

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN**  
**NHẬP MÔN THỊ GIÁC MÁY TÍNH**  
**CS231.M21.KHCL**

**ĐỀ TÀI: NHẬN DIỆN CẢM XÚC KHUÔN MẶT**

**GIẢNG VIÊN HƯỚNG DẪN: TS. MAI TIẾN DŨNG**

**SINH VIÊN THỰC HIỆN: BÙI QUỐC THỊNH – 20520934**  
**HOÀNG ĐÌNH HỮU - 20521384**

**TP. HỒ CHÍ MINH, 6/2022**

## MỤC LỤC

Phần 1. Giới thiệu về môn học.....	3
1.1 Khái niệm trí tuệ nhân tạo.....	3
1.2 Khái niệm thị giác máy tính.....	3
1.3 Khái niệm xử lý ảnh.....	3
Phần 2: Giới thiệu về đề tài.....	4
2.1 Ứng dụng nhận diện cảm xúc khuôn mặt trong đời sống và mục đích.....	4
Phần 3. Các kỹ thuật nhận diện cảm xúc khuôn mặt.....	5
3.1 Nhận diện cảm xúc khuôn mặt bằng CNN.....	5
3.2 Nhận diện cảm xúc khuôn mặt bằng SVM.....	10
3.3 Nhận diện cảm xúc khuôn mặt bằng thư viện DeepFace.....	13
Phần 4. Thực nghiệm.....	14
4.1 Input và output.....	14
4.2 Bộ dataset FER-2013.....	14
Phần 5: Kết quả.....	15
5.1 Kết quả CNN.....	15
5.2 Kết quả SVM.....	16
5.3 Kết quả DeepFace.....	17
Phần 6. So sánh.....	19
Tài liệu tham khảo.....	20

## **Phần 1: Giới thiệu môn học**

### **1.1 Khái niệm về Trí tuệ nhân tạo**

AI là trí thông minh được thể hiện bởi máy móc, không giống như trí thông minh tự nhiên được hiển thị bởi con người và động vật, liên quan đến ý thức và cảm xúc.

Học máy là nghiên cứu các thuật toán máy tính cải tiến tự động thông qua kinh nghiệm và dữ liệu. Nó được xem như một phần của trí tuệ nhân tạo.

Học sâu là một phần của họ các phương pháp học máy rộng hơn dựa trên mạng nơ-ron nhân tạo với học đại diện. Việc học có thể được giám sát, bán giám sát hoặc không giám sát.

### **1.2 Khái niệm về thị giác máy tính**

Thị giác máy tính (tiếng Anh: 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 và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định.

### **1.3 Khái niệm về xử lý hình ảnh**

Xử lý hình ảnh là việc sử dụng máy tính kỹ thuật số để xử lý hình ảnh kỹ thuật số thông qua một thuật toán. Từ đó, chúng ta có thể nâng cao các bức ảnh hoặc trích xuất thông tin hữu ích từ chúng.

## Phần 2: Giới thiệu về đề tài

Cảm xúc được thể hiện khi tương tác và giao tiếp với người khác. Nghiên cứu cách đọc chúng có thể là một nhiệm vụ khó khăn, vì vậy công nghệ được sử dụng để thực hiện công việc đó. Nhận dạng cảm xúc được sử dụng bởi nhiều tổ chức ngày nay, nhưng nó chính xác là gì?

Nhận dạng cảm xúc là một trong nhiều công nghệ nhận dạng khuôn mặt đã phát triển và lớn mạnh trong những năm qua. Hiện tại, phần mềm nhận dạng cảm xúc trên khuôn mặt được sử dụng để cho phép một chương trình nhất định kiểm tra và xử lý các biểu hiện trên khuôn mặt của con người. Sử dụng chức năng phân phối hình ảnh tiên tiến, phần mềm này hoạt động giống như bộ não con người, giúp nó có khả năng nhận biết cảm xúc.

Đó là AI hoặc "Trí tuệ nhân tạo" phát hiện và nghiên cứu các biểu hiện khuôn mặt khác nhau để sử dụng chúng với thông tin bổ sung được cung cấp cho họ. Điều này hữu ích cho nhiều mục đích khác nhau, bao gồm điều tra và phỏng vấn, đồng thời cho phép nhà chức trách phát hiện cảm xúc của một người chỉ bằng việc sử dụng công nghệ.

### **2.1 Ứng dụng nhận diện cảm xúc khuôn mặt trong đời sống và mục đích**

Những diễn giả, các nhà diễn thuyết, diễn viên hoặc ca sĩ luôn đứng trên sân khấu và thể hiện những phần trình diễn tốt nhất của họ. Tuy nhiên, bản thân những con người lại không thể nào biết rằng khán giả của họ có thật sự hài lòng với phần trình diễn hay không. Đó là lý do tại sao việc nhận diện cảm xúc khuôn mặt có thể hỗ trợ cho những vấn đề này.

