mahotas. OpenCV cũng triển khai LBP, nhưng chỉ trong bối cảnh nhận dạng khuôn mặt - trình trích xuất LBP cơ bản không được hiển thị để tính toán biểu đồ LBP thô. Việc triển khai LBP bằng gói scikit-image được đánh giá tốt nhất vì chúng cung cấp nhiều quyền kiểm soát hơn.

Chương 2. KỸ THUẬT TẠO VÀ PHÁT HIỆN KHUÔN MẶT GIẢ

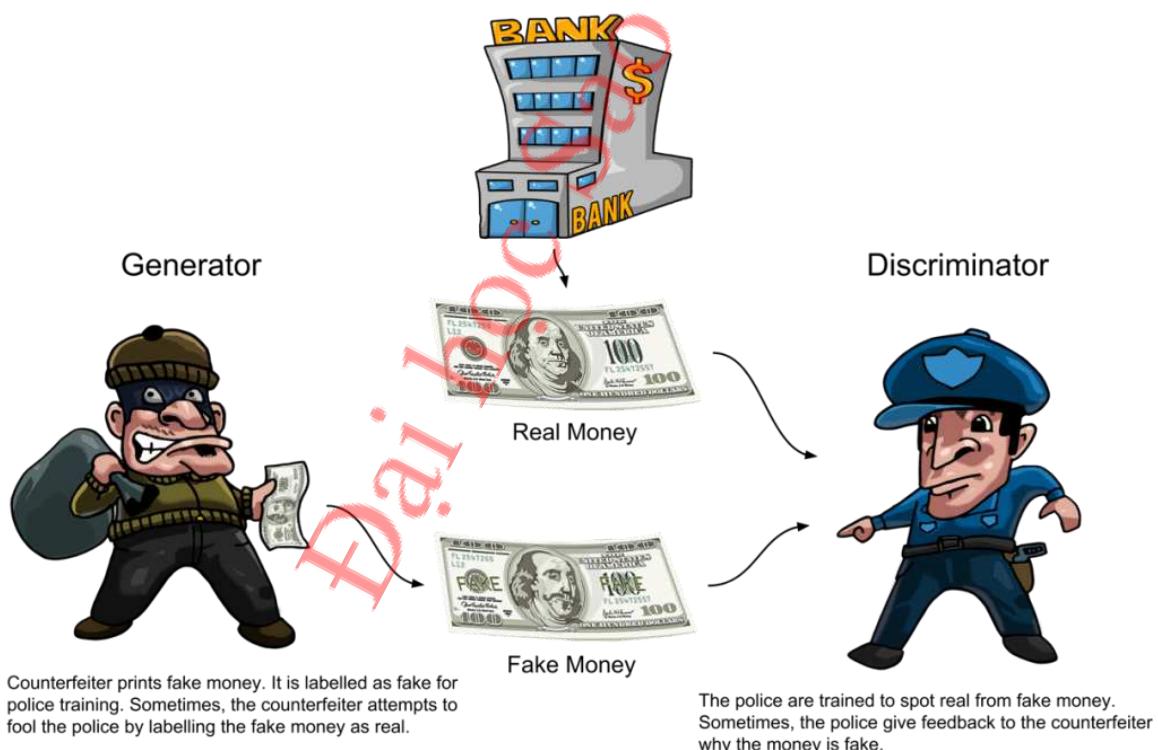
2.1. Kỹ thuật tạo khuôn mặt giả

Trong phần này, em sẽ trình bày và mô tả các kỹ thuật phổ biến mà tin tặc có thể sử dụng để tạo ra khuôn mặt giả nhằm đánh lừa hệ thống xác thực khuôn mặt. Một số phương pháp và công nghệ thường được áp dụng bao gồm: Mô hình học sâu sử dụng mạng GANs, Deepfake, 3D Printing.

2.1.1. Mô hình Generative Adversarial Networks (GAN)[3]

Mô hình Generative Adversarial Networks (GAN): GAN là một phương pháp học sâu mà có thể được sử dụng để tạo ra hình ảnh khuôn mặt giả. Mô hình bao gồm hai thành phần chính - một mô hình sinh tạo ảnh và một mô hình phân biệt (discriminator) để phân biệt giữa ảnh thật và ảnh giả. Quá trình đào tạo GAN có thể dẫn đến việc tạo ra khuôn mặt giả có độ chi tiết và tự nhiên. Tóm lại, GAN là mạng để sinh dữ liệu mới giống với dữ liệu trong cơ sở dữ liệu có sẵn và có hai mạng trong GAN là Generator và Discriminator.

Trong GAN, mạng Generator sinh ra các dữ liệu giống như thật thì Discriminator cố gắng phân biệt đâu là dữ liệu được sinh ra từ Generator và đâu là dữ liệu thật có.



Hình 2.1. Minh họa các mạng trong GAN

Ví dụ bài toán là dùng GAN để Generate làm ra tiền giả có thể chi tiêu được. Dữ liệu có là tiền thật.

Generator giống như người làm tiền giả còn Discriminator giống như cảnh sát. Người làm tiền giả sẽ cố gắng làm ra tiền giả mà cảnh sát cũng không phân biệt được. Còn cảnh sát sẽ phân biệt đâu là tiền thật và đâu là tiền giả. Mục tiêu cuối cùng là người làm tiền giả sẽ làm ra tiền mà cảnh sát cũng không phân biệt được đâu là thật và đâu là giả và thế là mang tiền đi tiêu được.

Trong quá trình train GAN thì cảnh sát có 2 việc: 1 là học cách phân biệt tiền nào là thật, tiền nào là giả, 2 là nói cho thằng làm tiền giả biết là tiền nó làm ra vẫn chưa qua mắt được và cần cải thiện hơn. Dần dần thì thằng làm tiền giả sẽ làm tiền giống tiền thật hơn và cảnh sát cũng thành thạo việc phân biệt tiền giả và tiền thật. Và mong đợi là tiền giả từ GAN sẽ đánh lừa được cảnh sát.

Ý tưởng của GAN bắt nguồn từ zero-sum non-cooperative game, hiểu đơn giản như trò chơi đối kháng 2 người (cờ vua, cờ tướng), nếu một người thắng thì người còn lại sẽ thua. Ở mỗi lượt thì cả 2 đều muốn maximize cơ hội thắng của mình và minimize cơ hội thắng của đối phương. Discriminator và Generator trong mạng GAN giống như 2 đối thủ trong trò chơi. Trong lý thuyết trò chơi thì GAN model converge khi cả Generator và Discriminator đạt tới trạng thái Nash equilibrium, tức là 2 người chơi đạt trạng thái cân bằng và đi tiếp các bước không làm tăng cơ hội thắng. “A strategy profile is a Nash equilibrium if no player can do better by unilaterally changing his or her strategy”

Hãng công nghệ Mỹ sản xuất chip máy tính rất quan trọng với công nghệ AI, song cũng sử dụng một phần đội ngũ kỹ sư phần mềm để phát triển các công cụ hữu ích, hoặc thử nghiệm cách mới để sử dụng phần cứng của họ.

Một số ứng dụng của GAN liên quan đến tạo khuôn mặt giả:

Generate Photographs of Human Faces

Ví dụ về ảnh mặt người do GAN sinh ra từ 2014 đến 2017. Mọi người có thể thấy chất lượng ảnh sinh ra tốt lên đáng kể theo thời gian.



Hình 2.2. Ảnh mặt GAN sinh ra qua các năm

Đến năm 2017, các ảnh có được sinh ra bởi GAN có độ sắc nét cao, các chi tiết trên khuôn mặt nhìn đã thật hơn so với các phiên bản trước.

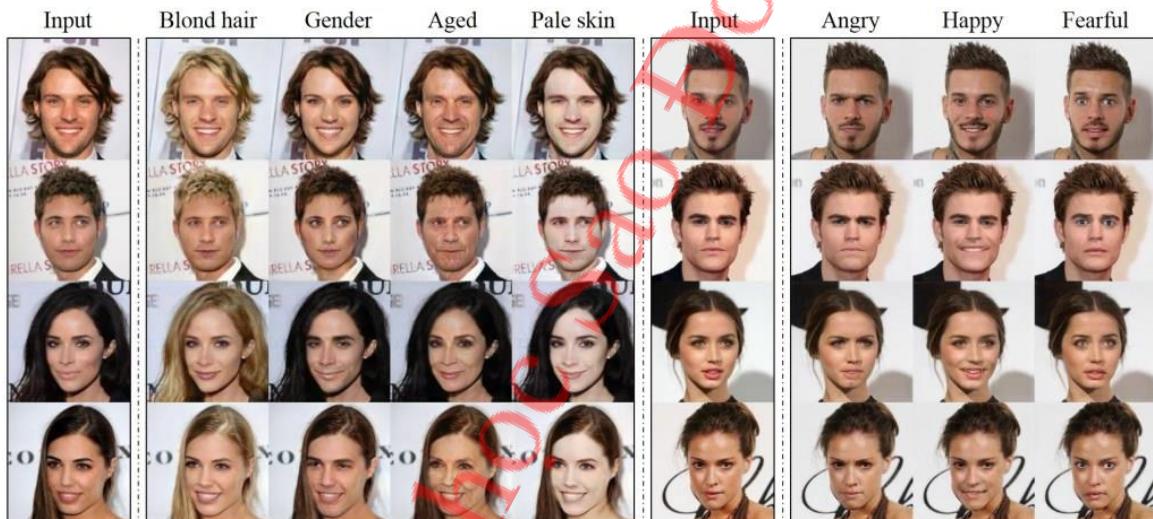
Đến năm 2018, ngoài chất lượng ảnh về độ sắc nét, ngay cả các đối tượng và nền của ảnh cũng được cải thiện rất nhiều.

Hình 2.3 là ảnh sinh ra bởi GAN năm 2018, phải để ý rất chi tiết mới có thể phân biệt được ảnh mặt đây là sinh ra hay ảnh thật.



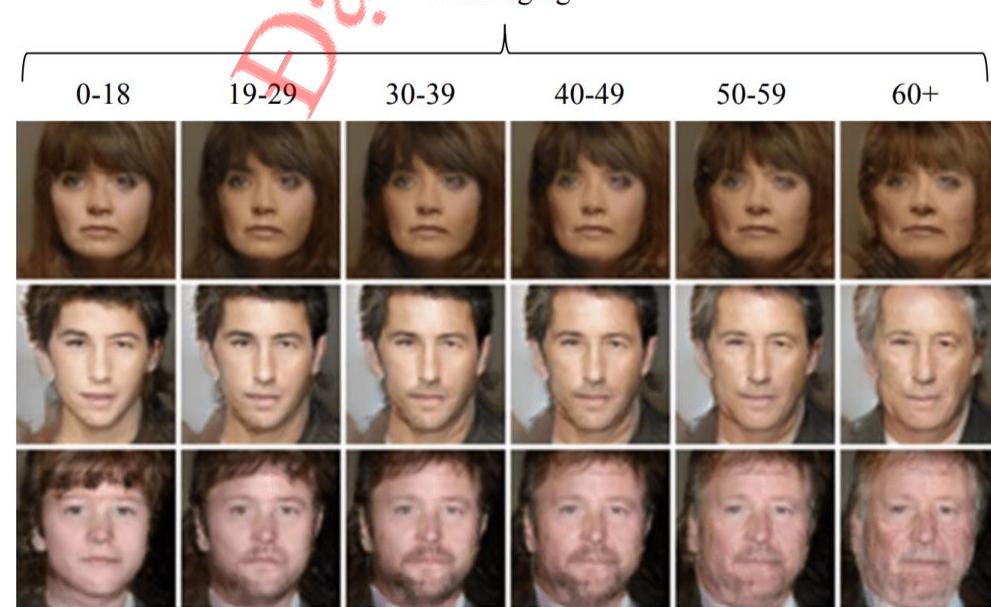
Hình 2.3. StyleGAN

Image editing: Có ứng dụng nổi tiếng là FaceApp. Ứng dụng này cho phép sửa các thuộc tính vào khuôn mặt như màu tóc, da, giới tính, cảm xúc hay độ tuổi.



Hình 2.4. Stargan

Face Aging



Hình 2.5. Age-cGAN

2.1.2. Deepfake Technology

Công nghệ Deepfake được xây dựng trên nền tảng machine learning (Học máy) mã nguồn mở của Google. Deepfake sẽ quét video và ảnh chân dung của một người sau đó hợp nhất với video riêng biệt nhờ AI và thay thế các chi tiết trên gương mặt như mắt, miệng, mũi với chuyển động gương mặt, giọng nói như thật. Càng có nhiều hình ảnh gốc thì AI càng có nhiều dữ liệu để thực hiện. Deepfake có thể gán khuôn mặt của người này sang người khác trong video với độ chân thực đến kinh ngạc. Trong deepfake AI, các thuật toán học sâu tự dạy cách giải quyết vấn đề với tập dữ liệu lớn, được sử dụng để hoán đổi khuôn mặt trong video, hình ảnh và nội dung kỹ thuật số khác để làm cho giả mạo có vẻ như thật.

Với việc công nghệ doppelganger giúp tạo ra một bản sao giống hệt chủ thể ngày càng phát triển với độ chân thực video cải tiến thì người dân khó có thể phân biệt được đâu là người thực đang nói trước ống kính và đâu là video AI. Hiện nay, Deepfake đang trở thành nỗi ám ảnh, là "bóng ma" trong thế giới ảo, được tội phạm mạng trên khắp thế giới dùng vào nhiều mục đích xấu không chỉ dừng lại ở mục đích lừa đảo.

Cách thức hoạt động của Deepfake liên quan chặt chẽ với trí tuệ nhân tạo. Từ đó, hình ảnh khuôn mặt của một số người nhất định (tạm gọi là người A) với chất lượng cao được thay thế hoàn toàn bằng khuôn mặt của một người khác (người B). Ảnh nén của A được đưa vào bộ giải mã của B. Bộ giải mã sau đó tái tạo lại khuôn mặt của người B với biểu cảm và hướng khuôn mặt của người A. Quá trình này được thực hiện liên tục, chi tiết đến khi cho ra sản phẩm “thật” nhất.

Deepfake Technology: Sử dụng mô hình học sâu để đào tạo và tạo ra video hoặc hình ảnh có chứa khuôn mặt của người khác. Công nghệ này có thể tạo ra những video giả mạo như người nổi tiếng đang nói những điều họ chưa từng nói.



Hình 2.6. Ví dụ Deepfake



Hình 2.7. Khuôn mặt giả Deepfake

Một số cách phòng tránh hiệu quả lừa đảo lợi dụng Deepfake:

Để không trở thành nạn nhân của các chiêu trò lừa đảo qua mạng xã hội nói chung và các chiêu trò sử dụng công nghệ Deepfake để lừa đảo nói riêng, người dân sử dụng mạng xã hội cần luôn tỉnh táo, cảnh giác. Khi nhận bất kỳ tin nhắn, cuộc gọi video với nội dung vay, mượn tiền qua các ứng dụng mạng xã hội. Tốt hơn hết, hãy bình tĩnh gọi điện thoại trực tiếp cho người thân để xác minh mà không qua các ứng dụng như Zalo, Messenger, Viber, Telegram... thậm chí gặp mặt trực tiếp khi có người cần hỏi vay tiền.

Bên cạnh đó, người dùng nên cân nhắc kỹ lưỡng trước khi click những đường link lạ và không rõ xuất xứ trên mạng xã hội để phòng tránh việc các đối tượng xấu đánh cắp tài khoản. Việc bảo vệ tài khoản mạng xã hội của bản thân và thiết lập các cơ chế bảo mật thông tin cũng là một trong những cách hiệu quả với các chiêu trò của tội phạm mạng.

Trên thị trường phổ thông, do năng lực tính toán của các ứng dụng Deepfake chưa hoàn hảo nên clip do AI tạo nên thường có dung lượng nhỏ, thời gian ngắn, chất lượng âm thanh và hình ảnh không cao. Trong những clip đó, khuôn mặt nhân vật thường khá "cứng" và ít cảm xúc hơn tự nhiên, hình thể cũng hạn chế di chuyển so với clip thông thường. Do đó, một trong những cách thông dụng để có thể kiểm tra phía bên gọi có sử dụng Deepfake AI để giả mạo trong những cuộc gọi video hay không là hãy yêu cầu họ quay mặt sang bên các góc 90 độ. Ngoài ra, còn có phương pháp khác là yêu cầu người gọi đưa tay trước mặt. Khi đưa tay lên khuôn mặt, sự chồng chéo của bàn tay và khuôn mặt khiến AI bị nhầm lẫn từ đó bộc lộ một số yếu tố có phần bất thường, “kì dị”.

