

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA



Báo cáo

HỆ THỐNG THÔNG MINH

Đề tài:

Hệ thống dự đoán tuổi, giới tính và cảm xúc

Giảng viên hướng dẫn:
PGS.TS Quản Thành Thơ

Nhóm 12:
Hoàng Mạnh Thành - 2170084
Trần Phạm Công Danh - 2170523

TP. HỒ CHÍ MINH, tháng 5 năm 2022

MỤC LỤC

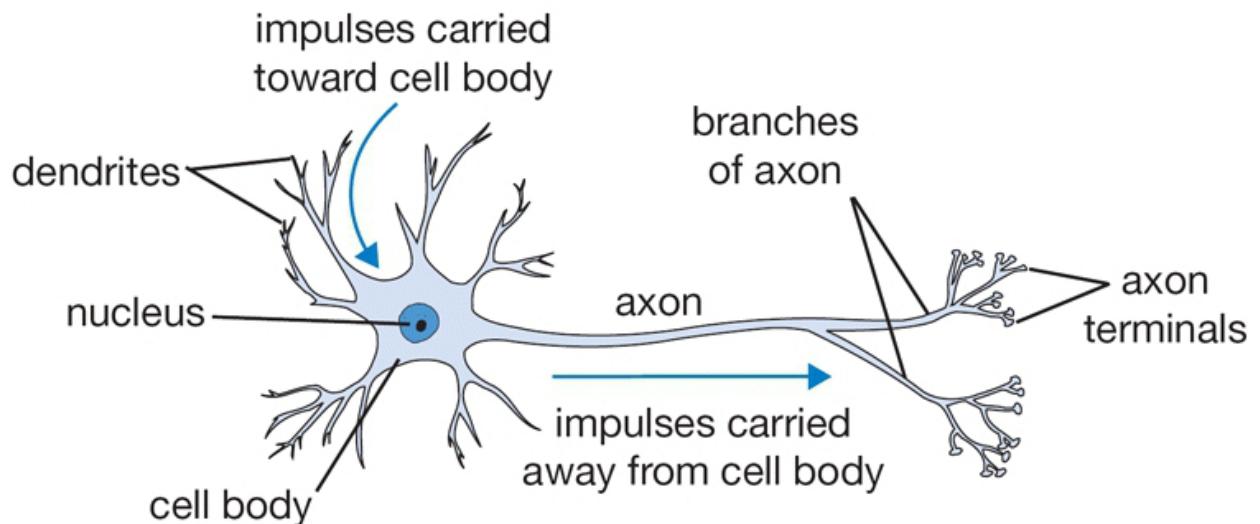
1. Mạng Nơ-ron nhân tạo	4
1.1 Perceptrons	4
1.2 Sigmoid Neurons	5
1.3 Kiến trúc mạng Nơron	7
2. Mạng Nơ-ron tích chập	9
2.1 Giới thiệu	9
2.2 Kiến trúc Mạng Nơ-ron tích chập	9
2.3 Rút trích đặc trưng	11
2.4 Phân lớp	12
2.5 Mô hình Mạng VGG-16 sử dụng trong đề tài	13
3. Đề tài: Hệ thống dự đoán tuổi, giới tính, cảm xúc	16
3.1 Mô tả hệ thống	16
3.2 Giới thiệu Dataset	16
3.3 Pre-processing dữ liệu	18
3.4 Xây dựng mô hình huấn luyện cho các dataset	19
3.4.1 Xây dựng, đánh giá mô hình nhận diện cảm xúc với FER-2013	19
3.4.1.1 Mô hình	19
3.4.1.2 Kết quả đánh giá trên tập train, validation, test.	22
3.4.1.3 Đánh giá confusion matrix, recall, precision, f1-score	23
3.4.1.4 Kết luận	27
3.4.2 Xây dựng, đánh giá mô hình dự đoán giới tính với UTKFace	27
3.4.2.1 Mô hình	27
3.4.2.2 Kết quả đánh giá trên tập train, validation, test.	29
3.4.2.3 Đánh giá confusion matrix, recall, precision, f1-score	30
3.4.2.4 Kết luận	32
3.4.3 Xây dựng, đánh giá mô hình dự đoán Tuổi với UTKFace	33
3.4.3.1 Mô hình	33
3.4.3.2 Kết quả đánh giá trên tập train, validation, test.	35
3.4.3.3 Đánh giá confusion matrix, recall, precision, f1-score	36
3.4.3.4 Kết luận	39
3.5 Xây dựng hệ thống	39
3.5.1 Giao diện người dùng	39
3.5.2 Back-end xử lý dự đoán	40
4. Giới hạn của dự án	40
TÀI LIỆU THAM KHẢO	41

1. Mạng Nơ-ron nhân tạo

Mạng nơ-ron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơ-ron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơ-ron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơ-ron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơ-ron.

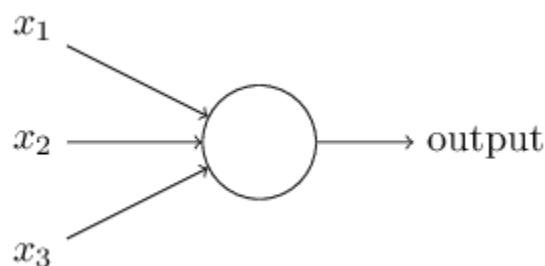
1.1 Perceptrons

Một mạng nơ-ron được cấu thành bởi các nơ-ron đơn lẻ được gọi là các perceptron. Nơ-ron nhân tạo được lấy cảm hứng từ nơ-ron sinh học như hình mô tả bên dưới:



Hình: 1.1.1: Nơ-ron sinh học

Như hình trên, ta có thể thấy một nơ-ron có thể nhận nhiều đầu vào và cho ra một kết quả duy nhất. Mô hình của perceptrons cũng tương tự như vậy:



Hình: 1.1.2: Cách hoạt động của Perceptrons

Một perceptron sẽ nhận một hoặc nhiều đầu vào dạng nhị phân và cho ra một kết quả **độc** dạng nhị phân duy nhất. Các đầu vào được điều phối tầm ảnh hưởng bởi các tham số trọng lượng

tương ứng \mathbf{w} của nó, còn kết quả đầu ra được quyết định dựa vào một ngưỡng quyết định \mathbf{b} nào đó:

$$o = \begin{cases} 0 & \text{if } \sum_i w_i x_i \leq \text{threshold} \\ 1 & \text{if } \sum_i w_i x_i > \text{threshold} \end{cases}$$

Đặt $b = -\text{threshold}$, ta có thể viết lại thành:

$$o = \begin{cases} 0 & \text{if } \sum_i w_i x_i + b \leq 0 \\ 1 & \text{if } \sum_i w_i x_i + b > 0 \end{cases}$$

Ví dụ: việc đưa ra quyết định có đi đến lễ hội âm nhạc hay không dựa trên 3 yếu tố:

1. Thời tiết có tốt không?
2. Có bạn đi cùng không?
3. Địa điểm tổ chức lễ hội có gần không?

Thì ta coi 4 yếu tố đầu vào là x_1, x_2, x_3 , và nếu $o=0$ thì ta không đi đến lễ hội còn $o=1$ thì ta đi. Giả sử mức độ quan trọng của 4 yếu tố trên lần lượt là $w_1 = 0.5, w_2 = 0.25, w_3 = 0.25$ và chọn ngưỡng $b = -0.5$ thì ta có thể thấy rằng việc trời nắng có ảnh hưởng tới 50% quyết định đi lễ hội và việc có bạn đi cùng ảnh hưởng tới 25% quyết định đi nhậu của ta. Nếu gắn $x_0 = 1$ và $w_0 = b$, ta còn có thể viết gọn lại thành:

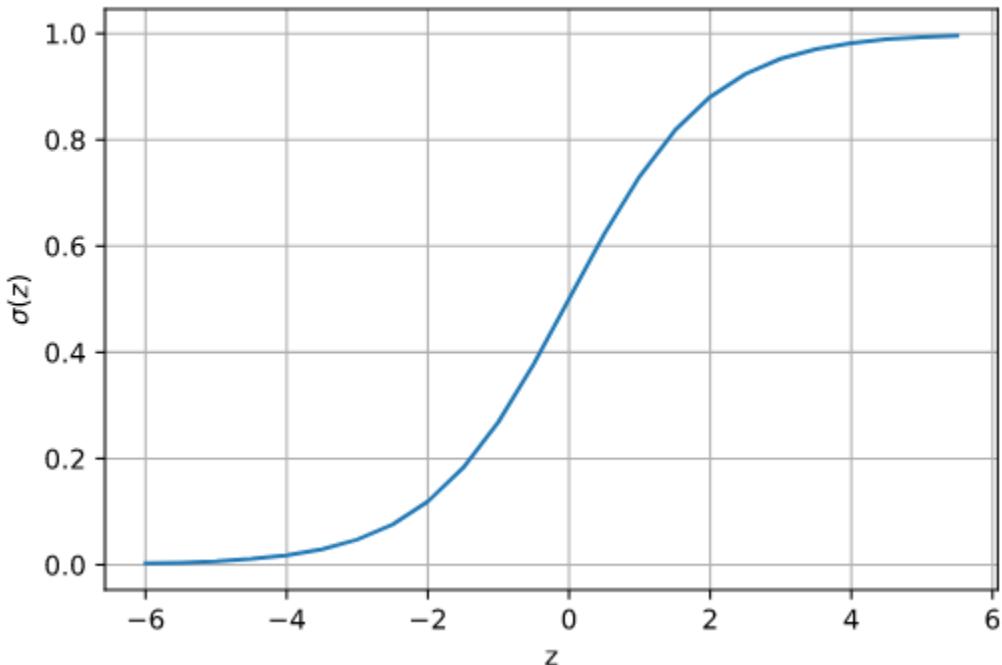
$$o = \begin{cases} 0 & \text{if } \mathbf{w}^\top \mathbf{x} \leq 0 \\ 1 & \text{if } \mathbf{w}^\top \mathbf{x} > 0 \end{cases}$$

1.2 Sigmoid Neurons

Với đầu vào và đầu ra dạng nhị phân, ta rất khó có thể điều chỉnh một lượng nhỏ đầu vào để đầu ra thay đổi chút ít, nên để linh động, ta có thể mở rộng chúng ra cả khoảng $[0, 1]$. Lúc này đầu ra được quyết định bởi một hàm sigmoid $\sigma(\mathbf{w}^\top \mathbf{x})$ có công thức như sau:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Đồ thị của hàm này cũng cân xứng rất đẹp thể hiện được mức độ công bằng của các tham số:



Hình: 1.2.1: Sigmoid Function

Đặt $z = \mathbf{w}^T \mathbf{x}$ thì công thức của perceptron lúc này sẽ có dạng:

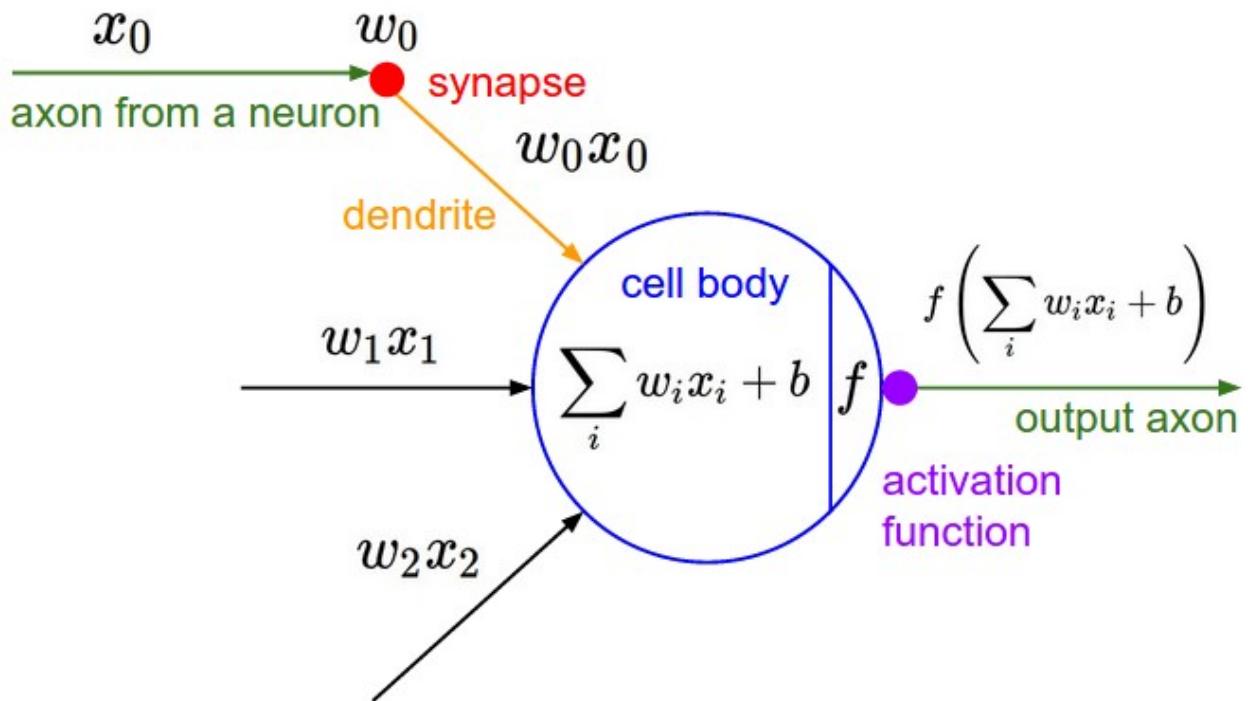
$$o = \sigma(z) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$

Tới đây thì ta có thể thấy rằng mỗi sigmoid neuron cũng tương tự như một bộ phân loại tuyến tính (logistic regression) bởi xác suất $P(y_i = 1 | \mathbf{x}_i; \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x})$.

