ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue button with white text

Description automatically generated

HỌC PHẦN ĐỒ ÁN CHUYÊN NGÀNH

**Phân tích thiết kế mạng xã hội**

Sinh viên thực hiện: Trương Gia Thành

Nguyễn Khánh Hòa

Nguyễn Hoàng Phương Vũ

Trần Tiến Thanh

Nguyễn Ngọc Tuấn Vũ

Giảng viên hướng dẫn: Trần Thanh Nhã

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2024

LỜI MỞ ĐẦU

Trước hết, nhóm chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy **Trần Thanh Nhã**, giảng viên bộ môn **Phân tích mạng truyền thông xã hội** tại Khoa Công nghệ Thông tin, trường Đại học Sài Gòn. Thầy không chỉ là người truyền đạt kiến thức mà còn là nguồn động lực lớn để nhóm chúng em vượt qua những thử thách trong quá trình thực hiện đồ án. Sự tận tâm, kiên nhẫn và những ý kiến đóng góp mang tính định hướng của thầy đã giúp nhóm em hiểu rõ hơn về vấn đề, đồng thời hoàn thiện đồ án một cách tốt nhất trong khả năng. Chúng em rất biết ơn vì sự hỗ trợ của thầy trong từng giai đoạn, từ việc xác định vấn đề, định hướng nghiên cứu, đến việc kiểm tra, chỉnh sửa nội dung và hình thức báo cáo.

Chúng em cũng xin gửi lời cảm ơn chân thành đến các thầy cô tại **trường Đại học Sài Gòn**, đặc biệt là các thầy cô trong **Khoa Công nghệ Thông tin**. Chính những bài giảng tận tâm của thầy cô, từ các môn đại cương giúp xây dựng nền tảng kiến thức cơ bản, đến các môn chuyên ngành trang bị kỹ năng chuyên sâu, đã mang đến cho chúng em một hành trang vững chắc. Kiến thức mà chúng em tích lũy được không chỉ dừng lại ở lý thuyết mà còn là nguồn cảm hứng để chúng em áp dụng vào thực tiễn, đặc biệt là trong đồ án này. Chúng em luôn trân trọng những nỗ lực không ngừng nghỉ của các thầy cô trong việc truyền đạt kiến thức, chia sẻ kinh nghiệm thực tế và khơi gợi niềm đam mê nghiên cứu trong lòng sinh viên.

Đồng thời, nhóm em cũng xin chân thành cảm ơn các bạn đồng hành, những người đã cùng nhóm chia sẻ những ý tưởng, hỗ trợ trong việc thu thập, xử lý dữ liệu và hoàn thiện báo cáo. Sự đoàn kết, tinh thần hợp tác và những ý kiến đóng góp từ các bạn đã giúp chúng em không ngừng tiến bộ.

Trong quá trình thực hiện đồ án, nhóm chúng em đã đối mặt với không ít khó khăn, từ việc lên kế hoạch, phân chia công việc, thu thập và xử lý dữ liệu, đến phân tích, trình bày và hoàn thiện báo cáo. Những khó khăn này   
không chỉ đến từ thời gian hạn hẹp hay lượng kiến thức cần áp dụng mà còn từ những kinh nghiệm thực tế chưa đủ chín chắn của nhóm. Tuy nhiên, nhờ sự hướng dẫn tận tình của thầy cô và sự hỗ trợ lẫn nhau trong nhóm, chúng em đã nỗ lực vượt qua và hoàn thành đồ án này.

Dù vậy, chúng em ý thức rằng đồ án của mình vẫn còn nhiều hạn chế và chưa thể đạt đến mức hoàn hảo. Do đó, chúng em rất mong nhận được những ý kiến đóng góp, nhận xét quý báu từ thầy cô và bạn bè để không ngừng học hỏi, hoàn thiện hơn trong tương lai. Những góp ý này không chỉ giúp cải thiện chất lượng đồ án mà còn là cơ hội để chúng em rèn luyện và phát triển bản thân, chuẩn bị tốt hơn cho những thử thách trong sự nghiệp sau này.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành nhất đến tất cả những ai đã đóng góp vào quá trình thực hiện đồ án này. Nhờ vào sự hướng dẫn, động viên và đồng hành của mọi người, chúng em đã có thể đi đến ngày hôm nay với một sản phẩm thể hiện sự cố gắng và tâm huyết của cả nhóm.

Xin kính chúc thầy cô luôn dồi dào sức khỏe, tràn đầy năng lượng và đạt được nhiều thành công hơn nữa trong sự nghiệp.

# MỤC LỤC

[MỤC LỤC 3](#_Toc184213744)

[Bảng phân chia công việc 5](#_Toc184213745)

[Chương 1. TỔNG QUAN VỀ ĐỒ ÁN 6](#_Toc184213746)

[1. Giới thiệu đề tài 6](#_Toc184213747)

[2. Lý do chọn đề tài 6](#_Toc184213748)

[1. Mở rộng kiến thức về trí tuệ nhân tạo và máy học 7](#_Toc184213749)

[2. Xu hướng công nghệ hiện đại 7](#_Toc184213750)

[3. Tính ứng dụng cao trong đời sống 8](#_Toc184213751)

[4. Tiếp cận tài nguyên công nghệ chất lượng cao 8](#_Toc184213752)

[5. Định hướng nghề nghiệp và phát triển tương lai 9](#_Toc184213753)

[6. Mục tiêu mong muốn đạt được 9](#_Toc184213754)

[Chương 2. NỘI DUNG LÝ THUYẾT 11](#_Toc184213755)

[1. Khái niệm cơ bản 11](#_Toc184213756)

[2. Trí tuệ nhân tạo và máy học 12](#_Toc184213757)

[3. Thuật toán sử dụng 13](#_Toc184213758)

[I. Mạng nơ-ron tích chập (CNN) 13](#_Toc184213759)

[4. Các công cụ và thư viện 20](#_Toc184213760)

[Chương 3. NỘI DUNG THỰC HÀNH 26](#_Toc184213761)

[1. Cách thu thập dữ liệu 26](#_Toc184213762)

[2. Tiền xử lí dữ liệu 31](#_Toc184213763)

[3. Input dataset 31](#_Toc184213764)

[4. Trình bày mô hình 32](#_Toc184213765)

[5. Đầu ra mô hình 48](#_Toc184213766)

[6. Xử lí ảnh và nhận diện khuôn mặt 51](#_Toc184213767)

[7. Cấu hình máy tính 56](#_Toc184213768)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 57](#_Toc184213769)

[TÀI LIỆU THAM KHẢO 59](#_Toc184213770)

# Bảng phân chia công việc

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Tên | Công Việc | % Góp |
| 3121410042 | Nguyễn Ngọc Tuấn Vũ | Thuyết Trình, tìm tài liệu | 90% |
| 3121410456 | Trương Gia Thành | Tìm tài liệu, làm pp, Kaggle build model nhận diện khuôn mặt, code python | 100% |
| 3121410026 | Nguyễn Khánh Hòa | Tìm tài liệu viết báo cáo | 85% |
| 3122410482 | Nguyễn Hoàng Phương Vũ | Code python nhận diện khuôn mặt, tìm tài liệu viết báo cáo | 90% |
| 3120410475 | Trần Tiến Thanh | Kaggle, build model nhận diện khuôn mặt | 90% |

# Chương 1. TỔNG QUAN VỀ ĐỒ ÁN

## **1. Giới thiệu đề tài**

Tên đề tài: Phân tích cảm xúc thông qua khuôn mặt

Phân tích cảm xúc thông qua khuôn mặt là một nhánh quan trọng của trí tuệ nhân tạo và xử lý hình ảnh, cho phép nhận diện và phân tích cảm xúc biểu lộ từ khuôn mặt của con người. Kỹ thuật này dựa vào các biểu hiện cơ mặt như cười, nhíu mày, nhăn trán để phân loại các trạng thái cảm xúc, thường được chia thành các nhóm chính: **vui vẻ**, **buồn bã**, **tức giận**, **ngạc nhiên**, **sợ hãi**, **ghê tởm** và **trung lập**.

## **2. Lý do chọn đề tài**

Nhóm em chọn đề tài Phân tích cảm xúc qua khuôn mặt bằng trí tuệ nhân tạo (AI) vì các lý do:

Trí tuệ nhân tạo (AI) là một trong những lĩnh vực tiên phong trong công nghệ hiện đại, đang từng bước thay đổi cách con người làm việc và tương tác với thế giới. Chính vì vậy, nhóm chúng em lựa chọn đề tài "Phân tích cảm xúc thông qua gương mặt bằng trí tuệ nhân tạo" với mong muốn học hỏi, mở rộng kiến thức và đóng góp vào xu hướng này.