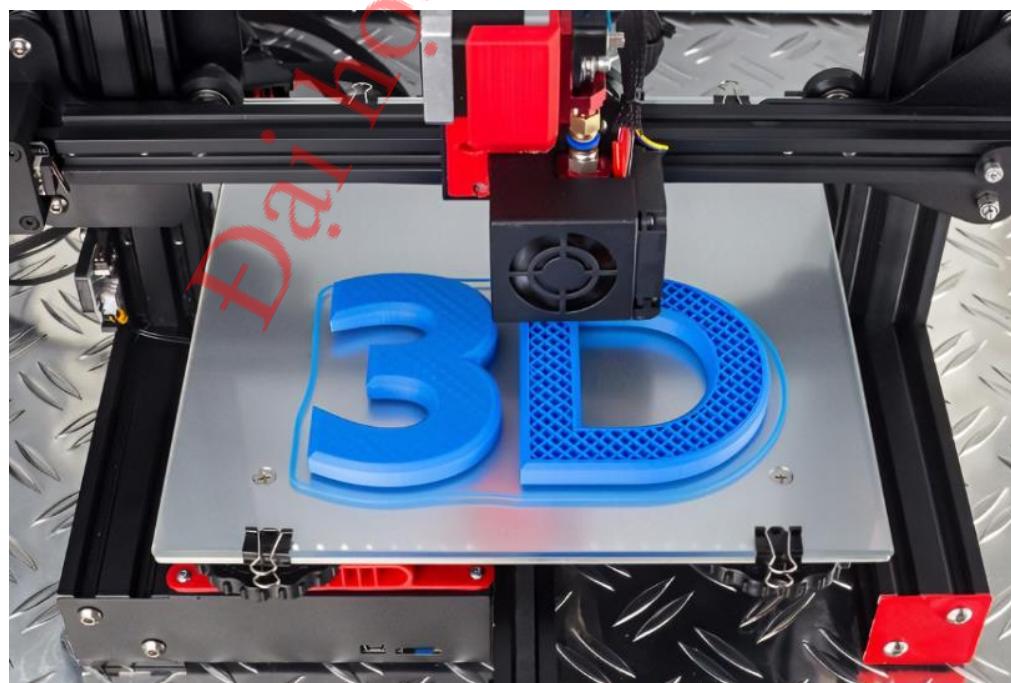


Hình 2.8. Dấu hiệu nhận biết DeepFake

2.1.3. 3D Printing và Silicone Masks

In 3D và Mặt nạ Silicone: Sử dụng công nghệ in 3D để tạo ra mô hình khuôn mặt và sau đó sử dụng silicone hoặc các vật liệu khác để tạo ra mặt nạ. Các kỹ thuật này có thể tạo ra khuôn mặt giả với độ chân thật cao.

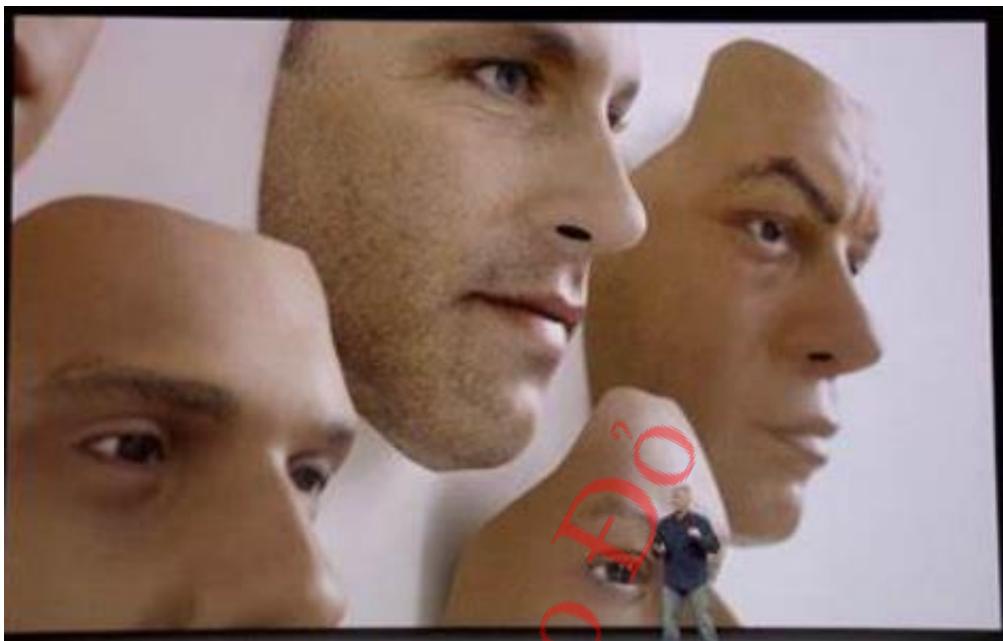
Công nghệ in 3D đã mở ra một cách tiếp cận mới trong việc tạo ra khuôn mặt giả. Đầu tiên, một mô hình 3D của khuôn mặt được tạo ra dựa trên các hình ảnh hoặc quét 3D. Mô hình này sau đó được in ra bằng máy in 3D để tạo ra một khuôn mặt giả cứng.



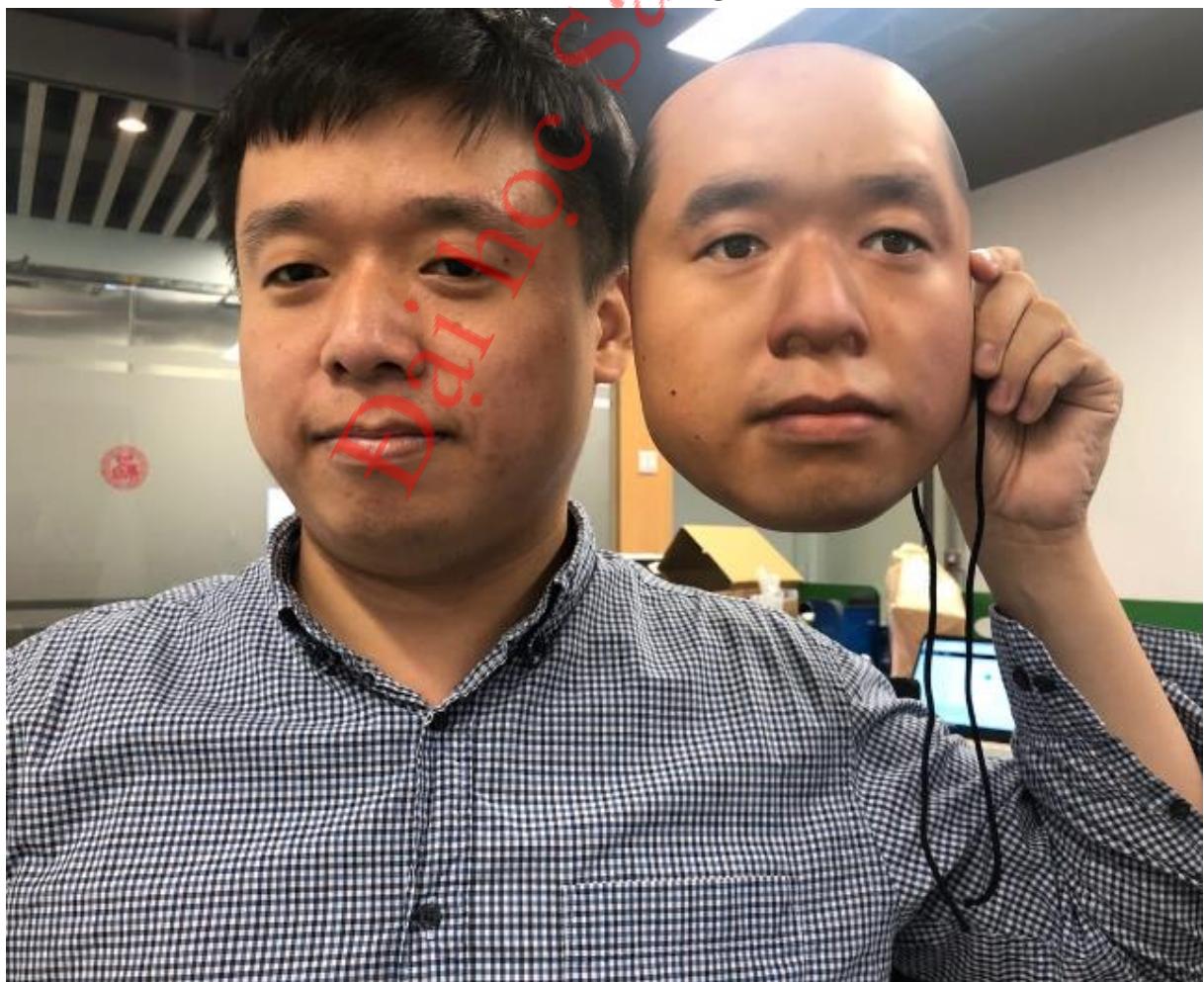
Hình 2.9. In 3D

Tiếp theo, silicone hoặc các vật liệu tương tự được sử dụng để tạo ra mặt nạ mềm mại và dẻo dai. Mặt nạ này có thể được đặt lên khuôn mặt thật của một người, tạo ra hiệu ứng như thế đó là khuôn mặt thật của họ.

Các kỹ thuật này có thể tạo ra khuôn mặt giả với độ chân thật cao, đến mức khó có thể phân biệt bằng mắt thường. Tuy nhiên, chúng cũng đặt ra thách thức lớn cho các hệ thống nhận dạng khuôn mặt, đòi hỏi sự cải tiến và phát triển liên tục để đối phó với những kỹ thuật ngày càng tinh vi hơn.



Hình 2.10. Khuôn mặt giả 3D



Hình 2.11. Công nghệ in 3D mặt giả

2.2. Kỹ thuật phát hiện khuôn mặt giả

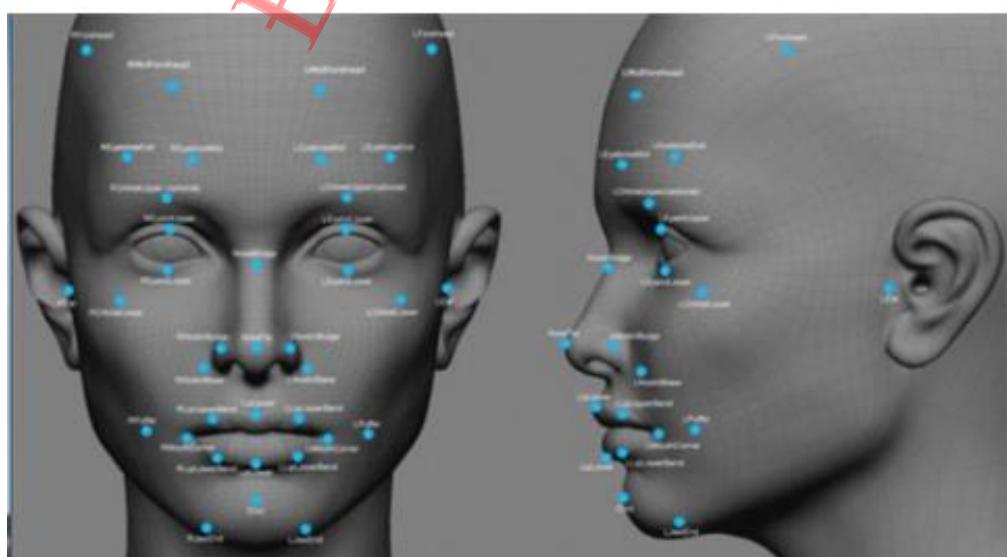
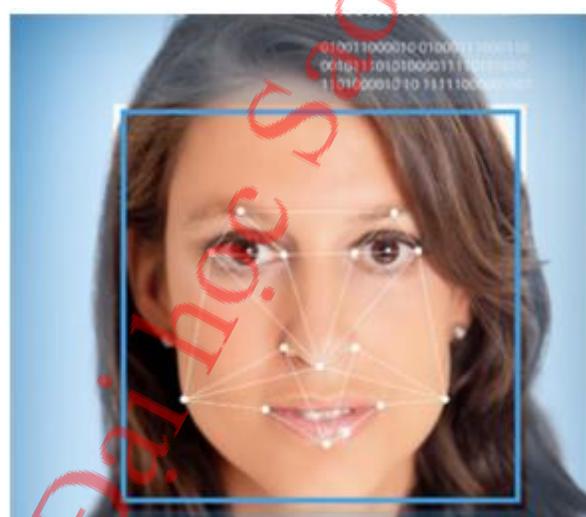
Trong phần này, em sẽ tìm hiểu về các kỹ thuật và phương pháp để phát hiện khuôn mặt giả, nhằm bảo vệ hệ thống xác thực khuôn mặt khỏi sự đánh lừa. Các phương pháp này có thể bao gồm:

2.2.1. Phân tích đặc điểm vùng khuôn mặt

Phân tích đặc điểm vùng khuôn mặt sử dụng các thuật toán để phân tích các đặc điểm cụ thể trên khuôn mặt như đường nét, cấu trúc da, và mô phỏng cấu trúc 3D để xác định tính chân thật của khuôn mặt.

Phương pháp này tập trung vào việc sử dụng các đặc điểm cụ thể của khuôn mặt để phát hiện khuôn mặt giả. Các điểm nổi bật như mắt, mũi, miệng, hoặc các đặc tính khác của khuôn mặt được sử dụng để xác định tính thật giả của một khuôn mặt. Mặc dù đây là một phương pháp truyền thống và có thể dễ bị đánh lừa bởi các kỹ thuật tạo mặt giả mới, nhưng nó vẫn đóng một vai trò quan trọng trong việc bảo vệ khỏi những phương pháp tạo mặt giả đơn giản.

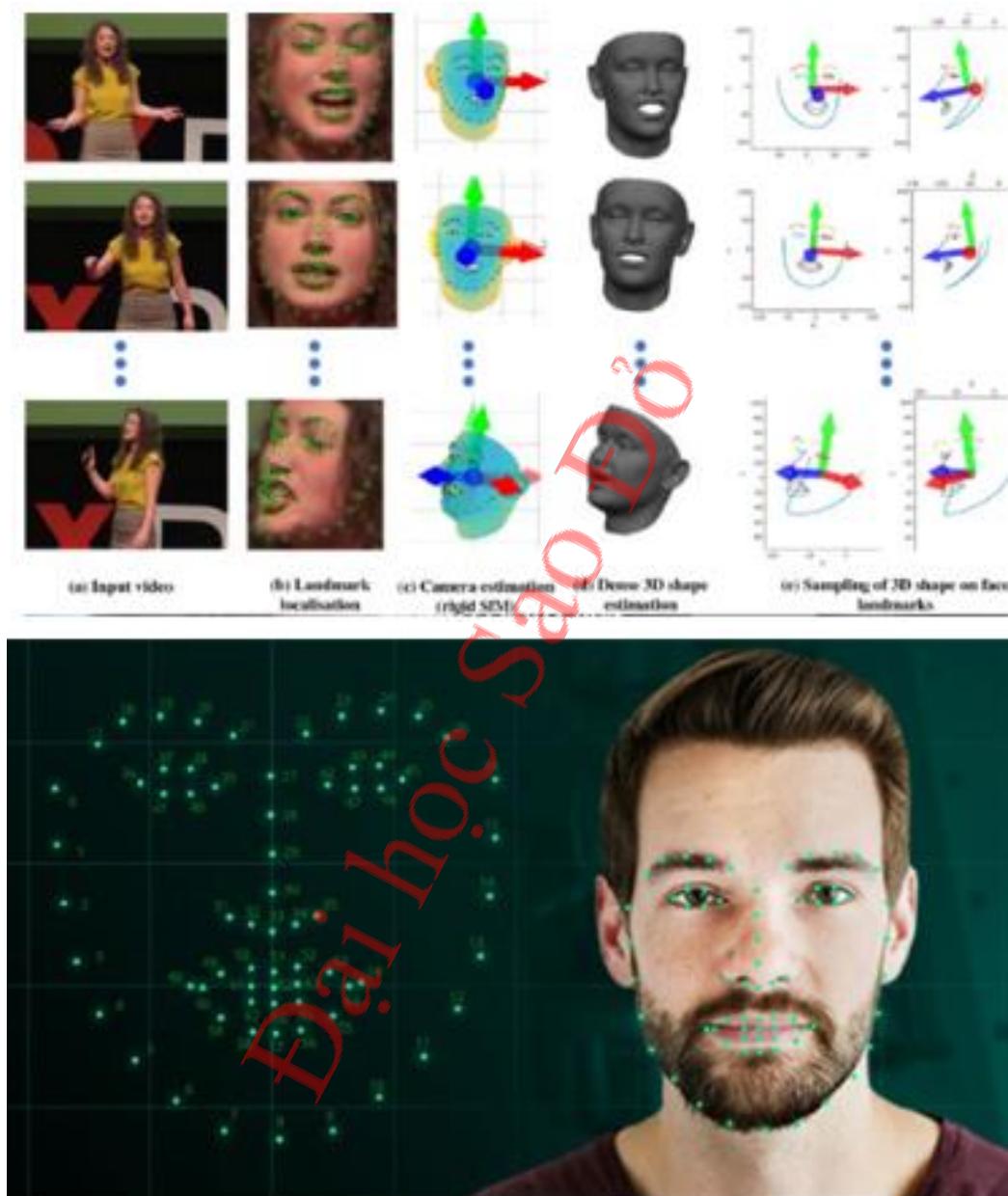
Tuy nhiên, phương pháp này có thể gặp khó khăn trong việc phát hiện các khuôn mặt giả chân thực và có thể bị đánh lừa bởi các khuôn mặt giả được tạo ra bằng công nghệ cao.



