**TRƯỜNG ĐẠI HỌC PHENIKAA   
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**Học Phần: *Mạng Noron và học sâu***

Đề tài: FACE EMOTION DETECTION

|  |  |  |
| --- | --- | --- |
| *Giảng viên hướng dẫn* | : | TS.Phạm Tiến Lâm |
| *Sinh viên thực hiện* | : | 19010042 – Phạm Ngọc Việt  19010002 – Trịnh Tuấn Anh |
|  |  | 19010000 – Lê Hữu Trung |
|  |  |  |
|  |  |  |

HÀ NỘI – 201...

**Mục Lục**

[**MỞ ĐẦU 4**](#_Toc77250837)

[**Chương 1: Giới thiệu đề tài 5**](#_Toc77250838)

[**1.1. Tên đề tài 5**](#_Toc77250839)

[**1.2. Mục tiêu và nội dung. 5**](#_Toc77250840)

[**1.2.1. Mục tiêu 5**](#_Toc77250841)

[**1.2.2. Nội dung 5**](#_Toc77250842)

[**1.3. Thành viên 5**](#_Toc77250843)

[**1.4. Phương thức thực hiện và kết quả 5**](#_Toc77250844)

[**1.4.1. Phương thức thực hiện 5**](#_Toc77250845)

[**1.4.2. Kết quả 5**](#_Toc77250846)

[**Chương 2: Tổng quan cơ sở lý thuyết và xử lý dữ liệu ảnh 6**](#_Toc77250847)

[**2.1. Ngôn ngữ lập trình Python 6**](#_Toc77250848)

[**Một số tính chất của Python 6**](#_Toc77250849)

[**Tính năng 6**](#_Toc77250850)

[**2.2. Tổng quan về thư viện Scikit - learn trong Python 7**](#_Toc77250851)

[**2.2.1. Khái niệm 7**](#_Toc77250852)

[**2.2.2. Thuật toán 8**](#_Toc77250853)

[**2.3. Tổng quan về Deep Learning và bài toán nhận diện trong thị giác máy tính 8**](#_Toc77250854)

[**2.3.1. Mạng nơron 8**](#_Toc77250855)

[**2.3.2. Định nghĩa Deep Learning 9**](#_Toc77250856)

[**2.3.3. Cách thức hoạt động của Deep Learning 10**](#_Toc77250857)

[**2.3.4. Các ưu điểm của Deep Learning 10**](#_Toc77250858)

[**2.3.5. Ứng dụng của Deep Learning 11**](#_Toc77250859)

[**2.3.6. Hạn chế của Deep Learning 12**](#_Toc77250860)

[**2.4. Thư viện Keras 12**](#_Toc77250861)

[**2.4.1. Giới thiệu thư viện Keras 12**](#_Toc77250862)

[**2.4.2. Một số tính năng của Keras 12**](#_Toc77250863)

[**2.5. Giới thiệu Convolutional Neural Network (CNN) 13**](#_Toc77250864)

[**2.5.1. Tổng quát 13**](#_Toc77250865)

[**2.5.2. Các lớp của CNN 13**](#_Toc77250866)

[**2.5.3. Cấu trúc của CNN 14**](#_Toc77250867)

[**2.6. Giới thiệu bài toán Object Detection 15**](#_Toc77250868)

[**2.7 Xử lý dữ liệu ảnh 16**](#_Toc77250869)

[**2.7.1.  Thư viện cần sử dụng 16**](#_Toc77250870)

[**2.7.2. Phương pháp sử dụng 17**](#_Toc77250871)

[**Chương 3: Bài toán nhận diện cảm xúc 19**](#_Toc77250872)

[**3.1. Giới thiệu 19**](#_Toc77250873)

[**3.2. Xác định bài toán 19**](#_Toc77250874)

[**Chương 4: Xây dựng các mô hình. 20**](#_Toc77250875)

[**4.1. Chuẩn bị dữ liệu. 20**](#_Toc77250876)

[**4.2. Mô hình đã xây dựng 21**](#_Toc77250877)

[**4.2.1. Sử dụng CNN Sequential với SparseCategoricalCrossentropy 21**](#_Toc77250878)

[**4.2.2. Sử dụng CNN Model với CategoricalCrossentropy 28**](#_Toc77250879)

[**\*\* Kết quả so sánh 2 model: 33**](#_Toc77250880)

[**Chương 5: Kết quả 34**](#_Toc77250881)

[**5.1. Kết quả thu được 34**](#_Toc77250882)

[**5.2. Ưu điểm, nhược điểm 35**](#_Toc77250883)

[**5.2.1. Ưu điểm 35**](#_Toc77250884)

[**5.2.2. Nhược điểm 35**](#_Toc77250885)

[**5.3. Hướng phát triển 35**](#_Toc77250886)

**MỞ ĐẦU**

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:

* Giáo dục: Phản ứng của người học trong thời gian thực và sự tham gia vào nội dung là giáo dục là một thước đo lường cho hiệu quả của bài giảng.
* Tiếp thị: Đây là một cách tuyệt vời để các công ty kinh doanh phân tích cách khách hàng phản hồi với quảng cáo, sản phẩm, bao bì và thiết kế cửa hàng của họ.
* Chơi game: Với sự ra đời của game thực tế ảo gần với trải nghiệm thực tế. Nhận dạng cảm xúc khuôn mặt đóng một vai trò quan trọng để cải thiện trải nghiệm chơi trò chơi.
* Bảo mật: Nó có thể giúp xác định hành vi đáng ngờ trong đám đông và có thể được sử dụng để ngăn chặn tội phạm và những kẻ khủng bố tiềm năng.
* Chăm sóc sức khỏe: Nó có thể hữu ích trong việc tự động hóa dịch vụ y tế. Cả sức khỏe thể chất và tinh thần có thể được phân tích thông qua ứng dụng này.
* Dịch vụ khách hàng: Quản lý dịch vụ khách hàng có thể hiệu quả hơn bằng cách sử dụng hệ thống nhận dạng cảm xúc khuôn mặt. Phân tích phản hồi của khách hàng và phản ứng của máy tính sẽ đảm bảo tương tác máy tính với con người trong cuộc sống thực.

Ngày nay, khi xã hội của chúng ta đang phát triển nhanh chóng, chúng ta có thể dễ dàng thấy rất nhiều thay đổi trong nhiều thứ xung quanh chúng ta. Từ cách chúng ta giao tiếp, làm việc hay cách chúng ta giải trí, tất cả đều khác. Một trong những lý do cho những thay đổi này là sự phát triển của công nghệ máy tính. Hiện nay, việc nhận diện khuôn mặt đang trở nên phổ biến. Các nhà phát hành đã và đang phát triển các phần mềm nhận diện, thay đổi cảm xúc khuôn mặt. Đề tài này là một phần của những phần mềm đó – nhận diện cảm xúc của mặt người.

**Chương 1: Giới thiệu đề tài**

**1.1. Tên đề tài**

Xây dựng mô hình học sâu để phát hiện Cảm xúc của khuôn mặt người.

**1.2. Mục tiêu và nội dung.**

**1.2.1. Mục tiêu**

* Xây dựng tập dữ liệu ảnh có kích thước đầu vào mặc định (48,48) là các cảm xúc khuôn mặt.
* Xây dựng mô hình học máy phát hiện cảm xúc khuôn mặt với độ chính xác cao.

**1.2.2. Nội dung**

* Ứng dụng xử lý ảnh để tìm ra bài toán phù hợp để nhận diện được các cảm xúc của khuôn mặt.
* Xây dựng mô hình Convolutional Neural Network (CNN) để nhận diện được các cảm xúc.

**1.3. Thành viên**

Bảng 1: Bảng thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Chức vụ** | **MSV** |
| **01** | Phạm Tiến Lâm | Giảng viên hướng dẫn |  |
| **02** | Phạm Ngọc Việt | Sinh viên | 19010042 |
| **03** | Trịnh Tuấn Anh | Sinh viên | 19010002 |
| **04** | Lê Hữu Trung | Sinh viên | 19010000 |

**1.4. Phương thức thực hiện và kết quả**

**1.4.1. Phương thức thực hiện**

* Nắm chắc kiến thức cơ bản về ngôn ngữ lập trình Python trên Windows.
* Tìm hiểu thư viện Scikit - learn trong Python.
* Chuẩn bị dữ liệu thô và Chuẩn hóa dữ liệu...
* Tìm hiểu CNN (Convolutional Neural Network) trong Keras để xây dựng mô hình học máy.
* Áp dụng và xây dựng mô hình CNN.

**1.4.2. Kết quả**

Tạo mô hình hình học sâu có thể phát hiện cảm xúc khuôn mặt của một người trong ảnh.

**Chương 2: Tổng quan cơ sở lý thuyết và xử lý dữ liệu ảnh**

**2.1. Ngôn ngữ lập trình Python**

Python là ngôn ngữ lập trình hướng đối tượng thông dụng dùng để viết các tiện ích hệ thống. Nó cũng được sử dụng như ngôn ngữ kết dính đóng vai trò tích hợp C và C++.

Được tạo ra bởi Guido van Rossum tại Amsterdam năm 1990, Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Python là ngôn ngữ có hình thức khá đơn giản và rõ ràng, do đó tạo nên sự dễ dàng tiếp cânh cho những lập trình viên mới bắt đầu.

Ban đầu, Python được phát triển để chạy trên nền Unix, nhưng rồi theo thời gian, nó đã được mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.

