

**SINH VIÊN: NGUYỄN MINH ĐỨC**

**NGÀNH KHOA HỌC MÁY TÍNH**

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**ĐỀ TÀI:**

**NGHIÊN CỨU MỘT SỐ KỸ THUẬT XỬ LÝ ẢNH, XÂY DỰNG**  
**ỨNG DỤNG NHẬN BIẾT CẢM XÚC QUA BIỂU CẢM KHUÔN**  
**MẶT**

<b>CBHD:</b>	<b>TS. ĐỖ MẠNH HÙNG</b>
<b>Sinh viên thực hiện:</b>	<b>NGUYỄN MINH ĐỨC</b>
<b>Mã sinh viên:</b>	<b>2020601169</b>
<b>Lớp:</b>	<b>2020DHK HMT01 – K15</b>

**Hà Nội – Năm 2024**

**SINH VIÊN: NGUYỄN MINH ĐỨC**

**NGÀNH KHOA HỌC MÁY TÍNH**

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**ĐỀ TÀI:**

**NGHIÊN CỨU MỘT SỐ KỸ THUẬT XỬ LÝ ẢNH, XÂY DỰNG  
ỨNG DỤNG NHẬN BIẾT CẢM XÚC QUA BIỂU CẢM KHUÔN  
MẶT**

<b>CBHD:</b>	<b>TS. ĐỖ MẠNH HÙNG</b>
<b>Sinh viên thực hiện:</b>	<b>NGUYỄN MINH ĐỨC</b>
<b>Mã sinh viên:</b>	<b>2020601169</b>
<b>Lớp:</b>	<b>2020DHKGMT01 – K15</b>

**Hà Nội – Năm 2024**

## MỤC LỤC

DANH MỤC HÌNH VẼ.....	iii
LỜI NÓI ĐẦU .....	iv
MỞ ĐẦU.....	1
NỘI DUNG .....	8
CHƯƠNG 1 TỔNG QUAN VỀ XỬ LÝ ẢNH.....	8
1.1 Xử lý ảnh, các vấn đề cơ bản trong xử lý ảnh.....	8
1.1.1 Xử lý ảnh .....	8
1.1.2 Các vấn đề cơ bản trong xử lý ảnh .....	12
1.2 Công nghệ nhận dạng khuôn mặt.....	19
1.2.1 Nhận dạng khuôn mặt là gì?.....	19
1.2.2 Một số lợi ích của hệ thống nhận dạng khuôn mặt .....	21
1.3 Giới thiệu về nhận dạng cảm xúc qua khuôn mặt.....	22
CHƯƠNG 2 CÁC PHƯƠNG PHÁP PHÁT HIỆN KHUÔN MẶT.....	24
2.1 Giới Thiệu.....	24
2.2 Phương Pháp Dựa Trên Mạng Nơ-ron.....	24
2.2.1 Mạng thần kinh (Neural Networks).....	24
2.2.2 Hàm mất mát và Tối ưu hóa (Loss function và Optimization) .....	35
2.2.3 Model feedforward và backpropagation.....	38
2.3 Phương Pháp Kết Hợp.....	39
2.4 Phương Pháp Dựa Trên Đặc Trưng.....	39
CHƯƠNG 3 THỰC NGHIỆM.....	41
3.1 Cơ sở dữ liệu .....	41

3.1.1	Bộ dataset Giới tính (Gender) .....	41
3.1.2	Bộ dataset Cảm xúc (Emotion) .....	42
3.2	Triển khai thực nghiệm .....	43
3.2.1	Triển khai xây dựng mô hình training với tập dữ liệu Cảm xúc .....	43
3.2.2	Triển khai xây dựng mô hình training với tập dữ liệu giới tính .....	52
3.2.3	Triển khai giao diện cơ bản cho ứng dụng nhận diện khuôn mặt ...	52
3.3	Kết quả thực nghiệm .....	57
3.3.1	Lưu model .....	57
3.3.2	Giao diện demo .....	57
3.3.3	Kết quả đạt được .....	58
KẾT LUẬN .....		59
TÀI LIỆU THAM KHẢO .....		60

## DANH MỤC HÌNH VẼ

Hình 1-1: Quá trình xử lý ảnh.....	8
Hình 1-2: Các bước cơ bản trong một hệ thống xử lý ảnh .....	9
Hình 1-3: Ảnh thu nhận và ảnh mong muốn.....	13
Hình 1-4: Công nghệ nhận dạng khuôn mặt .....	21
Hình 2-1: AI vs. ML vs. DL.....	25
Hình 2-2: Nơ-ron.....	26
Hình 2-3: Mô hình tổng quát Neural Network.....	27
Hình 2-4: ReLU Layer .....	31
Hình 2-5: Các biến thể của ReLU .....	32
Hình 2-6: Các phương pháp chuẩn hóa.....	33
Hình 2-7: Minh họa về Drop Layer .....	35
Hình 2-8: Model feedforward và backpropagation.....	38
Hình 3-1: Hình ảnh minh họa Male .....	41
Hình 3-2: Hình ảnh minh họa Female.....	42
Hình 3-3: Một số hình ảnh minh họa bộ dataset FER-2013 .....	43
Hình 3-4: Kết quả training emotion-model.....	51
Hình 3-5: Kết quả training gender-model .....	52
Hình 3-6: Ảnh lưu model .....	57
Hình 3-7: Giao diện.....	57
Hình 3-8: Kết quả nhận diện male-happy .....	58
Hình 3-9: Kết quả nhận diện male-neutral.....	58

## LỜI NÓI ĐẦU

Trong thời đại ngày nay, sự phát triển nhanh chóng của công nghệ thông tin đã mở ra những cánh cửa mới đầy tiềm năng cho nhiều lĩnh vực ứng dụng. Tôi tin rằng lĩnh vực xử lý ảnh đang trở thành một trong những lĩnh vực nổi bật, đặc biệt là trong việc nghiên cứu và phát triển các ứng dụng nhận biết cảm xúc thông qua biểu cảm khuôn mặt. Đây không chỉ là một đề tài nghiên cứu mà còn là một xu hướng quan trọng đánh dấu sự giao thoa giữa trí tuệ nhân tạo và tâm lý học.

Cùng với sự phổ biến của các phương tiện truyền thông xã hội và ứng dụng trực tuyến, nhu cầu trong việc hiểu và tương tác với cảm xúc con người đã trở nên ngày càng quan trọng. Các ứng dụng nhận biết cảm xúc qua biểu cảm khuôn mặt không chỉ có thể hỗ trợ trong việc phát triển trải nghiệm người dùng mà còn có ứng dụng rộng rãi trong lĩnh vực y tế, giáo dục, quảng cáo, và nhiều lĩnh vực khác.

Điều này đặt ra thách thức lớn đối với cộng đồng nghiên cứu về xử lý ảnh khi phải đối mặt với sự đa dạng và phức tạp của biểu cảm khuôn mặt. Trong bối cảnh này, tôi tin rằng đề tài nghiên cứu về "Một số kỹ thuật xử lý ảnh, xây dựng ứng dụng nhận biết cảm xúc qua biểu cảm khuôn mặt" trở nên hết sức quan trọng và hứa hẹn mang lại những đóng góp tích cực cho sự phát triển của lĩnh vực này.

Chương trình nghiên cứu của tôi sẽ tập trung vào việc phân tích, thiết kế, và triển khai các kỹ thuật xử lý ảnh mới nhằm nâng cao độ chính xác và hiệu suất của hệ thống nhận biết cảm xúc. Đồng thời, chúng tôi cũng sẽ khám phá ứng dụng thực tế của nghiên cứu này thông qua việc xây dựng một ứng dụng có khả năng nhận diện và đánh giá cảm xúc dựa trên biểu cảm khuôn mặt.

Tôi hy vọng rằng, thông qua đề tài này, chúng ta sẽ có cơ hội nhìn nhận sâu sắc hơn về sức mạnh của kỹ thuật xử lý ảnh và khả năng ứng dụng chúng trong việc hiểu và tương tác với thế giới xung quanh, đặc biệt là trong lĩnh vực nhận biết cảm xúc qua biểu cảm khuôn mặt.

## MỞ ĐẦU

### I. Lý do chọn đề tài

Trong những năm gần đây, công nghệ xử lý ảnh đã có những bước tiến vượt bậc, đặc biệt là trong lĩnh vực nhận dạng và phân loại hình ảnh. Một trong những ứng dụng quan trọng của công nghệ này là nhận dạng cảm xúc khuôn mặt, lĩnh vực có tiềm năng ứng dụng rộng rãi trong thực tế. Khuôn mặt của con người biểu hiện nhiều cảm xúc mà không cần phải nói ra lời, là một trong những phương tiện mạnh mẽ và tự nhiên nhất để con người truyền đạt và thể hiện cảm xúc.

Không giống như các hình thức giao tiếp phi ngôn ngữ khác, cảm xúc trên khuôn mặt là phổ quát và dễ dàng nhận biết. Hiện nay, việc nhận dạng và phân tích cảm xúc khuôn mặt tự động là một vấn đề thú vị và đầy thách thức, có ảnh hưởng to lớn đến xã hội. Cảm xúc và hành động trên khuôn mặt chiếm phần lớn trong giao tiếp phi ngôn ngữ của con người, bao gồm 93% cảm xúc giao tiếp của con người, trong đó 55% được thể hiện qua cử chỉ và hành động trên khuôn mặt. Cảm xúc khuôn mặt có thể được phân tích dễ dàng thông qua hình ảnh và máy tính có thể tương tác với con người như cách con người tương tác với nhau. Đó là lý do tại sao nhận dạng cảm xúc qua khuôn mặt ngày càng được sự quan tâm trong mọi lĩnh vực. Các nhà nghiên cứu đã chỉ ra rằng cảm xúc trên khuôn mặt là phổ quát và bẩm sinh trong tất cả các chủng tộc, giới tính và độ tuổi. Bên cạnh cảm xúc trung tính, có bảy cảm xúc cơ bản, bao gồm: trung tính, giận dữ, ghê tởm, sợ hãi, hạnh phúc, buồn và bất ngờ. Hệ thống nhận diện cảm xúc khuôn mặt có thể được sử dụng trong nhiều lĩnh vực khác nhau.

Ví dụ, hệ thống nhận diện cảm xúc có thể được sử dụng để phát hiện trạng thái tâm lý của học sinh, từ đó có những biện pháp giáo dục phù hợp. Ví dụ, hệ thống có thể phát hiện những học sinh đang buồn chán, từ đó giáo viên có thể đưa ra các biện pháp hỗ trợ kịp thời và hiệu quả. Hệ thống cũng có thể theo dõi tình

trạng sức khỏe của bệnh nhân, đặc biệt là những bệnh nhân mắc các bệnh tâm lý. Ví dụ, hệ thống có thể phát hiện những bệnh nhân đang lo lắng, từ đó bác sĩ có thể đưa ra các biện pháp điều trị thích hợp. Ngoài ra, hệ thống cũng có thể được sử dụng để phát hiện những hành vi đáng ngờ, từ đó ngăn ngừa các vụ tội phạm. Ví dụ, hệ thống có thể phát hiện những người đang tức giận, từ đó cảnh sát có thể có những biện pháp ngăn ngừa kịp thời. Hơn nữa, hệ thống này còn có thể cải thiện khả năng giao tiếp giữa con người và máy móc. Ví dụ, hệ thống có thể hiểu được cảm xúc người dùng, từ đó máy móc có thể đưa ra những phản hồi phù hợp.

Hệ thống nhận diện cảm xúc còn có thể ứng dụng trong các lĩnh vực giải trí và dịch vụ khách hàng. Trong ngành công nghiệp giải trí, các hệ thống này có thể giúp cải thiện trải nghiệm người dùng bằng cách cá nhân hóa nội dung dựa trên cảm xúc của khán giả. Ví dụ, một hệ thống phát hiện rằng khán giả đang buồn chán có thể thay đổi nội dung để phù hợp hơn, hoặc gợi ý các chương trình giải trí khác nhau. Trong lĩnh vực dịch vụ khách hàng, các hệ thống nhận diện cảm xúc có thể giúp nhân viên dịch vụ hiểu rõ hơn về trạng thái tâm lý của khách hàng, từ đó đưa ra các phản hồi kịp thời và phù hợp, nâng cao chất lượng dịch vụ.

Hệ thống nhận diện cảm xúc khuôn mặt cũng có thể đóng vai trò quan trọng trong lĩnh vực chăm sóc sức khỏe. Bằng cách theo dõi biểu cảm khuôn mặt của bệnh nhân, các bác sĩ và nhân viên y tế có thể phát hiện sớm các dấu hiệu của stress, lo âu, hoặc các vấn đề tâm lý khác, từ đó đưa ra biện pháp điều trị kịp thời. Điều này đặc biệt quan trọng trong việc chăm sóc người cao tuổi và những người có nguy cơ mắc các bệnh tâm lý.

Lĩnh vực nhận diện cảm xúc khuôn mặt vẫn còn đang trong giai đoạn phát triển, với nhiều thách thức cần được giải quyết. Tuy nhiên, với sự phát triển nhanh chóng của công nghệ xử lý ảnh và trí tuệ nhân tạo, các kỹ thuật nhận diện cảm xúc khuôn mặt đã được cải thiện đáng kể về độ chính xác và hiệu suất. Điều này làm tăng khả năng ứng dụng rộng rãi của các hệ thống này trong thực tế. Do đó, việc



nghiên cứu và phát triển đề tài "Nhận diện cảm xúc khuôn mặt người bằng xử lý ảnh" là khả thi và có nhiều triển vọng trong tương lai.

Với những lý do trên, tôi đã chọn đề tài này để nghiên cứu và phát triển.

## **II. Mục tiêu nghiên cứu**

Dựa trên những lý do đã nêu trong phần lý do chọn đề tài, tôi xác định các mục tiêu cần đạt được của đề tài như sau:

**Tạo ra một hệ thống nhận diện cảm xúc khuôn mặt người có độ chính xác cao:**

- Hệ thống cần có khả năng phân biệt chính xác các cảm xúc cơ bản của con người, bao gồm các trạng thái vui, buồn, giận, sợ và ngạc nhiên.
- Đảm bảo rằng hệ thống có thể hoạt động hiệu quả trong các tình huống thực tế và điều kiện khác nhau.

**Phát triển các kỹ thuật nhận dạng cảm xúc khuôn mặt mới:**

- Cải thiện độ chính xác và hiệu quả của hệ thống thông qua việc phát triển các kỹ thuật mới.
- Các kỹ thuật mới cần giải quyết được các thách thức hiện nay của lĩnh vực nhận diện cảm xúc khuôn mặt, như độ phức tạp của khuôn mặt, điều kiện ánh sáng không thuận lợi, và biểu cảm khuôn mặt tinh tế.
- Tích hợp các kỹ thuật học sâu (deep learning) và trí tuệ nhân tạo (AI) tiên tiến vào hệ thống.

**Ứng dụng hệ thống nhận diện cảm xúc khuôn mặt vào thực tế:**

- Thử nghiệm và đánh giá hệ thống trong các lĩnh vực như giáo dục, y tế, bảo mật và giao tiếp.
- Tạo ra các giải pháp thực tế và khả thi để áp dụng hệ thống vào cuộc sống hàng ngày, từ việc theo dõi trạng thái tâm lý của học sinh, tình trạng

sức khỏe của bệnh nhân, đến việc phát hiện các hành vi đáng ngờ và cải thiện khả năng giao tiếp giữa con người và máy móc.

Để đạt được các mục tiêu trên, tôi sẽ thực hiện các nhiệm vụ sau:

**Nghiên cứu các kỹ thuật nhận dạng cảm xúc khuôn mặt hiện có:**

- Đánh giá các phương pháp và kỹ thuật hiện đang được sử dụng trong lĩnh vực này.
- Phân tích các ưu điểm và nhược điểm của từng kỹ thuật để làm cơ sở cho việc phát triển các phương pháp mới.

**Xây dựng và phát triển các kỹ thuật nhận dạng cảm xúc khuôn mặt mới:**

- Sử dụng các công nghệ tiên tiến như học sâu và mạng nơ-ron để cải thiện độ chính xác và hiệu quả của hệ thống.
- Phát triển các mô hình học máy có khả năng học từ dữ liệu lớn và đa dạng.

**Thu thập và xây dựng bộ dữ liệu khuôn mặt biểu cảm:**

- Thu thập một tập dữ liệu phong phú và đa dạng về các biểu cảm khuôn mặt từ nhiều nguồn khác nhau.
- Xây dựng bộ dữ liệu chuẩn để sử dụng trong quá trình huấn luyện và thử nghiệm hệ thống.