Hình 2.12. Sinh trắc học

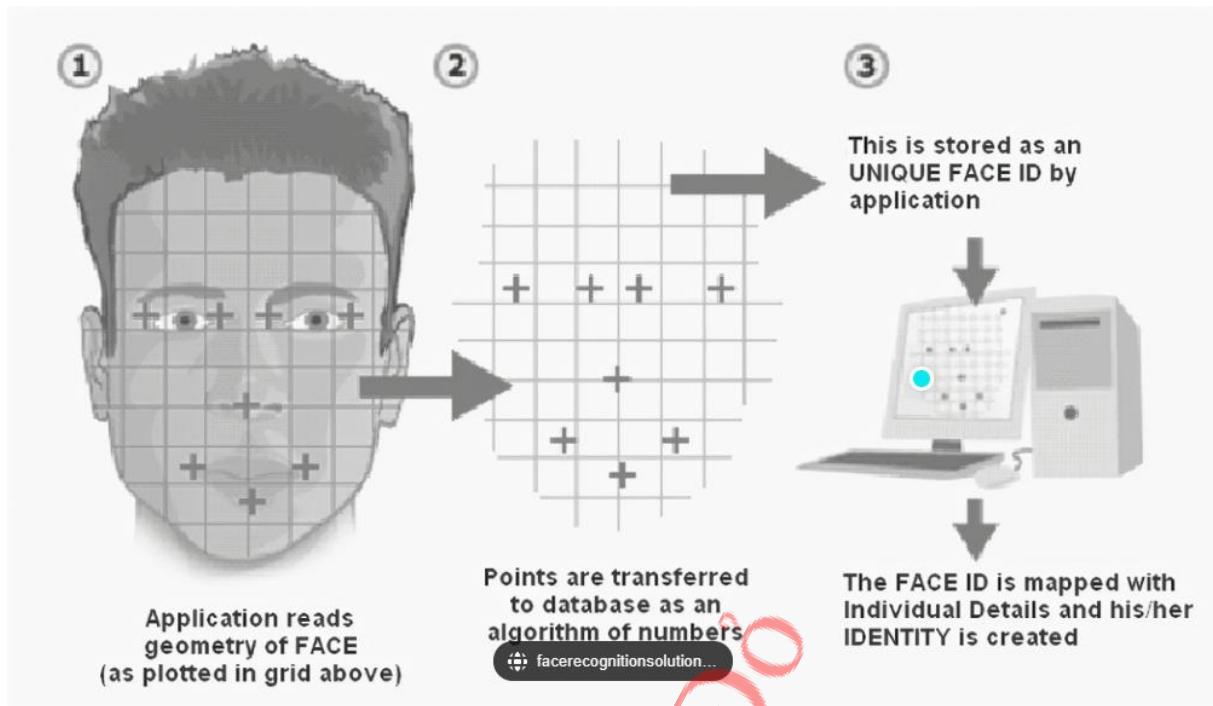
Công nghệ nhận dạng khuôn mặt hoạt động dựa trên công nghệ sinh trắc học, máy sử dụng các thuật toán phân tích, nhận diện, so sánh các điểm trên khuôn mặt với mẫu khuôn mặt đã đăng ký trước để xác định danh tính của một ai đó.

Toàn bộ quá trình nhận diện, xác thực, đối chiếu của công nghệ này đều được diễn ra tự động.



Hình 2.13. Phép phân tích thành phần chính

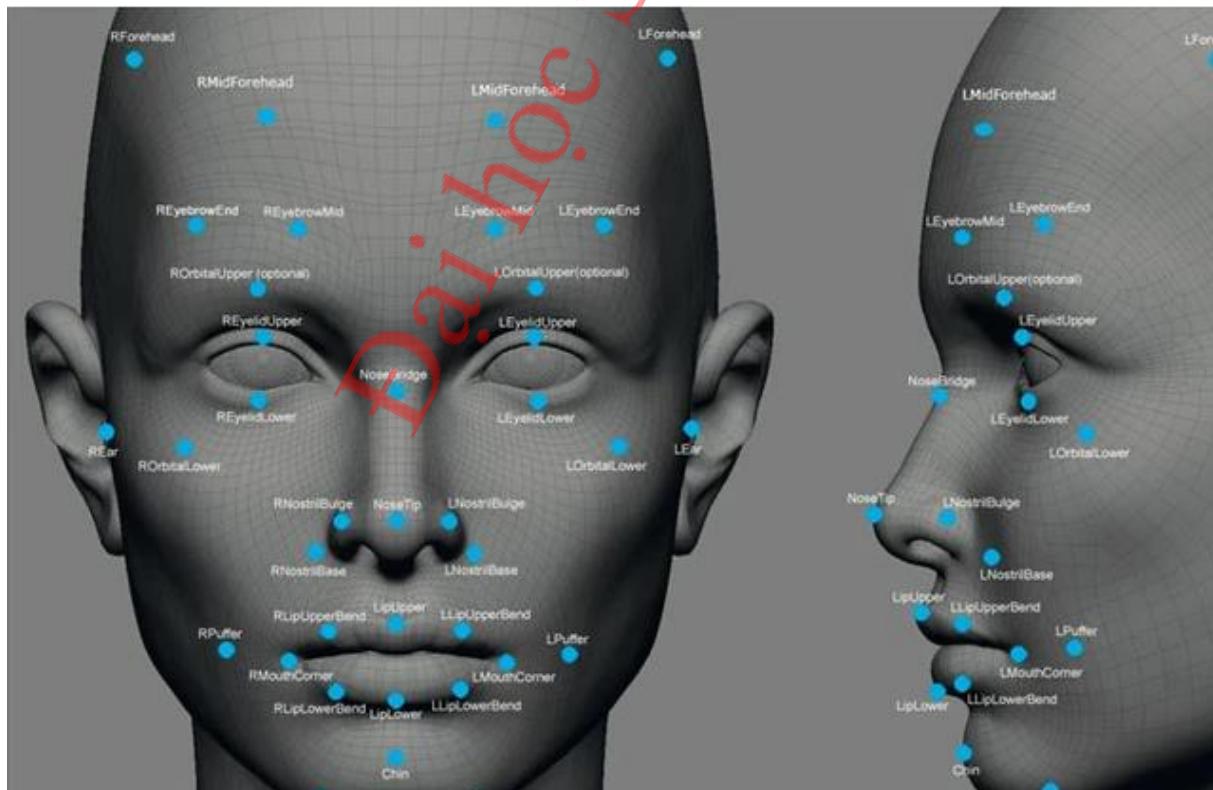
Hiện nay, các thuật toán phổ biến bao gồm Principal Component Analysis (phép phân tích thành phần chính) sử dụng các khuôn mặt riêng, Linear Discriminate Analysis (Phân tích biệt tuyến tính), Elastic Bunch Graph Matching sử dụng thuật toán Fisherface, các mô hình Markov ẩn, Multilinear Subspace Learning (Luyện nhớ không gian con đa tuyến) sử dụng đại diện cơ căng, và theo dõi liên kết động thần kinh. (theo Wikipedia)



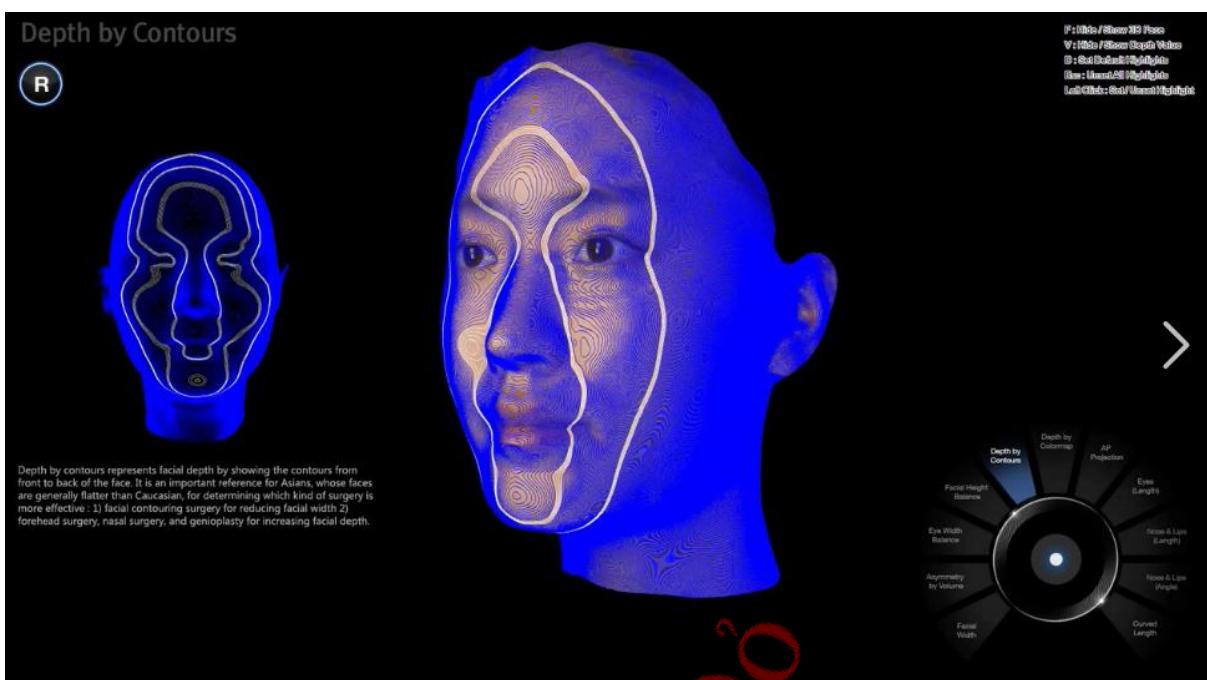
Hình 2.14. Ví dụ phân tích đặc điểm khuôn mặt

2.2.2. Công cụ quét 3D

Sử dụng Công nghệ Quét 3D: Sử dụng thiết bị quét 3D để xác định chiều sâu và cấu trúc 3D của khuôn mặt. Phương pháp này có thể giúp phân biệt giữa khuôn mặt thật và mô hình giả tạo.



Hình 2.15. Quét 3D đặc điểm khuôn mặt



Hình 2.16. Quét 3D khuôn mặt

2.2.3. Liveness Detection

Face Biometric (Sinh trắc học khuôn mặt) đang nhanh chóng được khách hàng và các tổ chức, doanh nghiệp chấp nhận như một phương thức xác minh danh tính tiện lợi. Tính năng công nghệ này hoạt động bằng cách so sánh các đặc điểm trên khuôn mặt của người dùng với một mẫu sinh trắc học đã đăng ký để xác minh danh tính. Tuy thay thế được cho các phương pháp bảo mật truyền thống như trả lời câu hỏi “bí mật”, nhập mã PIN, nhưng sinh trắc học khuôn mặt chưa đủ để kiểm tra đó có phải là khuôn mặt của thực thể sống không, hay chỉ là ảnh in chất lượng cao, video được quay lại.

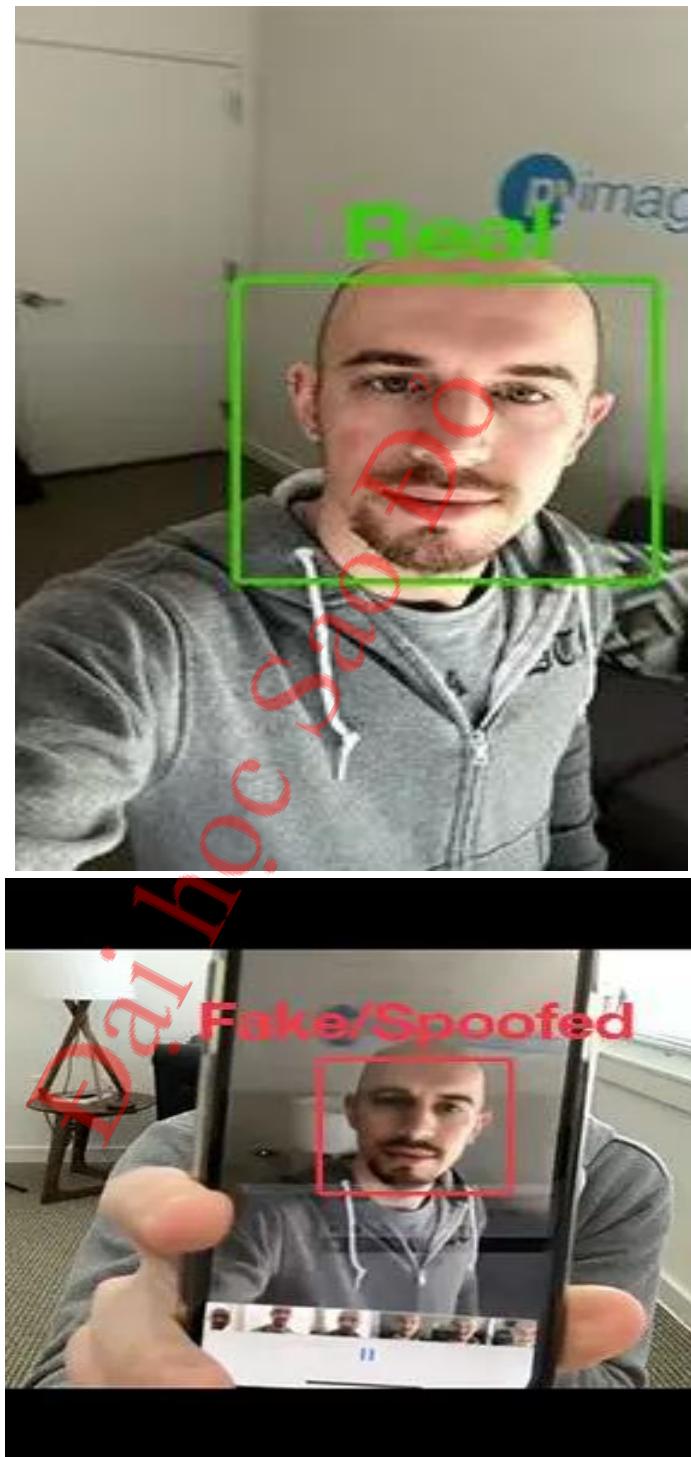
Để hỗ trợ, làm tăng hiệu quả của sinh trắc học khuôn mặt, Liveness Detection (Xác định thực thể sống) ra đời giúp đo lường, phân tích các đặc điểm và phản ứng vật lý nhằm xác định xem mẫu sinh trắc học có được chụp từ đối tượng sống có mặt tại điểm chụp hay không. Nói đơn giản hơn, đây là phương thức xác minh người thật, giúp ngăn chặn các hình thức giả mạo khuôn mặt bằng ảnh in, ảnh trên video, mặt nạ 3D... đảm bảo tính chính xác xuyên suốt trong quá trình định danh khách hàng điện tử (eKYC).

Liveness Detection: Sử dụng mô hình học sâu để phân biệt giữa khuôn mặt thật và giả bằng cách xem xét các đặc điểm động của khuôn mặt như chuyển động của mắt, nháy nháy, hoặc sự biến đổi của khuôn mặt theo thời gian.

Active Liveness Detection (Xác định thực thể sống chủ động) yêu cầu người dùng tham gia vào quá trình kiểm tra sự sống bằng cách thực hiện theo yêu cầu. Ví dụ một số hệ thống sẽ yêu cầu người dùng:

- Quay đầu sang phải/trái
- Gật đầu
- Nhìn theo một điểm sáng chuyển động trên màn hình
- Mỉm cười
- Đọc một câu hay một dãy số...

Passive Liveness Detection (Xác định thực thể sống thụ động) không yêu cầu hành động từ người dùng mà sẽ tự động xác định sự sống bằng các kỹ thuật, tự động phân biệt được ảnh, video là giả, mặt người là thật. Đối với camera có chất lượng cao, công nghệ này có thể phân định được qua chất lượng ảnh: ảnh blur (ảnh bị làm mờ), ảnh 3D TOF, Infrared, Lidar.



Hình 2.17. Phát hiện sự sống

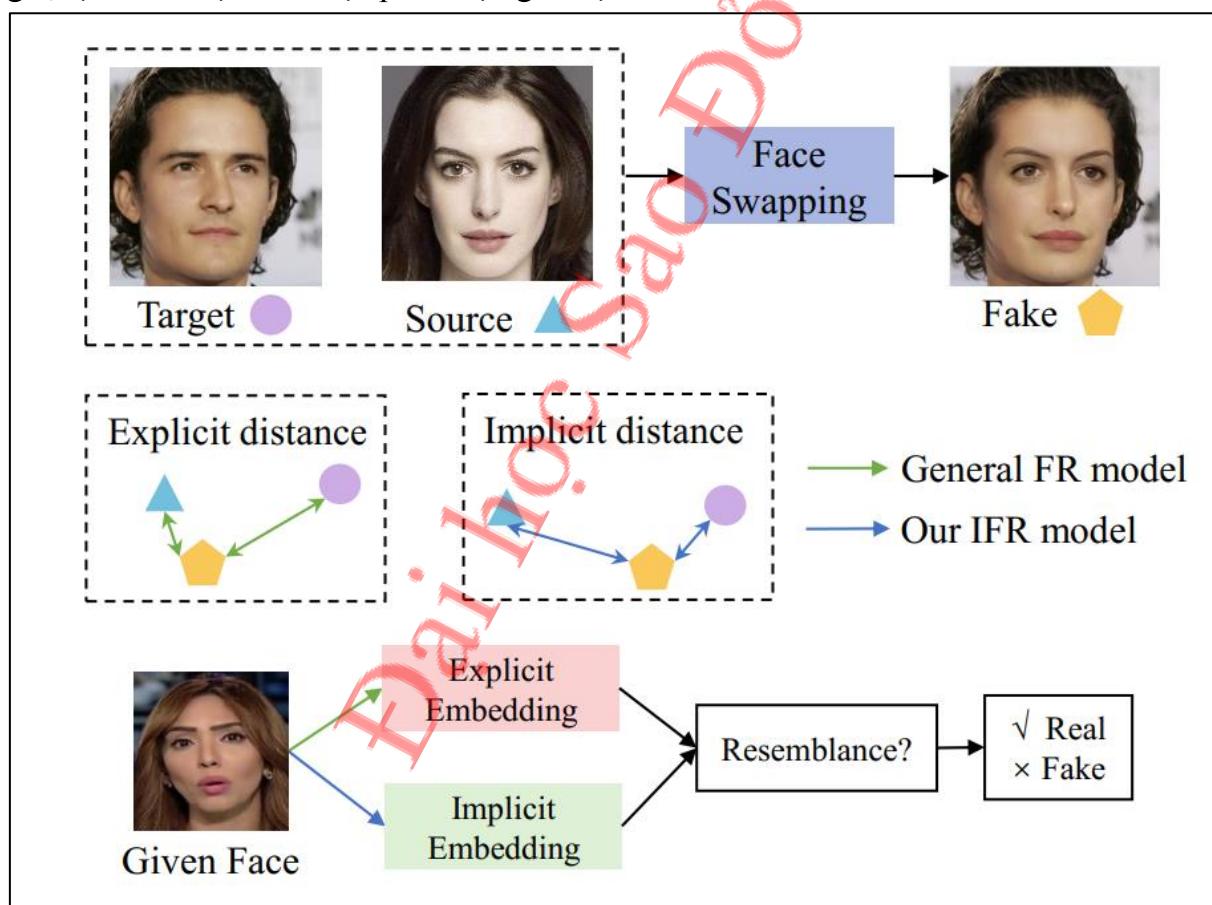
2.2.4. Phát hiện hoán đổi khuôn mặt Deepfake dựa trên nhận dạng ngầm[7]

Sự phát triển của deep learning đã thúc đẩy sự tiến bộ không ngừng của công nghệ giả mạo khuôn mặt. Đặc biệt đối với việc hoán đổi khuôn mặt, nó có thể thay thế khuôn mặt mục tiêu với khuôn mặt nguồn để tạo ra khuôn mặt giả mà mắt người không

thể phân biệt được. Với công nghệ này, kẻ tấn công có thể dễ dàng giả mạo các video chất lượng cao của những người nổi tiếng, và các nhân vật chính trị để đạt được các mục đích chính trị hoặc thương mại bất hợp pháp. Để giảm bớt việc lạm dụng hoán đổi khuôn mặt, đó là bởi khuôn mặt nguồn thông qua việc hoán đổi khuôn mặt để tạo ra khuôn mặt giả.