### **1. Mở rộng kiến thức về trí tuệ nhân tạo và máy học**

AI, đặc biệt là lĩnh vực máy học (Machine Learning), là một trong những công nghệ cốt lõi của thời đại số. Việc nghiên cứu và triển khai đề tài này giúp nhóm tiếp cận và hiểu rõ hơn về cách AI hoạt động, từ việc xử lý dữ liệu, nhận diện khuôn mặt đến phân tích cảm xúc phức tạp. Đây là một cơ hội quý báu để nhóm trau dồi các kỹ năng lập trình, phân tích và làm việc với các thuật toán học sâu (Deep Learning) – những yếu tố quan trọng trong bất kỳ ứng dụng AI nào.

### **2. Xu hướng công nghệ hiện đại**

Phân tích khuôn mặt kết hợp AI đang trở thành xu hướng công nghệ toàn cầu. Các tập đoàn lớn như Google, Microsoft, Meta đã và đang đầu tư mạnh mẽ vào lĩnh vực này, minh chứng cho tiềm năng ứng dụng rộng rãi của công nghệ trong nhiều lĩnh vực như bảo mật, y tế, giáo dục và giải trí. Nhóm nhận thấy đây là cơ hội để không chỉ theo kịp xu hướng mà còn góp phần khám phá thêm những giá trị mà công nghệ này có thể mang lại.

### **3. Tính ứng dụng cao trong đời sống**

Công nghệ phân tích cảm xúc qua khuôn mặt không chỉ dừng lại ở nghiên cứu mà còn có khả năng ứng dụng thực tiễn cao. Nó có thể giúp cải thiện trải nghiệm khách hàng, hỗ trợ chẩn đoán tâm lý trong y học, và nâng cao hiệu quả giáo dục thông qua việc theo dõi cảm xúc học sinh. Việc nghiên cứu đề tài này mang đến tiềm năng đóng góp cho các lĩnh vực đang ngày càng cần sự chính xác và cá nhân hóa.

### **4. Tiếp cận tài nguyên công nghệ chất lượng cao**

Một trong những lợi thế lớn khi thực hiện đề tài này là khả năng sử dụng các nền tảng và tài nguyên hỗ trợ như Kaggle. Kaggle không chỉ cung cấp kho dữ liệu phong phú mà còn tạo điều kiện cho nhóm làm việc trong môi trường lập trình trực tuyến hiện đại (Kaggle Kernels). Điều này giúp nhóm không chỉ xây dựng kỹ năng chuyên môn mà còn làm quen với các công cụ và quy trình làm việc chuẩn quốc tế.

### **5. Định hướng nghề nghiệp và phát triển tương lai**

Lựa chọn nghiên cứu đề tài này cũng là một bước đi chiến lược trong việc định hướng nghề nghiệp. Với nhu cầu ngày càng cao về chuyên gia AI, đặc biệt trong các ứng dụng liên quan đến xử lý hình ảnh và cảm xúc, nhóm tin rằng kiến thức và kinh nghiệm thu được sẽ là nền tảng vững chắc để tiến xa hơn trong sự nghiệp.

Tóm lại, việc lựa chọn đề tài không chỉ xuất phát từ sự yêu thích công nghệ mà còn từ những giá trị thực tiễn và tiềm năng phát triển mà nó mang lại. Đề tài "Phân tích cảm xúc thông qua gương mặt bằng trí tuệ nhân tạo" không chỉ giúp nhóm mở rộng kiến thức mà còn đóng góp vào xu hướng phát triển công nghệ, đáp ứng nhu cầu thực tiễn của xã hội hiện đại.

### **6. Mục tiêu mong muốn đạt được**

#### **I. Mục tiêu tổng quát**

Xây dựng một hệ thống trí tuệ nhân tạo có khả năng tự động nhận diện và phân tích cảm xúc của con người thông qua biểu cảm khuôn mặt, góp phần cải thiện tương tác giữa con người và máy móc, đồng thời ứng dụng hiệu quả trong các lĩnh vực như giáo dục, y tế, an ninh và giải trí.

#### **II. Mục tiêu cụ thể**

Phát triển mô hình AI có khả năng nhận diện các trạng thái cảm xúc cơ bản (hạnh phúc, buồn bã, tức giận, ngạc nhiên, sợ hãi, trung lập) với độ chính xác cao.

Đảm bảo mô hình hoạt động hiệu quả trong thời gian thực, xử lý tốt các dữ liệu hình ảnh và video từ nhiều nguồn khác nhau.

Xây dựng bộ dữ liệu khuôn mặt có tính đa dạng cao về giới tính, độ tuổi, để đảm bảo mô hình có thể áp dụng trên phạm vi rộng.

# Chương 2. NỘI DUNG LÝ THUYẾT

## **1. Khái niệm cơ bản**

**Phân tích cảm xúc khuôn mặt** là quá trình nhận diện và phân loại các cảm xúc của con người thông qua biểu cảm khuôn mặt. Mỗi cảm xúc cơ bản mà con người thể hiện đều có những đặc trưng riêng biệt trên khuôn mặt, bao gồm các sự thay đổi trong cơ mặt, nhắm mắt, nhíu mày, mỉm cười, hoặc các cử động khác. Những cảm xúc này thường được chia thành một số nhóm cơ bản, chẳng hạn như hạnh phúc, tức giận, buồn bã, ngạc nhiên, sợ hãi, ghê tởm và trung lập.

Phân tích cảm xúc khuôn mặt chủ yếu sử dụng các phương pháp trong **thị giác máy tính** và **trí tuệ nhân tạo** (AI) để phát hiện các đặc điểm này và xác định cảm xúc người dùng thông qua các thuật toán học máy. Việc nhận diện cảm xúc qua khuôn mặt có thể giúp hệ thống hiểu và phản hồi lại hành vi người dùng, mang lại những trải nghiệm tương tác tốt hơn, đặc biệt trong các ứng dụng như chăm sóc sức khỏe, giáo dục, an ninh, và giải trí.

**Các cảm xúc cơ bản thường gặp bao gồm:**

* **Hạnh phúc**: Được nhận diện qua nụ cười, mắt híp lại, cơ mặt mềm mại.
* **Buồn bã**: Môi cong xuống, mắt trũng sâu, đầu cúi xuống.
* **Tức giận**: Lông mày nhíu lại, miệng mím chặt, cơ mặt căng thẳng.
* **Ngạc nhiên**: Mắt mở rộng, miệng mở rộng hoặc nhướng lên.
* **Sợ hãi**: Mắt mở to, miệng mở nhẹ, cơ mặt căng thẳng.
* **Ghê tởm**: Mũi nhăn lại, môi cong xuống, cơ mặt có biểu hiện phản ứng mạnh.
* **Trung lập**: Biểu cảm khuôn mặt không có sự thay đổi rõ rệt, cơ mặt thư giãn.

## **2. Trí tuệ nhân tạo và máy học**

**Trí tuệ nhân tạo (AI)** là một lĩnh vực trong khoa học máy tính, nghiên cứu và phát triển các hệ thống máy tính có thể thực hiện các tác vụ thông minh mà trước đây chỉ có con người mới làm được. AI bao gồm nhiều kỹ thuật khác nhau, trong đó **học máy (Machine Learning)** là một trong những phương pháp quan trọng, cho phép máy tính học từ dữ liệu mà không cần phải lập trình cụ thể.

**Học máy (Machine Learning - ML)** là một nhánh của AI, trong đó máy tính học các mẫu từ dữ liệu để đưa ra dự đoán hoặc quyết định. Các hệ thống học máy có thể học và cải thiện qua thời gian dựa trên các dữ liệu đầu vào.

**Học sâu (Deep Learning - DL)** là một phương pháp học máy đặc biệt, sử dụng mạng nơ-ron nhân tạo với nhiều lớp (deep neural networks). Học sâu đã cách mạng hóa nhiều lĩnh vực như nhận diện hình ảnh, nhận diện giọng nói và phân tích cảm xúc qua khuôn mặt, vì nó có khả năng học và trích xuất các đặc trưng từ dữ liệu mà không cần sự can thiệp của con người.

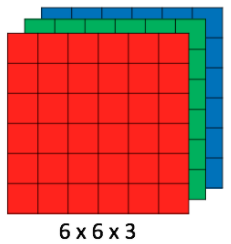
## **3. Thuật toán sử dụng**

### **I. Mạng nơ-ron tích chập (CNN)**

#### **a. Giới thiệu**

Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

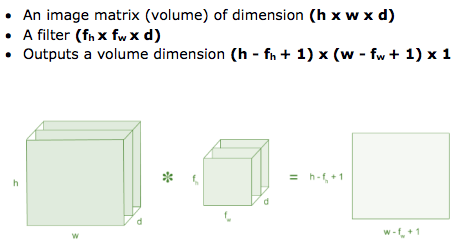
CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao,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



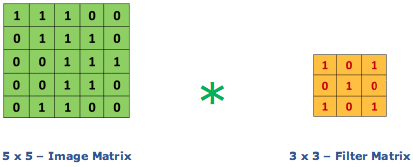
Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.