Ngoài hàm sigmoid ra, ta còn có thể một số hàm khác như tanh, ReLU để thay thế hàm sigmoid bởi dạng đồ thị của nó cũng tương tự như sigmoid. Một cách tổng quát, hàm perceptron được biểu diễn qua một hàm kích hoạt (activation function) $f(z)$ như sau:

$$o = f(z) = f(\mathbf{w}^T \mathbf{x})$$

Bằng cách biểu diễn như vậy, ta có thể coi neuron sinh học được thể hiện như sau:

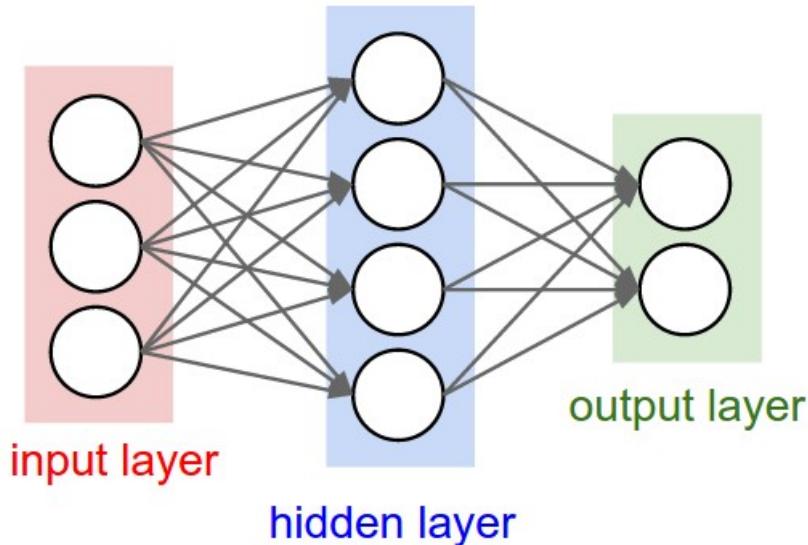


Hình: 1.2.2: Mô hình Noron

Một điểm cần lưu ý là các hàm kích hoạt buộc phải là **hàm phi tuyến**. Vì nếu nó là tuyến tính thì khi kết hợp với phép toán tuyến tính $w^T x$ thì kết quả thu được cũng sẽ là một thao tác tuyến tính dẫn tới chuyện nó trở nên vô nghĩa.

1.3 Kiến trúc mạng Noron

Mạng NN là sự kết hợp của của các tầng perceptron hay còn được gọi là perceptron đa tầng (multilayer perceptron) như hình vẽ bên dưới:

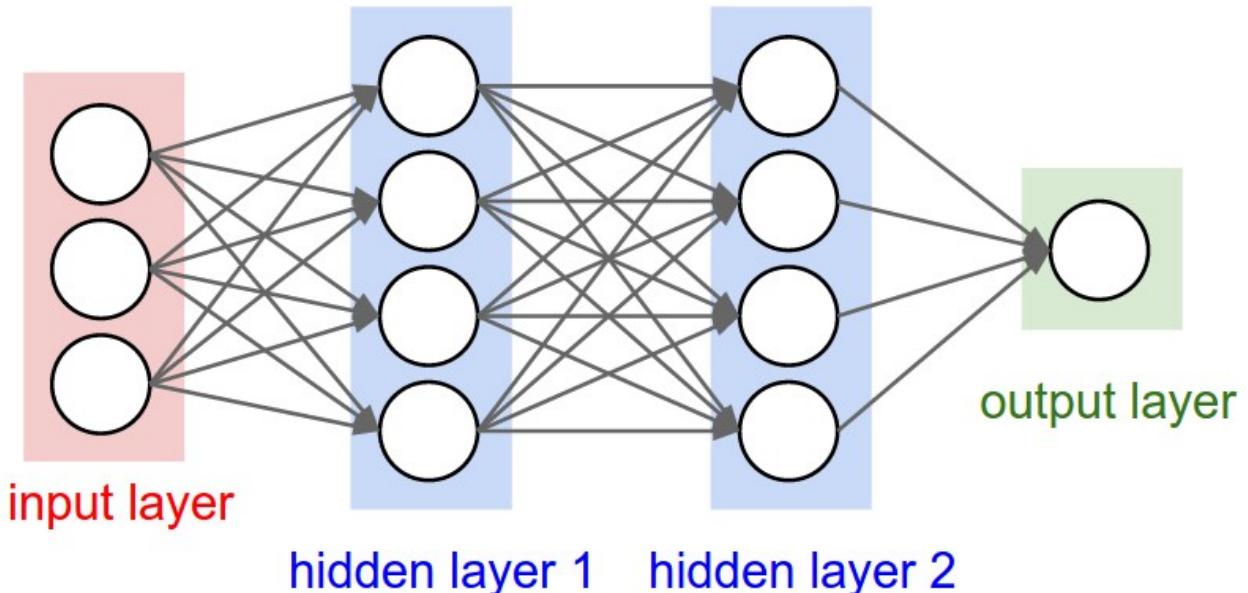


Hình: 1.3.1: Mạng Nơron

Một mạng NN sẽ có 3 kiểu tầng:

- *Tầng vào (input layer)*: Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
- *Tầng ra (output layer)*: Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
- *Tầng ẩn (hidden layer)*: Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

Một NN chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn như hình dưới:



Hình: 1.3.2: Mạng Nơron 2 tầng ẩn

Trong mạng NN, mỗi nút mạng là một sigmoid nơ-ron nhưng hàm kích hoạt của chúng có thể khác nhau. Tuy nhiên trong thực tế người ta thường để chúng cùng dạng với nhau để tính toán cho thuận lợi.

Ở mỗi tầng, số lượng các nút mạng (nơ-ron) có thể khác nhau tùy thuộc vào bài toán và cách giải quyết. Nhưng thường khi làm việc người ta để các tầng ẩn có số lượng nơ-ron bằng nhau. Ngoài ra, các nơ-ron ở các tầng thường được liên kết đôi một với nhau tạo thành mạng kết nối đầy đủ (full-connected network). Khi đó ta có thể tính được kích cỡ của mạng dựa vào số tầng và số nơ-ron. Ví dụ ở hình trên ta có:

- 4 tầng mạng, trong đó có 2 tầng ẩn.
- $3 + 4*2 + 1 = 12$ nút mạng.
- $(3*4 + 4*4 + 4*1) + (4+4+1) = 41$ tham số.

2. Mạng Nơ-ron tích chập

2.1 Giới thiệu

Để dạy thuật toán nhận diện đối tượng trong hình ảnh, ta sử dụng một loại Mạng Nơ-ron Nhân Tạo (Artificial Neural Network): Mạng Nơ-ron Tích Chập. Tên của nó được dựa trên phép tính quan trọng được sử dụng trong mạng đó là Tích Chập.

Mạng Nơ-ron Tích Chập lấy cảm hứng từ não người. Nghiên cứu trong những thập niên 1950 và 1960 của D.H Hubel và T.N Wiesel trên não của động vật đã đề xuất một mô hình mới cho việc cách mà động vật nhìn nhận thế giới. Trong báo cáo, hai ông đã diễn tả 2 loại tế bào nơ-ron trong não và cách hoạt động khác nhau: tế bào đơn giản (simple cell – S cell) và tế bào phức tạp (complex cell – C cell).

Các tế bào đơn giản được kích hoạt khi nhận diện các hình dáng đơn giản như đường nằm trong một khu vực cố định và một góc cạnh của nó. Các tế bào phức tạp có vùng tiếp nhận lớn hơn và đầu ra của nó không nhạy cảm với những vị trí cố định trong vùng.

Trong thị giác, vùng tiếp nhận của một nơ-ron tương ứng với một vùng trên võng mạc nơi mà sẽ kích hoạt nơ-ron tương ứng.

Năm 1980, Fukushima đề xuất mô hình mạng nơ-ron có cấp bậc gọi là neocognitron. Mô hình này dựa trên khái niệm về S cell và C cell. Mạng neocognitron có thể nhận diện mẫu dựa trên việc học hình dáng của đối tượng.

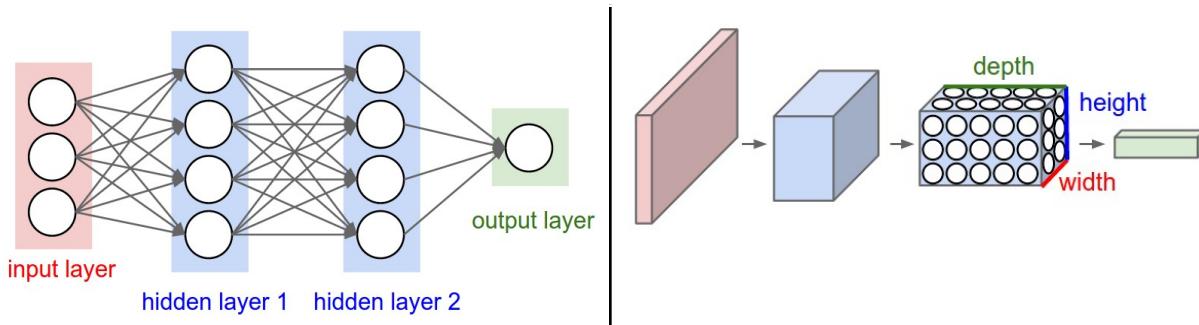
Sau đó vào năm 1998, Mạng Nơ-ron Tích Chập được giới thiệu bởi Bengio, Le Cun, Bottou và Haffner. Mô hình đầu tiên của họ được gọi tên là LeNet-5. Mô hình này có thể nhận diện chữ số viết tay.

2.2 Kiến trúc Mạng Nơ-ron tích chập

Mạng Nơ-ron Tích Chập có kiến trúc khác với Mạng Nơ-ron thông thường. Mạng Nơ-ron bình thường chuyển đổi đầu vào thông qua hàng loạt các tầng ẩn. Mỗi tầng là một tập các nơ-ron

và các tầng được liên kết đầy đủ với các nơ-ron ở tầng trước đó. Và ở tầng cuối cùng sẽ là tầng kết quả đại diện cho dự đoán của mạng.

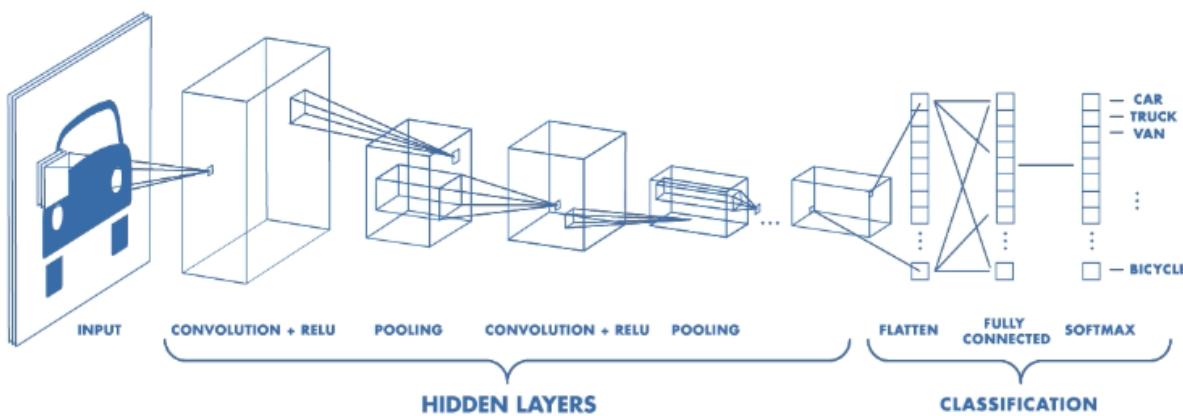
Đầu tiên, mạng Nơ-ron Tích Chập được chia thành 3 chiều: rộng, cao, và sâu. Kế đến, các nơ-ron trong mạng không liên kết hoàn toàn với toàn bộ nơ-ron kế đến nhưng chỉ liên kết tới một vùng nhỏ. Cuối cùng, một tầng đầu ra được tối giản thành véc-tơ của giá trị xác suất.