Về hình thức, khuôn mặt giả trông giống khuôn mặt nguồn thay vì khuôn mặt mục tiêu. Mô hình nhận dạng khuôn mặt chung (FR)

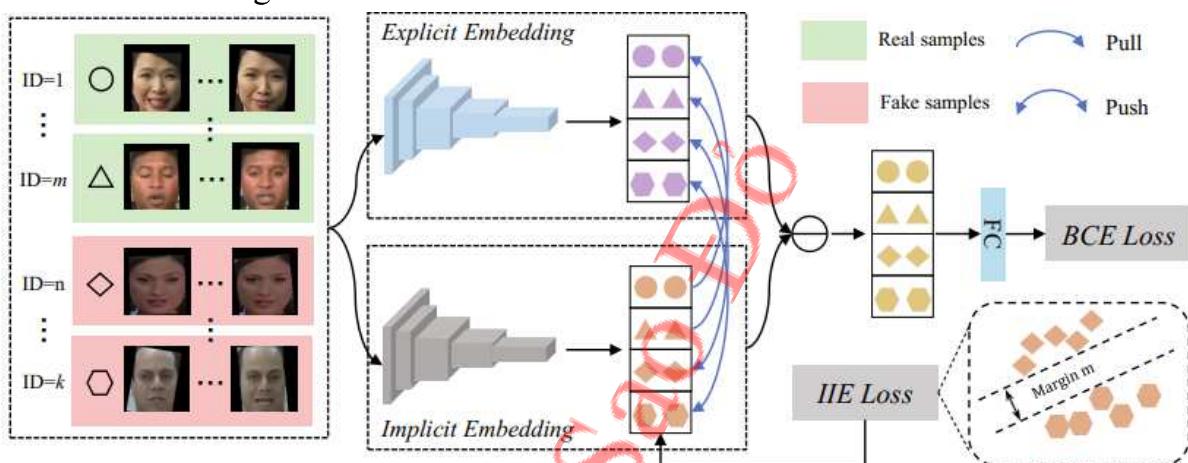
CosFace để thu được khoảng cách rõ ràng của các khuôn mặt này. Đặc biệt, do mặt giả được tổng hợp từ mặt nguồn và khuôn mặt mục tiêu, mong muốn khám phá nhận dạng khuôn mặt tiềm ẩn (IFR) mô hình có thể khai thác nhận dạng khuôn mặt mục tiêu tương ứng dựa trên khuôn mặt giả tạo. Với sự giống nhau giữa rõ ràng và tiềm ẩn nhung của khuôn mặt nhất định, ta có thể phân biệt nó một cách đáng kể là thật và giả, tạo điều kiện cho việc phát hiện giả mạo.



Hình 2.18. Nhận dạng khuôn mặt mục tiêu tương ứng dựa trên khuôn mặt giả tạo

Để giải quyết các vấn đề trên, xem xét việc phát hiện hoán đổi khuôn mặt từ góc độ nhận dạng khuôn mặt được hiển thị trong hình 2.18, việc hoán đổi khuôn mặt nhằm mục đích thay thế khuôn mặt target bằng khuôn mặt nguồn, tiếp tục tạo ra khuôn mặt giả điều đó thậm chí không thể phân biệt được bằng mắt người. Ở đây, em giới thiệu hai khái niệm mới về khuôn mặt giả, bao gồm nhận dạng rõ ràng và nhận dạng ngầm. Cụ thể, sự rõ ràng danh tính đại diện cho khuôn mặt giả trông như thế nào, nghĩa là nhận dạng khuôn mặt nguồn. Vì vậy, khoảng cách rõ ràng giữa khuôn mặt giả và khuôn mặt

thật có thể được đo lường bằng các mô hình nhận dạng khuôn mặt chung hiện có. Để nhận dạng ngầm, tôi cho rằng khuôn mặt giả có nguồn gốc từ mặt và mặt mục tiêu. Mặc dù nó trông giống như nguồn khuôn mặt, nó có thể chứa ít nhiều danh tính khuôn mặt mục tiêu trong đội hình. Em gọi thông tin khuôn mặt mục tiêu tiềm năng này là danh tính ngầm của khuôn mặt giả. Điều đáng lưu ý là danh tính ngầm của khuôn mặt thật phù hợp với danh tính rõ ràng của nó. Do đó, với một hình ảnh khuôn mặt, chúng ta nhúng tương ứng, nó sẽ được đưa vào các không gian đặc trưng nhận dạng rõ ràng và tiềm ẩn. Khoảng cách giữa rõ ràng và tiềm ẩn của nó Các đặc điểm được lấy làm cơ sở để đánh giá thật và giả. Pro video khoảng cách rất gần, hình ảnh đưa ra là thật, nếu không thì đó là hình ảnh giả.



Hình 2.19. Phác thảo về khuôn khổ định hướng nhận dạng tiềm ẩn được đề xuất

Các phương pháp hoán đổi khuôn mặt gần đây được hưởng lợi từ những tiến bộ trong học sâu. Ngay từ đầu, những người tìm kiếm xem việc hoán đổi khuôn mặt như một vấn đề chuyển đổi phong cách. Dưới sự hướng dẫn của các điểm mốc bì mặt, CNN có thể chuyển một hình ảnh khuôn mặt sang kiểu của hình ảnh khuôn mặt khác bằng một danh tính cụ thể. Kể từ đó, DeepFakes cỏ diễn đề xuất công việc hoán đổi khuôn mặt của bộ mã hóa-giải mã. Sau khi được huấn luyện, nó có thể hoán đổi khuôn mặt giữa hai người danh tính cụ thể nhưng không thể khai quát hóa cho người khác. Trên này cơ bản, một số phương pháp kết hợp các biểu hiện tiềm ẩn đã xuất hiện. Họ trích xuất các đặc điểm nhận dạng từ mặt nguồn và các đặc điểm thuộc tính từ mặt đích.

Tuy nhiên, biểu thức của khuôn mặt mục tiêu thường không được cung cấp trước ở đầu ra của bộ giải mã. Vấn đề khó khăn nhất với các phương pháp trên đòi hỏi phải đào tạo theo cặp các khuôn mặt được hoán đổi, điều này không thân thiện trong thực tế. ĐỂN khắc phục hạn chế trên, Nirkin et al. [7] đề xuất một cách tiếp cận mới dựa trên mạng lưới thần kinh tái diễn cho khuôn mặt tái hiện, có thể được áp dụng cho một hình ảnh hoặc một trình tự video. Các phương pháp ping hoán đổi khuôn mặt dựa trên tái thiết gần đây với GAN cũng đã cho thấy thành công. Họ là những người theo thuyết bất khả tri và có thể tạo ra những hình ảnh giả thực tế và chất lượng cao. Nhìn chung, các phương pháp hoán đổi khuôn mặt dựa trên học tập hiện có tuyên bố sẽ tách riêng danh tính của mặt ban đầu và gán nó cho mặt đích. Tuy nhiên, không có phương pháp tách

rồi thuần túy, do đó khuôn mặt giả chứa thông tin nhận dạng khuôn mặt mục tiêu tiềm năng. Để đạt được mục đích này, phương pháp của nhằm mục đích khám phá dấu hiệu tiềm năng này để phát hiện ping hoán đổi khuôn mặt.

2.2.5. Phát hiện hình ảnh Deepfake bằng kỹ thuật Deep Learning [8]

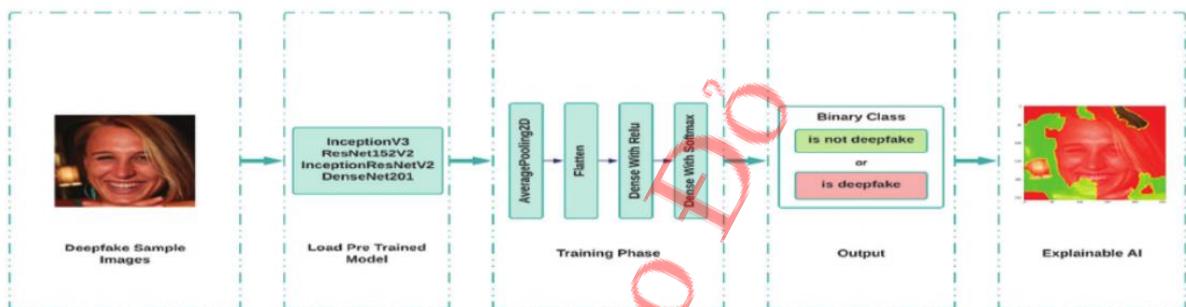
Một người dùng Reddit lần đầu tiên tạo các video clip bị chỉnh sửa có tên là “deepfake” bằng cách sử dụng TensorFlow. Sự nổi lên của deepfakes đã đặt câu hỏi về tính xác thực của bất kỳ nội dung xã hội kỹ thuật số nào. Ứng dụng được sử dụng rộng rãi nhất của Deepfake là FaceApp, ứng dụng này có chức năng hoán đổi khuôn mặt. Để xác thực, các nền tảng truyền thông xã hội như Facebook và Twitter sẽ phát hiện và xóa nội dung giả mạo.



Hình 2.20. Deepfake

Vì mối đe dọa của deepfake đã được xác định nên các phương pháp xác định deepfake là cần thiết. Các phương pháp tiếp cận ban đầu dựa vào các tính năng được tạo ra bắt nguồn từ các trực trắc và sai sót trong video giả mạo. Trong khi các phương pháp gần đây sử dụng Deep Learning (DL) để tự động phát hiện deepfake. Những mô hình DL này có thể được sử dụng để đạt được độ chính xác cực cao, thậm chí đôi khi còn vượt trội hơn con người. Kỹ thuật này mang lại lợi ích rất lớn cho việc nhận dạng giọng nói, nhận dạng đối tượng trực quan, phát hiện đối tượng và các lĩnh vực khác nhau [12]. Các thuật toán DL cung cấp tính linh hoạt và độ chính xác chưa từng có trong nhiều ứng dụng miền nhận dạng và bảo mật. Ngay cả khi chỉ được đào tạo về thông tin thống kê, các thuật toán như vậy đã cho thấy kết quả vượt trội trên các tập dữ liệu điểm chuẩn.

Nhưng bất chấp hiệu suất đáng kinh ngạc của các thuật toán Machine Learning (ML) này, chúng cũng có thể mắc lỗi. Do đó, câu hỏi về sự tin cậy và an toàn trong ML rất quan trọng. Do tầm quan trọng ngày càng tăng của thuật toán ML, Liên minh Châu Âu đã ban hành Quy định chung về bảo vệ dữ liệu (GDPR), trong đó bao gồm quyền giải thích. Vì vậy, các phương pháp ML phải có thể giải thích được bằng cách sử dụng các mô hình vốn có thể giải thích được hoặc bằng cách thiết lập các phương pháp tiếp cận mới. Vì chỉ phát hiện các hình ảnh deepfake với độ chính xác cao hơn có thể là chưa đủ, nên thông tin hoặc giải thích thích hợp cũng cần thiết để hiểu các đặc điểm chính xác để đưa ra dự đoán của chúng. Nhờ sức mạnh của AI có thể giải thích (XAI), hệ thống ML giờ đây sẽ có thể giải thích lý do của chúng, xác định điểm mạnh và điểm yếu cũng như dự đoán hành vi trong tương lai của chúng.

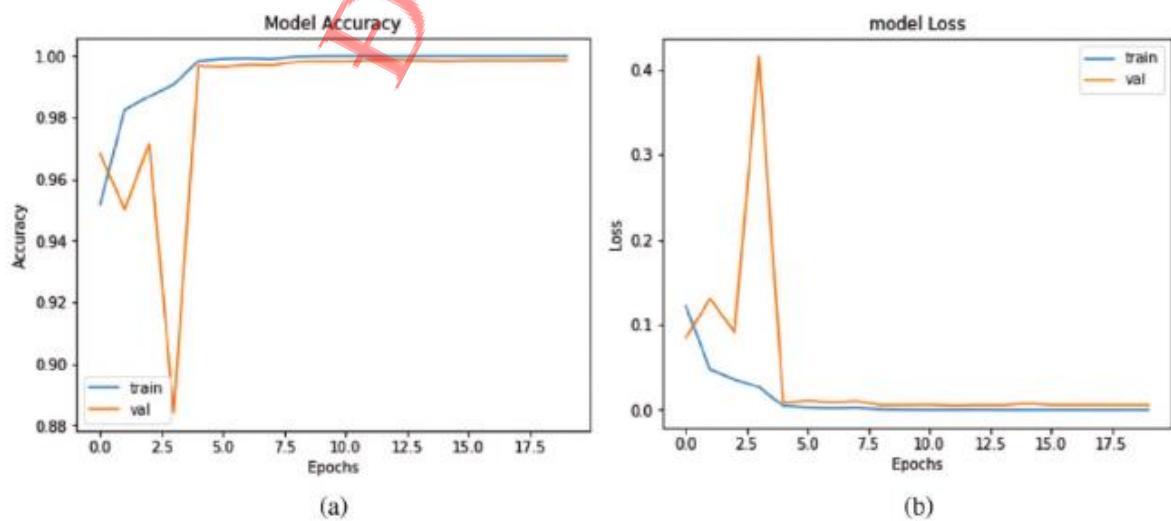


Hình 2.21. Kiến trúc nhận dạng Deepfake

Trọng số của mô hình được giữ nguyên, đó là một “ImageNet” để phát hiện sự so sánh nhất quán. Lớp trên cùng của mô hình bị bỏ qua và lớp dày đặc được thay đổi tương ứng đối với hai nơ-ron, có phân loại giả và thực. Vì TensorFlow cung cấp Học sâu khác nhau (DL), mỗi mô hình đều có mật độ và số lớp riêng. Với mục đích so sánh, tất cả các siêu tham số đều được giữ nguyên cho mục đích so sánh để cho phép so sánh chính xác giữa các phương pháp DL khác nhau. Không có mô hình nào khác có lớp trên cùng được đào tạo trước. Hình dạng đầu vào của tất cả các mô hình được giữ nguyên giống hệt như 224 px * 224 px với kênh 3 cho định dạng RGB. Lớp chập, lớp gộp, lớp hiệu chỉnh Relu và lớp được kết nối đầy đủ là các lớp khác nhau của mô hình CNN. Đối với mô hình đề xuất, lớp tống hợp trung bình toàn cầu và lớp kích hoạt của Relu đã được sử dụng với mật độ 512. Ngoài ra, để điều chỉnh siêu tham số, chuẩn hóa hàng loạt đã được sử dụng. Quá khớp mô hình là một vấn đề phổ biến đối với các tập dữ liệu lớn. Để ngăn vấn đề này xảy ra, phương pháp được đề xuất thực hiện giảm 0,2 trước khi huấn luyện mỗi lô nơ-ron. Các mô hình đã được đào tạo với trọng số trước đó là “ImageNet”. Mục đích của khung đề xuất là phát hiện các khuôn mặt giả sâu và sau đó xác thực mô hình bằng XAI. Sau khi thu thập hình ảnh, tập dữ liệu phải được chia thành tập huấn luyện, tập kiểm tra và tập xác thực, đồng thời việc phân tách được thực hiện để tập huấn luyện, tập kiểm tra và tập xác thực vẫn được cân bằng. Để có kết quả đào tạo tốt hơn, dữ liệu đã được xử lý trước. Để đảm bảo hình ảnh được xử lý đúng cách, nó phải được kiểm tra bằng cách vẽ đồ thị bằng Matplotlib

Các mô hình AI:

DL có thể giải thích được coi là mô hình hộp đen vì việc đạt được sự hiểu biết thấu đáo về hoạt động bên trong của mạng lưới thần kinh sâu là điều không thể khuất phục. Hầu hết thời gian, Trí tuệ nhân tạo (AI) chấp nhận kết quả mà không cần giải thích nhiều. Điều này đòi hỏi một hệ thống để giải thích các mô hình hộp đen này; do đó, AI có thể giải thích (XAI) đã xuất hiện. XAI đưa ra sự làm rõ thông qua trực quan hóa, phân tích, che giấu, giá trị số và trọng số của các tính năng. Thuật toán có thể giải thích cục bộ (LIME) được coi là phù hợp nhất cho mô hình hộp đen vì LIME là mô hình bất khả tri, có nghĩa là nó có thể được sử dụng cho các mô hình DeepLearning (DL) khác nhau. Vì phương pháp được đề xuất đã trải qua một số so sánh giữa các phương pháp DL khác nhau, điều quan trọng là chọn thuật toán XAI không xác định mô hình. Thuật toán LIME sử dụng lựa chọn mô-đun phụ để xuất kết quả của mô hình. Thuật toán có hai biến và đầu tiên, sử dụng giải thích tuyến tính asparse, thuật toán giải thích trọng số của các tính năng chịu trách nhiệm. Thứ hai, tầm quan trọng của các tính năng này được tính toán trực tiếp và sau đó được tối ưu hóa bằng thuật toán tham lam. Hàm Theargmax trả về trọng lượng tính năng cuối cùng. Việc đọc hình ảnh mẫu được thực hiện thông qua mô-đun Python có tên là “pillow”, trong đó hình ảnh mẫu được chuyển đổi và một chiều bổ sung được thêm vào mảng của hình ảnh mẫu. Sau đó, mô hình phải dự đoán ảnh mẫu bằng hàm Argmax. Để phân đoạn ảnh mẫu, thuật toán phân đoạn ảnh scikit này được sử dụng với tính năng ranh giới đánh dấu. Thuật toán phân đoạn có tỷ lệ tham số là 0,2 và khoảng cách tối đa là 200. Sau khi phân đoạn, hình ảnh mẫu được vẽ bằng thanh màu. Sau đó, hình ảnh mẫu đã được xác minh bằng cách sử dụng mặt nạ 3D. Để triển khai mô hình LIME và giải thích hộp đen của học sâu, cần phải tạo Người giải thích Hình ảnh Lime. Ví dụ: khi giải thích một trường hợp, người giải thích lấy mảng hình ảnh mẫu asan, kết quả dự đoán của hình ảnh mẫu, số lượng nhãn của nó và số lượng mẫu. Trong phương pháp được đề xuất, sau khi tạo phần giải thích, phân tích trực quan của phần giải thích phải được hiển thị bằng cách sử dụng một ranh giới. Mặt nạ hình ảnh đã được xác minh bằng cách sử dụng cả lượng chỉ dương và lượng bình thường.