**Một số tính chất của Python**

* **Interpreted**: nhờ chức năng thông dịch mà trình thông dịch (Interpreter) của Python có thể xử lý lệnh tại thời điểm chạy chương trình (runtime). Nhờ đó mà không cần biên dịch chương trình trước khi thực hiện nó (tương tự như Perl và PHP).
* **Interactive**: tính năng tương tác của Python giúp tương tác trực tiếp với trình thông dịch của nó ngay tại dấu nhắc lệnh. Cụ thể: Có thể thực hiện lệnh một cách trực tiếp tại dấu nhắc của Python.
* **Object-Oriented**: Python hỗ trợ mạnh cho phong cách lập trình hướng đối tượng và kỹ thuật lập trình gói mã trong đối tượng.
* **Beginner's Language**: Python được xem là ngôn ngữ lập trình dành cho những ai mới làm quen với việc lập trình trên máy tính, nhưng nó hỗ trợ mạnh cho việc phát triển nhiều loại ứng dụng khác nhau, từ các chương trình xử lý văn bản đơn giản đến các ứng dụng web, đến các chương trình game, …

**Tính năng**

*Ngôn ngữ lập trình đơn giản, dễ học*: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

*Miễn phí, mã nguồn mở*: Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

*Ngôn ngữ thông dịch cấp cao*: Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa, ... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

*Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến*: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

*Hướng đối tượng*: Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

**2.2. Tổng quan về thư viện Scikit - learn trong Python**

**2.2.1. Khái niệm**

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

Thư viện được cấp phép bản quyền chuẩn FreeBSD và chạy được trên nhiều nền tảng Linux. Scikit-learn được sử dụng như một tài liệu để học tập.

Để cài đặt scikit-learn trước tiên phải cài thư viện SciPy (Scientific Python). Những thành phần gồm:

* [Numpy](https://codelearn.io/sharing/tim-hieu-thu-vien-numpy-trong-python): Gói thư viện xử lý dãy số và ma trận nhiều chiều.
* SciPy: Gói các hàm tính toán logic khoa học.
* [Matplotlib](https://codelearn.io/sharing/ve-bieu-do-voi-thu-vien-matplotlib-p1): Biểu diễn dữ liệu dưới dạng đồ thị 2 chiều, 3 chiều.
* IPython Notebook: dùng để tương tác trực quan với Python.
* SymPy: Gói thư viện các kí tự toán học.
* [Pandas](https://codelearn.io/sharing/xu-ly-du-lieu-voi-pandas-trong-python): Xử lý, phân tích dữ liệu dưới dạng bảng.

Những thư viện mở rộng của SciPy thường được đặt tên dạng SciKits. Như thư viện này là gói các lớp, hàm sử dụng trong thuật toán học máy thì được đặt tên là scikit-learn.

Scikit-learn hỗ trợ mạnh mẽ trong việc xây dựng các sản phẩm. Nghĩa là thư viện này tập trung sâu trong việc xây dựng các yếu tố: dễ sử dụng, dễ code, dễ tham khảo, dễ làm việc, hiệu quả cao.

Mặc dù được viết cho Python nhưng thực ra các thư viện nền tảng của scikit-learn lại được viết dưới các thư viện của C để tăng hiệu suất làm việc. Ví dụ như: Numpy (Tính toán ma trận), LAPACK, LibSVM và Cython.

**2.2.2. Thuật toán**

Thư viện tập trung vào việc mô hình hóa dữ liệu. Nó không tập trung vào việc truyền tải dữ liệu, biến đổi hay tổng hợp dữ liệu. Những công việc này dành cho thư viện Numpy và Pandas.

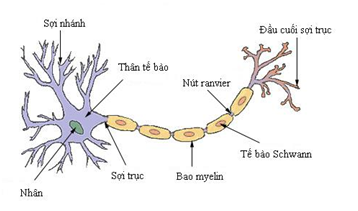
Sau đâu là một số nhóm thuật toán được xây dựng bởi thư viện scikit-learn:

* **Clustering**: Nhóm thuật toán Phân cụm dữ liệu không gán nhãn. Ví dụ thuật toán KMeans
* **Cross Validation**: Kiểm thử chéo, đánh giá độ hiệu quả của thuật toán học giám sát sử dụng dữ liệu kiểm thử (validation data) trong quá trình huấn luyện mô hình.
* **Datasets**: Gồm nhóm các Bộ dữ liệu được tích hợp sẵn trong thư viện. Hầu như các bộ dữ liệu đều đã được chuẩn hóa và mang lại hiêu suất cao trong quá trình huấn luyện như iris, digit, ...
* **Dimensionality Reduction**: Mục đích của thuật toán này là để Giảm số lượng thuộc tính quan trọng của dữ liệu bằng các phương pháp như tổng hợp, biểu diễn dữ liệu và lựa chọn đặc trưng. Ví dụ thuật toán PCA (Principal component analysis).
* **Ensemble methods**: Các Phương pháp tập hợp sử dụng nhiều thuật toán học tập để có được hiệu suất dự đoán tốt hơn so với bất kỳ thuật toán học cấu thành nào.
* **Feature extraction**: Trích xuất đặc trưng. Mục đích là để định nghĩa các thuộc tình với dữ liệu hình ảnh và dữ liệu ngôn ngữ.
* **Feature selection**: Trích chọn đặc trưng. Lựa chọn các đặc trưng có ý nghĩa trong việc huấn luyện mô hình học giám sát.
* **Parameter Tuning**: Tinh chỉnh tham số. Các thuật toán phục vụ việc lựa chọn tham số phù hợp để tối ưu hóa mô hình.
* **Manifold Learning**: Các thuật toán học tổng hợp và Phân tích dữ liệu đa chiều phức tạp.
* **Supervised Models**: Học giám sát. Mảng lớn các thuật toán học máy hiện nay. Ví dụ như linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines và decision trees.

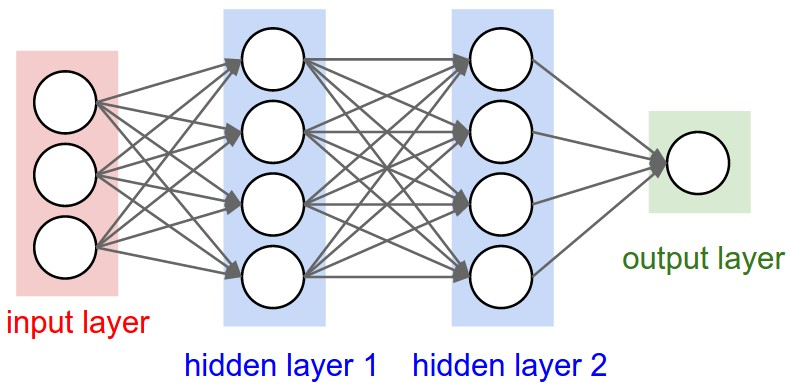
**2.3. Tổng quan về Deep Learning và bài toán nhận diện trong thị giác máy tính**

**2.3.1. Mạng nơron**

Mạng Neural là một [mô hình toán học](https://vi.wikipedia.org/wiki/M%C3%B4_h%C3%ACnh_to%C3%A1n_h%E1%BB%8Dc) hay [mô hình tính toán](https://vi.wikipedia.org/wiki/M%C3%B4_h%C3%ACnh_t%C3%ADnh_to%C3%A1n) được xây dựng dựa trên các [mạng neural sinh học](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A1ng_n%C6%A1-ron_sinh_h%E1%BB%8Dc&action=edit&redlink=1). Nó gồm có một nhóm các [neural nhân tạo](https://vi.wikipedia.org/w/index.php?title=N%C6%A1-ron_nh%C3%A2n_t%E1%BA%A1o&action=edit&redlink=1) (nút) nối với nhau, và xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút.



Mạng nơ-ron nhân tạo.



Thuật toán mạng nơ-ron.

Layer đầu tiên là input layer, các layer ở giữa được gọi là hidden layer, layer cuối cùng được gọi là output layer. Các hình tròn được gọi là node.

Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer – 1 (Không tính input layer).

Ví dụ như ở hình trên có 1 input layer, 2 hidden layer và 1 output layer. Số lượng layer của mô hình là 3 layer.

Mỗi node trong hidden layer và output layer :

* Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng.
* Mỗi node có 1 hệ số bias b riêng.
* Diễn ra 2 bước: tính tổng linear và áp dụng activation function.

**2.3.2. Định nghĩa Deep Learning**

Deep learning (còn gọi là học sâu). Khái niệm này được lấy cảm hứng từ chức năng giống với các tế bào não của con người gọi là tế bào thần kinh. Deep Learning là tập con của Machine Learning và nó có tác dụng hỗ trợ cho máy tính tự huấn luyện chính nó để có thể thực hiện mọi tác vụ tương tự như con người. Điều này chính là giúp máy tính bắt chước con người cách học hỏi và suy nghĩ.

Các hệ thống của Deep Learning có khả năng cải thiện được những hiệu suất  của chúng với quyền truy cập vào dữ liệu sẽ được nhiều hơn. Thông thường, phiên bản máy sẽ có nhiều kinh nghiệm hơn; những máy móc đã có đủ kinh nghiệm thì sẽ được mang đi phục vụ cho những công việc như: lái xe, phát hiện người, ...

Deep Learning có hỗ trợ cho việc dịch ngôn ngữ, phân loại các hình ảnh, nhận dạng giọng nói. Chính vì thế, nó có thể được ứng dụng để giải quyết mọi nhu cầu cần nhận dạng mẫu mà không cần đến sự can thiệp của con người.

Deep Learning hoạt động dựa vào mạng lưới thần kinh nhân tạo và nó bao gồm nhiều lớp chứa các  dữ liệu mô phỏng cách thức vận hành của não người.