**Thử nghiệm và đánh giá hệ thống nhận diện cảm xúc khuôn mặt:**

- Thử nghiệm hệ thống trên nhiều tập dữ liệu khác nhau để đảm bảo tính tổng quát và độ chính xác cao.
- Đánh giá hiệu suất của hệ thống trong các điều kiện thực tế và trong các lĩnh vực ứng dụng khác nhau.

Với những nỗ lực của tôi, tôi hy vọng đề tài này sẽ góp phần phát triển lĩnh vực nhận diện cảm xúc khuôn mặt, mang lại những ứng dụng thiết thực và có ý nghĩa trong thực tế.

### **III. Phương pháp nghiên cứu**

Để đạt được các mục tiêu và nhiệm vụ đã nêu, tôi sẽ sử dụng các phương pháp nghiên cứu sau:

#### **Nghiên cứu lý thuyết:**

- Nghiên cứu các tài liệu khoa học, các bài báo nghiên cứu và các công trình nghiên cứu trước đây về nhận diện cảm xúc khuôn mặt.
- Xem xét các phương pháp và kỹ thuật nhận diện cảm xúc khuôn mặt đã được phát triển, cũng như các bộ dữ liệu khuôn mặt biểu cảm và các chỉ số đánh giá hiệu quả của hệ thống nhận diện cảm xúc khuôn mặt.

#### **Nghiên cứu thực nghiệm:**

- Thu thập và xây dựng bộ dữ liệu khuôn mặt biểu cảm bằng cách chụp ảnh khuôn mặt của các tình nguyện viên biểu lộ các cảm xúc cơ bản.
- Thử nghiệm và đánh giá các kỹ thuật nhận dạng cảm xúc khuôn mặt dựa trên bộ dữ liệu này.
- Sử dụng kết quả thực nghiệm để tối ưu hóa và cải tiến các kỹ thuật nhận dạng.

#### **Phương pháp suy luận:**

- Sử dụng các phương pháp suy luận logic để phát triển các kỹ thuật nhận dạng cảm xúc khuôn mặt mới.
- Giải quyết các thách thức hiện nay của lĩnh vực nhận diện cảm xúc khuôn mặt, chẳng hạn như độ phức tạp của khuôn mặt, điều kiện ánh sáng không thuận lợi và biểu cảm khuôn mặt tinh tế.

Cụ thể, các phương pháp nghiên cứu sẽ được sử dụng như sau:

### **Nghiên cứu lý thuyết:**

Tôi sẽ nghiên cứu sâu rộng các tài liệu khoa học liên quan đến nhận diện cảm xúc khuôn mặt, bao gồm việc tìm hiểu về các phương pháp nhận dạng cảm xúc khuôn mặt đã được phát triển, các bộ dữ liệu khuôn mặt biểu cảm đã được công bố, và các chỉ số đánh giá hiệu quả của hệ thống nhận diện cảm xúc khuôn mặt.

Qua đó, tôi sẽ tổng hợp các kiến thức và thông tin quan trọng để làm nền tảng cho việc phát triển các kỹ thuật nhận dạng mới.

### **Nghiên cứu thực nghiệm:**

Tôi sẽ tiến hành thu thập và xây dựng bộ dữ liệu khuôn mặt biểu cảm bằng cách chụp ảnh khuôn mặt của các tình nguyện viên khi họ biểu lộ các cảm xúc cơ bản như vui, buồn, giận, sợ và ngạc nhiên.

Bộ dữ liệu này sẽ được sử dụng để thử nghiệm và đánh giá các kỹ thuật nhận dạng cảm xúc khuôn mặt đã được phát triển.

Qua quá trình thử nghiệm và đánh giá, tôi sẽ thu thập các kết quả và phản hồi để tối ưu hóa và cải tiến các kỹ thuật nhận dạng.

### **Phương pháp suy luận:**

Tôi sẽ áp dụng các phương pháp suy luận logic để phát triển các kỹ thuật nhận dạng cảm xúc khuôn mặt mới, nhằm giải quyết các thách thức hiện nay của lĩnh vực nhận diện cảm xúc khuôn mặt.

Các kỹ thuật mới sẽ được thiết kế để cải thiện độ chính xác và hiệu quả của hệ thống nhận diện, đồng thời xử lý tốt hơn các yếu tố ảnh hưởng như độ phức tạp của khuôn mặt và điều kiện ánh sáng không thuận lợi.

Với việc sử dụng các phương pháp nghiên cứu trên, tôi hy vọng sẽ đạt được các mục tiêu và nhiệm vụ đã đề ra, góp phần vào sự phát triển của lĩnh vực nhận diện cảm xúc khuôn mặt và mang lại những ứng dụng thiết thực trong thực tế.

## NỘI DUNG

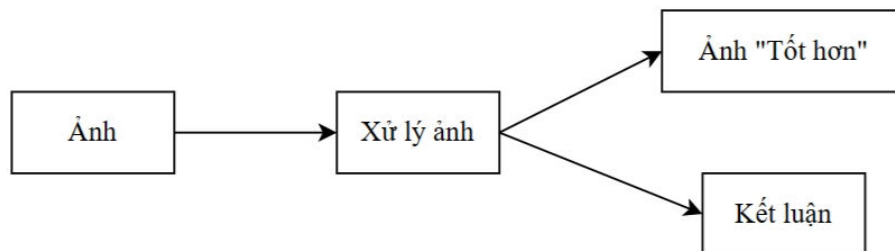
### CHƯƠNG 1 TỔNG QUAN VỀ XỬ LÝ ẢNH

#### 1.1 Xử lý ảnh, các vấn đề cơ bản trong xử lý ảnh

##### 1.1.1 Xử lý ảnh

Con người nhận thức thế giới qua các giác quan, trong đó thị giác giữ vai trò quan trọng nhất. Với sự phát triển vượt bậc của công nghệ phần cứng máy tính trong những năm gần đây, các ứng dụng xử lý hình ảnh và đồ họa đã không ngừng mở rộng và có nhiều ứng dụng thiết thực trong cuộc sống. Xử lý hình ảnh và đồ họa đóng vai trò then chốt trong giao tiếp giữa con người và máy móc.

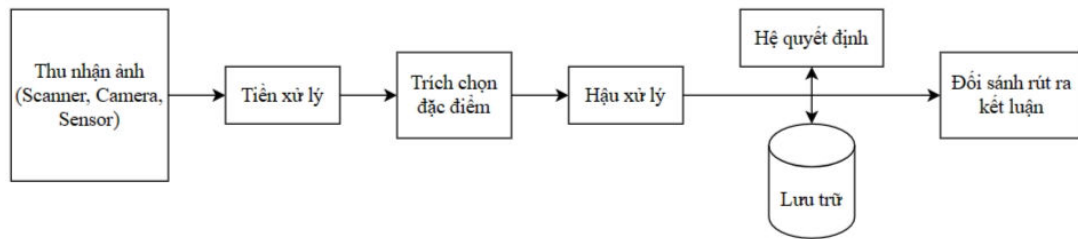
Quá trình xử lý hình ảnh được xem như việc thao tác trên ảnh đầu vào để đạt được kết quả như mong đợi. Kết quả đầu ra của quá trình này có thể là một hình ảnh "cải tiến" hoặc một kết luận cụ thể.



*Hình 1-1: Quá trình xử lý ảnh*

Một bức ảnh có thể được xem như một tập hợp các điểm ảnh, và mỗi điểm ảnh có thể được coi là một chỉ số về cường độ sáng hoặc một đặc trưng tại một vị trí cụ thể nào đó của đối tượng trong không gian. Nó có thể được coi như một hàm  $n$  biến  $P(c_1, c_2, \dots, c_n)$ . Vì vậy, trong xử lý ảnh, ảnh có thể được coi là ảnh  $n$  chiều.

Sơ đồ tổng quát của một hệ thống xử lý ảnh:



*Hình 1-2: Các bước cơ bản trong một hệ thống xử lý ảnh*

#### 1.1.1.1 Phần thu nhận ảnh (Image Acquisition)

Ảnh có thể được thu nhận từ camera màu hoặc đen trắng. Thông thường, ảnh từ camera là ảnh tương tự (được tạo ra bởi camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh gồm 25 dòng), hoặc có thể là ảnh số hóa (sử dụng công nghệ CCD – Charge Coupled Device), một loại photodiode tạo ra cường độ sáng cho từng điểm ảnh.

Camera thường sử dụng phương pháp quét dòng để tạo ra ảnh hai chiều. Chất lượng của một bức ảnh phụ thuộc vào thiết bị thu và các yếu tố môi trường như ánh sáng và cảnh quan.

#### 1.1.1.2 Tiền xử lý (Image Processing)

Sau khi được thu nhận, ảnh có thể gặp nhiễu và độ tương phản thấp, do đó cần được đưa vào bộ tiền xử lý để cải thiện chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu và tăng cường độ tương phản để làm cho ảnh trở nên rõ nét hơn.

Nhận dạng và nội suy ảnh (Image Recognition and Interpretation):

- **Nhận dạng ảnh** là quá trình xác định hình ảnh. Quá trình này thường được thực hiện bằng cách so sánh với các mẫu chuẩn đã được học hoặc lưu trữ từ trước.

- **Nội suy** là việc đưa ra các phán đoán dựa trên nhận dạng. Ví dụ: một dãy số và các đường gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại.

Có nhiều phương pháp phân loại hình ảnh khác nhau. Theo lý thuyết về nhận dạng, các mô hình toán học về hình ảnh được phân chia thành hai loại cơ bản:

- Nhận dạng theo tham số
- Nhận dạng theo cấu trúc

Một số đối tượng nhận dạng phổ biến hiện nay trong khoa học và công nghệ bao gồm: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản, nhận dạng vân tay, nhận dạng mã vạch, và nhận dạng khuôn mặt.

### 1.1.1.3 Cơ sở tri thức (Knowledge Base)

Như đã đề cập trước đó, hình ảnh là một đối tượng phức tạp với nhiều yếu tố như đường nét, độ sáng tối, dung lượng điểm ảnh và môi trường thu ảnh, dẫn đến việc xuất hiện nhiễu. Trong quá trình xử lý và phân tích ảnh, ngoài việc đơn giản hóa các phương pháp toán học để tiện lợi cho xử lý, người ta còn mong muốn bắt chước quy trình tiếp nhận và xử lý hình ảnh theo cách của con người.

Một trong những mục tiêu chính của xử lý hình ảnh là nâng cao chất lượng ảnh, làm cho nó rõ ràng và sắc nét hơn. Điều này bao gồm việc loại bỏ nhiễu và tăng cường độ tương phản. Các kỹ thuật xử lý ảnh này không chỉ áp dụng trong các lĩnh vực như y tế, quân sự, và an ninh mà còn phổ biến trong các ứng dụng hằng ngày như chỉnh sửa ảnh và video.

Hiện nay, nhiều bước trong quy trình xử lý đã được thực hiện theo các phương pháp trí tuệ nhân tạo. Nhờ vào sự phát triển của công nghệ AI, các hệ thống nhận dạng và xử lý hình ảnh ngày càng trở nên thông minh hơn, có khả năng tự học và cải thiện hiệu suất qua thời gian. Các thuật toán học sâu (deep learning)



và mạng neuron nhân tạo (neural networks) đã và đang tạo ra những bước tiến vượt bậc trong lĩnh vực này.

Các cơ sở tri thức đóng vai trò quan trọng trong việc mô hình hóa và xử lý hình ảnh. Bằng cách tích hợp những hiểu biết từ các lĩnh vực khác nhau như toán học, khoa học máy tính, và trí tuệ nhân tạo, chúng ta có thể xây dựng các hệ thống xử lý ảnh hiệu quả và chính xác hơn. Điều này không chỉ giúp cải thiện chất lượng hình ảnh mà còn mở ra nhiều ứng dụng mới, từ tự động hóa trong công nghiệp đến nhận diện khuôn mặt trong các hệ thống an ninh.

#### **1.1.1.4 Mô tả (biểu diễn ảnh)**

Sau khi số hoá, hình ảnh sẽ được lưu trữ vào bộ nhớ hoặc chuyển sang các giai đoạn xử lý tiếp theo để phân tích. Việc lưu trữ ảnh trực tiếp từ các ảnh thô đòi hỏi dung lượng bộ nhớ rất lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Do đó, thông thường các ảnh thô này được biểu diễn lại (hay còn gọi là mã hoá) theo các đặc điểm của ảnh, được gọi là các đặc trưng ảnh (Image Features), bao gồm biên ảnh (Boundary) và vùng ảnh (Region).

Các đặc trưng ảnh giúp đơn giản hóa và giảm dung lượng bộ nhớ cần thiết trong khi vẫn giữ lại những thông tin quan trọng nhất để phân tích và xử lý. Các biên ảnh là những đường nét phân định giữa các vùng khác nhau trong ảnh, và các vùng ảnh là những khu vực có các thuộc tính giống nhau. Nhờ vào các đặc trưng này, quá trình xử lý ảnh trở nên hiệu quả hơn, giảm thiểu tài nguyên cần thiết mà vẫn đảm bảo độ chính xác cao.

Việc mã hoá các đặc trưng ảnh cũng tạo điều kiện thuận lợi cho các thuật toán nhận dạng và phân loại ảnh, giúp hệ thống nhận biết và phân tích các đối tượng trong ảnh một cách chính xác và nhanh chóng hơn. Điều này có ý nghĩa quan trọng trong nhiều ứng dụng thực tiễn, từ y tế đến an ninh, từ sản xuất công nghiệp đến các hệ thống thị giác máy tính trong robot.

### 1.1.2 Các vấn đề cơ bản trong xử lý ảnh

#### 1.1.2.1 Một số khái niệm cơ bản

- **Ảnh và điểm ảnh**

Điểm ảnh được coi là một dấu hiệu hoặc cường độ sáng tại một tọa độ cụ thể trong không gian của đối tượng. Hình ảnh, do đó, là một tập hợp của các điểm ảnh này. Mỗi điểm ảnh đóng góp vào tổng thể của bức ảnh, với từng cường độ sáng và màu sắc tạo nên sự phân biệt và chi tiết của hình ảnh.

Thông qua quá trình xử lý hình ảnh, các điểm ảnh này được phân tích và điều chỉnh để cải thiện chất lượng tổng thể của bức ảnh. Chẳng hạn, việc tăng độ tương phản hoặc loại bỏ nhiễu có thể giúp các điểm ảnh rõ nét hơn, làm cho bức ảnh trở nên sắc sảo và dễ nhận diện hơn.

Điều này rất quan trọng trong nhiều ứng dụng, từ việc phân tích y tế, nơi mỗi điểm ảnh có thể chứa thông tin quan trọng về sức khỏe, đến các hệ thống nhận diện khuôn mặt, nơi từng điểm ảnh đóng vai trò trong việc xác định danh tính của một người.

- **Mức xám, màu**

Đây là các giá trị có thể có của các điểm ảnh trong một hình ảnh. Mức xám đề cập đến cường độ sáng trong ảnh đen trắng, trong khi màu bao gồm sự phối hợp của các giá trị cường độ sáng cho mỗi kênh màu (thường là đỏ, xanh lục và xanh lam) trong ảnh màu.

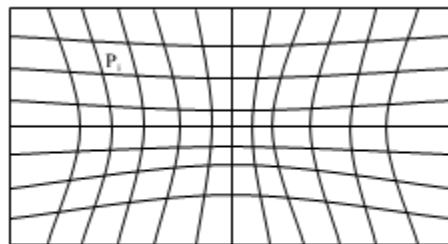
Mức xám thường được biểu diễn bằng các giá trị từ 0 (đen) đến 255 (trắng) trong hệ màu 8-bit, hoặc theo các mức độ khác trong các hệ màu có độ phân giải cao hơn. Ảnh màu, mặt khác, bao gồm nhiều tổ hợp của các giá trị cường độ cho các kênh màu, tạo ra một phổ rộng các sắc thái và màu sắc.

Việc hiểu và thao tác với các mức xám và màu là nền tảng của nhiều kỹ thuật xử lý hình ảnh, từ việc cải thiện chất lượng hình ảnh đến việc nhận dạng và phân loại đối tượng trong ảnh.

