**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI 09:  XÂY DỰNG HỆ THỐNG PHÂN LOẠI CẢM XÚC QUA KHUÔN MẶT CỦA CON NGƯỜI**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã sinh viên** | **Lớp hành chính** |
| **1** | **Đinh Quang Phong** | **20210882** | **DCCNTT12.10.3** |
| **2** | **Phạm Trung Anh** | **20210705** | **DCCNTT12.10.3** |
| **3** | **Vũ Thanh Hải** | **20210721** | **DCCNTT12.10.3** |
| **4** | **Lương Hữu Anh** | **20210678** | **DCCNTT12.10.3** |
| **5** | **Vũ Hoàng Công Anh** | **20210856** | **DCCNTT12.10.3** |

**Bắc Ninh, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**

**Mã đề thi: 09**

**Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Lớp tín chỉ: CNPM.03.K12.03.LT.C04.1\_LT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** |
| **1** | **Đinh Quang Phong** | **20210882** |  |  |
| **2** | **Phạm Trung Anh** | **20210705** |  |  |
| **3** | **Vũ Thanh Hải** | **20210721** |  |  |
| **4** | **Lương Hữu Anh** | **20210678** |  |  |
| **5** | **Vũ Hoàng Công Anh** | **20210856** |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM 1**  *(Ký và ghi rõ họ tên)* | **CÁN BỘ CHẤM 2**  *(Ký và ghi rõ họ tên)* |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 09**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín chỉ: CNPM.03.K12.03.LT.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Đinh Quang Phong** | **Phạm Trung Anh** | **Vũ Hoàng Công Anh** | **Lương Hữu Anh** | **Vũ Thanh Hải** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **20210882** | **20210705** | **20210856** | **20210678** | **20210721** |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |  |

**MỤC LỤC**

[LỜI NÓI ĐẦU 8](#_Toc184472226)

[CHƯƠNG 1: NHẬN DẠNG ĐỐI TƯỢNG 9](#_Toc184472227)

[1.1.1. Mục tiêu nhận dạng 9](#_Toc184472228)

[1.1.2. Các vấn đề đặt ra 9](#_Toc184472229)

[1.1.3 Các công việc trong bài toán nhận dạng 10](#_Toc184472230)

[1.2 Các kĩ thuật nhận dạng nhận dạng đối tượng 10](#_Toc184472231)

[1.3 Ngôn Ngữ Lập Trình Và Thư Viện 20](#_Toc184472232)

[1.3.1 python 20](#_Toc184472233)

[1.3.2 Các Thư Viện 21](#_Toc184472234)

[CHƯƠNG 2: XÂY DỰNG HỆ THỐNG 22](#_Toc184472235)

[2.1. Bài toán 22](#_Toc184472236)

[2.1.1 Nhận dạng cảm xúc từ khuôn mặt 22](#_Toc184472237)

[2.1.2 Tiền xử lý dữ liệu 22](#_Toc184472238)

[2.1.2 Tiền xử lý dữ liệu 22](#_Toc184472239)

[2.1.3 Tăng cường dữ liệu (Data Augmentation) 22](#_Toc184472240)

[2.1.4 Xây dựng mô hình CNN 23](#_Toc184472241)

[2.1.5 Huấn luyện mô hình 23](#_Toc184472242)

[2.1.6 Đánh giá mô hình 23](#_Toc184472243)

[2.1.7 Nhận dạng cảm xúc thời gian thực 23](#_Toc184472244)

[2.2 Các kĩ thuật Deeplearning được dùng trong bài toán mô tả ảnh 23](#_Toc184472245)

[2.2.1 Mạng nơ ron tích chập - CNN 23](#_Toc184472246)

[2.2.1.1 Tích chập (Covolution) 24](#_Toc184472247)

[2.2.1.2 Kiến trúc mạng CNN 25](#_Toc184472248)

[CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM 27](#_Toc184472250)

[3.1 Dữ Liệu 27](#_Toc184472251)

[3.1.1 Thông tin cơ bản về FER-2013 27](#_Toc184472252)

[3.1.2 Kỹ thuật tiền xử lý dữ liệu (Data Preprocessing) 27](#_Toc184472253)

[3.1.2.1 Chuẩn hóa dữ liệu (Normalization) 28](#_Toc184472254)

[3.1.2.2 Reshape dữ liệu 28](#_Toc184472255)

[3.1.2.3 One-Hot Encoding cho nhãn 28](#_Toc184472256)

[3.1.2.4 Data Augmentation (Tăng cường dữ liệu) 28](#_Toc184472257)

[3.1.2.5 Resizing (nếu cần thiết) 28](#_Toc184472258)

[3.1.2.6 Transfer learning cho dữ liệu ảnh đầu vào 29](#_Toc184472259)

[3.1.3 Chia dữ liệu Train-Test-Validation 30](#_Toc184472260)

[3.2 Độ đo đánh giá 30](#_Toc184472261)

[3.2.1 Độ chính xác (Accuracy) 30](#_Toc184472262)

[3.2.3 Độ nhạy (Recall) 31](#_Toc184472263)

[3.2.4 Ma trận nhầm lẫn (Confusion Matrix) 31](#_Toc184472265)

[3.2.5 Thời gian chạy mô hình 33](#_Toc184472266)

[3.3 kết quả thực nghiệm 33](#_Toc184472267)

[3.3.1 Kết quả huấn luyện mô hình 33](#_Toc184472268)

# 

# LỜI NÓI ĐẦU

Cảm xúc là một phần không thể thiếu trong cuộc sống của con người, ảnh hưởng sâu sắc đến hành vi, tư duy và quyết định. Khả năng nhận diện và phân loại cảm xúc qua khuôn mặt không chỉ là một chủ đề nghiên cứu thú vị mà còn mang lại nhiều ứng dụng thiết thực trong các lĩnh vực như giáo dục, y tế, marketing và tương tác người-máy.

Trong bối cảnh phát triển nhanh chóng của công nghệ trí tuệ nhân tạo, hệ thống nhận diện và phân loại cảm xúc dựa trên khuôn mặt đang trở thành một xu hướng quan trọng. Những tiến bộ trong học sâu (deep learning) và thị giác máy tính đã tạo điều kiện cho việc xây dựng các mô hình phân tích biểu cảm khuôn mặt với độ chính xác cao.

Bài viết này trình bày về việc xây dựng một hệ thống phân loại cảm xúc qua khuôn mặt, bao gồm các bước từ thu thập dữ liệu, xử lý hình ảnh, thiết kế mô hình cho đến đánh giá hiệu suất. Chúng tôi hy vọng nghiên cứu này không chỉ cung cấp một góc nhìn toàn diện về công nghệ nhận diện cảm xúc mà còn góp phần phát triển các ứng dụng thực tế, giúp nâng cao chất lượng cuộc sống và tối ưu hóa các tương tác trong xã hội hiện đại.

# CHƯƠNG 1: NHẬN DẠNG ĐỐI TƯỢNG

**1.1.Bài toán nói chung**

## 1.1.1. Mục tiêu nhận dạng

- Mục tiêu chính của bài toán là phân loại các cảm xúc của con người thông qua các đặc trưng trên khuôn mặt. Các cảm xúc phổ biến như vui, buồn, giận dữ, sợ hãi, ngạc nhiên, kinh tởm, hoặc trung tính.

- Nhận diện khuôn mặt là bước đầu tiên để xác định các đối tượng trong ảnh, sau đó sẽ tiến hành phân tích cảm xúc dựa trên các đặc trưng khuôn mặt như mắt, miệng, biểu cảm cơ mặt,….

## 1.1.2. Các vấn đề đặt ra

*a. Thu thập và xử lý dữ liệu*

- Dữ liệu huấn luyện: Thu thập dữ liệu khuôn mặt có gắn nhãn cảm xúc. Điều này có thể đến từ các bộ dữ liệu công khai như FER-2013, AffectNet, hoặc cần tự thu thập dữ liệu từ các nguồn khác.

- Tiền xử lý ảnh: Cần chuẩn bị và xử lý ảnh đầu vào, bao gồm cắt ảnh khuôn mặt, chuẩn hóa kích thước, điều chỉnh độ sáng, độ tương phản và các yếu tố môi trường để đảm bảo sự ổn định trong phân tích.

*b. Nhận diện khuôn mặt (Face Detection)*

- Phát hiện khuôn mặt trong ảnh là một bước quan trọng để xác định vùng khuôn mặt cần phân tích.

- Sử dụng các kỹ thuật như Haar cascades, HOG + SVM, hoặc các mạng nơ-ron sâu như Mạng CNN (Convolutional Neural Networks) để phát hiện khuôn mặt trong ảnh hoặc video.

*c. Trích xuất đặc trưng (Feature Extraction)*

- Các đặc trưng cần trích xuất từ khuôn mặt có thể bao gồm đặc điểm hình dạng (dựa trên các điểm đặc trưng như mắt, mũi, miệng), đặc trưng hình ảnh, và biểu cảm cơ mặt.

- Một trong các phương pháp phổ biến là sử dụng Deep Learning, đặc biệt là Mạng CNN để tự động học các đặc trưng từ ảnh khuôn mặt.

*d. Phân loại cảm xúc*

- Sau khi trích xuất các đặc trưng từ khuôn mặt, hệ thống cần phân loại cảm xúc dựa trên những đặc trưng đó.

- Các thuật toán phân loại có thể bao gồm: SVM (Support Vector Machine), Mạng nơ-ron sâu (Deep Neural Networks), Random Forest, K-Nearest Neighbors (KNN).

Việc lựa chọn mô hình phân loại cần cân nhắc đến độ chính xác, tốc độ, và khả năng thích ứng với các tình huống thực tế.

*e. Đánh giá hiệu suất hệ thống*

- Đánh giá độ chính xác của hệ thống qua các chỉ số như accuracy, precision, recall, F1-score.

- Hệ thống cần được kiểm thử trên bộ dữ liệu mới để đảm bảo tính tổng quát và độ ổn định khi ứng dụng trong các tình huống thực tế.