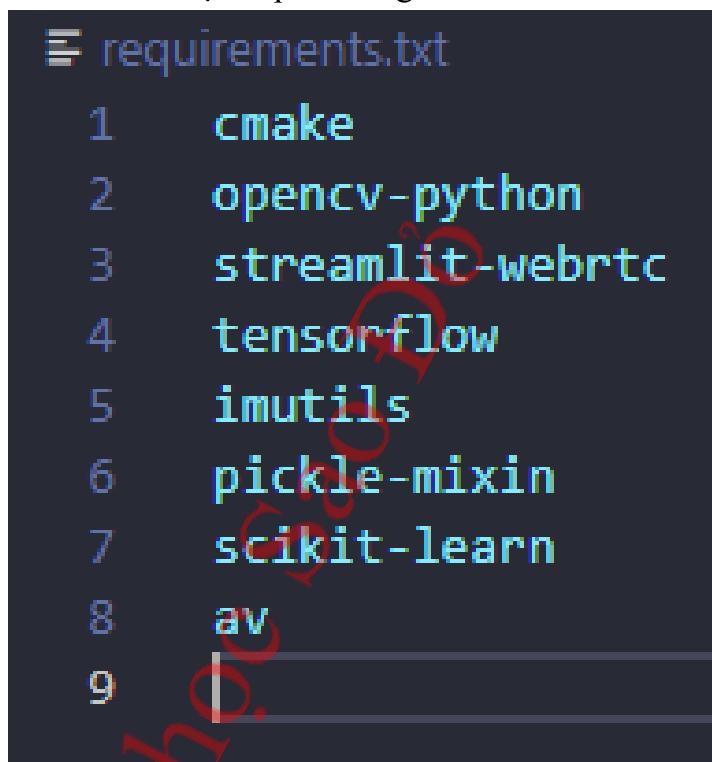
Hình 2.22. InceptionResNetV2. (a) độ chính xác và (b) biểu đồ tổn hao

Chương 3. XÂY DỰNG ỨNG DỤNG

3.1. Lựa chọn công cụ

Để xây dựng ứng dụng phát hiện khuôn mặt giả, em sử dụng các công cụ sau:

- PyCharm Community Edition 2023.2.1
- Python 3.8
- Module venv được tích hợp sẵn trong Python hỗ trợ tạo môi trường ảo Python.
- Công cụ Roboflow hỗ trợ gán nhãn và định dạng dữ liệu.
- Thư viện Tensorflow hỗ trợ deep learning.



```

requirements.txt

1 cmake
2 opencv-python
3 streamlit-webrtc
4 tensorflow
5 imutils
6 pickle-mixin
7 scikit-learn
8 av
9

```

Hình 3.1. Thư viện sử dụng trong dự án

3.2. Mô hình mạng CNN phát hiện khuôn mặt giả mạo

Để xây dựng ứng dụng phát hiện khuôn mặt giả mạo, em cần thực hiện các bước như sau:

Bước 1: Chuẩn bị cơ sở dữ liệu.

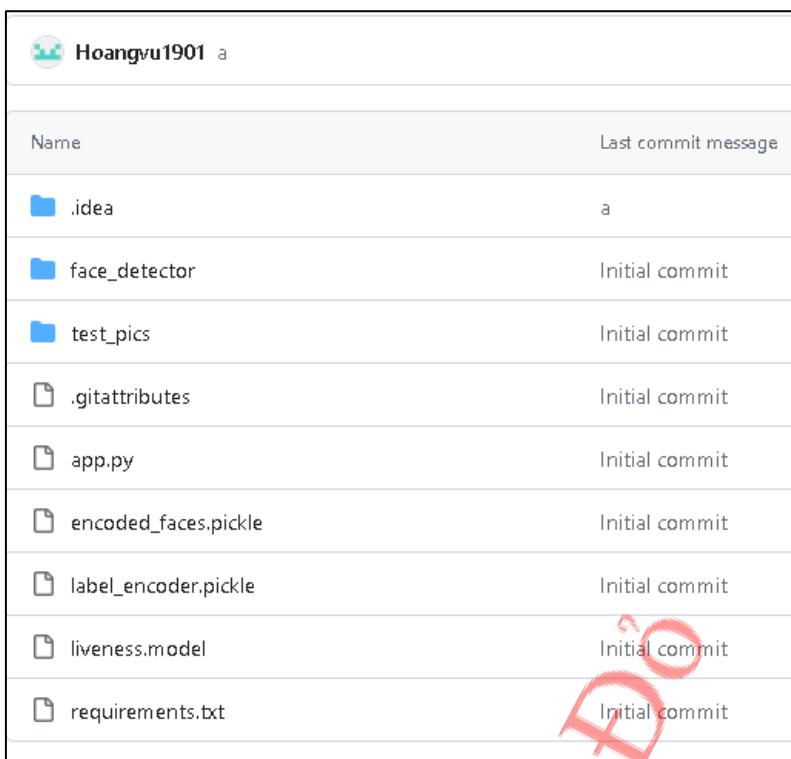
Bước 2: Triển khai mạng CNN có khả năng phát hiện khuôn mặt giả mạo.

Bước 3: Huấn luyện mạng CNN.

Bước 4: Sử dụng mô hình đã huấn luyện vào dự đoán.

Trong các phần tiếp theo, em sẽ trình bày chi tiết các bước thực hiện của ứng dụng. Để bài toán trở nên đơn giản, trong phạm vi của đồ án này, em sẽ minh họa việc phân biệt khuôn mặt thật và khuôn mặt giả mạo trên màn hình. Thuật toán có thể dễ dàng được mở rộng sang các loại khuôn mặt giả mạo khác như ảnh in, ảnh in có độ phân giải cao,...

Cấu trúc của dự án gồm:



The screenshot shows a GitHub repository named "Hoangvu1901". The repository contains the following files and their last commit messages:

Name	Last commit message
.idea	a
face_detector	Initial commit
test_pics	Initial commit
.gitattributes	Initial commit
app.py	Initial commit
encoded_faces.pickle	Initial commit
label_encoder.pickle	Initial commit
liveness.model	Initial commit
requirements.txt	Initial commit

Hình 3.2. Cấu trúc của dự án

3.3. Cơ sở dữ liệu

Cơ sở dữ liệu của ứng dụng được em thu thập trực tiếp bằng điện thoại cá nhân. Dữ liệu gồm 2 video:

- Video 1: Độ dài 25 giây đi xung quanh nhà.
- Video 2: Sử dụng điện thoại quay lại video 1 đang phát trên màn hình máy tính.

Bộ dữ liệu gồm hai tập dữ liệu (Positive - ảnh mặt thật và Negative - ảnh mặt giả) trong đó:

- Ảnh thật: chụp từ video selfie bằng điện thoại (video 1). Về lý tưởng, tập dữ liệu này nên có khuôn mặt của nhiều người và của nhiều dân tộc với nhiều đặc trưng về màu da, khuôn mặt khác nhau.
- Ảnh giả mạo: Chụp từ máy ảnh quay từ màn hình máy tính (video 2). Cũng đề xuất có sự đa dạng như mong muốn ở tập dữ liệu ảnh thật.

Trong các ảnh thật và ảnh giả mạo, sử dụng máy dò khuôn mặt Caffe để xác định vùng quan tâm ROI (Region of Interest) của khuôn mặt. Mô hình này đã được huấn luyện trước và đặt trong thư mục face_detector của dự án.

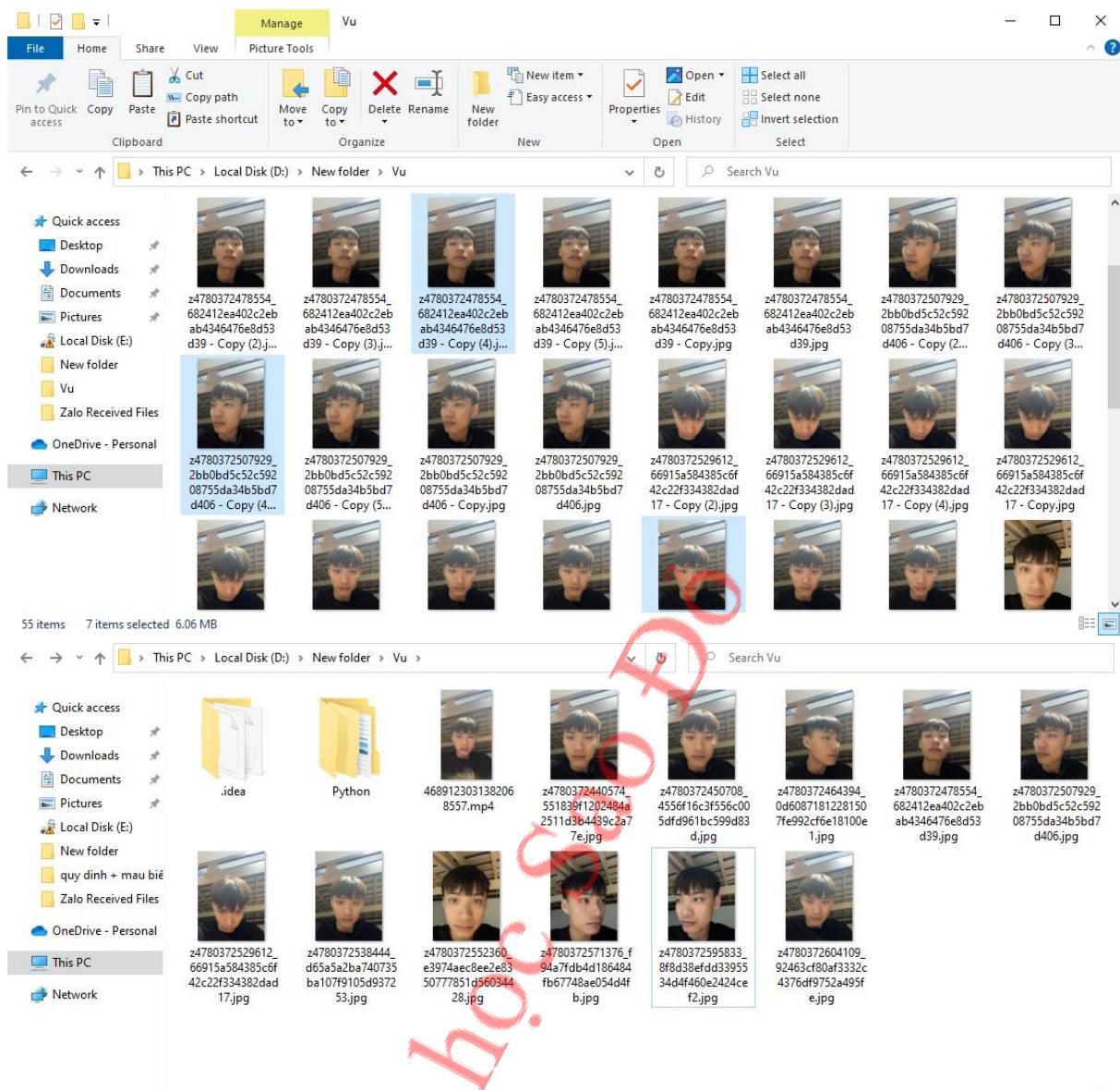
Triển khai LivenessNet:

Trong thư mục "pyimagesearch/", em triển khai lớp LivenessNet để phân loại giữa khuôn mặt "thật" và "giả mạo".

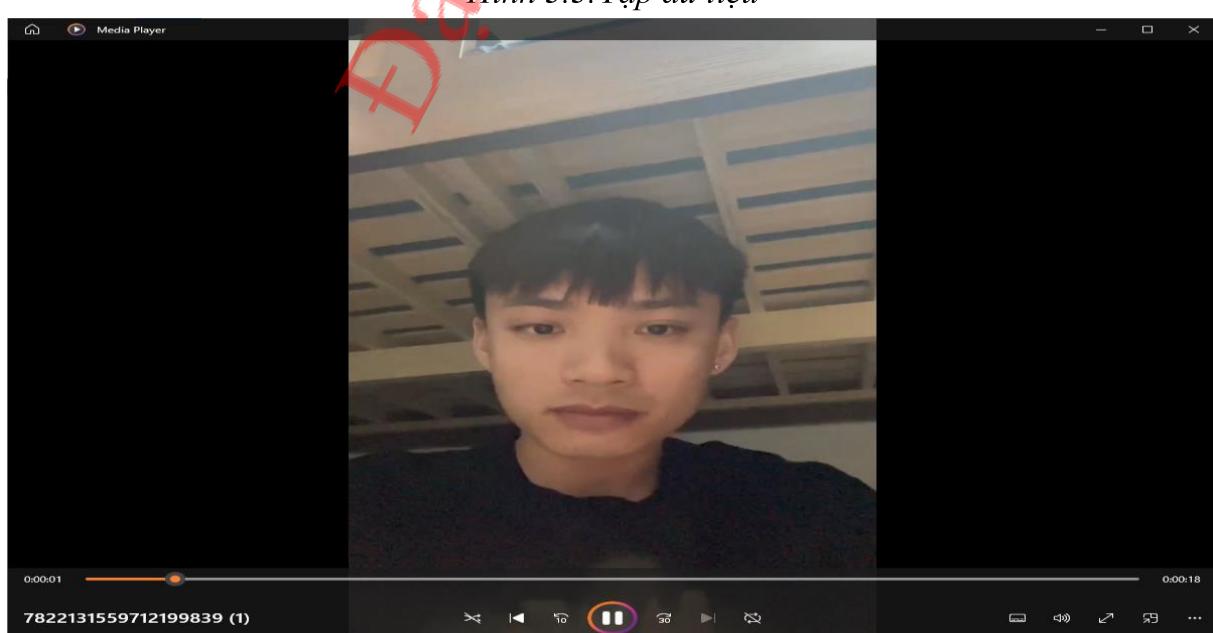
Chuẩn bị đầu vào cho LivenessNet:

Cung cấp hai video đầu vào trong thư mục "Video/" để đào tạo trình phân loại LivenessNet.

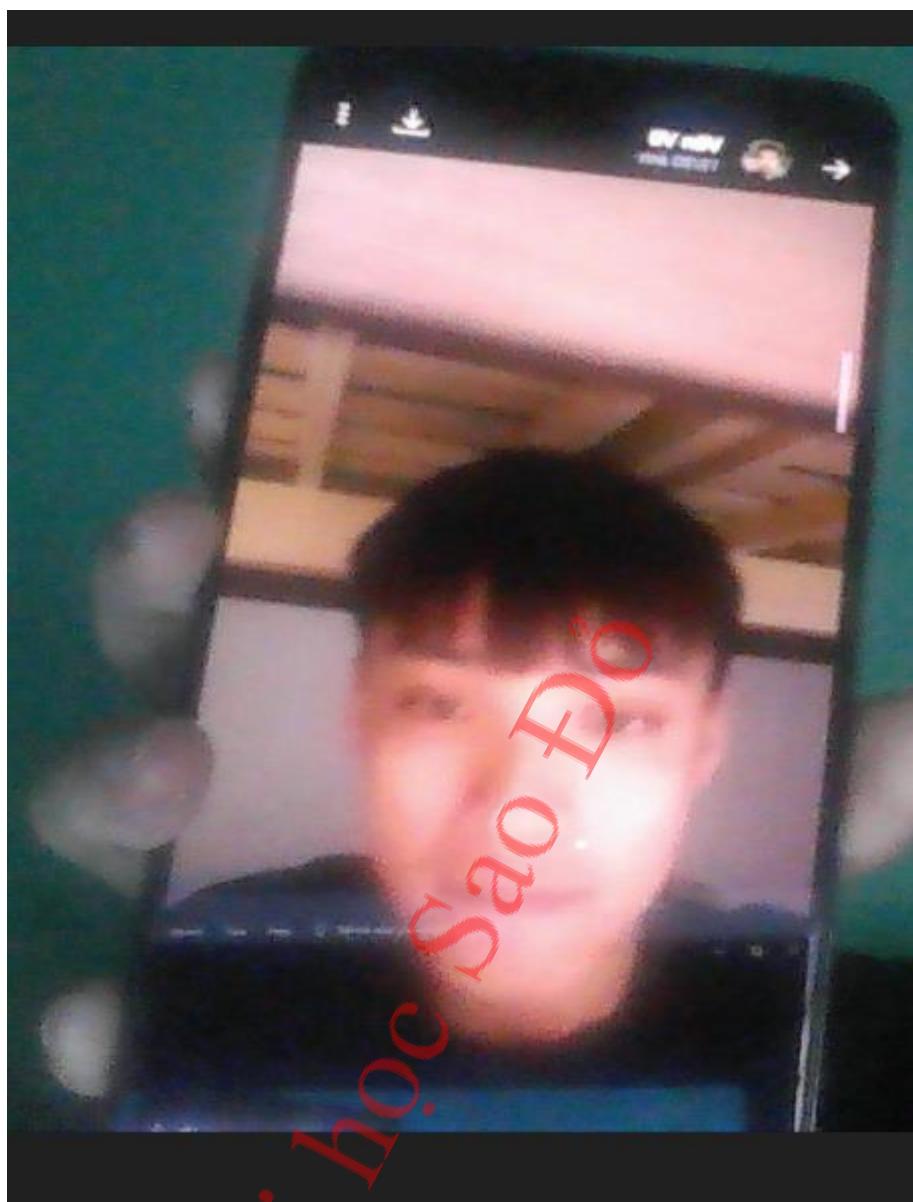
Cuối cùng, em đã áp dụng trình phát hiện khuôn mặt cho cả hai tập video để trích xuất ROI khuôn mặt riêng lẻ cho cả hai lớp, giúp xây dựng bộ dữ liệu phát hiện sự sống.