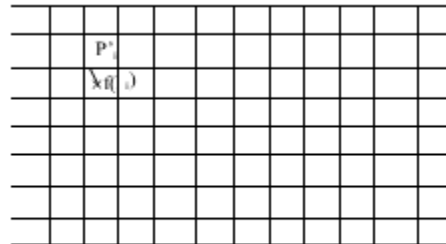
### 1.1.2.2 Nắn chỉnh biến dạng

Hình ảnh thu nhận thường bị biến dạng do các yếu tố từ thiết bị quang học và điện tử. Những biến dạng này có thể do nhiều nguyên nhân, chẳng hạn như sai lệch trong hệ thống ống kính, hiện tượng quang sai, hoặc nhiễu tín hiệu điện tử. Các thiết bị quang học không hoàn hảo có thể gây ra các vấn đề như làm mờ ảnh, méo hình hoặc tạo ra các điểm sáng không mong muốn. Tương tự, thiết bị điện tử có thể làm xuất hiện nhiễu hoặc làm giảm độ chính xác của tín hiệu hình ảnh.

Để khắc phục những vấn đề này, các kỹ thuật xử lý hình ảnh thường được áp dụng, bao gồm cả việc điều chỉnh và hiệu chỉnh hình ảnh sau khi thu nhận. Các phương pháp như lọc nhiễu, chỉnh sửa độ sáng, cân chỉnh màu sắc và sửa méo có thể được sử dụng để cải thiện chất lượng hình ảnh và giảm thiểu biến dạng.



**Ảnh thu nhận**



**Ảnh mong muốn**

*Hình 1-3: Ảnh thu nhận và ảnh mong muốn*

Để khắc phục các biến dạng trong hình ảnh, người ta sử dụng các phép chiếu. Các phép chiếu này thường được xây dựng dựa trên một tập các điểm điều khiển. Các điểm điều khiển này đóng vai trò quan trọng trong việc xác định cách thức hình ảnh được điều chỉnh và biến đổi để giảm thiểu méo mó và cải thiện chất lượng.

Phép chiếu trong xử lý hình ảnh có thể giúp điều chỉnh hình học của ảnh, làm cho các đối tượng trong ảnh khớp đúng với thực tế hơn. Điều này rất hữu ích trong các ứng dụng như bản đồ, kiến trúc và thiết kế đồ họa, nơi độ chính xác hình học là điều rất quan trọng. Các phép chiếu còn được sử dụng để tái cấu trúc ảnh từ các góc nhìn khác nhau, giúp cho việc phân tích và xử lý ảnh trở nên linh hoạt và hiệu quả hơn.

Ngoài ra, việc sử dụng các điểm điều khiển trong phép chiếu cũng giúp tối ưu hóa quá trình xử lý và tiết kiệm tài nguyên, bởi vì chỉ những điểm quan trọng được điều chỉnh, thay vì toàn bộ hình ảnh.

Giả sử  $(P_i, P'_i)$   $i = 1, n$  có  $n$  tập điều khiển.

Tìm hàm

$$f: P_i \mapsto f(P_i)$$

Sao cho

$$\sum_{i=1}^n \|f(P_i) - P'_i\|^2 \rightarrow \min$$

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm  $f$  có dạng:

$$f(x, y) = (a_1x + b_1y + c_1, a_2x + b_2y + c_2)$$

$$\begin{aligned} \phi = \sum_{i=1}^n (f(P_i) - P'_i)^2 &= \sum_{i=1}^n [(a_1x_i + b_1y_i + c_1 - x'_i)^2 \\ &\quad + (a_2x_i + b_2y_i + c_2 - y'_i)^2] \end{aligned}$$

Để  $\phi \rightarrow \min$

$$\begin{cases} \frac{\partial \phi}{\partial a_1} = 0 \\ \frac{\partial \phi}{\partial b_1} = 0 \\ \frac{\partial \phi}{\partial c_1} = 0 \end{cases} \Leftrightarrow \begin{cases} \sum_{i=1}^n a_i x_i^2 + \sum_{i=1}^n b_i x_i y_i + \sum_{i=1}^n c_i x_i = \sum_{i=1}^n x_i x'_i \\ \sum_{i=1}^n a_i x_i y_i + \sum_{i=1}^n b_i x_i y_i + \sum_{i=1}^n c_i x_i = \sum_{i=1}^n x_i x'_i \\ \sum_{i=1}^n a_i x_i^2 + \sum_{i=1}^n b_i x_i y_i + \sum_{i=1}^n c_i x_i = \sum_{i=1}^n x_i x'_i \end{cases}$$

Giải hệ phương trình tuyến tính tìm được  $a_1, b_1, c_1$

Tương tự tìm được  $a_2, b_2, c_2$

Xác định được hàm f

Khử nhiễu

Trong quá trình thu nhận ảnh, có hai loại nhiễu cơ bản:

- **Nhiều hệ thống:** Đây là loại nhiễu có quy luật, thường xuất hiện do các vấn đề liên quan đến thiết bị hoặc các yếu tố có hệ thống. Nhiễu này có thể được loại bỏ hoặc giảm thiểu bằng các phép biến đổi cụ thể. Ví dụ, việc sử dụng các kỹ thuật như biến đổi Fourier hoặc biến đổi wavelet có thể giúp khử nhiễu hệ thống hiệu quả.
- **Nhiều ngẫu nhiên:** Đây là loại nhiễu xuất hiện do các nguyên nhân không rõ ràng, chẳng hạn như các vết bẩn hoặc sự biến động ngẫu nhiên trong quá trình thu nhận ảnh. Để khắc phục loại nhiễu này, các phương pháp lọc thường được áp dụng. Các phép lọc như lọc trung bình, lọc trung vị, hoặc các kỹ thuật lọc phức tạp hơn có thể giúp làm sạch hình ảnh và giảm thiểu ảnh hưởng của nhiễu ngẫu nhiên.

Việc khử nhiễu là một bước quan trọng trong xử lý hình ảnh, giúp cải thiện chất lượng ảnh và làm cho các bước phân tích và nhận dạng sau đó trở nên chính xác hơn.

### 1.1.2.3 Chỉnh mức xám

Nhằm khắc phục tính không đồng đều do hệ thống gây ra, việc chỉnh mức xám là một bước quan trọng trong xử lý ảnh. Thông thường, có hai hướng tiếp cận chính:

- **Giảm số mức xám:** Phương pháp này thực hiện bằng cách nhóm các mức xám gần nhau thành một bó. Nếu chỉ còn hai mức xám, kết quả sẽ là ảnh đen trắng. Một ứng dụng phổ biến của kỹ thuật này là khi in ảnh màu trên máy in đen trắng. Kỹ thuật giảm số mức xám giúp tiết kiệm tài nguyên và đơn giản hóa quá trình xử lý ảnh khi chất lượng cao không phải là yêu cầu bắt buộc.
- **Tăng số mức xám:** Phương pháp này thực hiện bằng kỹ thuật nội suy để tạo ra các mức xám trung gian. Kỹ thuật nội suy này giúp tăng cường độ mịn và chi tiết của hình ảnh, làm cho ảnh trở nên mượt mà hơn. Điều này đặc biệt hữu ích trong các ứng dụng yêu cầu độ chính xác cao và hình ảnh chất lượng tốt, chẳng hạn như trong y học, truyền thông hình ảnh và thiết kế đồ họa.

Chỉnh mức xám không chỉ giúp cải thiện chất lượng hình ảnh mà còn giúp cân bằng và đồng nhất các giá trị cường độ sáng, từ đó làm cho quá trình phân tích và xử lý tiếp theo trở nên hiệu quả hơn.

### 1.1.2.4 Trích chọn đặc điểm

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Dưới đây là một số đặc điểm của ảnh:

**Đặc điểm không gian:** Bao gồm phân bố mức xám, phân bố xác suất, biên độ, và điểm uốn. Những đặc điểm này giúp xác định cấu trúc và bố cục không gian của hình ảnh.

**Đặc điểm biến đổi:** Các đặc điểm này được trích chọn thông qua việc thực hiện lọc vùng (zonal filtering). Các bộ lọc vùng, hay còn gọi là “mặt nạ đặc điểm” (feature mask), thường là các khe hẹp có hình dạng khác nhau như chữ nhật, tam giác, cung tròn, v.v. Chúng giúp xác định các biến đổi quan trọng trong hình ảnh.

**Đặc điểm biên và đường biên:** Đặc trưng cho đường biên của đối tượng và rất hữu ích trong việc trích chọn các thuộc tính bất biến được sử dụng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ các toán tử như toán tử gradient, toán tử la bàn, toán tử Laplace, và toán tử “chéo không” (zero crossing). Chúng giúp xác định các biên và đường viền trong hình ảnh, làm rõ ranh giới giữa các đối tượng.

Việc trích chọn hiệu quả các đặc điểm này giúp nhận dạng các đối tượng trong ảnh một cách chính xác, đồng thời tăng tốc độ tính toán và giảm dung lượng bộ nhớ lưu trữ. Điều này rất quan trọng trong các ứng dụng thực tiễn, từ nhận diện khuôn mặt đến phân loại hình ảnh trong y tế và công nghiệp.

#### 1.1.2.5 Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, và được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người tiên phong trong lĩnh vực này, đã định nghĩa rằng: “Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity) được xác định một cách mơ hồ (vaguely defined) và có thể gán cho nó một tên gọi nào đó”. Ví dụ, mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó, một chữ viết, khuôn mặt người, hoặc một ký đồ tín hiệu tiếng nói.

Để nhận dạng hoặc phân loại một mẫu cụ thể, có thể sử dụng hai phương pháp chính:

**Phân loại có mẫu (supervised classification):** Chẳng hạn như phân tích phân biệt (discriminant analysis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định trước.

**Phân loại không có mẫu (unsupervised classification hay clustering):** Trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này tại thời điểm phân loại vẫn chưa được biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chính:

1. Thu nhận dữ liệu và tiền xử lý.
2. Biểu diễn dữ liệu.
3. Nhận dạng, ra quyết định.

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng bao gồm:

1. Đối sánh mẫu dựa trên các đặc trưng được trích chọn.
2. Phân loại thống kê.
3. Đối sánh cấu trúc.
4. Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng thực tế, rõ ràng không thể chỉ sử dụng một cách tiếp cận đơn lẻ để đạt hiệu quả phân loại “tối ưu.” Do đó, việc kết hợp nhiều phương pháp và cách tiếp cận khác nhau là cần thiết. Các phương thức phân loại tổ hợp (ensemble methods) hay được sử dụng khi nhận dạng và đã đạt được những kết quả khả quan thông qua thiết kế các hệ thống lai (hybrid systems), bao gồm nhiều mô hình kết hợp.

Giải quyết bài toán nhận dạng trong các ứng dụng mới phát sinh từ cuộc sống không chỉ tạo ra thách thức về thuật toán mà còn đặt ra yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả các ứng dụng này là những đặc điểm đặc trưng



cần thiết thường rất nhiều, không thể do chuyên gia đề xuất mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

#### 1.1.2.6 Nén ảnh

Nhằm giảm thiểu không gian lưu trữ, việc nén ảnh thường được thực hiện theo hai khuynh hướng: nén có bảo toàn thông tin và nén không bảo toàn thông tin. Nén không bảo toàn thường có khả năng nén cao hơn nhưng khả năng phục hồi dữ liệu gốc thì kém hơn. Dựa trên hai khuynh hướng này, có bốn cách tiếp cận cơ bản trong nén ảnh:

- **Nén ảnh thống kê:** Kỹ thuật nén này dựa vào việc thống kê tần suất xuất hiện của các giá trị điểm ảnh, từ đó áp dụng chiến lược mã hóa thích hợp. Một ví dụ điển hình cho kỹ thuật mã hóa này là định dạng \*.TIF.
- **Nén ảnh không gian:** Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa, lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này là mã nén \*.PCX.
- **Nén ảnh sử dụng phép biến đổi:** Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn, và thường hiệu quả hơn. Định dạng \*.JPG là ví dụ điển hình cho kỹ thuật nén này, sử dụng các phép biến đổi như DCT (Discrete Cosine Transform) để nén ảnh hiệu quả.
- **Nén ảnh Fractal:** Kỹ thuật này sử dụng tính chất Fractal của các đối tượng ảnh, thể hiện sự lặp lại của các chi tiết. Quá trình nén sẽ tính toán để chỉ cần lưu trữ phần gốc của ảnh và quy luật sinh ra ảnh theo nguyên lý Fractal, giúp nén ảnh một cách tối ưu mà vẫn giữ được các chi tiết quan trọng.

### 1.2 Công nghệ nhận dạng khuôn mặt

#### 1.2.1 Nhận dạng khuôn mặt là gì?

Nhận dạng khuôn mặt là một công nghệ sử dụng phần mềm để xác định hoặc xác nhận danh tính của một người dựa trên đặc điểm khuôn mặt của họ. Công

nghe này hoạt động bằng cách xác định và đo lường các đặc điểm khuôn mặt từ hình ảnh hoặc video.

Công nghệ nhận dạng khuôn mặt có thể thực hiện các chức năng chính sau:

**Xác định khuôn mặt người trong hình ảnh hoặc video:** Phát hiện sự hiện diện của khuôn mặt người trong các hình ảnh hoặc video, thường được sử dụng trong các ứng dụng giám sát và bảo mật.

**So sánh khuôn mặt trong hai hình ảnh:** Xác định xem khuôn mặt xuất hiện trong hai hình ảnh có phải là cùng một người hay không. Điều này hữu ích trong các hệ thống xác thực và kiểm tra danh tính.

**Tìm kiếm khuôn mặt trong bộ sưu tập hình ảnh:** Tìm kiếm và so sánh khuôn mặt trong một cơ sở dữ liệu lớn các hình ảnh hiện có. Điều này được áp dụng trong các ứng dụng an ninh và quản lý dữ liệu.

Các hệ thống bảo mật sinh trắc học sử dụng công nghệ nhận dạng khuôn mặt để nhận dạng cá nhân duy nhất trong các quy trình đăng nhập hoặc triển khai người dùng, cũng như để tăng cường việc xác thực người dùng. Các thiết bị di động và cá nhân cũng thường sử dụng công nghệ này để bảo mật thiết bị, chẳng hạn như mở khóa điện thoại bằng khuôn mặt.

Nhận dạng khuôn mặt không chỉ là một công cụ bảo mật mạnh mẽ mà còn có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ quản lý nhân sự đến tiếp thị và trải nghiệm người dùng cá nhân hóa.



*Hình 1-4: Công nghệ nhận dạng khuôn mặt*

## **1.2.2 Một số lợi ích của hệ thống nhận dạng khuôn mặt**

### **1.2.2.1 Bảo mật hiệu quả**

Nhận dạng khuôn mặt là hệ thống xác minh nhanh chóng và hiệu quả. Công nghệ này nhanh và tiện lợi hơn so với những công nghệ sinh trắc học khác như quét vân tay hoặc võng mạc. Số lượng điểm tiếp xúc trong nhận dạng khuôn mặt cũng ít hơn so với việc nhập mật khẩu hoặc PIN, làm cho quá trình xác thực trở nên nhanh chóng và dễ dàng hơn. Công nghệ này hỗ trợ xác thực nhiều yếu tố để cung cấp thêm lớp bảo mật bổ sung, giúp nâng cao mức độ an toàn của hệ thống.

### **1.2.2.2 Cải thiện độ chính xác**

Nhận dạng khuôn mặt là phương thức xác định cá nhân chính xác hơn so với việc chỉ sử dụng số điện thoại, địa chỉ email, địa chỉ gửi thư hoặc địa chỉ IP. Ví dụ, hiện nay, hầu hết các dịch vụ giao dịch từ cổ phiếu cho đến tiền điện tử đều tín nhiệm công nghệ nhận dạng khuôn mặt để bảo vệ khách hàng và tài sản của họ. Nhờ công nghệ này, việc xác thực người dùng trở nên chính xác hơn, giảm thiểu nguy cơ gian lận và lạm dụng.

### **1.2.2.3 Tích hợp dễ dàng hơn**

Công nghệ nhận dạng khuôn mặt tương thích và tích hợp dễ dàng với hầu hết các phần mềm bảo mật hiện có. Ví dụ, điện thoại thông minh với camera trước có thể tích hợp các thuật toán hoặc mã phần mềm nhận dạng khuôn mặt, giúp người dùng dễ dàng sử dụng tính năng này để bảo mật thiết bị. Việc tích hợp này không chỉ nâng cao mức độ bảo mật mà còn tăng cường trải nghiệm người dùng, làm cho việc sử dụng các thiết bị và hệ thống trở nên thuận tiện hơn.

Công nghệ nhận dạng khuôn mặt không chỉ mang lại nhiều lợi ích trong việc bảo mật mà còn cải thiện độ chính xác và hiệu quả trong nhiều ứng dụng khác nhau, từ quản lý an ninh đến các dịch vụ tài chính và cá nhân.