#### **b. Lớp tích chập - convolution layer**

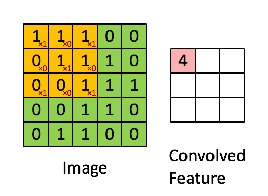
Tích chập là lớp đầu tiên để trích xuất các tính năng từ hình ảnh đầu vào. Tích chập duy trì mối quan hệ giữa các pixel bằng cách tìm hiểu các tính năng hình ảnh bằng cách sử dụng các ô vương nhỏ của dữ liệu đầu vào. Nó là 1 phép toán có 2 đầu vào như ma trận hình ảnh và 1 bộ lọc hoặc hạt nhân.



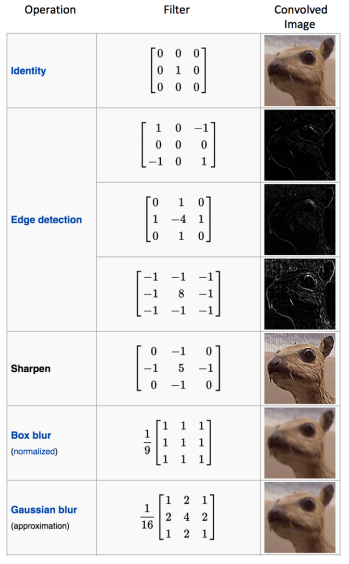
Sau đó, lớp tích chập của ma trận hình ảnh 5 x 5 nhân với ma trận bộ lọc 3 x 3 gọi là 'Feature Map' như hình bên dưới.



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.



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.



#### **c. Bước nhảy - stride**

Stride là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.

#### **d. Đường viền - padding**

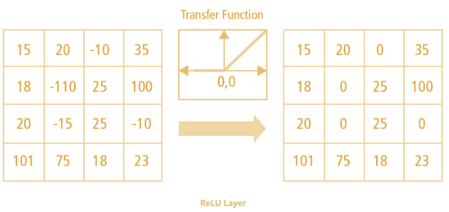
Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:

1. Chèn thêm các số 0 vào 4 đường biên của hình ảnh (padding)
2. Cắt bớt hình ảnh tại những điểm không phù hợp với kernel.

#### **e. Hàm phi tuyến - ReLU**

ReLU viết tắt của Rectified Linear Unit, là 1 hàm phi tuyến. Với đầu ra là: ƒ (x) = max (0, x).

Tại sao ReLU lại quan trọng: ReLU giới thiệu tính phi tuyến trong ConvNet. Vì dữ liệu trong thế giới mà chúng ta tìm hiểu là các giá trị tuyến tính không âm.



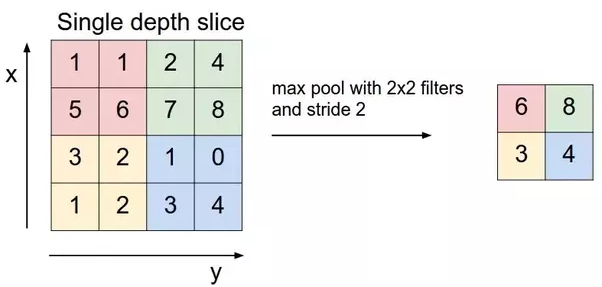
Có 1 số hà phi tuyến khác như tanh, sigmoid cũng có thể được sử dụng thay cho ReLU. Hầu hết người ta thường dùng ReLU vì nó có hiệu suất tốt.

#### **f. Lớp gộp - Pooling Layer**

Lớp pooling sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu xuống làm giảm kích thước của mỗi map nhưng vẫn giữ lại thông tin quan trọng. Các pooling có thể có nhiều loại khác nhau:

* Max Pooling
* Average Pooling
* Sum Pooling

Max pooling lấy phần tử lớn nhất từ ma trận đối tượng, hoặc lấy tổng trung bình. Tổng tất cả các phần tử trong map gọi là sum pooling



## **4. Các công cụ và thư viện**

Kaggle (Kaggle.com) là nền tảng trực tuyến nổi tiếng dành cho các nhà khoa học dữ liệu, chuyên gia máy học và cộng đồng yêu thích phân tích dữ liệu. Kaggle cung cấp môi trường làm việc và công cụ hỗ trợ mạnh mẽ để thực hiện các dự án khoa học dữ liệu mà không cần cấu hình phần mềm phức tạp. Những tính năng nổi bật của Kaggle bao gồm:  
**Kho dữ liệu phong phú**: Kaggle cung cấp hàng nghìn bộ dữ liệu miễn phí từ nhiều lĩnh vực, từ phân tích cảm xúc, xử lý ngôn ngữ tự nhiên, đến nhận diện khuôn mặt. Bạn có thể dễ dàng tìm kiếm và sử dụng các bộ dữ liệu này cho dự án của mình.

#### **Môi trường làm việc trực tuyến (Kaggle Notebooks):**

Cho phép bạn viết và chạy mã Python hoặc R ngay trên nền tảng.

Tích hợp sẵn các thư viện phổ biến như TensorFlow, PyTorch, NumPy, Pandas, và Matplotlib.

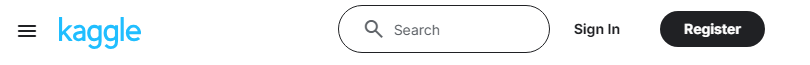
Hỗ trợ GPU/TPU miễn phí để tăng tốc huấn luyện các mô hình học sâu.

**Các cuộc thi khoa học dữ liệu**: Kaggle tổ chức nhiều cuộc thi phân tích dữ liệu với giải thưởng hấp dẫn. Đây là cơ hội để bạn học hỏi, thực hành và giao lưu với cộng đồng chuyên gia toàn cầu.

**Cộng đồng học hỏi**: Kaggle có một diễn đàn lớn, nơi bạn có thể chia sẻ kinh nghiệm, hỏi đáp, và tìm kiếm lời khuyên từ các chuyên gia.

Về cách sử dụng:

1. Đăng ký tài khoản miễn phí góc phải phía trên của trang



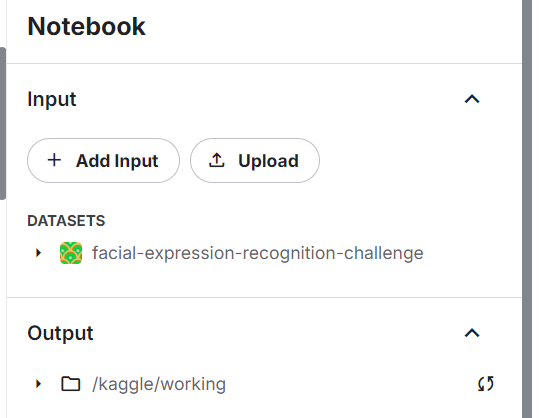
Sau khi đăng ký, bạn có thể truy cập vào toàn bộ dữ liệu, môi trường làm việc và các tài liệu học tập.

##### **Kaggle Notebooks**

Kaggle Notebooks là môi trường làm việc trực tuyến, cho phép bạn viết mã mà không cần cài đặt phần mềm trên máy tính. Tuy nhiên khi bạn mới bắt đầu tiếp xúc với Kaggle thì chúng ta nên chọn tìm kiếm những bài viết có sẵn để tham khảo hoặc chúng ta có thể chọn Copy and edit để có thể xuất ra 1 notebook riêng để ta có thể chỉnh sửa mà không có sự thay đổi gì đến bản gốc

###### Datasets

Kaggle sỡ hữu lượng lớn dữ liệu có sẵn trong các bài viết. Do đó chúng ta có thể dễ dàng tìm kiếm để tải xuống hoặc tiện hơn là thêm thẳng vào notebook **(Add Input)** để chúng ta có thể dễ dàng thực thi



##### **Python**

**Python** là một ngôn ngữ lập trình bậc cao, được thiết kế với triết lý đơn giản, dễ đọc, và dễ học. Được tạo ra bởi **Guido van Rossum** vào năm 1991, Python ngày càng phổ biến và được sử dụng rộng rãi trong các lĩnh vực như khoa học dữ liệu, học máy, trí tuệ nhân tạo, phát triển web, phân tích dữ liệu, và tự động hóa.

Tại sao Python được ưa chuộng?

###### **Cú pháp đơn giản:**

Python rất dễ học, ngay cả với người mới bắt đầu lập trình.

Cú pháp giống ngôn ngữ tự nhiên, giúp mã nguồn dễ đọc và bảo trì.