- Đối phó với các tình huống thực tế: Dữ liệu đầu vào có thể có thay đổi về ánh sáng, góc nhìn khuôn mặt, hoặc cảm xúc người tham gia. Cần phát triển các phương pháp robust để xử lý những yếu tố này.

## 1.1.3 Các công việc trong bài toán nhận dạng

*a. Thu thập và chuẩn bị dữ liệu*

- Thu thập bộ dữ liệu có gắn nhãn cảm xúc.

- Tiền xử lý ảnh để chuẩn bị cho việc huấn luyện (cắt ảnh khuôn mặt, chuẩn hóa kích thước, v.v.).

*b. Xây dựng mô hình phát hiện khuôn mặt*

- Sử dụng các phương pháp như Haar Cascades hoặc Mạng CNN để nhận diện khuôn mặt trong ảnh.

*c. Trích xuất đặc trưng và huấn luyện mô hình phân loại*

- Trích xuất các đặc trưng từ khuôn mặt, bao gồm các biểu cảm cơ mặt.

Huấn luyện mô hình phân loại cảm xúc sử dụng các thuật toán như SVM, Mạng CNN, hoặc các phương pháp học sâu.

*d. Kiểm thử và tối ưu hóa hệ thống*

- Đánh giá hệ thống qua các chỉ số như accuracy và F1-score.

Tối ưu hóa mô hình cho độ chính xác và hiệu suất cao nhất, xử lý các tình huống thực tế với các yếu tố như ánh sáng, góc nhìn, và thay đổi cảm xúc.

*e. Triển khai và ứng dụng*

- Xây dựng một hệ thống ứng dụng có thể nhận diện cảm xúc của người dùng qua video hoặc ảnh chụp.

- Đảm bảo hệ thống hoạt động mượt mà và có thể áp dụng trong các tình huống thực tế, như trong các dịch vụ khách hàng, nhận diện cảm xúc trong giao tiếp, v.v.

## 1.2 Các kĩ thuật nhận dạng nhận dạng đối tượng

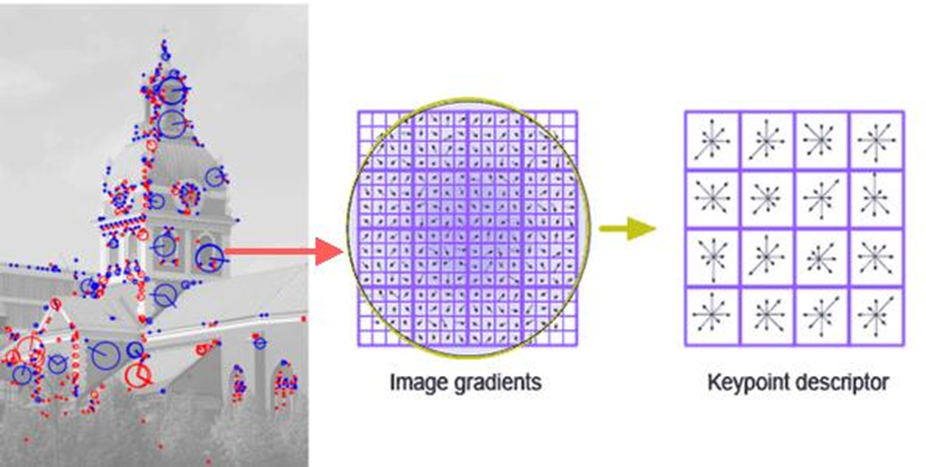
**1.Kỹ thuật truyền thống**

*a. SIFT (Scale-Invariant Feature Transform)*

- Tư tưởng:  
 SIFT (Biến đổi đặc trưng bất biến theo tỉ lệ) là một thuật toán phát hiện và mô tả các điểm đặc trưng trong ảnh.

1. Phát hiện điểm đặc trưng (keypoints) dựa trên việc tìm các điểm cực trị ở nhiều thang đo khác nhau.
2. Mô tả điểm đặc trưng bằng cách sử dụng histogram hướng gradient cục bộ.
3. Các điểm đặc trưng này không thay đổi khi ảnh bị phóng to, thu nhỏ, xoay, hoặc thay đổi ánh sáng.

- Hình minh họa:



- Ưu điểm:

* Đặc trưng ổn định với thay đổi về kích thước, góc xoay và ánh sáng.
* Hiệu quả cao cho các bài toán nhận dạng đối tượng, ghép ảnh (image stitching).

- Nhược điểm:

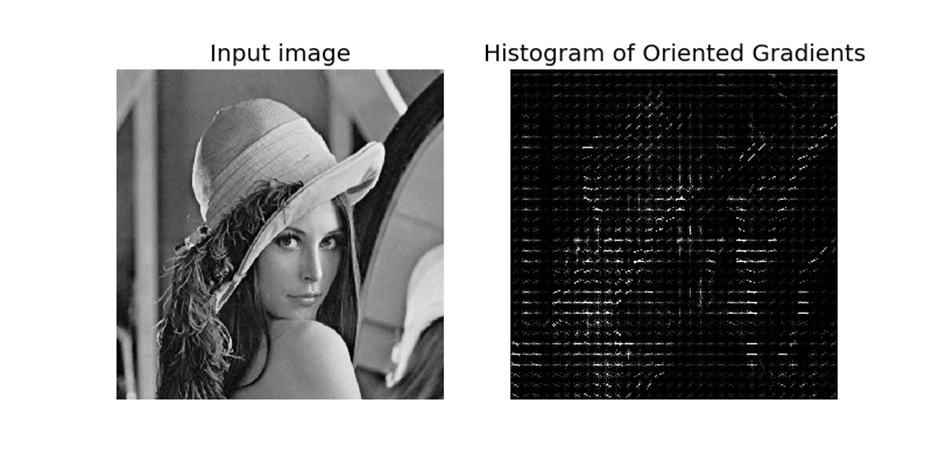
* Tốn tài nguyên và thời gian xử lý với ảnh có độ phân giải cao.
* Không phù hợp cho các ứng dụng thời gian thực.

*b. HOG (Histogram of Oriented Gradients)*

- Tư tưởng:  
 HOG (Biểu đồ gradient định hướng) chia ảnh thành các ô nhỏ (cells), tính gradient theo hướng trong mỗi ô, sau đó gom thành histogram.

1. HOG biểu diễn hình dạng đối tượng dựa trên sự phân bố của hướng gradient.
2. Các đặc trưng được chuẩn hóa để tăng khả năng nhận dạng trong điều kiện ánh sáng thay đổi.

- Hình minh họa:



- Ưu điểm:

* Hiệu quả trong nhận dạng hình dạng (như người đi bộ, phương tiện).
* Tính toán nhanh hơn các kỹ thuật như SIFT.

- Nhược điểm:

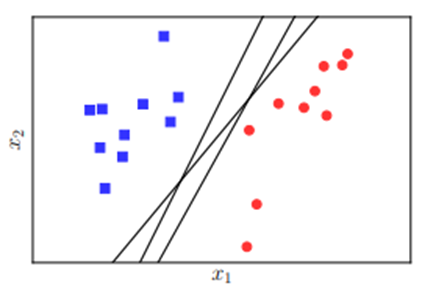
* Hiệu quả giảm khi đối tượng bị thay đổi góc nhìn lớn.
* Không hiệu quả trong trường hợp ảnh phức tạp hoặc nhiễu mạnh.

*c. SVM (Support Vector Machine)*

- Tư tưởng:  
 SVM (Máy vector hỗ trợ) là một mô hình phân loại dựa trên ý tưởng tìm siêu phẳng tối ưu để phân chia dữ liệu thành các lớp.

1. Dữ liệu đầu vào được ánh xạ vào không gian cao chiều (kernel trick).
2. Tìm siêu phẳng với khoảng cách tối đa giữa các điểm gần nhất của mỗi lớp (margin).

- Hình minh họa:



- Ưu điểm:

* Hiệu quả với dữ liệu có số chiều cao và ít nhiễu.
* Có thể sử dụng với dữ liệu không tuyến tính nhờ các kernel (RBF, polynomial).

- Nhược điểm:

* Khó áp dụng cho dữ liệu lớn vì yêu cầu tài nguyên cao.
* Nhạy cảm với dữ liệu không cân bằng (imbalanced data).

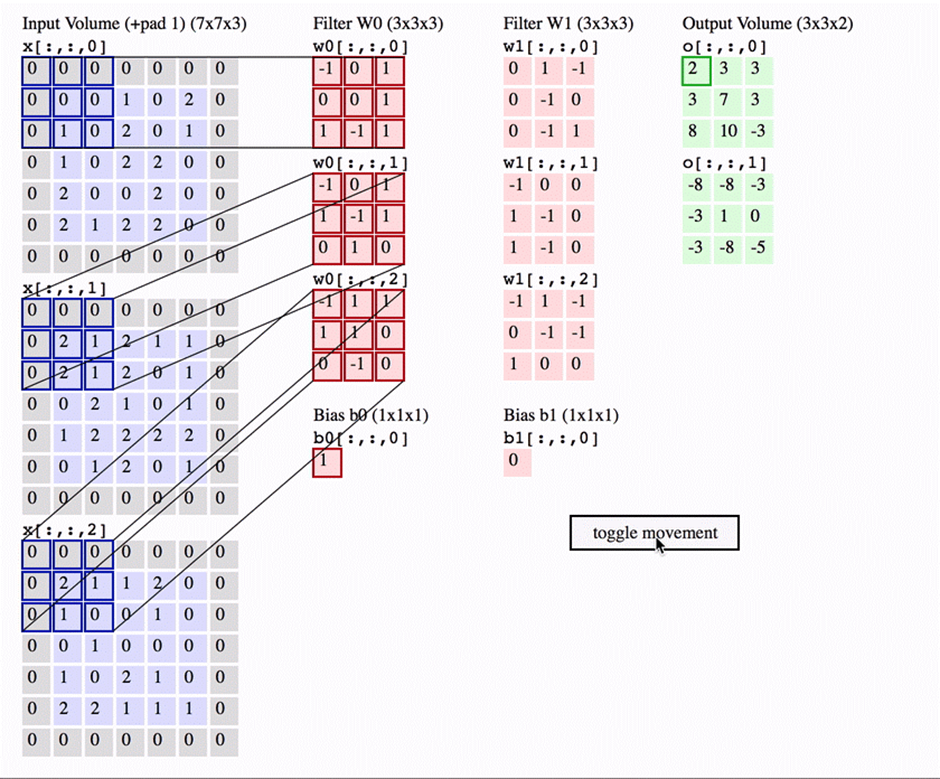
**2. Kỹ thuật học sâu (Deep Learning)**

*a. CNN (Convolutional Neural Network)*

- Tư tưởng:  
 CNN (Mạng nơ-ron tích chập) là một loại mạng nơ-ron chuyên biệt, hoạt động tốt trong các bài toán nhận dạng hình ảnh.