### **1.3Giới thiệu về nhận dạng cảm xúc qua khuôn mặt**

Khuôn mặt của con người biểu hiện nhiều cảm xúc mà không cần phải nói ra, là một trong những phương tiện mạnh mẽ và tự nhiên nhất để con người truyền đạt cảm xúc. Không giống như các hình thức giao tiếp phi ngôn ngữ khác, cảm xúc trên khuôn mặt mang tính phổ quát. Hiện nay, nhận dạng và phân tích cảm xúc khuôn mặt tự động là một vấn đề thú vị và đầy thách thức, có ảnh hưởng to lớn đến xã hội.

Cảm xúc trên khuôn mặt và hành động của chúng ta là phương tiện giao tiếp phi ngôn ngữ, chiếm 93% cảm xúc giao tiếp của con người, trong đó 55% thể hiện qua cử chỉ khuôn mặt và hành động. Cảm xúc khuôn mặt có thể được phân tích dễ dàng thông qua hình ảnh khuôn mặt, cho phép máy tính tương tác với con người một cách tự nhiên như cách con người tương tác với nhau. Đó là lý do tại sao nhận dạng cảm xúc qua khuôn mặt ngày càng được quan tâm trong mọi lĩnh vực.

Các nhà nghiên cứu đã chỉ ra rằng cảm xúc trên khuôn mặt là phổ quát và bẩm sinh trong tất cả các chủng tộc, giới tính và độ tuổi. Bao gồm cả cảm xúc trung tính, có bảy cảm xúc cơ bản là: trung tính, giận dữ, ghê tởm, sợ hãi, hạnh

phức, buồn và bất ngờ. Nhận dạng cảm xúc qua khuôn mặt có nhiều ứng dụng trong các lĩnh vực khác nhau:

**Giáo dục:** Phản ứng của người học trong thời gian thực và sự tham gia vào nội dung giáo dục là thước đo hiệu quả của bài giảng.

**Tiếp thị:** Đây là cách tuyệt vời để các công ty kinh doanh phân tích cách khách hàng phản hồi với quảng cáo, sản phẩm, bao bì và thiết kế cửa hàng của họ.

**Chơi game:** Với sự ra đời của game thực tế ảo gần với trải nghiệm thực tế, nhận dạng cảm xúc khuôn mặt đóng vai trò quan trọng trong việc cải thiện trải nghiệm chơi trò chơi.

**Bảo mật:** Công nghệ này giúp xác định hành vi đáng ngờ trong đám đông và có thể được sử dụng để ngăn chặn tội phạm và những kẻ khủng bố tiềm năng.

**Chăm sóc sức khỏe:** Công nghệ này hữu ích trong việc tự động hóa dịch vụ y tế. Cả sức khỏe thể chất và tinh thần có thể được phân tích thông qua ứng dụng này.

**Dịch vụ khách hàng:** Quản lý dịch vụ khách hàng có thể hiệu quả hơn nhờ hệ thống nhận dạng cảm xúc khuôn mặt. Phân tích phản hồi của khách hàng và phản ứng của máy tính sẽ đảm bảo tương tác thực tế giữa máy tính và con người.

Hệ thống nhận diện cảm xúc khuôn mặt được sử dụng nhiều trong cuộc sống, bao gồm điều trị y tế, giao tiếp phi ngôn ngữ, đánh giá đau của bệnh nhân, phát hiện nói dối, và giám sát trạng thái của người lái xe để phát hiện dấu hiệu buồn ngủ, mệt mỏi, giúp cảnh báo và đảm bảo an toàn.

## **CHƯƠNG 2      CÁC PHƯƠNG PHÁP PHÁT HIỆN KHUÔN MẶT**

### **2.1Giới Thiệu**

Trong chương này, tôi sẽ tập trung giới thiệu và phân tích các phương pháp phát hiện khuôn mặt hiện đại, nhằm xác định những tiến bộ và thách thức trong lĩnh vực này. Việc hiểu rõ về các kỹ thuật này sẽ là cơ sở quan trọng cho việc xây dựng hệ thống nhận biết cảm xúc dựa trên biểu cảm khuôn mặt.

Nhận diện khuôn mặt hiện là một trong những ứng dụng gần gũi nhất của Thị giác máy tính. Công nghệ này được tích hợp trên đa thiết bị, đa nền tảng, đa lĩnh vực, đem đến cho người dùng cuối những trải nghiệm nhanh chóng, tiện lợi và vẫn đảm bảo an toàn khi tiến hành xác thực trên các hệ thống.

### **2.2Phương Pháp Dựa Trên Mạng Nơ-ron**

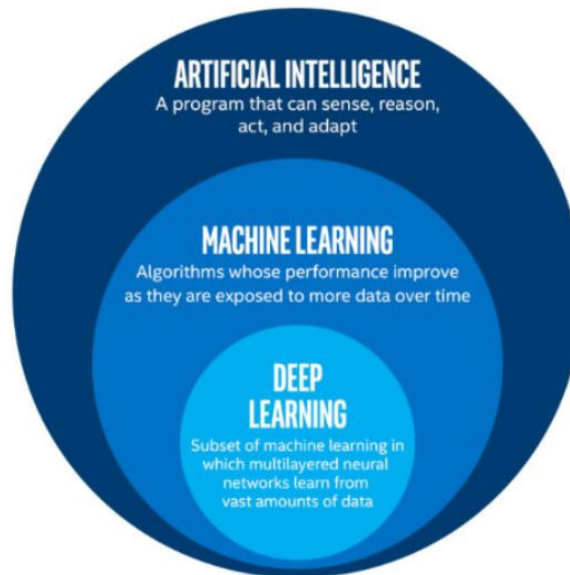
Sự phát triển mạnh mẽ của mạng nơ-ron sâu đã tạo ra nhiều cơ hội mới trong lĩnh vực phát hiện khuôn mặt. Trong phần này, tôi sẽ trình bày về việc tích hợp các mô hình Convolutional Neural Network (CNN) và các biến thể của nó như Single Shot MultiBox Detector (SSD) và Region-based CNN (R-CNN) để cải thiện độ chính xác trong quá trình nhận diện.

#### **2.2.1 Mạng thần kinh (Neural Networks)**

##### **2.2.1.1 Giới thiệu chung về deep learning**

Deep learning là một nhánh chuyên sâu của Machine learning. Trong khoảng thập kỷ vừa qua, lĩnh vực này đã chứng kiến sự phát triển vượt bậc, mở ra nhiều ứng dụng thực tế đáng kinh ngạc. Deep learning không chỉ thay đổi cách chúng ta xử lý dữ liệu mà còn tạo ra nhiều giải pháp tiên tiến trong các lĩnh vực như y tế, tài chính, vận tải và nhiều ngành công nghiệp khác. Những tiến bộ này đã tạo ra một cuộc cách mạng trong khả năng phân tích và xử lý thông tin, mang lại lợi ích to lớn cho người dùng và các doanh nghiệp trên toàn thế giới. Với khả năng học hỏi từ lượng dữ liệu khổng lồ, Deep learning đã và đang giúp đẩy nhanh

quá trình chuyển đổi số và tạo ra những giá trị mới trong thời đại công nghệ hiện nay.



*Hình 2-1: AI vs. ML vs. DL*

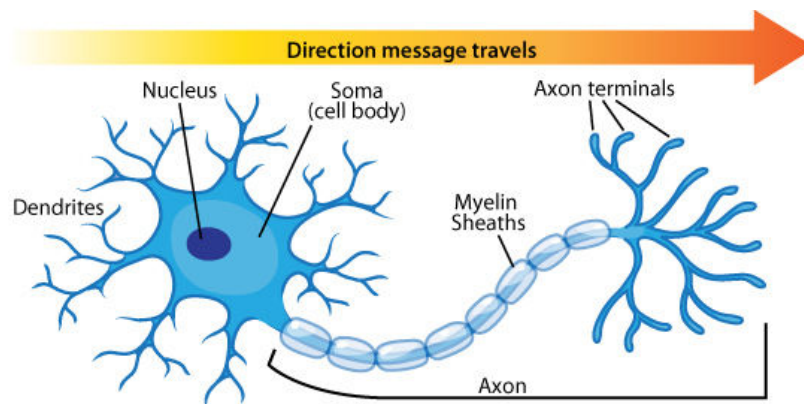
Hai lý do chính dẫn đến sự phát triển vượt bậc của Deep learning trong những năm trở lại đây là:

- Sự bùng nổ của dữ liệu: Ngày càng có nhiều dữ liệu được sinh ra từ nhiều nguồn khác nhau, bao gồm dữ liệu hình ảnh, văn bản, âm thanh và video. Các mô hình deep learning tận dụng rất tốt sự phong phú và đa dạng của nguồn dữ liệu này, giúp cải thiện độ chính xác và hiệu quả trong việc học tập và phân tích.
- Sự phát triển của phần cứng máy tính: Song hành với sự gia tăng của dữ liệu là sự tiến bộ vượt bậc trong công nghệ phần cứng máy tính. Các hệ thống phần cứng hiện đại với sức mạnh xử lý mạnh mẽ và các GPU chuyên dụng cho việc huấn luyện mô hình deep learning đã giúp đẩy nhanh quá trình huấn luyện hàng tỷ dữ liệu. Điều này không chỉ giúp rút ngắn thời gian

đào tạo mô hình mà còn nâng cao khả năng học tập và phân tích của các mô hình này.

### 2.2.1.2 Kiến trúc mạng nơ ron - Neural network

Neural network, hay còn gọi là mạng nơ ron, được lấy cảm hứng từ cấu trúc mạng nơ ron sinh học trong bộ não con người. Đây là một hệ thống mạng gồm rất nhiều nơ ron nhân tạo được kết nối với nhau theo một trật tự nhất định, nhằm mục đích mô phỏng cách thức hoạt động và xử lý thông tin của bộ não. Mỗi nơ ron trong mạng có vai trò truyền tải và xử lý thông tin, góp phần vào việc học hỏi, phân tích và ra quyết định của hệ thống. Các nơ ron được sắp xếp thành các lớp khác nhau, bao gồm lớp đầu vào, lớp ẩn và lớp đầu ra, với mỗi lớp đảm nhiệm những chức năng riêng biệt nhưng liên kết chặt chẽ với nhau. Với khả năng tự học và cải thiện qua từng lần huấn luyện, neural network đã và đang trở thành công cụ quan trọng trong nhiều lĩnh vực như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và nhiều ứng dụng khác.



**Hình 2-2: Nơ-ron**

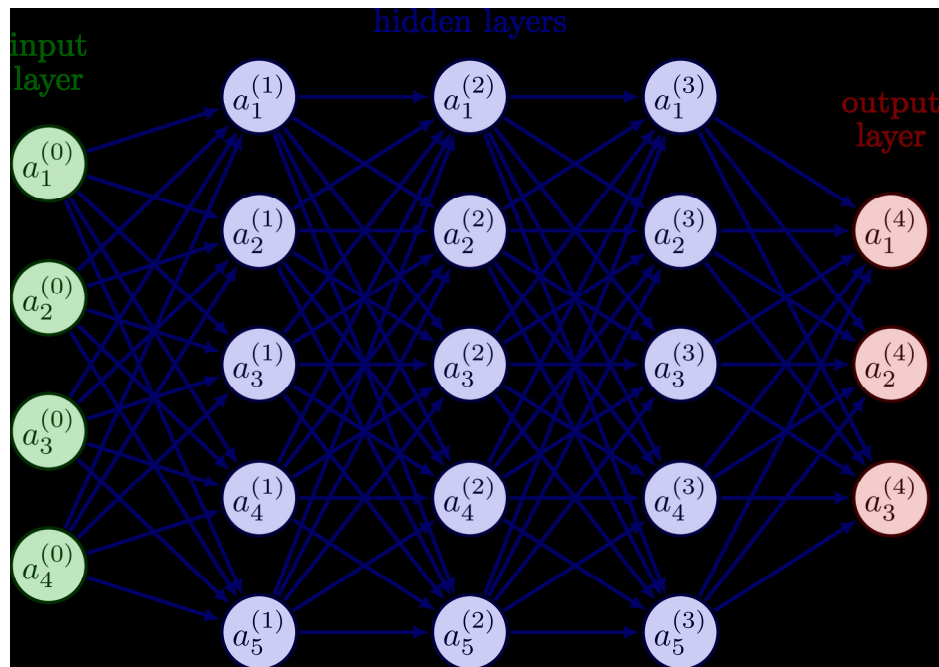
Nơ-ron là đơn vị cơ bản cấu tạo hệ thống thần kinh và là một phần quan trọng nhất của não. Trong não người, có khoảng 10 triệu nơ-ron, và mỗi nơ-ron kết nối với 10.000 nơ-ron khác. Mỗi nơ-ron bao gồm phần thân (soma) chứa nhân, các tín hiệu đầu vào qua sợi nhánh (dendrites) và các tín hiệu đầu ra qua sợi trục (axon) kết nối với các nơ-ron khác. Hiểu đơn giản, mỗi nơ-ron nhận dữ liệu đầu



vào qua sợi nhánh và truyền dữ liệu đầu ra qua sợi trục, đến các sợi nhánh của các nơ-ron khác. Mỗi nơ-ron nhận xung điện từ các nơ-ron khác qua sợi nhánh, và nếu xung điện đủ lớn để kích hoạt nơ-ron, tín hiệu sẽ đi qua sợi trục đến các sợi nhánh của các nơ-ron khác.

Ở mỗi nơ-ron cần quyết định có kích hoạt nơ-ron đấy hay không. Tuy nhiên, trong deep learning, chúng ta chỉ lấy cảm hứng từ não bộ và cách nó hoạt động, chứ không phải bắt chước toàn bộ các chức năng của nó. Mục tiêu chính là sử dụng các mô hình này để giải quyết những bài toán mà chúng ta đặt ra.

Trong neural network, các nơ-ron nhân tạo được kết nối với nhau tạo thành các lớp (layers). Lớp nhận đầu vào được gọi là input layer, các lớp tính toán ở giữa được gọi là hidden layers, và lớp trả đầu ra được gọi là output layer. Các lớp này hoạt động phối hợp với nhau để học hỏi, xử lý thông tin và đưa ra kết quả.



**Hình 2-3: Mô hình tổng quát Neural Network**

Trong neural network, có rất nhiều loại layer khác nhau, mỗi loại có chức năng riêng biệt. Trong nội dung bài này, chúng ta sẽ nghiên cứu về một số loại

layer cơ bản nhất. Các layer này đóng vai trò quan trọng trong việc xây dựng và huấn luyện mô hình neural network.

Layer đầu tiên được gọi là input layer, nơi dữ liệu đầu vào được đưa vào mạng. Tiếp theo là các hidden layers, đây là các lớp tính toán ở giữa giúp xử lý và phân tích thông tin từ input layer. Cuối cùng là output layer, nơi kết quả của quá trình học tập và phân tích được xuất ra. Một mạng neural network bắt buộc phải có input layer và output layer, trong khi các hidden layers có thể có hoặc không, tùy thuộc vào yêu cầu của bài toán.

Các hình tròn trong các layer được gọi là nodes, hay còn gọi là neurons. Những nodes này liên kết với tất cả các nodes ở các layer trước đó thông qua các trọng số riêng (weights). Mỗi node đều có một hệ số bias riêng và sau đó sẽ tính tổng các trọng số nhân với dữ liệu đầu vào từ các layer trước, rồi áp dụng một hàm kích hoạt (activation function) để truyền tín hiệu tới các nodes của layer tiếp theo.

Việc hiểu rõ về các loại layer và chức năng của chúng là rất quan trọng để có thể thiết kế và huấn luyện một neural network hiệu quả. Các mô hình deep learning sử dụng cấu trúc này để giải quyết nhiều bài toán phức tạp, từ nhận diện hình ảnh đến xử lý ngôn ngữ tự nhiên.