Có số lượng rất lớn những bệnh nhân trong bệnh viện nhưng bác sĩ và y tá thì lại có hạn nên việc quản lý và kiểm tra bệnh nhân gây khá nhiều trở ngại. Việc ứng dụng nhận diện cảm xúc khuôn mặt có thể hỗ trợ các bác sĩ cũng như y tá không phải túc trực thường xuyên mà vẫn có thể nhận biết được những biểu hiện của bệnh nhân để từ đó có thể quan sát tình trạng của bệnh nhân thông qua các cử chỉ trên khuôn mặt.

Những tên tội phạm hoặc khủng bố luôn có những hành tung thất thường, người dân cũng thế. Việc ứng dụng nhận diện cảm xúc khuôn mặt vào giám sát công dân có thể cho các cơ quan chức trách cũng như cơ quan hành pháp nhận biết được những mối nguy trước mắt có thể xảy ra.

## Phần 3: Các kỹ thuật nhận diện cảm xúc khuôn mặt

Để có thể nhận diện được cảm xúc khuôn mặt thì chúng ta có hai hình thức chính là thông qua phương thức tầm nhìn:

- Camera/ Video
- Hình ảnh

Hoặc là phương thức Tín hiệu sinh học/Sinh lý học

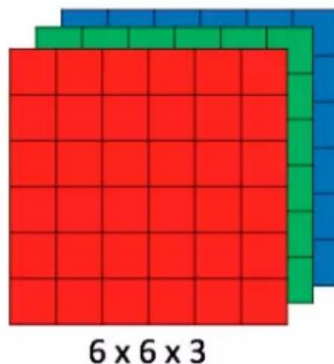
- PPG
- Điện tâm đồ
- Điện não đồ

Ở đây, với mục tiêu ứng dụng thị giác máy tính nên chúng ta sẽ ứng dụng hình thức tầm nhìn.

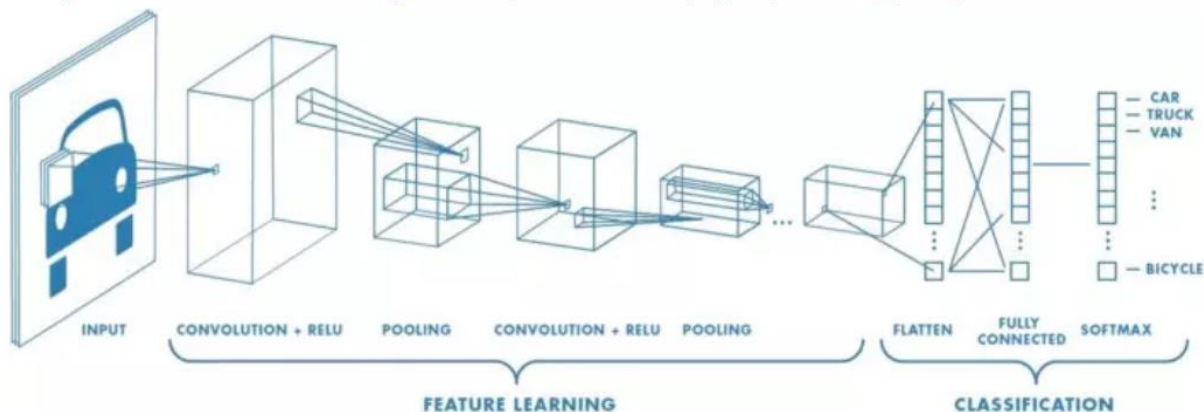
### 3.1 Nhận diện cảm xúc khuôn mặt bằng CNN

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  $H \times W \times D$  (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB  $6 \times 6 \times 3$  (3 ở đây là giá trị RGB).

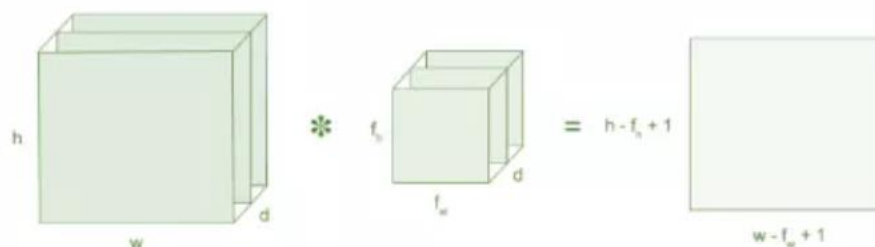


Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ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



Tích chập (Convolution Layer) 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 vuông 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.