Hình 3.3. Tập dữ liệu



Hình 3.4. Hình ảnh được quay từ điện thoại



Hình 3.5. Hình ảnh giả mạo được quay lại từ camera máy tính

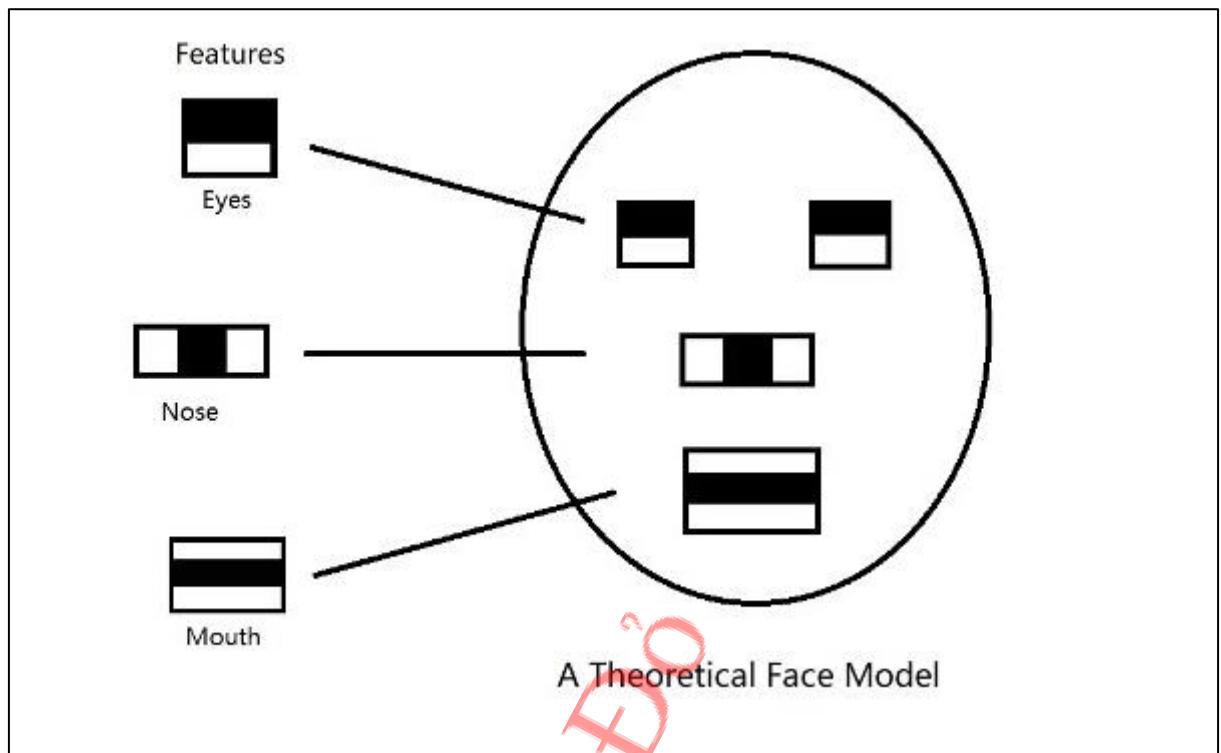
3.4. Xây dựng module

3.4.1. Module nhận diện khuôn mặt

Haar Cascade là thuật toán phát hiện đối tượng được giới thiệu bởi Paul Viola và Michael Jones nhằm phát hiện khuôn mặt trong ảnh hoặc video. Một chức năng phân tầng được đào tạo bằng cách sử dụng nhiều bức ảnh âm và dương, sau này có thể được sử dụng để xác định bất kỳ đối tượng hoặc khuôn mặt nào. Các file được đào tạo sẵn này có sẵn trong repo OpenCV GitHub.

Sử dụng phương thức tiếp cận cửa sổ trượt, một cửa sổ có kích thước cố định sẽ lặp lại hình ảnh từ trái sang phải, từ trên xuống dưới. Ở mỗi giai đoạn, cửa sổ dừng lại và phân loại xem khu vực có chứa khuôn mặt hay không.

OpenCV, một công cụ thị giác máy, hoạt động với mô hình Haar Cascade được huấn luyện trước để phân loại các đối tượng. Mỗi giai đoạn kiểm tra 5 đối tượng, hai đối tượng cạnh, hai đối tượng đường thẳng và một đối tượng bốn ô đan xen.



Hình 3.6. Nhận diện khuôn mặt

Đây là quy trình cần thực hiện để xây dựng một công cụ nhận diện khuôn mặt:

- Tải mô hình Haar Cascade Frontal Face
- Khởi tạo camera
- Đọc khung hình từ camera
- Chuyển đổi hình ảnh sang thang độ xám
- Lấy tọa độ khuôn mặt
- Vẽ một hình chữ nhật và đặt thông điệp thích hợp
- Hiển thị đầu ra

OpenCV là một thư viện thị giác máy mã nguồn mở và machine learning. Nó có hơn 2.500 thuật toán đã được tối ưu cho rất nhiều ứng dụng. Trong đó bao gồm nhận dạng, phát hiện khuôn mặt/đôi tượng, phân loại...

Rất nhiều công ty lớn như Google, IBM và cả Yahoo đã sử dụng OpenCV trong ứng dụng của họ. Dẫu vậy, trước khi bắt đầu bạn cũng nên cân nhắc về việc đảm bảo an toàn cho dữ liệu riêng tư của mình.

Để cài đặt OpenCV trong Python, sử dụng câu lệnh sau:

```
pip install opencv-python
```

Lập trình công cụ phát hiện khuôn mặt bằng Python

Bạn cần thực hiện các bước sau để lập trình một công cụ phát hiện khuôn mặt bằng Python:

Tải file Haar Cascade Frontal Face Default XML và đặt nó vào chung vị trí với chương trình Python của bạn.

Nhập thư viện OpenCV:

```
# importing the required libraries
import cv2
```

Lưu file thuật toán Haar Cascade Frontal Face để dễ tham khảo

```
# loading the haar case algorithm file into alg variable
alg = "haarcascade_frontalface_default.xml"
```

Sử dụng lớp CascadeClassifier để tải một file XML vào OpenCV:

```
# passing the algorithm to OpenCV
haar_cascade = cv2.CascadeClassifier(alg)
```

Lấy video từ camera. Đưa giá trị 0 vào hàm VideoCapture() để sử dụng camera chính của bạn.

```
# capturing the video feed from the camera
cam = cv2.VideoCapture(0)
```

Thiết lập vòng lặp vô hạn để đọc từng frame của đầu vào camera. Hàm read() trả về hai tham số. Giá trị đầu tiên là kiểu boolean để cho biết hoạt động có thành công thứ hai. Tham số thứ hai chứa khung thực tế mà bạn sẽ làm việc. Lưu khung này trong biến img.

```
while True:
    _, img = cam.read()
```

Cài đặt văn bản thông báo mặc định là Face not detected. Khi phát hiện ra khuôn mặt thì cập nhật giá trị vào biến đó.

```
text = "Face not detected"
```

Đầu vào từ thế giới thực có nhiều màu sắc nhưng trong định dạng BGR. BGR là viết tắt của blue, green và red (xanh, xanh lá và đỏ). Điều này sẽ khiến ứng dụng thị giác máy phải xử lý rất nhiều thứ. Do vậy, để giảm bớt khối lượng quy trình, chúng ta sử dụng định dạng màu xám.

```
# convert each frame from BGR to Grayscale
grayImg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Đưa các khung hình và mã chuyển đổi định dạng, COLOR_BGR2GRAY, vào cvtColor() để thay đổi từng khung hình của video từ màu sắc sang xám.

Sử dụng detectMultiScale() để phát hiện khuôn mặt. Phương thức này sử dụng ba tham số làm đầu vào. Đầu tiên là nguồn ảnh, grayImg. Tham số thứ hai là scaleFactor. Nó chỉ định mức độ bạn phải giảm kích thước ảnh ở mỗi tỷ lệ ảnh. Sử dụng giá trị mặc định 1.3 làm hệ số tỷ lệ. Hệ số tỷ lệ càng cao thì tốc độ thực hiện càng nhanh do càng ít bước cần thực hiện. Tuy nhiên, khả năng thiếu khuôn mặt cũng cao hơn. Tham số thứ ba là minNeighbors. Tham số này chỉ định số lượng hàm xóm mà mỗi hình chữ nhật phải có để giữ lại nó. Giá trị càng cao thì khả năng phát hiện sai khuôn mặt càng thấp nhưng cũng có tỷ lệ bỏ lỡ các khuôn mặt không rõ ràng cao hơn.

```
# detect faces using Haar Cascade
face = haar_cascade.detectMultiScale(grayImg, 1.3, 4)
```

Khi phát hiện ra khuôn mặt, sẽ nhận được 4 tọa độ. x đại diện cho tọa độ x, y đại diện cho tọa độ y, w đại diện cho chiều rộng và h đại diện cho chiều cao. Cập nhật thông báo văn bản là Face Detected và vẽ hình chữ nhật dựa trên các tọa độ đó. Màu sắc của hình chữ nhật là green (theo định dạng BGR) và độ dày đường viền hình chữ nhật là 2 pixel.

```
# draw a rectangle around the face and update the text to Face Detected
for (x, y, w, h) in face:
    text = "Face Detected"
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

Tùy chọn in văn bản trên bảng điều khiển đầu ra. Hiển thị văn bản trên màn hình bằng cách sử dụng khung hình đã chụp làm nguồn, văn bản thu thập được trong văn bản trên, kiểu font chữ của FONT_HERSHEY_SIMPLEX, hệ số tỷ lệ font chữ là 1, màu xanh lam, độ dày 2 pixel và kiểu dòng AA.

```
# display the text on the image
print(text)
image = cv2.putText(img, text, (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255, 0, 0), 2, cv2.LINE_AA)
```

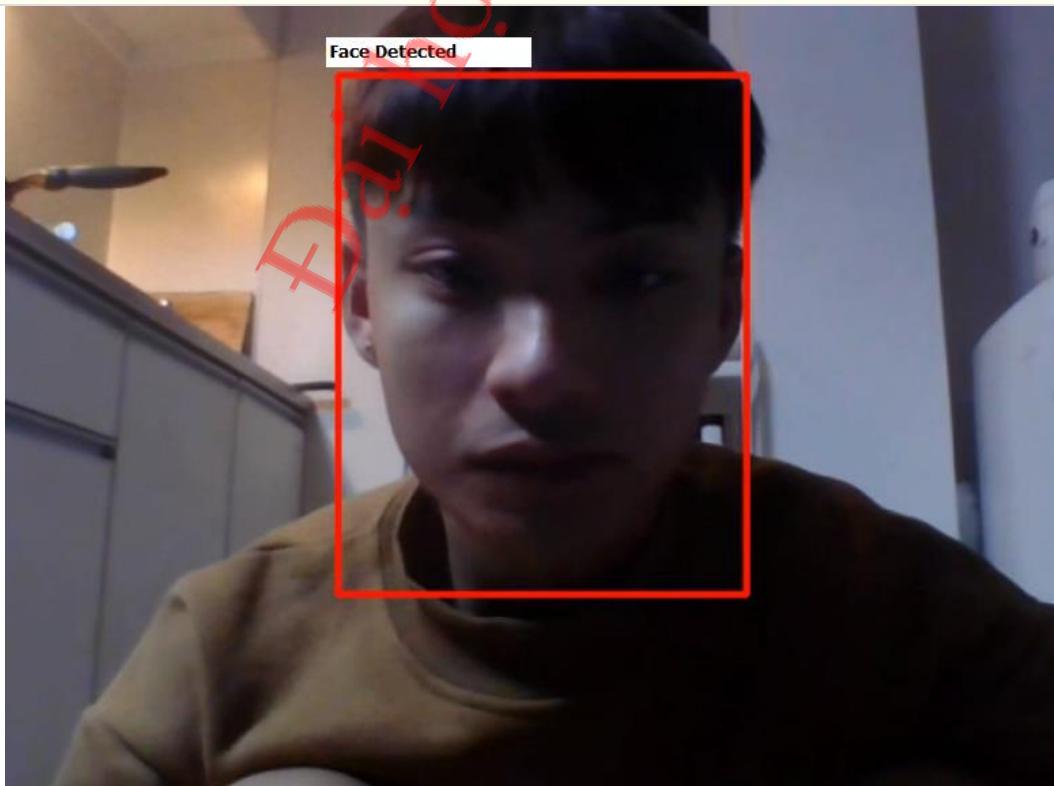
Hiển thị cửa sổ với tiêu đề Face Detection và hình ảnh. Sử dụng phương thức waitKey() để hiển thị cửa sổ trong 10 mili giây và kiểm tra một lần nhấn phím. Nếu người dùng nhấn phím Esc (Giá trị ASCII 27), hãy thoát khỏi vòng lặp.

```
# display the output window and press escape key to exit
cv2.imshow("Face Detection", image)
key = cv2.waitKey(10)

if key == 27:
    break
```

Cuối cùng, giải phóng đối tượng camera khỏi chương trình Python và đóng tất cả các cửa sổ.

```
cam.release()
cv2.destroyAllWindows()
```



Hình 3.7. Phát hiện khuôn mặt

3.4.2. Module phát hiện khuôn mặt giả

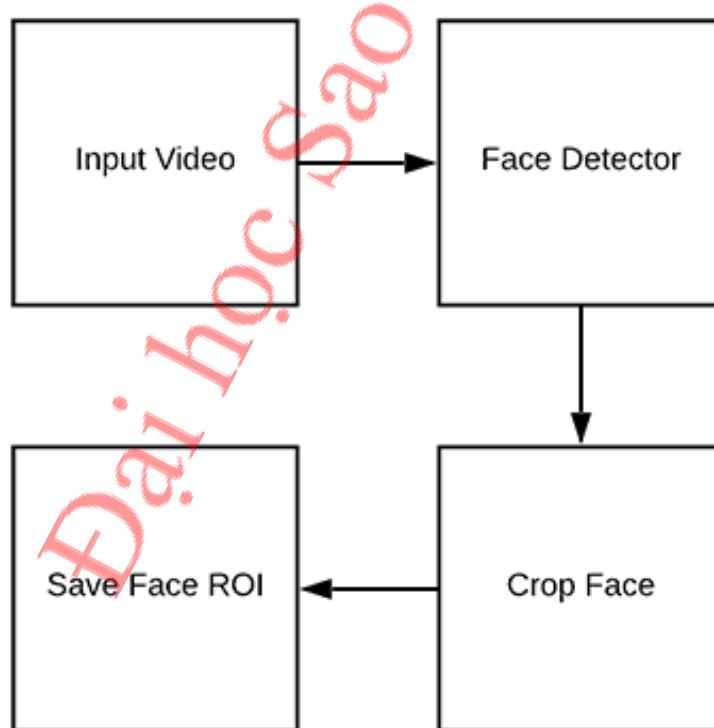
Để xây dựng một module nhận diện khuôn mặt: quá trình xử lý video để phát hiện và cắt khuôn mặt. Dưới đây là các bước chi tiết:

- Video đầu vào: Đây là bước đầu tiên trong quy trình, nơi video được cung cấp như là dữ liệu đầu vào.

- Bộ phát hiện khuôn mặt: Trong bước này, một thuật toán hoặc mô hình học máy được sử dụng để phát hiện khuôn mặt trong video. Thuật toán này có thể dựa trên nhiều kỹ thuật khác nhau, bao gồm nhưng không giới hạn ở các mạng neural tích chập (CNNs), Haar cascades, hoặc các phương pháp khác.

- Cắt khuôn mặt: Khi một khuôn mặt được phát hiện, nó được cắt ra khỏi hình ảnh gốc. Điều này thường được thực hiện bằng cách sử dụng tọa độ của hộp giới hạn xung quanh khuôn mặt.

- Lưu vùng quan tâm (ROI) của khuôn mặt: Cuối cùng, khuôn mặt đã được cắt ra được lưu lại như là một ROI (Vùng quan tâm). ROI sau đó có thể được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như huấn luyện mô hình học máy, phân tích hình ảnh, hoặc xác thực danh tính.



Hình 3.8. Quá trình xử lý video

Theo thứ tự xuất hiện trong hình:

- gather_examples.py: Tập lệnh này lấy ROI khuôn mặt từ các tệp video đầu vào và giúp em tạo bộ dữ liệu sống về khuôn mặt học sâu.

- train.py: Như tên tệp chỉ ra, tập lệnh này sẽ đào tạo trình phân loại LivenessNet của em. Chúng ta sẽ sử dụng Keras và TensorFlow để huấn luyện mô hình. Quá trình đào tạo dẫn đến một vài tệp:

- le.pickle: Bộ mã hóa nhãn lớp của em.