### **2.2.1.3 Linear layer**

Linear layer là lớp đơn giản nhất, nhưng là nền tảng quan trọng để xây dựng các neural network. Cụ thể, linear layer thực hiện phép biến đổi tuyến tính (linear transformation) thông qua phép toán nhân ma trận. Trong quá trình này, các đầu vào của mỗi node được nhân với trọng số tương ứng và cộng thêm hệ số bias, từ đó tạo ra đầu ra của node. Linear layer đóng vai trò quan trọng trong việc tạo ra cấu trúc cơ bản của neural network, giúp mô hình học được các mối quan hệ tuyến tính trong dữ liệu. Mặc dù đơn giản, nhưng linear layer là phần không thể thiếu trong các mạng deep learning phức tạp hơn, vì nó cung cấp nền tảng cho các phép biến đổi phức tạp và việc học các mẫu trong dữ liệu.

$$y = WX$$

Trong đó:

- $X$  là ma trận đầu vào của linear layer
- $W$  là trọng số của neural network tại linear layer đó
- $y$  là ma trận đầu ra của linear layer, kết quả của phép biến đổi tuyến tính

#### 2.2.1.4 Một số hàm kích hoạt (Activation layer)

Chúng ta đã cùng nhau nghiên cứu về một số các logistic activation layer như Sigmoid, Softmax và Tanh. Bên cạnh các layer này, còn có rất nhiều các activation layer khác, mỗi loại có những đặc điểm và ứng dụng riêng biệt. Các activation layer được đặt xen kẽ giữa các linear layer, giúp cho quá trình tính toán của các linear layer trở nên ý nghĩa và hiệu quả hơn. Điều này có nghĩa là, nếu không có các activation layer nằm giữa các linear layer, việc chồng nhiều linear layer lên nhau cũng không khác gì so với việc chỉ sử dụng một linear layer duy nhất. Activation layer giúp mô hình có khả năng học các mối quan hệ phi tuyến tính trong dữ liệu, làm tăng tính phức tạp và khả năng biểu diễn của mô hình. Một số activation layer phổ biến khác bao gồm ReLU (Rectified Linear Unit), Leaky ReLU, và ELU (Exponential Linear Unit), mỗi loại có những ưu điểm và ứng dụng riêng trong việc giải quyết các bài toán khác nhau trong lĩnh vực deep learning.

##### 2.2.1.4.1 ReLU

Khác với các activation layer như Sigmoid hay Softmax, thường được đặt ở layer cuối cùng trong neural network để chuyển đổi đầu ra thành xác suất, ReLU (Rectified Linear Unit) là một activation layer phổ biến thường được đặt giữa các linear layer trong mạng neural network. ReLU có ưu điểm là đơn giản và hiệu quả, giúp mô hình học được các đặc điểm phức tạp từ dữ liệu bằng cách kích hoạt các nơ-ron chỉ khi đầu vào của chúng dương. Điều này giúp giảm thiểu vấn đề

vanishing gradient, một vấn đề thường gặp khi sử dụng các activation layer như Sigmoid và Tanh.

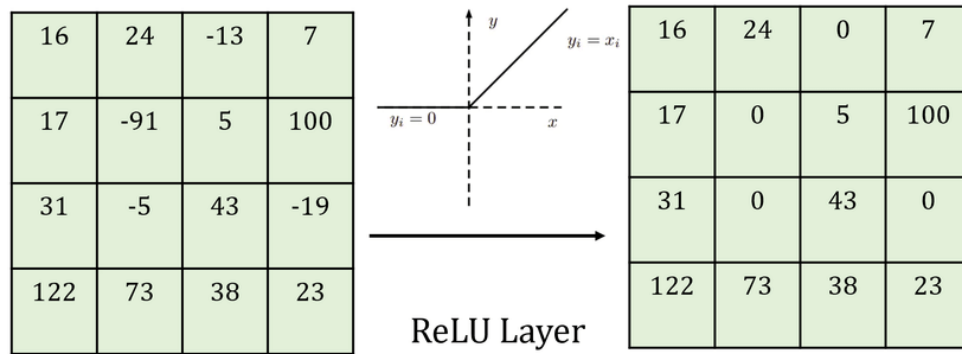
ReLU hoạt động bằng cách gán giá trị 0 cho tất cả các đầu vào âm và giữ nguyên các giá trị đầu vào dương. Vì tính chất này, ReLU giúp tăng tốc độ huấn luyện mô hình và cải thiện hiệu suất tổng thể. Ngoài ReLU, còn có các biến thể khác như Leaky ReLU và Parametric ReLU (PReLU), mỗi loại có những điều chỉnh nhất định để khắc phục một số hạn chế của ReLU trong các trường hợp cụ thể.

$$y = \max(0, x)$$

Trong đó:

- $x$  là giá trị đầu vào của hàm ReLU
- $y$  là giá trị đầu ra của hàm ReLU

Khi sử dụng hàm ReLU (Rectified Linear Unit) cho một vector hoặc ma trận, chúng ta áp dụng hàm ReLU cho từng phần tử trên vector hoặc ma trận đó. Điều này có nghĩa là mỗi phần tử sẽ được tính toán riêng lẻ, và hàm ReLU sẽ gán giá trị 0 cho bất kỳ phần tử nào có giá trị âm, trong khi giữ nguyên giá trị của các phần tử dương. Quá trình này giúp loại bỏ các giá trị âm, làm tăng tính phi tuyến của mô hình và hỗ trợ việc học các đặc điểm phức tạp từ dữ liệu. Việc áp dụng ReLU theo từng phần tử giúp tăng tốc độ tính toán và tối ưu hóa quá trình huấn luyện của neural network, làm cho mô hình trở nên hiệu quả hơn trong việc xử lý các bài toán thực tế.



Hình 2-4: ReLU Layer

#### 2.2.1.4.2 Leaky ReLU

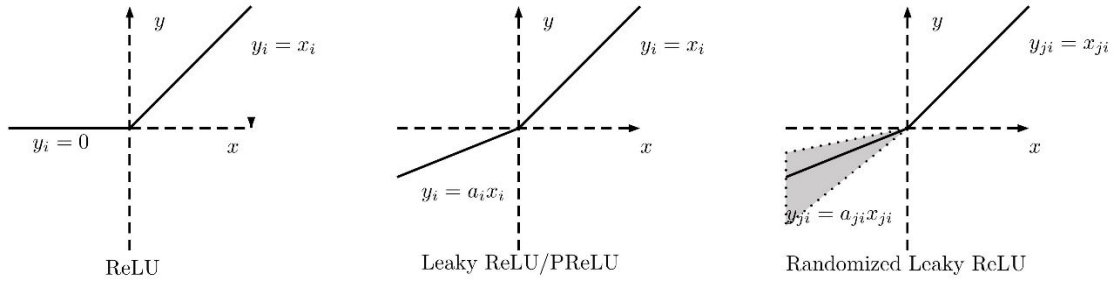
Một vấn đề khi sử dụng hàm ReLU (Rectified Linear Unit) là hiện tượng vanishing gradient tại những vị trí có giá trị nhỏ hơn hoặc bằng 0. Điều này xảy ra khi gradient của các nơ-ron trong mạng bị tiêu biến, làm cho việc huấn luyện mô hình trở nên kém hiệu quả. Để khắc phục vấn đề này, Leaky ReLU đã được đề xuất như một giải pháp cải tiến.

Leaky ReLU là một biến thể của ReLU, trong đó các giá trị âm không bị gán hoàn toàn bằng 0 mà được nhân với một hệ số nhỏ, thường là một giá trị rất nhỏ như 0.01. Điều này giúp duy trì gradient ngay cả khi đầu vào âm, giúp tránh hiện tượng vanishing gradient và cải thiện quá trình huấn luyện mô hình. Leaky ReLU đã chứng minh hiệu quả trong nhiều trường hợp, giúp tăng tốc độ huấn luyện và độ chính xác của các neural network.

$$y = \max(\gamma x, x)$$

Trong đó

- $x$  là giá trị đầu vào của hàm ReLU
- $\gamma$  là giá trị rất nhỏ, thường được lựa chọn là 0.1
- $y$  là giá trị đầu ra của hàm ReLU



**Hình 2-5: Các biến thể của ReLU**

### 2.2.1.5 Chuẩn hóa (Normalization layer)

Đối với các neural network phức tạp và có kích thước mô hình lớn, quá trình huấn luyện thường gặp nhiều khó khăn và trở nên bất ổn định. Sự bất ổn định này có thể dẫn đến việc neural network có kết quả huấn luyện rất kém, gây khó khăn trong việc triển khai các ứng dụng thực tiễn.

Sự ra đời của các normalization layer, cụ thể là Batch normalization layer, đã giúp cải thiện đáng kể tình trạng này. Normalization layer nói chung có vai trò chuẩn hoá đầu ra của mỗi layer trong neural network, từ đó giúp ổn định hoá quá trình huấn luyện. Batch normalization, một trong những kỹ thuật phổ biến nhất, thực hiện việc chuẩn hoá bằng cách điều chỉnh và duy trì sự ổn định của các giá trị đầu ra qua các batch dữ liệu huấn luyện.

Việc sử dụng Batch normalization không chỉ giúp giảm thiểu sự thay đổi không mong muốn trong quá trình huấn luyện mà còn tăng tốc độ huấn luyện và cải thiện hiệu suất của mô hình. Bằng cách giảm thiểu sự thay đổi trong việc phân phối các đầu ra của mỗi layer, normalization layer giúp mô hình học tập một cách hiệu quả hơn, từ đó đạt được kết quả tốt hơn và ổn định hơn.

<b>Input:</b> Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\};$	
Parameters to be learned: $\gamma, \beta$	
<b>Output:</b> $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$	
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$	// mini-batch mean
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$	// mini-batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$	// normalize
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i)$	// scale and shift

**Algorithm 1:** Batch Normalizing Transform, applied to activation  $x$  over a mini-batch.

*Hình 2-6: Các phương pháp chuẩn hóa*

Ngoài Batch normalization layer, còn có một số loại normalization layer khác như Layer normalization layer, Instance normalization layer và Group normalization layer. Mỗi loại này có những đặc điểm và ứng dụng riêng biệt:

**Layer normalization layer:** Loại normalization này áp dụng chuẩn hoá lên tất cả các nơ-ron trong một lớp (layer) dựa trên thông số thống kê của chính lớp đó. Điều này giúp ổn định quá trình huấn luyện cho từng lớp một cách độc lập và thường được sử dụng trong các mạng có cấu trúc tuần tự như Recurrent Neural Network (RNN).

**Instance normalization layer:** Được áp dụng cho từng mẫu dữ liệu (instance) trong batch, loại normalization này thường được sử dụng trong các bài toán xử lý hình ảnh và style transfer, giúp duy trì các đặc điểm cụ thể của từng ảnh trong quá trình huấn luyện.

**Group normalization layer:** Chia nhỏ các nơ-ron trong một lớp thành các nhóm và thực hiện chuẩn hoá trên từng nhóm. Điều này giúp tối ưu hoá và ổn định

quá trình huấn luyện, đặc biệt hiệu quả trong các mô hình có kích thước batch nhỏ, nơi Batch normalization không thể thực hiện tốt.

Các normalization layer này đóng vai trò quan trọng trong việc cải thiện hiệu suất và độ ổn định của neural network, giúp mô hình học nhanh hơn và đạt được kết quả tốt hơn.

#### **2.2.1.6 Dropout layer**

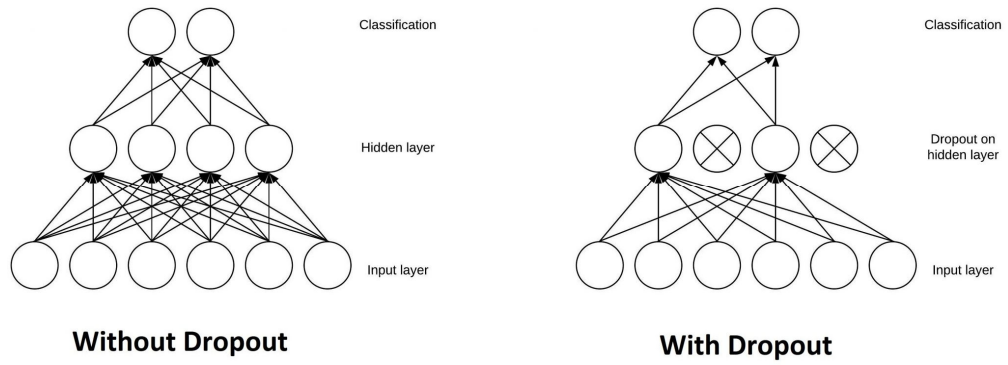
Chúng ta đã thảo luận về hiện tượng overfit trong các mô hình machine learning, và trong neural network, hiện tượng này cũng là một vấn đề nan giải. Để giảm bớt vấn đề này, Dropout layer đã được giới thiệu như một giải pháp hiệu quả.

Dropout layer hoạt động bằng cách ngẫu nhiên chọn một số các nơ-ron trong neural network và gán giá trị của chúng bằng 0 trong quá trình huấn luyện. Cụ thể hơn, Dropout layer ngẫu nhiên lựa chọn một số các trọng số của mô hình và gán giá trị bằng 0. Bằng cách này, mạng neural network sẽ giảm được độ phức tạp, từ đó giảm thiểu hiện tượng overfit.

Việc loại bỏ ngẫu nhiên các nơ-ron trong quá trình huấn luyện giúp mô hình không phụ thuộc quá nhiều vào bất kỳ nơ-ron nào, tăng cường khả năng tổng quát hóa của mô hình. Điều này cải thiện hiệu suất của mạng neural network khi áp dụng vào dữ liệu mới mà nó chưa từng thấy trước đây, giúp giảm thiểu sự chênh lệch giữa kết quả huấn luyện và kết quả thử nghiệm.

Dropout layer đã trở thành một kỹ thuật phổ biến và hiệu quả trong việc xây dựng các neural network mạnh mẽ và đáng tin cậy, đặc biệt là khi xử lý các bài toán phức tạp và yêu cầu khả năng tổng quát hóa cao.





Hình 2-7: Minh họa về Drop Layer

## 2.2.2 Hàm mất mát và Tối ưu hóa (Loss function và Optimization)

### 2.2.2.1 Hàm mất mát

Đối với bài toán Regression, chúng ta có một số hàm loss phổ biến như sau:

- Mean Absolute Error (MAE): MAE tính toán tổng các giá trị tuyệt đối của sai số giữa giá trị dự đoán và giá trị thực tế, sau đó lấy giá trị trung bình. MAE ít nhạy cảm với các giá trị ngoại lai hơn so với MSE.

$$MAE(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

- Mean Squared Error (MSE): Đây là hàm loss được sử dụng phổ biến nhất trong các bài toán hồi quy. MSE tính toán tổng bình phương sai số giữa giá trị dự đoán và giá trị thực tế, sau đó lấy giá trị trung bình. Hàm này nhạy cảm với các giá trị ngoại lai (outliers).

$$MSE(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

- Huber Loss: Huber Loss là sự kết hợp giữa MSE và MAE, giúp giảm thiểu ảnh hưởng của các giá trị ngoại lai. Huber Loss hoạt động như MAE khi sai số nhỏ và như MSE khi sai số lớn.

- **Log-Cosh Loss:** Log-Cosh Loss sử dụng hàm log của cosh (hyperbolic cosine) để tính toán sai số. Hàm này kết hợp các đặc tính của cả MSE và MAE, giúp ổn định quá trình huấn luyện và ít nhạy cảm với các giá trị ngoại lai.
- **Quantile Loss:** Quantile Loss thường được sử dụng trong các bài toán dự đoán khoảng tin cậy (confidence intervals). Hàm này tính toán sai số dựa trên phân vị (quantiles) của giá trị dự đoán và giá trị thực tế.

Đối với bài toán Classification, chúng ta có một số hàm loss phổ biến như sau:

- **Cross-Entropy Loss:** Đây là hàm loss phổ biến nhất trong các bài toán phân loại. Nó đo lường sự khác biệt giữa phân phối dự đoán của mô hình và phân phối thực tế của nhãn. Cross-Entropy Loss thường được sử dụng cho các bài toán phân loại nhị phân và phân loại đa lớp.
- **Hinge Loss:** Hinge Loss thường được sử dụng trong các mô hình SVM (Support Vector Machines). Hàm này thích hợp cho các bài toán phân loại nhị phân và giúp tối đa hóa margin giữa các lớp.
- **Kullback-Leibler Divergence (KL Divergence):** KL Divergence đo lường sự khác biệt giữa hai phân phối xác suất. Hàm này thường được sử dụng trong các mô hình phân loại và các bài toán học không giám sát.
- **Focal Loss:** Focal Loss là một biến thể của Cross-Entropy Loss, được thiết kế để xử lý các bài toán có sự mất cân bằng lớp lớn. Hàm này giảm trọng số của các mẫu dễ phân loại và tăng trọng số của các mẫu khó phân loại.
- **Binary Cross-Entropy Loss:** Đây là một phiên bản đặc biệt của Cross-Entropy Loss, được sử dụng trong các bài toán phân loại nhị phân. Binary Cross-Entropy Loss đo lường độ lệch giữa giá trị dự đoán và giá trị thực tế cho mỗi lớp.