1. Lớp tích chập: Trích xuất các đặc trưng cục bộ từ ảnh.
2. Lớp pooling: Giảm kích thước đặc trưng mà không làm mất thông tin quan trọng.
3. Lớp fully connected: Tổng hợp và phân loại đặc trưng.

-Hình minh họa:



- Ưu điểm:

* Tự động trích xuất đặc trưng, không cần can thiệp thủ công.
* Hiệu quả cao trong các bài toán thị giác (ảnh, video).

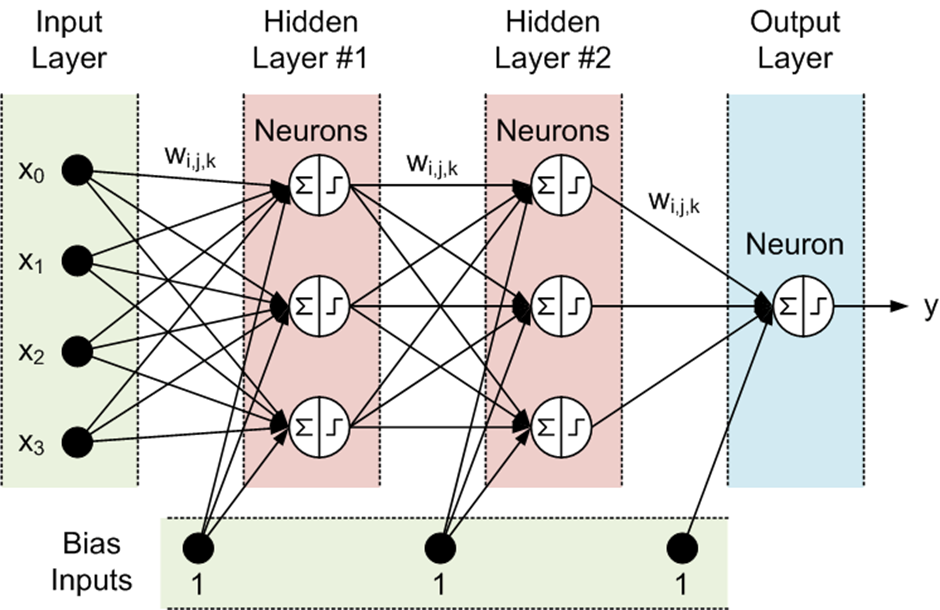
- Nhược điểm:

* Yêu cầu dữ liệu lớn và thời gian huấn luyện lâu.
* Rất tốn tài nguyên phần cứng (GPU/TPU).

*b. RNN (Recurrent Neural Network)*

- Tư tưởng:  
 RNN (Mạng nơ-ron hồi tiếp) học dữ liệu tuần tự bằng cách duy trì một trạng thái ẩn, cho phép mạng sử dụng thông tin từ các bước trước đó. Các biến thể như LSTM (Long Short-Term Memory) khắc phục vấn đề gradient biến mất.

- Hình minh họa:



- Ưu điểm:

* Phù hợp với dữ liệu chuỗi như văn bản, âm thanh, thời gian.
* Có khả năng ghi nhớ mối quan hệ dài hạn trong chuỗi (nhờ LSTM/GRU).

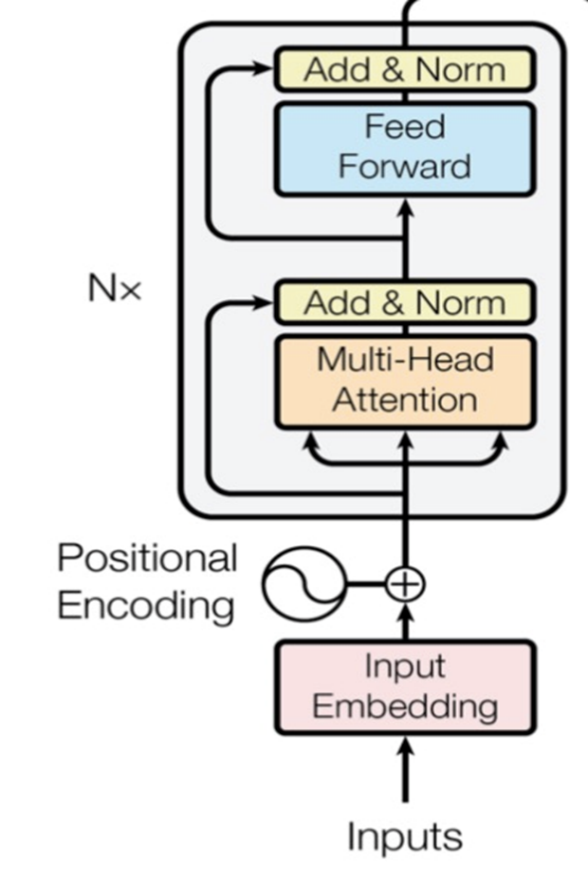
- Nhược điểm:

* Hiệu suất giảm với chuỗi quá dài.
* Huấn luyện chậm hơn CNN.

*c. Transformer (Attention Mechanism)*

- Tư tưởng:  
 Transformer sử dụng cơ chế attention để xác định tầm quan trọng của từng phần dữ liệu trong chuỗi, giúp xử lý song song tốt hơn RNN.

- Hình minh họa:



- Ưu điểm:

* Hiệu quả với dữ liệu chuỗi dài (NLP, thời gian).
* Hoạt động song song, tăng tốc độ xử lý.

- Nhược điểm:

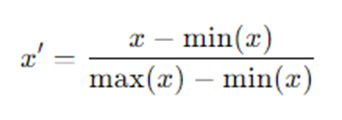
* Rất tốn tài nguyên tính toán và bộ nhớ.

**3. Kỹ thuật xử lý trước dữ liệu (Data Preprocessing)**

*a. Chuẩn hóa dữ liệu (Normalization)*

- Tư tưởng:

Dữ liệu đầu vào thường có đơn vị, thang đo khác nhau, điều này có thể làm sai lệch kết quả của các thuật toán. Chuẩn hóa là quá trình đưa các đặc trưng về cùng một thang đo (thường từ 0 đến 1 hoặc -1 đến 1). Công thức phổ biến là:



- Ưu điểm:

* Giảm tác động của các đặc trưng có giá trị lớn hoặc đơn vị khác nhau.
* Tăng độ ổn định và tốc độ hội tụ của mô hình học máy.

- Nhược điểm:

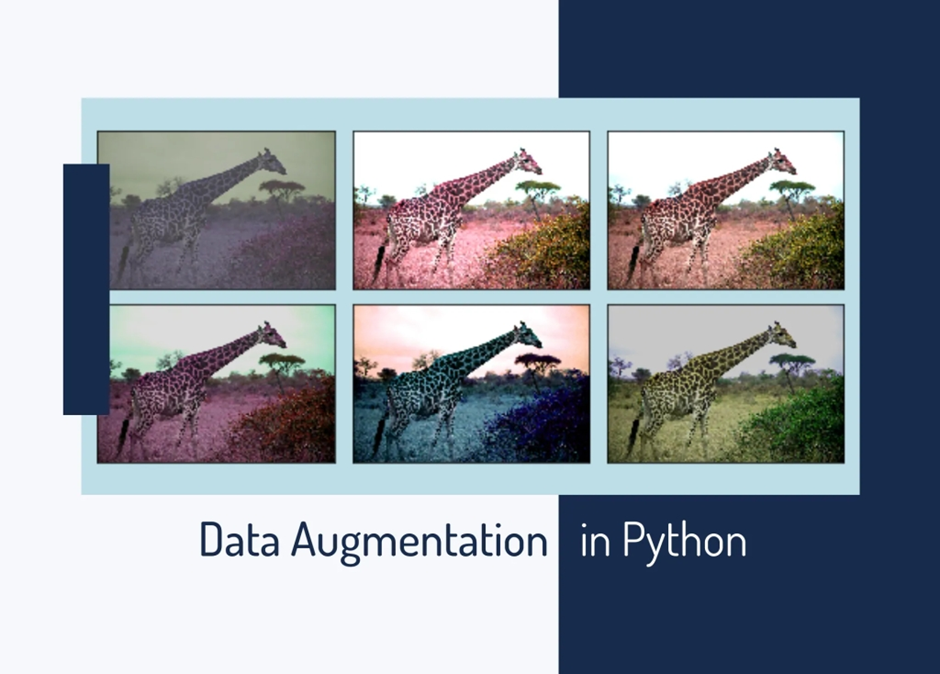
* Có thể làm mất ý nghĩa vật lý của dữ liệu nếu áp dụng không đúng.

*b. Tăng cường dữ liệu (Data Augmentation)*

- Tư tưởng:

Trong các bài toán nhận dạng, đặc biệt là học sâu, dữ liệu thường không đủ phong phú. Tăng cường dữ liệu tạo ra các phiên bản mới từ dữ liệu hiện tại bằng cách áp dụng các phép biến đổi như: xoay ảnh, lật ảnh, thay đổi độ sáng, thêm nhiễu Gaussian.

- Hình minh họa:



- Ưu điểm:

* Tăng tính đa dạng cho tập dữ liệu mà không cần thu thập thêm dữ liệu thực.
* Giảm nguy cơ quá khớp (overfitting).