Hình 2.2.1: Minh họa Mạng Nơ-ron thông thường và Mạng Nơ-ron tích chập

CNNs gồm hai thành phần:

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



Hình 2.2.2: Kiến trúc của CNN

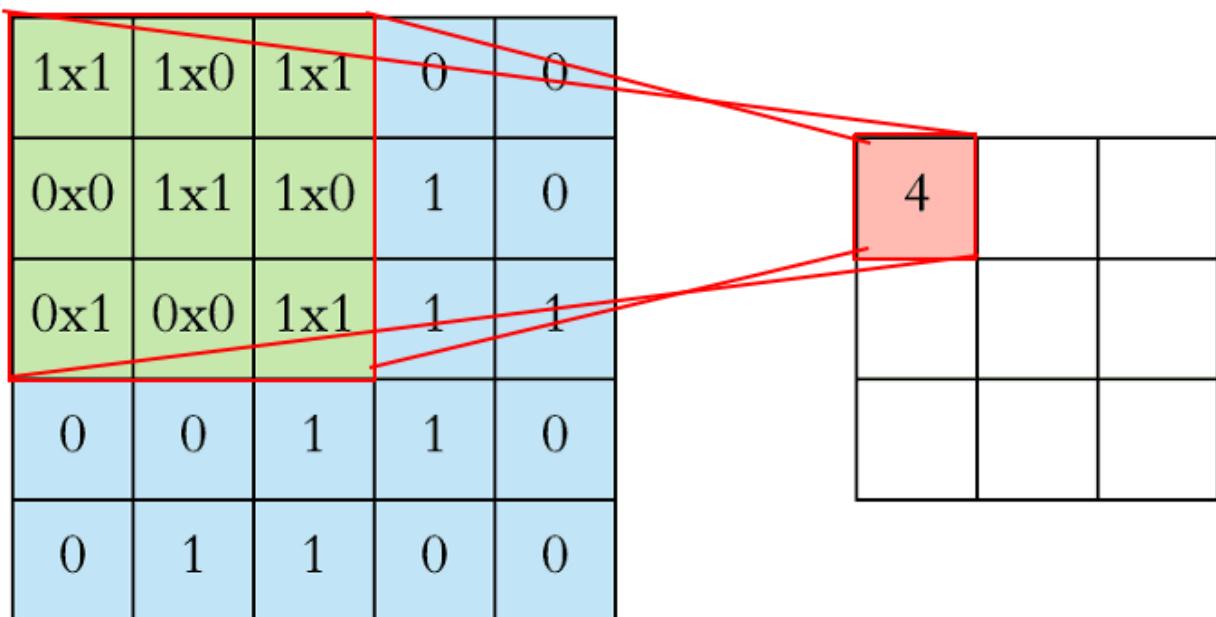
2.3 Rút trích đặc trưng

Tích chập là một khối quan trọng trong CNN. Thuật ngữ tích chập được dựa trên một phép hợp nhất toán học của hai hàm tạo thành hàm thứ ba. Phép toán này kết hợp hai tập thông tin khác nhau.

Trong trường hợp CNN, tích chập được thực hiện trên giá trị đầu vào của dữ liệu và **kernel/filter** (thuật ngữ này được sử dụng khác nhau tùy tình huống) để tạo ra một **bản đồ đặc trưng** (feature map).

Ta thực hiện phép tích chập bằng cách trượt kernel/filter theo dữ liệu đầu vào. Tại mỗi vị trí, ta tiến hành phép nhân ma trận và tính tổng các giá trị để đưa vào bản đồ đặc trưng.

Trong hình dưới đây, thành phần kernel/filter (màu xanh lá) trượt trên đầu vào (màu xanh dương) và kết quả được trả về bản đồ đặc trưng (màu đỏ). Kernel/filter có kích thước là 3×3 trong ví dụ này.



Hình 2.3.1: Phép tích chập

Trong thực tế, tích chập được thực hiện hiện trên không gian 3 chiều. Vì mỗi hình ảnh được biểu diễn dưới dạng 3 chiều: rộng, cao, và sâu. Chiều sâu ở đây chính là giá trị màu sắc của hình (RGB).

Ta thực hiện phép tích chập trên đầu vào nhiều lần khác nhau. Mỗi lần sử dụng một kernel/filter khác nhau. Kết quả ta sẽ thu được những bản đồ đặc trưng khác nhau. Cuối cùng, ta kết hợp toàn bộ bản đồ đặc trưng này thành kết quả cuối cùng của tầng tích chập.

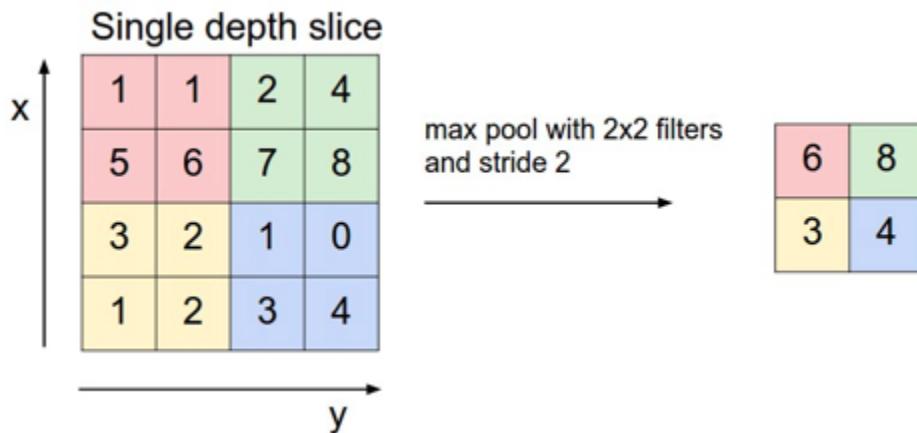
Tương tự như mạng nơ-ron thông thường, ta sử dụng một hàm kích hoạt (activate function) để có đầu ra dưới dạng phi tuyến. Trong trường hợp CNN, đầu ra của phép tích chập sẽ đi qua hàm kích hoạt nào đó ví dụ như hàm **ReLU** (rectified linear units).

Trong quá trình trượt kernel/filter trên dữ liệu đầu vào, ta sẽ quy định một bước nhảy (stride) với mỗi lần di chuyển. Thông thường ta lựa chọn thường chọn bước nhảy là 1. Nếu kích thước bước nhảy tăng, kernel/filter sẽ có ít ô trùng lắp.

Bởi vì kích thước đầu ra luôn nhỏ hơn đầu vào nên ta cần một phép xử lý đầu vào để đầu ra không bị co giãn. Đơn giản ta chỉ cần thêm một lè nhỏ vào đầu vào. Một lè với giá trị 0 sẽ được thêm vào xung quanh đầu vào trước khi thực hiện phép tích chập.

Thông thường, sau mỗi tầng tích chập, ta sẽ cho kết quả đi qua một **tầng hợp nhất** (pooling layer). Mục đích của tầng này là để nhanh chóng giảm số chiều. Việc này giúp giảm thời gian học và hạn chế việc **overfitting**.

Một phép hợp nhất đơn giản thường được dùng đó là **max pooling**, phép này lấy giá trị lớn nhất của một vùng để đại diện cho vùng đó. Kích thước của vùng sẽ được xác định trước để giảm kích thước của bản đồ đặc trưng nhanh chóng nhưng vẫn giữ được thông tin cần thiết.



Hình 2.3.1: Max pooling kích thước 2x2

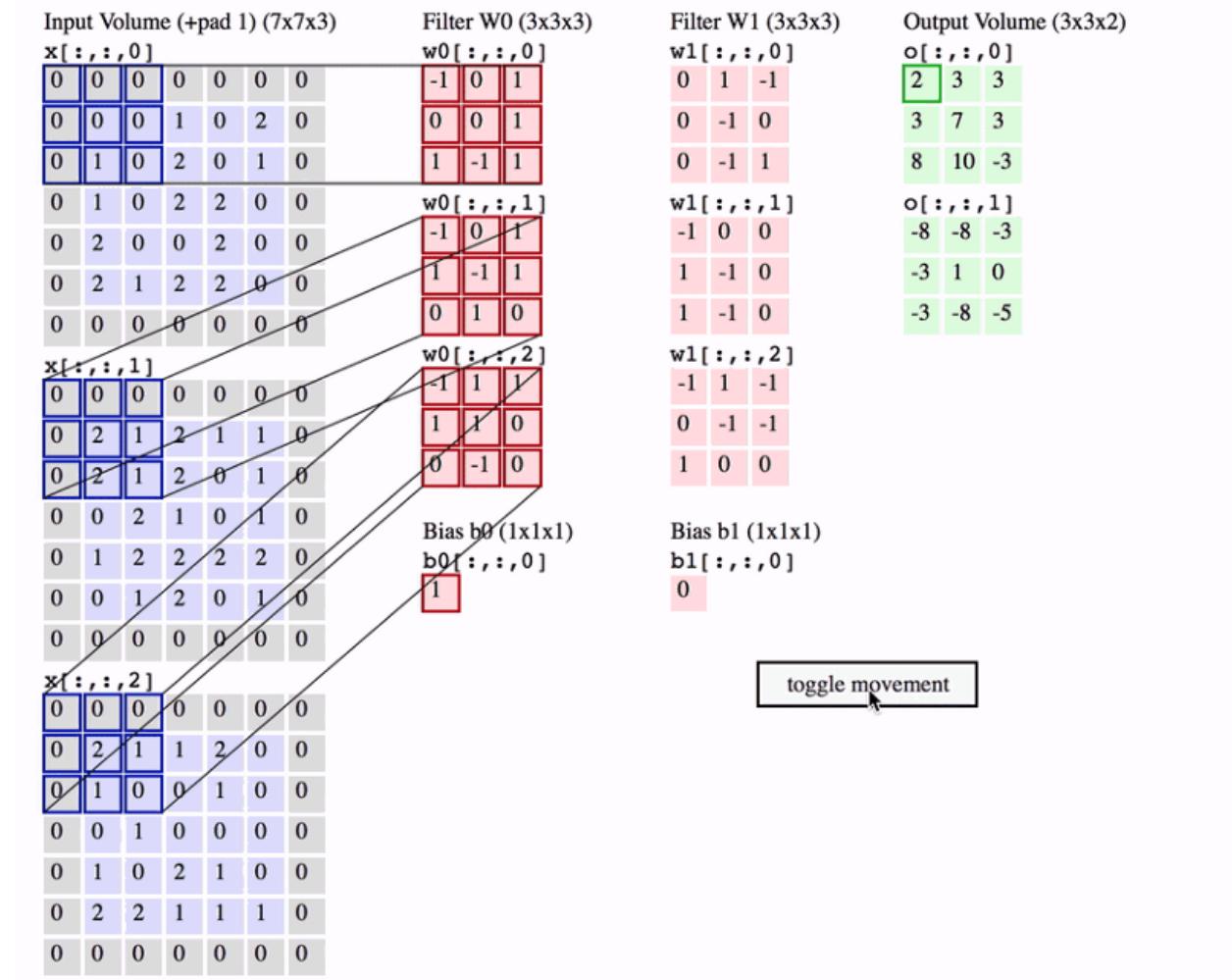
Tổng kết lại khi sử dụng CNN, ta cần chú ý đến 4 siêu tham số quan trọng:

- Kích thước kernel/filter
- Số lượng kernel/filter
- Kích thước bước nhảy (stride)
- Kích thước lè (padding)