### 2.2.2.2 Optimization

Chúng ta đã cùng nhau nghiên cứu về khá nhiều các thuật toán tối ưu hóa khác nhau biến thể từ Gradient Descent. Một số phương pháp phổ biến bao gồm:

**Stochastic Gradient Descent (SGD):** Phương pháp này cập nhật trọng số của mô hình dựa trên một mẫu ngẫu nhiên từ dữ liệu huấn luyện thay vì toàn bộ tập dữ liệu, giúp giảm thời gian tính toán và tăng tốc độ hội tụ.

**Momentum:** Thuật toán này cải tiến SGD bằng cách sử dụng một động lượng để tăng tốc quá trình tối ưu hóa, giúp mô hình vượt qua các thung lũng của hàm mất mát một cách nhanh chóng hơn.

**Nesterov Accelerated Gradient (NAG):** NAG là một biến thể của Momentum, giúp tăng tốc quá trình tối ưu hóa bằng cách điều chỉnh hướng di chuyển của gradient dựa trên thông tin về động lượng, giúp cải thiện hiệu suất và độ chính xác của mô hình.

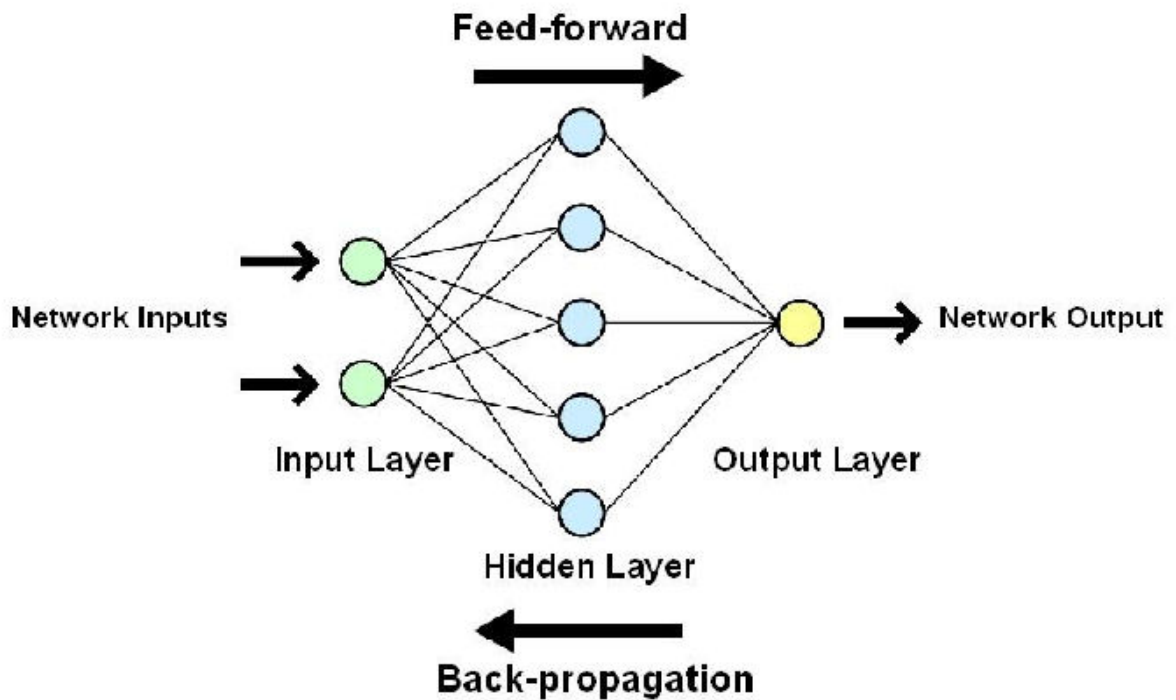
Ngoài ra, còn có các thuật toán tối ưu hóa khác như:

**Adaptive Gradient Algorithm (AdaGrad):** Phương pháp này điều chỉnh tốc độ học tập cho từng tham số dựa trên lịch sử gradient, giúp mô hình học nhanh hơn khi gradient lớn và chậm lại khi gradient nhỏ.

**Root Mean Square Propagation (RMSProp):** RMSProp cải thiện AdaGrad bằng cách giới hạn sự tăng tốc độ học tập, giúp tránh tình trạng tốc độ học tập giảm quá nhanh và tăng hiệu suất huấn luyện mô hình.

**Adaptive Moment Estimation (Adam):** Adam kết hợp các ý tưởng của Momentum và RMSProp, điều chỉnh tốc độ học tập dựa trên trung bình của gradient và trung bình bình phương của gradient, giúp mô hình hội tụ nhanh hơn và ổn định hơn.

### 2.2.3 Model feedforward và backpropagation



*Hình 2-8: Model feedforward và backpropagation*

Feedforward là quá trình neural network nhận đầu vào từ input layer, thực hiện các phép tính toán qua các linear layer, activation layer và normalization layer để trả đầu ra ở output layer. Quá trình này giúp mạng neural network xử lý và chuyển đổi dữ liệu từ đầu vào thành đầu ra thông qua một loạt các phép biến đổi và tính toán.

Ngược lại với feedforward, sau khi đưa kết quả dự đoán của neural network vào hàm loss và tính toán giá trị loss, Backpropagation là quá trình tính toán giá trị đạo hàm của hàm loss theo từng trọng số của neural network. Cụ thể, backpropagation giúp xác định cách thay đổi các trọng số để giảm thiểu giá trị loss. Sau khi tính được giá trị gradient tương ứng với mỗi trọng số, neural network thực hiện cập nhật lại các trọng số này theo thuật toán tối ưu dựa trên gradient descent. Việc này giúp cải thiện hiệu suất của mô hình bằng cách điều chỉnh các trọng số theo hướng giảm thiểu hàm loss, từ đó nâng cao độ chính xác của dự đoán.

## 2.3 Phương Pháp Kết Hợp

Để đạt được hiệu suất tối ưu, nhiều phương pháp đã được kết hợp và tối ưu hóa một cách hiệu quả. Trong phần này, tôi sẽ trình bày về những nghiên cứu tiên tiến nhất trong việc kết hợp giữa các phương pháp dựa trên đặc trưng (feature-based methods) và mạng nơ-ron (neural networks). Các nghiên cứu này đã chứng minh rằng sự kết hợp giữa hai phương pháp này có thể mang lại hiệu quả vượt trội so với việc sử dụng một phương pháp duy nhất. Chúng ta sẽ cùng xem xét các cách tiếp cận đa dạng trong việc kết hợp các đặc trưng cụ thể của dữ liệu với khả năng học sâu của mạng nơ-ron để nâng cao hiệu suất mô hình. Đồng thời, chúng ta cũng sẽ đánh giá hiệu suất của các phương pháp này qua nhiều tiêu chí khác nhau, từ độ chính xác, tốc độ huấn luyện cho đến khả năng tổng quát hóa trên dữ liệu mới.

## 2.4 Phương Pháp Dựa Trên Đặc Trưng

Phương pháp này tập trung vào việc xác định các đặc trưng quan trọng trên khuôn mặt, như điểm mắt, mũi, và miệng, để từ đó xác định vị trí của khuôn mặt trong ảnh. Tôi sẽ trình bày chi tiết về cách các thuật toán như Viola-Jones và Histogram of Oriented Gradients (HOG) được áp dụng, cũng như ưu và nhược điểm của từng phương pháp.

### **Thuật toán Viola-Jones:**

**Áp dụng:** Thuật toán Viola-Jones là một trong những phương pháp phát hiện khuôn mặt sớm nhất và được sử dụng rộng rãi. Nó sử dụng đặc trưng Haar-like để phát hiện các vùng quan trọng trên khuôn mặt như mắt, mũi và miệng. Viola-Jones dựa trên việc huấn luyện một cây phân loại đơn giản, sử dụng các đặc trưng Haar-like, để nhận diện khuôn mặt.

**Ưu điểm:**

- Tốc độ xử lý nhanh, phù hợp cho các ứng dụng thời gian thực.
- Được tích hợp trong nhiều thư viện phát hiện khuôn mặt như OpenCV

**Nhược điểm:**

- Hiệu suất kém trong điều kiện ánh sáng thay đổi và góc nhìn khác nhau.
- Độ chính xác không cao khi phát hiện các khuôn mặt nhỏ hoặc bị che khuất một phần.

**Thuật toán Histogram of Oriented Gradients (HOG):**

**Áp dụng:** HOG là một phương pháp phát hiện đặc trưng hình dạng của đối tượng thông qua việc tính toán các gradient hướng. Đối với phát hiện khuôn mặt, HOG chia ảnh thành các ô nhỏ, tính toán hướng của các gradient trong từng ô và sử dụng biểu đồ hướng này để mô tả các đặc trưng của khuôn mặt.

**Ưu điểm:**

- Độ chính xác cao trong việc phát hiện các đặc trưng hình dạng.
- Tương đối ổn định với sự thay đổi ánh sáng và góc nhìn.

**Nhược điểm:**

- Tốc độ xử lý chậm hơn so với Viola-Jones.
- Cần sự điều chỉnh cẩn thận để đạt được hiệu suất tốt nhất.

## CHƯƠNG 3 THỰC NGHIỆM

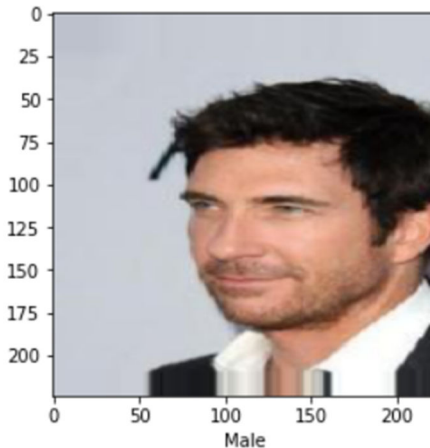
### 3.1 Cơ sở dữ liệu

#### 3.1.1 Bộ dataset Giới tính (Gender)

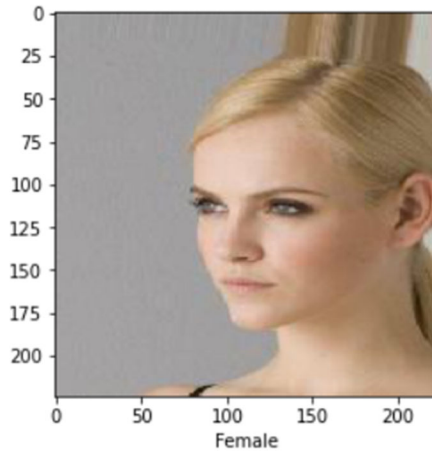
Link: <https://www.kaggle.com/datasets/ashishjangra27/gender-recognition-200k-images-celeba/data>

Tập dữ liệu này được sử dụng để phân loại giới tính dựa trên hình ảnh. Bộ dữ liệu bao gồm gần 200,000 hình ảnh với tổng dung lượng khoảng 1,3 GB, được chia thành ba tập dữ liệu chính: Train (Huấn luyện), Test (Kiểm tra), và Validation (Xác thực). Mỗi tập dữ liệu này gồm hai tập con (label) được chia thành hai thư mục riêng biệt là Female (Giới tính nữ) và Male (Giới tính nam).

Tập dữ liệu này rất hữu ích cho các nghiên cứu và phát triển các mô hình học sâu (deep learning) trong lĩnh vực nhận dạng giới tính, giúp các nhà khoa học dữ liệu và các nhà nghiên cứu có thể kiểm tra và cải thiện các thuật toán của mình.



*Hình 3-1: Hình ảnh minh họa Male*



*Hình 3-2: Hình ảnh minh họa Female*

### 3.1.2 Bộ dataset Cảm xúc (Emotion)

Link: <https://www.kaggle.com/datasets/msambare/fer2013>

Dữ liệu bao gồm các hình ảnh khuôn mặt với thang độ xám 48x48 pixel. Các khuôn mặt trong ảnh đã được đăng ký tự động sao cho khuôn mặt ít nhiều được căn giữa và chiếm cùng một khoảng không gian trong mỗi hình ảnh.

Nhiệm vụ của tập dữ liệu này là phân loại từng khuôn mặt dựa trên cảm xúc thể hiện trên nét mặt. Cảm xúc được phân loại thành một trong bảy loại:

(0=Tức giận, 1=Ghê tởm, 2=Sợ hãi, 3=Vui vẻ, 4=Buồn, 5=Bất ngờ, 6=Bình thường) .

Tập huấn luyện bao gồm 28.709 ví dụ, cung cấp một khối lượng dữ liệu phong phú để huấn luyện các mô hình học sâu. Tập kiểm tra công khai bao gồm 3.589 ví dụ, được sử dụng để đánh giá hiệu suất của các mô hình đã được huấn luyện.

Bộ dữ liệu FER-2013 rất hữu ích cho các nghiên cứu và phát triển các mô hình nhận diện cảm xúc từ hình ảnh khuôn mặt, giúp các nhà khoa học dữ liệu và các nhà nghiên cứu có thể kiểm tra và cải thiện các thuật toán của mình.





*Hình 3-3: Một số hình ảnh minh họa bộ dataset FER-2013*

### 3.2 Triển khai thực nghiệm

#### 3.2.1 Triển khai xây dựng mô hình training với tập dữ liệu Cảm xúc

##### 3.2.1.1 Import các thư viện cần thiết

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Dropout, BatchNormalization
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.callbacks import EarlyStopping
```

import tensorflow as tf: Nhập thư viện TensorFlow, một khung công tác mã nguồn mở mạnh mẽ dành cho học máy và học sâu.

`from tensorflow.keras.preprocessing.imageimport ImageDataGenerator:` Nhập lớp `ImageDataGenerator` từ Keras để tạo và xử lý dữ liệu hình ảnh theo thời gian thực, bao gồm các phép biến đổi và tăng cường dữ liệu.

`from tensorflow.keras.modelsimport Sequential:` Nhập lớp `Sequential` từ Keras để tạo mô hình tuần tự, đây là một dạng mô hình gồm các lớp được xếp nối tiếp nhau.

`from tensorflow.keras.layersimport Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization:`

`Conv2D`: Lớp tích chập 2D dùng để trích xuất đặc trưng từ ảnh đầu vào.

`MaxPooling2D`: Lớp gộp tối đa 2D dùng để giảm kích thước không gian của đặc trưng, giảm số lượng tham số và tính toán trong mạng.

`Flatten`: Lớp dẹt hóa, chuyển đổi các tensor 2D thành một vector 1D, thường được sử dụng sau các lớp tích chập và gộp.

`Dense`: Lớp kết nối đầy đủ, thường được sử dụng ở cuối mô hình để thực hiện phân loại hoặc hồi quy.

`Dropout`: Lớp bỏ ngẫu nhiên các đơn vị mạng, giúp ngăn chặn overfitting bằng cách tạm thời loại bỏ các đơn vị trong quá trình huấn luyện.

`BatchNormalization`: Lớp chuẩn hóa theo batch, giúp cải thiện tốc độ huấn luyện và ổn định của mạng.

`from tensorflow.keras.optimizersimport RMSprop:` Nhập bộ tối ưu hóa `RMSprop` từ Keras, một thuật toán tối ưu hóa giúp điều chỉnh trọng số của mô hình để giảm thiểu hàm mất mát.

`from tensorflow.keras.callbacksimport EarlyStopping:` Nhập callback `EarlyStopping` từ Keras, dùng để dừng quá trình huấn luyện sớm nếu mô hình không còn cải thiện hiệu suất trên tập kiểm tra.

### 3.2.1.2 Tạo bộ tăng cường dữ liệu

```
train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

`rescale=1.0/255`: Tái tỷ lệ các giá trị pixel của ảnh về khoảng từ 0 đến 1 bằng cách chia các giá trị cho 255. Điều này là cần thiết vì các giá trị pixel mặc định nằm trong khoảng từ 0 đến 255.

`rotation_range=20`: Xoay ngẫu nhiên ảnh trong phạm vi từ -20 đến 20 độ, giúp tạo ra các biến thể của ảnh ban đầu và giảm thiểu hiện tượng overfitting.

`width_shift_range=0.2`: Dịch chuyển ngẫu nhiên ảnh theo chiều ngang trong khoảng 20% của chiều rộng ảnh, giúp tạo ra các biến thể mới và cải thiện khả năng học của mô hình.

`height_shift_range=0.2`: Dịch chuyển ngẫu nhiên ảnh theo chiều dọc trong khoảng 20% của chiều cao ảnh, tương tự như `width_shift_range` nhưng theo chiều dọc.

