|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BỘ CÔNG THƯƠNG**  **ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  IMG_256  **BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**  **HỌC PHẦN: TRÍ TUỆ NHÂN TẠO**  **ĐỀ TÀI:**  **NHẬN DIỆN CẢM XÚC KHUÔN MẶT NGƯỜI**  **BẰNG CNN**   |  |  |  | | --- | --- | --- | | Sinh viên thực hiện: | Nguyễn Thị Ly | 2021607970 | |  | Nguyễn Xuân Lập | 2022602350 | |  | Nguyễn Năng Bình | 2022602699 | |  | Hoàng Anh Đăng | 2024601297 | |  |  |  | | Lớp, Khóa: | 20242IT6094010 |  | | Nhóm | 3 |  | | Giảng viên hướng dẫn**:** Ths. Mai Thanh Hồng | |  |   **Hà Nội, 2025** |

## KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ ÁN

Tên lớp : 20242IT6094010

Họ và tên sinh viên : Nguyễn Thị Ly, Nguyễn Xuân Lập, Nguyễn Năng Bình, Hoàng Anh Đăng

Tên nhóm : Nhóm 3

Tên đề tài : Nhận dạng cảm xúc khuôn mặt người bằng CNN.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Người thực hiện** | **Nội dung công việc** | **Kết quả đạt được** | **Phương pháp thực hiện** |
| 1 | Cả nhóm | Xác định đề tài nhóm | Hoàn thành | Tìm hiểu trong giáo trình, internet, youtube… |
| 2 | Cả nhóm | Tìm hiểu về Python và Google Colab | Hoàn thành | Tìm hiểu trong giáo trình,  internet,... |
| 3 | Nguyễn Thị Ly | Tìm hiểu dataset của bài toán | Hoàn thành | Tìm hiểu trong giáo trình,  internet,... |
| 4 | Nguyễn Năng Bình | Tìm hiểu về tiền xử lý dữ liệu | Hoàn thành | Tìm hiểu trong giáo trình,  internet,... |
| 5 | Nguyễn Xuân Lập  Hoàng Anh Đăng | Tìm hiểu về mạng CNN, logic mờ | Hoàn thành | Tìm hiểu trong giáo trình,  internet,... |
| 6 | Nguyễn Thị Ly  Nguyễn Xuân Lập | Viết báo cáo chương 1 và 2 | Hoàn thành | Microsoft Word |
| 7 | Nguyễn Năng Bình  Hoàng Anh Đăng | Viết báo cáo chương 3 và 4 | Hoàn thành | Microsoft Word |
|  | Cả nhóm | Hoàn thiện báo cáo | Hoàn thành | Microsoft Word |

Ngày 09 tháng 06 năm 2025

## XÁC NHẬN CỦA GIẢNG VIÊN

Mai Thanh Hồng

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_bookmark0)

[LỜI NÓI ĐẦU 2](#_bookmark1)

[PHẦN 1. GIỚI THIỆU CHUNG 3](#_bookmark2)

* 1. [Giới thiệu về nhận dạng cảm xúc khuôn mặt 3](#_bookmark3)
     1. [Sơ lược về nhận dạng cảm xúc khuôn mặt 3](#_bookmark4)
     2. [Các cảm xúc trên khuôn mặt 4](#_bookmark5)
     3. [Phương pháp nhận dạng cảm xúc 5](#_bookmark8)
  2. [Giới thiệu mạng nơron tích chập (CNN) 8](#_bookmark12)
     1. [Bộ dữ liệu FER2013 8](#_bookmark13)
     2. [Mạng nơron tích chập (Convolutional Neural Networks) 8](#_bookmark14)
  3. [Giới thiệu về Fuzzy Logic 12](#_bookmark20)
     1. [Khái niệm: 12](#_bookmark21)
     2. [Lý do sử dụng Fuzzy Logic 12](#_bookmark22)
     3. [Ứng dụng Fuzzy Logic trong nhận diện cảm xúc khuôn mặt 12](#_bookmark23)

[PHẦN 2. CƠ SỞ LÝ THUYẾT 15](#_bookmark24)

* 1. [Giới thiệu về Deep Learning 15](#_bookmark25)
     1. [Khái quát về Deep Learning 15](#_bookmark26)
     2. [Mô hình hoạt động của Deep Learning 15](#_bookmark27)
     3. [Các lớp thành phần của Deep Learning 16](#_bookmark29)
  2. [Một số mô hình học sâu (Deep Learning) 16](#_bookmark30)
     1. [Mạng nơron tái tạo (RNN) 16](#_bookmark31)
     2. [Autoencoder 16](#_bookmark32)
     3. [Mạng nơron sâu (DNN) 17](#_bookmark33)
     4. [Mạng học sâu niềm tin (DBN) 17](#_bookmark34)
     5. [Mạng nơron tích chập (CNN) 17](#_bookmark35)
  3. [Mô hình mạng nơron tích chập (CNN) trong nhận dạng cảm xúc khuôn](#_bookmark36) [mặt 18](#_bookmark36)
     1. [Khái quát mạng nơron tích chập CNN 18](#_bookmark37)
     2. [Các kiểu lớp mạng nơron tích chập CNN 20](#_bookmark40)
     3. [CNN trong nhận diện cảm xúc khuôn mặt người 25](#_bookmark49)
  4. [Giới thiệu về Fuzzy Logic 27](#_bookmark50)
     1. [Khái niệm cơ bản về Fuzzy Logic 27](#_bookmark51)
     2. [Thành phần của hệ thống Fuzzy Logic 28](#_bookmark53)
     3. [Ứng dụng của Fuzzy Logic trong nhận dạng cảm xúc khuôn mặt 29](#_bookmark55)

[PHẦN 3. THIẾT KẾ HỆ THỐNG 32](#_bookmark57)

* 1. [Tệp cơ sở dữ liệu 32](#_bookmark58)
  2. [Môi trường và công cụ sử dụng 34](#_bookmark62)
     1. [Python 34](#_bookmark63)
     2. [Thư viện Keras 35](#_bookmark64)
     3. [Thư viện Numpy 35](#_bookmark65)
     4. [Thư viện Matplotlib 36](#_bookmark66)
     5. [Google Colab 36](#_bookmark68)
     6. [Kaggle 38](#_bookmark70)
  3. [Hệ thống sử dụng mạng nơron tích chập (CNN) 39](#_bookmark71)
     1. [Phân tích dữ liệu 39](#_bookmark72)
     2. [Xử lý dữ liệu 40](#_bookmark74)
     3. [Xây dựng mô hình CNN 43](#_bookmark75)
     4. [Xây dựng mô hình Fuzzy Logic 47](#_bookmark76)

[PHẦN 4. SO SÁNH KẾT QUẢ 53](#_bookmark77)

* 1. [Tiêu chí so sánh 53](#_bookmark78)
  2. [Phân tích kết quả so sánh giữa CNN và Fuzzy Logic 54](#_bookmark79)
  3. [Đánh giá tổng quan 55](#_bookmark80)

[KẾT LUẬN 56](#_bookmark81)

[TÀI LIỆU THAM KHẢO 57](#_bookmark82)

# DANH SÁCH CÁC HÌNH ẢNH

[Hình 1. 1. Bảy cảm xúc cơ bản của con người 5](#_bookmark7)

[Hình 1. 2. Tổng quan hệ thống nhận diện cảm xúc 5](#_bookmark9)

[Hình 1. 3. Kiến trúc hệ thống nhận dạng cảm xúc khuôn mặt bằng phương pháp truyền](#_bookmark10) [thống 6](#_bookmark10)

[Hình 1. 4. Kiến trúc hệ thống nhận dạng cảm xúc khuôn mặt bằng phương pháp deep](#_bookmark11) [learning 7](#_bookmark11)

[Hình 1. 5. Các tầng (layer) trong CNN là 3 chiều 9](#_bookmark15)

[Hình 1. 6. Ví dụ minh họa về cấu trúc CNNs – LeNet- 5 9](#_bookmark16)

[Hình 1. 7. Minh họa cách thức tích chập của một ảnh RGB và ma trận kernel 10](#_bookmark17)

[Hình 1. 8. Mô phỏng quá trình tích chập trong CNN 11](#_bookmark18)

[Hình 1. 9. Minh họa kỹ thuật Pooling trong mô hình CNN 12](#_bookmark19)

[Hình 2. 1. Mô hình hoạt động của Deep Learning 15](#_bookmark28)

[Hình 2. 2. Ví dụ về xử lý đầu vào của CNN 19](#_bookmark38)

[Hình 2. 3. Ví dụ về một mô hình CNN 19](#_bookmark39)

[Hình 2. 4. Hoạt động của lớp tích chập 20](#_bookmark41)

[Hình 2. 5. Ví dụ về một ma trận hình ảnh và một ma trận bộ lọc 20](#_bookmark42)

[Hình 2. 6. Cách hoạt động của ma trận bộ lọc với ma trận hình ảnh 21](#_bookmark43)

[Hình 2. 7. Xử lý hình ảnh với các Kernel khác nhau 22](#_bookmark44)

[Hình 2. 8. Xử lý ảnh với Max-pooling 23](#_bookmark45)

[Hình 2. 9. Xử lý ảnh với Average-Pooling 24](#_bookmark46)

[Hình 2. 10. Xử lý ảnh với Flatten 25](#_bookmark47)

[Hình 2. 11. Mô hình mạng CNN hoàn chỉnh 25](#_bookmark48)

[Hình 2. 12. Fuzzy logic là gì? 27](#_bookmark52)

[Hình 2. 13. Thành phần của hệ thống Fuzzy Logic 28](#_bookmark54)

[Hình 2. 14. Ví dụ về dùng Fuzzy Logic nhận diện cảm xúc 30](#_bookmark56)

[Hình 3. 1. Một số hình ảnh trong bộ dữ liệu FER-2013 33](#_bookmark59)

[Hình 3. 2. Số lượng ảnh của từng nhãn cảm xúc trong bộ dữ liệu FER-2013 33](#_bookmark60)

[Hình 3. 3. Một số hình ảnh trong phải khuôn mặt người trong bộ dữ liệu FER-2013 .34](#_bookmark61) [Hình 3. 4. Một số hình ảnh về biểu đồ của thư viện Matplotlib 36](#_bookmark67)

[Hình 3. 5. Giao diện Google Colab 37](#_bookmark69)

[Hình 3. 6. Biểu đổ thể hiện số lượng hình ảnh của từng nhãn và từng tập dữ liệu 39](#_bookmark73)

# DANH SÁCH CÁC BẢNG BIỂU

[Bảng 1.1. Biểu cảm khuôn mặt 4](#_bookmark6)

## LỜI CẢM ƠN

Lời đầu tiên, nhóm 3 chúng em xin gửi lời cảm ơn chân thành tới cô Mai Thanh Hồng vì đã tận tình hướng dẫn và chia sẻ những kiến thức quý báu trong suốt quá trình thực hiện bài tập lớn với chủ đề "Nhận diện cảm xúc khuôn mặt người bằng CNN".

Nhờ sự hướng dẫn chi tiết của cô, chúng em không chỉ hiểu rõ hơn về cách áp dụng mạng nơ-ron tích chập (CNN) vào bài toán nhận diện cảm xúc mà còn học được cách giải quyết vấn đề thực tiễn thông qua các phương pháp hiện đại trong lĩnh vực trí tuệ nhân tạo. Những gợi ý và phản hồi kịp thời của cô đã giúp chúng em cải thiện bài làm, nâng cao kỹ năng nghiên cứu cũng như khả năng triển khai thực tế.

Chúng em xin gửi đến cô lời cảm ơn chân thành nhất và hy vọng sẽ tiếp tục nhận được sự chỉ bảo từ cô trong những chặng đường học tập tiếp theo.

Chúng em xin chân thành cảm ơn!

**Nhóm sinh viên thực hiện Nhóm 3**

## LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, trí tuệ nhân tạo (AI) đã và đang đóng vai trò quan trọng trong nhiều lĩnh vực của cuộc sống, từ y tế, giáo dục đến các ứng dụng thực tiễn như nhận diện cảm xúc, một yếu tố quan trọng trong giao tiếp giữa con người với con người. Bài toán nhận diện cảm xúc khuôn mặt không chỉ giúp cải thiện trải nghiệm người dùng trong các ứng dụng thông minh mà còn mở ra nhiều cơ hội nghiên cứu mới trong lĩnh vực tương tác người - máy.

Dựa trên sức mạnh của mạng nơ-ron tích chập (Convolutional Neural Networks

- CNN), một trong những phương pháp hiệu quả nhất trong xử lý hình ảnh, nhóm chúng em đã chọn đề tài "Nhận diện cảm xúc khuôn mặt bằng CNN" làm chủ đề nghiên cứu trong bài tập lớn này. Báo cáo tập trung vào việc tìm hiểu, xây dựng mô hình CNN, đồng thời triển khai thử nghiệm trên các tập dữ liệu phổ biến để đánh giá hiệu quả của phương pháp.

Quá trình thực hiện và hoàn thiện báo cáo là kết quả của sự cố gắng, nỗ lực nghiên cứu, thảo luận và áp dụng kiến thức học được từ lớp học, các nguồn tài liệu truy xuất, cũng như những trải nghiệm cá nhân của từng thành viên trong nhóm. Chúng em hy vọng bài báo cáo này sẽ góp một phần nhỏ bé vào việc tìm hiểu và ứng dụng AI trong cuộc sống. Chúng em rất mong nhận được những ý kiến đóng góp từ thầy cô và bạn đọc, để báo cáo này càng trở nên hoàn thiện hơn.

*Nhóm chúng em xin chân thành cảm ơn!*

## PHẦN 1. GIỚI THIỆU CHUNG

### Giới thiệu về nhận dạng cảm xúc khuôn mặt

### Sơ lược về nhận dạng cảm xúc khuôn mặt

Phân loại và nhận diện cảm xúc là lĩnh vực đã được nghiên cứu trong nhiều năm qua với nhiều ứng dụng trong nhiều lĩnh vực khác nhau gắn liền với các hệ thống tương tác người máy. Trong máy học, phân loại biểu cảm là một bài toán khó, tuy nhiên, đối với con người, vấn đề này có thể giải quyết ngay lập tức. Các thách thức chính là: hình ảnh biểu cảm của cùng một người ở cùng một biểu cảm vẫn có thể khác nhau ở những điều kiện ánh sáng, môi trường và góc quay. Những biến đổi này càng lớn khi các đối tượng nghiên cứu càng đa dạng.

Khuôn mặt của con người biểu hiện nhiều cảm xúc mà không cần phải nói ra. Đó là một trong những phương tiện mạnh mẽ và tự nhiên nhất để con người truyền đạt thể hiện cảm xúc. Không giống như các hình thức giao tiếp phi ngôn ngữ khác, cảm xúc trên khuôn mặt nó 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, bao gồm 93% cảm xúc giao tiếp của con người, trong đó 55% thể hiện cử chỉ khuôn mặt và hành động của con người.

Cảm xúc khuôn mặt có thể được phân tích dễ dàng thông qua hình ảnh khuôn mặt 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. Thêm cảm xúc trung tính là có bảy cảm xúc cơ bản, gồm: trung tính, giận dữ, ghê tởm, sợ hãi, hạnh phúc, buồn và bất ngờ. Nhận dạng cảm xúc qua khuôn mặt có ứng dụng trong các lĩnh vực khác nhau

### Các cảm xúc trên khuôn mặt

Nhận biết cảm xúc từ nét mặt có một số lợi thế như:

1. Cận theo hướng tự nhiên nhất để xác định trạng thái cảm xúc của khuôn mặt.
2. Nhiều bộ dữ liệu có sẵn cho biểu hiện cảm xúc trên khuôn mặt.
3. Nhiều công cụ hỗ trợ xác định cảm xúc khuôn mặt có sẵn. Nhận biết cảm xúc từ nét mặt cũng có một số nhược điểm như:
   1. Không thể cung cấp thông tin ngữ cảnh, do đó đôi khi kết quả bị sai lệch.
   2. Kết quả phát hiện cảm xúc phụ thuộc vào chất lượng hình ảnh hoặc video.
   3. Chuyển động liên quan đến cảm xúc khuôn mặt có thể được đối tượng cố tình làm giả như các diễn viên …

Bảng dưới đây cho biết biểu cảm trên khuôn mặt thể hiện 7 cảm xúc chính của con người:



*Bảng 1.1. Biểu cảm khuôn mặt*



*Hình 1. 1. Bảy cảm xúc cơ bản của con người*

### Phương pháp nhận dạng cảm xúc

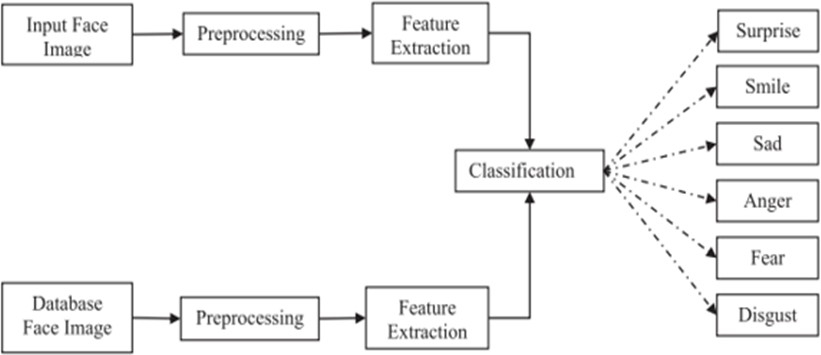
Nhận dạng cảm xúc trên khuôn mặt (Facial Emotion Recognition) là một công nghệ được sử dụng để phân tích cảm xúc theo các nguồn khác nhau, chẳng hạn như hình ảnh và video. Bài toán này thuộc về nhóm công nghệ gọi là “tính toán cảm xúc”, một lĩnh vực nghiên cứu đa ngành về khả năng của máy tính để nhận biết và giải thích cảm xúc và trạng thái tình cảm của con người và nó thường được xây dựng dựa trên công nghệ Trí tuệ nhân tạo.