2.4 Phân lớp

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

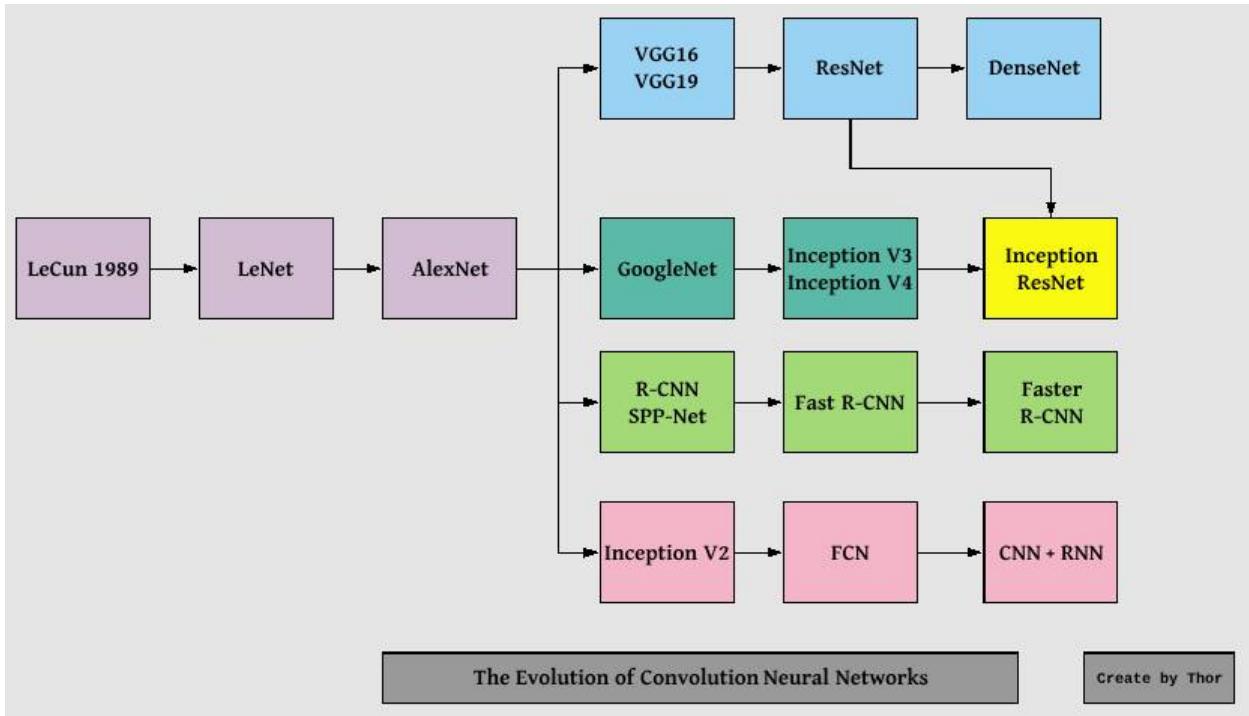
Kết quả thu được cuối cùng cũng sẽ là một véc-tơ với các giá trị xác suất cho việc dự đoán như mạng nơ-ron thông thường.



Hình 2.4.1: Minh họa cho tích chập với 2 filter, 3 cột, 2 bước nhảy, $lè = 1$

2.5 Mô hình Mạng VGG-16 sử dụng để tài

Từ mạng CNN cơ bản người ta có thể tạo ra rất nhiều architect khác nhau, từ những mạng neural cơ bản 1 đến 2 layer đến 100 layer. Các mạng CNN tiêu biểu như: LeNet-5 (1988), AlexNet (2012), VGG-16 (2014), GoogleNet-V3 (2015), ResNet-50 (2015), DenseNet (2016).



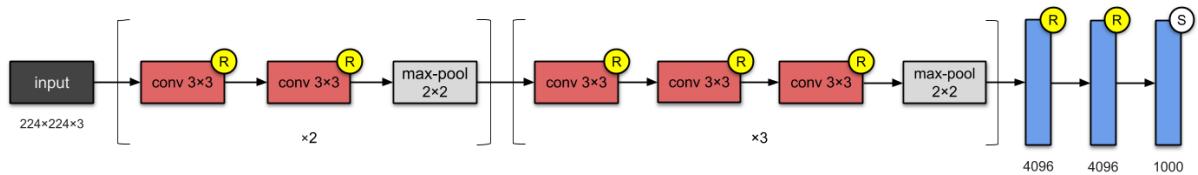
Hình 2.5.1: Các mô hình mạng CNN

Với VGG-16, quan điểm về một mạng nơ ron sâu hơn sẽ giúp ích cho cải thiện độ chính xác của mô hình tốt hơn. Về kiến trúc thì VGG-16 vẫn dũ các đặc điểm của AlexNet nhưng có những cải tiến:

- Kiến trúc VGG-16 sâu hơn, bao gồm 13 layers tích chập 2 chiều (thay vì 5 so với AlexNet) và 3 lớp kết nối đầy đủ.
- Lần đầu tiên trong VGG-16 chúng ta xuất hiện khái niệm về khối tích chập (block). Đây là những kiến trúc gồm một tập hợp các layers CNN được lặp lại giống nhau. Kiến trúc khối đã khởi nguồn cho một dạng kiến trúc hình mẫu rất thường gặp ở các mạng CNN kể từ đó.
- VGG-16 cũng kế thừa lại hàm activation ReLU ở AlexNet.
- VGG-16 cũng là kiến trúc đầu tiên thay đổi thứ tự của các block khi xếp nhiều layers CNN + max pooling thay vì xen kẽ chỉ một layer CNN + max pooling.
- VGG-16 chỉ sử dụng các bộ lọc kích thước nhỏ 3x3 thay vì nhiều kích thước bộ lọc như AlexNet. Kích thước bộ lọc nhỏ sẽ giúp giảm số lượng tham số cho mô hình và mang lại hiệu quả tính toán hơn.

Mạng VGG-16 sâu hơn so với AlexNet và số lượng tham số của nó lên tới 138 triệu tham số. Đây là một trong những mạng mà có số lượng tham số lớn nhất. Kết quả của nó hiện đang xếp thứ 2 trên bộ dữ liệu ImageNet validation ở thời điểm public. Ngoài ra còn một phiên bản nữa của VGG-16 là VGG-19 tăng cường thêm 3 layers về độ sâu.

Bắt đầu từ VGG-16, một hình mẫu chung cho các mạng CNN trong các tác vụ học có giám sát trong xử lý ảnh đã bắt đầu hình thành đó là các mạng trở nên sâu hơn và sử dụng các block dạng *[Conv2D*n + Max Pooling]*.



Hình 2.5.2 Kiến trúc mạng VGG-16

3. Đề tài: Hệ thống dự đoán tuổi, giới tính, cảm xúc

3.1 Mô tả hệ thống

Người dùng tải video clip hoặc ảnh từ máy lên hệ thống hoặc sử dụng hình ảnh từ Webcam. Hệ thống sẽ nhận diện khuôn mặt người trong video/webcam và phán đoán khoảng tuổi, giới tính, cảm xúc.

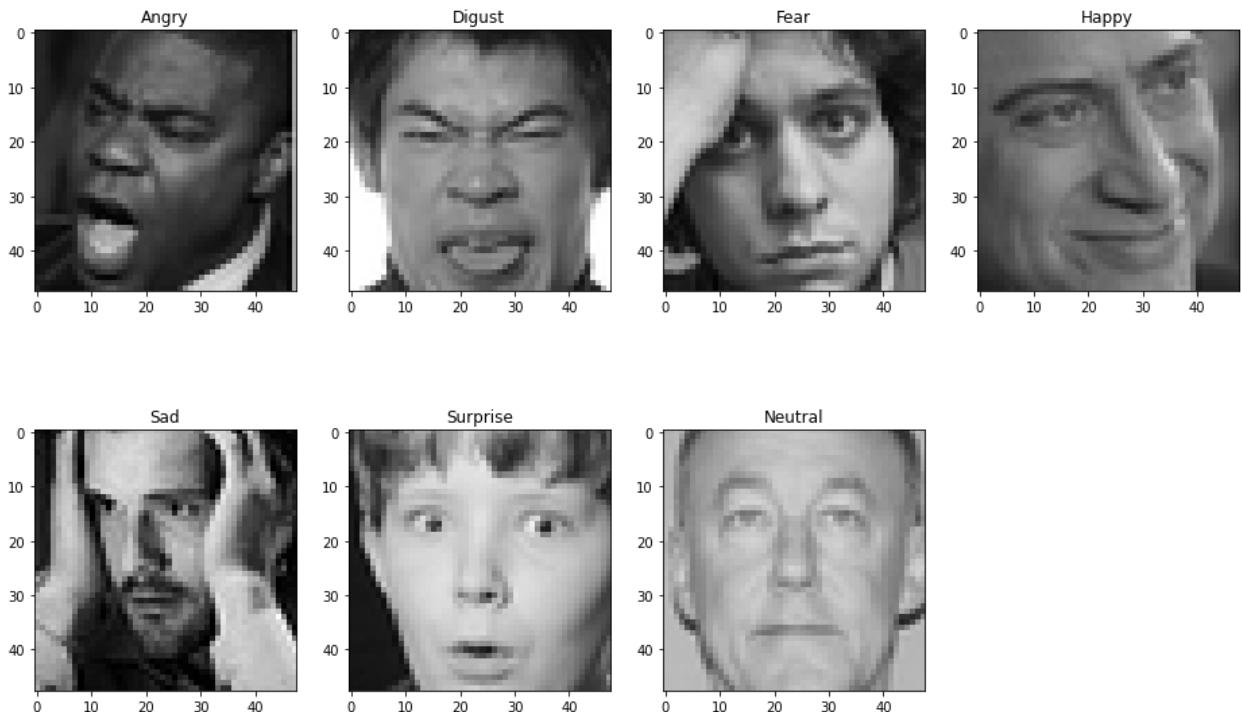
Hệ thống sẽ có hai ứng dụng chính:

1. Sử dụng tích hợp trong camera ở quầy thanh toán nhà hàng để đánh giá độ hài lòng của khách hàng. Do hầu hết các đánh giá bằng câu hỏi, có thể khách hàng sẽ không trả lời thật lòng.
2. Phát triển mở rộng: dữ liệu dự đoán được lưu trữ vào Data Warehouse, làm giàu và sử dụng để phục vụ cho phân tích sở thích của khách hàng cho các nhà hàng, giúp các nhà hàng hiểu rõ khách hàng hơn.

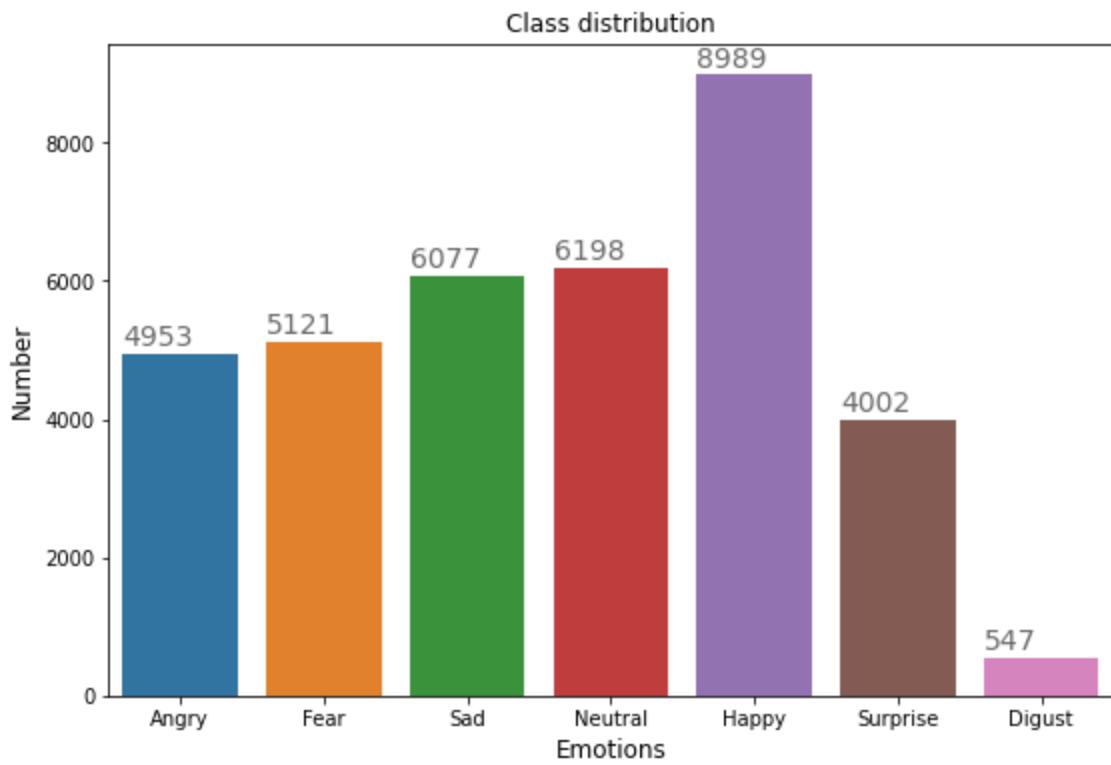
Kỹ thuật dự đoán sử dụng: mạng CNN. Hình trong training set sẽ được qua các mạng tích chập để trích xuất những đặc trưng, rồi qua các lớp fully connected rồi cho ra kết quả.