`shear_range=0.2`: Áp dụng biến dạng xé cho ảnh trong khoảng 20%, giúp tạo ra các biến thể khác nhau của ảnh ban đầu.

`zoom_range=0.2`: Phóng to hoặc thu nhỏ ngẫu nhiên ảnh trong khoảng từ 80% đến 120% kích thước ban đầu, giúp mô hình học cách nhận diện các đối tượng ở các kích thước khác nhau.

`horizontal_flip=True`: Lật ngẫu nhiên ảnh theo chiều ngang, giúp tăng cường tập dữ liệu bằng cách tạo ra các phiên bản phản chiếu của ảnh ban đầu.

`fill_mode='nearest'`: Khi áp dụng các biến đổi như dịch chuyển, xoay hay xé, các giá trị pixel mới được thêm vào sẽ được điền bằng các giá trị pixel gần nhất, giúp duy trì tính liên tục của ảnh.

`ImageDataGenerator` giúp tạo và xử lý dữ liệu hình ảnh theo thời gian thực, tăng cường dữ liệu bằng cách tạo ra các biến thể ngẫu nhiên của ảnh ban đầu, từ đó cải thiện khả năng học của mô hình học sâu.

### 3.2.1.3 Áp dụng data augmentation cho tập huấn luyện

```
train_generator = train_datagen.flow_from_directory(
    directory=r"/kaggle/input/fer2013/train",
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical'
)
```

`directory=r"/kaggle/input/fer2013/train"`: Chỉ định đường dẫn đến thư mục chứa tập dữ liệu huấn luyện. Trong trường hợp này, nó là đường dẫn đến thư mục "train" trong kho lưu trữ dữ liệu Kaggle "fer2013".

`target_size=(150, 150)`: Chuyển đổi kích thước của tất cả các hình ảnh trong thư mục về kích thước (150, 150) pixels trước khi đưa vào mô hình.

`batch_size=32`: Chỉ định kích thước lô (batch size), nghĩa là số lượng mẫu hình ảnh được xử lý cùng một lúc. Ở đây, lô có kích thước 32.

`class_mode='categorical'`: Chỉ định chế độ phân loại, trong trường hợp này là 'categorical', nghĩa là các nhãn được mã hóa dưới dạng phân loại (one-hot encoding).

Hàm `train_datagen.flow_from_directory` tạo ra một bộ sinh dữ liệu từ các hình ảnh trong thư mục chỉ định, với các thuộc tính cụ thể như chuyển đổi kích thước, kích thước lô, và chế độ phân loại. Bộ sinh dữ liệu này sẽ tự động đọc ảnh từ thư mục, áp dụng các biến đổi và tăng cường đã chỉ định, và tạo ra các lô dữ liệu để huấn luyện mô hình học máy.

- Tạo bộ tăng cường dữ liệu cho tập validation (không thay đổi dữ liệu)

```
validation_datagen = ImageDataGenerator(rescale=1.0/255)
```

ImageDataGenerator: Đây là một lớp trong Keras được sử dụng để tạo và xử lý dữ liệu hình ảnh theo thời gian thực.

rescale=1.0/255: Tham số này dùng để tái tỷ lệ các giá trị pixel của ảnh về khoảng từ 0 đến 1 bằng cách chia các giá trị cho 255. Điều này là cần thiết vì các giá trị pixel mặc định nằm trong khoảng từ 0 đến 255.

Đoạn mã này tạo ra một đối tượng ImageDataGenerator cho tập dữ liệu xác thực (validation data). Đối tượng này sẽ tự động áp dụng việc tái tỷ lệ cho tất cả các ảnh trong tập dữ liệu xác thực trước khi chúng được đưa vào mô hình để đánh giá.

- Áp dụng data augmentation cho tập validation

```
validation_generator = validation_datagen.flow_from_directory(
    directory=r"/kaggle/input/fer2013/test",
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical'
)
```

directory=r"/kaggle/input/fer2013/test": Chỉ định đường dẫn đến thư mục chứa tập dữ liệu kiểm tra. Trong trường hợp này, nó là đường dẫn đến thư mục "test" trong kho lưu trữ dữ liệu Kaggle "fer2013".

target\_size=(150, 150): Chuyển đổi kích thước của tất cả các hình ảnh trong thư mục về kích thước (150, 150) pixels trước khi đưa vào mô hình.

batch\_size=32: Chỉ định kích thước lô (batch size), nghĩa là số lượng mẫu hình ảnh được xử lý cùng một lúc. Ở đây, lô có kích thước 32.

class\_mode='categorical': Chỉ định chế độ phân loại, trong trường hợp này là 'categorical', nghĩa là các nhãn được mã hóa dưới dạng phân loại (one-hot encoding).

Hàm `validation_datagen.flow_from_directory` tạo ra một bộ sinh dữ liệu từ các hình ảnh trong thư mục chỉ định, với các thuộc tính cụ thể như chuyển đổi kích thước, kích thước lô, và chế độ phân loại. Bộ sinh dữ liệu này sẽ tự động đọc ảnh từ thư mục, áp dụng việc tái tỷ lệ và tạo ra các lô dữ liệu để kiểm tra mô hình học máy.

### 3.2.1.4 Khởi tạo mô hình CNN

Khởi tạo một mô hình Sequential, trong đó các lớp sẽ được xếp chồng lên nhau theo thứ tự.

```
# Khởi tạo mô hình CNN
model = Sequential()

# Các Lớp Convolutional
model.add(Conv2D(16, (3, 3), activation='relu', input_shape=(150,
150, 3)))
model.add(MaxPooling2D(2, 2))
```

`Conv2D(16, (3, 3), activation='relu', input_shape=(150, 150, 3))`: Thêm một lớp tích chập với 16 bộ lọc kích thước (3, 3), sử dụng hàm kích hoạt ReLU, và đầu vào là các ảnh kích thước (150, 150, 3) (chiều rộng, chiều cao, và số kênh màu).

`MaxPooling2D(2, 2)`: Thêm một lớp gộp tối đa với kích thước lưới (2, 2) để giảm kích thước không gian của đặc trưng.

Tương tự, các lớp tích chập và gộp tối đa tiếp theo được thêm vào để trích xuất và giảm kích thước đặc trưng từ ảnh đầu vào:

```
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
```

```
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
```

Chuyển đổi các đặc trưng 2D thành một vector 1D để chuẩn bị cho các lớp kết nối đầy đủ.

```
# Flatten
model.add(Flatten())

# Fully-connected layers
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
```

Dense(128, activation='relu'): Thêm một lớp kết nối đầy đủ với 128 đơn vị và hàm kích hoạt ReLU.

Dropout(0.5): Thêm lớp bỏ ngẫu nhiên 50% các đơn vị, giúp ngăn chặn overfitting.

```
# Output Layer
model.add(Dense(7, activation='softmax'))
```

Thêm một lớp kết nối đầy đủ với 7 đơn vị đầu ra (tương ứng với 7 lớp cảm xúc) và sử dụng hàm kích hoạt Softmax để phân loại đa lớp.

### 3.2.1.5 Sử dụng EarlyStopping để dừng quá trình huấn luyện

```
# Compile mô hình
model.compile(optimizer=RMSprop(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
# Sử dụng EarlyStopping để dừng quá trình huấn luyện sớm nếu không có cải tiến đáng kể
early_stopping = EarlyStopping(patience=3, monitor='val_loss',
                                restore_best_weights=True)
```

optimizer=RMSprop(learning\_rate=0.001): Sử dụng bộ tối ưu hóa RMSprop với tốc độ học là 0.001. RMSprop là một thuật toán tối ưu hóa giúp điều chỉnh trọng số của mô hình để giảm thiểu hàm mất mát.

`loss='categorical_crossentropy'`: Sử dụng hàm mất mát crossentropy cho các bài toán phân loại đa lớp. Hàm này đo lường sự khác biệt giữa các phân phối xác suất dự đoán và thực tế.

`metrics=['accuracy']`: Sử dụng độ chính xác làm chỉ số đánh giá cho quá trình huấn luyện và kiểm tra mô hình.

`patience=3`: Chỉ định số epoch cần chờ trước khi dừng quá trình huấn luyện nếu không có cải tiến đáng kể. Ở đây, `patience` được đặt là 3.

`monitor='val_loss'`: Theo dõi hàm mất mát trên tập xác thực (validation loss) để quyết định dừng huấn luyện sớm.

`restore_best_weights=True`: Khôi phục trọng số tốt nhất của mô hình khi quá trình huấn luyện kết thúc. Điều này giúp đảm bảo mô hình đạt được hiệu suất tốt nhất có thể.

Đoạn mã này chuẩn bị mô hình để huấn luyện bằng cách xác định bộ tối ưu hóa, hàm mất mát và chỉ số đánh giá. Ngoài ra, nó còn thiết lập cơ chế EarlyStopping để dừng quá trình huấn luyện sớm nếu mô hình không có cải tiến đáng kể, giúp tránh overfitting và tối ưu hóa hiệu suất mô hình.

### 3.2.1.6 Huấn luyện mô hình

*# Huấn Luyện mô hình*

```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // train_generator.batch_size,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=validation_generator.n //
validation_generator.batch_size,
    callbacks=[early_stopping]
)
```

`train_generator`: Bộ sinh dữ liệu huấn luyện, cung cấp các lô dữ liệu huấn luyện theo thời gian thực.



`steps_per_epoch=train_generator.n // train_generator.batch_size`: Số bước mỗi epoch được xác định bằng tổng số mẫu trong tập huấn luyện chia cho kích thước lô. Đây là số lần mô hình sẽ cập nhật trọng số trong mỗi epoch.

`epochs=20`: Số lần toàn bộ dữ liệu huấn luyện sẽ được đưa qua mô hình. Trong trường hợp này, mô hình sẽ huấn luyện trong 20 epoch.

`validation_data=validation_generator`: Bộ sinh dữ liệu xác thực, cung cấp các lô dữ liệu xác thực theo thời gian thực.

`validation_steps=validation_generator.n//validation_generator.batch_size`: Số bước mỗi epoch cho tập xác thực được xác định bằng tổng số mẫu trong tập xác thực chia cho kích thước lô.

`callbacks=[early_stopping]`: Sử dụng callback EarlyStopping đã được định nghĩa trước đó để dừng quá trình huấn luyện sớm nếu không có cải tiến đáng kể.

Đoạn mã này khởi động quá trình huấn luyện mô hình bằng cách sử dụng dữ liệu huấn luyện và xác thực từ các bộ sinh dữ liệu đã định nghĩa. Mô hình sẽ huấn luyện trong 20 epoch (hoặc ít hơn nếu EarlyStopping kích hoạt) và lưu lại thông tin về quá trình huấn luyện vào biến history để có thể theo dõi sau này.

### 3.2.1.7 Đánh giá độ chính xác mô hình

```
# Đánh giá độ chính xác của mô hình
score = model.evaluate(train_generator,verbose=0)
print('Sai số kiểm tra là: ',score[0])
print('Độ chính xác kiểm tra là: ',score[1])
```

### 3.2.1.8 Kết quả training model

Sai số kiểm tra là: 1.350556492805481

Độ chính xác kiểm tra là: 0.4932251274585724

*Hình 3-4: Kết quả training emotion-model*

### 3.2.2 Triển khai xây dựng mô hình training với tập dữ liệu giới tính

Xây dựng tương tự với emotion-model

Sai số kiểm tra là: 0.195728600025177

Độ chính xác kiểm tra là: 0.9189985394477844

*Hình 3-5: Kết quả training gender-model*

### 3.2.3 Triển khai giao diện cơ bản cho ứng dụng nhận diện khuôn mặt

#### 3.2.3.1 Import các thư viện cần thiết

```
import cv2
import os
import sys
import warnings
import logging
import queue
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.utils import set_random_seed
import numpy as np
import cvlib
import tkinter as tk
from tkinter import *
from PIL import ImageTk, Image
from tkinter import messagebox
import threading
```

import cv2: Nhập thư viện OpenCV, dùng cho các tác vụ thị giác máy tính như xử lý ảnh, quay video và phân tích.

import os: Cung cấp cách tương tác với hệ điều hành, như đọc hoặc ghi tệp, và quản lý thư mục.

import sys: Cung cấp quyền truy cập vào các tham số và hàm đặc thù của hệ thống, bao gồm tương tác với trình thông dịch Python.

import warnings: Cho phép hiển thị các thông báo cảnh báo để nhận biết các vấn đề tiềm ẩn.

`import logging`: Cho phép ghi lại các thông báo để theo dõi sự kiện xảy ra trong quá trình thực thi chương trình, hữu ích cho việc gỡ lỗi và giám sát.

`import queue`: Cung cấp cấu trúc dữ liệu hàng đợi cho lập trình đa luồng, cho phép trao đổi dữ liệu an toàn giữa các luồng.

`from tensorflow.keras.models import load_model`: Nhập hàm để tải các mô hình đã huấn luyện sẵn trong Keras API của TensorFlow.

`from tensorflow.keras.preprocessing.image import img_to_array`: Nhập tiện ích để chuyển đổi ảnh thành mảng NumPy, hữu ích cho xử lý ảnh trong các tác vụ học máy.

`from tensorflow.keras.utils import set_random_seed`: Nhập tiện ích để đặt ngẫu nhiên hạt giống (seed) cho việc tái sản xuất kết quả trong TensorFlow.

`import numpy as np`: Nhập thư viện NumPy, gói cơ bản cho tính toán khoa học với Python, đặc biệt là cho các thao tác mảng và ma trận.

`import cvlib`: Nhập thư viện CVlib, cung cấp API cao cấp cho các tác vụ thị giác máy tính.

`import tkinter as tk`: Nhập thư viện Tkinter, thư viện giao diện người dùng (GUI) chuẩn trong Python.

`from tkinter import *`: Nhập tất cả các mô-đun và lớp từ Tkinter để xây dựng ứng dụng GUI.

`from PIL import ImageTk, Image`: Nhập các mô-đun ImageTk và Image từ Thư viện Hình ảnh Python (PIL) để xử lý ảnh trong các ứng dụng Tkinter.

`from tkinter import messagebox`: Nhập mô-đun messagebox từ Tkinter để tạo các hộp thông báo pop-up trong ứng dụng GUI.

`import threading`: Nhập mô-đun threading, cung cấp hỗ trợ cho việc thực thi mã đồng thời bằng cách sử dụng các luồng.