- Nhược điểm:

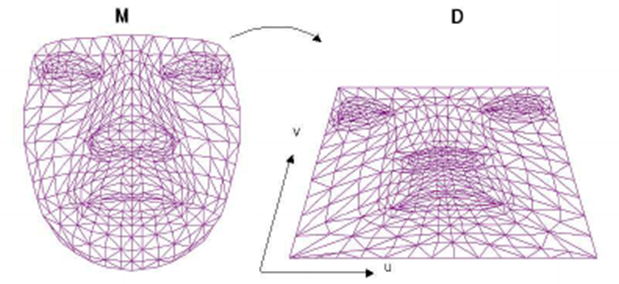
* Một số phép biến đổi có thể làm thay đổi bản chất của dữ liệu (ví dụ: xoay chữ số 6 thành 9).
* Tốn thời gian tiền xử lý.

*c. Giảm chiều dữ liệu (Dimensionality Reduction)*

- Tư tưởng:

Dữ liệu đầu vào có thể chứa nhiều đặc trưng không quan trọng hoặc dư thừa. Các phương pháp giảm chiều như PCA (Principal Component Analysis) giúp giữ lại các đặc trưng quan trọng nhất và loại bỏ nhiễu.

- Hình minh họa:



-  Ưu điểm:

* Giảm độ phức tạp tính toán và tài nguyên xử lý.
* Loại bỏ nhiễu, tăng hiệu suất của mô hình.

- Nhược điểm:

* Có thể mất thông tin quan trọng nếu giảm chiều không hợp lý.

**4. Kỹ thuật kết hợp (Hybrid Techniques)**

*a. Ensemble Learning (Học ghép nhóm)*

- Tư tưởng:  
 Ensemble Learning kết hợp nhiều mô hình học máy để cải thiện độ chính xác và tính ổn định của hệ thống. Các phương pháp phổ biến:

1. Bagging (Bootstrap Aggregating): Tạo các tập dữ liệu con từ tập gốc, huấn luyện nhiều mô hình độc lập (ví dụ: Random Forest), sau đó trung bình kết quả hoặc dùng bỏ phiếu đa số.
2. Boosting: Xây dựng mô hình từng bước, trong đó mỗi mô hình mới tập trung sửa lỗi của mô hình trước (ví dụ: AdaBoost, XGBoost).

- Ưu điểm:

* Cải thiện độ chính xác so với từng mô hình đơn lẻ.
* Giảm hiện tượng quá khớp (overfitting).

- Nhược điểm:

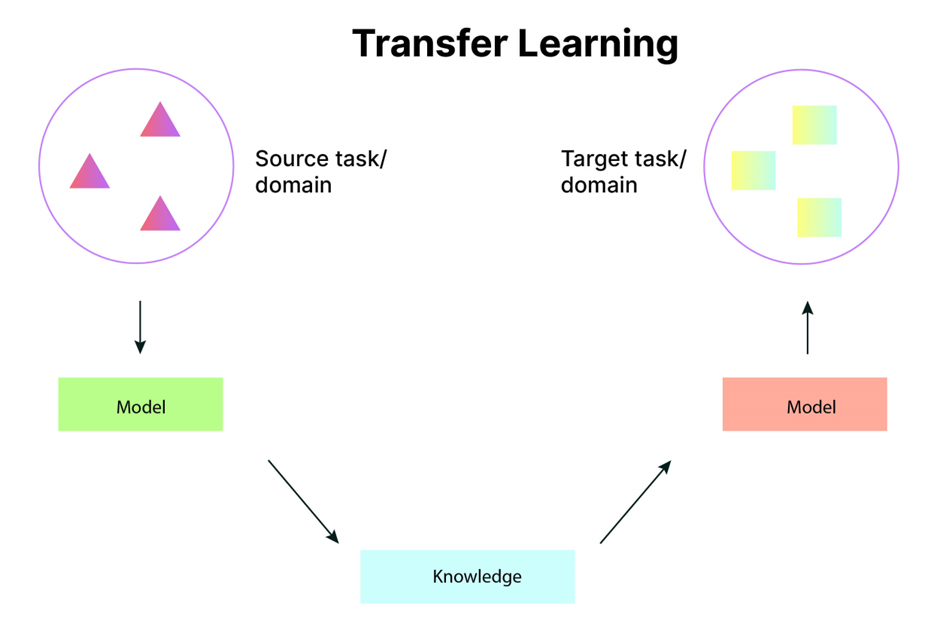
* Tăng độ phức tạp và tài nguyên tính toán.
* Khó triển khai trong thời gian thực với Boosting.

*b. Transfer Learning (Học chuyển giao)*

- Tư tưởng:  
 Transfer Learning tái sử dụng kiến thức từ một bài toán nguồn (source task) cho bài toán đích (target task).

1. Sử dụng một mô hình đã được huấn luyện trên tập dữ liệu lớn (ví dụ: ResNet, BERT).
2. Tinh chỉnh (fine-tuning) mô hình trên dữ liệu của bài toán cụ thể.

- Hình minh họa:



- Ưu điểm:

* Tiết kiệm thời gian và tài nguyên huấn luyện.
* Hiệu quả cao ngay cả với dữ liệu nhỏ.

- Nhược điểm:

* Phụ thuộc vào chất lượng của mô hình tiền huấn luyện.
* Có thể gặp hiện tượng "catastrophic forgetting" nếu tinh chỉnh không đúng cách.

*c. AutoML (Automated Machine Learning)*

- Tư tưởng:  
 AutoML tự động hóa quá trình thiết kế, chọn mô hình, tinh chỉnh siêu tham số và huấn luyện, giảm sự can thiệp thủ công. Các công cụ phổ biến như Google AutoML, Auto-Keras giúp tăng hiệu quả triển khai các giải pháp học máy.

- Hình minh họa:  
 Sơ đồ quy trình AutoML với các bước tự động: lựa chọn mô hình, tối ưu siêu tham số, huấn luyện và đánh giá.

- Ưu điểm:

* Tiết kiệm thời gian và công sức.
* Dễ dàng tiếp cận với người mới bắt đầu.

- Nhược điểm:

* Giới hạn trong khả năng tùy chỉnh.
* Yêu cầu tài nguyên cao để thử nghiệm nhiều cấu hình.

**5. Kỹ thuật tối ưu hóa và cải tiến mô hình**

*a. Hyperparameter Tuning (Tinh chỉnh siêu tham số)*

- Tư tưởng:  
 Tinh chỉnh các siêu tham số như tốc độ học (learning rate), số lớp, số lượng nơ-ron, kích thước batch... nhằm đạt hiệu suất tốt nhất cho mô hình.

* Grid Search: Dò tìm toàn bộ không gian siêu tham số theo lưới định sẵn.
* Random Search: Lựa chọn ngẫu nhiên một số cấu hình siêu tham số để thử nghiệm.
* Bayesian Optimization: Dự đoán các siêu tham số tốt nhất dựa trên các lần thử nghiệm trước đó.

- Ưu điểm:

* Cải thiện đáng kể hiệu suất mô hình.
* Các phương pháp như Bayesian Optimization tiết kiệm tài nguyên hơn Grid Search.

- Nhược điểm:

* Grid Search tốn nhiều tài nguyên với không gian lớn.
* Random Search không đảm bảo tìm được cấu hình tốt nhất.

b. Early Stopping

- Tư tưởng:  
 Dừng huấn luyện sớm khi mô hình không còn cải thiện trên tập xác thực, nhằm tránh hiện tượng quá khớp (overfitting).

1. Theo dõi hiệu suất trên tập xác thực sau mỗi epoch.
2. Dừng lại nếu hiệu suất không tăng trong một số epoch liên tiếp.

- Hình minh họa:  
 Đồ thị lỗi theo số epoch, cho thấy lỗi tập xác thực bắt đầu tăng trong khi lỗi tập huấn luyện tiếp tục giảm.

- Ưu điểm:

* Ngăn chặn quá khớp hiệu quả.
* Giảm thời gian huấn luyện.

- Nhược điểm:

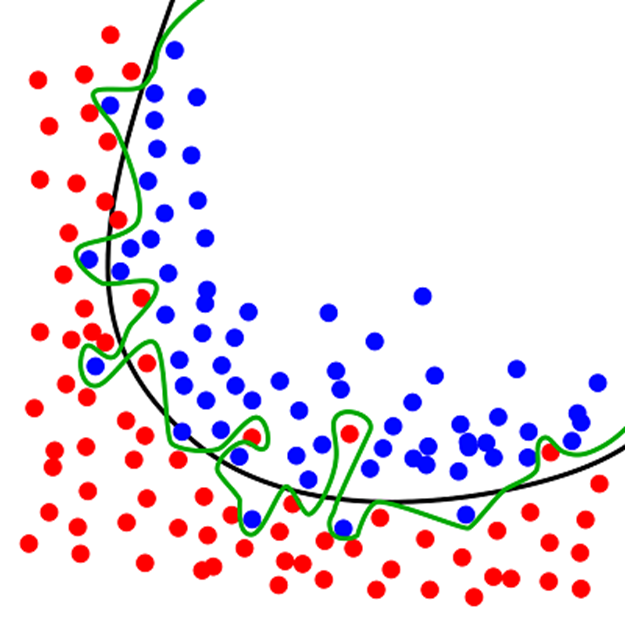
* Có thể dừng quá sớm trước khi đạt hiệu suất tối ưu.

*c. Regularization (Chính quy hóa)*

- Tư tưởng:  
 Regularization thêm ràng buộc vào quá trình huấn luyện để giảm hiện tượng quá khớp. Các phương pháp phổ biến:

* L1 Regularization (Lasso): Thêm ràng buộc dựa trên tổng giá trị tuyệt đối của các trọng số.
* L2 Regularization (Ridge): Thêm ràng buộc dựa trên tổng bình phương của các trọng số.
* Dropout: Ngẫu nhiên bỏ qua một số nơ-ron trong mỗi lần huấn luyện.

- Hình minh họa:



- Ưu điểm:

* Giảm hiện tượng quá khớp mà không làm mất hiệu suất trên tập huấn luyện.
* Dễ áp dụng với các mô hình học sâu.

- Nhược điểm:

* Regularization mạnh có thể làm giảm độ chính xác trên tập huấn luyện.