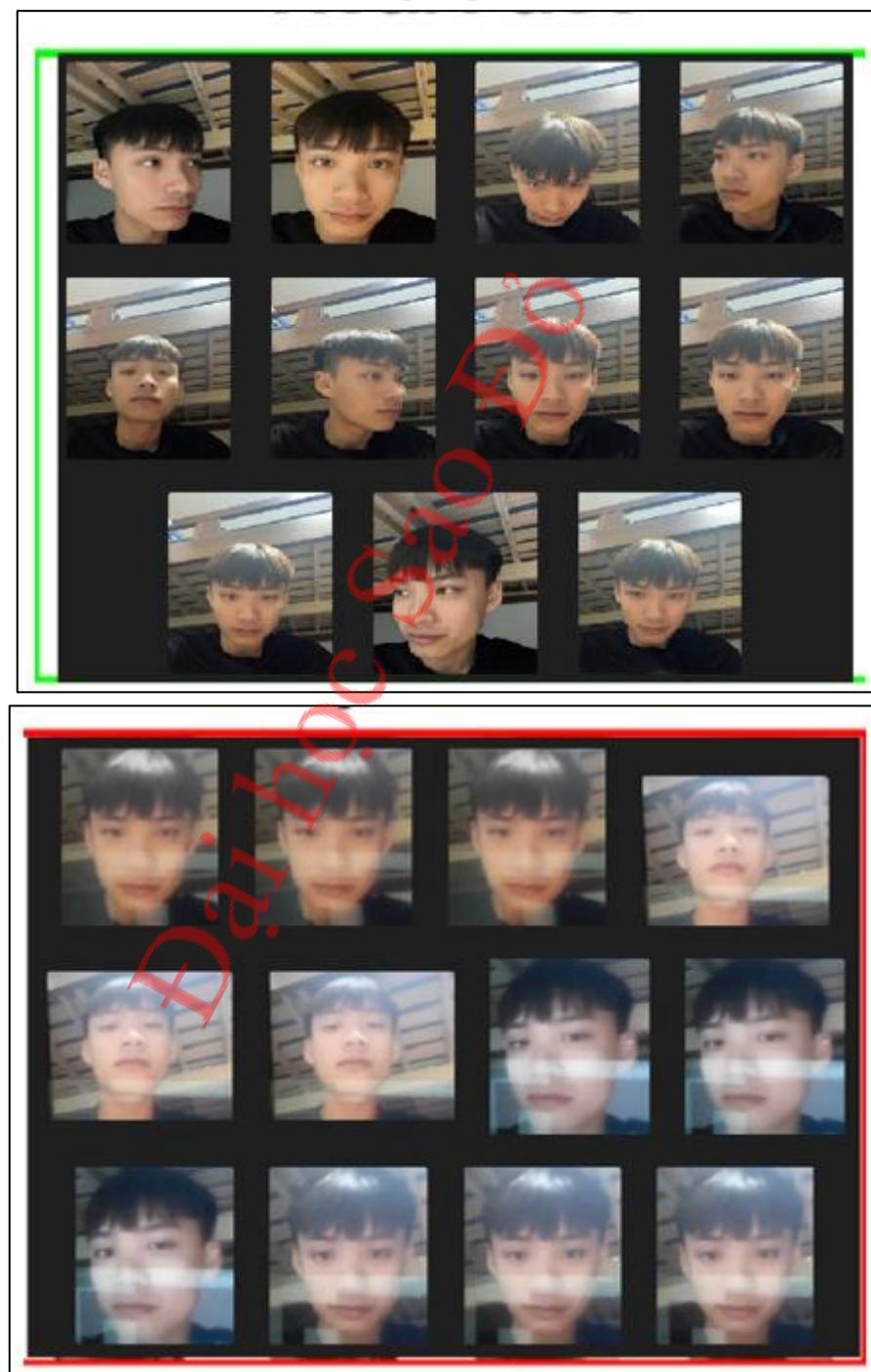
- liveness.model : Mô hình Keras nối tiếp của em phát hiện sự sống động của khuôn mặt.

- liveness_demo.py: Kịch bản trình diễn của em sẽ kích hoạt webcam của bạn để lấy khung hình để tiến hành phát hiện sự sống của khuôn mặt trong thời gian thực.

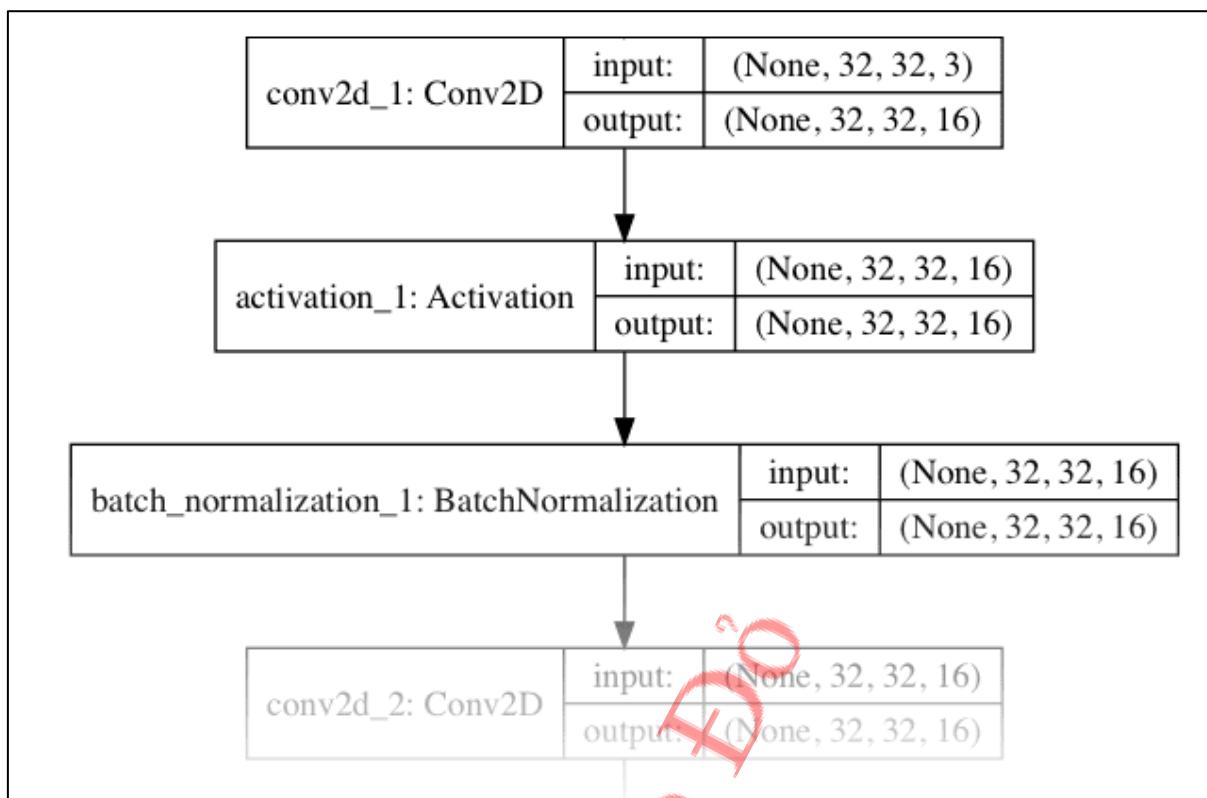
Mục tiêu cuối cùng nếu tập lệnh này sẽ là điền vào hai thư mục:

Fake/: Chứa ROI khuôn mặt từ fake.mp4 tệp

Real/: Giữ ROI khuôn mặt từ Real.mp4 tệp.



Hình 3.9. Dữ liệu trích xuất từ video



Hình 3.10. Kiến trúc LivenessNet

Hình 3.10 mô tả kiến trúc của một mạng nơ-ron tích chập (Convolutional Neural Network - CNN) được gọi là “LivenessNet”. Dưới đây là các chi tiết:

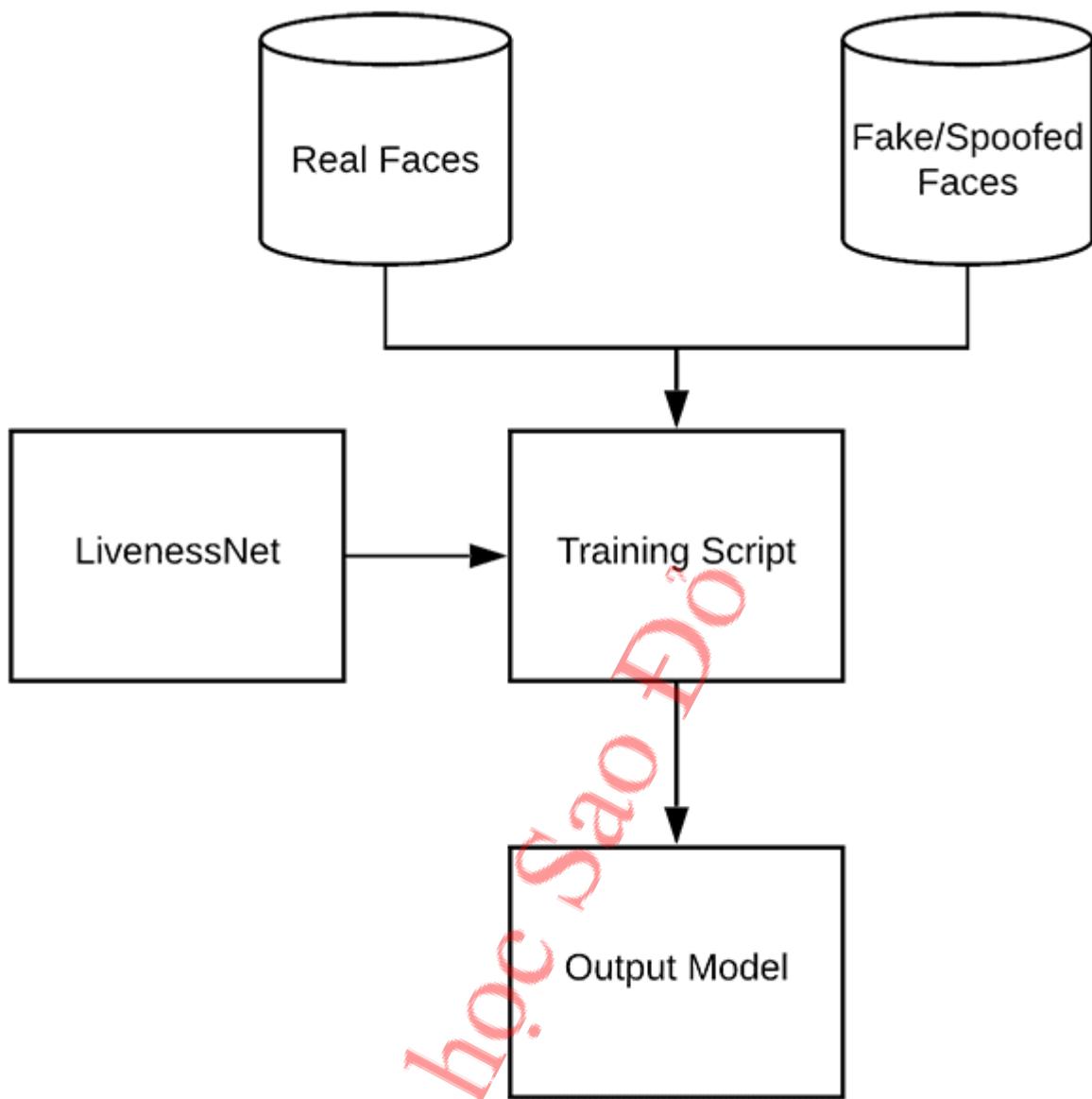
- conv2d_1: Conv2D: Đây là lớp đầu tiên của mạng, nhận đầu vào là một hình ảnh có kích thước (None, 32, 32, 3) (chiều rộng 32, chiều cao 32, và 3 kênh màu) và trả về một tensor có kích thước (None, 32, 32, 16). Lớp này áp dụng một bộ lọc tích chập lên hình ảnh đầu vào.

- activation_1: Activation: Lớp này áp dụng một hàm kích hoạt, thường là ReLU (Rectified Linear Unit), lên đầu ra của lớp Conv2D trước đó để giới hạn giá trị đầu ra trong một phạm vi cụ thể.

- batch_normalization_1: BatchNormalization: Lớp này chuẩn hóa đầu ra của lớp Activation trước đó bằng cách điều chỉnh và tỷ lệ dữ liệu, giúp tăng tốc độ học và cải thiện hiệu suất của mạng.

- conv2d_2: Conv2D: Đây là lớp Conv2D thứ hai trong mạng, nhận đầu vào từ lớp BatchNormalization và áp dụng một bộ lọc tích chập khác.

Mạng “LivenessNet” này có thể được sử dụng để phát hiện sự sống trong các hình ảnh hoặc video, như xác định xem một khuôn mặt là thực sự hay chỉ là hình ảnh



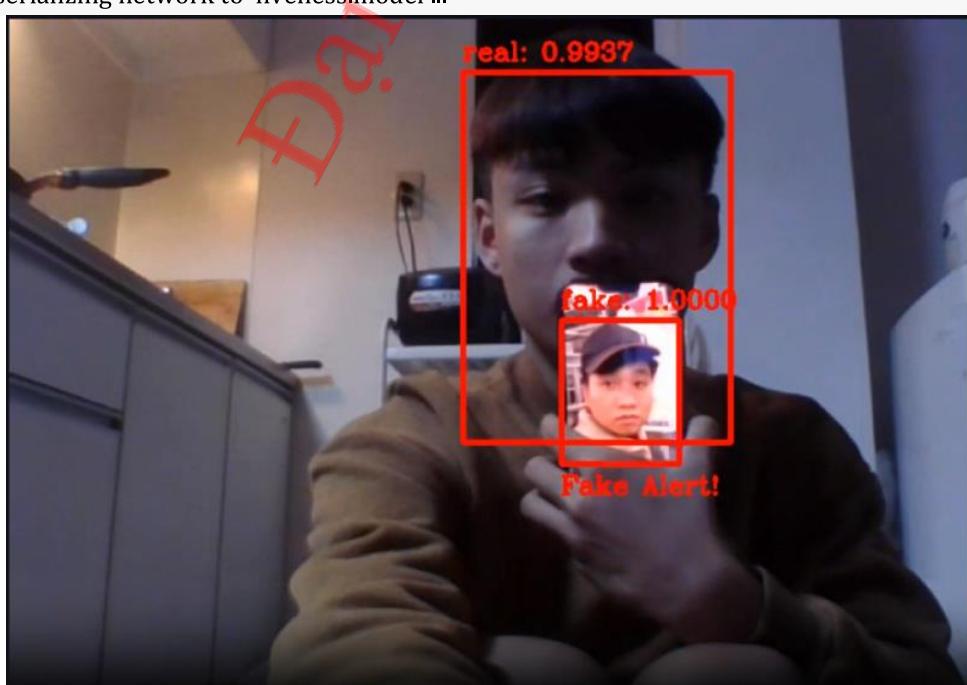
Hình 3.11. Quy trình huấn luyện module phân biệt khuôn mặt

Hình 3.11 mô tả một quy trình huấn luyện mô hình để phân biệt giữa khuôn mặt thật và khuôn mặt giả/spoofed sử dụng LivenessNet. Dưới đây là các bước chi tiết:

- Khuôn mặt thật và Khuôn mặt giả/spoofed: Đây là hai tập dữ liệu đầu vào cho quy trình. Tập dữ liệu “Khuôn mặt thật” chứa các hình ảnh của khuôn mặt thật, trong khi tập dữ liệu “Khuôn mặt giả/spoofed” chứa các hình ảnh của khuôn mặt giả.
- LivenessNet: Đây là mạng nơ-ron tích chập (Convolutional Neural Network - CNN) được sử dụng để phân loại khuôn mặt thật và khuôn mặt giả. Nó nhận đầu vào từ cả hai tập dữ liệu và trả về đầu ra dưới dạng các dự đoán về xác suất của mỗi khuôn mặt là thật hay giả.
- Training Script: Đây là bước tiếp theo trong quy trình, nơi đầu ra của LivenessNet được cung cấp cho một kịch bản huấn luyện. Kịch bản này sẽ sử dụng dữ liệu đầu vào để huấn luyện mô hình phân loại.
- Output Model: Đây là kết quả cuối cùng của quy trình, là một mô hình đã được huấn luyện có khả năng phân biệt giữa khuôn mặt thật và khuôn mặt giả.

Quá trình huấn luyện:

```
$ python train.py --dataset dataset --model liveness.model --le le.pickle
[INFO] loading images...
[INFO] compiling model...
[INFO] training network for 50 epochs...
Epoch 1/50
29/29 [=====] - 0s 13ms/step - loss: 1.2583 - accuracy: 0.5511 - val_loss: 0.6862 - val_accuracy: 0.6026
Epoch 2/50
29/29 [=====] - 0s 4ms/step - loss: 1.0099 - accuracy: 0.6089 - val_loss: 0.7426 - val_accuracy: 0.4487
Epoch 3/50
29/29 [=====] - 0s 4ms/step - loss: 0.8485 - accuracy: 0.6933 - val_loss: 0.8468 - val_accuracy: 0.4487
...
Epoch 48/50
29/29 [=====] - 0s 4ms/step - loss: 0.3170 - accuracy: 0.8800 - val_loss: 0.0666 - val_accuracy: 0.9872
Epoch 49/50
29/29 [=====] - 0s 3ms/step - loss: 0.2724 - accuracy: 0.8889 - val_loss: 0.0413 - val_accuracy: 1.0000
Epoch 50/50
29/29 [=====] - 0s 3ms/step - loss: 0.3573 - accuracy: 0.8533 - val_loss: 0.0293 - val_accuracy: 1.0000
[INFO] evaluating network...
precision recall f1-score support
fake 1.00 1.00 1.00 35
real 1.00 1.00 1.00 43
accuracy 1.00 78
macro avg 1.00 1.00 1.00 78
weighted avg 1.00 1.00 1.00 78
[INFO] serializing network to 'liveness.model'...
```



Hình 3.12. Kết quả của module phân biệt khuôn mặt

3.5. Xây dựng website demo

Thiết lập một lớp xử lý video (trong đoạn mã là videoprocessor) để xử lý các khung hình từ luồng video.