### 3.2.3.2 Tạo cửa sổ bằng thư viện Tkinter

```
# Tạo cửa sổ tkinter
root = tk.Tk()
root.geometry('1200x700')
root.resizable(False, False)
root.configure(padx=10, pady=10)
root.title('Ứng dụng phân tích khuôn mặt - Dự đoán giới tính & cảm xúc')
root.configure(bg="#EAEDED")
root.iconphoto(True, PhotoImage(file=ICON_PATH))

is_running = False
frame_queue = queue.Queue(maxsize=10) # Giới hạn số lượng khung hình trong hàng đợi để tránh lỗi bộ nhớ
```

### 3.2.3.3 Hàm xử lý camera

#### 3.2.3.3.1 Hàm khởi động Camera

```
# Hàm khởi động camera
def use_camera():
    global is_running
    is_running = True
    start_button.config(state="disabled")
    stop_button.config(state="normal")
    exit_button.config(state="normal")

# Tạo luồng xử lý camera
worker_thread = threading.Thread(target=camera_worker,
daemon=True)
if not hasattr(threading.current_thread(), "_workers"):
    threading.current_thread()._workers = []
threading.current_thread()._workers.append(worker_thread)
worker_thread.start()
```

#### 3.2.3.3.2 Hàm thoát chương trình

```
# Hàm thoát chương trình
def quit_program():
    answer = messagebox.askyesno("Quit", "Bạn có chắc muốn thoát không?")
    if answer:
        global is_running
        is_running = False
        if hasattr(threading.current_thread(), "_workers"):
            for thread in threading.current_thread()._workers:
                thread.join()
        root.destroy()
```

### 3.2.3.3 Hàm để dừng Camera

```
# Hàm để dừng camera
def cancel_feed():
    global is_running
    is_running = False
    start_button.config(state="normal")
    stop_button.config(state="disabled")

    if hasattr(threading.current_thread(), "_workers"):
        for thread in threading.current_thread()._workers:
            thread.join()
```

### 3.2.3.3.4 Luồng xử lý Camera

```
# Luồng xử lý camera
def camera_worker():
    global is_running
    capture = cv2.VideoCapture(0, cv2.CAP_DSHOW) # Sử dụng
DirectShow backend cho các hệ thống Windows

    while is_running:
        ret, frame = capture.read()

        # Phát hiện khuôn mặt
        faces, confidences = cvlib.detect_face(frame)

        for face, confidence in zip(faces, confidences):
            (startX, startY) = face[0], face[1]
            (endX, endY) = face[2], face[3]

            # Vẽ hình chữ nhật xung quanh khuôn mặt
            cv2.rectangle(frame, (startX, startY), (endX, endY), (0,
0, 255), 2) # Màu BGR

            # Cắt phần vùng khuôn mặt đã phát hiện
            face_crop = np.copy(frame[startY:endY, startX:endX])

            # Bỏ qua những khuôn mặt có kích thước quá nhỏ
            if face_crop.shape[0] < 10 or face_crop.shape[1] < 10:
                continue

            # Tiền xử lý khuôn mặt cho dự đoán giới tính
            face_crop = cv2.resize(face_crop, (150, 150),
interpolation=cv2.INTER_AREA)
            face_crop = face_crop.astype("float") / 255.0
            face_crop = img_to_array(face_crop)
            face_crop = np.expand_dims(face_crop, axis=0)
```

```

# Dự đoán giới tính và cảm xúc
conf_model_gender = gender_model.predict(face_crop)[0]
idx_model_gender = np.argmax(conf_model_gender)
label_model_gender = gender_labels[idx_model_gender]

conf_model_emotion = emotion_model.predict(face_crop)[0]
idx_model_emotion = np.argmax(conf_model_emotion)
label_model_emotion = emotion_labels[idx_model_emotion]

# Gắn nhãn giới tính và cảm xúc lên ảnh
label = "{}, {}".format(label_model_gender,
label_model_emotion)
label = "{}: {}".format(label_model_gender,
label_model_emotion)
Y = startY - 10 if startY - 10 > 10 else startY + 10
cv2.putText(frame, label, (startX, Y),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 1, cv2.LINE_AA)

break # Chỉ xử lý khuôn mặt đầu tiên được phát hiện

# Chuyển đổi ảnh từ định dạng BGR của OpenCV sang định dạng
ảnh PIL
image = Image.fromarray(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB))
image = image.resize((640, 480),
Image.Resampling.BICUBIC) # Sử dụng
BICUBIC để tăng chất lượng khi thu nhỏ hoặc phóng to ảnh

# Đưa khung hình vào hàng đợi để luồng giao diện xử lý
if not frame_queue.full():
    frame_queue.put(image)

# Lắng nghe phím 'q' để thoát
if cv2.waitKey(1) & 0xFF == ord('q'):
    is_running = False
    continue
capture.release()
cv2.destroyAllWindows()

```

### 3.2.3.3.5 Hàm xử lý hàng đợi

```

# Hàm xử lý hàng đợi và hiển thị khung hình mới nhất lên giao diện
def process_queue():
    if not frame_queue.empty():
        image = frame_queue.get()

# Cập nhật khung hình trên giao diện

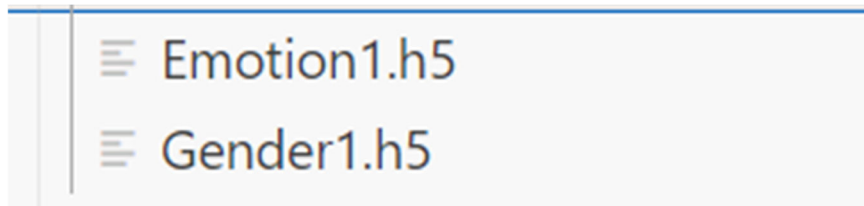
```

```
imgtk = ImageTk.PhotoImage(image=image)
image_label.config(image=imgtk)
image_label.image = imgtk
```

```
# Lên lịch kiểm tra khung hình tiếp theo
root.after(10, process_queue)
```

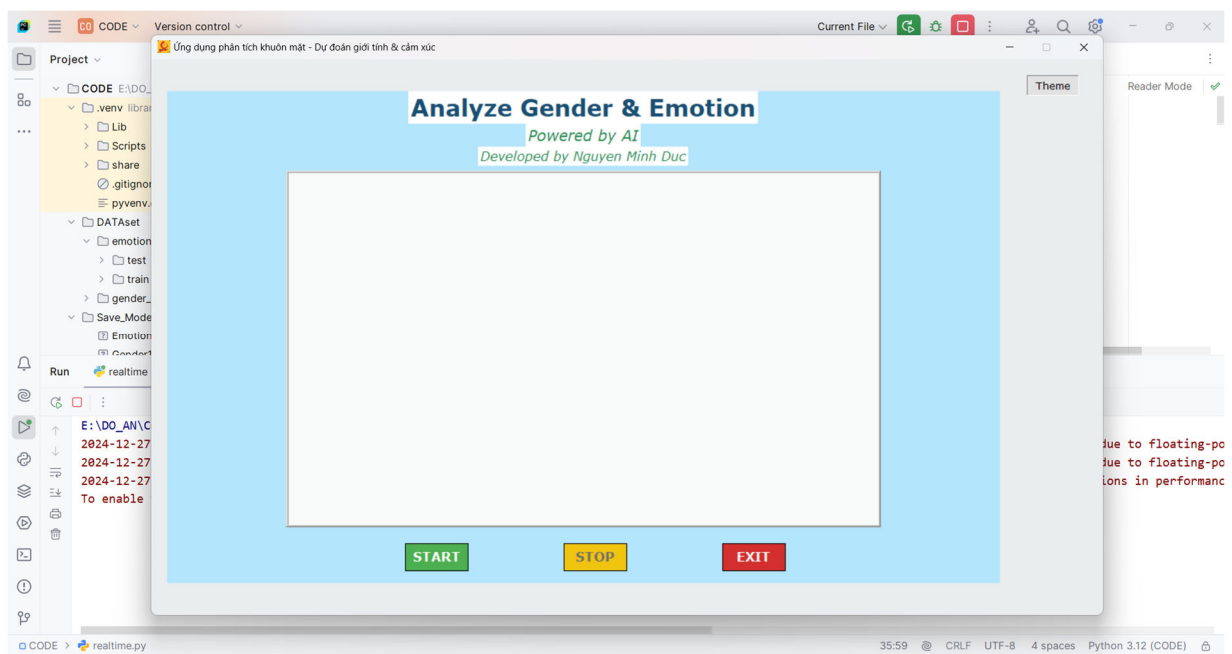
### 3.3 Kết quả thực nghiệm

#### 3.3.1 Lưu model



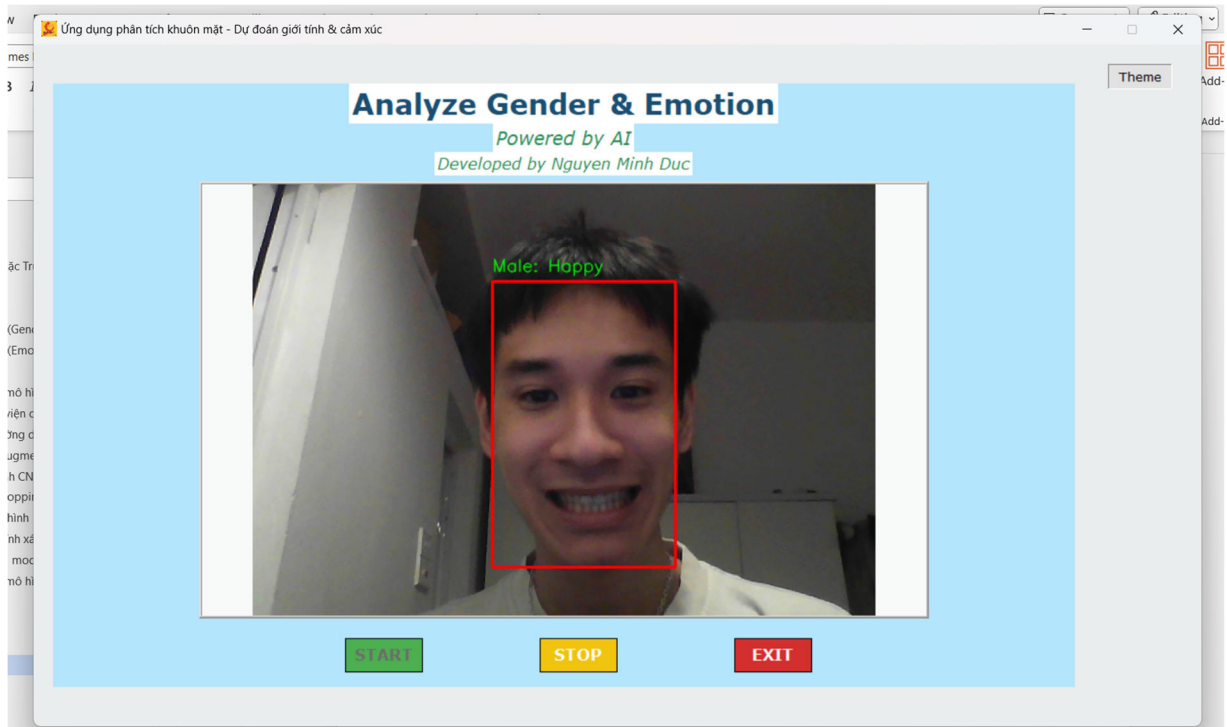
Hình 3-6: Ảnh lưu model

#### 3.3.2 Giao diện demo

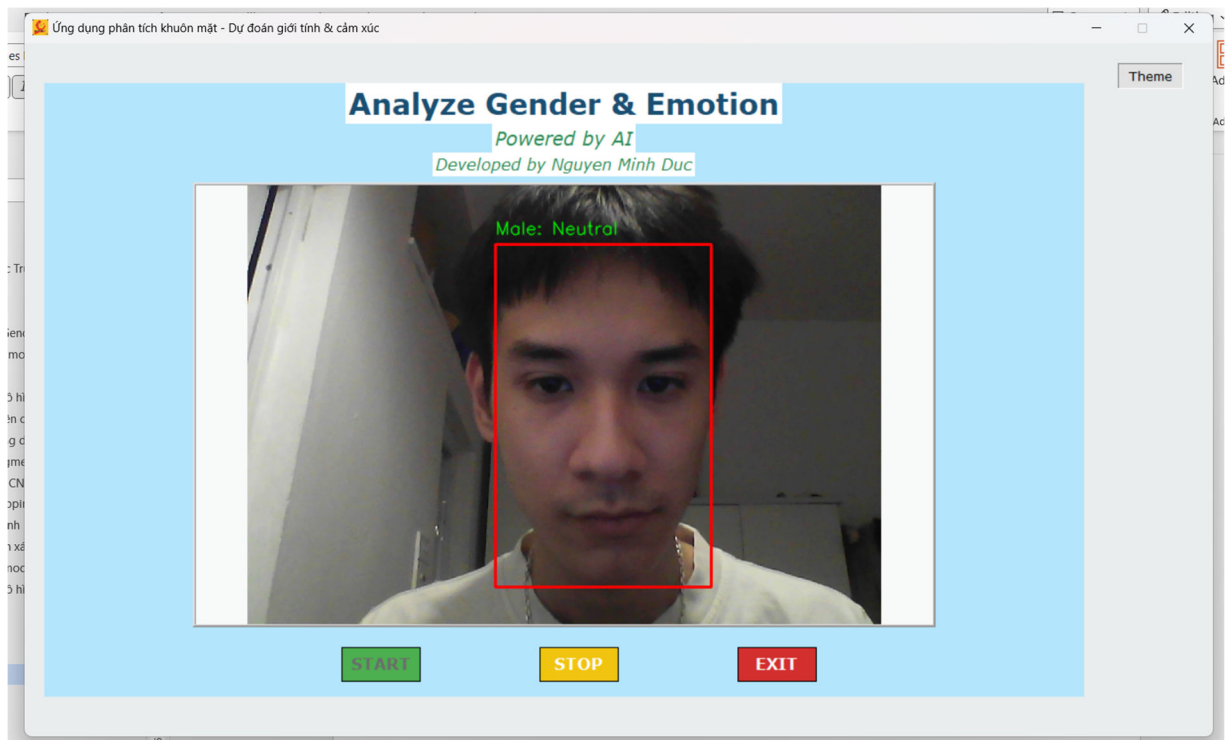


Hình 3-7: Giao diện

### 3.3.3 Kết quả đạt được



*Hình 3-8: Kết quả nhận diện male-happy*



*Hình 3-9: Kết quả nhận diện male-neutral*



## KẾT LUẬN

Kết quả nghiên cứu đã chứng minh rằng việc áp dụng một số kỹ thuật xử lý ảnh vào quá trình nhận biết cảm xúc qua biểu cảm khuôn mặt mang lại hiệu suất đáng kể. Những kỹ thuật này không chỉ giúp cải thiện độ chính xác trong việc nhận diện cảm xúc mà còn tăng cường khả năng xử lý và đánh giá thông tin từ hình ảnh khuôn mặt.

Trong quá trình thực hiện nghiên cứu, chúng tôi đã ứng dụng các phương pháp tiên tiến như học sâu (deep learning), trí tuệ nhân tạo (AI) và các thuật toán phân loại để xây dựng mô hình nhận biết cảm xúc. Mô hình đã được huấn luyện trên một tập dữ liệu đa dạng, gồm nhiều biểu cảm khác nhau trên khuôn mặt, để đảm bảo tính tổng quát và khả năng ứng dụng rộng rãi.

Kết quả đánh giá cho thấy mô hình của chúng tôi có khả năng nhận biết cảm xúc với độ chính xác cao, vượt qua nhiều mô hình khác và cảm nhận chính xác các biểu cảm phức tạp trên khuôn mặt. Sự kết hợp giữa các phương pháp xử lý ảnh và mô hình học sâu đã đóng góp tích cực vào khả năng hiểu biểu cảm và nâng cao hiệu suất của ứng dụng.

Ứng dụng nhận biết cảm xúc thông qua biểu cảm khuôn mặt của chúng tôi có thể được tích hợp vào nhiều lĩnh vực, từ giao tiếp trực tuyến, giáo dục, đến y tế và giải trí. Việc này mở ra nhiều cơ hội để tận dụng các thông tin ẩn sau biểu cảm khuôn mặt, tăng cường trải nghiệm người dùng và cải thiện tương tác giữa con người và máy tính.

Trong tương lai, chúng tôi đề xuất tiếp tục nghiên cứu và phát triển để cải thiện độ chính xác và đa dạng của mô hình, cũng như mở rộng ứng dụng của nó trong các lĩnh vực mới. Đồng thời, sự tích hợp của các kỹ thuật mới và việc tối ưu hóa hiệu suất sẽ là những hướng phát triển quan trọng trong nghiên cứu này.

## TÀI LIỆU THAM KHẢO

- [1] [https://viblo.asia/p/nguyen-cuu-va-ung-dung-cac-ky-thuat-nhan-dang-cam-xuc-qua-khuon-mat-ORNZqdmeK0n?form=MG0AV3#\\_i-gioi-thieu-ve-nhan-dang-cam-xuc-qua-khuon-mat-0](https://viblo.asia/p/nguyen-cuu-va-ung-dung-cac-ky-thuat-nhan-dang-cam-xuc-qua-khuon-mat-ORNZqdmeK0n?form=MG0AV3#_i-gioi-thieu-ve-nhan-dang-cam-xuc-qua-khuon-mat-0)
- [2] <https://www.tinhoccoban.net/2018/10/tu-hoc-xu-ly-anh-nhap-mon-xu-ly-anh.html>
- [3] <https://viblo.asia/p/nhung-cau-hoi-trong-phong-van-deep-learning-p1-YWOZrbPpZQ0?form=MG0AV3>
- [4] <https://aws.amazon.com/vi/what-is/facial-recognition/>
- [5] <https://arxiv.org/abs/2001.00562>
- [6] <https://www.sciencedirect.com/science/article/pii/S2210832715000260>
- [7] <https://www.springer.com/gp/book/9783319249474>
- [8] <https://ieeexplore.ieee.org/document/7506114>
- [9] <https://www.sciencedirect.com/topics/computer-science/facial-emotion-recognition>
- [10] <https://dl.acm.org/doi/10.1145/3197026.3197043>
- [11] <https://dl.acm.org/doi/10.1145/3054316>
- [12] [https://link.springer.com/chapter/10.1007/978-3-319-40349-6\\_21](https://link.springer.com/chapter/10.1007/978-3-319-40349-6_21)
- [13] <https://www.mdpi.com/1424-8220/20/8/2390>
- [14] <https://ieeexplore.ieee.org/document/8595366>