## 1.3 Ngôn Ngữ Lập Trình Và Thư Viện

## 1.3.1 python

**1. Tổng quan về Python**

- Python là một ngôn ngữ lập trình bậc cao, đa năng, hướng đối tượng, và có khả năng mở rộng.

- Tác giả: Guido van Rossum, phát hành lần đầu vào năm 1991.

- Mục tiêu: Đơn giản, dễ đọc và dễ học.

- Triết lý: Tập trung vào tính đơn giản và dễ hiểu. Python được thiết kế để đọc gần giống ngôn ngữ tự nhiên, giúp người lập trình dễ dàng viết và hiểu code.

**2. Đặc điểm nổi bật**

-  Cú pháp đơn giản: Giúp giảm thời gian viết mã và dễ bảo trì.

- Mã nguồn mở và miễn phí: Có thể tải, sử dụng và phân phối miễn phí.

- Đa nền tảng: Chạy trên nhiều hệ điều hành như Windows, macOS, Linux, và thậm chí trên các thiết bị nhúng.

- Hỗ trợ lập trình hướng đối tượng (OOP), lập trình thủ tục và lập trình hàm (Functional Programming).

**3. Ứng dụng của Python**

- Phát triển Web: Flask, Django.

- Khoa học Dữ liệu và Machine Learning: NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras.

- Tự động hóa và Script: Automate các tác vụ như đọc ghi file, xử lý dữ liệu.

- Phát triển Game: Pygame.

- Ứng dụng Desktop: Tkinter, PyQt.

- Phân tích Dữ liệu và Tính toán khoa học: SciPy, Pandas.

- Xử lý ảnh và video: OpenCV, Pillow.

- IoT và Robotics: Raspberry Pi với các thư viện như GPIO Zero.

## 1.3.2 Các Thư Viện

**1. Keras**

- Mục đích: Thư viện mã nguồn mở cho xây dựng và huấn luyện mô hình học máy, đặc biệt là các mạng nơron nhân tạo.

- Tính năng:

* API dễ sử dụng, hỗ trợ xây dựng mô hình nhanh.
* Hỗ trợ nhiều backend như TensorFlow, CNTK.

- Ứng dụng: Bạn sử dụng để xây dựng mô hình CNN nhận dạng chó và mèo.

**2. Pillow**

- Mục đích: Thư viện xử lý ảnh mạnh mẽ, mở rộng từ thư viện PIL (Python Imaging Library).

- Tính năng:

* Mở, chỉnh sửa, và lưu hình ảnh ở nhiều định dạng.
* Thực hiện các phép toán trên ảnh như thay đổi kích thước, cắt, lọc.

- Lỗi gặp phải: Trên Windows, chương trình không tìm thấy Pillow dù đã cài đặt.

**3. Convolutional Neural Networks (CNN)**

- Mục đích: Mô hình học sâu chuyên xử lý dữ liệu hình ảnh, được sử dụng để phân loại hình ảnh chó và mèo trong dự án của bạn.

- Cấu trúc cơ bản:

* Convolutional Layer: Trích xuất đặc trưng từ ảnh.
* Pooling Layer: Giảm kích thước dữ liệu, tránh overfitting.
* Fully Connected Layer: Kết nối toàn bộ mạng, đưa ra dự đoán cuối cùng.

- Ứng dụng: Bạn đã xây dựng và huấn luyện trên tập dữ liệu gồm 80 ảnh huấn luyện và 20 ảnh validation cho mỗi lớp (chó và mèo).

# CHƯƠNG 2: XÂY DỰNG HỆ THỐNG

## 2.1. Bài toán

## 2.1.1 Nhận dạng cảm xúc từ khuôn mặt

* Mục tiêu: Xây dựng một hệ thống nhận dạng cảm xúc khuôn mặt qua thời gian thực, phân loại các cảm xúc như:
  + Giận dữ (*Angry*)
  + Ghê tởm (*Disgust*)
  + Sợ hãi (*Fear*)
  + Hạnh phúc (*Happy*)
  + Buồn bã (*Sad*)
  + Ngạc nhiên (*Surprise*)
  + Bình thường (*Neutral*)
* Đầu vào:
  + Hình ảnh hoặc video từ webcam.
  + Dataset đã gắn nhãn các cảm xúc khuôn mặt.
* Đầu ra:
  + Tên cảm xúc được nhận diện trên khuôn mặt trong thời gian thực.

## 2.1.2 Tiền xử lý dữ liệu

Hình ảnh từ dataset được xử lý trước khi đưa vào mô hình:

* Chuyển đổi sang thang độ xám (grayscale): Giảm số lượng kênh màu từ 3 (RGB) xuống còn 1, giúp giảm kích thước dữ liệu và đơn giản hóa xử lý.
* Chuẩn hóa kích thước ảnh: Tất cả ảnh được resize về cùng kích thước (48x48 pixel) để phù hợp với mô hình.
* Chuẩn hóa giá trị điểm ảnh: Chia giá trị điểm ảnh cho 255 để đưa về khoảng [0, 1], giúp cải thiện tốc độ huấn luyện.

## 2.1.2 Tiền xử lý dữ liệu

Hình ảnh từ dataset được xử lý trước khi đưa vào mô hình:

* Chuyển đổi sang thang độ xám (grayscale): Giảm số lượng kênh màu từ 3 (RGB) xuống còn 1, giúp giảm kích thước dữ liệu và đơn giản hóa xử lý.
* Chuẩn hóa kích thước ảnh: Tất cả ảnh được resize về cùng kích thước (48x48 pixel) để phù hợp với mô hình.
* Chuẩn hóa giá trị điểm ảnh: Chia giá trị điểm ảnh cho 255 để đưa về khoảng [0, 1], giúp cải thiện tốc độ huấn luyện.

## 2.1.3 Tăng cường dữ liệu (Data Augmentation)

Sử dụng ImageDataGenerator để tạo ra các biến thể của dữ liệu gốc, giúp tăng tính đa dạng và giảm overfitting. Các kỹ thuật bao gồm:

* Xoay ảnh (*rotation*).
* Dịch chuyển ảnh (*width/height shift*).

Lật ngang ảnh (*horizontal flip*).

## 2.1.4 Xây dựng mô hình CNN

Mô hình mạng nơ-ron tích chập (*Convolutional Neural Network - CNN*) được sử dụng để nhận diện cảm xúc từ khuôn mặt:

* Convolutional Layer: Trích xuất các đặc trưng quan trọng từ hình ảnh (ví dụ: góc cạnh, đường viền).
* Pooling Layer: Giảm kích thước dữ liệu để tăng hiệu suất.
* Dense Layer: Kết hợp các đặc trưng và đưa ra dự đoán về cảm xúc.
* Dropout: Giảm overfitting bằng cách ngẫu nhiên bỏ qua một số nơ-ron trong quá trình huấn luyện.
* Output Layer: Sử dụng hàm kích hoạt softmax để phân loại hình ảnh thành 7 cảm xúc.

## 2.1.5 Huấn luyện mô hình

* Dữ liệu huấn luyện (train) được đưa vào mô hình để học các đặc trưng cảm xúc.
* Dữ liệu xác thực (validation) được sử dụng để kiểm tra hiệu suất mô hình sau mỗi epoch.
* Sử dụng hàm mất mát categorical\_crossentropy và tối ưu hóa bằng adam.

## 2.1.6 Đánh giá mô hình

* Vẽ biểu đồ để quan sát độ chính xác (accuracy) và lỗi (loss) qua các epoch.
* Lưu mô hình đã huấn luyện (emotion\_model.h5) để sử dụng trong giai đoạn nhận diện thời gian thực.

## 2.1.7 Nhận dạng cảm xúc thời gian thực

* Webcam: Sử dụng OpenCV để thu hình từ webcam.
* Phát hiện khuôn mặt: Dùng cascade classifier để xác định vùng khuôn mặt.
* Tiền xử lý ảnh:
  + Cắt ảnh khuôn mặt từ frame webcam.
  + Resize và chuẩn hóa khuôn mặt.
* Dự đoán cảm xúc:
  + Sử dụng mô hình CNN đã huấn luyện để dự đoán cảm xúc.
* Hiển thị kết quả:
  + Vẽ khung xung quanh khuôn mặt và hiển thị tên cảm xúc nhận diện được.

## 2.2 Các kĩ thuật Deeplearning được dùng trong bài toán mô tả ảnh

Nhận dạng biểu cảm khuôn mặt sử dụng cơ chế encoder-decoder để huấn luyện nhận dạng biểu cảm khuôn mặt. Cụ thể: giai đoạn mã hóa (encoder), mạng nơ-ron tích chập (CNN) thường được sử dụng để phát hiện các đối tượng trong khuôn mặt do lớp tích chập cuối cùng của các mạng này cung cấp biểu diễn phong phú của hình ảnh, lớp này được sử dụng làm vectơ đặc trưng.

## 2.2.1 Mạng nơ ron tích chập - CNN

Mạng Mạng nơ ron tích chập Mạng nơ-ron tích chập (Convolutional neural network-

CNN) là một trong những mô hình deep learning tiên tiến được sử dụng phổ biến hiện nay.

## 2.2.1.1 Tích chập (Covolution)

Tích chập là phương pháp được sử dụng trong tín hiệu số. Dựa theo nguyên lý biến đổi thông tin, các nha khoa học đã sử dụng trong xử lý ảnh và video số. Convolution là kỹ thuật quan trọng trong xử lý ảnh, được sử dụng chính yếu trong các phép toán trên ảnh như: đạo hàm ảnh, làm trơn ảnh, trích xuất biên cạnh trong ảnh.

Convolution là toán tử mà ta thực hiện xoay cửa sổ 180 độ rồi sau đó áp dụng phép tương quan. Correlation là toán tử tìm sự tương quan của các vùng trên ảnh gốc. Vùng pixel trên ảnh gốc có mẫu càng giống với cửa sổ thì giá trị tại điểm tương ứng của ảnh đầu ra càng lớn.