- An image matrix (volume) of dimension **( $h \times w \times d$ )**
- A filter ( **$f_h \times f_w \times d$** )
- Outputs a volume dimension **( $h - f_h + 1$ )  $\times$  ( $w - f_w + 1$ )  $\times$  1**



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.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

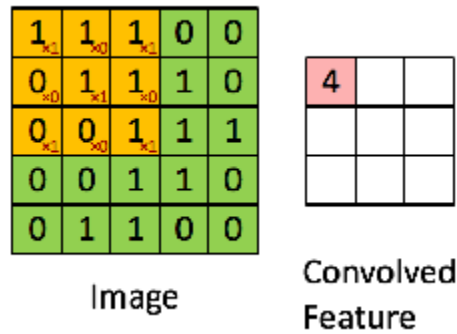
\*

1	0	1
0	1	0
1	0	1



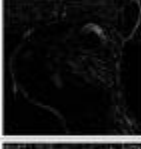




**5 x 5 – Image Matrix**

**3 x 3 – Filter Matrix**

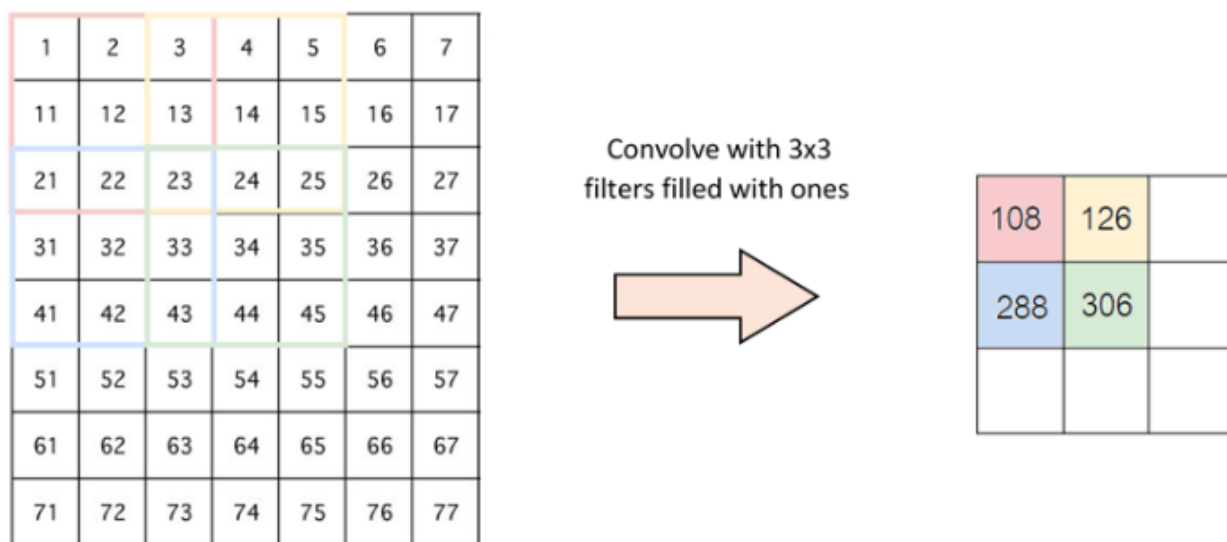
Sau đó, lớp tích chập của ma trận hình ảnh 5 x 5 nhân với ma trận bộ lọc 3 x 3 gọi là 'Feature Map' như hình bên dưới.



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.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Bước nhảy (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.

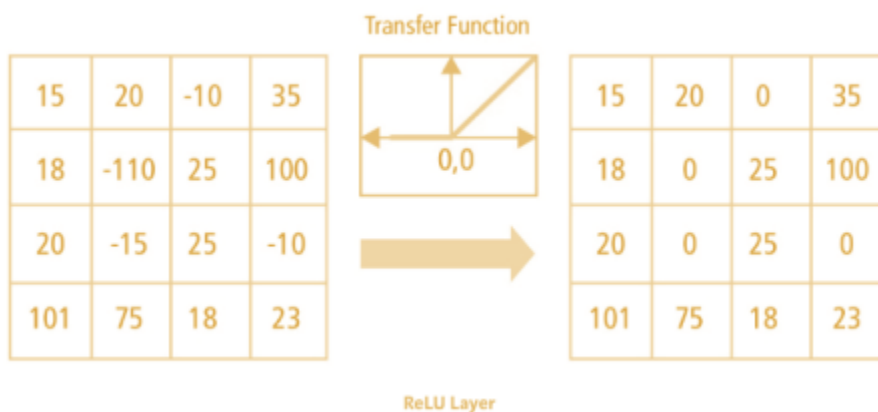


Đường viền (Padding). Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:

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

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

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 hàm 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.