###### **Thư viện phong phú:**

Python có một hệ sinh thái mạnh mẽ với hàng ngàn thư viện như:

**NumPy**, **Pandas**: Xử lý dữ liệu.

**Matplotlib**, **Seaborn**: Vẽ biểu đồ và trực quan hóa dữ liệu.

**TensorFlow**, **PyTorch**, **scikit-learn**: Học máy và AI.

**OpenCV**, **dlib**: Xử lý ảnh và nhận diện khuôn mặt.

###### **Đa nền tảng:**

Python chạy được trên hầu hết các hệ điều hành như Windows, macOS, và Linux.

###### **Cộng đồng mạnh mẽ:**

-Python có một cộng đồng lớn, luôn sẵn sàng hỗ trợ.

-Tài liệu phong phú, diễn đàn hoạt động sôi nổi như Stack Overflow, Reddit.

###### **Dễ tích hợp và mở rộng:**

-Python dễ dàng tích hợp với các ngôn ngữ khác như C, C++, hoặc Java.

###### **Sử dụng trong nhiều lĩnh vực:**

* **Khoa học dữ liệu và học máy:** Tạo mô hình, phân tích dữ liệu.
* **Phát triển web:** Frameworks như Django, Flask.
* **Tự động hóa:** Tạo các công cụ và kịch bản tự động.
* **Xử lý ảnh và video:** Phân tích và nhận diện khuôn mặt, vật thể.

# Chương 3. NỘI DUNG THỰC HÀNH

## **1. Cách thu thập dữ liệu**

Phân tích cảm xúc thông qua khuôn mặt là một bài toán trong lĩnh vực xử lí hình ảnh và máy học (machine learning). Mục tiêu là để nhận diện và phân loại các trạng thái cảm xúc như: hạnh phúc (happy), buồn (sad), trung lập (neutral), ghê tởm (disgust), nóng giận (angry), ngạc nhiên (surprise), sợ hãi (fear).

Dữ liệu cần thu thập bao gồm:

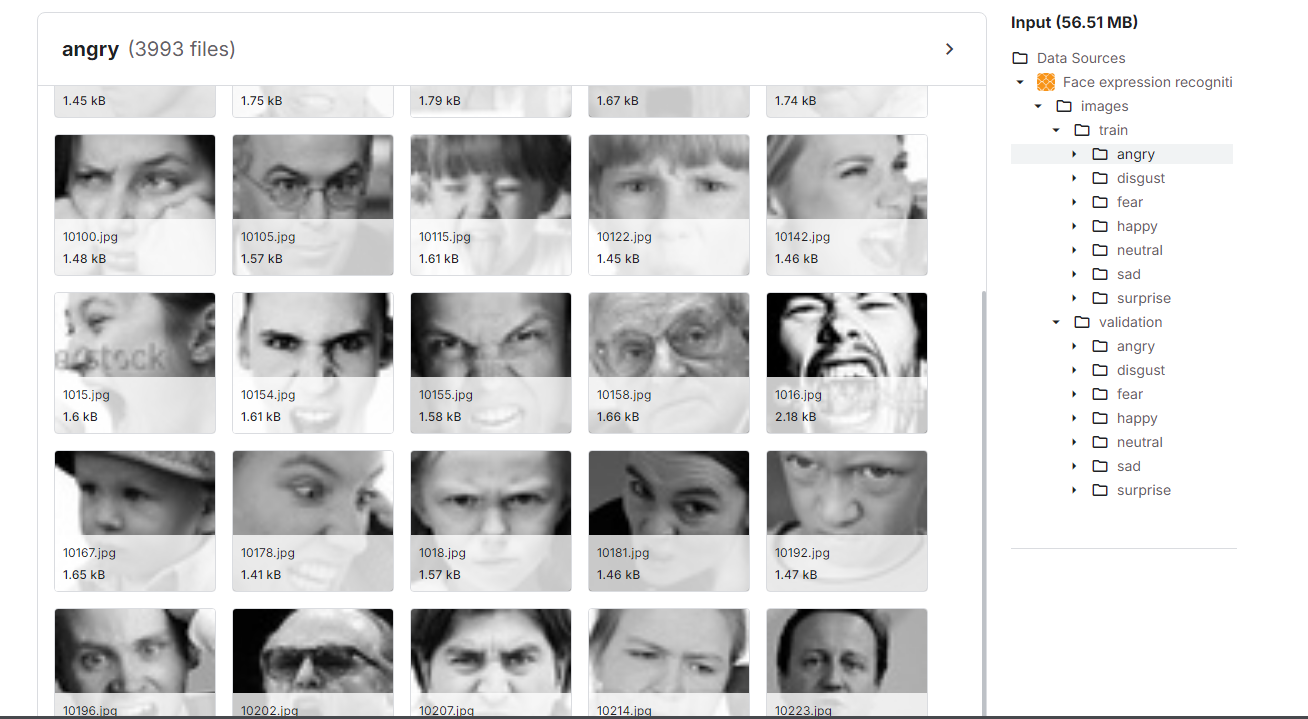
* Ảnh của khuôn mặt hiển thị rõ 7 cảm xúc
* Video ghi lại cảm xúc của người trong thời gian thực

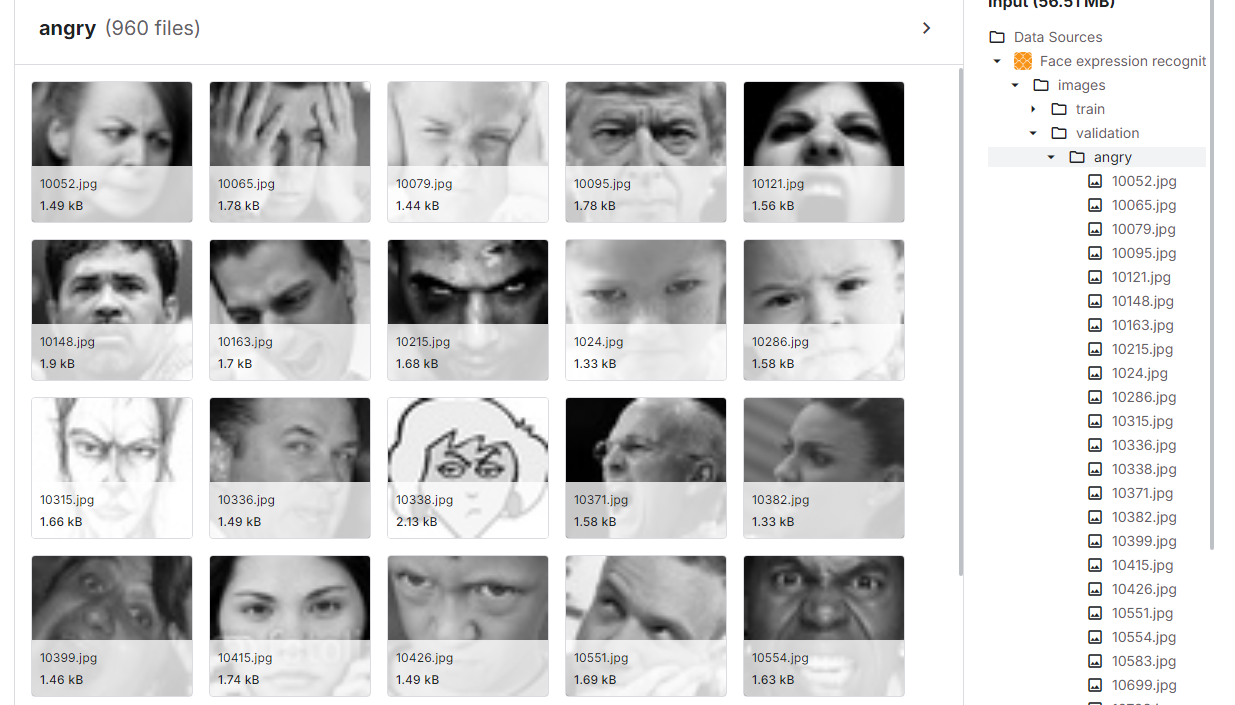
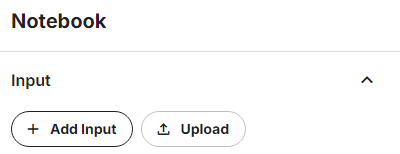
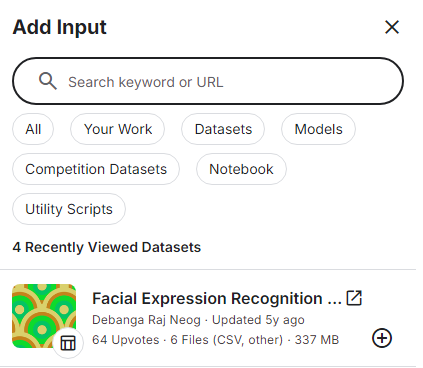
Cách thực hiện:

Bước 1: Bạn nên tạo sẵn 2 thư mục (folders) riêng biệt, một dùng để train và một dùng để test. Mỗi thư mục gồm 7 thư mục con chứa 7 cảm xúc khác nhau



Bước 2: Bạn có thể tải ảnh khuôn mặt từ các nguồn như Facebook, Google Image, Instagram,... . Sau đó nhét vào đúng folder trạng thái của khuôn mặt mà bạn cho là phù hợp. Ví dụ như folder Angry của bài phân tích cảm xúc khuôn mặt của tôi:

Bước 3: Làm folder test (ở đây folder của tôi là validation) tương tự với file train, tuy nhiên bạn không được để hình ảnh ở folder train và test giống nhau. Có thể ở folder test sẽ ít hơn hoặc nhiều hơn nhưng hãy chắc chắn là không được giống nhau.