Công thức tích chập giữa hàm ảnh f (x, y) và bộ lọc k (x, y) (kích thước mxn) được biểu diễn bởi công thức 2.1 sau:

𝑚/2 𝑛/2

𝑘(𝑥, 𝑦) ∗ 𝑓(𝑥, 𝑦) = ∑ ∑ 𝑘(𝑢, 𝑣)𝑓(𝑥 − 𝑢, 𝑦 − 𝑣) (2.1)

𝑢=−𝑚/2𝑣=−𝑛/2

Thành phần không thể thiếu của phép tích chập là ma trận kernel (bộ lọc). Điểm neo (anchor point) của kernel sẽ quyết định vùng ma trận tương ứng trên ảnh để tích chập, thông thường anchor point được chọn là tâm của kernel. Giá trị mỗi phần tử trên kernel được xem như là hệ số tổ hợp với lần lượt từng giá trị độ xám của điểm ảnh trong vùng tương ứng với kernel.

Phép tích chập được hình dung thực hiện bằng việc dịch chuyển ma trận kernel lần lượt qua tất cả các điểm ảnh trong ảnh, bắt đầu từ góc bên trái trên của ảnh. Và đặt anchor point tương ứng tại điểm ảnh đang xét. Ở mỗi lần dịch chuyển, thực hiện tính toán kết quả mới cho điểm ảnh đang xét bằng công thức tích chập.

Ví dụ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | = | 5 | 1 | 3 |
| 0 | 1 | 1 | 2 | 3 | 2 |
| 1 | 0 | 1 | 2 | 2 | 4 |

X

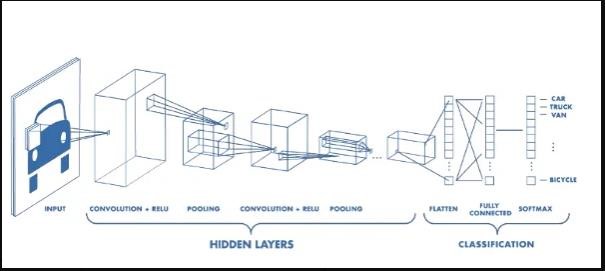
*Quá trình nhân tích chập đầu vào 6\*6, kenel 3\*3, kết quả 3\*3*

## 2.2.1.2 Kiến trúc mạng CNN

Kiến trúc mạng được chia thành 3 chiều: rộng, cao, và sâu. Các nơ-ron trong mạng có thể không liên kết hoàn toàn với toàn bộ nơ-ron kế tiếp theo. Cuối cùng, một tầng đầu ra được tối giản thành véc-tơ của giá trị xác suất. Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm phi tuyến như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

CNNs gồm hai thành phần:

* Phần tầng ẩn hay phần rút trích đặc trưng: trong phần này, mạng sẽ tiến hành tính toán hàng loạt phép tích chập và phép hợp nhất (pooling) để phát hiện các đặc trưng. Ví dụ: nếu ta có hình ảnh con ngựa vằn, thì trong phần này mạng sẽ nhận diện các sọc vằn, hai tai, và bốn chân của nó.
* Phần phân lớp: tại phần này, một lớp với các liên kết đầy đủ sẽ đóng vai trò như một bộ phân lớp các đặc trưng đã rút trích được trước đó. Tầng này sẽ đưa ra xác suất của một đối tượng trong hình.



# *Hình 2.2. Kiến trúc mạng tổng quát cho bài toán phân lớp*

CNNs có khả năng trích rút đặc trưng

Trong cấu trúc CNN, tích chập được thực hiện trên giá trị đầu vào của dữ liệu và kernel/filter để tạo ra một bản đồ đặc trưng của ảnh. Thực hiện phép tích chập bằng cách trượt kernel/filter theo dữ liệu đầu vào. Tại mỗi vị trí, tiến hành phép nhân ma trận và tính tổng các giá trị để đưa vào bản đồ đặc trưng. Tương tự như mạng nơ-ron thông thường, ta sử dụng một hàm kích hoạt để có đầu ra dưới dạng phi tuyến. Trong quá trình trượt kernel/filter trên dữ liệu đầu vào, ta sẽ quy định một bước nhảy (stride) với mỗi lần di chuyển. Thông thường ta lựa chọn thường chọn bước nhảy là 1. Nếu kích thước bước nhảy tăng, kernel/filter sẽ có ít ô trùng lặp. Kích thước đầu ra luôn nhỏ hơn đầu vào nên ta cần một phép xử lí đầu vào để đầu ra không bị co giãn bằng cách thêm một lề ở đầu vào. Một lề với giá trị 0 sẽ được thêm vào xung quanh đầu vào trước khi thực hiện phép tích chập. Thông thường, sau mỗi tầng tích chập, ta sẽ cho kết quả đi qua một tầng hợp nhất (pooling layer). Mục đích của tầng này là để nhanh chóng giảm số chiều. Việc này giúp giảm thời gian học và hạn chế việc overfitting. Pooling layer sẽ chọn được tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

##### Khả năng phân lớp đối tượng

Trong phần phân lớp, ta sử dụng một vài tầng với kết nối đầy đủ để xử lí kết quả của phần tích chập. Vì đầu vào của mạng liên kết đầy đủ là 1 chiều, ta cần làm phẳng đầu vào trước khi phân lớp. Tầng cuối cùng trong mạng CNN là một tầng liên kết đầy đủ, phần này hoạt động tương tự như mạng nơ-ron thông thường.

Kết quả thu được cuối cùng cũng sẽ là một véc-tơ với các giá trị xác suất cho việc dự đoán như mạng nơ-ron thông thường.

# 

# CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

## 3.1 Dữ Liệu

## 3.1.1 Thông tin cơ bản về FER-2013

* Nguồn gốc: FER-2013 được công bố tại cuộc thi *Challenges in Representation Learning: Facial Expression Recognition Challenge* năm 2013.
* Kích thước dữ liệu:
  + Tổng số hình ảnh: 35.887 ảnh.
  + Mỗi hình ảnh là khuôn mặt của một người, được chụp trong điều kiện ánh sáng, góc nhìn và biểu cảm khác nhau.
* Định dạng hình ảnh:
  + Màu: Grayscale (thang độ xám).
  + Kích thước: 48x48 pixels.
  + Định dạng: Các ảnh được lưu trữ dưới dạng số học trong tệp .csv.
* Số lượng lớp cảm xúc: 7 lớp:

1.Angry (Tức giận)

2. Disgust (Ghê tởm)

3. Fear (Sợ hãi)

4.Happy (Vui vẻ)

5. Sad (Buồn bã)

6. Surprise (Ngạc nhiên)

7. Neutral (Trung tính)



*3.1.1 : hình ảnh tệp dữ liệu*

## 3.1.2 Kỹ thuật tiền xử lý dữ liệu (Data Preprocessing)

Khi làm việc với dữ liệu nhận dạng cảm xúc như FER-2013, Tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết các vấn đề về học máy. Hầu hết các bộ dữ liệu được sử dụng trong các vấn đề liên quan đến học máy cần được xử lý, làm sạch và chuyển đổi trước khi đào tạo các thuật toán học máy trên các bộ dữ liệu đó. Các kỹ thuật tiền xử lý dữ liệu hiện tại bao gồm: Xử lý dữ liệu bị thiếu, mã hóa các biến phân loại, chuẩn hóa dữ liệu, chia tỷ lệ dữ liệu, v.v.. Dưới đây là các bước chi tiết:

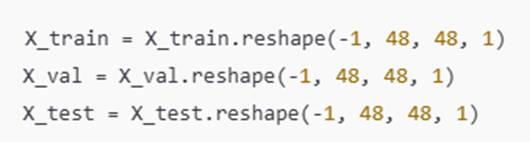
## 3.1.2.1 Chuẩn hóa dữ liệu (Normalization)

* Dữ liệu hình ảnh trong FER-2013 có giá trị pixel từ 0-255. Để mô hình học tốt hơn, ta sẽ chuẩn hóa giá trị pixel về khoảng [0, 1]:



## 3.1.2.2 Reshape dữ liệu

* Dữ liệu FER-2013 là ảnh grayscale (thang độ xám), cần reshape để thêm trục "channel" (1 kênh màu).



## 3.1.2.3 One-Hot Encoding cho nhãn

* Nhãn cảm xúc trong dataset được đánh số từ 0-6, cần chuyển đổi thành dạng ma trận nhị phân (one-hot encoding) để phù hợp với đầu ra của mô hình:

## 3.1.2.4 Data Augmentation (Tăng cường dữ liệu)

* Với dữ liệu FER-2013, augmentation giúp giảm tình trạng overfitting. Các kỹ thuật phổ biến:
  + Lật ngang (Horizontal Flip): Đảo ngược khuôn mặt.
  + Zoom (Zoom Range): Phóng to/thu nhỏ ảnh.
  + Dịch chuyển (Shift Range): Dịch khuôn mặt theo chiều ngang/dọc.
  + Xoay (Rotation): Xoay ảnh một góc nhỏ.
  + Thay đổi độ sáng (Brightness Adjustment): Giả lập các điều kiện ánh sáng khác nhau.

## 3.1.2.5 Resizing (nếu cần thiết)

* Nếu hình ảnh không phải 48x48 (như khi dùng dataset khác), cần resize ảnh về kích thước chuẩn

## 3.1.2.6 Transfer learning cho dữ liệu ảnh đầu vào

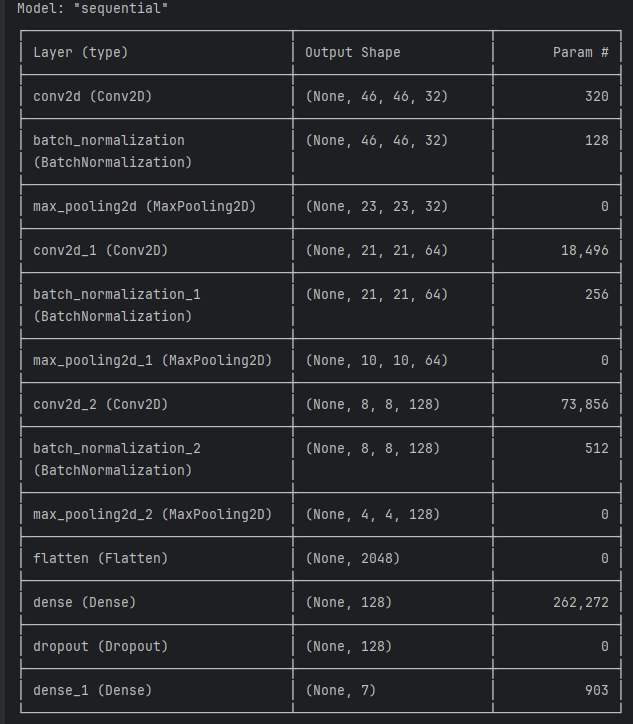
Transfer learning là một kỹ thuật trong học sâu, trong đó một mô hình đã được huấn luyện trên một tập dữ liệu lớn được tái sử dụng để giải quyết bài toán mới. Thay vì huấn luyện từ đầu (từ đầu vào ngẫu nhiên), bạn có thể tận dụng kiến thức đã học từ một mô hình đã huấn luyện trước đó và điều chỉnh cho phù hợp với dữ liệu của mình.