*Hình 1. 2. Tổng quan hệ thống nhận diện cảm xúc*

#### Phương pháp truyền thống

Hệ thống nhận dạng cảm xúc qua khuôn mặt với phương pháp truyền thống thì xử lý bài qua các giai đoạn: tiền xử lý hình ảnh khuôn mặt, trích xuất đặc trưng và phân loại.



*Hình 1. 3. Kiến trúc hệ thống nhận dạng cảm xúc khuôn mặt bằng phương pháp*

*truyền thống*

Tiền xử lý là quá trình được sử dụng để cải thiện hiệu suất của hệ thống nhận dạng cảm xúc qua khuôn mặt và được thực hiện các loại quy trình khác nhau: căn chỉnh độ rõ, chia tỷ lệ hình ảnh, điều chỉnh độ tương phản và sử dụng các quy trình nâng cao để cải thiện các khung biểu thức.

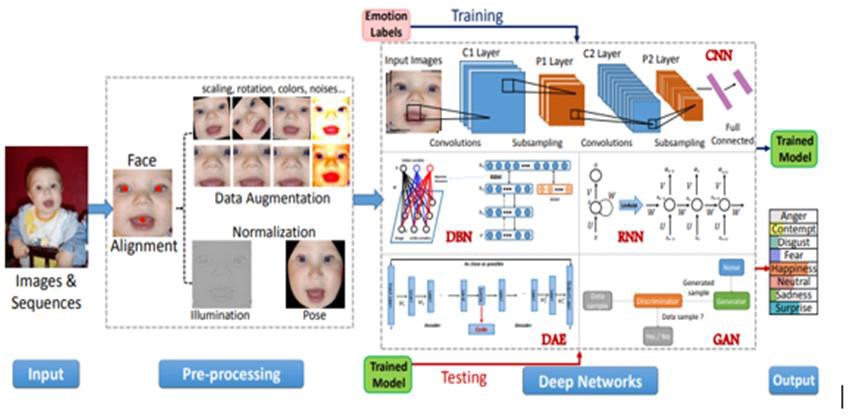
Trích xuất đặc trưng trong thị giác máy tính là một giai đoạn quan trọng, nó phát hiện ra việc chuyển từ mô tả đồ họa sang mô tả dữ liệu ẩn, trích chọn những đặc trưng riêng nhất của hình ảnh, sau đó những mô tả dữ liệu này có thể được sử dụng làm đầu vào cho bài toán phân loại.

Phân loại là giai đoạn cuối cùng của hệ thống nhận diện cảm xúc qua khuôn mặt (FER), để phân loại ra các loại cảm xúc trên khuôn mặt: hạnh phúc, buồn bã, bất ngờ, tức giận, sợ hãi, ghê tởm và bình thường. Sử dụng các phương pháp phân loại như: Cây quyết định (ID3), SVM, HMM (Hidden Markov Model) ... thì phân loại SVM cho độ

chính xác và phân loại tốt nhất. Chính vì vậy, mình chọn SVM đại diện cho phương pháp truyền thống để sử dụng cho hệ thống nhận diện của mình.

#### 1.1.3.2 Phương pháp hiện đại

Trong phần này, sẽ mô tả các bước chính phổ biến trong hệ thống nhận dạng cảm xúc qua khuôn mặt thực hiện qua các giai đoạn: tiền xử lý, phân lớp sử dụng học sâu. Những năm gần đây, học sâu có độ chính xác hơn phương pháp truyền thống vì nó không phải qua bước trích xuất các đặc trưng một cách tường minh, nó sẽ thực hiện đi kèm với phương pháp phân loại.



*Hình 1. 4. Kiến trúc hệ thống nhận dạng cảm xúc khuôn mặt bằng phương pháp deep learning*

Tiền xử lý ảnh: cũng phải xử lý 1 số vấn đề của ảnh đầu vào hệ thống, xử lý trước quá trình training. Các bước thực hiện: Căn chỉnh khuôn mặt để phát hiện khuôn mặt, tăng dữ liệu hình ảnh đảm bảo đủ dữ liệu training, cuối cùng là chuẩn hóa dữ liệu khuôn mặt. Sử dụng các phương pháp CNN, DBN, DAE, RNN, GAN...

Phân loại: Trong phương pháp truyền thống bước trích xuất đặc trưng và bước phân loại tính năng là độc lập với nhau, trong Deep learning có thể thực hiện FER theo cách từ đầu đến cuối. Một lớp mất được thêm vào cuối mạng để điều chỉnh lỗi lan truyền ngược, sau đó xác suất dự đoán của từng mẫu có thể được mạng trực tiếp xuất ra.

### Giới thiệu mạng nơron tích chập (CNN)

### Bộ dữ liệu FER2013

Dữ liệu FER-2013 được công bố bởi trang Kaggle trong khuôn khổ workshop của hội thảo ICML 2013. Dữ liệu gồm các ảnh đa cấp xám cỡ 48x48 chỉ gồm khuôn mặt hầu như được căn giữa ảnh và tỉ lệ khuôn mặt được điều chỉnh chiếm phần lớn diện tích của ảnh. Một ảnh sẽ được gán nhãn nằm một trong bảy loại cảm xúc giá trị từ 0 đến 6 (0: giận dữ, 1: căm phẫn, 2: sợ hãi, 3: hạnh phúc, 4: buồn rầu, 5: bất ngờ, 6: trung lập).

Bộ dữ liệu này gồm 28.709 mẫu huấn luyện, mẫu kiểm tra công khai có 3.589 ảnh. Khi thực hiện đánh giá mô hình, Kaggle sẽ sử dụng một bộ kiểm tra khác cũng có

3.589 ảnh, vì vậy kết quả đánh giá của ban giám khảo có thể có sai lệch so với sử dụng bộ test công khai, một số trường hợp đặc biệt sai lệch có thể lên đến 5%.

Chúng tôi sử dụng bộ dữ liệu này cho mô hình thử nghiệm vì bộ dữ liệu có số mẫu khá lớn, phù hợp với việc huấn luyện với mạng học sâu, vốn đòi hỏi nhiều mẫu hơn các phương pháp học máy thông thường. Ngoài ra, bộ dữ liệu được cấu trúc dễ dàng xử lý bởi thư viện Keras/TensorFlow và có nhiều kết quả đối chứng khi thực hiện so sánh mô hình của chúng tôi với các kết quả của những nhóm nghiên cứu khác.

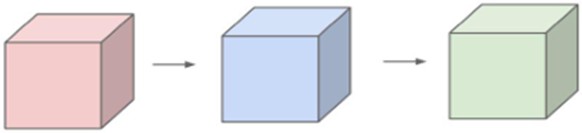
Cảm xúc của con người là muôn hình vạn trạng. Mỗi người điều có mỗi cách bày tỏ cảm xúc của mình khác nhau. Chính vì thế, dữ liệu cần để huấn luyện cho mô hình nhận diện cảm xúc khuôn mặt người là rất lớn. Trong đề tài này, tập dữ liệu FER2013 được lựa chọn sử dụng.

#### Mạng nơron tích chập (Convolutional Neural Networks)

* + - 1. ***Khái niệm:***

Mạng nơ ron tích chập là một trong những mạng truyền thẳng đặc biệt. Mạng nơ ron tích chập là một mô hình học sâu phổ biến và tiên tiến nhất hiện nay. Hầu hết các hệ thống nhận diện và xử lý ảnh hiện nay đều sử dụng mạng nơ ron tích chập vì tốc độ xử lý nhanh và độ chính xác cao. Trong mạng nơ ron truyền thống, các tầng được coi là một chiều, thì trong mạng nơ ron tích chập, các tầng được coi là 3 chiều, gồm: chiều cao, chiều rộng và chiều sâu. Mạng nơ ron tích chập có hai khái niệm quan trọng: kết

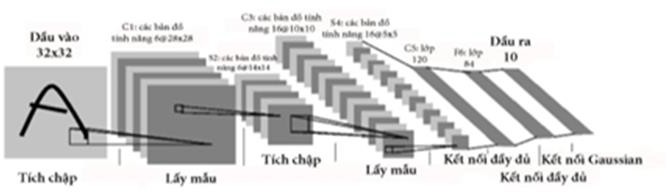
nối cục bộ và chia sẻ tham số. Những khái niệm này góp phần giảm số lượng trọng số cần được huấn luyện, do đó tăng nhanh được tốc độ tính toán.



*Hình 1. 5. Các tầng (layer) trong CNN là 3 chiều*

#### Mô hình mạng nơron tích chập

Một kiến trúc CNN bao gồm các lớp: convolution layer, pooling layer và fully connected layer. Ở giữa các lớp convolution và pooling thường có các hàm kích hoạt phi tuyến. Ảnh khi đưa vào mạng sẽ được lan truyền qua tầng convolution layer, giá trị tính được từ các tầng convolution sẽ đi qua một hàm kích hoạt, sau đó giá trị này sẽ được lan truyền qua pooling layer. Cuối cùng ảnh sẽ được lan truyền đến tầng fully connected layer và đi qua hàm kích hoạt Softmax, thường thì cuối cùng sẽ thu được một vector chứa xác suất phần trăm thuộc về các lớp đối với các bài toán phân loại. Ví dụ minh họa về một kiến trúc mạng nơ ron tích chập đầy đủ:



*Hình 1. 6. Ví dụ minh họa về cấu trúc CNNs – LeNet- 5*

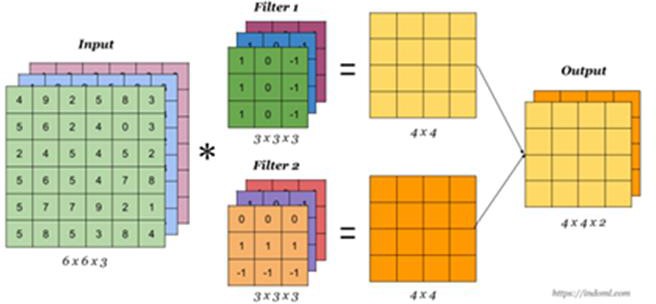
− Convolution layer

Convolution layer là lớp quan trọng nhất và cũng là lớp đầu tiên của của mô hình CNN. Lớp này có chức năng chính là phát hiện các đặc trưng có tính không gian hiệu

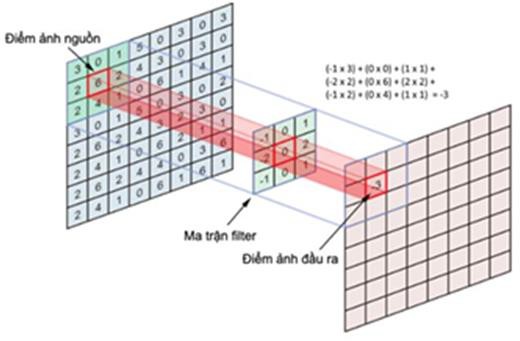
quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, bộ filters, và receptive field, feature map. Conv layer nhận đầu vào là một ma trận 3 chiều và một bộ filters cần phải học. Bộ filters này sẽ trượt qua từng vị trí trên bức ảnh để tính tích chập (convolution) giữa bộ filter và phần tương ứng trên bức ảnh. Phần tương ứng này trên bức ảnh gọi là receptive field, tức là vùng mà một neuron có thể nhìn thấy để đưa ra quyết định, và ma trận cho ra bởi quá trình này được gọi là feature map.

Khi đưa ảnh vào mạng, bộ filter sẽ quét qua toàn bộ ảnh cho nên các đặc trưng cơ bản của ảnh như là góc, cạnh, màu sắc và texture sẽ được mạng phát hiện ra bất kể nó nằm ở vị trí nào trong ảnh. Do đó tầng convolution được xem như là một bộ 11 trích chọn đặc trưng (feature detector) vì nó có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh đầu vào.

Tích chập sẽ trích xuất đặc trưng của ảnh đầu vào qua các vùng ảnh nhỏ. Các vùng này được gọi là Local Receptive Field (LRF). Tích chập sẽ tính toán trên các LRF chồng lấp lên nhau. Độ chồng lắp này phụ thuộc vào hệ số trượt S (stride) của từng kiến trúc mạng cụ thể. Nếu sử dụng với hệ số trượt S = α, thì tương ứng LRF (bằng kích thước với kernel) sẽ dịch chuyển α đơn vị pixel sau mỗi lần tích chập.



*Hình 1. 7. Minh họa cách thức tích chập của một ảnh RGB và ma trận kernel*



*Hình 1. 8. Mô phỏng quá trình tích chập trong CNN*

− Các tham số của lớp tích chập

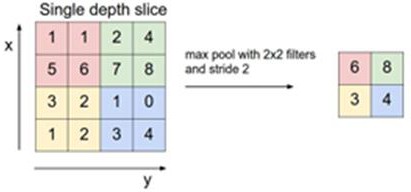
Các tham số cơ bản của tầng convolution chính là kích thước filter, stride và padding. Trong đó quan trọng nhất chính là kích thước bộ filter, vì nó tỉ lệ thuận với số tham số cần học tại mỗi tầng convolution và là tham số quyết định receptive field của tầng này. Kích thước filter phổ biến thường dùng là 3x3.

− Hàm kích hoạt

Hàm kích hoạt là một hàm số nhận vào một giá trị đầu vào và kết quả là một giá trị có miền giá trị nằm trên một khoảng (hay nửa khoảng) nào đó. Hàm kích hoạt rất quan trọng bởi vì nó sẽ tăng khả năng dự đoán của mạng neural và giúp mô hình học được các quan hệ phi tuyến phức tạp tiềm ẩn trong dữ liệu. Thông thường hàm kích hoạt sử dụng ở giữa các tầng convolution và pooling là hàm Relu.

− Lớp Pooling

Pooling giúp cho mạng giảm số lượng tham số, từ đó giúp đơn giản hóa quá trình tính toán của CNN và qua đó góp phần giải quyết vấn đề overfitting khi huấn luyện mạng.



*Hình 1. 9. Minh họa kỹ thuật Pooling trong mô hình CNN*

### Giới thiệu về Fuzzy Logic

### Khái niệm:

Fuzzy Logic (Logic mờ) là một kỹ thuật phổ biến trong các hệ thống xử lý thông tin không rõ ràng hoặc không chắc chắn, rất phù hợp với bài toán nhận diện cảm xúc khuôn mặt, nơi các trạng thái cảm xúc thường không hoàn toàn rõ ràng hoặc dễ phân loại. Thay vì chỉ sử dụng các nhãn cảm xúc cố định (như "vui", "buồn", "giận dữ"), Fuzzy Logic cho phép biểu diễn cảm xúc một cách linh hoạt hơn bằng cách mô hình hóa sự mờ nhòe giữa các trạng thái.

### Lý do sử dụng Fuzzy Logic

**Tính không chắc chắn trong cảm xúc**:

− Cảm xúc không phải lúc nào cũng phân biệt rõ ràng; ví dụ, một khuôn mặt có thể thể hiện sự pha trộn giữa "buồn" và "lo lắng".

− Fuzzy Logic giúp giải quyết bài toán này bằng cách gán một mức độ thuộc về (membership degree) cho mỗi cảm xúc thay vì chỉ định cảm xúc duy nhất.

### Tính linh hoạt:

− Fuzzy Logic có thể tích hợp các yếu tố cảm xúc không tuyến tính hoặc các đặc điểm phức tạp mà mô hình cứng nhắc (như các phương pháp phân loại truyền thống) khó xử lý.

### Ứng dụng Fuzzy Logic trong nhận diện cảm xúc khuôn mặt

**Xác định đặc trưng khuôn mặt:**

− Các đặc trưng như độ cong môi, độ nhướng mày, khoảng cách giữa mắt và miệng được trích xuất từ hình ảnh khuôn mặt.

độ 0.5.

− Những đặc trưng này được biểu diễn dưới dạng các giá trị mờ (fuzzy values), ví dụ: "hơi cong", "rất cong", "không cong".