Bộ phận mạng thần kinh nhân tạo này giống với não người bao gồm: Các node (nó là đơn vị thần kinh trong mạng thần kinh nhân tạo) là những neuron thần kinh. Hầu hết các node bản thân thường chỉ có khả năng trả lời cho những câu hỏi  đơn giản và cơ bản nhất; với những tác vụ khó thì chúng sẽ tiến hành liên kết với nhau để trả lời. Bạn có thể dạy hoặc có thể huấn luyện chúng bằng những  thuật toán cụ thể.

Những node giải đáp các câu hỏi phức tạp thì sẽ được gọi là mạng lưới thần kinh sâu (DNN), nó được định nghĩa như sau: Mạng lưới thần kinh sâu có khả năng thực hiện các hoạt động phức tạp như biểu diễn, trừu tượng có mang các ý nghĩa về âm thanh, văn bản và hình ảnh. Chúng được đánh giá là lĩnh vực phát triển nhất trong Machine Learning.

**2.3.3. Cách thức hoạt động của Deep Learning**

Các dòng thông tin sẽ được trải qua nhiều lớp cho đến lớp sau cùng. Lấy quy trình học của con người làm ví dụ cụ thể. Qua các lớp đầu tiên sẽ tập trung vào việc học các khái niệm cụ thể hơn trong khi các lớp sâu hơn sẽ sử dụng thông tin đã học để nghiên cứu và phân tích sâu hơn trong các khái niệm trừu tượng . Quy trình xây dựng biểu diễn dữ liệu này được gọi là trích xuất tính năng.

Kiến trúc phức tạp của việc học sâu được cung cấp từ mạng lưới thần kinh sâu với khả năng thực hiện trích xuất tính năng tự động. Ngược lại, trong học máy thông thường còn gọi là học nông, nhiệm vụ này được thực hiện khi truy xuất các thuật toán cụ thể.

**2.3.4. Các ưu điểm của Deep Learning**

* Deep Learning có độ chính xác cao, có thể học sâu rộng và đạt được độ chính xác ở mức công nhận vô cùng cao. Điều này sẽ đảm bảo cho thiết bị điện tử tiêu dùng có thể đáp ứng được mọi nhu cầu cũng như mong  đợi từ phía người dùng. Deep Learning rất trọng về độ an toàn đối với những  mô hình xe không người lái.
* Các dữ liệu của Deep Learning sẽ được dán nhãn bởi: Việc phát triển của xe không người lái đòi hỏi hàng triệu hình ảnh cũng như hàng ngàn giờ để xem video.
* GPU có hiệu suất cao và có kiến thức song song nên rất hiệu quả cho Deep Learning. Nếu như kết hợp với điện toán đám mây hoặc các cụm thì nó cho phép nhóm phát triển có thể giảm xuống tối đa thời gian đào tạo đối với mạng lưới học tập sau vài tuần hoặc vài giờ.

**2.3.5. Ứng dụng của Deep Learning**

* Ứng dụng trong các hệ thống gợi ý

Các nền tảng lớn hiện nay như Facebook, Lazada, Tiki…đều có hệ thống gợi ý rất mạnh giúp tăng đáng kể độ tương tác của người dùng chúng dựa trên các dữ liệu của người dùng phát sinh ra khi sử dụng và tương tác trên các thiết bị có kết nối internet để gợi ý thêm những sản phẩm họ sẽ thích. Ví dụ như trên các nền tảng mua sắm, gợi ý các bài quảng cáo hoặc được tài trợ như trên Facebook hay các khóa học mà người học quan tâm như trên các nền tảng học online.

* Ứng dụng trong ngành công nghệ cao

Hiện nay, Robot còn  có thể hợp tác với các hoạt động của con người và chúng có thể thực hiện được các nhiệm vụ riêng biệt phù hợp với thế mạnh của chúng. Robot đang góp phần thay thế con người trong việc thực hiện các công việc khó khăn hơn. Đây chính  là phát minh tuyệt vời nhờ ứng dụng Deep Learning.

* Ứng dụng trong nông nghiệp

Hiện nay, nhờ có Deep Learning mà người nông dân có thể triển khai những thiết bị có khả năng phân biệt được cỏ dại với cây trồng. Từ đó, các loại máy móc phun thuốc diệt cỏ có thể chọn lọc phun lên  cỏ dại để đảm bảo cho cây trồng không bị ảnh hưởng.

* Ứng dụng trong y tế và chăm sóc sức khỏe

Deep Learning phát huy được hiệu quả siêu nổi bật của mình bằng cách hiển thị dữ liệu chất lượng cao và  đo lường khả năng của các mạng thần kinh rồi thực hiện phân loại hình ảnh. Deep Learning có khả năng phân loại ung thư da chuẩn xác và hiệu quả như một bác sĩ da liễu chuyên nghiệp.

Đồng thời, các nhà nghiên cứu về căn bệnh ung thư cũng đang ứng dụng Deep Learning để có thể tự động phát hiện ra các tế bào phát bệnh, nó được thực hiện như sau: chế tạo kính hiển vi tiên tiến để có thể tạo ra bộ dữ liệu có chiều cao phù hợp, được dùng để đào tạo một ứng dụng học sâu để  có thể xác định tế bào ung thư.

Ngoài ra, Deep Learning còn có bước tiến quan trọng trong việc cải thiện chất lượng sức khỏe của bệnh nhân bằng sự dự đoán với các sự kiện y tế từ dữ liệu hồ sơ sức khỏe điện tử.

* Ứng dụng trong công nghiệp

Deep Learning có khả năng cải thiện sự an toàn cho các công nhân xung máy móc hạng nặng bằng cách: tự động phát hiện nếu như người hoặc vật nằm trong khoảng cách không an toàn đối với máy.

**2.3.6. Hạn chế của Deep Learning**

* Deep learning luôn đòi hỏi một lượng dữ liệu đầu vào khổng lồ để máy tính học hỏi. Quy trình này mất nhiều thời gian và sức mạnh xử lý chỉ có các server (máy chủ) cỡ lớn mới làm được. Nếu không có đủ dữ liệu đầu vào hoặc có đủ dữ liệu đầu vào, nhưng lại không có đủ sức mạnh để xử lý thì mọi thứ không thể diễn ra đúng như ý định và kết quả máy tính đưa ra cuối cùng cũng không chính xác.
* Deep learning vẫn chưa thể nhận biết được những thứ phức tạp, hoặc tương tự nhau. Lý do là vì hiện chưa có kĩ thuật nào đủ tốt nhằm giúp trí tuệ nhân tạo có thể rút ra những kết luận đó một cách logic bởi chúng chưa có được khả năng nhận biết như con người.
* Trí tuệ nhân tạo mang lại rất nhiều giá trị cho cuộc sống loài người nhưng cũng tiềm ẩn những nguy cơ. Rất nhiều chuyên gia lo lắng rằng, khi trí tuệ nhân tạo đạt tới 1 ngưỡng tiến hóa nào đó thì đó cũng là thời điểm loài người bị tận diệt. Rất nhiều các bộ phim đã khai thác đề tài này với nhiều góc nhìn khác nhau, nhưng qua đó đều muốn cảnh báo loài người về mối nguy hiểm đặc biệt này. Dự báo cho rằng, từ 5 đến 10 năm nữa ngành khoa học này sẽ phát triển lên tới đỉnh cao. Hãy cùng chờ đợi những thành tựu mới nhất của loài người về lĩnh vực này.

**2.4. Thư viện Keras**

**2.4.1. Giới thiệu thư viện Keras**

Keras là một thư viện nơron mã nguồn mở được viết bằng ngôn ngữ lập trình Python. Nó có khả năng chạy trên đầu trang của Tensorflow, Microsoft Cognitive Toolkit, R, Theano hoặc StripeML. Được thiết kế để cho phép thử nghiệm nhanh với các mạng thần kinh sâu, nó tập trung vào việc thân thiện với người dùng, mô-đun và mở rộng.

Keras được coi là một thư viện ‘high-level’ với phần ‘low-level’ (còn được gọi là backend) có thể là TensorFlow, CNTK, hoặc Theano (sắp tới Theano sẽ không được duy trì nâng cấp nữa). Keras có cú pháp đơn giản hơn TensorFlow rất nhiều. Với mục đích giới thiệu về các mô hình nhiều hơn là cách sử dụng các thư viện deep learning, tôi sẽ chọn Keras với TensorFlow là ‘backend’.