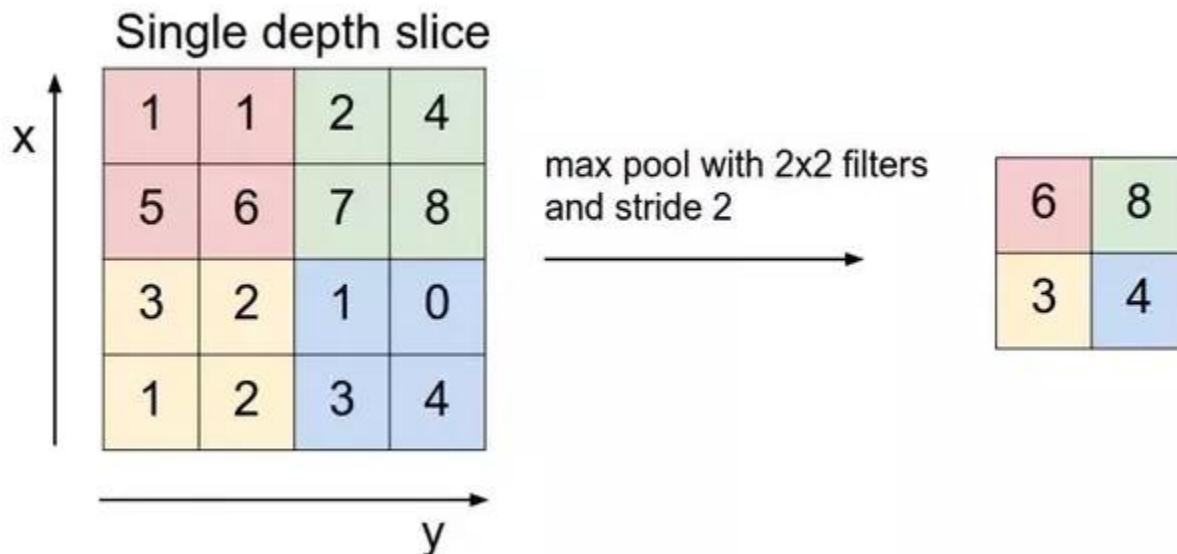
Lớp gộp (Pooling layer) sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu xuống làm giảm kích thước của mỗi map nhưng vẫn giữ lại thông tin quan trọng. Các pooling có thể có nhiều loại khác nhau:

- 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.



Trong đề tài này, chúng tôi đã tự xây dựng mô hình CNN riêng để thực hiện việc huấn luyện. Mô hình được xây dựng như hình dưới:

```
''' First layer '''
emotion_model.add(Conv2D(filters=64,kernel_size=(5,5),input_shape=(48, 48, 1),activation='relu',padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(Conv2D(filters=64,kernel_size=(5,5),activation='relu',padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(MaxPooling2D(pool_size=(2,2)))
emotion_model.add(Dropout(0.5))

''' Second layer '''
emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.5))

'''Extra layer'''
emotion_model.add(Conv2D(filters=256,kernel_size=(3,3),activation='relu',padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(Conv2D(filters=256,kernel_size=(3,3),activation='relu',padding='same',kernel_initializer='he_normal'))
emotion_model.add(BatchNormalization())
emotion_model.add(MaxPooling2D(pool_size=(2,2)))
emotion_model.add(Dropout(0.5))

'''Fully connected layer'''
emotion_model.add(Flatten())
emotion_model.add(Dense(128, activation='relu', kernel_initializer='he_normal'))
emotion_model.add(Dropout(0.6))
emotion_model.add(Dense(7, activation='softmax'))
```

Mô hình được xây dựng với 2 lớp chính, 1 lớp thêm và 1 lớp Fully Connected dựa trên mô hình Sequential.

Mô hình Sequential liên quan đến việc xác định một lớp Sequential và thêm từng lớp vào mô hình theo cách tuyến tính, từ đầu vào đến đầu ra.

Convolutional Layers: Conv2D là convolution dùng để lấy feature từ ảnh với các tham số:

- filters: số filter của convolution.
- kernel\_size: kích thước window search trên ảnh.
- strides: số bước nhảy trên ảnh.
- activation: chọn activation như linear, softmax, relu, tanh, sigmoid. Đặc điểm mỗi hàm các bạn có thể search thêm để biết cụ thể nó như thế nào.
- padding: có thể là "valid" hoặc "same". Với same thì có nghĩa là padding =1.
- Kernel\_initializer: xác định cách đặt trọng số ngẫu nhiên ban đầu của các lớp Keras. Bộ khởi tạo cho ma trận trọng số hạt nhân (xem keras.initializers). Mặc định là 'glorot\_uniform'.

Pooling Layers: sử dụng để làm giảm param khi train, nhưng vẫn giữ được đặc trưng của ảnh:

- pool\_size: kích thước ma trận để lấy max hay average.
- Ngoài ra còn có: MaxPooling2D, AveragePooling1D, 2D (lấy max, trung bình) với từng size.

Batch Normalization là một phương pháp hiệu quả khi training một mô hình mạng nơ ron. Mục tiêu của phương pháp này chính là việc muốn chuẩn hóa các feature (đầu ra của mỗi layer sau khi đi qua các activation) về trạng thái zero-mean với độ lệch chuẩn 1.

Lớp Dropout đặt ngẫu nhiên các đơn vị đầu vào thành 0 với tần suất tỷ lệ ở mỗi bước trong thời gian đào tạo, điều này giúp ngăn ngừa việc trang bị quá mức. Các đầu vào không được đặt thành 0 được tăng tỷ lệ lên  $1 / (1 - \text{tỷ lệ})$  sao cho tổng trên tất cả các đầu vào là không thay đổi.

Làm phẳng (Flatten) một tensor có nghĩa là loại bỏ tất cả các kích thước ngoại trừ một kích thước. Một lớp Flatten trong Keras định hình lại tensor để có hình dạng bằng số phần tử có trong tensor.