Bước 4: Nếu bạn cảm thấy quá nhiều hình ảnh để thu thập thì bạn có để chọn add input vào search datasets để thêm thẳng vào bài code 

Hoặc là có thể tải datasets từ các nguồn khác như github, reddit, các diễn đàn công nghệ,... và chọn upload vào bài.

## **2. Tiền xử lí dữ liệu**

**1. Kiểm tra**

- Đảm bảo tất cả hình ảnh đều có định dạng hợp lệ (đuôi .jpg)

- Loại bỏ các hình ảnh bị lỗi hoặc không hiển thị được.

- Nên chọn hình ảnh biểu hiện cảm xúc rõ ràng của khuôn mặt.

**2. Định dạng kích thước hình ảnh**

- Resize tất cả ảnh về cùng kích thước 48x48 pixel theo yêu cầu của mô hình.

## **3. Input dataset**

|  |  |  |  |
| --- | --- | --- | --- |
|  | File train | File validation | Tỉ lệ (train/validation) |
| Angry | 3993 files .jpg | 960 files .jpg | 0,240420736 |
| Disgust | 436 files .jpg | 111 files .jpg | 0,254587156 |
| Happy | 7164 files .jpg | 1825 files .jpg | 0,254745952 |
| Sad | 4938 files .jpg | 1139 files .jpg | 0.230660186 |
| Surprise | 3205 files .jpg | 797 files .jpg | 0,248673947 |
| Fear | 4103 files .jpg | 1018 files .jpg | 0,248111138 |
| Neutral | 4982 files .jpg | 1216 files .jpg | 0,244078683 |
| Tổng cộng | 28821 files .jpg | 7066 files .jpg | 0,245168454 |

Tỉ lệ 24,51%

## **4. Trình bày mô hình**

Mô hình bao gồm 7 lớp:

- Importing Libraries



Lớp trên chuẩn bị các **thư viện cần thiết** để thực hiện một bài toán Deep Learning, cụ thể là xử lý dữ liệu hình ảnh và xây dựng một mô hình mạng nơ-ron tích chập (CNN).

Mục đích chính của code:

1. **Phân tích và trực quan hóa dữ liệu** (hình ảnh hoặc bảng dữ liệu).
2. **Xử lý dữ liệu hình ảnh**, bao gồm đọc ảnh, chuyển đổi sang dạng ma trận và áp dụng kỹ thuật tăng cường dữ liệu (data augmentation).
3. **Xây dựng mô hình Deep Learning**, sử dụng các lớp như Conv2D, Dense, Dropout, MaxPooling để phát hiện và phân loại đặc trưng từ hình ảnh.
4. **Sử dụng các thuật toán tối ưu hóa** (Adam, SGD, RMSprop) để cải thiện hiệu suất trong quá trình huấn luyện.

Nó là phần chuẩn bị cơ bản cho các bài toán như **phân loại cảm xúc từ khuôn mặt**, **phân loại hình ảnh**, hoặc các tác vụ liên quan đến thị giác máy tính.

- Displaying Images



Đoạn code trên thực hiện việc **hiển thị hình ảnh mẫu** từ tập dữ liệu liên quan đến **nhận diện cảm xúc trên khuôn mặt**.

Chức năng chính:

* **Đọc và hiển thị 9 hình ảnh** thuộc lớp cảm xúc cụ thể (ở đây là **"disgust"**, tức giận dữ) từ thư mục chứa dữ liệu huấn luyện.
* Sử dụng thư viện matplotlib để tạo lưới 3x3 và hiển thị các hình ảnh đã được resize về kích thước chuẩn *48×4848 \times 48*48×48 (theo biến picture\_size).

Ý nghĩa:

* Hỗ trợ **kiểm tra dữ liệu đầu vào**, đảm bảo các hình ảnh trong tập dữ liệu được xử lý và phân loại đúng trước khi đưa vào mô hình học máy.

- Making Training and Validation Data



Đoạn code trên xử lý **chuẩn bị dữ liệu huấn luyện và kiểm tra** cho mô hình Deep Learning bằng cách:

Chức năng chính:

1. **Tạo generator dữ liệu:**
   1. **ImageDataGenerator**: Cung cấp dữ liệu hình ảnh cho mô hình dưới dạng batch (nhóm), giúp giảm tải bộ nhớ và hỗ trợ xử lý dữ liệu lớn.
2. **Định nghĩa tập huấn luyện (train\_set) và kiểm tra (test\_set):**
   1. **Dữ liệu từ thư mục train và validation:** Đọc hình ảnh từ các thư mục tương ứng.
   2. **Chuyển đổi hình ảnh:**
      1. Resize tất cả hình ảnh về kích thước *48×48×48 \times 48×*48×48.
      2. Sử dụng chế độ màu xám (grayscale) để xử lý dữ liệu ảnh đen trắng.
   3. **Phân loại:** Ánh xạ nhãn (label) vào định dạng **one-hot encoding** với class\_mode='categorical'.
   4. **Shuffle:** Xáo trộn dữ liệu trong tập huấn luyện (shuffle=True), nhưng không xáo trộn dữ liệu kiểm tra (shuffle=False).

Ý nghĩa:

Đoạn code này **chuẩn bị dữ liệu đầu vào** để mô hình có thể huấn luyện và đánh giá hiệu quả trên tập dữ liệu được định dạng sẵn.

- Model Building



Đoạn code này nói về việc **xây dựng một mô hình mạng nơ-ron tích chập (CNN)** để giải quyết bài toán **phân loại hình ảnh**. Cụ thể, nó được thiết kế để **phân loại 7 lớp cảm xúc trên khuôn mặt**, dựa trên dữ liệu ảnh xám có kích thước 48×48.

Vấn đề chính:

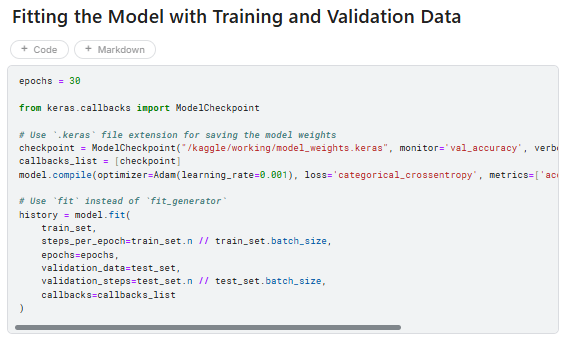
1. **Bài toán cần giải quyết:**
   1. Phân loại hình ảnh khuôn mặt thành 7 loại cảm xúc khác nhau, chẳng hạn như "vui", "buồn", "giận dữ", "sợ hãi", v.v.
2. **Phương pháp giải quyết:**
   1. Sử dụng mô hình **CNN**, một loại mạng nơ-ron chuyên xử lý dữ liệu hình ảnh, để tự động trích xuất và học các đặc trưng từ hình ảnh.
3. **Quy trình thực hiện:**
   1. Xây dựng các **lớp tích chập (Conv2D)** để trích xuất đặc trưng.
   2. Sử dụng các lớp như **BatchNormalization, MaxPooling, Dropout** để ổn định và tối ưu quá trình huấn luyện.
   3. Cuối cùng, thêm các lớp fully connected (Dense) để thực hiện phân loại.

Ứng dụng:

Đoạn code này có thể áp dụng cho các bài toán thực tế như:

* **Nhận diện cảm xúc trên khuôn mặt.**
* **Phân loại hình ảnh trong thị giác máy tính.**
* **Hệ thống tương tác thông minh**, ví dụ chatbot hoặc ứng dụng dựa trên nhận diện cảm xúc.

- Fitting the Model with Training and Validation Data



Đoạn code này dùng để **huấn luyện mô hình học sâu (Deep Learning model)** với dữ liệu huấn luyện và kiểm tra, đồng thời lưu lại trọng số của mô hình trong quá trình huấn luyện.