3.2 Giới thiệu Dataset

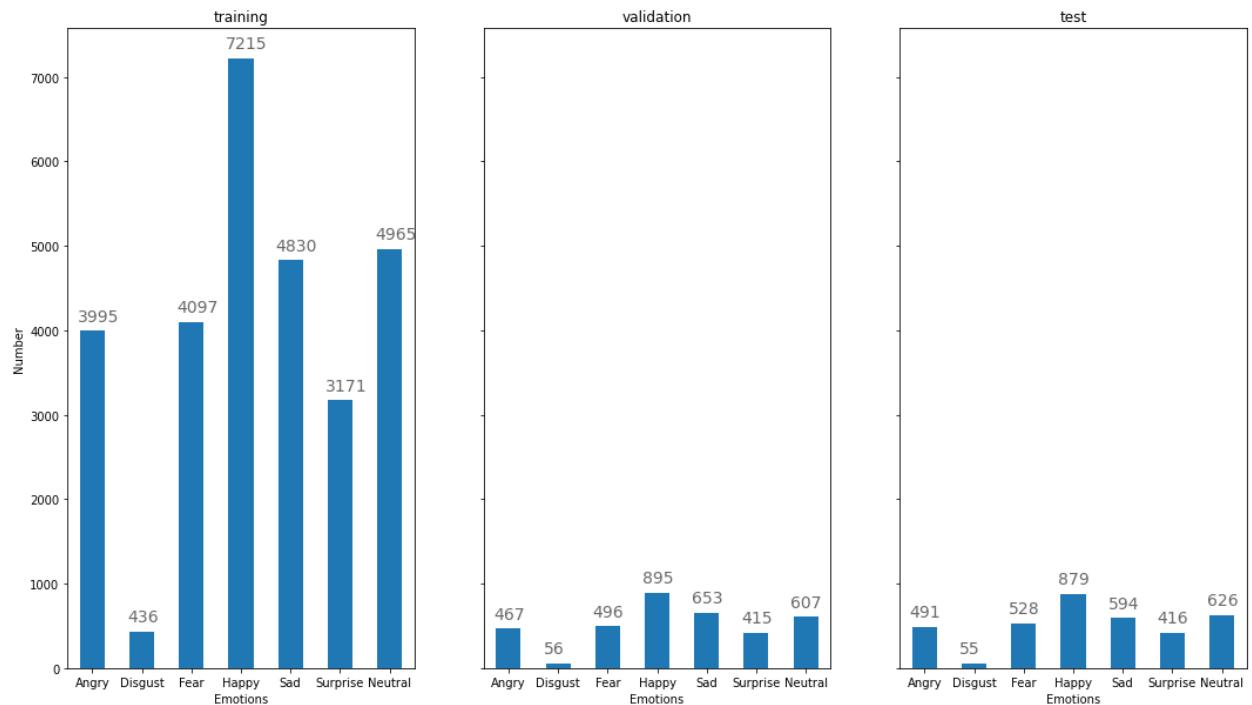
FER-2013[4]: Đây là dataset có 28709 ảnh thuộc training set, 3589 thuộc public test set(validation set) và 3589 thuộc private test set (test set), được phân lớp thành 7 cảm xúc là (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).



Hình 3.2.1: Mẫu các cảm xúc trong dataset FER-2013

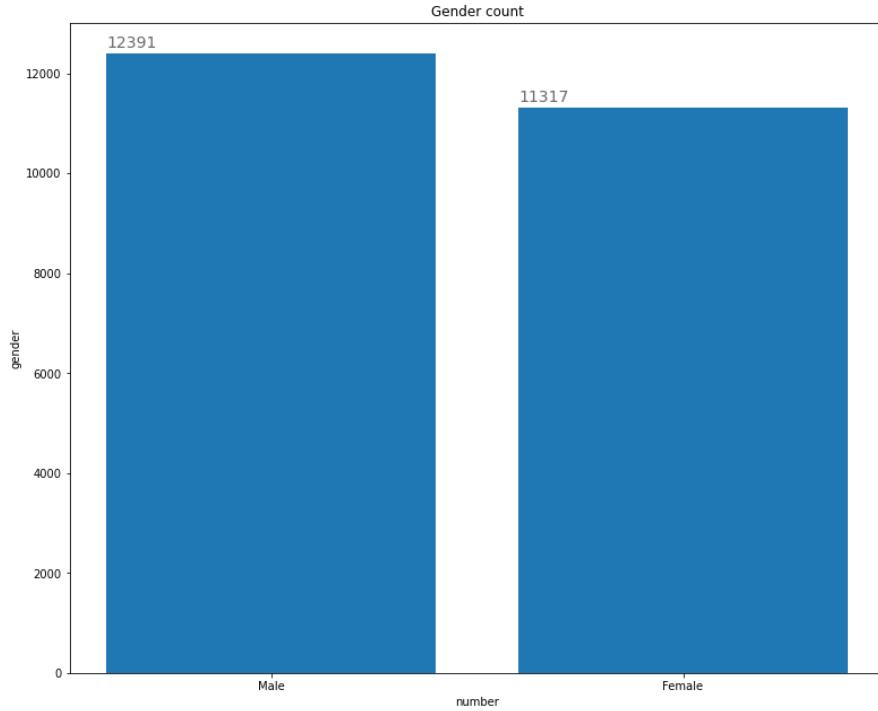


Hình 3.2.2 Biểu đồ số mẫu các cảm xúc trong FER-2013 dataset

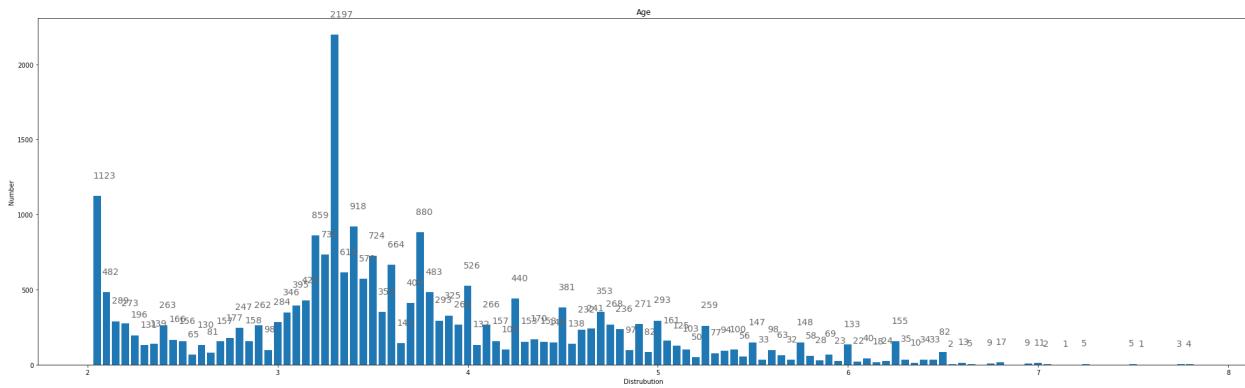


Hình 3.2.3: Biểu đồ số mẫu các cảm xúc trong training, validation và test set

UTK Face[5]: Đây là dataset có đến 23708 nghìn bức ảnh với kích thước, được phân lớp về giới tính, chủng tộc và tuổi từ 1 đến 116.



Hình 3.2.4: Tần số Nam và nữ trong UTKFace dataset



Hình 3.2.5: Tần số từng tuổi trong tập UTKFace

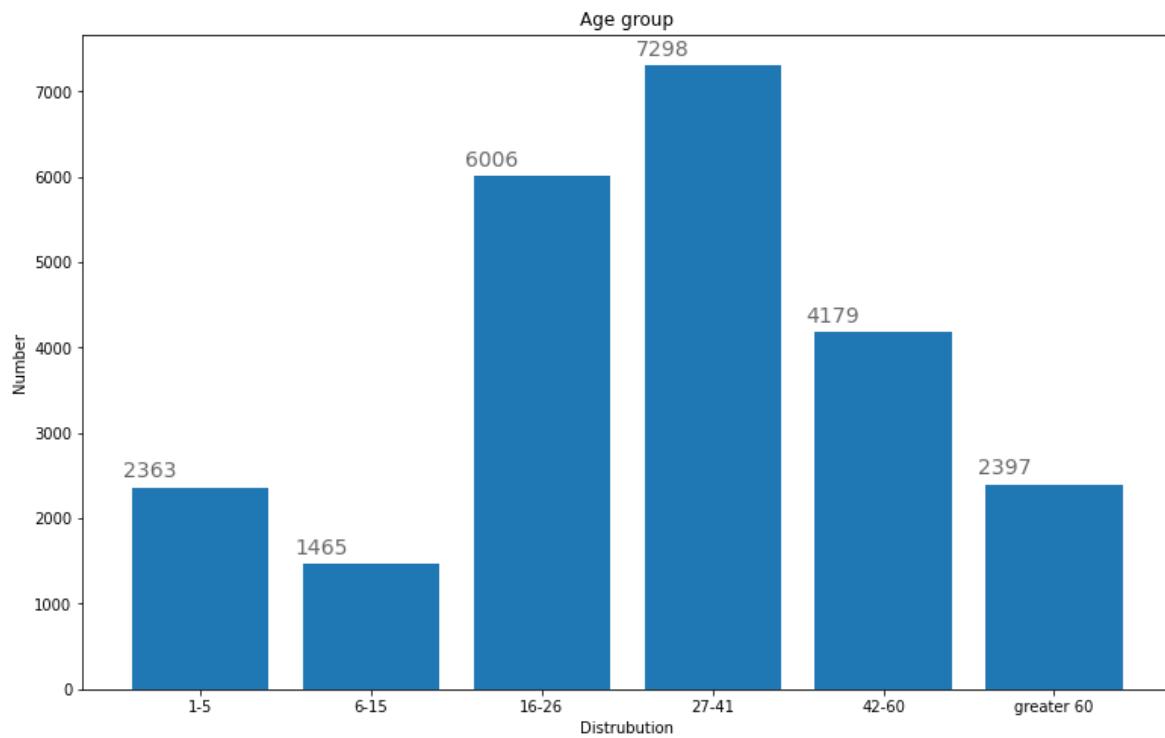
3.3 Pre-processing dữ liệu

- Loại bỏ những ảnh không có mặt người trong tập dataset FER-2013. Vì dataset đã được chia sẵn thành training, public test, private test với tỷ lệ 80%, 10%, 10%. Nên dataset sẽ

được chia thành 3 tập với training, validation, test set với tỷ lệ 80%, 10%, 10% với training, validation, test lần lượt là training, public test, private test.

- Với dataset UTKFace:

- Giới tính chia thành 2 nhóm: nam và nữ, với tỷ lệ các training, validation, test lần lượt là: 64%, 16%, 20%.
- Nhóm tuổi được chia thành training, validation, test với tỷ lệ lần lượt là: 64%, 16%, 20%. Tuổi trong tập dataset gốc được gom thành 6 nhóm:
 - + Nhóm 1: từ 1 đến 5 tuổi.
 - + Nhóm 2: từ 6 đến 15 tuổi.
 - + Nhóm 3: từ 16 đến 26 tuổi.
 - + Nhóm 4: từ 27 đến 41 tuổi.
 - + Nhóm 5: từ 42 đến 60 tuổi.
 - + Nhóm 6: lớn hơn 60 tuổi.