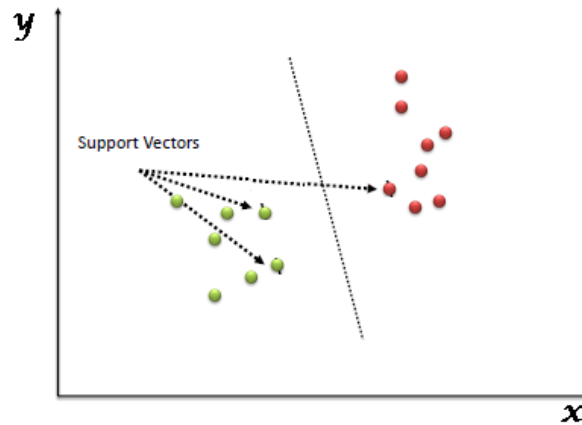
Dense (): Layer này cũng như một layer neural network bình thường, với các tham số sau:

- units: số chiều output, như số class sau khi train (chó, mèo, lợn, gà).
- activation: chọn activation đơn giản với sigmoid thì output có 1 class.
- use\_bias: có sử dụng bias hay không (True or False).

### **3.2 Nhận diện cảm xúc khuôn mặt bằng SVM**

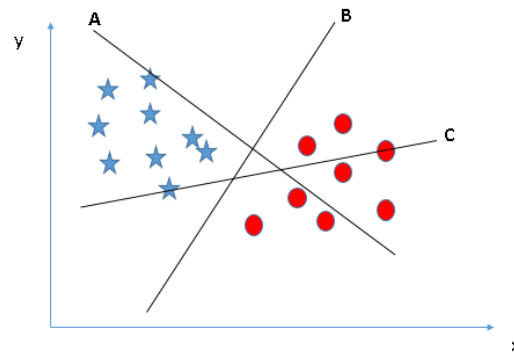
SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất.



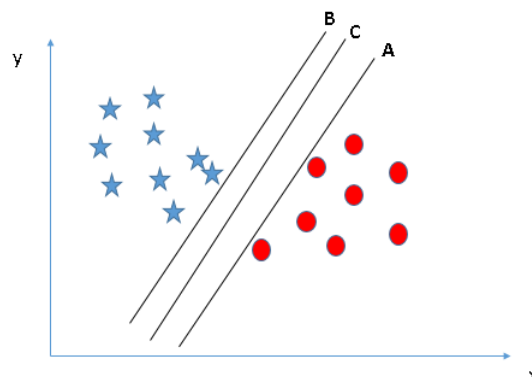
Xác định "Làm sao để vẽ-xác định đúng hyper-plane". Chúng ta sẽ theo các tiêu chí sau:

Có 3 đường hyper-lane (A, B và C). Bây giờ đường nào là hyper-lane đúng cho nhóm ngôi sao và hình tròn.



Quy tắc số một để chọn 1 hyper-lane, chọn một hyper-plane để phân chia hai lớp tốt nhất. Trong ví dụ này chính là đường B.

Quy tắc thứ hai chính là xác định khoảng cách lớn nhất từ điều gần nhất của một lớp nào đó đến đường hyper-plane. Khoảng cách này được gọi là "Margin", Hãy nhìn hình bên dưới, trong đây có thể nhìn thấy khoảng cách margin lớn nhất đây là đường C. Cần nhớ nếu chọn làm hyper-lane có margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu.



Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Trong ví dụ quả táo quả lê đặt trên mặt bán, margin chính là khoảng cách giữa cây que và hai quả táo và lê gần nó nhất. Điều quan trọng ở đây đó là phương pháp SVM luôn cố gắng cực đại hóa margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả táo và lê. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào.

Cuối cùng, chúng tôi đã thử sử dụng Biểu đồ Gradients định hướng (HOG) để mô tả sự phân bố của gradient và hướng cạnh trong hình ảnh trước khi xử lý chúng. Ý tưởng đằng sau việc sử dụng HOG là những cảm xúc khác nhau sẽ có độ chuyển màu khác nhau và riêng biệt, đặc biệt là xung quanh vùng miệng và mắt. Mặc dù HOG không giúp ích đáng kể cho độ chính xác của SVM, nhưng nó đã nâng độ chính xác của bộ phân loại lên đến độ chính xác SVM cao nhất của chúng tôi là 53,81%.

HOG (histogram of oriented gradients) là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh, dùng để detect một đối tượng.

Hog được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một object trong ảnh. Bài toán tính toán Hog thường gồm 5 bước:

- Chuẩn hóa hình ảnh trước khi xử lý.
- Tính toán gradient theo cả hướng x và y.
- Lấy phiếu bầu cùng trọng số trong các cell.
- Chuẩn hóa các block.
- Thu thập tất cả các biểu đồ cường độ gradient định hướng để tạo ra feature vector cuối cùng.



### **3.3 Nhận diện cảm xúc khuôn mặt bằng thư viện DeepFace**

Deepface là một framework nhận dạng khuôn mặt và phân tích thuộc tính khuôn mặt (tuổi, giới tính, cảm xúc và chủng tộc) nhẹ cho python. Đây là một framework nhận dạng khuôn mặt kết hợp bao gồm các mô hình hiện đại: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib và SFace.