Kỹ thuật này đặc biệt hữu ích khi bạn có một lượng dữ liệu hạn chế, vì các mô hình đã huấn luyện từ trước (chẳng hạn như mô hình phân loại ảnh) đã học được các đặc trưng chung có thể áp dụng cho nhiều bài toán khác nhau.

Dưới đây là các bước chi tiết để thực hiện Transfer Learning với dữ liệu ảnh đầu vào trong bài toán nhận dạng cảm xúc khuôn mặt.

Các bước thực hiện Transfer Learning

1. Chọn mô hình đã huấn luyện trước (Pretrained Model)
   * Các mô hình như VGG16, ResNet50, InceptionV3, hoặc MobileNet đã được huấn luyện trên các tập dữ liệu lớn như ImageNet. Chúng có thể giúp bạn nhận diện các đặc trưng cơ bản như cạnh, kết cấu, hình dạng.
2. Tải mô hình đã huấn luyện trước
   * Sử dụng mô hình đã được huấn luyện trên ImageNet mà không cần huấn luyện lại toàn bộ mạng.
3. Chỉnh sửa mô hình (Fine-tuning)
   * Loại bỏ các lớp cuối cùng của mô hình cũ và thay thế chúng bằng một số lớp mới để phù hợp với bài toán của bạn (ví dụ: số lớp đầu ra tương ứng với các cảm xúc khác nhau).
4. Huấn luyện mô hình (Fine-tuning)
   * Bạn chỉ cần huấn luyện lại các lớp mới này với dữ liệu của bạn, giữ lại các lớp đã huấn luyện trước. Điều này giúp giảm thời gian huấn luyện và cải thiện hiệu quả mô hình.



*3.1.2.6 : một mô hình CNN (Convolutional Neural Network)*

## 3.1.3 Chia dữ liệu Train-Test-Validation

Dataset FER-2013 có sẵn thông tin về việc chia tập dữ liệu:

* Training Set: 80% dữ liệu (28.709 ảnh).
* Validation Set: 10% dữ liệu (3.589 ảnh).
* Test Set: 10% dữ liệu (3.589 ảnh).

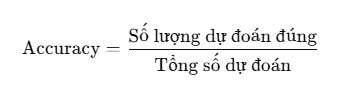
## 3.2 Độ đo đánh giá

Trong các bài toán nhận dạng cảm xúc khuôn mặt, để đánh giá hiệu suất của mô hình, chúng ta cần sử dụng các chỉ số đo lường. Những chỉ số này giúp đánh giá mô hình có hoạt động tốt hay không, hiệu quả đến mức nào, và thời gian thực hiện ra sao. Dưới đây là giải thích chi tiết từng độ đo và cách áp dụng.

## 3.2.1 Độ chính xác (Accuracy)

Độ chính xác là tỷ lệ các dự đoán chính xác so với tổng số dự đoán. Đây là một trong những chỉ số đánh giá cơ bản và dễ hiểu nhất, đặc biệt khi bài toán có các lớp cân bằng (số lượng mẫu trong mỗi lớp gần bằng nhau).

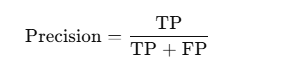
**Công thức tính**:



Trong mô hình của bạn, **accuracy** sẽ cho bạn biết tỷ lệ phần trăm các cảm xúc được dự đoán chính xác từ tổng số khuôn mặt mà mô hình nhận diện.

**3.2.2 Độ chính xác (Precision)**

**Độ chính xác** đo lường mức độ chính xác của các dự đoán "dương tính" của mô hình. Đây là tỷ lệ giữa số lượng dự đoán đúng của một lớp cụ thể (True Positives - TP) và tổng số dự đoán là dương tính (TP + False Positives - FP):



**Ý nghĩa**: Độ chính xác cao có nghĩa là mô hình ít dự đoán sai lớp cảm xúc. Cũng rất quan trọng khi bài toán yêu cầu giảm thiểu các dự đoán sai.

## 3.2.3 Độ nhạy (Recall)

**Độ nhạy** (hay còn gọi là **Sensitivity**) đo lường khả năng của mô hình trong việc nhận diện đúng các đối tượng thực sự của lớp mục tiêu. Đây là tỷ lệ giữa số lượng dự đoán đúng (TP) và tổng số đối tượng thực sự thuộc lớp đó (TP + False Negatives - FN):



**Ý nghĩa**: Độ nhạy cao cho thấy mô hình không bỏ sót quá nhiều đối tượng thực sự thuộc lớp cảm xúc. Đây là chỉ số quan trọng khi bạn không muốn bỏ qua các tình huống quan trọng.

## 3.2.4 Ma trận nhầm lẫn (Confusion Matrix)

**Ý nghĩa**:

* Ma trận nhầm lẫn cho thấy số lượng mẫu được dự đoán đúng và sai ở từng nhãn (cảm xúc).
* **Hàng**: Nhãn thực tế.
* **Cột**: Nhãn dự đoán.
* Mỗi ô trong ma trận cho biết số lượng mẫu.

**Ví dụ**: Nếu bài toán nhận dạng cảm xúc khuôn mặt có 7 cảm xúc: **Angry (Tức giận)**, **Disgust (Ghê tởm)**, **Fear (Sợ hãi)**, **Happy (Vui vẻ)**, **Sad (Buồn bã)**, **Surprise (Ngạc nhiên)**, và **Neutral (Trung tính)**, thì ma trận nhầm lẫn sẽ được thể hiện như sau:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Actual \ Predicted | Angry   |  | | --- | |  |  |  | | --- | |  | | Disgust   |  | | --- | |  |  |  | | --- | |  | | Fear | Happy | Sad | Surprise | Neutral |
| Angry | 50 | 3 | 2 | 0 | 5 | 1 | 4 |
| Disgust | 2 | 47 | 5 | 0 | 3 | 1 | 2 |
| Fear | 3 | 4 | 40 | 2 | 6 | 1 | 4 |
| Happy | 1 | 0 | 1 | 55 | 2 | 3 | 2 |
| Sad | 4 | 2 | 3 | 1 | 50 | 2 | 4 |
| Surprise | 1 | 1 | 1 | 4 | 2 | 51 | 2 |
| Neutral | 3 | 2 | 4 | 1 | 3 | 1 | 46 |

Phân tích ví dụ

* Angry (Tức giận):
  + Có 50 ảnh Angry được mô hình dự đoán đúng.
  + Có 3 ảnh Angry bị nhầm là Disgust, 2 ảnh nhầm là Fear, và 4 ảnh nhầm là Neutral.
* Disgust (Ghê tởm):
  + Có 47 ảnh Disgust được dự đoán đúng.
  + Có 5 ảnh nhầm là Fear, và 2 ảnh nhầm là Neutral.
* Fear (Sợ hãi):
  + Có 40 ảnh Fear được dự đoán đúng.
  + Có 6 ảnh Fear bị nhầm là Sad và 4 ảnh nhầm là Neutral.
* Happy (Vui vẻ):
  + Có 55 ảnh Happy được dự đoán đúng.
  + Một số nhầm lẫn: 1 ảnh nhầm là Fear, 2 ảnh nhầm là Sad, và 2 ảnh nhầm là Neutral.
* Sad (Buồn bã):
  + Có 50 ảnh Sad được dự đoán đúng.
  + Một số nhầm lẫn: 4 ảnh bị dự đoán là Angry, 2 ảnh là Disgust, và 4 ảnh là Neutral.
* Surprise (Ngạc nhiên):
  + Có 51 ảnh Surprise được dự đoán đúng.
  + Một số nhầm lẫn: 1 ảnh nhầm là Angry và 2 ảnh nhầm là Neutral.
* Neutral (Trung tính):
  + Có 46 ảnh Neutral được dự đoán đúng.
  + Một số nhầm lẫn: 3 ảnh nhầm là Angry, 2 ảnh nhầm là Disgust, và 4 ảnh nhầm là Fear.

## 3.2.6 Thời gian chạy mô hình

Thời gian chạy mô hình là một chỉ số quan trọng trong bài toán nhận diện cảm xúc qua webcam, vì mô hình cần có khả năng dự đoán trong thời gian thực.

* **Thời gian dự đoán (Inference Time)**: Thời gian mà mô hình cần để đưa ra dự đoán cho một bức ảnh hoặc khung hình từ webcam.
* **Thời gian huấn luyện (Training Time)**: Thời gian cần thiết để huấn luyện mô hình trên toàn bộ tập dữ liệu.

**Ý nghĩa**: Thời gian chạy thấp giúp mô hình hoạt động nhanh chóng và hiệu quả trong ứng dụng thời gian thực, như nhận diện cảm xúc qua webcam.