Hình 3.3.1: Phân bố các nhóm tuổi

3.4 Xây dựng mô hình huấn luyện cho các dataset

3.4.1 Xây dựng, đánh giá mô hình nhận diện cảm xúc với FER-2013

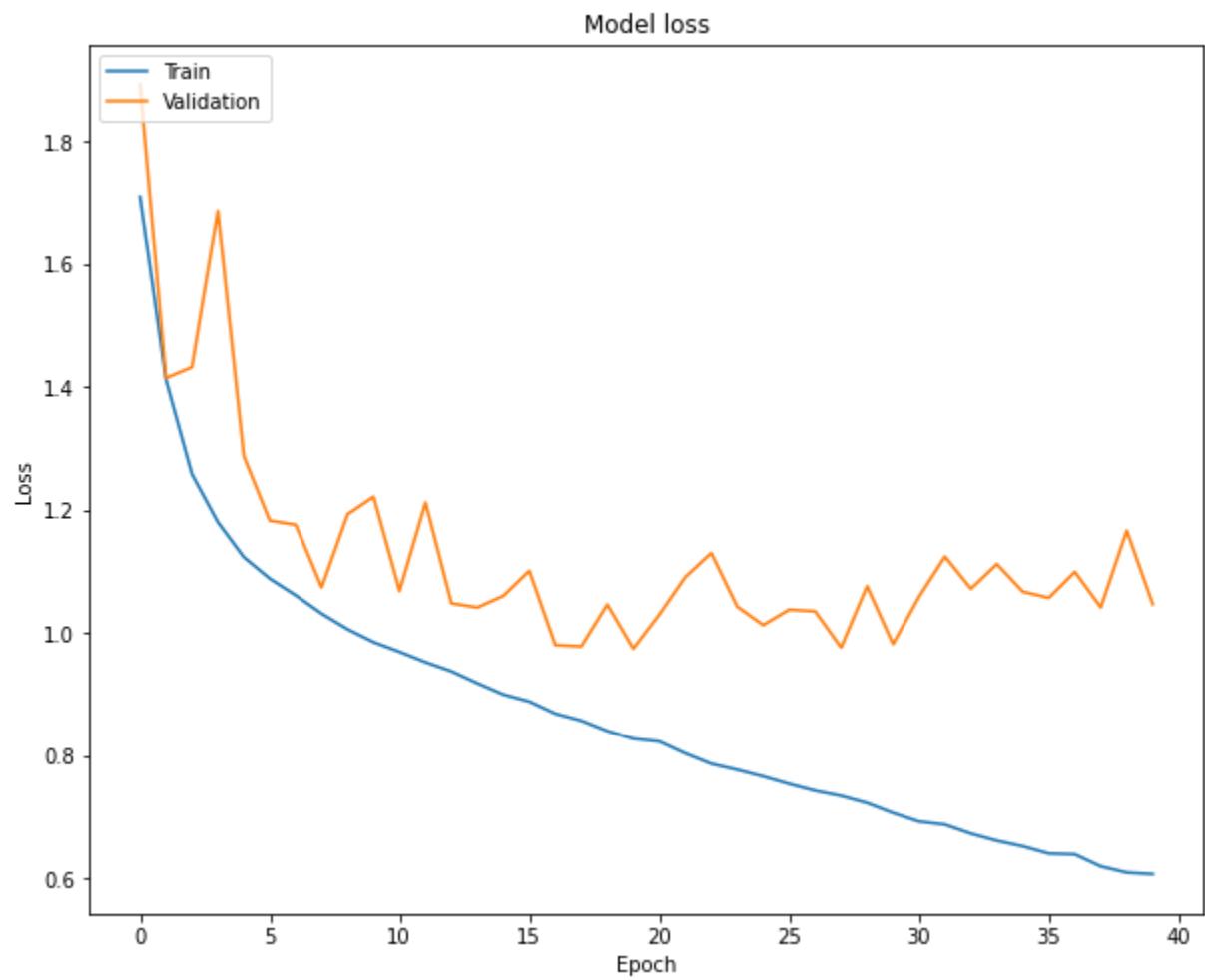
3.4.1.1 Mô hình

Model: "sequential_1"

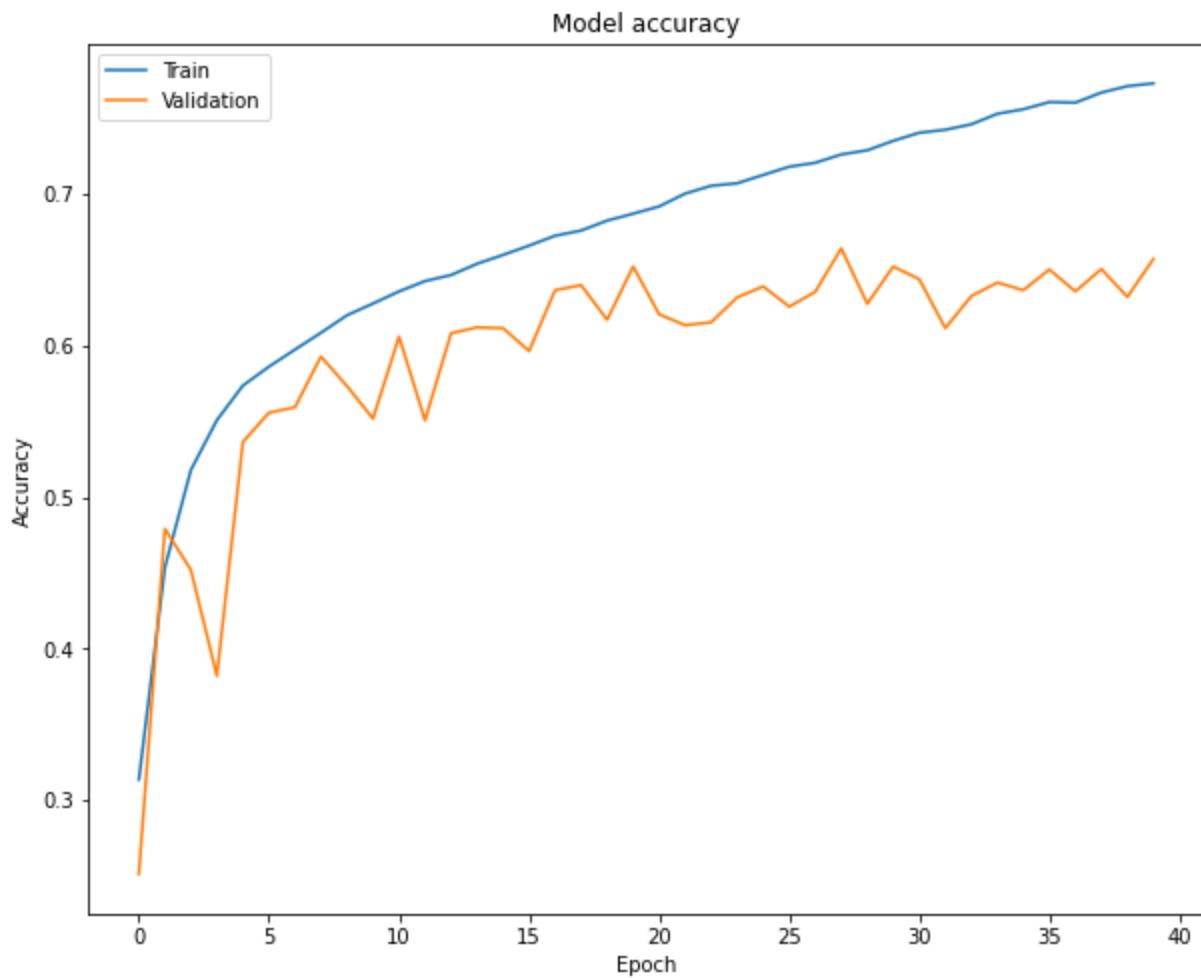
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 256)	2560
batch_normalization (BatchN ormalization)	(None, 46, 46, 256)	1024
activation (Activation)	(None, 46, 46, 256)	0
conv2d_1 (Conv2D)	(None, 46, 46, 256)	590080
batch_normalization_1 (Bathc hNormalization)	(None, 46, 46, 256)	1024
activation_1 (Activation)	(None, 46, 46, 256)	0
max_pooling2d (MaxPooling2D)	(None, 23, 23, 256)	0
conv2d_2 (Conv2D)	(None, 23, 23, 128)	295040
batch_normalization_2 (Bathc hNormalization)	(None, 23, 23, 128)	512
activation_2 (Activation)	(None, 23, 23, 128)	0
conv2d_3 (Conv2D)	(None, 23, 23, 128)	147584
batch_normalization_3 (Bathc hNormalization)	(None, 23, 23, 128)	512
activation_3 (Activation)	(None, 23, 23, 128)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 11, 11, 128)	0
conv2d_4 (Conv2D)	(None, 11, 11, 64)	73792
batch_normalization_4 (Bathc hNormalization)	(None, 11, 11, 64)	256
activation_4 (Activation)	(None, 11, 11, 64)	0
conv2d_5 (Conv2D)	(None, 11, 11, 64)	36928

batch_normalization_5 (Batch Normalization)	(None, 11, 11, 64)	256
activation_5 (Activation)	(None, 11, 11, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 512)	819712
batch_normalization_6 (Batch Normalization)	(None, 512)	2048
activation_6 (Activation)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
batch_normalization_7 (Batch Normalization)	(None, 256)	1024
activation_7 (Activation)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
batch_normalization_8 (Batch Normalization)	(None, 128)	512
activation_8 (Activation)	(None, 128)	0
dense_3 (Dense)	(None, 7)	903
=====		
Total params:	2,137,991	
Trainable params:	2,134,407	
Non-trainable params:	3,584	

3.4.1.2 Kết quả đánh giá trên tập train, validation, test.



Hình 3.4.1.2.1: Loss trên training set và validation set

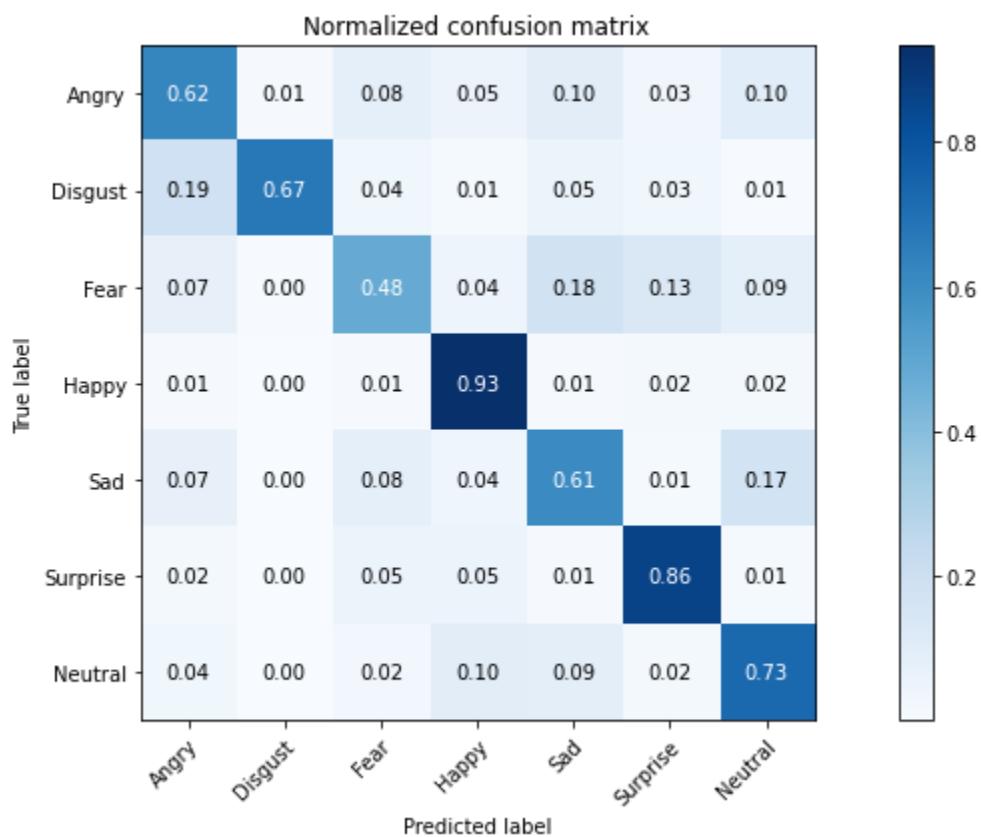


Hình 3.4.1.2.2: Độ chính xác trên training set và validation set

3.4.1.3 Đánh giá confusion matrix, recall, precision, f1-score

Classification Report					
	precision	recall	f1-score	support	
Angry	0.70	0.62	0.66	3995	
Disgust	0.78	0.67	0.72	436	
Fear	0.64	0.48	0.55	4097	
Happy	0.84	0.93	0.89	7215	
Sad	0.63	0.61	0.62	4830	
Surprise	0.74	0.86	0.79	3171	
Neutral	0.67	0.73	0.70	4965	
accuracy				0.72	28709
macro avg		0.72	0.70	0.70	28709
weighted avg		0.72	0.72	0.72	28709