Các thí nghiệm cho thấy con người có độ chính xác 97,53% đối với các tác vụ nhận dạng khuôn mặt trong khi các mô hình đó đã đạt và vượt qua mức độ chính xác đó.

Nếu chạy tính năng nhận dạng khuôn mặt với DeepFace, chúng ta sẽ có quyền truy cập vào một loạt các tính năng: Xác minh khuôn mặt, Nhận dạng khuôn mặt, Phân tích thuộc tính khuôn mặt, Phân tích khuôn mặt theo thời gian thực.

## **Phần 4: Thực nghiệm**

### **4.1 Input và output**

Input: Hình ảnh chính diện của khuôn mặt bao gồm mắt, mũi, miệng, tai.

Output: Một trong bảy cảm xúc trong dữ liệu huấn luyện.

### **4.2 Bộ dataset FER-2013**

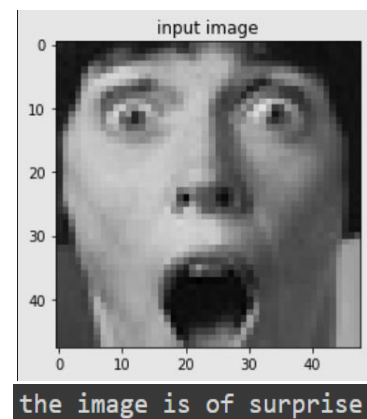
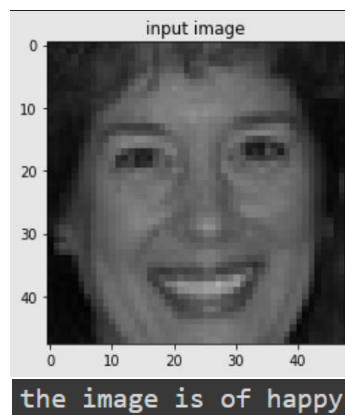
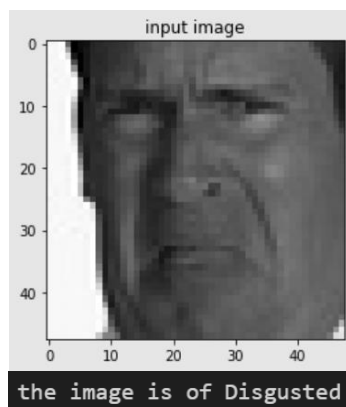
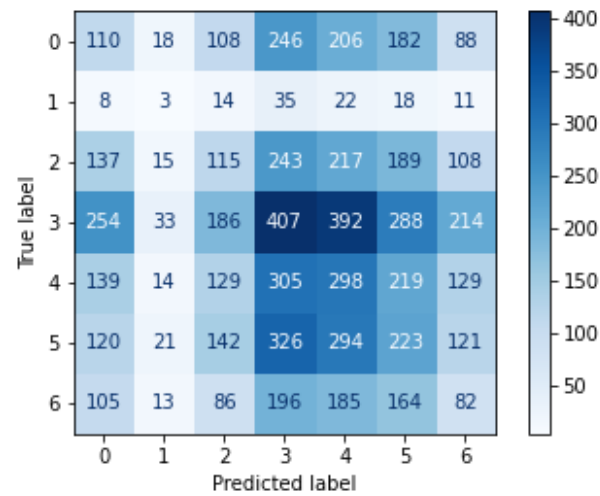
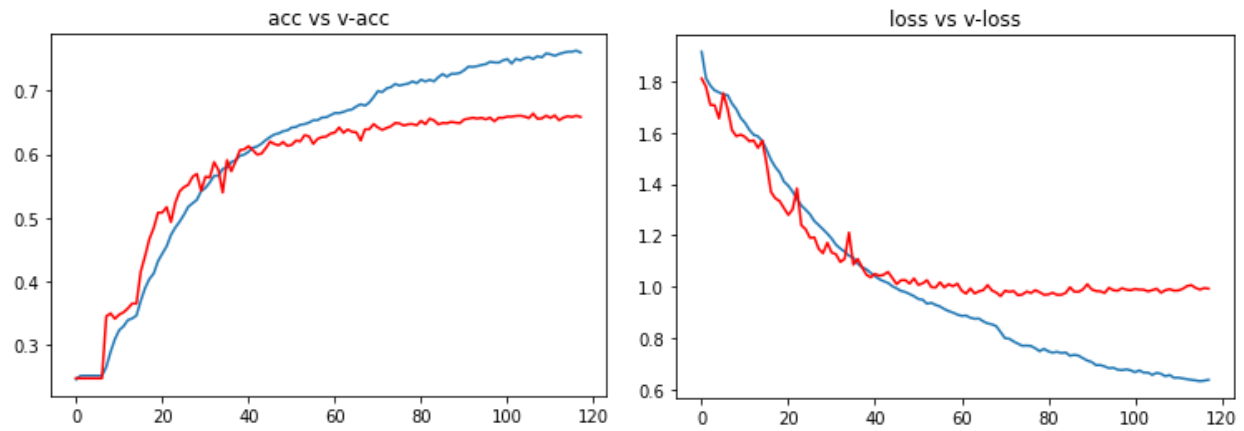
Dataset FER-2013 được tạo bằng cách thu thập kết quả tìm kiếm hình ảnh trên Google về từng cảm xúc và từ đồng nghĩa của cảm xúc