### Xây dựng hệ thống suy diễn mờ:

− Một tập các **quy tắc mờ** (fuzzy rules) được xây dựng để liên kết các đặc trưng khuôn mặt với các cảm xúc.

*Ví dụ:*

Nếu *miệng rất cong* và *mắt mở to*, thì cảm xúc là "vui" với mức độ 0.8.

Nếu *miệng hơi cong* và *lông mày nhướng nhẹ*, thì cảm xúc là "hài lòng" với mức

### Kết hợp với CNN:

* 1. CNN được sử dụng để trích xuất các đặc trưng từ hình ảnh khuôn mặt một cách tự động và chính xác.
  2. Các đặc trưng này sau đó được chuyển đổi thành các giá trị mờ làm đầu vào cho hệ suy diễn mờ.
  3. Kết quả cuối cùng là mức độ thuộc về của từng cảm xúc, cho phép hệ thống xác định các cảm xúc chính và phụ.

### Xử lý đầu ra mờ:

* 1. Đầu ra từ hệ thống Fuzzy Logic thường là một tập các mức độ thuộc về cho từng cảm xúc (ví dụ: "vui: 0.7", "ngạc nhiên: 0.2").
  2. Có thể sử dụng phương pháp giải mờ (defuzzification) để chọn cảm xúc chính, hoặc trực tiếp hiển thị các mức độ cảm xúc để phân tích sâu hơn.
     1. **Ưu điểm của việc sử dụng Fuzzy Logic**
        1. **Mô hình hóa cảm xúc phức tạp:** Phù hợp với các trạng thái cảm xúc phức hợp hoặc không rõ ràng.
        2. **Tăng cường tính tương tác:** Cho phép phản hồi tinh tế hơn, đặc biệt trong các ứng dụng như robot xã hội, chăm sóc sức khỏe, và giáo dục.
        3. **Kết hợp linh hoạt:** Có thể tích hợp với các mô hình học sâu như CNN để cải thiện hiệu quả.
     2. **Thách thức khi áp dụng Fuzzy Logic**
        1. **Tốn thời gian thiết kế hệ thống:** Việc xây dựng tập quy tắc mờ đòi hỏi kiến thức chuyên môn và kinh nghiệm.
        2. **Khả năng mở rộng hạn chế:** Đối với các tập dữ liệu lớn và phức tạp, hệ thống Fuzzy Logic cần được tối ưu hóa để xử lý hiệu quả.
        3. **Tính toán phức tạp:** Khi số lượng đặc trưng và quy tắc tăng, hệ thống có thể trở nên khó kiểm soát.
     3. **Ví dụ thực tiễn**
        1. **Ứng dụng trong robot giao tiếp:** Fuzzy Logic giúp robot xác định cảm xúc của người dùng dựa trên các biểu hiện khuôn mặt phức tạp và phản hồi phù hợp.
        2. **Phân tích tâm lý học:** Hỗ trợ phân tích các trạng thái cảm xúc phức tạp của bệnh nhân dựa trên biểu hiện khuôn mặt.

Fuzzy Logic kết hợp với các công nghệ hiện đại như CNN không chỉ tăng độ chính xác mà còn mở ra nhiều cơ hội phát triển các hệ thống nhận diện cảm xúc thân thiện và hiệu quả hơn.

## PHẦN 2. CƠ SỞ LÝ THUYẾT

### Giới thiệu về Deep Learning

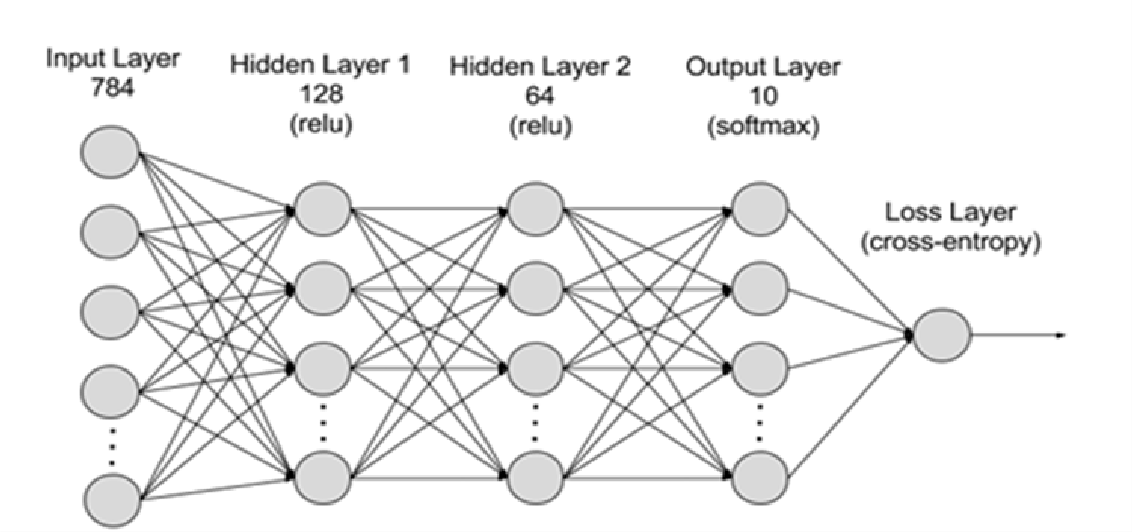
### Khái quát về Deep Learning

Deep learning (Học sâu) là một phương thức trong lĩnh vực [trí tuệ nhân tạo (AI)](https://aws.amazon.com/what-is/artificial-intelligence/), được sử dụng để dạy máy tính xử lý dữ liệu theo cách được lấy cảm hứng từ bộ não con người. Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Có thể sử dụng các phương pháp học sâu để tự động hóa các tác vụ thường đòi hỏi trí tuệ con người, chẳng hạn như mô tả hình ảnh hoặc chép lời một tập tin âm thanh.

Deep Learning tập trung vào việc xây dựng và đào tạo các mô hình máy học có khả năng học từ dữ liệu phức tạp. Deep Learning sử dụng các mạng nơ-ron sâu với nhiều lớp để tự động học và biểu diễn các đặc trưng phức tạp của dữ liệu.

### Mô hình hoạt động của Deep Learning

Deep Learning hoạt động theo cách mô phỏng cách não bộ con người hoạt động trong việc xử lý thông tin. Các chương trình máy tính sử dụng Deep Learning sẽ trải qua quá trình học tương tự như quá trình học của một đứa trẻ. Mô hình sử dụng nhiều lớp nơron để biểu diễn các mức độ trừu tượng khác nhau của dữ liệu. Các lớp này bao gồm lớp đầu vào, các lớp ẩn (hidden layers), và lớp đầu ra. Trong quá trình đào tạo, mô hình điều chỉnh trọng số để tối ưu hóa khả năng dự đoán hoặc phân loại của nó. Các vòng lặp tiếp tục cho đến khi đầu ra đạt đến một mức độ chính xác chấp nhận được.



*Hình 2. 1. Mô hình hoạt động của Deep Learning*

### Các lớp thành phần của Deep Learning

Các mô hình Deep Learning thường bao gồm các thành phần chính như sau:

− Lớp đầu vào (Input Layer): Là lớp là lớp đầu tiên trong mô hình học sâu, nhận dữ liệu đầu vào và chuyển nó vào mạng. Lớp này có thể là ảnh, văn bản, âm thanh hoặc bất kỳ dữ liệu nào phù hợp với bài toán cụ thể.

− Lớp ẩn (Hidden Layers): Là các lớp nơron nằm giữa lớp đầu vào và lớp đầu ra. Các mạng học sâu có hàng trăm lớp ẩn có thể được dùng để phân tích một vấn đề từ nhiều góc độ khác nhau. Chúng thực hiện các phép biến đổi phức tạp để học các đặc trưng của dữ liệu.

− Lớp đầu ra (Output Layer):Là lớp cuối cùng trong mô hình học sâu, bao gồm các nút xuất dữ liệu, đưa ra dự đoán hoặc kết quả của mô hình.

### Một số mô hình học sâu (Deep Learning)

### Mạng nơron tái tạo (RNN)

Mạng nơron tái tạo (Recurrent neural networks - RNN) mở rộng khả năng của mạng nơron truyền thống và được thiết kế để lập mô hình dữ liệu dạng tuần tự. RNN được tạo thành từ các nơron: các nút xử lý dữ liệu kết hợp cùng nhau để thực hiện các tác vụ phức tạp. Các nơron được tổ chức dưới dạng lớp đầu vào, đầu ra và ẩn. Lớp đầu vào nhận thông tin để xử lý và lớp đầu ra cung cấp kết quả. Quá trình xử lý dữ liệu, phân tích và dự đoán diễn ra trong lớp ẩn. Để đưa ra quyết định, mỗi đơn vị RNN dựa vào đầu vào hiện tại của nó và đầu ra của đầu vào trước đó. RNN được sử dụng rộng rãi trong các lĩnh vực khác nhau như xử lý giọng nói, nhận dạng hoạt động của con người, dự đoán chữ viết tay và hiểu ngữ nghĩa. Đối với hệ thống IDS, RNN có thể được sử dụng cho bài toán phân lớp có giám sát hoặc trích xuất đặc trưng. IDS dựa trên RNN được một số nghiên cứu đề xuất để phân loại nhị phân và phân lớp đa lớp trên tập dữ liệu NSL-KDD. Mô hình đã được thử nghiệm bằng cách sử dụng một số nút ẩn và tốc độ học khác nhau. Kết quả cho thấy rằng tốc độ học và số lượng các nút ẩn ảnh hưởng đến độ chính xác của mô hình.

### Autoencoder

Autoencoder là một mạng nơ-ron nhân tạo (ANN) có khả năng tự học các biểu diễn hiệu quả của dữ liệu đầu vào mà không cần nhãn. Những biểu diễn này thường có chiều nhỏ hơn đáng kể so với dữ liệu gốc, giúp autoencoder trở thành công cụ hữu ích

trong các bài toán giảm chiều dữ liệu. Ngoài ra, autoencoder còn được sử dụng để phát hiện và trích xuất các đặc trưng tiềm ẩn, hỗ trợ việc huấn luyện các mô hình trong các bài toán khác.

Một số nghiên cứu đã đề xuất sử dụng autoencoder cải tiến, cụ thể là mô hình stacked sparse autoencoder (SSAE) kết hợp với máy vector hỗ trợ (SVM) để xây dựng hệ thống phát hiện xâm nhập (IDS). Trong đó, SSAE được dùng để trích xuất đặc trưng, còn SVM đảm nhận vai trò phân loại. Kết quả thực nghiệm trên tập dữ liệu NSL-KDD cho thấy phương pháp này đạt hiệu suất phân loại vượt trội so với các kỹ thuật trích chọn đặc trưng và các thuật toán học máy khác.

### Mạng nơron sâu (DNN)

Mạng nơ-ron sâu (Deep Neural Network - DNN) là một dạng mạng nơ-ron nhân tạo nâng cao, thuộc lĩnh vực học sâu, với cấu trúc "sâu" và phức tạp hơn đáng kể so với mạng nơ-ron truyền thống. Điểm khác biệt chính của DNN là số lượng lớn các nút trong mỗi lớp và số lượng lớp ẩn nhiều hơn, cho phép mô hình hóa các mối quan hệ phi tuyến tính và biểu diễn dữ liệu tốt hơn.

Nhiều nghiên cứu đã chứng minh hiệu quả vượt trội của mạng nơ-ron sâu trong việc phát hiện xâm nhập mạng (IDS). Các thử nghiệm trên các tập dữ liệu như KDDCup 99, NSL-KDD, Kyoto, UNSW-NB15, WSN-DS, và CICIDS 2017 cho thấy DNN đạt

được độ chính xác cao hơn so với các phương pháp truyền thống, khẳng định tiềm năng của nó trong lĩnh vực an ninh mạng.

### Mạng học sâu niềm tin (DBN)

Mạng học sâu niềm tin (Deep belief net-DBN) là một mô hình mạng nơ-ron nhân tạo nhiều lớp. Quá trình huấn luyện mạng DBN gồm hai pha: tiền huấn luyện và hiệu chỉnh trọng số. Trong pha tiền huấn luyện, máy học Boltzmann được sử dụng để khởi tạo trọng số tốt nhất cho mô hình với dữ liệu không cần được gán nhãn. Trong pha hiệu chỉnh trọng số, DBN tiếp tục được huấn luyện bằng phương pháp lan truyền ngược cổ điển với dữ liệu được gán nhãn. Đối với IDS, DBN được sử dụng cho các nhiệm vụ trích xuất đặc trưng và phân lớp.

### Mạng nơron tích chập (CNN)

Mạng nơ-ron tích chập (Convolutional neural network - CNN) là một dạng cụ thể của mạng nơ ron sâu. Mạng nơ ron tích chập có một lớp vào, một lớp ra và nhiều

lớp ẩn khác nhau. Các lớp ẩn gồm các loại như: lớp tích chập, lớp giảm kích thước, lớp sửa dữ liệu, lớp chuẩn hóa, lớp kết nối đầy đủ... Trong đó, lớp tích chập được sử dụng nhằm tạo liên kết giữa các lớp liền kề trong phạm vi nhỏ, giới hạn trong một vùng cục bộ. Điều này giúp giảm đáng kể các việc tính toán các hàm truyền giữa các lớp mà vẫn duy trì được mối liên hệ giữa các nơron để trích xuất đặc trưng của dữ liệu ở các lớp sau đó của mạng. Một số nghiên cứu đề xuất một IDS hiệu quả dựa trên CNN. Ý tưởng là đầu tiên thực hiện trích xuất đặc trưng bằng cách sử dụng kỹ thuật phân tích thành phần và autoencoder. Sau đó biến đổi vectơ một chiều (tập đặc trưng) thành ma trận hai chiều và đưa vào mạng nơ ron tích chập. Các thử nghiệm được thực hiện trên tập dữ liệu KDD Cup'99, cho thấy hiệu quả của nó về thời gian thực hiện bởi các thuật toán trong giai đoạn huấn luyện và kiểm tra.

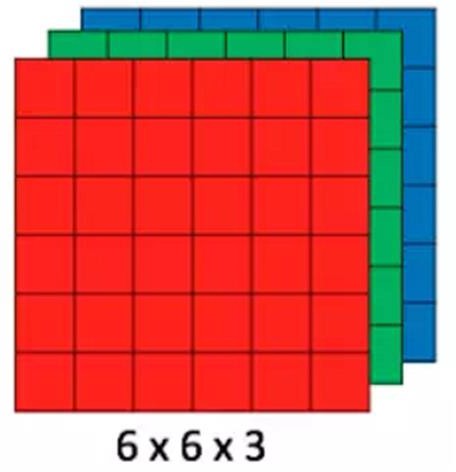
### Mô hình mạng nơron tích chập (CNN) trong nhận dạng cảm xúc khuôn mặt

### Khái quát mạng nơron tích chập CNN

Convolutional Neural Network (CNN – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao, nhờ khả năng nhận diện và phân tích các đặc điểm quan trọng trong hình ảnh.

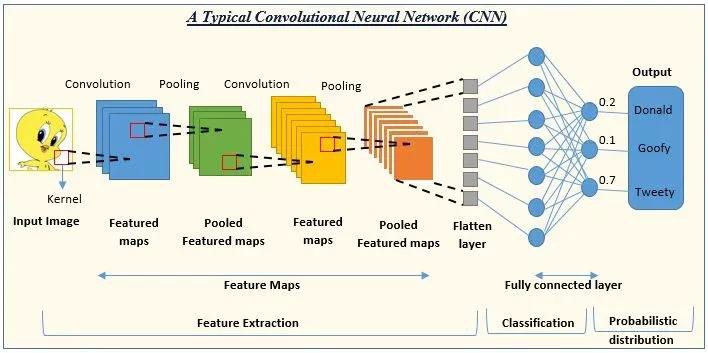
CNN hoạt động bằng các xử lý dữ liệu hình ảnh thông qua các lớp tích chập, giúp trích xuất các đặc điểm nổi bật từ hình ảnh một các hiệu quả.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).



*Hình 2. 2. Ví dụ về xử lý đầu vào của CNN*

CNN hoạt động bằng cách đưa ảnh đầu vào qua một chuỗi các bước: lớp tích chập phát hiện đặc trưng cơ bản như cạnh và góc, lớp kích hoạt giúp tăng cường các giá trị quan trọng, và lớp gộp giảm kích thước dữ liệu trong khi giữ lại thông tin cốt lõi. Quy trình này được lặp đi lặp lại để học các đặc trưng phức tạp hơn. Cuối cùng, các đặc trưng được tổng hợp tại các lớp kết nối đầy đủ và chuyển qua hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luông CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.