**2.4.2. Một số tính năng của Keras**

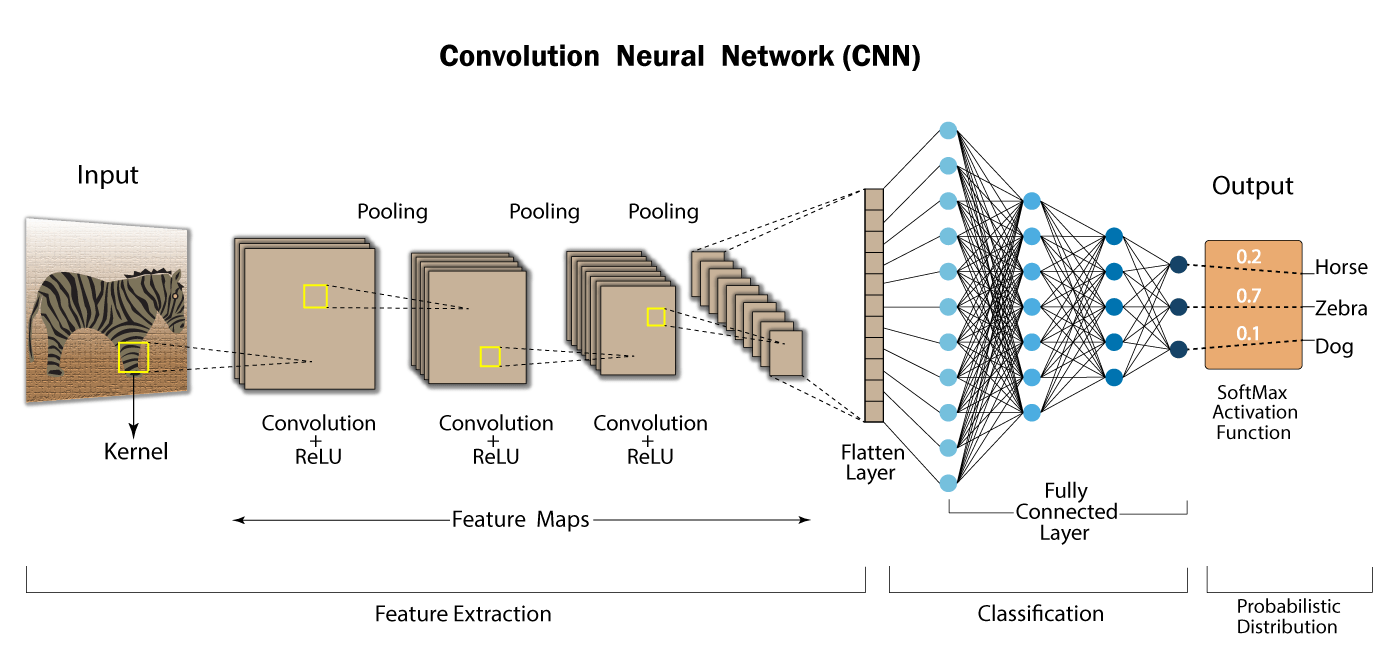
* Keras ưu tiên trải nghiệm của người lập trình.
* Keras đã được sử dụng rộng rãi trong doanh nghiệp và cộng đồng nghiên cứu.
* Keras giúp dễ dàng biến các thiết kế thành sản phẩm.
* Keras hỗ trợ huấn luyện trên nhiều GPU phân tán.
* Keras hỗ trợ đa backend engines và không giới hạn bạn vào một hệ sinh thái.

**2.5. Giới thiệu Convolutional Neural Network (CNN)**

**2.5.1. Tổng quát**

Convolutional Neural Network (CNN) còn gọi là mạng nơ ron tích chập. Đây là một trong những mô hình của Deep Learning (Deep Learning là tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hoá ở mức độ cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp hoặc bằng cách khác). Tác dụng của thuật toán này chính là giúp chúng ta tạo ra được những hệ thống thông minh có sự phản ứng với độ chính xác cao. Thêm vào đó là khả năng áp dụng được vào đời sống thực tiễn. Tức là chúng ta sử dụng CNN để xác định xem hình ảnh đó là gì , tức là nó có thể là cái này hoặc cái kia.

**2.5.2. Các lớp của CNN**



Sơ đồ các lớp của CNN

* Convolutional layer

Có thể nói đây là một lớp cực kỳ quan trọng trong CNN, bởi ở lớp này sẽ thực hiện mọi phép tính toán. Một khái niệm cần nhắc đến ở Convolutional layer là filter map, stride, padding, feature map

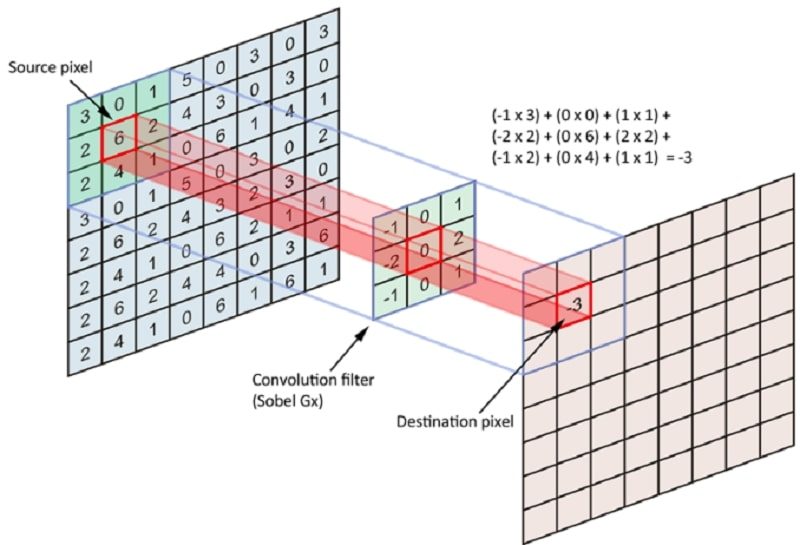
* CNN sử dụng những filter để áp vào những vùng của hình ảnh. Các filter map này chính là một ma trận 3 chiều, trong đó bao gồm những con số và các con số đó chính là parameter.
* Stride có thể hiểu là khi bạn dịch chuyển filter map theo pixel dựa vào một giá trị từ trái sang phải. Stride đó chính là chỉ sự dịch chuyển này.
* Padding là những giá trị 0 được thêm vào lớp input.
* Feature map: thể hiện kết quả mỗi lần filter map quét qua input. Mỗi lần quét như thế sẽ xảy ra quá trình tính toán.
* Pooling layer
* Nếu như đầu vào quá lớn, các lớp pooling layer sẽ được xếp vào giữa các lớp Convolutional layer để làm giảm các parameter.
* Pooling layer có 2 loại phổ biến là max pooling và average pooling.
* Ở đây, khi sử dụng lớp max pooling thì số lượng parameter giảm đi. Khi đó CNN gồm nhiều filter map, mỗi filter map đó sẽ cho max pooling khác nhau.
* Relu layer
* Relu layer chính là một hàm kích hoạt trong neural network. Hàm kích hoạt còn được gọi là activation function. Tác dụng chính của hàm kích hoạt này chính là việc mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function có các hàm cơ bản như: Sigmoid, Tanh, Relu, Leaky relu, Maxout.
* Đặc biệt là trong việc huấn luyện các mạng neuron thì relu có những ưu điểm khá nổi bật. Có thể kể đến việc tính toán nhanh hơn…
* Fully connected layer

Dùng để đưa ra kết quả.

**2.5.3. Cấu trúc của CNN**

Mạng CNN gồm nhiều lớp Convolutional chồng lên nhau, sử dụng các hàm tính toán để kích hoạt trọng số. Mỗi lớp sau khi được kích hoạt sẽ cho ra kết quả trừu tượng cho các lớp tiếp theo. Mỗi layer kế tiếp chính là thể hiện kết quả của layer trước đó,

Thông qua quá trình training, các lớp layer CNN tự động học các giá trị được thể hiện qua các lớp filter.



Lớp layer CNN tự động.

Cấu trúc cơ bản của CNN gồm 3 phần chính: Local receptive field, share weights and bias, pooling.

* Local receptive field: là các trường cục bộ. Tác dụng của lớp mày chính là nó giúp chúng ta lọc các dữ liệu, thông tin của ảnh và chọn được những vùng ảnh có giá trị sử dụng nhất.
* Share weights and bias: là trọng số chia sẻ. Làm giảm tối đa số lượng các tham số.
* Pooling layer: lớp tổng hợp. Đây gần như là lớp cuối cùng, có tác dụng làm đơn giản hoá thông tin đầu ra. Tức là, sau khi hoàn tất các quá trình tính toán và quét các lớp thì sẽ đi đến lớp pooling layer để giảm bớt thông tin không cần thiết, sau đó cho ra kết quả mà chúng ta mong muốn.

**2.6. Giới thiệu bài toán Object Detection**

Một trong những lĩnh vực quan trọng của Trí tuệ nhân tạo (Artificial Intelligence) là thị giác máy (Computer Vision). Computer Vision là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh, phát hiện các đối tượng, tạo ảnh, siêu phân giải hình ảnh và nhiều hơn vậy. Object Detection có lẽ là khía cạnh sâu sắc nhất của thị giác máy do số lần sử dụng trong thực tế.

Object Detection đề cập đến khả năng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng. Object Detection đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái. Có nhiều cách để nhận diện đối tượng có thể được sử dụng cũng như trong nhiều lĩnh vực thực hành. Giống như mọi công nghệ khác, một loạt các ứng dụng sáng tạo và tuyệt vời của Object Detection sẽ đến từ các lập trình viên và các nhà phát triển phần mềm.

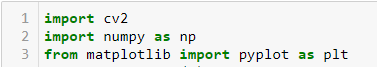
Bắt đầu sử dụng các phương pháp nhận diện đối tượng hiện đại trong các ứng dụng và hệ thống, cũng như xây dựng các ứng dụng mới dựa trên các phương pháp này. Việc triển nhận diện đối tượng sớm liên quan đến việc sử dụng các thuật toán cổ điển, giống như các thuật toán được hỗ trợ trong OpenCV, thư viện computer vision phổ biến. Tuy nhiên, các thuật toán cổ điển này không thể đạt được hiệu suất đủ để làm việc trong các điều kiện khác nhau.