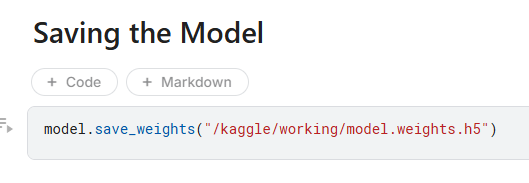
Các bước trong đoạn code:

1. **Cài đặt số lượng epoch (lần lặp) cho quá trình huấn luyện:**
   1. epochs = 30: Số lần huấn luyện mô hình trên toàn bộ dữ liệu.
2. **Sử dụng ModelCheckpoint từ Keras:**
   1. ModelCheckpoint là một callback giúp lưu lại trọng số của mô hình ở các điểm kiểm tra, như khi mô hình đạt được độ chính xác (accuracy) tốt nhất trên tập xác nhận (validation set).
   2. **monitor='val\_accuracy'**: Theo dõi độ chính xác trên tập xác nhận.
   3. **verbose=1**: In ra thông tin khi lưu trọng số.
   4. **save\_best\_only=True**: Chỉ lưu trọng số khi mô hình cải thiện độ chính xác trên tập xác nhận.
3. **Biên dịch mô hình:**
   1. **model.compile**: Định nghĩa thuật toán tối ưu (Adam với learning rate là 0.001), hàm mất mát (categorical crossentropy cho bài toán phân loại đa lớp), và các chỉ số (accuracy) để đánh giá mô hình trong quá trình huấn luyện.
4. **Huấn luyện mô hình với model.fit:**
   1. **train\_set**: Dữ liệu huấn luyện.
   2. **steps\_per\_epoch=train\_set.n // train\_set.batch\_size**: Số bước huấn luyện mỗi epoch, phụ thuộc vào số lượng mẫu và batch size.
   3. **validation\_data=test\_set**: Dữ liệu kiểm tra dùng để đánh giá mô hình sau mỗi epoch.
   4. **validation\_steps=test\_set.n // test\_set.batch\_size**: Số bước kiểm tra cho mỗi epoch.
   5. **callbacks=callbacks\_list**: Đưa vào ModelCheckpoint để lưu trọng số của mô hình.

Mục đích:

* Đoạn code này giúp huấn luyện mô hình, theo dõi hiệu suất của mô hình trên dữ liệu xác nhận và lưu lại trọng số của mô hình khi đạt được kết quả tốt nhất.

Saving the Model



Lớp này thực hiện việc **lưu trọng số của mô hình** vào một tệp tin. Cụ thể, nó sử dụng hàm save\_weights() của Keras để lưu các trọng số đã học được trong quá trình huấn luyện mô hình vào một tệp với định dạng .h5.

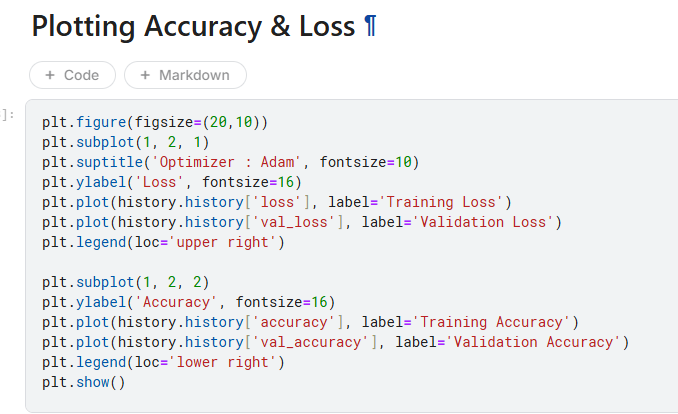
Giải thích chi tiết:

* **model.save\_weights("/kaggle/working/model.weights.h5")**:
  + **model**: Đây là mô hình đã được huấn luyện (ví dụ mô hình học sâu).
  + **save\_weights()**: Hàm này lưu lại trọng số của mô hình vào một tệp. Trọng số (weights) là các giá trị được điều chỉnh trong quá trình huấn luyện để tối ưu hóa mô hình.
  + **"/kaggle/working/model.weights.h5"**: Đây là đường dẫn và tên tệp lưu trữ trọng số của mô hình. Định dạng tệp .h5 là chuẩn để lưu trữ các trọng số trong Keras.

Mục đích:

* **Lưu trữ trọng số của mô hình** để có thể sử dụng lại sau này mà không cần phải huấn luyện lại từ đầu. Bạn có thể tải trọng số này về và áp dụng vào mô hình tương tự, giúp tiết kiệm thời gian huấn luyện và làm việc với mô hình đã học.

- Plotting Accuracy & Loss

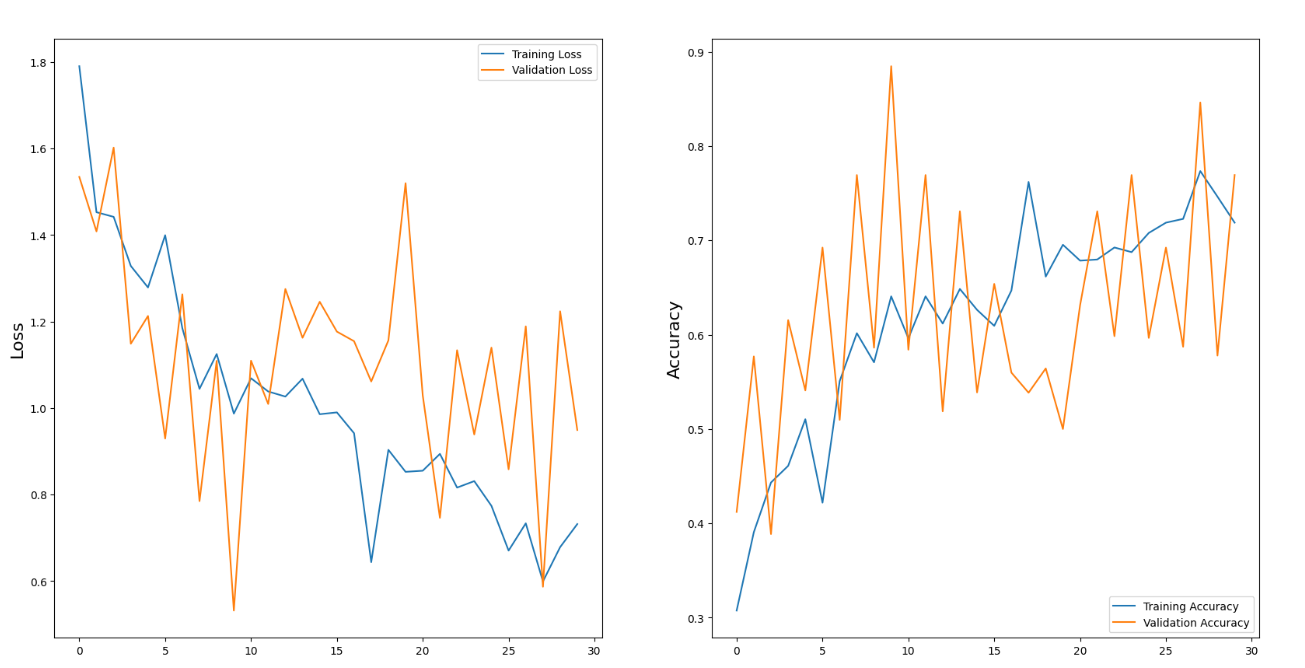


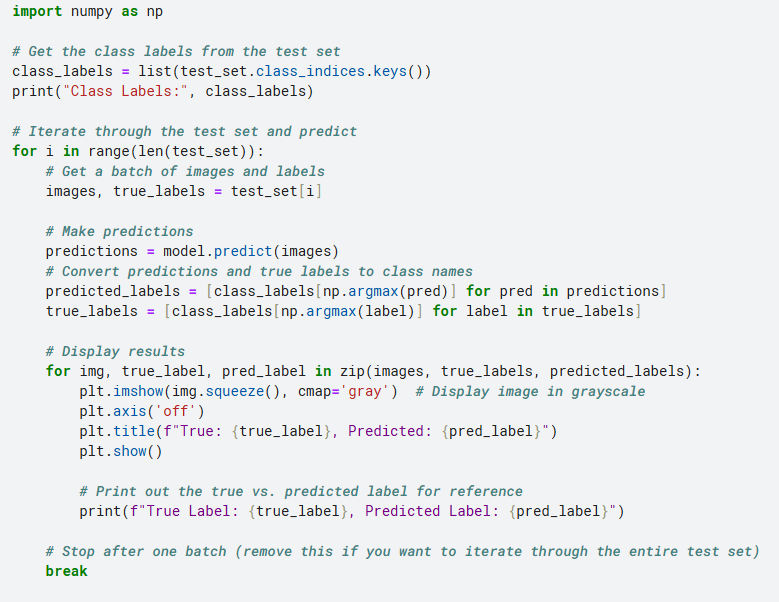
Đoạn code này vẽ đồ thị để **hiển thị quá trình huấn luyện của mô hình** trong suốt các epoch, bao gồm:

1. **Đồ thị 1 (Loss)**:
   1. **Hiển thị Loss** của mô hình trong quá trình huấn luyện và kiểm tra (validation) theo từng epoch.
   2. history.history['loss']: Loss của tập huấn luyện.
   3. history.history['val\_loss']: Loss của tập kiểm tra.
   4. Đặt tiêu đề cho đồ thị là "Optimizer: Adam", và chú thích trục Y là "Loss".
2. **Đồ thị 2 (Accuracy)**:
   1. **Hiển thị Accuracy** của mô hình trong quá trình huấn luyện và kiểm tra theo từng epoch.
   2. history.history['accuracy']: Accuracy của tập huấn luyện.
   3. history.history['val\_accuracy']: Accuracy của tập kiểm tra.
   4. Đặt tiêu đề trục Y là "Accuracy".