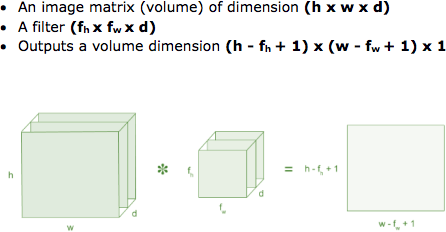
*Hình 2. 3. Ví dụ về một mô hình CNN*

### Các kiểu lớp mạng nơron tích chập CNN

Các kiểu lớp mạng nơ-ron tích chập (CNN) chủ yếu bao gồm lớp tích chập (Convolutional Layer), lớp rút gọn (Pooling Layer), và lớp kết nối đầy đủ (Fully Connected Layer). Dưới đây là mô tả chi tiết về mỗi kiểu lớp này:

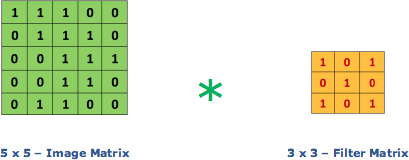
#### Lớp tích chập (Convolutional layer)

Lớp tích chập là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Phép toán tích chập bao gồm một bộ lọc (hoặc hạt nhân) trượt qua dữ liệu đầu vào, thực hiện phép nhân từng phần tử và cộng các kết quả để tạo ra bản đồ đặc điểm.



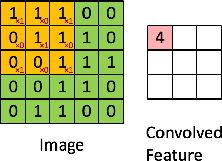
*Hình 2. 4. Hoạt động của lớp tích chập*

Xem xét 1 ma trận 5 x 5 có giá trị pixel là 0 và 1. Ma trận bộ lọc 3 x 3 như hình bên dưới.



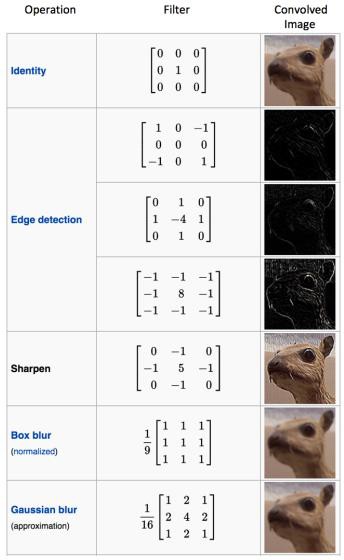
*Hình 2. 5. Ví dụ về một ma trận hình ảnh và một ma trận bộ lọc*

Sau đó, lớp tích chập của ma trận hình ảnh 5 x 5 nhân với ma trận bộ lọc 3 x 3 gọi là 'Feature Map' như hình bên dưới.



*Hình 2. 6. Cách hoạt động của ma trận bộ lọc với ma trận hình ảnh*

Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể thực hiện các hoạt động như phát hiện cạnh, làm mờ và làm sắc nét bằng cách áp dụng các bộ lọc. Ví dụ dưới đây cho thấy hình ảnh tích chập khác nhau sau khi áp dụng các Kernel khác nhau.



*Hình 2. 7. Xử lý hình ảnh với các Kernel khác nhau*

Các thành phần chính của một lớp tích chập:

− Bộ lọc (Kernels): Là các ma trận nhỏ, có thể học được, trích xuất các tính năng cụ thể từ dữ liệu đầu vào. Ví dụ, một bộ lọc có thể phát hiện các cạnh ngang, trong khi một bộ lọc khác có thể phát hiện các cạnh dọc. Trong quá trình đào tạo, các giá trị của các bộ lọc này được điều chỉnh để tối ưu hóa quá trình trích xuất tính năng.

− Stride: xác định mức độ bộ lọc di chuyển trong quá trình hoạt động tích chập. Bước tiến bằng 1 có nghĩa là bộ lọc di chuyển một pixel tại một thời điểm, trong khi bước tiến bằng 2 có nghĩa là nó di chuyển hai pixel tại một

thời điểm. Bước tiến lớn hơn dẫn đến bản đồ tính năng đầu ra nhỏ hơn và tính toán nhanh hơn.

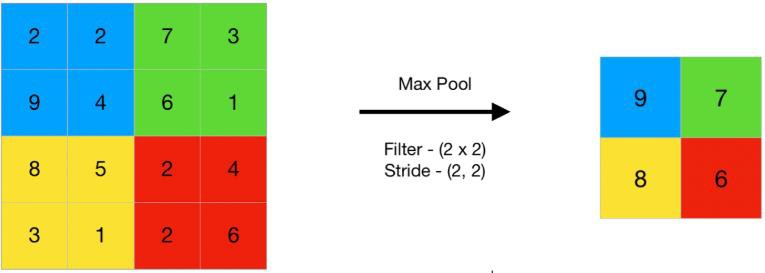
− Padding: liên quan đến việc thêm các pixel bổ sung xung quanh dữ liệu đầu vào để kiểm soát kích thước không gian của bản đồ đặc trưng đầu ra. Có hai loại padding phổ biến: ‘valid’ padding,, không thêm pixel bổ sung và ‘same’ padding, thêm pixel để đảm bảo bản đồ đặc điểm đầu ra có cùng kích thước với đầu vào.

− Activation Function: Sau phép toán tích chập, một hàm kích hoạt, thường là Rectified Linear Unit (ReLU) được đưa vào mô hình mạng nơ-ron.

#### Lớp rút gọn (Pooling layer)

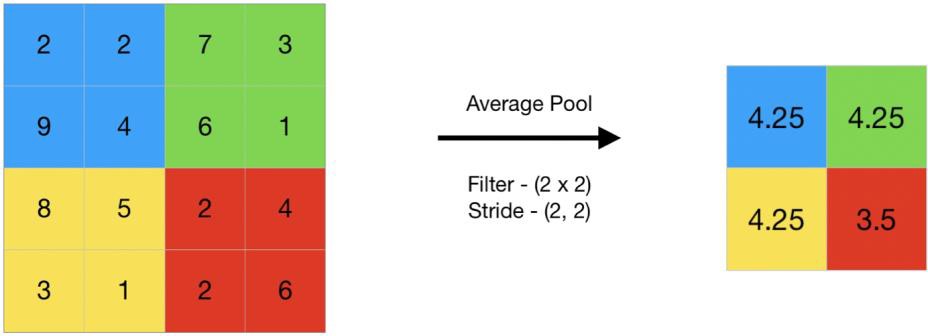
Pooling layer làm giảm số lượng tham số cần học và lượng tính toán được thực hiện. Pooling layer tổng hợp các đặc trưng có trong một vùng của bản đồ đặc trưng được tạo bởi lớp tích chập. Vì vậy, các thao tác tiếp theo được thực hiện trên các đặc trưng đã được tổng hợp thay vì các đặc trưng được định vị chính xác do lớp tích chập tạo ra. Điều này làm cho mô hình trở nên mạnh mẽ hơn trước các biến thể về vị trí của các đặc trưng trong hình ảnh đầu vào.

Các loại lớp rút gọn (Pooling layer):

− Max Pooling: là hoạt động chọn phần tử lớn nhất từ vùng bản đồ đặc trưng được bộ lọc bởi bao phủ. Do đó, đầu ra sau lớp max-pooling sẽ là một bản đồ đặc trưng chứa các đặc trưng nổi bật nhất của bản đồ đặc trưng trước đó.

*Hình 2. 8. Xử lý ảnh với Max-pooling*

− Average-Pooling: tính trung bình của các phần tử có trong vùng của bản đồ đặc trưng được bộ lọc bao phủ. Do đó, trong khi max-pooling cung cấp đặc trưng nổi bật nhất trong một vùng cụ thể của bản đồ đặc trưng, thì pooling trung bình cho ra giá trị trung bình của các đặc trưng có trong vùng đó.



*Hình 2. 9. Xử lý ảnh với Average-Pooling*

#### Lớp kết nối đầy đủ (Fully connected layer)

Sau khi ảnh được đi qua nhiều lớp tích chập (Convolutional layer) và lớp rút gọn (Pooling layer), mô hình sẽ học được các đặc điểm tương đối của ảnh như mắt, mũi, khuôn mặt, và các đặc trưng khác. Sau đó, dữ liệu đầu ra của lớp cuối cùng sẽ được chuyển đổi thành một vector.

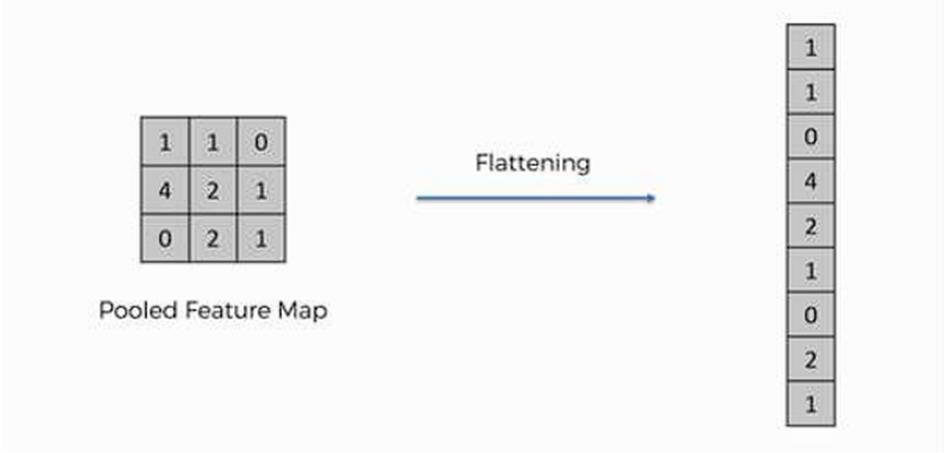
Trong mô hình mạng CNN, các lớp kết nối đầy đủ được đặt ở cuối mô hình và được sử dụng để tối ưu hóa dữ liệu, ví dụ như độ chính xác của lớp. Các thành phần chính của lớp kết nối đầy đủ (Fully connected layer)

− Nơ-ron: Đơn vị cơ bản nhận thông tin đầu vào từ tất cả các nơron của lớp trước và gửi thông tin đầu ra đến tất cả các nơron của lớp tiếp theo.

− Trọng lượng: Mỗi kết nối giữa các nơron có một trọng số liên quan, biểu thị sức mạnh và ảnh hưởng của một nơron lên nơron khác.

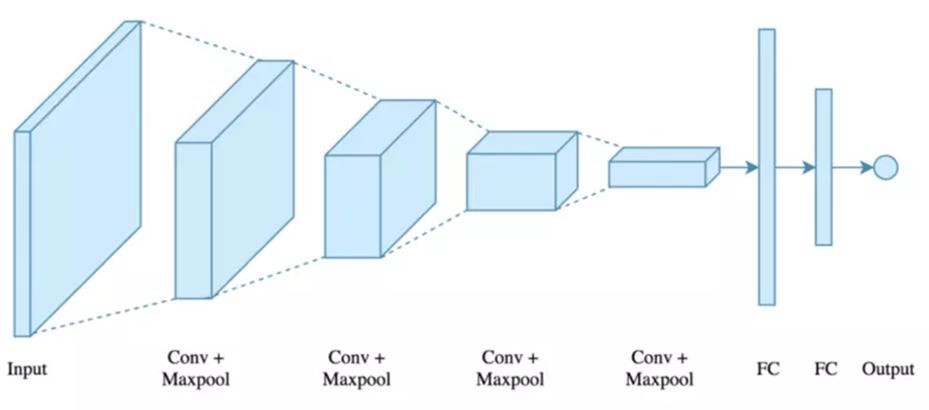
− Độ lệch: Một thuật ngữ độ lệch cho mỗi nơron giúp điều chỉnh đầu ra cùng với tổng trọng số của các đầu vào.

− Hàm kích hoạt: Các hàm như ReLU, Sigmoid hoặc Tanh đưa tính phi tuyến tính vào mô hình, cho phép mô hình học được các mẫu và hành vi phức tạp.



*Hình 2. 10. Xử lý ảnh với Flatten*

Sau đó sử dụng các lớp fully connected layer để kết hợp các đặc điểm của ảnh thì sẽ ra được output của mô hình.



*Hình 2. 11. Mô hình mạng CNN hoàn chỉnh*

#### CNN trong nhận diện cảm xúc khuôn mặt người

Cảm xúc là một khía cạnh cơ bản của giao tiếp và hành vi của con người. Chúng được thể hiện thông qua biểu cảm khuôn mặt, ngôn ngữ cơ thể và giọng nói. Mặc dù tất cả những tín hiệu này đều quan trọng, nhưng biểu cảm khuôn mặt thường là chỉ báo cảm xúc dễ thấy và đáng tin cậy nhất. Phát hiện cảm xúc bằng CNN chủ yếu tập trung vào việc phân tích biểu cảm khuôn mặt để xác định trạng thái cảm xúc của một cá nhân.

Mạng nơ-ron tích chập (CNN) là một loại kiến trúc mạng nơ-ron học sâu được thiết kế riêng để xử lý dữ liệu dạng lưới, chẳng hạn như hình ảnh và video. CNN đã cách mạng hóa lĩnh vực thị giác máy tính và được sử dụng rộng rãi cho nhiều tác vụ khác

nhau, bao gồm phân loại hình ảnh, phát hiện đối tượng, nhận dạng khuôn mặt và tạo hình ảnh. Chúng đặc biệt hiệu quả trong việc nắm bắt các phân cấp không gian của các tính năng trong dữ liệu.

Kiến trúc đơn giản của một CNN tiêu biểu để phân loại hình ảnh

#### − Lớp đầu vào:

+ Nhận dữ liệu hình ảnh thô. Hình ảnh thường được biểu diễn dưới dạng lưới các điểm ảnh với ba kênh màu (đỏ, lục và lam - RGB)

+ Kích thước của lớp đầu vào khớp với kích thước của hình ảnh đầu vào (Ví dụ: 28x28x1 cho hình ảnh 28x28 pixel với kênh RGB)

#### − Các lớp tích chập (Convolutional layers):

+ Các lớp tích chập bao gồm nhiều bộ lọc. Mỗi bộ lọc quét qua hình ảnh đầu vào bằng cửa sổ trượt.

+ Phép tính tích chập tính toán vô hướng giữa bộ lọc và vùng đầu vào. Các hàm kích hoạt đưa tính phi tuyến vào mạng.

+ Các lớp Max-Pooling giảm kích thước không gian (chiều rộng và chiều cao) để giảm độ phức tạp trong tính toán.

#### − Các lớp gộp (Pooling Layers):

+ Các lớp gộp (ví dụ: MaxPooling hoặc AveragePooling) làm giảm kích thước không gian của các biểu đồ đặc trưng trong khi vẫn giữ lại các thông tin quan trọng.

+ Việc gộp nhóm giúp mạng lưới mạnh mẽ hơn trước sự thay đổi về vị trí hoặc kích thước của các đối tượng trong đầu vào.

− ***Làm phẳng lớp (Flatten Layer):*** Lớp làm phẳng định hình lại đầu ra của các lớp trước thành một vectơ 1D, cho phép đưa vào một lớp dày đặc.

#### − Các lớp được kết nối đầy đủ (Fully Connected Layers):

+ Sau nhiều lớp tích chập (convolutional layers) và lớp pooling, các mạng nơ-ron tích chập (CNN) thường có một hoặc nhiều lớp kết nối đầy đủ (còn gọi là dense layers).

+ Các lớp kết nối đầy đủ (fully connected layers) kết hợp các đặc trưng cao cấp được học từ các lớp trước đó và đưa ra dự đoán cuối cùng. Trong các tác vụ phân loại, các lớp này đưa ra xác suất của các lớp.

− ***Hàm mất mát (Loss Function):*** Các mạng nơ-ron tích chập (CNN) được huấn luyện bằng một hàm mất mát (ví dụ, hàm mất mát phân loại chéo (categorical cross-entropy) cho các tác vụ phân loại) đo lường sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. Mục tiêu trong quá trình huấn luyện là tối thiểu hóa hàm mất mát này.

#### − Truyền ngược và tối ưu hóa (Backpropagation and Optimization):

+ Các mạng nơ-ron tích chập (CNN) được huấn luyện bằng cách sử dụng thuật toán lan truyền ngược (backpropagation) và các thuật toán tối ưu hóa (ví dụ: gradient descent ngẫu nhiên hoặc các biến thể của nó) để cập nhật các tham số của mạng (trọng số và độ lệch) và giảm thiểu hàm mất mát.

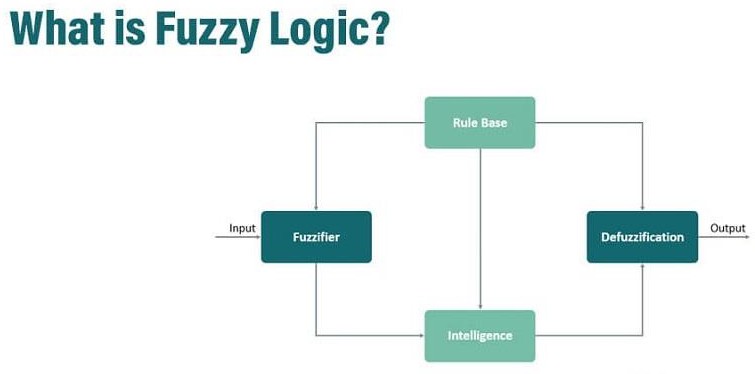
#### − Đầu ra của Mô hình (Model Output):

+ Đầu ra cuối cùng là một phân phối xác suất trên các lớp.