Việc áp dụng đột phát và nhanh chóng của deep learning vào năm 2012 đã đưa vào sự tồn tại các thuật toán và phương pháp phát hiện đối tượng như R-CNN, Fast-RCNN, Faster-RCNN, RetinaNet, SSD và YOLO. Sử dụng các phương pháp và thuật toán này, dựa trên deep learning và cũng dựa trên việc học máy đòi hỏi rất nhiều kiến thức về toán học và việc học sâu. Có hàng triệu chuyên gia lập trình và các nhà phát triển phần mềm muốn tích hợp và tạo ra các sản phẩm mới sử dụng object detection. Nhưng công nghệ này xa tầm tay của họ và phức tạp để hiểu và sử dụng thực tế của nó.

* Họ các mô hình object detection dựa trên Region-Based Convolutional Neural Network (R-CNNs) gồm các lớp mô hình: R-CNN, Fast R-CNN và Faster R-CNN là những mô hình sơ khai, có tốc độ xử lý chậm. Thuật toán dựa trên 2 phần xử lý riêng biệt là phát hiện các region proposal và phân loại hình ảnh.
* Lớp các mô hình YOLO có tốc độ thời gian xử lý thực. Là công nghệ của các mô hình object detection dựa trên Region-Based Convolutional Neural Network (R-CNNs) gồm các lớp mô hình: R-CNN, Fast R-CNN và Faster R-CNN là những mô hình sơ khai, có tốc độ xử lý chậm. Thuật toán dựa trên 2 phần xử lý riêng biệt là phát hiện các region proposal và phân loại hình ảnh.
* Lớp các mô hình YOLO có tốc độ thời gian xử lý thực. Là công nghệ state-of-art nhất hiện nay có tốc độ xử lý realtime, phát hiện được lên tới 9000 loại đối tượng.state-of-art nhất hiện nay có tốc độ xử lý realtime, phát hiện được lên tới 9000 loại đối tượng.

**2.7 Xử lý dữ liệu ảnh**

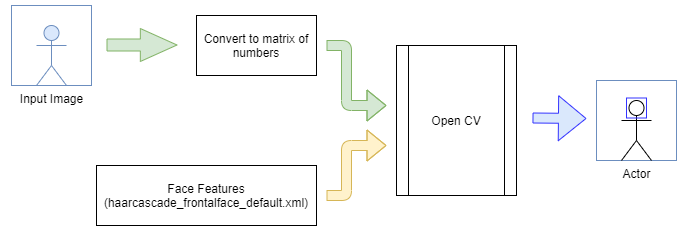
**2.7.1.  Thư viện cần sử dụng**

****

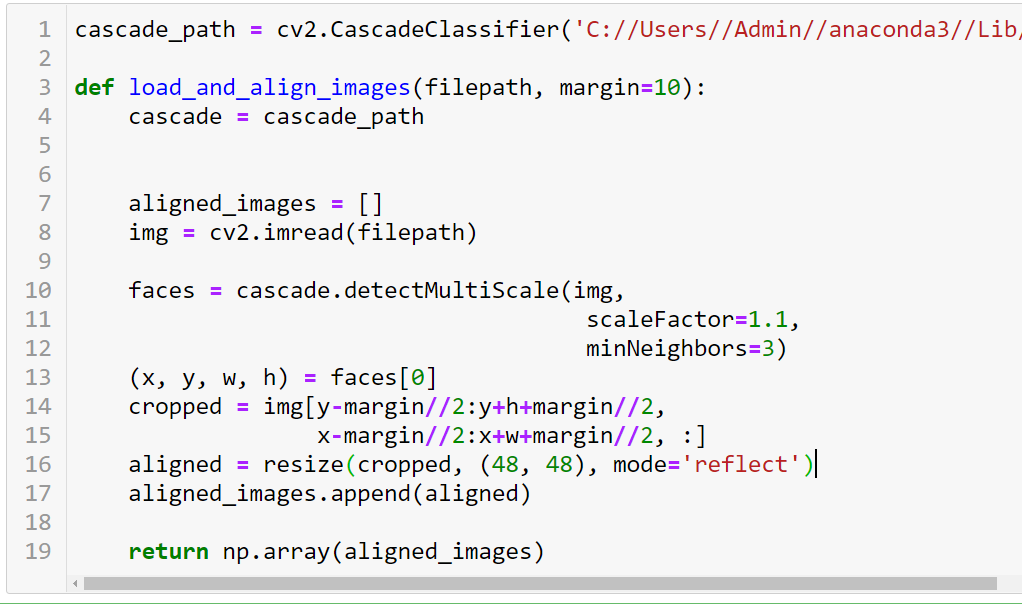
* Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.
* OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực.
* matplotlib.pyplot là một thư viện vẽ đồ họa được sử dụng cho đồ họa 2D bằng ngôn ngữ lập trình python . Nó có thể được sử dụng trong các tập lệnh python, shell, máy chủ ứng dụng web và các bộ công cụ giao diện người dùng đồ họa khác.

**2.7.2. Phương pháp sử dụng**

Bước 1: sử dụng “ **haarcascade\_frontalface\_default.xml** ” để phát hiện khuôn mặt trong hình ảnh được truyền vào.



Bước 2: Kết hợp sử dụng hàm **load\_and\_align\_images** để cắt phần ảnh được phát hiện.



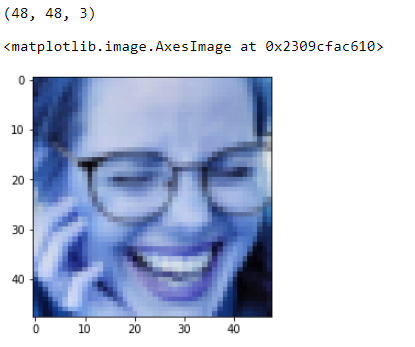
Hàm sẽ trả về một numpy array của ảnh khuôn mặt được phát hiện với size (48,48).

(x,y là tọa độ trái trên còn w,h là độ dài và chiều cao của ảnh đã được nhận diện so với ảnh gốc)

**Ảnh ban đầu (gốc)**



**Ảnh đã được nhận diện và resize**

****

**Chương 3: Bài toán nhận diện cảm xúc**

**3.1. Giới thiệu**

Trong những năm gần đây,xử lí hình ảnh đang được nghiên cứu và phát triển với tốc độ nhanh chóng bởi các trung tâm nghiên cứu, trường đại học cũng như học viện. Trong đó, nhận dạng và phân loại hình ảnh là một trong những lĩnh vực được theo đuổi một cách tích cực. Ý tưởng cốt lỗi từ việc nhận dạng và phân loại hình ảnh là phân tích ảnh từ dữ liệu được thu bởi các cảm biến hình ảnh như camera, webcam. Nhờ hệ thống xử lý hình ảnh mà con người đã giảm bớt khối công việc cũng như tăng sự chính xác trong việc đưa ra các quyết định liên quan đến xử lý ảnh trên nhiều lĩnh vực: quân sự và quốc phòng, các hệ thống kỹ nghệ hóa sinh, giải phẫu, các hệ thống thông minh, robotics, các hệ thống an ninh và bảo mật.

Điều quan trọng nhất trong xã hội hiện nay là các thiết bị điện tử hầu như tự động hóa, thông minh, hiểu ý con người mà không cần phải thiết bị điều khiển trung gian nào. Các thuật toán nhận dạng và xử lý ngày càng được sử dụng rộng rãi. Vì vậy từ những vấn đề trên, bọn em đã thực hiện đề tài ”***Nhận diện và phân loại cảm xúc khuôn mặt người***” bằng xử lý ảnh

**3.2. Xác định bài toán**

Bài toán nhận diện cảm xúc khuôn mặt người thuộc dạng bài toán Nhận diện và phân loại (Object Detection and Classification). Object Detection đề cập đến khả năng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng. Object Detection đã được sử dụng rộng rãi để phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật và xe không người lái. Có nhiều cách để nhận diện đối tượng có thể được sử dụng cũng như trong nhiều lĩnh vực thực hành.

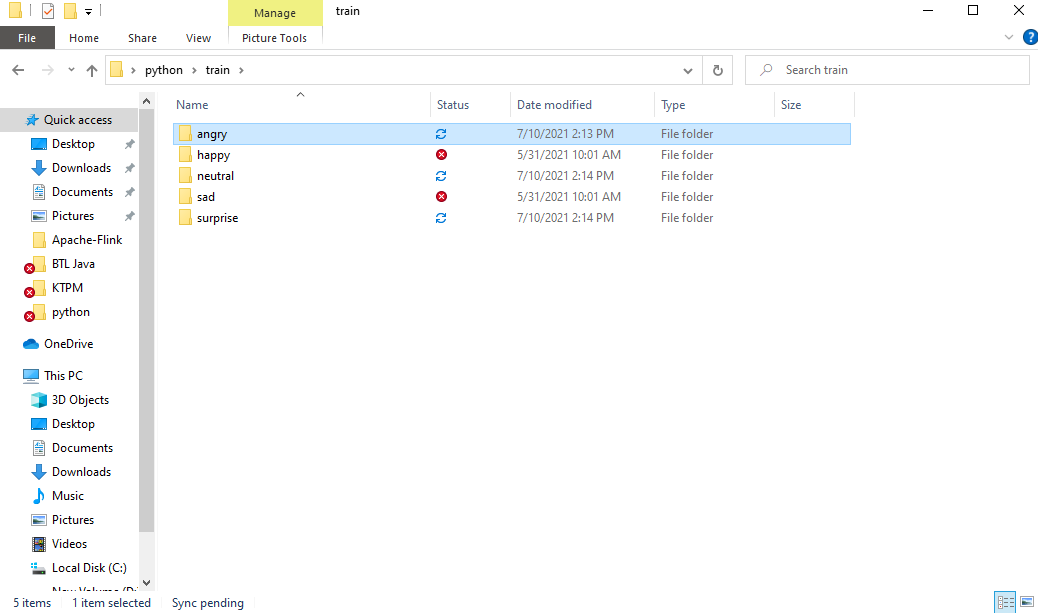
Bài toán phân lớp (Classification) là quá trình phân lớp một đối tượng dữ liệu vào một hay nhiều lớp đã cho trước nhờ một mô hình phân lớp (model). Mô hình này được xây dựng dựa trên một tập dữ liệu được xây dựng trước đó có gán nhãn (hay còn gọi là tập huấn luyện). Quá trình phân lớp là quá trình gán nhãn cho đối tượng dữ liệu.