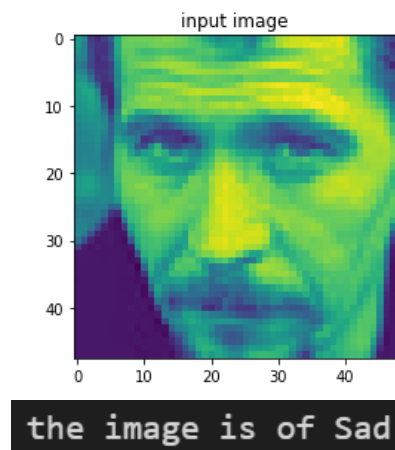
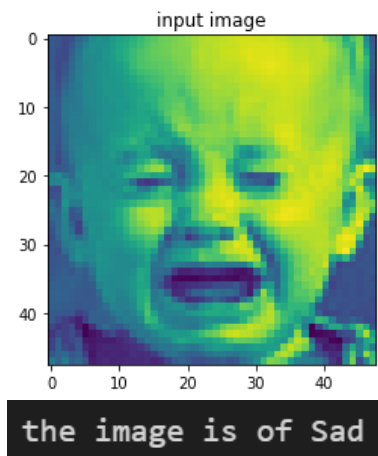
Dữ liệu bao gồm hình ảnh thang độ xám 48x48 pixel của khuôn mặt. Các khuôn mặt đã được đăng ký tự động để khuôn mặt được căn giữa nhiều hơn hoặc ít hơn và chiếm khoảng không gian như nhau trong mỗi hình ảnh.

Nhiệm vụ là phân loại từng khuôn mặt dựa trên cảm xúc thể hiện trên nét mặt thành một trong bảy loại (0 = Giận dữ, 1 = Chán ghét, 2 = Sợ hãi, 3 = Hạnh phúc, 4 = Buồn, 5 = Bất ngờ, 6 = Trung lập). Tập huấn luyện bao gồm 28.709 ví dụ và tập kiểm tra công khai bao gồm 7178 ví dụ.

## Phần 5: Kết quả

### 5.1 Kết quả CNN





Kết quả sau khi thực hiện theo phương thức CNN được độ chính xác 76,35% và độ chính xác trên tập test là 66.09%.

## 5.2 Kết quả SVM

Kết quả sau khi thực hiện theo phương thức SVM được độ chính xác 53,21% với kernel rbf và 53,81% với kernel poly.

Kernel rbf

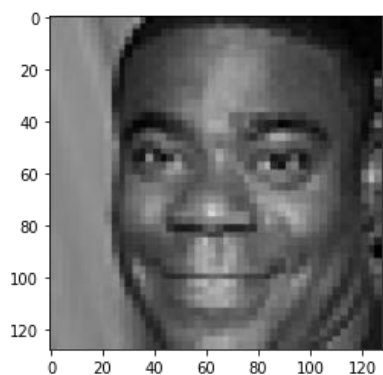
```
[[ 346  0  80 169 154 181 28]
 [ 34 22 14 13 8 17 3]
 [ 116 0 311 140 142 227 88]
 [ 53 0 63 1416 98 121 23]
 [ 81 0 73 191 660 203 25]
 [ 136 0 85 210 237 555 24]
 [ 42 0 66 92 65 56 510]]

0.5321816662022848
```

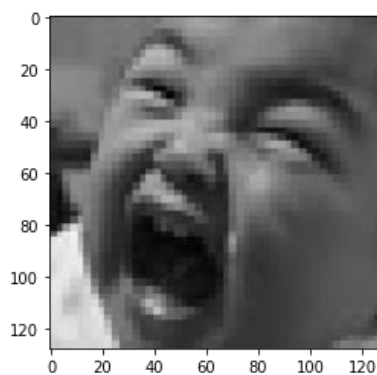
Kernel Poly

```
[[ 429  6 113 130 116 148 16]
 [ 26 55 10 7 3 9 1]
 [ 152 8 417 101 108 167 71]
 [ 98 2 79 1321 131 108 35]
 [ 146 1 120 180 587 170 29]
 [ 199 7 173 149 207 490 22]
 [ 46 0 78 66 40 37 564]]

0.5381721928113681
```