+ Trong quá trình huấn luyện, mô hình được tối ưu hóa bằng cách sử dụng một hàm mất mát (ví dụ: hàm mất mát phân loại chéo - categorical cross- entropy) để đưa ra dự đoán gần nhất có thể với nhãn thực tế (ground truth labels).

### Giới thiệu về Fuzzy Logic

### Khái niệm cơ bản về Fuzzy Logic

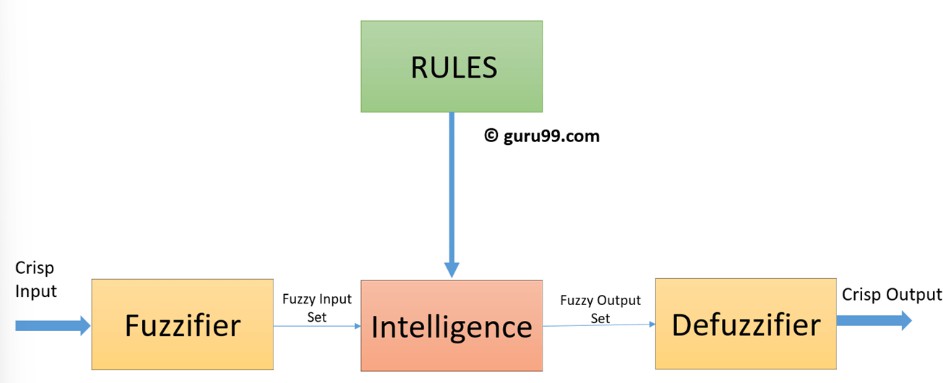


*Hình 2. 12. Fuzzy logic là gì?*

Fuzzy Logic, hay Logic mờ, là một khái niệm được sử dụng trong lĩnh vực trí tuệ nhân tạo (AI) nhằm xử lý và ra quyết định trong các tình huống không rõ ràng hoặc không chắc chắn.

Không giống như logic cổ điển chỉ có hai giá trị rõ ràng là CÓ (TRUE) và KHÔNG (FALSE), Fuzzy Logic cho phép các giá trị nằm giữa hai trạng thái này, giúp máy tính có thể mô phỏng khả năng suy luận gần gũi với con người. Khái niệm này được phát minh bởi Lotfi Zadeh, một nhà khoa học máy tính, và đã trở thành một phần quan trọng trong nhiều ứng dụng hiện đại.

### Thành phần của hệ thống Fuzzy Logic



*Hình 2. 13. Thành phần của hệ thống Fuzzy Logic*

Fuzzy Logic có 4 thành phần chính:

− Fuzzification(mờ hóa): được sử dụng để chuyển đổi các đầu vào, tức là các số chính xác (crisp numbers) thành các tập mờ (fuzzy sets). Các đầu vào chính xác này về cơ bản là các đầu vào được đo bởi cảm biến và đưa vào hệ thống điều khiển để xử lý, chẳng hạn như nhiệt độ, áp suất, số vòng quay (rpm),...

− RuleBase (quy tắc mờ): Chứa tập hợp các quy tắc và điều kiện IF-THEN do các chuyên gia cung cấp để điều chỉnh hệ thống ra quyết định, dựa trên thông tin ngôn ngữ.

− Inference Engine (công cụ suy luận): xác định mức độ khớp của đầu vào mờ hiện tại với từng quy tắc và quyết định quy tắc nào sẽ được kích hoạt theo trường đầu vào. Dựa trên % khớp, nó xác định quy tắc nào cần triển khai theo trường đầu vào nhất định. Sau đó, các quy tắc được áp dụng sẽ được kết hợp để phát triển các hành động kiểm soát.

− Defuzzification(giải mờ): sau khi các suy luận được thực hiện, kết quả đầu ra sẽ vẫn ở dạng mờ. Mô-đun Defuzzification có nhiệm vụ chuyển đổi các tập mờ này thành giá trị rõ ràng và có thể sử dụng được. Đây là bước cuối cùng trong quá trình ra quyết định của hệ thống.

# Ứng dụng của Fuzzy Logic trong nhận dạng cảm xúc khuôn mặt

#### Sự khác biệt giữa logic truyền thống và logic mờ

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Logic truyền thống** | **Logic mờ** |
| Giá trị Đúng/ Sai | Chỉ có 2 giá trị Đúng hoặc Sai | Các giá trị nằm trên một dải liên  tục từ 0 đến 1, đại diện cho mức độ đúng của một mệnh đề. |
| Độ linh hoạt | Cứng nhắc, chỉ phân loại rõ ràng giữa đúng hoặc sai. | Linh hoạt, cho phép các mức độ đúng sai khác nhau, phù hợp với  thực tế phức tạp. |
| Khả năng xử lý thông tin mập mờ | Không thể xử lý thông tin mập mờ hoặc không rõ  ràng. | Có thể xử lý thông tin mập mờ và các mức độ không chắc chắn. |
| Ứng dụng | Toán học, máy tính, lập trình, các bài toán xác định  rõ ràng. | Điều khiển hệ thống, trí tuệ nhân tạo, phân tích tài chính, các hệ  thống tự động. |

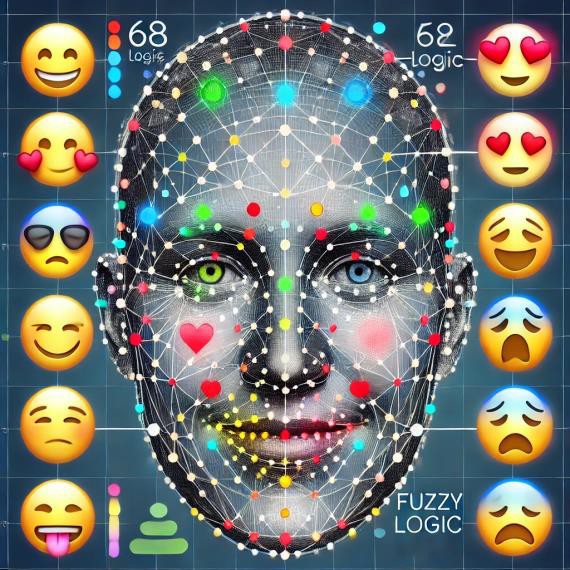
*Bảng 1.2. Sự khác biệt giữa logic truyền thống và logic mờ*

#### Tại sao nên sử dụng Fuzzy Logic trong nhận diện cảm xúc khuôn mặt người?

Cảm xúc là một quá trình phức tạp và không luôn luôn rõ ràng. Một người có thể cảm thấy "vui vẻ một chút" hay "buồn nhưng không quá buồn". Các biểu hiện cảm xúc không phải lúc nào cũng chính xác và dễ phân loại. Fuzzy Logic giúp mô hình hóa cảm xúc khi chúng không có ranh giới rõ ràng, chẳng hạn "cảm thấy một chút giận dữ" hay "vừa vui vừa lo lắng".

Con người thường không biểu hiện cảm xúc một cách rõ ràng và rạch ròi. Ví dụ, một người có thể cảm thấy "hơi vui" nhưng cũng "hơi lo lắng" trong cùng một tình

huống. Fuzzy Logic có thể mô phỏng các tình huống này tốt hơn so với các hệ thống phân loại chính xác hoặc nhị phân.

Thay vì phân loại cảm xúc một cách cứng nhắc (chỉ vui, buồn, giận dữ), Fuzzy Logic có thể đưa ra một thang đo cảm xúc, chẳng hạn như "50% vui vẻ, 30% lo lắng", điều này phản ánh thực tế hơn.

*Hình 2. 14. Ví dụ về dùng Fuzzy Logic nhận diện cảm xúc*

Hệ thống nhận diện cảm xúc dựa trên Fuzzy Logic bao gồm các bước chính:

Đưa vào hình ảnh có khuôn mặt cần nhận diện cảm xúc vào và thực hiện những kỹ thuật tiền xử lý dữ liệu để lọc nhiễu và loại bỏ những thông tin không cần thiết, trích xuất những đặc trưng.

Xây dựng hệ thống suy luận mờ:

1. **Xây dựng tập mờ (Fuzzification):** Chuyển đổi các dữ liệu đầu vào thành các giá trị mờ.
2. **Biểu diễn dữ liệu mờ**: Các đặc trưng như độ mở của miệng(cười), mức độ mở mắt, hay hướng chân mày được chuyển thành các giá trị mờ thông qua hàm thành viên (Membership Function).
3. **Hệ thống suy luận mờ (Fuzzy Inference System)**: Áp dụng các luật mờ để nhận diện cảm xúc khuôn mặt, ví dụ:

− Nếu miệng cười lớn và mắt mở vừa phải → cảm xúc = *vui*.

− Nếu môi nhếch nhẹ và chân mày cau lại → cảm xúc = *tức giận*.

− Nếu miệng không cười và mắt mở nhỏ → cảm xúc = *buồn*

− Nếu miệng mở to và mắt mở to → cảm xúc = *ngạc nhiên*

1. **Giải mờ (Defuzzification)**: Hệ thống đưa ra mức độ khả năng thuộc về các cảm xúc như vui, buồn, tức giận, hay ngạc nhiên.

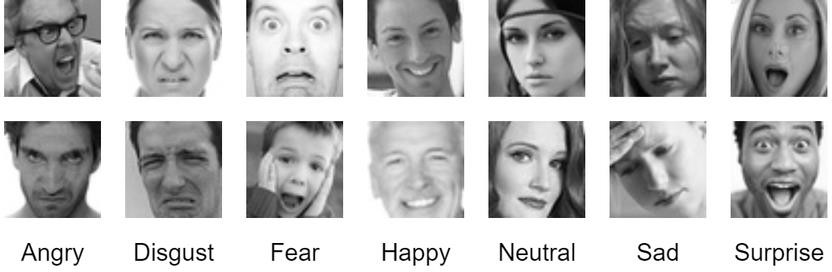
## PHẦN 3. THIẾT KẾ HỆ THỐNG

### Tệp cơ sở dữ liệu

Trong các mô hình học sâu, bộ dữ liệu đóng vai trò nền tảng và không thể thiếu, quyết định trực tiếp đến hiệu quả hoạt động của hệ thống. Đối với một hệ thống nhận diện biểu cảm khuôn mặt, bộ dữ liệu chứa đựng các thông tin quan trọng, bao gồm những đặc trưng cần thiết để mô hình có thể học và phân tích chính xác. Quy trình chuẩn bị bộ dữ liệu không chỉ dừng lại ở việc thu thập, mà còn bao gồm việc làm sạch, tổ chức và tiền xử lý để đảm bảo tính nhất quán và độ tin cậy. Chỉ sau khi hoàn tất các bước này, bộ dữ liệu mới sẵn sàng phục vụ cho việc trích xuất các đặc điểm và hiển thị kết quả một cách hiệu quả, giúp hệ thống đạt được hiệu suất tối ưu.

Hiện nay, có nhiều bộ dữ liệu phổ biến trên internet hỗ trợ cho việc nhận diện cảm xúc khuôn mặt, chẳng hạn như Cohn-Kanade (CK), Japanese Female Facial Expressions (JAFFE), hay Facial Expression Research Database (FERG). Những bộ dữ liệu này đã được sử dụng rộng rãi trong các nghiên cứu và ứng dụng thực tế. Tuy nhiên, trong hệ thống nhận diện cảm xúc khuôn mặt mà nhóm em đang phát triển, nhóm đã quyết định sử dụng bộ dữ liệu Facial Expression Recognition Challenge (FER-2013) từ nền tảng Kaggle, bởi nó đáp ứng tốt các yêu cầu về tính đa dạng và quy mô cần thiết cho dự án.

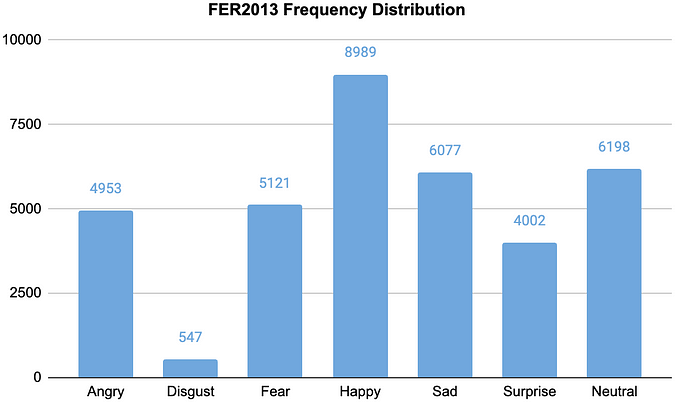
Bộ dữ liệu FER-2013 bao gồm các hình ảnh khuôn mặt đen trắng với kích thước 48x48 pixel. Bộ dữ liệu đã được phân loại thành từng hình ảnh với nhãn là một trong bảy loại cảm xúc: tức giận (Angry), ghê tởm (Disgust), sợ hãi (Fear), hạnh phúc (Happy), buồn (Sad), ngạc nhiên (Surprise), và trung tính (Neutral). Bộ dữ liệu được chia thành hai phần chính: tập huấn luyện gồm 28.709 hình ảnh và tập kiểm tra công khai với 3.589 hình ảnh.



*Hình 3. 1. Một số hình ảnh trong bộ dữ liệu FER-2013*

Quá trình biên soạn bộ dữ liệu FER2013 được thực hiện bằng cách tiến hành các lượt tìm kiếm hình ảnh trên Google với từ khóa là từng cảm xúc và các từ đồng nghĩa của chúng. (Quinn, M. & Sivesind, G. & Reis, G., n.d.) Cách tiếp cận này nhằm đảm bảo thu thập được một phạm vi biểu cảm khuôn mặt đa dạng, nhưng đồng thời cũng có nguy cơ dẫn đến việc biểu cảm trong dữ liệu không hoàn toàn chính xác.

Mặc dù có những hạn chế tiềm ẩn, FER2013 vẫn là một trong những bộ dữ liệu được sử dụng rộng rãi trong lĩnh vực này nhờ khả năng bao quát các trạng thái cảm xúc khác nhau. Tuy nhiên, một thách thức cố hữu của bộ dữ liệu này là sự mất cân bằng về số lượng hình ảnh giữa các nhãn cảm xúc. Hình dưới đây minh họa tần suất phân phối của các nhãn cảm xúc trong bộ dữ liệu.



*Hình 3. 2. Số lượng ảnh của từng nhãn cảm xúc trong bộ dữ liệu FER-2013*

Như có thể thấy từ trên, một số cảm xúc có sự đại diện không cân xứng so với các cảm xúc khác, điều này ảnh hưởng đến sự ổn định và công bằng của các mô hình

được huấn luyện. Sự mất cân đối trong bộ dữ liệu này gây ra những tác động đáng kể đối với việc đánh giá các mô hình học sâu được huấn luyện trên FER2013, vì các chỉ số hiệu suất có thể bị nghiêng về các lớp cảm xúc chiếm ưu thế, dẫn đến kết quả thiên lệch và đánh giá không chính xác về hiệu quả của mô hình.

Bên cạnh đó, bộ dữ liệu còn tồn tại một số hình ảnh không phải là khuôn mặt người, điều này có thể làm giảm chất lượng và độ chính xác của quá trình huấn luyện mô hình.



*Hình 3. 3. Một số hình ảnh trong phải khuôn mặt người trong bộ dữ liệu FER-2013*

### Môi trường và công cụ sử dụng

Trong đề tài về nhận dạng cảm xúc khuôn mặt bằng mạng nơ-ron tích chập (CNN), nhóm em sẽ sử dụng thư viện Keras để xây dựng mô hình từ đầu trên nền tảng Google Colab, với ngôn ngữ lập trình Python.

### Python

Python là một ngôn ngữ lập trình bậc cao, đa năng và dễ sử dụng, được thiết kế với cú pháp đơn giản, giúp người dùng dễ dàng học và áp dụng trong nhiều lĩnh vực khác nhau. Với sự hỗ trợ mạnh mẽ từ cộng đồng lập trình viên toàn cầu, Python đã nhanh chóng trở thành một trong những ngôn ngữ phổ biến nhất trên thế giới. Một điểm nổi bật của Python là thư viện phong phú và đa dạng, cho phép nó được sử dụng rộng rãi trong các lĩnh vực như phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, học sâu (deep learning), và học máy (machine learning).

Các thư viện như NumPy, Pandas, và Matplotlib giúp xử lý và phân tích dữ liệu nhanh chóng, trong khi các công cụ mạnh mẽ như TensorFlow, PyTorch và Keras hỗ trợ xây dựng và triển khai các mô hình học máy và mạng nơ-ron phức tạp. Python cũng có khả năng tương thích tốt với nhiều nền tảng, từ máy tính cá nhân đến các máy chủ trên đám mây, làm cho nó trở thành lựa chọn hàng đầu cho các nhà phát triển và nhà nghiên cứu.