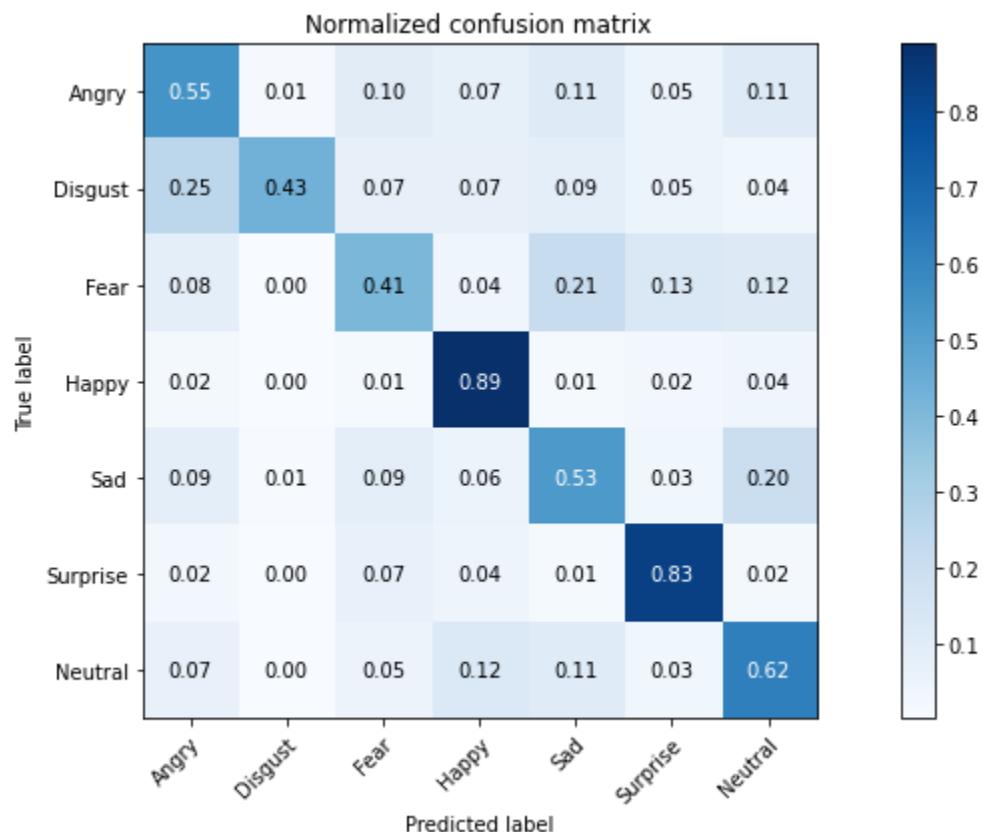
Hình 3.4.1.3.1: Recall, precision, f1-score của training set



Hình 3.4.1.3.2: Confusion matrix training set

Classification Report					
	precision	recall	f1-score	support	
Angry	0.58	0.55	0.57	467	
Disgust	0.63	0.43	0.51	56	
Fear	0.53	0.41	0.46	496	
Happy	0.81	0.89	0.85	895	
Sad	0.58	0.53	0.55	653	
Surprise	0.69	0.83	0.75	415	
Neutral	0.57	0.62	0.59	607	
accuracy			0.65	3589	
macro avg	0.63	0.61	0.61	3589	
weighted avg	0.64	0.65	0.64	3589	

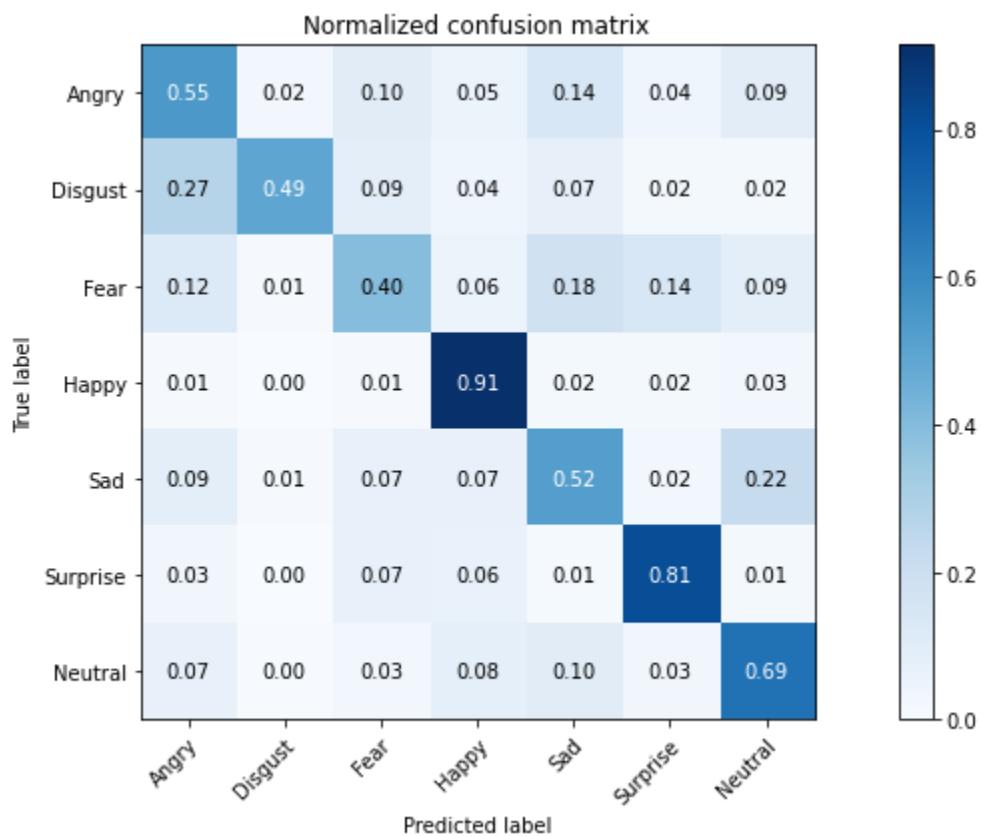
Hình 3.4.1.3.3: Recall, precision, f1-score của validation set



Hình 3.4.1.3.4: Confusion matrix Validation set

Classification Report					
	precision	recall	f1-score	support	
Angry	0.57	0.55	0.56	491	
Disgust	0.52	0.49	0.50	55	
Fear	0.58	0.40	0.47	528	
Happy	0.82	0.91	0.87	879	
Sad	0.55	0.52	0.53	594	
Surprise	0.70	0.81	0.75	416	
Neutral	0.62	0.69	0.65	626	
accuracy			0.66	3589	
macro avg	0.62	0.62	0.62	3589	
weighted avg	0.65	0.66	0.65	3589	

Hình 3.4.1.3.5: Recall, precision, f1-score của test set



Hình 3.4.1.3.6: Confusion matrix test set

3.4.1.4 Kết luận

Mô hình huấn luyện đạt độ chính xác là 66% cho tập test, với các điểm recall, precision, f1-score đa số đạt trên 50%, chỉ có lớp fear là thấp nhất với các điểm recall, precision, f1-score lần lượt là 0.58, 0.40, 0.47.

Random test trên test set:



3.4.2 Xây dựng, đánh giá mô hình dự đoán giới tính với UTKFace

3.4.2.1 Mô hình

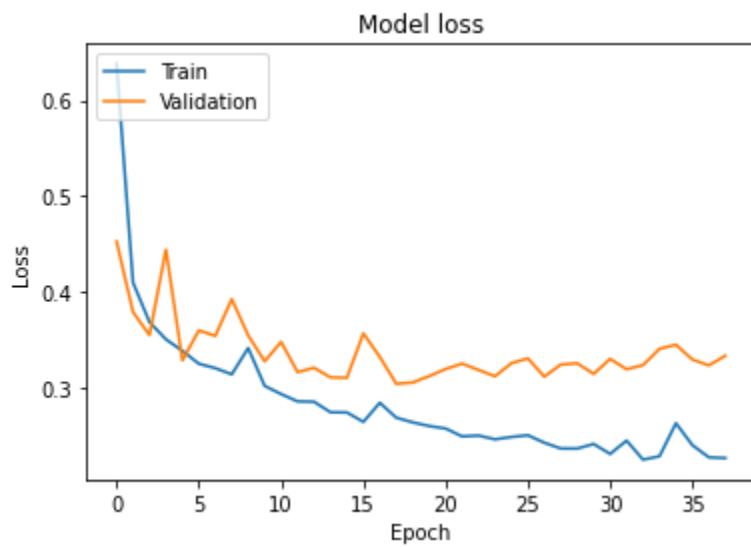
Giới tính

Model: "model_1"

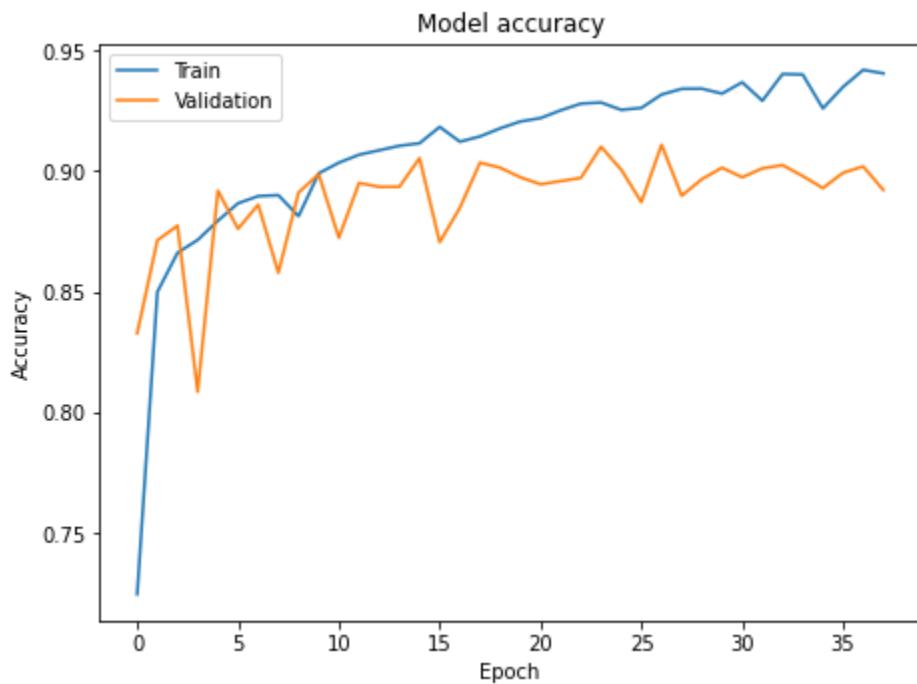
Layer (type)	Output Shape	Param #
<hr/>		
input_2 (InputLayer)	[(None, 48, 48, 3)]	0
conv2d_4 (Conv2D)	(None, 48, 48, 32)	896
dropout_5 (Dropout)	(None, 48, 48, 32)	0
activation_4 (Activation)	(None, 48, 48, 32)	0
max_pooling2d_4 (MaxPooling 2D)	(None, 24, 24, 32)	0
conv2d_5 (Conv2D)	(None, 24, 24, 64)	18496
dropout_6 (Dropout)	(None, 24, 24, 64)	0

activation_5 (Activation)	(None, 24, 24, 64)	0
max_pooling2d_5 (MaxPooling 2D)	(None, 12, 12, 64)	0
conv2d_6 (Conv2D)	(None, 12, 12, 128)	73856
dropout_7 (Dropout)	(None, 12, 12, 128)	0
activation_6 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_6 (MaxPooling 2D)	(None, 6, 6, 128)	0
conv2d_7 (Conv2D)	(None, 6, 6, 256)	295168
dropout_8 (Dropout)	(None, 6, 6, 256)	0
activation_7 (Activation)	(None, 6, 6, 256)	0
max_pooling2d_7 (MaxPooling 2D)	(None, 3, 3, 256)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 64)	147520
dropout_9 (Dropout)	(None, 64)	0
sex_out (Dense)	(None, 1)	65
<hr/>		
Total params:	536,001	
Trainable params:	536,001	
Non-trainable params:	0	