Truyền các khung hình từ luồng video qua cả hai module để nhận diện khuôn mặt và đánh giá tính thật giả.

Hiển thị kết quả trực tiếp trên video và cung cấp thông báo khi phát hiện khuôn mặt giả.

Để tích hợp cả hai module và hiển thị kết quả trực tiếp trên video, ta sử dụng lớp videoprocessor:

class VideoProcessor:

```
def recv(self, frame):
    frm = frame.to_ndarray(format="bgr24")
    # Lặp qua các khung hình từ luồng video
    frm = imutils.resize(frm, width=800)
    # ... (các bước xử lý khác)
    # Lặp qua các phát hiện
    for i in range(0, detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > args['confidence']:
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype('int')
            # ... (các bước xử lý khác)
            if label_name == 'fake':
                cv2.putText(frm, "Alert! Khuôn mặt giả!", (startX, endY + 25),
                           cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
            # ... (các bước xử lý khác)
    return av.VideoFrame.from_ndarray(frm, format='bgr24')
```

Lớp VideoProcessor nhận các khung hình từ luồng video, thực hiện xử lý nhận diện khuôn mặt và đánh giá tính thật giả, và hiển thị kết quả trực tiếp trên video.

Hiển thị video thời gian thực:

Để hiển thị video thời gian thực và kết quả nhận dạng khuôn mặt, chúng ta sử dụng thư viện streamlit_webrtc. Đây là một công cụ mạnh mẽ để tạo giao diện web thời gian thực trong ứng dụng streamlit.

```
94
95     webrtc_streamer(key="key", video_processor_factory=VideoProcessor, rtc_configuration={
96         "iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]
97     }, sendback_audio=False, video_receiver_size=1)
```

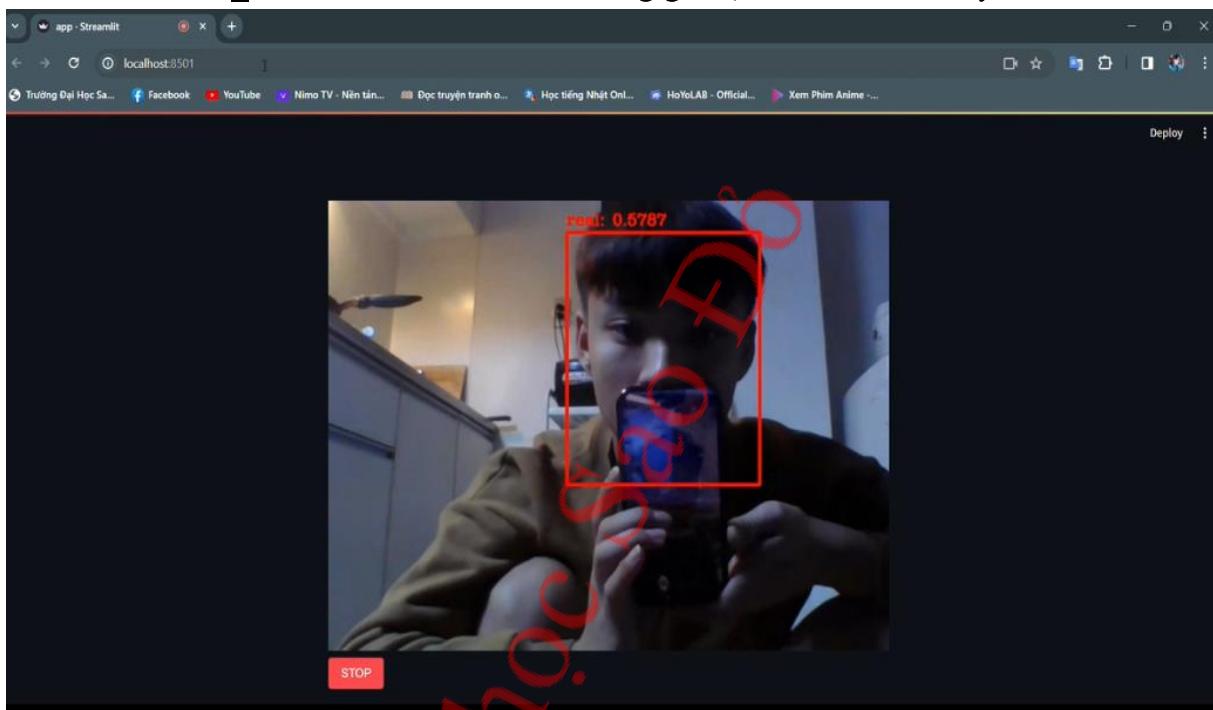
Hình 3.13. Module Streamlist

```
# Sử dụng thư viện streamlit_webrtc để hiển thị video thời gian thực
webrtc_streamer(key="key", video_processor_factory=VideoProcessor,
rtc_configuration={
```

```
"iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}],  
, sendback_audio=False, video_receiver_size=1)
```

Trong đoạn mã trên:

- webrtc_streamer được sử dụng để tạo một giao diện webRTC, cho phép chúng ta stream video thời gian thực từ camera máy tính.
- video_processor_factory=VideoProcessor kết hợp với lớp VideoProcessor đã được định nghĩa trước đó để xử lý các khung hình video.
- rtc_configuration là cấu hình WebRTC, đảm bảo kết nối đúng đắn.
- sendback_audio=False để tắt chức năng gửi lại âm thanh từ máy tính.



Hình 3.14. Triển khai website demo

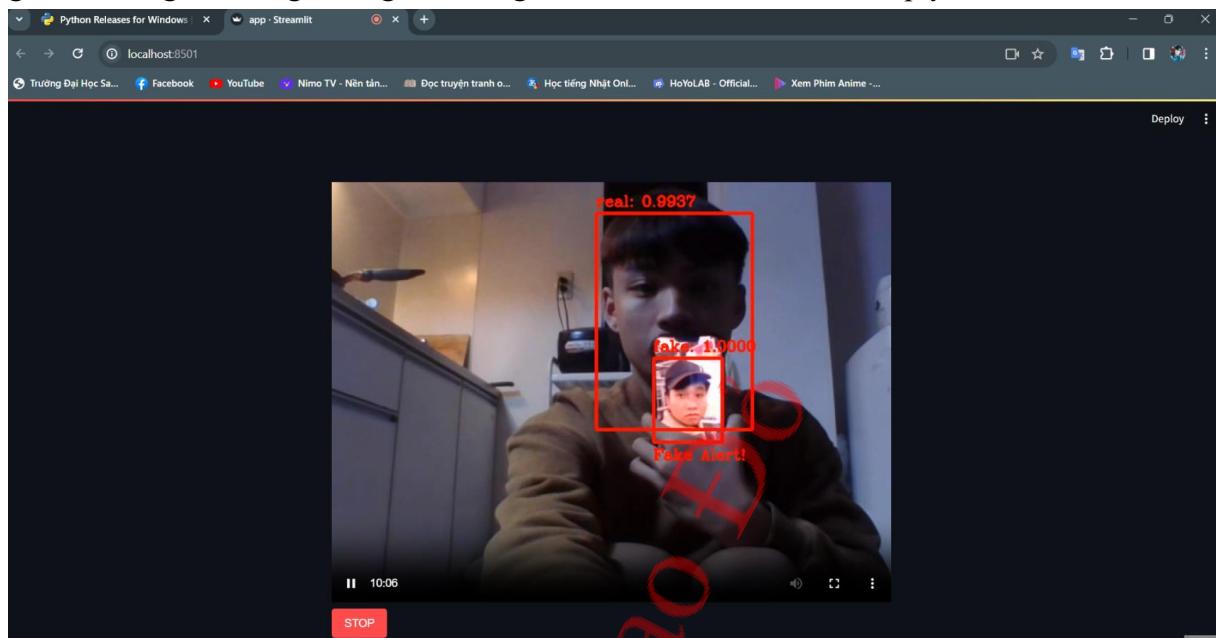
```
Bo [INFO] unknown, real  
Bo [INFO] [*****] - 0s 20ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 31ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 29ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 28ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 31ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 20ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 24ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 22ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 21ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 27ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 20ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 27ms/step  
Bo [INFO] Unknown, fake  
Bo [INFO] [*****] - 0s 32ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 28ms/step  
Bo [INFO] Unknown, fake  
Bo [INFO] [*****] - 0s 24ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 22ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 24ms/step  
Bo [INFO] Unknown, fake  
Bo [INFO] [*****] - 0s 20ms/step  
Bo [INFO] Unknown, real  
Bo [INFO] [*****] - 0s 22ms/step  
Bo [INFO] Unknown, real
```

Hình 3.15. Kết quả trả về sau khi chạy ứng dụng

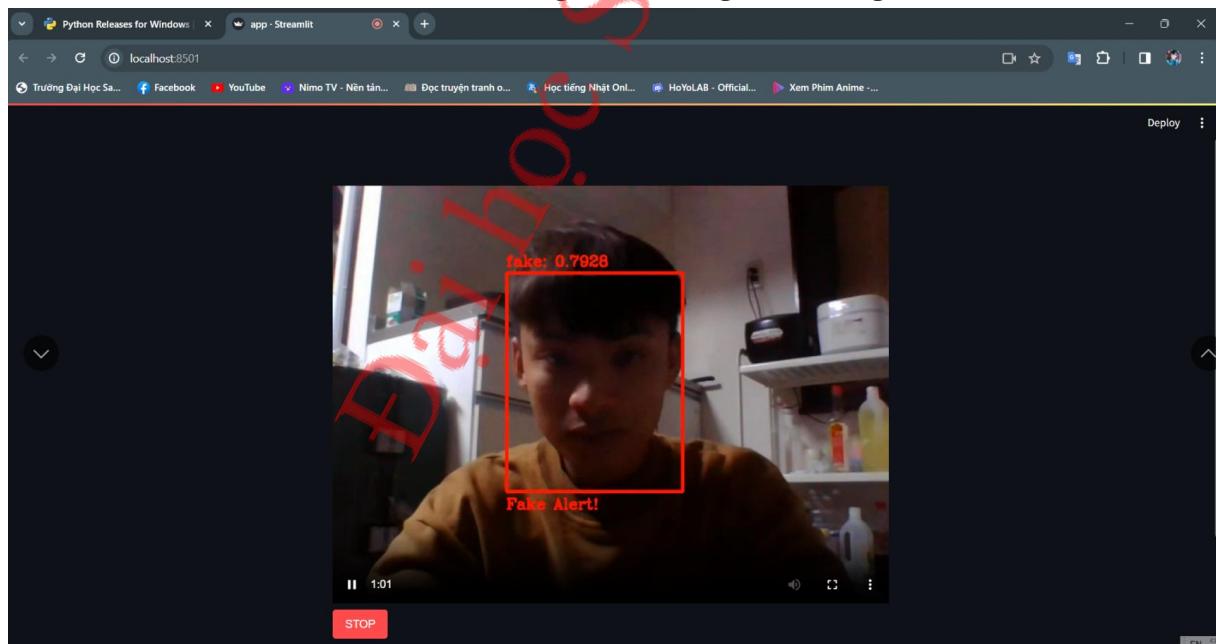
3.6. Thủ nghiệm

Nhận diện tính thật/giả

Hệ thống thực hiện nhận diện khuôn mặt từ luồng video và áp dụng mô hình máy học để phân loại tính thật hay giả của khuôn mặt. Kết quả được hiển thị trực tiếp trên giao diện người dùng, bao gồm thông tin về tên và xác suất của quyết định.



Hình 3.16. Trong môi trường thiếu sáng



Hình 3.17. Trong điều kiện tối

Bảng 3.1. Hiệu suất và độ chính xác

Môi trường thử nghiệm	Ánh sáng	Tốc độ phản hồi	Độ chính xác	Ghi chú
Ánh sáng bình thường	Đủ sáng	23~25ms	99%	Thử nghiệm video thời gian thực kết quả cho ra chính xác

Môi trường thử nghiệm	Ánh sáng	Tốc độ phản hồi	Độ chính xác	Ghi chú
Ánh sáng yếu	Thiếu sáng	23~25ms	95%	Thử nghiệm video thời gian thực cho ra kết quả tương đối chính xác
Môi trường quá tối	Không đủ ánh sáng	23~25ms	80%	Trong môi trường quá tối kết quả đưa ra không được khả quan, còn sai sót

Hệ thống đã được kiểm thử hiệu suất dưới nhiều điều kiện khác nhau, bao gồm cả ánh sáng môi trường và độ nhiễu. Hiệu suất của hệ thống ổn định, và độ chính xác trong việc phân biệt khuôn mặt tính thật và giả mạo đáng tin cậy.

Đại học Sao Đỏ

KẾT LUẬN

1. Kết quả đạt được

Hiệu suất đáng kể: Trong đề tài này, em đã xây dựng một hệ thống nhận diện khuôn mặt và phân biệt tính thật giả với hiệu suất đáng kể. Hệ thống của em có thể hoạt động trong ứng dụng thời gian thực trên trình duyệt web và có thể xử lý các tình huống phức tạp về ánh sáng và đa dạng khuôn mặt.

Tích hợp linh hoạt: Để tạo ra một giao diện người dùng trực quan và dễ sử dụng, em đã tích hợp hệ thống với thư viện streamlit_webrtc. Việc tích hợp này cho phép người dùng thực hiện việc nhận diện khuôn mặt và phân biệt tính thật giả trên trình duyệt web.

2. Hạn chế của đề tài

Mặc dù hệ thống của em đã đạt được nhiều thành tựu, tuy nhiên còn một số hạn chế cần được cải thiện để tăng tính ổn định và mở rộng khả năng ứng dụng.

Cải thiện hiệu suất: Để tối ưu hóa mô hình nhận diện khuôn mặt và có hiệu suất tốt hơn trong điều kiện ánh sáng đặc biệt và đa dạng, em cần cải thiện hiệu suất của hệ thống.

Mở rộng tính năng: Để mở rộng khả năng ứng dụng của hệ thống, em có thể bổ sung các tính năng mới như xác thực bằng giọng nói, nhận dạng khuôn mặt từ xa hoặc tích hợp với các phương pháp xác thực khác.

Nâng cao tính ổn định: Để nâng cao tính ổn định của hệ thống, em cần phải tối ưu hóa các tham số và cải thiện khả năng đáp ứng của hệ thống.

3. Kết luận

Em đã xây dựng thành công một hệ thống nhận diện khuôn mặt và phân biệt tính thật giả với hiệu suất đáng kể và tích hợp linh hoạt. Tuy nhiên, còn một số hạn chế cần được cải thiện để tăng tính ổn định và mở rộng khả năng ứng dụng của hệ thống. Em hy vọng rằng đề tài này sẽ cung cấp thông tin hữu ích cho những ai quan tâm đến lĩnh vực nhận diện khuôn mặt và phân biệt tính thật giả.

TÀI LIỆU THAM KHẢO

- [1] Đại học Sao Đỏ (2022), *Giáo trình Xử lý ảnh*
- [2] Đặng Nguyên Châu, Đỗ Hồng Tuấn (2017), *Tổng quan các phương pháp nhận dạng khuôn mặt dựa trên đặc trưng cạnh*, ISSN 1859-1531, Tạp chí khoa học và công nghệ đại học Đà Nẵng, Số 05(114).2017-Qュênh 2.
- [3] Nguyễn Thanh Tuấn (2020), *Deeplearning cơ bản*, tái bản lần 2, ebook, <https://drive.google.com/file/d/1INjzISABdoc7SRq8tg-xkCRRZRABPCKi/view>
- [4] B. Takács (1998), *Comparing face images using the modified Hausdorff distance*, *Pattern Recognition*, Vol. 31, 1998, pp. 1873-1881.
- [5] Y. Gao and M. K. Leung (2002), *Face recognition using line edge map*, IEEE Trans. on Pattern and Analysis and Machine Intelligence, Vol. 24, No. 6, Jun 2002, pp. 764-779.
- [6] Heath, M.D., Sarkar, S., Sanocki, T., and Bowyer, K.W.(1998), *Comparison of edge detectors: A methodology and initial study*, *Comput. Vis. Image Underst.*, Vol. 69, 1998, pp. 38–54.
- [7] Nevatia, R., and Babu, K.R. (1980), *Linear feature extraction and description*, *Comput. Graph. Image Process.*, Vol. 13, 1980, pp. 257–269.
- [8] Leung, M.K.H., and Yang, Y.H. (1990), *Dynamic two-strip algorithm in curve fitting*, *Pattern Recognition*, Vol. 23, 1990, pp. 69–79.
- [9] Huttenlocher, D.P., Klanderman, G.A., and Rucklidge, W.J. (1993), *Comparing images using the Hausdorff distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, 1993, pp. 850–863.
- [10] Y. Gao (2003), *Efficiently comparing face images using a modified Hausdorff distance*, IEE Proc. Vision, Image and Signal Processing, Vol. 150, No. 6, Dec 2003, pp. 346-350.
- [11] C. Silva, T. Bouwmans, C. Frelicot (2015), *An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos*, VISAPP 2015, Berlin, Germany.
- [12] Tereza Soukupová and Jan Čech (2016), *Real-Time Eye Blink Detection using Facial Landmarks*, 21st Computer Vision Winter Workshop
- [13] Saptarshi Chakraborty, Dhrubajyoti Das (2014), *An overview of face liveness detection*, *International Journal on Information Theory (IJIT)*, Vol.3, No.2, April 2014
- [14] Adrian Rosebrock (2019), *Liveness Detection with OpenCV*, <https://pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>