## 3.3 kết quả thực nghiệm

## 3.3.1 Kết quả huấn luyện mô hình

Epoch 1/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.2263 - loss: 1.9834

449/449 ━━━━━━━━━━━━━━━━━━━━ 78s 168ms/step - accuracy: 0.2264 - loss: 1.9831 - val\_accuracy: 0.2545 - val\_loss: 1.8872

Epoch 2/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 73s 163ms/step - accuracy: 0.2578 - loss: 1.8177 - val\_accuracy: 0.3020 - val\_loss: 1.7359

Epoch 3/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 161ms/step - accuracy: 0.2677 - loss: 1.7644 - val\_accuracy: 0.3006 - val\_loss: 1.7229

Epoch 4/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 70s 157ms/step - accuracy: 0.2903 - loss: 1.7214 - val\_accuracy: 0.2719 - val\_loss: 1.7548

Epoch 5/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 71s 158ms/step - accuracy: 0.3179 - loss: 1.6740 - val\_accuracy: 0.2816 - val\_loss: 1.8028

Epoch 6/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 70s 156ms/step - accuracy: 0.3468 - loss: 1.6292 - val\_accuracy: 0.4354 - val\_loss: 1.4673

Epoch 7/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 69s 154ms/step - accuracy: 0.3652 - loss: 1.5893 - val\_accuracy: 0.4553 - val\_loss: 1.3941

Epoch 8/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 64s 143ms/step - accuracy: 0.3841 - loss: 1.5476 - val\_accuracy: 0.4468 - val\_loss: 1.4113

Epoch 9/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 64s 143ms/step - accuracy: 0.4025 - loss: 1.5253 - val\_accuracy: 0.4823 - val\_loss: 1.4034

Epoch 10/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 64s 143ms/step - accuracy: 0.4101 - loss: 1.5084 - val\_accuracy: 0.4451 - val\_loss: 1.3970

Epoch 11/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 67s 150ms/step - accuracy: 0.4206 - loss: 1.4832 - val\_accuracy: 0.4401 - val\_loss: 1.4381

Epoch 12/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 68s 151ms/step - accuracy: 0.4290 - loss: 1.4611 - val\_accuracy: 0.4695 - val\_loss: 1.3683

Epoch 13/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 70s 156ms/step - accuracy: 0.4324 - loss: 1.4512 - val\_accuracy: 0.4887 - val\_loss: 1.3503

Epoch 14/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 69s 153ms/step - accuracy: 0.4498 - loss: 1.4263 - val\_accuracy: 0.4812 - val\_loss: 1.3622

Epoch 15/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 161ms/step - accuracy: 0.4536 - loss: 1.4173 - val\_accuracy: 0.4859 - val\_loss: 1.3743

Epoch 16/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 74s 165ms/step - accuracy: 0.4592 - loss: 1.3986 - val\_accuracy: 0.5134 - val\_loss: 1.2574

Epoch 17/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 161ms/step - accuracy: 0.4637 - loss: 1.3966 - val\_accuracy: 0.4486 - val\_loss: 1.4102

Epoch 18/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 160ms/step - accuracy: 0.4774 - loss: 1.3726 - val\_accuracy: 0.5306 - val\_loss: 1.2267

Epoch 19/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 160ms/step - accuracy: 0.4762 - loss: 1.3709 - val\_accuracy: 0.5359 - val\_loss: 1.2168

Epoch 20/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 73s 163ms/step - accuracy: 0.4849 - loss: 1.3526 - val\_accuracy: 0.5259 - val\_loss: 1.2357

Epoch 21/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 72s 161ms/step - accuracy: 0.4861 - loss: 1.3431 - val\_accuracy: 0.5164 - val\_loss: 1.3344

Epoch 22/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 73s 162ms/step - accuracy: 0.4910 - loss: 1.3395 - val\_accuracy: 0.5322 - val\_loss: 1.2552

Epoch 23/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 71s 158ms/step - accuracy: 0.4944 - loss: 1.3258 - val\_accuracy: 0.5419 - val\_loss: 1.1851

Epoch 24/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 71s 158ms/step - accuracy: 0.4962 - loss: 1.3234 - val\_accuracy: 0.5430 - val\_loss: 1.1971

Epoch 25/25

449/449 ━━━━━━━━━━━━━━━━━━━━ 70s 156ms/step - accuracy: 0.4971 - loss: 1.3268 - val\_accuracy: 0.5252 - val\_loss: 1.2374

113/113 ━━━━━━━━━━━━━━━━━━━━ 5s 41ms/step - accuracy: 0.5283 - loss: 1.2363

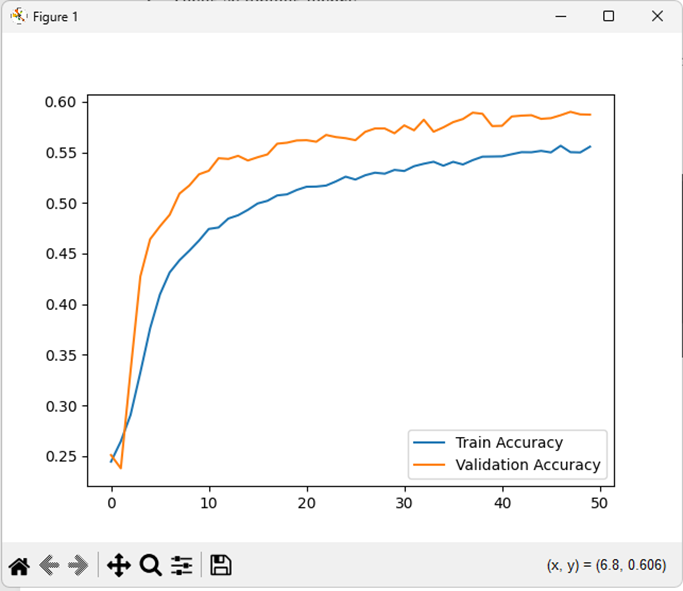
Thông số training model:

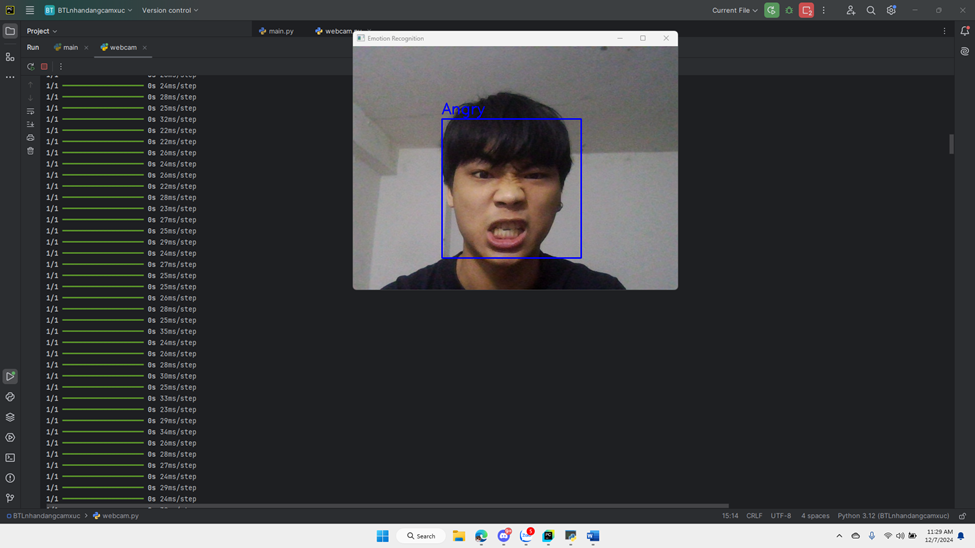
 ✓ Số Epoch: 25.

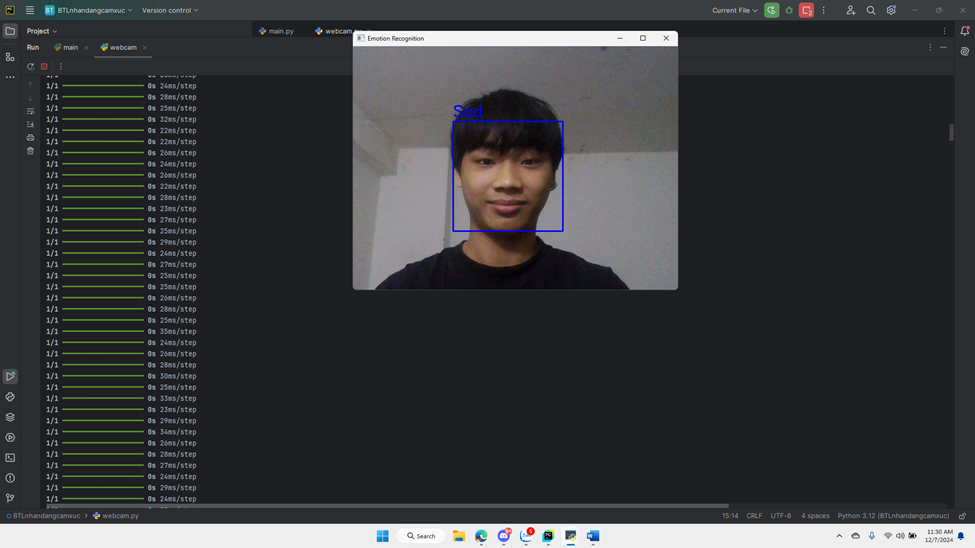
✓ Đầu ra: Model được training cho ra kết quả ước tính với các trọng số như:

Validation Loss: 1.2374

Validation Accuracy: 0.5252



****

****

****

**TÀI LIỆU THAM KHẢO**

*1.* [*https://mctt.vn/thuat-toan-xu-ly-hinh-anh*](https://mctt.vn/thuat-toan-xu-ly-hinh-anh)

*2.* [*https://dvms.com.vn/tin-tuc/tin-nganh/73466-gioi-thieu-giai-thuat-sift-de-nhan-dang-anh.html#gsc.tab=0*](https://dvms.com.vn/tin-tuc/tin-nganh/73466-gioi-thieu-giai-thuat-sift-de-nhan-dang-anh.html#gsc.tab=0)

*3.* [*https://luanvanhay.org/lam-thue/svm-thuat-toan-ho-tro-may-vec-to-support-vector-machines/*](https://luanvanhay.org/lam-thue/svm-thuat-toan-ho-tro-may-vec-to-support-vector-machines/)

*4,* [*https://nttuan8.com/bai-13-recurrent-neural-network/*](https://nttuan8.com/bai-13-recurrent-neural-network/)

*5. https://pdaotao.duytan.edu.vn/uploads/Mark/giao%20trinh%20xu%20ly%20anh.pdf*