Ngoài ra, Python còn nổi bật nhờ khả năng tương tác với các công cụ tính toán hiệu năng cao và các tài nguyên như GPU, giúp tăng tốc độ xử lý trong các ứng dụng yêu cầu tính toán phức tạp, chẳng hạn như xử lý hình ảnh, phân tích dữ liệu lớn hoặc huấn luyện các mô hình trí tuệ nhân tạo. Sự linh hoạt, tính mở rộng, và khả năng phát triển nhanh của Python đã khiến nó trở thành ngôn ngữ lý tưởng trong cả nghiên cứu lẫn ứng dụng thực tiễn.

### Thư viện Keras

Keras là một API mạng nơ-ron cấp cao, được phát triển bởi Google và viết bằng ngôn ngữ Python. Thư viện này hỗ trợ nhiều chương trình phụ trợ như TensorFlow, Theano, và CNTK, cho phép người dùng khai thác các ưu điểm của từng thư viện cho các nhiệm vụ khác nhau. Mục tiêu chính của Keras là đơn giản hóa việc xây dựng và thử nghiệm các mô hình học sâu, đặc biệt phù hợp với người mới bắt đầu hoặc các nhà nghiên cứu không chuyên sâu về Machine Learning.

Với Keras, việc xây dựng mạng nơ-ron trở nên dễ dàng nhờ khả năng xếp chồng và cấu hình các lớp một cách trực quan, hỗ trợ nhiều kiến trúc mạng nơ-ron khác nhau. Ngoài ra, Keras còn tích hợp sẵn các chức năng quan trọng như tiền xử lý dữ liệu, đánh giá hiệu suất mô hình, và trực quan hóa, giúp tối ưu hóa quy trình làm việc. Điểm mạnh khác của Keras là khả năng chạy mượt mà trên cả CPU và GPU, mang lại tính linh hoạt cao và hiệu suất vượt trội cho nhiều dự án học sâu.

Đối với nhóm chúng em, những người mới tiếp cận Deep Learning, việc lựa chọn Keras là một quyết định hợp lý. Thư viện này không chỉ đơn giản, dễ sử dụng mà còn hỗ trợ đầy đủ các công cụ cần thiết để triển khai và tối ưu hóa mô hình một cách hiệu quả.

### Thư viện Numpy

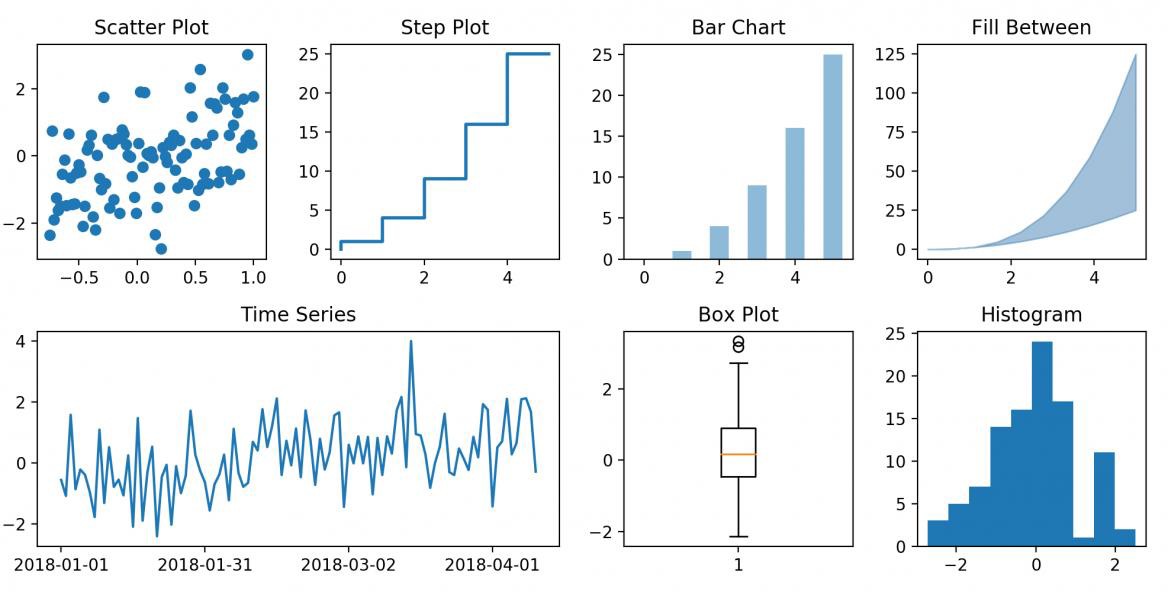
NumPy (Numerical Python) là một thư viện mã nguồn mở trong Python, được sử dụng để xử lý các phép tính số học và đại số tuyến tính với hiệu suất cao. NumPy nổi bật nhờ khả năng làm việc với mảng đa chiều (ndarray), cung cấp các phép toán nhanh chóng hơn so với các cấu trúc dữ liệu thông thường như list.

Thư viện này hỗ trợ nhiều hàm tích hợp để thực hiện các phép toán đại số, thống kê, và xử lý ma trận. NumPy cũng tương thích tốt với các thư viện học sâu như TensorFlow hay PyTorch, giúp tăng tốc tính toán trên GPU. Với khả năng vector hóa,

NumPy tối ưu hóa xử lý dữ liệu lớn, giảm thời gian xử lý đáng kể. Trong đề tài này chúng em sẽ sử dụng NumPy kết hợp với thư viện Matplotlib (thư viện vẽ đồ thị).

### Thư viện Matplotlib

Matplotlib là một thư viện Python mã nguồn mở được sử dụng chủ yếu để tạo ra các biểu đồ và đồ họa trực quan. Thư viện này hỗ trợ nhiều loại biểu đồ 2D như đồ thị đường, biểu đồ cột, biểu đồ tròn, và histogram. Matplotlib cho phép người dùng tùy chỉnh nhiều yếu tố trong đồ họa như màu sắc, kích thước, kiểu đường, nhãn trục, và tiêu đề, giúp tạo ra các hình ảnh trực quan rõ ràng và dễ hiểu.

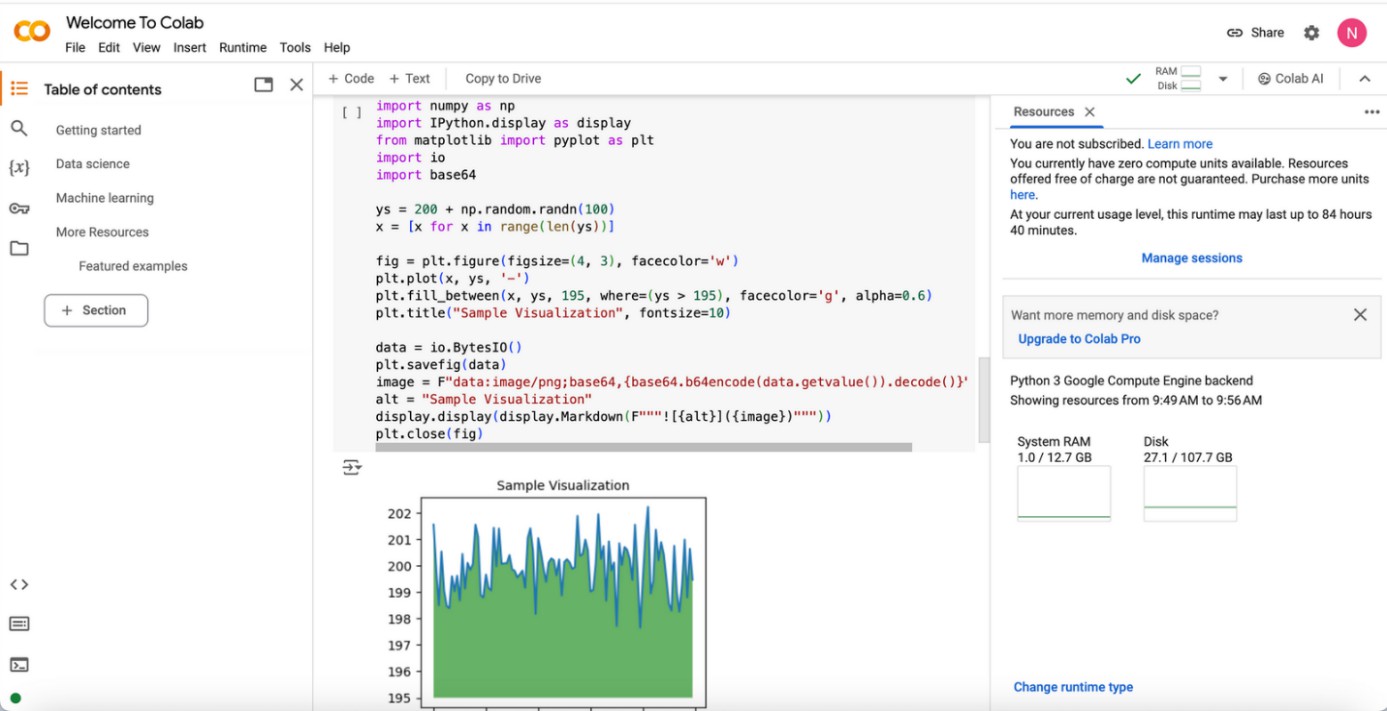


*Hình 3. 4. Một số hình ảnh về biểu đồ của thư viện Matplotlib*

Matplotlib có thể tích hợp tốt với các thư viện khác như NumPy và Pandas, giúp xử lý và trực quan hóa dữ liệu một cách hiệu quả. Thư viện này cũng hỗ trợ xuất các biểu đồ dưới nhiều định dạng như PNG, PDF, SVG, và EPS, rất tiện lợi cho việc chia sẻ và lưu trữ kết quả.

#### Google Colab

Google Colab (hoặc Google Colaboratory) là một dịch vụ miễn phí do Google cung cấp, cho phép người dùng chạy mã Python trong trình duyệt mà không cần phải cài đặt phần mềm hay cấu hình môi trường. Dịch vụ này đặc biệt hữu ích cho các dự án học máy, khoa học dữ liệu và nghiên cứu, vì nó cung cấp môi trường làm việc tích hợp với nhiều thư viện Python mạnh mẽ như TensorFlow, Keras, PyTorch, NumPy, Pandas, và Matplotlib.



*Hình 3. 5. Giao diện Google Colab*

Một trong những tính năng nổi bật của Google Colab là khả năng cung cấp GPU (Graphics Processing Unit) và TPU (Tensor Processing Unit) miễn phí, giúp tăng tốc quá trình huấn luyện các mô hình học sâu mà không cần sở hữu phần cứng đắt tiền. Điều này đặc biệt hữu ích cho những người làm việc với các mô hình phức tạp hoặc xử lý các bộ dữ liệu lớn.

Google Colab sử dụng định dạng Jupyter Notebook, cho phép người dùng viết mã Python, thực thi nó và trực quan hóa kết quả trong cùng một tệp. Người dùng có thể dễ dàng thêm văn bản, hình ảnh, đồ họa và biểu đồ vào notebook để giải thích kết quả. Ngoài ra, các notebook trong Google Colab có thể được lưu trực tiếp trên Google Drive, giúp dễ dàng truy cập và chia sẻ với đồng nghiệp hoặc bạn bè. Điều này hỗ trợ việc làm việc nhóm và hợp tác trong các dự án nghiên cứu.

Colab cũng hỗ trợ việc kết nối với các dịch vụ lưu trữ đám mây như Google Drive hoặc GitHub, giúp đồng bộ hóa và lưu trữ dữ liệu, mã nguồn và kết quả mô phỏng. Người dùng có thể tải lên hoặc tải xuống các tệp dễ dàng, hoặc sử dụng các lệnh hệ thống Unix để quản lý tệp trong môi trường làm việc.

Với giao diện dễ sử dụng và khả năng tích hợp mạnh mẽ với các công cụ và thư viện phổ biến, Google Colab là một công cụ lý tưởng cho việc phát triển mô hình học máy, phân tích dữ liệu, và nghiên cứu mà không cần cài đặt hoặc cấu hình môi trường phức tạp.

# Kaggle

**Kaggle** được ví như "Airbnb cho các nhà khoa học dữ liệu" – nơi họ dành thời gian buổi tối và cuối tuần của mình. Đây là một nền tảng cộng đồng được xây dựng để thu hút, nuôi dưỡng, đào tạo và thách thức các nhà khoa học dữ liệu trên toàn thế giới trong việc giải quyết các vấn đề về khoa học dữ liệu, học máy và phân tích dự đoán. Kaggle có hơn 536.000 thành viên hoạt động đến từ 194 quốc gia và nhận khoảng

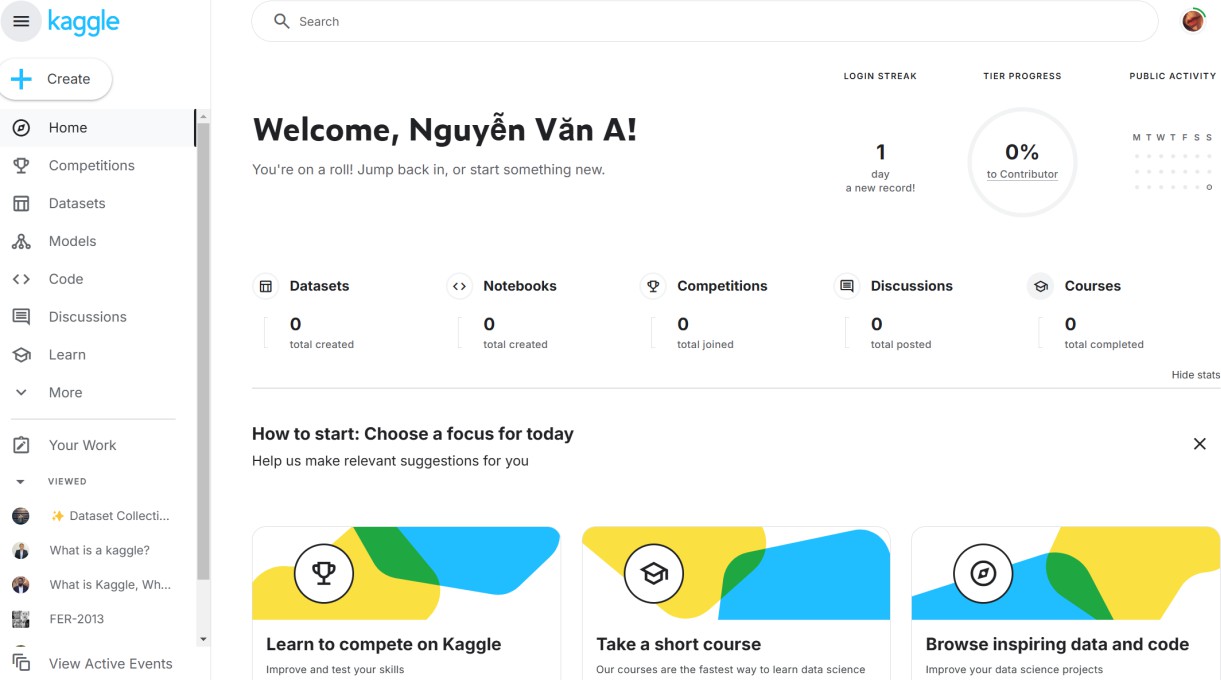
150.000 bài nộp mỗi tháng. Được khởi đầu tại Melbourne, Úc, Kaggle chuyển trụ sở đến Thung lũng Silicon vào năm 2011, huy động được 11 triệu đô la từ những nhà đầu tư nổi tiếng như Hal Varian (Kinh tế trưởng tại Google), Max Levchin (Paypal), Index và Khosla Ventures, và cuối cùng được Google mua lại vào tháng 3 năm 2017. Kaggle là điểm đến hàng đầu cho những người đam mê khoa học dữ liệu trên toàn thế giới, nơi họ cạnh tranh giành giải thưởng và nâng cao thứ hạng Kaggle của mình. Hiện tại, chỉ có 94 **Grandmaster Kaggle** trên thế giới.

**Kaggle** cho phép các nhà khoa học dữ liệu và các nhà phát triển khác tham gia vào các cuộc thi học máy, viết và chia sẻ mã, cũng như lưu trữ dữ liệu. Các loại bài toán trên Kaggle rất đa dạng, từ dự đoán khả năng mắc bệnh ung thư thông qua phân tích hồ sơ bệnh nhân, đến phân tích cảm xúc từ đánh giá phim và tác động của chúng đến phản ứng của khán giả.

Các bài toán này có thể đến từ nhiều nguồn, bao gồm cả những bài tập mang tính giáo dục hoặc các bài toán thực tế mà các công ty đang cố gắng giải quyết. Kaggle tạo ra môi trường cạnh tranh bằng cách trao giải thưởng và thứ hạng cho người thắng cuộc. Các giải thưởng không chỉ là tiền mặt mà còn có thể là cơ hội việc làm hoặc sản phẩm miễn phí từ công ty tổ chức cuộc thi.