Mục đích:

* Giúp theo dõi hiệu suất của mô hình qua các epoch, so sánh độ mất mát (loss) và độ chính xác (accuracy) giữa tập huấn luyện và tập kiểm tra, từ đó đánh giá sự cải thiện của mô hình.





Đoạn code này thực hiện các bước sau:

1. **Lấy nhãn lớp từ tập kiểm tra** (test\_set.class\_indices.keys()).
2. **Duyệt qua từng batch trong tập kiểm tra**, thực hiện dự đoán cho các hình ảnh trong batch đó.
3. **Dự đoán nhãn của các hình ảnh** bằng mô hình đã được huấn luyện.
4. **Chuyển các dự đoán và nhãn thật** thành tên lớp tương ứng từ class\_labels.
5. **Hiển thị hình ảnh** cùng với nhãn thật và nhãn dự đoán, sau đó in ra thông tin về nhãn thật và nhãn dự đoán.

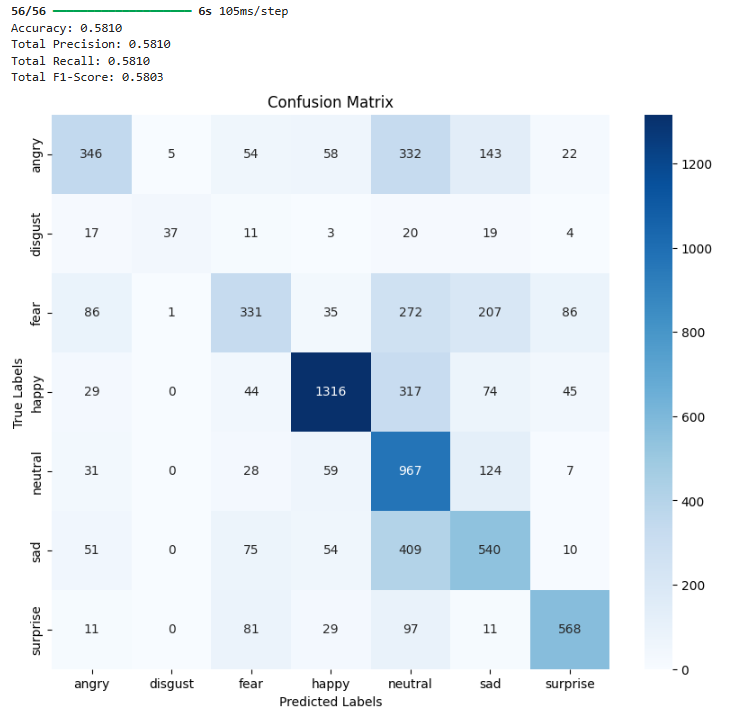
Mục đích của đoạn code này là **hiển thị kết quả dự đoán của mô hình** đối với một batch dữ liệu trong tập kiểm tra, so sánh nhãn thật và nhãn dự đoán.



Đoạn code này thực hiện các bước sau để đánh giá hiệu suất của mô hình trên tập kiểm tra:

1. **Dự đoán nhãn cho tập kiểm tra** bằng mô hình (model.predict(test\_set)), sau đó chuyển đổi kết quả thành nhãn lớp bằng cách sử dụng np.argmax.
2. **Tính độ chính xác** (accuracy) giữa nhãn thật và nhãn dự đoán bằng accuracy\_score, sau đó in ra giá trị độ chính xác.
3. **Tạo báo cáo phân loại** (classification report) để tính toán các chỉ số như độ chính xác, độ hồi tưởng (recall), và F1-score cho từng lớp, đồng thời in các chỉ số tổng quát (precision, recall, F1).
4. **Tính ma trận nhầm lẫn** (confusion matrix) giữa nhãn thật và nhãn dự đoán, và hiển thị nó dưới dạng heatmap bằng seaborn và matplotlib.

Mục đích của đoạn code này là **đánh giá mô hình bằng các chỉ số phân loại** (accuracy, precision, recall, F1-score) và hiển thị ma trận nhầm lẫn để quan sát các dự đoán chính xác và sai của mô hình.

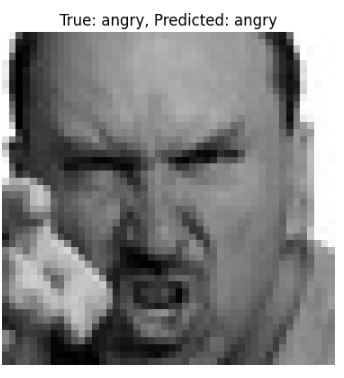


## **5. Đầu ra mô hình**

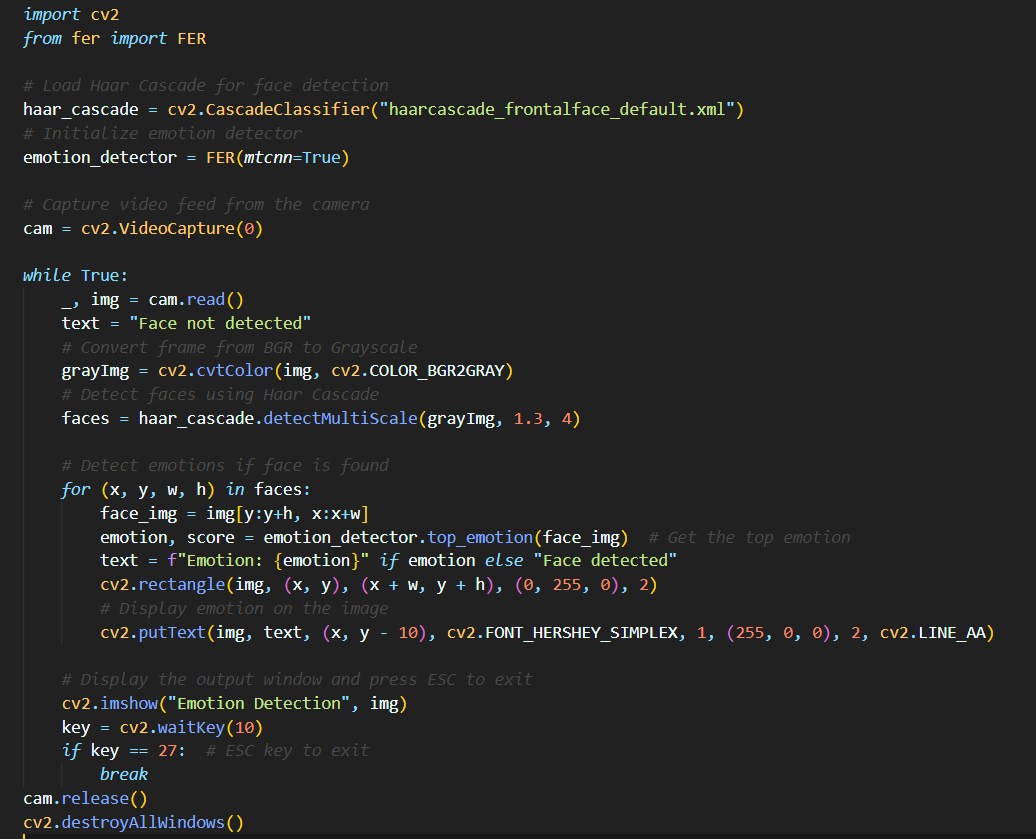








## **6. Xử lí ảnh và nhận diện khuôn mặt**



+ Công nghệ và thư viện sử dụng:

1. Python: Ngôn ngữ lập trình chính

2. OpenCV (cv2):

- Một thư viện mã nguồn mở cho xử lí ảnh và video

- Được dùng để xử lí webcam, phát hiện khuôn mặt, và vẽ hộp xung quanh khuôn mặt.

3. FER (Facial Expression Recognition):

- Một thư viện nhận diện cảm xúc khuôn mặt

- Dùng để phân tích biểu cảm khuôn mặt và trả về cảm xúc như hạnh phúc, buồn, giận dữ,...