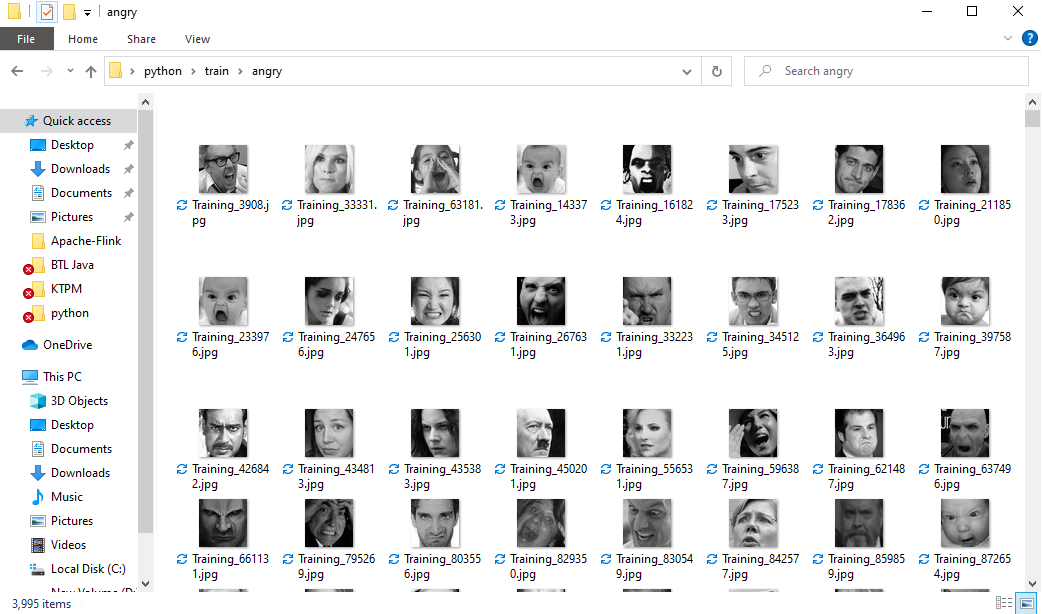
Chính vì thế, ở Bài toán Nhận diện và phân loại cảm xúc khuôn mặt người bọn em đã xây dựng ý tưởng bắng cách cắt khuôn mặt trong ảnh theo tỉ lệ 48x48 rồi chấm điểm ảnh dựa trên model CNN để xác định ảnh sẽ thuộc class nào trong những class đã được train.

**Chương 4: Xây dựng các mô hình.**

**4.1. Chuẩn bị dữ liệu.**

Dữ liệu ban đầu sẽ là các ảnh chứa trong các founder có kích thước (48,48) đã được nhận diện khuôn mặt.



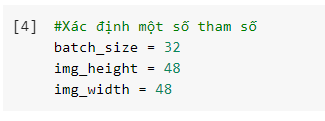
****

**4.2. Mô hình đã xây dựng**

**4.2.1. Sử dụng** [**CNN Sequential**](https://www.tensorflow.org/api_docs/python/tf/keras/Sequential) **với SparseCategoricalCrossentropy**

**Bước 1: Tạo tập dữ liệu.**

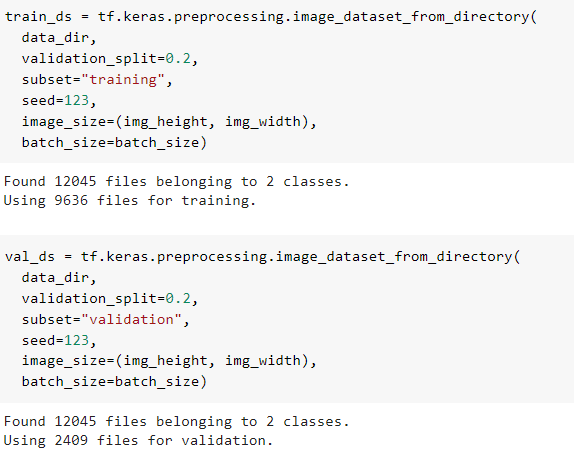
Xác định một số tham số cho bộ tải:

****

**Bước 2: chia bộ dữ liệu sử dụng 80% hình ảnh để đào tạo và 20% để xác thực.**

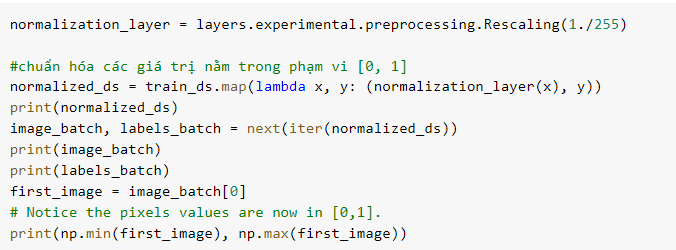
Lấy trực tiếp từ founder bằng thư viện:

**tf.keras.preprocessing.image\_dataset\_from\_directory**

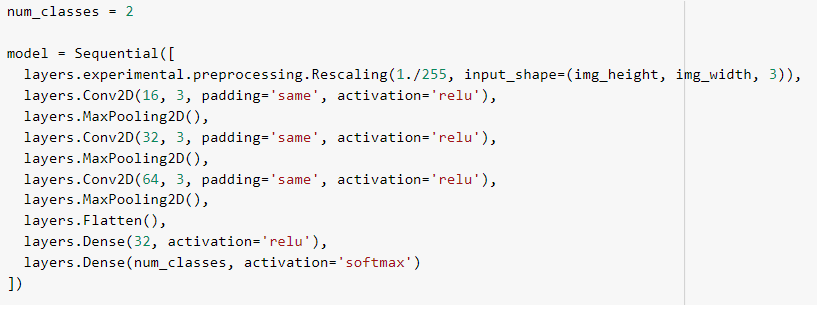
****

**Bước 3: Chuẩn hóa dữ liệu:**

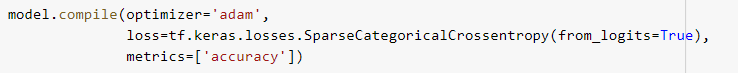
Các giá trị kênh RGB nằm trong phạm vi [0, 255] . Điều này không lý tưởng cho một mạng nơ-ron. Tại đây, sẽ chuẩn hóa các giá trị nằm trong phạm vi [0, 1] bằng cách sử dụng lớp Thay đổi tỷ lệ.

****

**Bước 4: Xây dựng mô hình với Sequential**

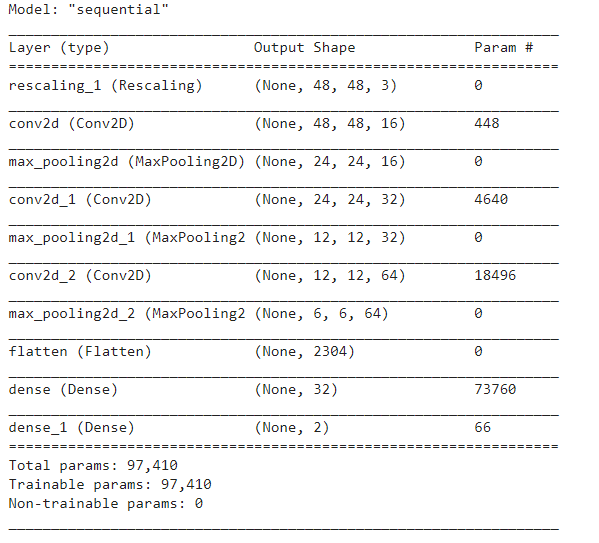
****

**Bước 5: Biên dịch**

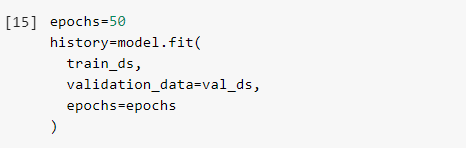


Ở đây, sử dụng hàm mất mát (loss) là 'sparse\_categorical\_crossentropy'. Categorical crossentropy dùng để tính toán mất mát giữa các lable và dự đoán. Sử dụng 'sparse \_categorical\_crossentropy' này khi có hai hoặc nhiều lớp lable. Các lable sẽ được **cung cấp dưới dạng một số nguyên**.

**Bước 6: model.summary()**

****

**Bước 7: Đào tạo mô hình**

****

****

Kết quả đào tạo cũng khá tốt với phần Train nhưng phần Test không ổn cho lắm val\_loss đang khá cao.

**Biểu đồ tổng hợp kết quả đào tạo**



**Overfitting**

Trong các biểu đồ ở trên, độ chính xác của quá trình huấn luyện đang tăng tuyến tính theo thời gian, trong khi độ chính xác của việc xác nhận là khoảng **80%** trong quá trình huấn luyện. Ngoài ra, sự khác biệt về độ chính xác giữa đào tạo và độ chính xác xác thực là đáng chú ý.