**Cộng đồng Kaggle** đang phát triển nhanh chóng với hơn một triệu thành viên (gọi là Kagglers). Họ đã nộp hơn 4 triệu mô hình học máy trong các cuộc thi, chia sẻ hơn 1.000 bộ dữ liệu, đăng hơn 170.000 bài viết trên diễn đàn và hơn 250 tập lệnh. Sự tăng trưởng nhanh chóng này, theo người sáng lập, là nhờ vào nội dung chất lượng cao, dữ liệu, và mã nguồn được cộng đồng Kaggle chia sẻ.

Hầu hết người dùng Kaggle rất tích cực, đóng góp hơn 4.000 bài viết diễn đàn mỗi tháng và hơn 3.500 bài nộp hàng ngày. Đây là nơi lý tưởng cho các nhà khoa học dữ liệu và kỹ sư học máy trên toàn thế giới.



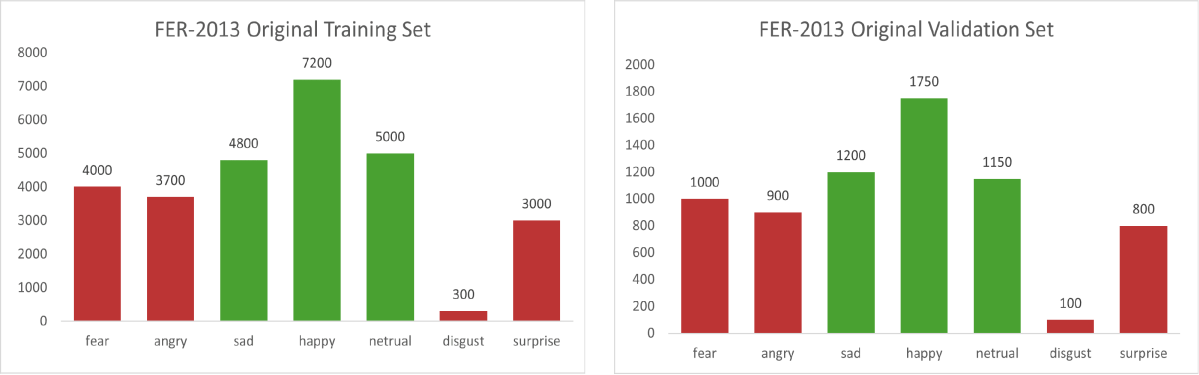
*Hình 3.6. Trang chủ của Kaggle*

### Hệ thống sử dụng mạng nơron tích chập (CNN)

#### Phân tích dữ liệu

Trước khi đưa dữ liệu vào mô hình, việc phân tích dữ liệu là một bước rất quan trọng. Chúng em sẽ tiến hành phân tích bộ dữ liệu và lựa chọn phương pháp phù hợp để chuẩn bị dữ liệu cho quá trình huấn luyện mô hình.

Bộ dữ liệu FER-2013 gồm tập huấn luyện gồm 28.709 hình ảnh và tập kiểm tra công khai với 3.589 hình ảnh. với kích thước 48x48 pixel. Dữ liệu này được chia thành hai tập con: Tập huấn luyện (Train) và Tập kiểm tra (Test). Mỗi tập dữ liệu này lại có 7 thư mục con, tương ứng với 7 cảm xúc khuôn mặt đã được phân loại và gán nhãn, bao gồm các cảm xúc như giận dữ, ghê tởm, sợ hãi, hạnh phúc, buồn bã, ngạc nhiên và trung tính.



*Hình 3. 7. Biểu đổ thể hiện số lượng hình ảnh của*

*từng nhãn và từng tập dữ liệu*

Trong biểu đồ phân tích dữ liệu trên, chúng ta có thể nhận thấy bộ dữ liệu này khá mất cân bằng. Cảm xúc "khó chịu" (Disgust) chỉ có rất ít hình ảnh, trong khi cảm xúc "hạnh phúc" (Happy) lại chiếm số lượng hình ảnh rất lớn. Điều này có thể gây khó khăn cho mô hình khi học và nhận diện cảm xúc "khó chịu", vì số lượng mẫu ít ỏi. Hơn nữa, sự mất cân bằng này có thể khiến mô hình dễ dàng nhận nhầm các cảm xúc khác, đặc biệt là cảm xúc "hạnh phúc", vì nó xuất hiện quá nhiều trong tập dữ liệu.

#### Xử lý dữ liệu

Xử lý dữ liệu là một bước quan trọng trong quy trình xây dựng mô hình nhận dạng cảm xúc khuôn mặt. Bước này đảm bảo rằng dữ liệu được chuẩn hóa và định dạng một cách phù hợp để cung cấp cho mô hình học sâu, giúp cải thiện hiệu quả huấn luyện và giảm thiểu các lỗi tiềm ẩn do dữ liệu không đồng nhất.

### Liên kết Google Drive với Google Colab

Vì dữ liệu huấn luyện và kiểm tra được lưu trên Google Drive, chúng em liên kết Google Colab với Google Drive để truy cập dữ liệu dễ dàng:

Lệnh trên sẽ gắn Google Drive vào Colab tại thư mục /content/drive. Sau đó, bạn cần xác định vị trí chính xác của dữ liệu trên Google Drive.

### Khởi tạo các generator

* + - 1. Tăng cường dữ liệu và chuẩn hóa cho huấn luyện:

− Sử dụng ImageDataGenerator với các tham số như:

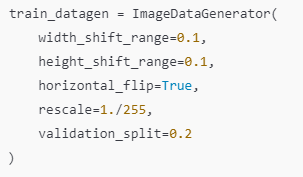
+ width\_shift\_range=0.1: Dịch ảnh ngang lên đến 10%.

+ height\_shift\_range=0.1: Dịch ảnh dọc lên đến 10%.

+ horizontal\_flip=True: Lật ngang ảnh.

+ rescale=1./255: Chia pixel cho 255 để chuẩn hóa giá trị về [0, 1].

+ validation\_split=0.2: Tách 20% dữ liệu làm tập xác thực.



* + - 1. Chỉ chuẩn hóa cho tập kiểm tra: Không cần tăng cường dữ liệu: test\_datagen = ImageDataGenerator(rescale=1./255)

### Sinh dữ liệu từ thư mục

1. Tập huấn luyện (train\_gen):

− Dùng flow\_from\_directory() để lấy dữ liệu từ train\_dir.

− Các tham số quan trọng:

+ directory: Đường dẫn đến thư mục dữ liệu.

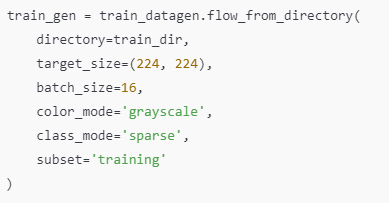
+ target\_size: Resize ảnh về kích thước 224x224.

+ batch\_size: Số ảnh trong mỗi batch (16 ảnh).

+ color\_mode: Ảnh được chuyển thành grayscale (ảnh xám).

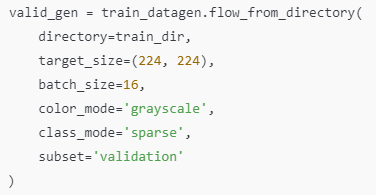
+ class\_mode: Nhãn được trả về dưới dạng chỉ số (sparse).

+ subset: Lấy phần dữ liệu là "training".



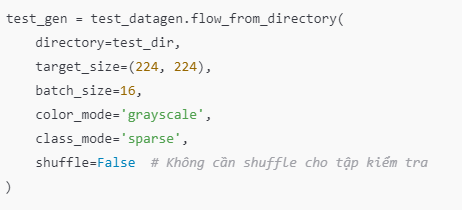
1. Tập xác thực (valid\_gen):

+ Tương tự train\_gen, nhưng subset='validation':



1. Tập kiểm tra (test\_gen):

− Dùng test\_datagen và không cần subset:



### Hoạt động của các generator

1. Thông báo khi sinh dữ liệu:

− Khi chạy các đoạn mã trên, bạn sẽ thấy:



− Các số lượng ảnh này tùy thuộc vào dữ liệu trong thư mục.

1. Lấy batch dữ liệu từ generator:
   * Generator trả về dữ liệu dưới dạng các batch.
   * Mỗi batch gồm:

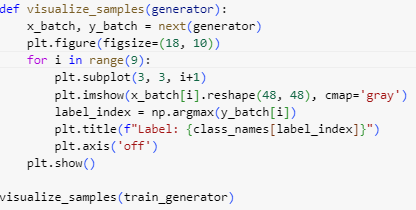
+ x\_batch: Ảnh đã được tăng cường/chuẩn hóa (dạng mảng NumPy).

+ y\_batch: Nhãn (dạng số).



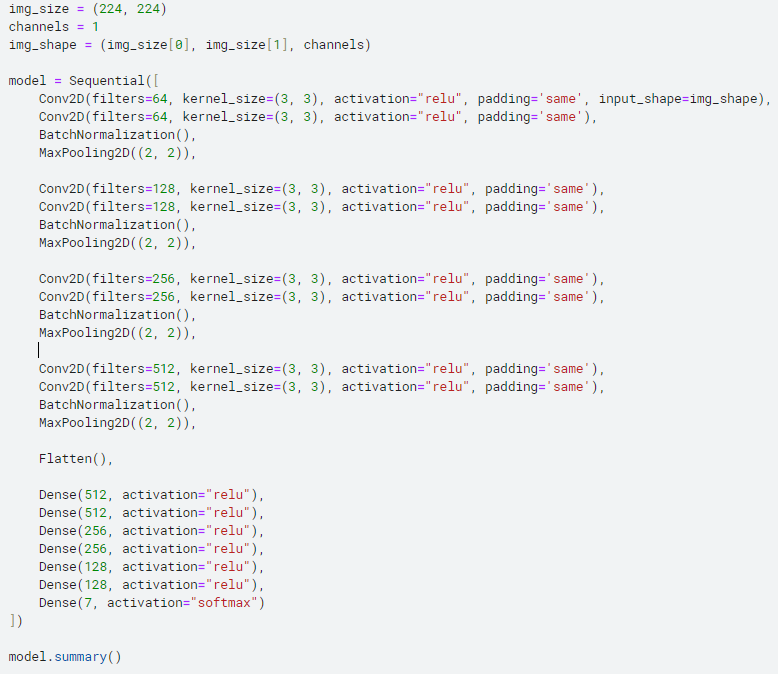
### Hiển thị ảnh mẫu từ generator:

− Hiển thị ảnh từ x\_batch cùng nhãn tương ứng:



### Xây dựng mô hình CNN

Mô hình CNN nhóm em lựa chọn là một mạng nơ-ron tích chập sâu (Deep Convolutional Neural Network) được thiết kế để phân loại ảnh thành 7 lớp (số lớp đầu ra là 7). Đây là một thiết kế khá mạnh mẽ, phù hợp cho các bài toán phân loại ảnh phức tạp như nhận diện cảm xúc từ ảnh khuôn mặt.



### Đầu vào:



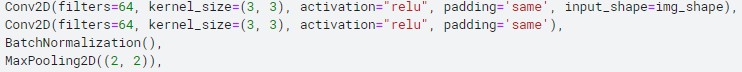
*- Kích thước đầu vào (input\_shape):*

*+ Ảnh có kích thước 224x224*.

+ Có 1 kênh màu (ảnh xám - grayscale).

### Khối tích chập và gộp (Convolution + Pooling Blocks)

− Khối 1 (64 filters):



+ Hai lớp tích chập (Conv2D):

* Sử dụng 64 bộ lọc (filters) có kích thước 3x3.
* Kích hoạt bằng hàm ReLU để giữ lại các đặc trưng phi tuyến tính.
* padding='same': Đầu ra có kích thước giống đầu vào (không làm giảm chiều không gian).

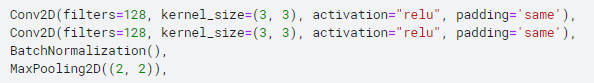
+ BatchNormalization:

* Chuẩn hóa đầu ra của các lớp tích chập, giúp tăng tốc độ huấn luyện và ổn định mô hình.

+ MaxPooling2D:

* Phép gộp cực đại (max-pooling) với kích thước 2x2, giảm kích thước không gian của ảnh (giảm 1/2 chiều dài và rộng).

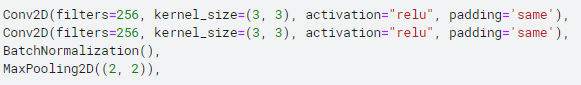
− Khối 2 (128 filters):



+ Tương tự Khối 1 nhưng với 128 bộ lọc.

+ Số lượng bộ lọc tăng lên giúp mô hình học được nhiều đặc trưng phức tạp hơn khi kích thước không gian giảm đi.

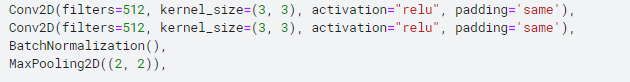
− - Khối 3 (256 filters):



+ Tăng số lượng bộ lọc lên 256.

+ Lúc này, ảnh đã giảm kích thước đáng kể, và các lớp này tập trung học các đặc trưng cấp cao hơn (các chi tiết nhỏ hoặc mẫu phức tạp).

− Khối 4 (512 filters):



+ 512 bộ lọc: Đây là khối mạnh nhất trong mô hình, học các đặc trưng cấp cao như toàn bộ khuôn mặt, các vùng cảm xúc lớn.

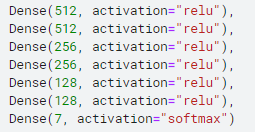
+ Ảnh sau khối này sẽ có kích thước rất nhỏ, thường là 7x7 hoặc nhỏ hơn tùy thuộc vào số khối.

### Flatten Layer

− Chuyển đổi tensor 2D hoặc 3D thành vector 1D.

− Ví dụ: Nếu đầu vào của lớp này là (7, 7, 512), thì đầu ra sẽ là một vector có kích thước 7×7×512=250887 \times 7 \times 512 = 250887×7×512=25088.

### Các lớp kết nối đầy đủ (Fully Connected Layers)



− Nhiều lớp Dense liên tiếp:

+ Tăng dần số lượng đơn vị ẩn từ 512 → 256 → 128.

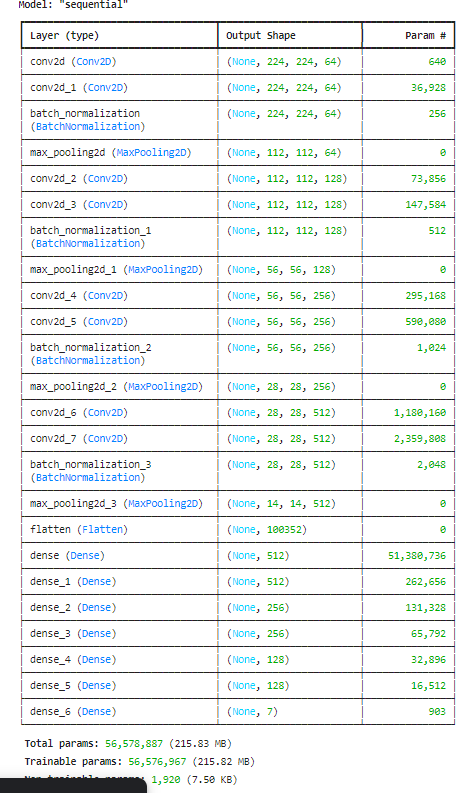
+ Mỗi lớp đều dùng hàm kích hoạt ReLU để học các đặc trưng phi tuyến.

− Lớp đầu ra (Dense(7, activation="softmax")):

+ Kết nối đầy đủ với 7 đơn vị đầu ra, tương ứng với 7 lớp cần phân loại.

+ Hàm softmax: Tính xác suất dự đoán cho từng lớp.

Sau khi xây dựng mô hình, chúng em sử dụng hàm model.summary() để tóm tắt lại của mô hình, như trong hình ảnh dưới đây. Số lượng tham số cần được huấn luyện ở cột cuối cùng.



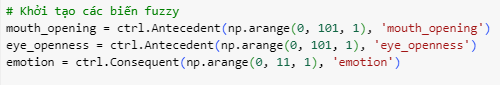
### Xây dựng mô hình Fuzzy Logic Khởi tạo các biến fuzzy

− Đầu tiên, chúng ta khai báo các biến Antecedent (tiền đề) và Consequent (kết luận) với các giá trị nằm trong các khoảng xác định:

+ mouth\_opening: Đại diện cho độ mở miệng, với giá trị từ 0 đến 100.

+ eye\_openness: Đại diện cho độ mở mắt, với giá trị từ 0 đến 100.

+ emotion: Đại diện cho mức độ cảm xúc, với giá trị từ 0 đến 10.