4. Haar Cascade Classifier

- Mô hình học máy dùng để phát hiện khuôn mặt trong ảnh hoặc video

+ Chức năng chính

1. Mở webcam và đọc video theo thời gian thực:

- Sử dụng cv2.VideoCapture(0) để truy cập camera

- Đọc từng khung hình (frame) từ video

1. Phát hiện khuôn mặt:

- Sử dụng tệp haarcascade\_frontalface\_default.xml chứ mô hình học máy Haar Cascade để phát hiện khuôn mặt trong ảnh hoặc video

1. Nhận diện cảm xúc khuôn mặt

- Thư viện FER được dùng để phân tích từng khuôn mặt trong ảnh và xác định cảm xúc chính

1. Hiển thị kết quả

- Nếu phát hiện khuôn mặt, sẽ vẽ hình chữ nhật xung quanh và hiển thị cảm xúc trên khung hình.

- Dùng OpenCV để hiển thị khung video với kết quả cảm xúc

5. Thoát ứng dụng:

- Người dùng nhấn phím ESC để dừng camera và thoát

+ Ý nghĩa của từng đoạn trong mã

1. Tải mô hình phát hiện khuôn mặt (Haar Cascade):

haar\_cascade = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")

* Đây là một mô hình Haar Cascade được huấn luyện trước để phát hiện khuôn mặt

2. Khởi tạo FER để nhận diện cảm xúc:

emotion\_detector = FER(mtcnn=True)

* + FER sử dụng mạng nơ-ron tích chập (CNN) để nhận diện cảm xúc từ khuôn mặt.
  + mtcnn=True: Sử dụng MTCNN (Multi-task Cascaded Convolutional Networks) để phát hiện khuôn mặt, cải thiện độ chính xác.

1. Đọc từ webcam:

cam = cv2.VideoCapture(0)

* Khởi động camera và lấy từng khung hình để xử lý.

4. Phát hiện khuôn mặt trong từng khung hình:

faces = haar\_cascade.detectMultiScale(grayImg, 1.3, 4)

* + detectMultiScale() tìm tất cả khuôn mặt trong ảnh.
  + Các tham số:
* 1.3: Tỷ lệ scale ảnh để tìm khuôn mặt.
* 4: Số bước cần thiết để xác nhận khuôn mặt.

5. Phân tích cảm xúc từng khuôn mặt:

emotion, score = emotion\_detector.top\_emotion(face\_img)

* Lấy ảnh của từng khuôn mặt và dự đoán cảm xúc

1. Hiển thị cảm xúc và vẽ khung chữ nhật xung quanh khuôn mặt

cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.putText(img, text, (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE\_AA)

* + cv2.rectangle(): Vẽ hình chữ nhật xung quanh khuôn mặt.
  + cv2.putText(): Hiển thị cảm xúc dự đoán trên khung hình.

1. Hiển thị và thoát ứng dụng:

cv2.imshow("Emotion Detection", img)

key = cv2.waitKey(10)

if key == 27: # ESC key

Break

* + Hiển thị khung video đã xử lý.
  + Nhấn phím ESC để dừng chương trình.

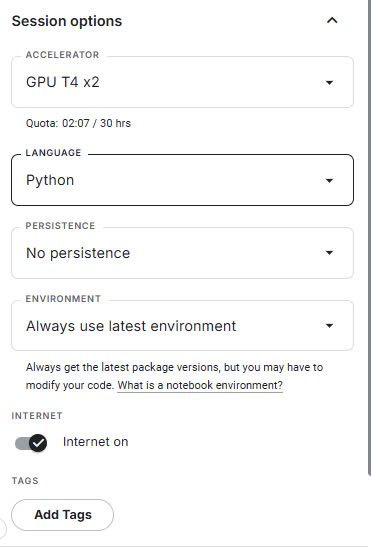
+ Ứng dụng thực tế:

* + **Giáo dục**: Theo dõi cảm xúc của học sinh trong lớp học trực tuyến.
  + **Bán lẻ**: Đánh giá phản hồi cảm xúc của khách hàng khi xem sản phẩm.
  + **Y tế**: Theo dõi trạng thái cảm xúc để hỗ trợ trị liệu tâm lý.
  + **Truyền thông**: Phân tích cảm xúc khán giả khi xem nội dung.

## **7. Cấu hình máy tính**

Chỉ cần máy chạy được + 8gb ram.

Cấu hình Kaggle, cần phải verify account Kaggle thì mới có thể chỉnh được Session options.



# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Kết luận:**

Đề tài "Phân tích cảm xúc thông qua khuôn mặt bằng trí tuệ nhân tạo" là một nghiên cứu quan trọng trong lĩnh vực AI và xử lý hình ảnh, với tiềm năng ứng dụng rộng rãi trong các ngành như y tế, giáo dục, an ninh và giải trí. Việc phát triển hệ thống nhận diện và phân tích cảm xúc thông qua biểu cảm khuôn mặt không chỉ góp phần cải thiện tương tác giữa con người và máy móc mà còn giúp tăng cường trải nghiệm người dùng trong nhiều lĩnh vực khác nhau. Nhóm đã đưa ra lý do hợp lý để lựa chọn đề tài này, bao gồm cơ hội mở rộng kiến thức về AI, nghiên cứu các công nghệ tiên tiến và ứng dụng thực tiễn trong đời sống, cùng với khả năng sử dụng các công cụ hỗ trợ hiện đại như Kaggle.

**Hướng phát triển trong tương lai:**

1. **Cải thiện độ chính xác của mô hình**: Trong tương lai, nhóm có thể tập trung vào việc cải thiện độ chính xác của mô hình AI, đặc biệt trong việc phân loại các trạng thái cảm xúc phức tạp hơn hoặc các biểu cảm khuôn mặt tinh tế mà hệ thống hiện tại có thể chưa nhận diện được chính xác.
2. **Mở rộng bộ dữ liệu**: Cần xây dựng bộ dữ liệu đa dạng hơn về các yếu tố như chủng tộc, văn hóa và môi trường, để mô hình có thể hoạt động hiệu quả trên một phạm vi rộng hơn, đáp ứng tốt nhu cầu của người dùng toàn cầu.
3. **Ứng dụng trong các lĩnh vực mới**: Phát triển ứng dụng công nghệ phân tích cảm xúc trong các lĩnh vực như chăm sóc sức khỏe tâm lý, hỗ trợ người khuyết tật, cải thiện giao tiếp trong môi trường học tập từ xa, và hỗ trợ các hệ thống an ninh, nhận diện đối tượng trong các tình huống phức tạp.
4. **Nâng cao khả năng nhận diện trong điều kiện thực tế**: Hệ thống có thể được cải thiện để nhận diện cảm xúc trong môi trường không hoàn hảo, như trong ánh sáng yếu hoặc với các yếu tố che khuất khuôn mặt.
5. **Tăng cường sự tương tác người-máy**: Tích hợp hệ thống với các giao diện người dùng (UI) thông minh, tạo ra những phản hồi tức thời dựa trên cảm xúc của người dùng, giúp các ứng dụng như trò chuyện ảo, trợ lý AI hay trò chơi điện tử trở nên tinh tế và gần gũi hơn với người dùng.
6. **Nghiên cứu và phát triển các mô hình AI đa nhiệm**: Phát triển các mô hình có khả năng phân tích không chỉ cảm xúc mà còn các yếu tố khác như động lực học khuôn mặt, phân tích ngữ nghĩa trong các tình huống giao tiếp phức tạp.

Nhóm nghiên cứu có thể tiếp tục khám phá những hướng đi mới này, giúp công nghệ phân tích cảm xúc thông qua khuôn mặt ngày càng hoàn thiện và được ứng dụng rộng rãi hơn trong xã hội.

# TÀI LIỆU THAM KHẢO

1. **Nguyễn Tuấn Anh - Bùi Mạnh Hùng - Lê Thị Minh Phương**, “ *Phương Pháp Nghiên Cứu Khoa Học*”, Nhà Xuất bản Xây Dựng/Trường Đại học Kiến Trúc Hà Nội, Năm xuất bản 2021.
2. **Chung Pham Van** - [[Deep Learning] Tìm hiểu về mạng tích chập (CNN)](https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2) 10/12/2020
3. **Kama** - [Network centrality using networkx](https://www.kaggle.com/code/rahulgoel1106/network-centrality-using-networkx) – 2022
4. **Farneet Singh -** [CK+ Facial Emotion Recognition - 96.46% Accuracy](https://www.kaggle.com/code/farneetsingh24/ck-facial-emotion-recognition-96-46-accuracy) – 12/2023
5. **Lê Công Trí -** [**Emotion Classification: CNN using Keras**](https://www.kaggle.com/code/lcngtr/emotion-classification-cnn-using-keras) **- 7/2021**