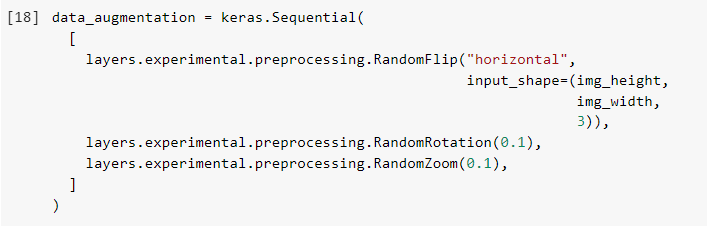
Khi có một số lượng nhỏ các ví dụ đào tạo, mô hình đôi khi học hỏi từ các yếu tố ánh sáng hoặc các chi tiết không mong muốn từ các ví dụ đào tạo — đến mức nó tác động tiêu cực đến hiệu suất của mô hình đối với các ví dụ mới. Hiện tượng này được gọi là overfitting. Có nghĩa là mô hình sẽ gặp khó khăn khi tổng quát hóa trên một tập dữ liệu mới.

Có nhiều cách để chống lại tình trạng này trong quá trình luyện tập. Sẽ sử dụng tăng cường dữ liệu và thêm Dropout vào mô hình.

**Tăng dữ liệu**

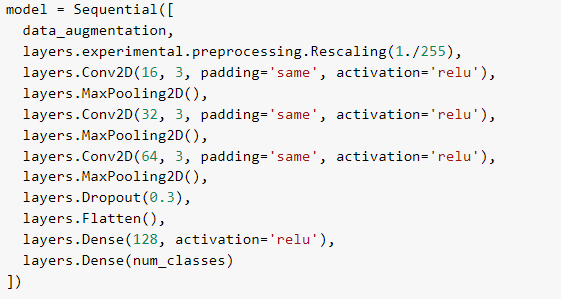
Overfitting thường xảy ra khi có một số lượng nhỏ các ví dụ đào tạo. Tăng cường dữ liệu áp dụng phương pháp tạo dữ liệu đào tạo bổ sung từ các ví dụ hiện có của bạn bằng cách tăng cường chúng bằng cách sử dụng các phép biến đổi ngẫu nhiên mang lại hình ảnh trông đáng tin cậy. Điều này giúp mô hình hiển thị nhiều khía cạnh của dữ liệu hơn và khái quát hóa tốt hơn.

Thực hiện tăng dữ liệu bằng cách sử dụng các lớp từ tf.keras.layers.experimental.preprocessing . Chúng có thể được đưa vào bên trong mô hình của bạn giống như các lớp khác và chạy trên GPU.

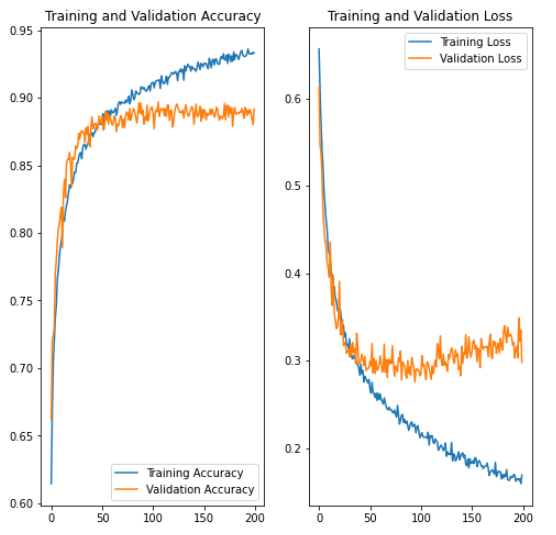




**Sử dụng Dropout 30 %:**

****

**Đào tạo tiếp mô hình và sau 200 epoch:**

****

Sau việc tăng dữ liệu và Dropout thì ta thấy val\_acc đã lên đến gần **88%**. Và Loss cũng đã giảm đáng kể.

### 4.2.2. Sử dụng CNN Model với CategoricalCrossentropy

**Bước 1: Đọc dữ liệu vào và chuẩn hóa:**



X là dữ liệu ảnh với kích thước (48,48,3)

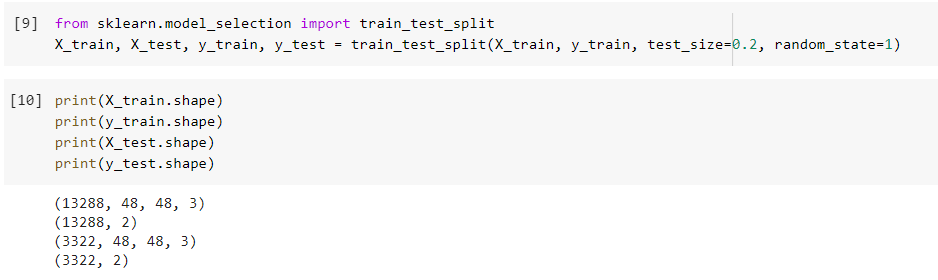
y là: [[1,0] [0,1]]

(với [1,0] : là Happy và [0,1]: là Sad)

**Chuẩn hóa dữ liệu đầu vào trong khoảng [0,1]**



**Bước 2: Tăng dữ liệu đầu vào và chia dữ liệu Train - Test:**



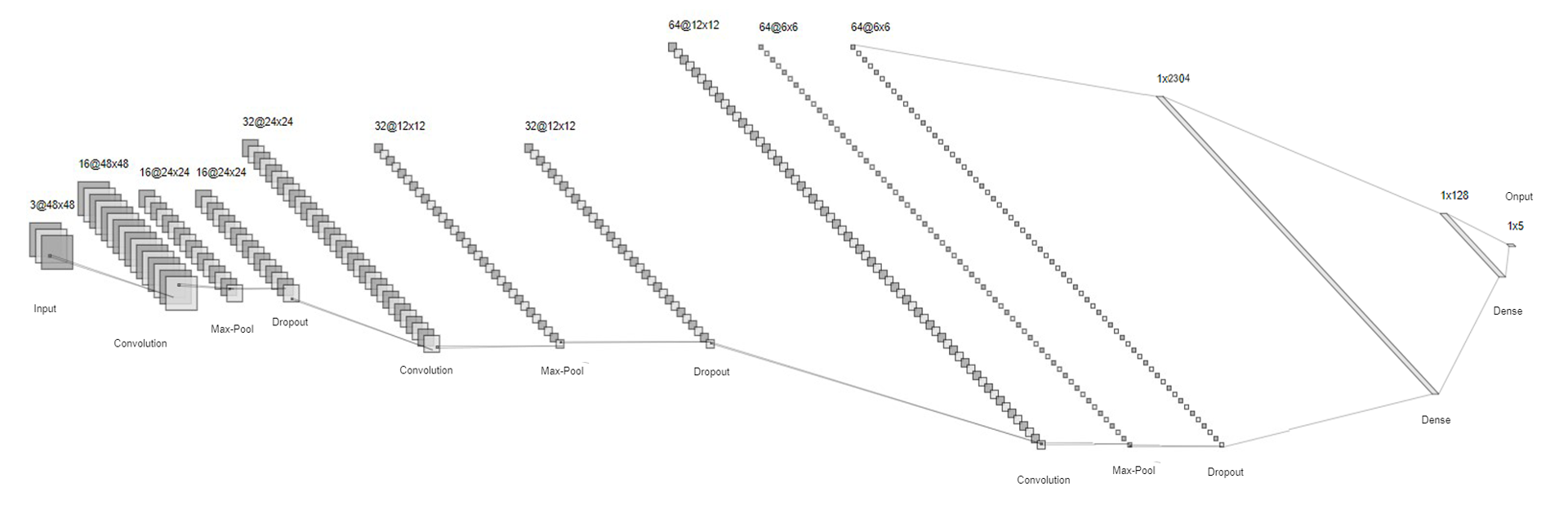
Tập dữ liệu Train chiếm 80%.

Tập dữ liệu Test chiếm 20%.

**Bước 3: Xây dựng mô hình CNN:**

|  |
| --- |
| batch\_size = 16  img\_height = 48  img\_width = 48  num\_classes=2  inp = Input(shape = (img\_height, img\_width, 3))  cnn = Conv2D(filters = 16, kernel\_size = 3,padding='same', activation = 'relu') (inp)  pooling = MaxPooling2D((2,2)) (cnn)  drop=Dropout(0.2)(pooling)  cnn = Conv2D(filters = 32, kernel\_size = 3, padding='same', activation = 'relu') (drop)  pooling = MaxPooling2D((2,2)) (cnn)  drop=Dropout(0.2)(pooling)  cnn = Conv2D(filters = 64, kernel\_size = 3, padding='same', activation = 'relu') (drop)  pooling = MaxPooling2D((2,2)) (cnn)  drop=Dropout(0.2)(pooling)  f = Flatten()(drop)  fc1 = Dense(units = 128, activation = 'relu') (f)  out = Dense(units = num\_classes, activation = 'softmax') (fc1)  model = Model(inputs = inp, outputs = out)  model.summary() |

Mô hình CNN của có input đầu vào là ảnh có kích thước 48\*48 với 3 kênh màu RGB. Sau đó tạo lớp tích chập 2D  kết hợp với dữ liệu đầu vào để tạo ra các đầu ra mới. Tiếp đó, sử dụng max pooling để  giảm giá trị đầu vào bằng cách lấy giá trị lớn nhất trong từng pool\_size. Rồi sử dụng Dropout với 20%. Tương tự lặp lại thêm 2 bước tích chập, pooling và Drop ta được kết quả như hình sau:



**Sơ đồ các lớp của mô hình CNN.**

Sau các bước Convolution và Max\_pool thì ta làm phẳng shape bằng Flatten().