### Định nghĩa hàm thành viên fuzzy cho miệng, mắt và cảm xúc

Tiếp theo, chúng ta định nghĩa các hàm thành viên (membership functions) cho ba yếu tố: miệng, mắt và cảm xúc. Đây là các hàm xác định mức độ của các yếu tố trong phạm vi đã cho.

− Miệng (mouth\_opening):

+ small: Độ mở miệng nhỏ (0 - 50).

+ medium: Độ mở miệng vừa (25 - 75).

+ large: Độ mở miệng lớn (50 - 100).

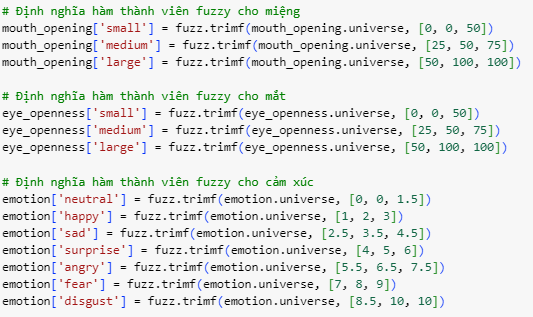
− Mắt (eye\_openness):

+ small: Độ mở mắt nhỏ (0 - 50).

+ medium: Độ mở mắt vừa (25 - 75).

+ large: Độ mở mắt lớn (50 - 100).

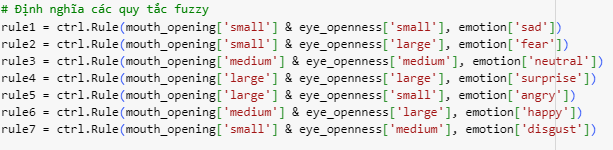
− Cảm xúc (emotion): Được phân loại thành 7 nhóm cảm xúc khác nhau (neutral, happy, sad, surprise, angry, fear, disgust). Mỗi cảm xúc có một hàm thành viên fuzzy riêng.



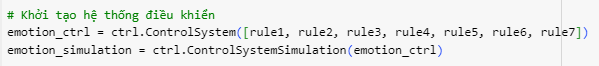
### Định nghĩa các quy tắc fuzzy

Tiếp theo, chúng ta tạo các quy tắc logic mờ (fuzzy rules). Mỗi quy tắc sẽ kết hợp các điều kiện (giá trị của miệng và mắt) và kết luận (cảm xúc):

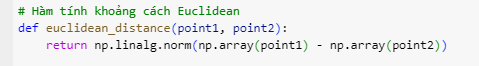
* Quy tắc 1: Nếu miệng nhỏ và mắt nhỏ → cảm xúc là "sad" (buồn).
* Quy tắc 2: Nếu miệng nhỏ và mắt lớn → cảm xúc là "fear" (sợ).
* Quy tắc 3: Nếu miệng vừa và mắt vừa → cảm xúc là "neutral" (trung tính).
* Quy tắc 4: Nếu miệng lớn và mắt lớn → cảm xúc là "surprise" (ngạc nhiên).
* Quy tắc 5: Nếu miệng lớn và mắt nhỏ → cảm xúc là "angry" (giận dữ).
* Quy tắc 6: Nếu miệng vừa và mắt lớn → cảm xúc là "happy" (hạnh phúc).
* Quy tắc 7: Nếu miệng nhỏ và mắt vừa → cảm xúc là "disgust" (ghê tởm).



### Khởi tạo hệ thống điều khiển fuzzy

Sau khi định nghĩa các quy tắc, chúng ta tạo một hệ thống điều khiển fuzzy sử dụng các quy tắc trên. Hệ thống này sẽ tính toán kết quả cảm xúc dựa trên các đầu vào từ miệng và mắt.

### Hàm tính khoảng cách Euclidean

Chúng ta sử dụng khoảng cách Euclidean để tính toán độ mở miệng và mắt từ các điểm đặc trưng trên khuôn mặt.

### Chức năng phát hiện cảm xúc

Hàm detect\_emotion(image) nhận một bức ảnh làm đầu vào, sử dụng OpenCV và dlib để phát hiện khuôn mặt và các điểm đặc trưng (landmarks) của khuôn mặt. Sau đó, nó tính toán độ mở miệng và mắt, chuẩn hóa giá trị và đưa vào mô hình fuzzy để tính toán cảm xúc.

− **Các bước**:

+ Chuyển ảnh thành ảnh xám (grayscale).

+ Phát hiện khuôn mặt trong ảnh.

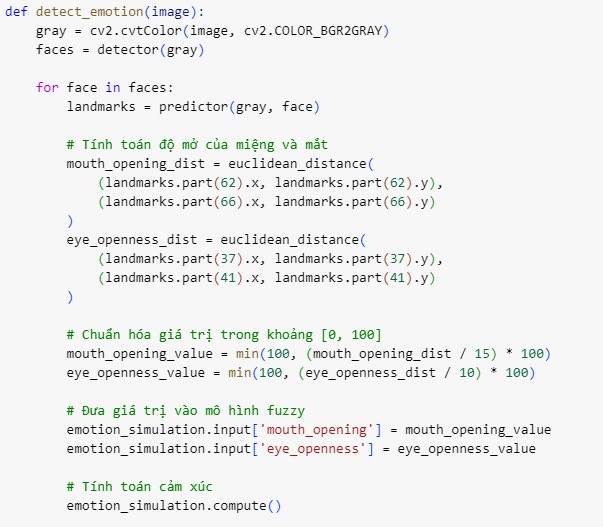
+ Xác định các điểm đặc trưng (landmarks) trên khuôn mặt.

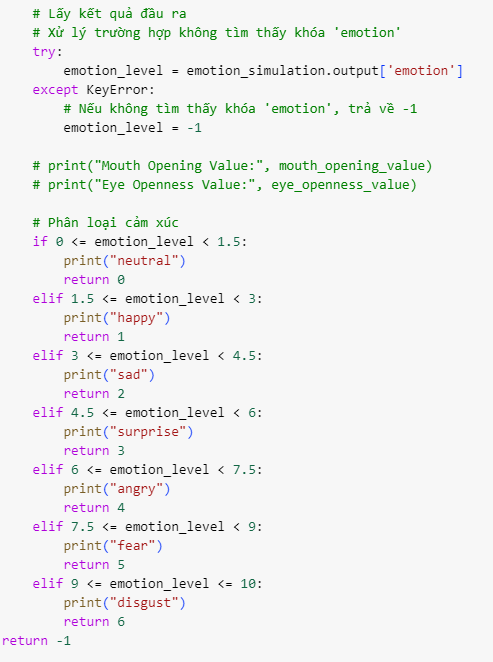
+ Tính toán độ mở miệng và mắt bằng cách sử dụng khoảng cách Euclidean.

+ Chuẩn hóa giá trị của độ mở miệng và mắt vào phạm vi [0, 100].

+ Đưa giá trị vào mô hình fuzzy và tính toán cảm xúc.

+ Dựa trên giá trị cảm xúc tính toán được, phân loại cảm xúc là "neutral", "happy", "sad", "surprise", "angry", "fear", hoặc "disgust".





## PHẦN 4. SO SÁNH KẾT QUẢ

Trong phần này, chúng ta sẽ so sánh hai phương pháp nhận dạng cảm xúc sử dụng Convolutional Neural Networks (CNN) và Fuzzy Logic. Các tiêu chí so sánh sẽ bao gồm độ chính xác, tốc độ xử lý, và hiệu quả trên các loại dữ liệu khác nhau. Chúng ta cũng sẽ phân tích và đánh giá tổng quan về các phương pháp này.

### Tiêu chí so sánh

* + 1. **Độ chính xác**:

− **CNN**: Mạng nơ-ron tích chập (CNN) có khả năng học và trích xuất đặc trưng mạnh mẽ từ dữ liệu hình ảnh, do đó thường cho kết quả chính xác cao trong các bài toán nhận dạng cảm xúc. CNN có khả năng tự động học các đặc trưng hình ảnh từ dữ liệu mà không cần phải chỉ định các đặc trưng thủ công.

− **Fuzzy Logic**: Fuzzy Logic không có khả năng học từ dữ liệu, mà dựa vào các quy tắc đã được xác định trước. Do đó, kết quả của hệ thống fuzzy logic có thể không chính xác như CNN khi xử lý các tình huống phức tạp hoặc khi dữ liệu không tuân theo các quy tắc đã định.

### Tốc độ xử lý:

− **CNN**: Việc xử lý hình ảnh qua các lớp CNN đòi hỏi nhiều phép toán, đặc biệt khi mô hình có nhiều lớp và tham số. Điều này có thể dẫn đến thời gian xử lý lâu hơn, đặc biệt khi áp dụng trên các máy tính có phần cứng hạn chế. Tuy nhiên, nếu sử dụng GPU hoặc phần cứng chuyên dụng, CNN có thể xử lý nhanh hơn.

− **Fuzzy Logic**: Fuzzy Logic có tốc độ xử lý nhanh hơn vì không yêu cầu các phép toán phức tạp như CNN. Nó chỉ dựa vào các quy tắc và phép tính đơn giản như **tính** toán giá trị fuzzy và suy diễn. Điều này giúp hệ thống fuzzy logic thực hiện nhanh chóng và ít tốn tài nguyên tính toán hơn.

### Hiệu quả trên các loại dữ liệu khác nhau:

− **CNN:** CNN rất hiệu quả với dữ liệu hình ảnh có độ phức tạp cao, vì chúng có khả năng học các đặc trưng hình ảnh phức tạp mà không cần phải xác định trước. Tuy nhiên, khi dữ liệu không đầy đủ hoặc không có sự đa dạng trong hình ảnh, CNN có thể gặp phải vấn đề overfitting hoặc không đạt được hiệu suất tối ưu.

− **Fuzzy Logic:** Hệ thống fuzzy logic có thể hoạt động tốt trong các trường hợp dữ liệu rõ ràng, dễ phân loại, hoặc khi có quy tắc rõ ràng. Tuy nhiên, khi dữ liệu có sự đa dạng lớn hoặc chứa nhiều yếu tố không xác định (như ảnh mờ hoặc bị che khuất), Fuzzy Logic có thể gặp khó khăn trong việc suy diễn chính xác cảm xúc.

### Phân tích kết quả so sánh giữa CNN và Fuzzy Logic

− **Độ chính xác**:

+ **CNN** thường cho kết quả chính xác cao hơn, đặc biệt là khi làm việc với dữ liệu hình ảnh phức tạp và đa dạng. Các mô hình CNN như VGG, ResNet hoặc Inception đã được chứng minh là rất hiệu quả trong việc nhận diện các đối tượng và cảm xúc từ hình ảnh.

+ **Fuzzy Logic,** mặc dù có thể đạt được kết quả tốt khi các yếu tố cảm xúc có thể được xác định rõ ràng qua các quy tắc logic, nhưng thiếu khả năng học từ dữ liệu, do đó không thể đạt được độ chính xác cao khi dữ liệu không tuân theo các quy tắc mờ đã định.

− **Tốc độ xử lý**:

+ **CNN**: Mặc dù CNN có thể xử lý các hình ảnh một cách chính xác và mạnh mẽ, nhưng tốc độ xử lý sẽ chậm hơn so với fuzzy logic, đặc biệt khi số lượng lớp trong mô hình tăng lên. Việc tính toán và tối ưu hóa mô hình CNN yêu cầu tài nguyên tính toán cao và thời gian huấn luyện lâu dài.

+ **Fuzzy Logic**: Fuzzy Logic có tốc độ xử lý nhanh hơn vì các quy tắc logic mờ đơn giản và không yêu cầu quá nhiều tính toán. Điều này làm cho Fuzzy Logic thích hợp cho các ứng dụng thời gian thực hoặc các thiết bị với phần cứng yếu.

### − Hiệu quả trên các loại dữ liệu khác nhau:

+ **CNN**: CNN có thể xử lý hiệu quả hầu hết các loại dữ liệu hình ảnh, đặc biệt là những dữ liệu có sự biến đổi lớn như ánh sáng, góc nhìn và khuôn mặt thay đổi. Tuy nhiên, CNN cần một lượng lớn dữ liệu huấn luyện và có thể gặp vấn đề với dữ liệu không đầy đủ.

+ **Fuzzy Logic**: Fuzzy Logic hoạt động hiệu quả khi có một tập hợp các quy tắc rõ ràng và dữ liệu đầu vào không quá phức tạp. Tuy nhiên, khi dữ liệu đầu vào

quá mơ hồ hoặc có sự không chắc chắn, kết quả có thể không chính xác hoặc không đáng tin cậy.

### Đánh giá tổng quan

− **CNN**:

Ưu điểm:

+ Độ chính xác cao, đặc biệt trong các bài toán nhận dạng hình ảnh phức tạp.

+ Có khả năng học từ dữ liệu và tự động trích xuất đặc trưng.

+ Thích hợp cho các ứng dụng yêu cầu phân tích hình ảnh phức tạp.

Nhược điểm:

+ Yêu cầu tài nguyên tính toán cao và thời gian huấn luyện lâu.

+ Cần một lượng lớn dữ liệu huấn luyện.

+ Tốc độ xử lý chậm hơn so với các phương pháp đơn giản như fuzzy logic.

− **Fuzzy Logic**: Ưu điểm:

+ Tốc độ xử lý nhanh, đặc biệt khi các quy tắc và đầu vào rõ ràng.

+ Phù hợp với các bài toán có các quy tắc logic mờ đã được xác định trước.

+ Dễ triển khai và ít tốn tài nguyên tính toán.

Nhược điểm:

+ Độ chính xác thấp hơn khi đối mặt với các tình huống phức tạp hoặc dữ liệu không rõ ràng.

+ Không thể học từ dữ liệu mới, cần phải xác định trước các quy tắc.

+ Không hiệu quả khi xử lý các loại dữ liệu phức tạp hoặc có sự biến đổi lớn.

# KẾT LUẬN

Trong đề tài này, chúng em đã nghiên cứu và triển khai hai phương pháp nhận dạng cảm xúc từ khuôn mặt: **mạng nơ-ron tích chập (CNN)** và **logic mờ (Fuzzy Logic)**. Cả hai phương pháp đều mang lại những kết quả đáng chú ý, đồng thời thể hiện được ưu nhược điểm riêng biệt khi áp dụng vào bài toán nhận diện cảm xúc.

**CNN**: Học sâu các đặc trưng phức tạp, đạt độ chính xác cao, nhưng đòi hỏi tài nguyên lớn và thời gian xử lý dài.

**Fuzzy Logic**: Đơn giản, nhanh chóng, phù hợp với các hệ thống có tài nguyên hạn chế, nhưng phụ thuộc nhiều vào quy tắc thủ công.

Hạn chế chính gồm dữ liệu chưa đa dạng và CNN cần tối ưu hơn để tránh mất cân bằng.

Hướng phát triển:

− Tăng cường dữ liệu và sử dụng các mô hình CNN tiên tiến.

− Mở rộng hệ thống quy tắc và tích hợp fuzzy logic với học máy.

− Ứng dụng thực tiễn như nhận diện cảm xúc qua camera phục vụ giáo dục, an ninh, và kinh doanh.

Đề tài giúp chúng em hiểu rõ tiềm năng của hai phương pháp, đồng thời hướng đến việc kết hợp cả CNN và Fuzzy Logic để tối ưu hiệu quả.

# TÀI LIỆU THAM KHẢO

1. ThS. Trần Hùng Cường – Ths. Nguyễn Phương Nga, *Giáo trình Trí tuệ nhân tạo*, Nhà xuất bản Trường Đại học Công nghiệp Hà Nội.
2. N T Tuan (2019), *Convolutional neural network,* từ https://nttuan8.com/bai-6- convolutional-neural-network/.
3. To Duc Thang (2020), *Làm quen với Keras,* từ https://viblo.asia/p/lam-quen-voi- keras-gGJ59mxJ5X2.
4. Phạm Minh Quyền, Phùng Thanh Huy, Đỗ Duy Tân, Huỳnh Hoàng Hà, Trương Quang Phúc (2021), *Implementation of facial emotion recognition using cnn on jetson tx2,* Tạp Chí Khoa Học Giáo Dục Kỹ Thuật.
5. *Machine learning,* truy cập 15/11/2024, từ https://en.wikipedia.org/wiki/Machine\_learning.
6. what-is/deep-learning, truy cập từ 8/11/2024 từ https://aws.amazon.com/vi/what- is/deep-learning.
7. *Emotion Detection Using Convolutional Neural Networks (CNNs)* truy cập 05/12/2024, từ [https://www.geeksforgeeks.org/emotion-detection-using-convolutional-](http://www.geeksforgeeks.org/emotion-detection-using-convolutional-) neural-networks-cnns/

*Fuzzy Logic | Introduction* truy cập 04/12/2024, từ [https://www.geeksforgeeks.org/fuzzy-logic-introduction/](http://www.geeksforgeeks.org/fuzzy-logic-introduction/)