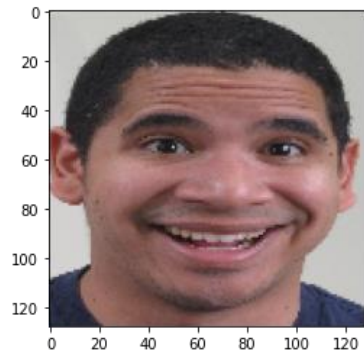


happy  
happy

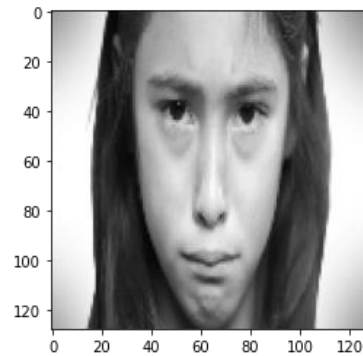


angry  
angry

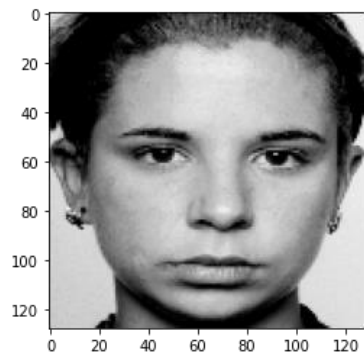




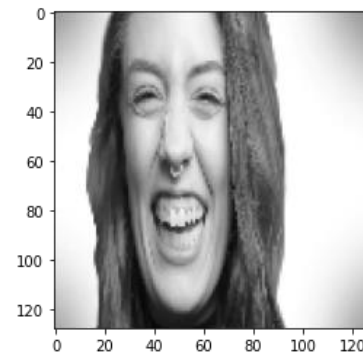
sad  
sad



sad  
sad



neutral  
neutral



sad  
sad

### 5.3 Kết quả DeepFace

Vì Deepface là một gói nhận dạng khuôn mặt lại. Nó hiện bao gồm nhiều mô hình nhận dạng khuôn mặt hiện đại: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib và SFace. Cấu hình mặc định sử dụng mô hình VGG-Face.

Model	LFW Score	YTF Score
Facenet512	99.65%	-
SFace	99.60%	-
ArcFace	99.41%	-
Dlib	99.38 %	-
Facenet	99.20%	-
VGG-Face	98.78%	97.40%
<i>Human-beings</i>	97.53%	-
OpenFace	93.80%	-
DeepID	-	97.05%



## **Phần 6: So sánh**

Có thể thấy được sự chênh lệch rõ rệt trong các mô hình khác nhau, mỗi mô hình có ưu và nhược điểm khác nhau.

Thư viện DeepFace cho ra độ chính xác cao nhất cũng như thể hiện sự ổn định cao nhất trong thực nghiệm lần demo thời gian thực.

Kế tiếp là mô hình CNN tự xây dựng bởi nhóm và cuối cùng là phương pháp SVM.

Kết luận cho thấy phương pháp sử dụng SVM và đặc trưng HOG vẫn còn nhiều hạn chế nên không phù hợp với bài toán đề ra.

## Tài liệu tham khảo

[1] Real-time Emotion Recognition from Facial Expressions by Minh-An Quinn, Grant Sivesind, Guilherme Reis in CS229 - Stanford University

<http://cs229.stanford.edu/proj2017/final-reports/5243420.pdf>

[2] Project description of DeepFace

<https://pypi.org/project/deepface/>

[3] Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville published by MIT Press, 2016

[4] Stanford University's Course — CS231n: Convolutional Neural Network for Visual Recognition by Prof. Fei-Fei Li, Justin Johnson, Serena Yeung

[5] Mayank Mishra, "Convolutional Neural Networks, Explained", Medium, Aug 27, 2020

<https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939#:~:text=A%20CNN%20typically%20has%20three,and%20a%20fully%20connected%20layer.>

[6] Keras layers API

<https://keras.io/api/layers/>

[7] [Deep Learning] Tìm hiểu về mạng tích chập (CNN) by Chung Pham Van

<https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>

[8] Giới thiệu về Support Vector Machine (SVM) by Huynh Chi Trung

<https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVEImB>

[9] Emotion Recognition: Introduction to Emotion Reading Technology

<https://recfaces.com/articles/emotion-recognition>

[10] Nhận diện cảm xúc khuôn mặt đơn giản với Keras by To Duc Thang

<https://viblo.asia/p/nhan-dien-cam-xuc-khuon-mat-don-gian-voi-keras-V3m5WvRwlO7>

[11] Thị giác máy tính

[https://vi.wikipedia.org/wiki/Thị\\_giác\\_máy\\_tính](https://vi.wikipedia.org/wiki/Thị_giác_máy_tính)

[12] Support Vector Machines Part 1 (of 3): Main Ideas!!!

<https://www.youtube.com/watch?v=efR1C6CvhmE&t=4s>

[13] Realtime Face Emotion Recognition | Python | OpenCV | Step by Step Tutorial for beginners by DeepLearning\_by\_PhDScholar

<https://www.youtube.com/watch?v=fkgpvkqcoJc&t=1654s>

[14] Emotion Detection using CNN | Emotion Detection Deep Learning project |Machine Learning | Data Magic by Data Magic (by Sunny Kusawa)

<https://www.youtube.com/watch?v=UHdRxHPRBng&t=1437s>

[15] Emotion Detection using Convolutional Neural Networks and OpenCV | Keras | Realtime

<https://www.youtube.com/watch?v=Bb4Wvl57Llk&t=1s>