Các lớp được kết nối đầy đủ được xác định bằng cách sử dụng [lớp Dense](https://keras.io/layers/core/) . Chúng ta có thể chỉ định số lượng nơron hoặc nút trong lớp làm đối số đầu tiên và chỉ định hàm kích hoạt bằng cách sử dụng đối số activation. Và sử dụng chức năng kích hoạt đơn vị tuyến tính được chỉnh lưu được gọi là ReLU trên lớp đầu và softmax trong lớp 2. Từ đây ta xác định được inputs và outputs của mô hình.

**Bước 4: Biên dịch mô hình:**

|  |
| --- |
| opt = keras.optimizers.Adam(learning\_rate=0.001)  model.compile(loss='categorical\_crossentropy', optimizer =opt, metrics=['accuracy']) |

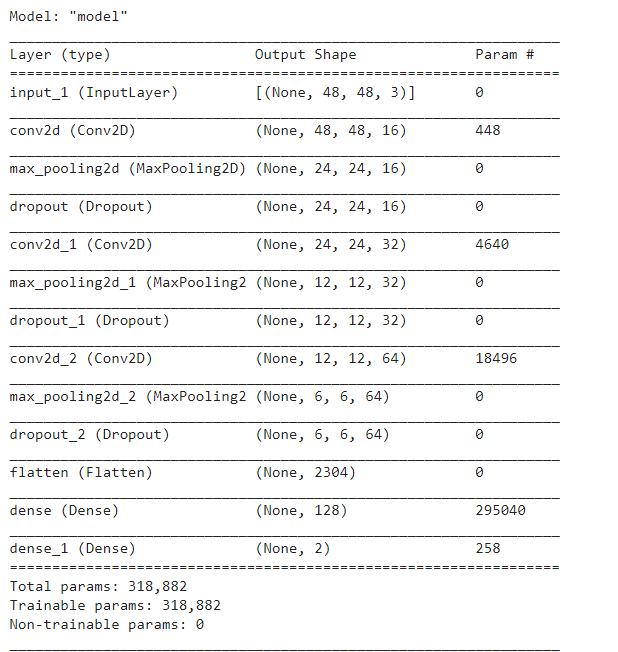
Ở đây, sử dụng hàm mất mát (loss) là 'categorical\_crossentropy'. Categorical crossentropy dùng để tính toán mất mát giữa các lable và dự đoán. Sử dụng 'categorical\_crossentropy' này khi có hai hoặc nhiều lớp lable. Các lable sẽ được **cung cấp dưới dạng one\_hot**.

Và sử dụng công cụ tối ưu hóa Adam để đạt đến mức tối thiểu toàn cầu trong khi đào tạo mô hình. Nếu bị mắc kẹt trong cực tiểu cục bộ trong khi đào tạo thì trình tối ưu hóa adam sẽ giúp thoát khỏi cực tiểu cục bộ (local minimum) và đạt tới điểm tối ưu nhất (flat minimum).

Ngoài ra có một chức năng được sử dụng để đánh giá hiệu xuất của mô hình là metrics.

Có thể kiểm tra tóm tắt của mô hình mà tôi đã đào tạo bằng cách sử dụng đoạn mã sau:

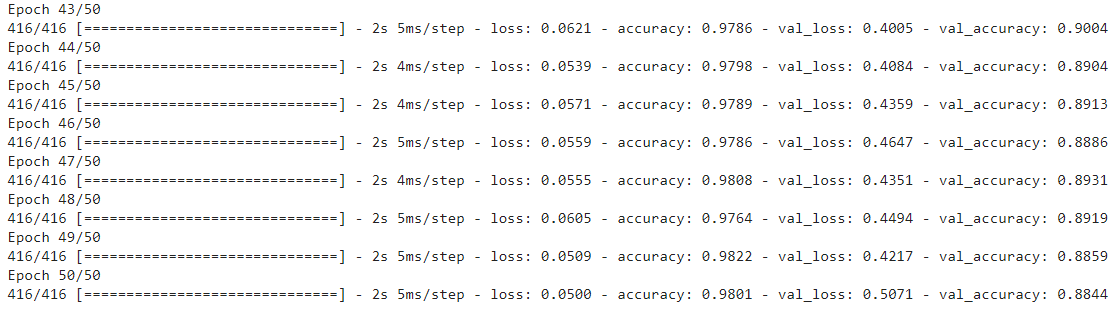
|  |
| --- |
| model.summary( ) |

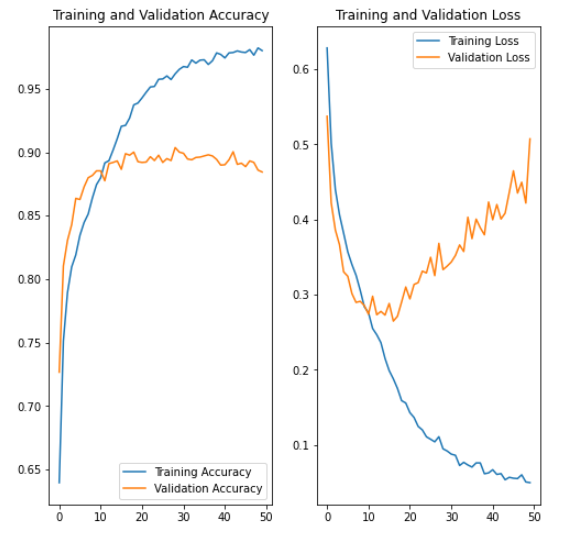


**Tóm tắt mô hình.**

**Bước 5: Đào tạo mô hình:**

|  |
| --- |
| epochs = 50  history = model.fit(    X\_train,y\_train,    epochs=epochs,    validation\_data=(X\_test,y\_test)  ) |





Trong các biểu đồ ở trên, độ chính xác của quá trình huấn luyện đang tăng tuyến tính theo thời gian, trong khi độ chính xác của việc xác nhận là khoảng **90%** trong quá trình huấn luyện.

### \*\* Kết quả so sánh 2 model:

Thực hiện test model bằng 100 ảnh rõ nét cảm xúc khuôn mặt người. Trong đó 50 ảnh là Happy, 50 ảnh là Sad.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Model 1** | | | **Model 2** | | |
| True | False | (%) | True | False | (%) |
| **Happy** | 49 | 1 | 98 | 49 | 1 | 98 |
| **Sad** | 39 | 11 | 78 | 42 | 7 | 84 |

Cho chạy trên 2 model khác nhau và cho ra được kết quả khá là khả quan:

Với 50 ảnh Happy thì 2 model cho tỉ lệ đúng lên đến: **98%** và đều nhận diện sai cùng 1 ảnh thứ 46.

Với 50 ảnh Sad thì 2 model cho tỷ lệ đúng hơi thấp: vì dữ liệu test có cả ảnh không đầy đủ khuôn mặt và có ảnh không detect được mặt. Cảm xúc không rõ.

**Chương 5: Kết quả**

**5.1. Kết quả thu được**

Sau quá trình tìm hiểu đề tài và nghiên cứu đề tài. Tôi đã tìm hiểu được rất nhiều phương pháp khác nhau để giải quyết bài toán phát hiện biển số xe để có thể ứng dụng và phục vụ tại trường Đại học Phenikaa. Mô hình của tối được xây dựng trên thuật toán CNN. Với kết quả trả về tương đối chính xác.

**Bảng thể hiện kết quả sau khi chạy mô hình.**

|  |  |  |
| --- | --- | --- |
| Ảnh gốc | Ảnh detect | Kết quả dự đoán |
|  |  | Model 1:  Model 2: |
|  |  | Model 1:    Model 2: |
|  |  | Model 1:    Model 2: |

**5.2. Ưu điểm, nhược điểm**

**5.2.1. Ưu điểm**

* Mô hình nhận diện tương đối chính xác trên hầu hết các ảnh.
* Nhận diện được 1 cảm xúc trên 1 khung hình.
* Nhận diện được mặt ở nhiều trạng thái khác nhau như: nghiêng về nhiều hướng khác nhau.
* Mô hình nhận diện tốt với dữ liệu ảnh thực tế.
* Dễ sử dụng
* Có thể sử dụng để nhận biết khuôn mặt trong ảnh và cả video

**5.2.2. Nhược điểm**

* Mô hình chưa thể nhận diện chính xác được tất cả các ảnh do hạn chế của lượng dữ liệu huấn luyện.
* Mô hình nhận diện yếu khi mặt ở khu vực viền ảnh hoặc bị cắt.
* Mô hình nhận diện rất yếu khi ảnh đầu vào ở nơi thiếu ánh sáng, bị mờ, chất lượng hình ảnh không tốt, ... có thể nhận diện sai hoặc không nhận diện được.
* Mô hình nhận diện yếu khi có vật thể che (khẩu trang, kính,...) hoặc bị cắt do khung hình.
* Mô hình nhận diện chưa tốt trong các điều kiện môi trường như tối, hoặc sáng chói.

**5.3. Hướng phát triển**

* Cần bổ sung thêm nhiều dữ liệu để tối ưu hóa mô hình như dữ liệu ảnh buổi tối, ảnh mờ, ảnh nhiễu, ảnh sáng chói, ảnh ở nhiều vị trí khác nhau trên khung hình, ảnh bị vật thể che, ...
* Thực thi trên nhiều class hơn 5 class, 7 class để tối đa hóa nhận dạng được các biểu cảm khuôn mặt.
* Có thể áp dụng thêm vào nhiều lĩnh vực dùng để đánh giá sản phẩm người dùng, chất lượng phục vụ cũng như theo dõi sức khỏe con người.