3.4.2.2 Kết quả đánh giá trên tập train, validation, test.



Hình 3.4.2.2.1: Loss trên training set và validation set

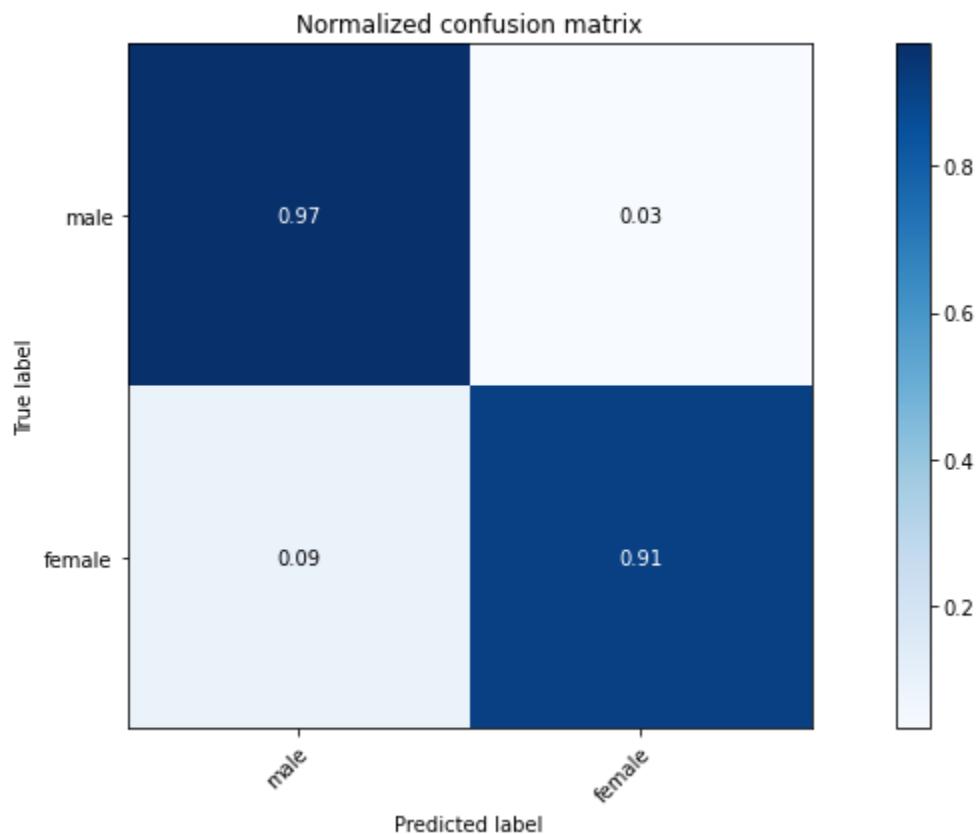


Hình 3.4.2.2.2: Độ chính xác trên training set và validation set

3.4.2.3 Đánh giá confusion matrix, recall, precision, f1-score

	precision	recall	f1-score	support
male	0.92	0.97	0.94	7925
female	0.96	0.91	0.93	7247
accuracy			0.94	15172
macro avg	0.94	0.94	0.94	15172
weighted avg	0.94	0.94	0.94	15172

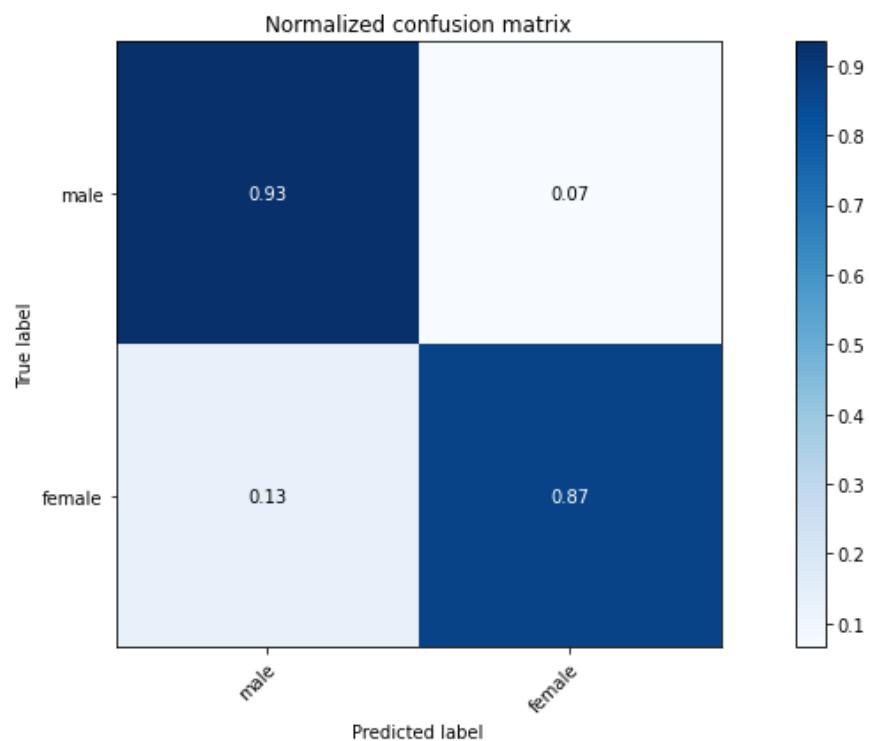
Hình 3.4.2.3.1: Recall, precision, f1-score của training set



Hình 3.4.2.3.2: Confusion matrix training set

	precision	recall	f1-score	support
male	0.89	0.93	0.91	2007
female	0.92	0.87	0.89	1787
accuracy			0.90	3794
macro avg	0.91	0.90	0.90	3794
weighted avg	0.90	0.90	0.90	3794

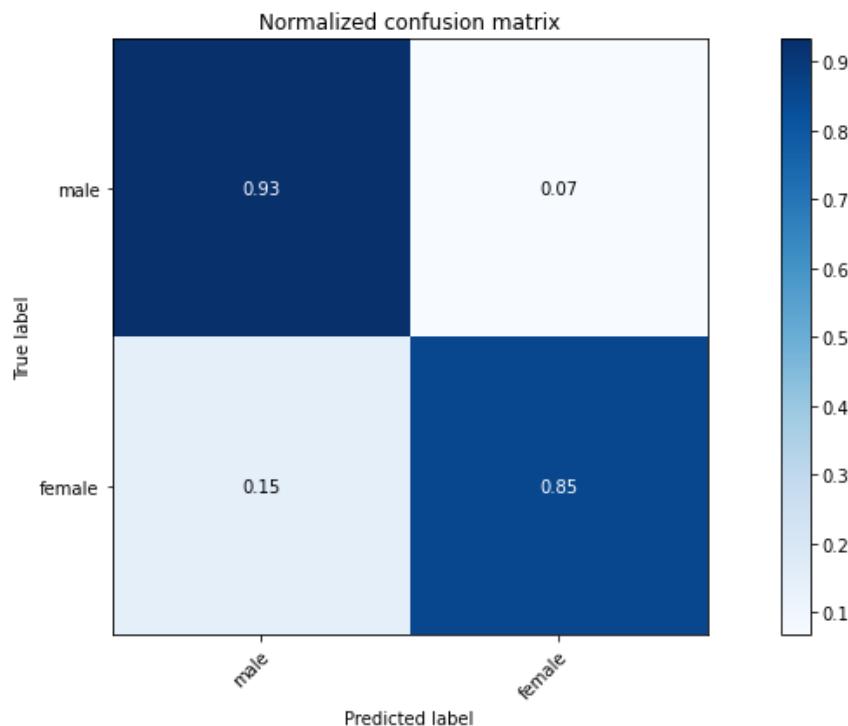
Hình 3.4.2.3.3: Recall, precision, f1-score của validation set



Hình 3.4.2.3.4: Confusion matrix validation set

	precision	recall	f1-score	support
male	0.87	0.93	0.90	2459
female	0.92	0.85	0.89	2283
accuracy			0.89	4742
macro avg	0.90	0.89	0.89	4742
weighted avg	0.90	0.89	0.89	4742

Hình 3.4.2.3.5: Recall, precision, f1-score của test set

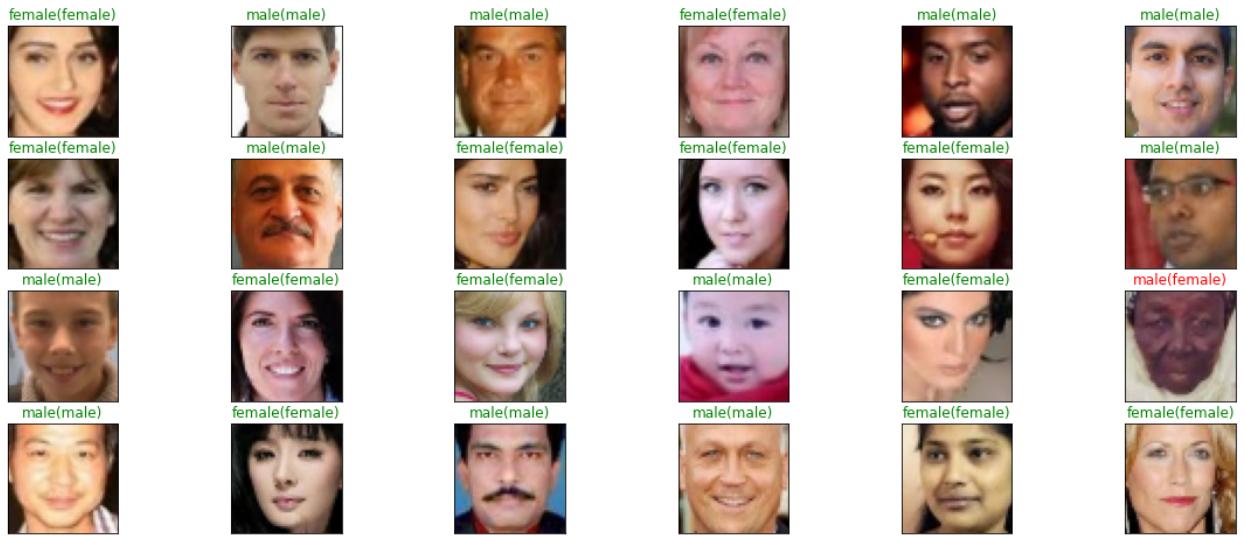


Hình 3.4.2.3.6: Confusion matrix test set

3.4.2.4 Kết luận

Mô hình giới tính được huấn luyện tốt, các chỉ số recall, precision, f1-score của 2 giới đều trên 84% cả 3 tập train, validation và test set. Độ chính xác trên tập test là 89%.

Random test trên test set



3.4.3 Xây dựng, đánh giá mô hình dự đoán Tuổi với UTKFace

3.4.3.1 Mô hình

Mô hình xây dựng huấn luyện theo nhóm tuổi

Model: "model"

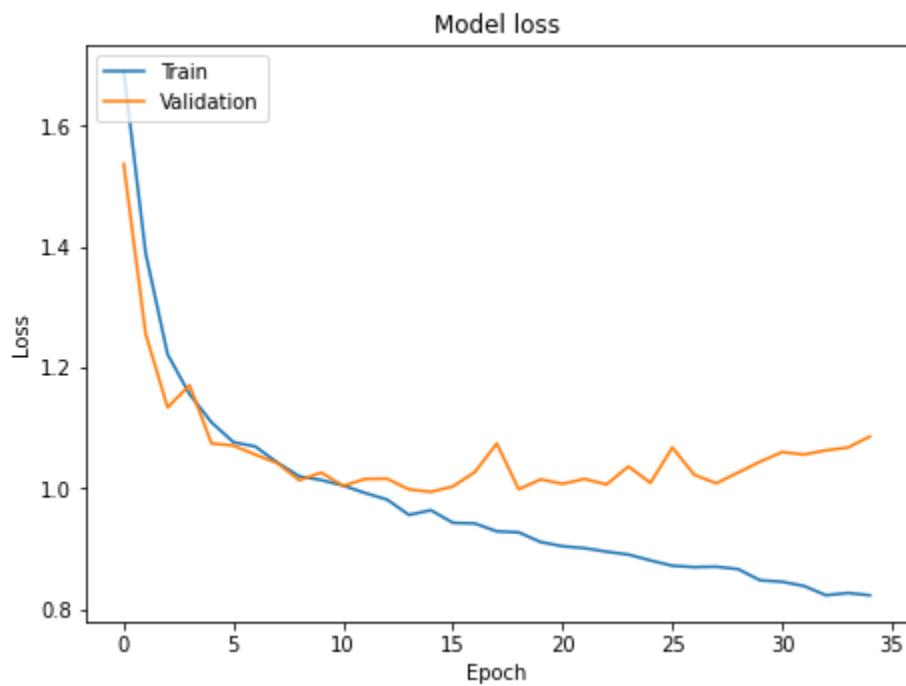
Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 48, 48, 3)]	0
conv2d (Conv2D)	(None, 48, 48, 48)	1344
dropout (Dropout)	(None, 48, 48, 48)	0
activation (Activation)	(None, 48, 48, 48)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 48)	0
conv2d_1 (Conv2D)	(None, 24, 24, 96)	41568
dropout_1 (Dropout)	(None, 24, 24, 96)	0
activation_1 (Activation)	(None, 24, 24, 96)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 12, 12, 96)	0

conv2d_2 (Conv2D)	(None, 12, 12, 192)	166080
dropout_2 (Dropout)	(None, 12, 12, 192)	0
activation_2 (Activation)	(None, 12, 12, 192)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 6, 6, 192)	0
conv2d_3 (Conv2D)	(None, 6, 6, 384)	663936
dropout_3 (Dropout)	(None, 6, 6, 384)	0
activation_3 (Activation)	(None, 6, 6, 384)	0
max_pooling2d_3 (MaxPooling 2D)	(None, 3, 3, 384)	0
flatten (Flatten)	(None, 3456)	0
dense (Dense)	(None, 64)	221248
dropout_4 (Dropout)	(None, 64)	0
age_out (Dense)	(None, 6)	390

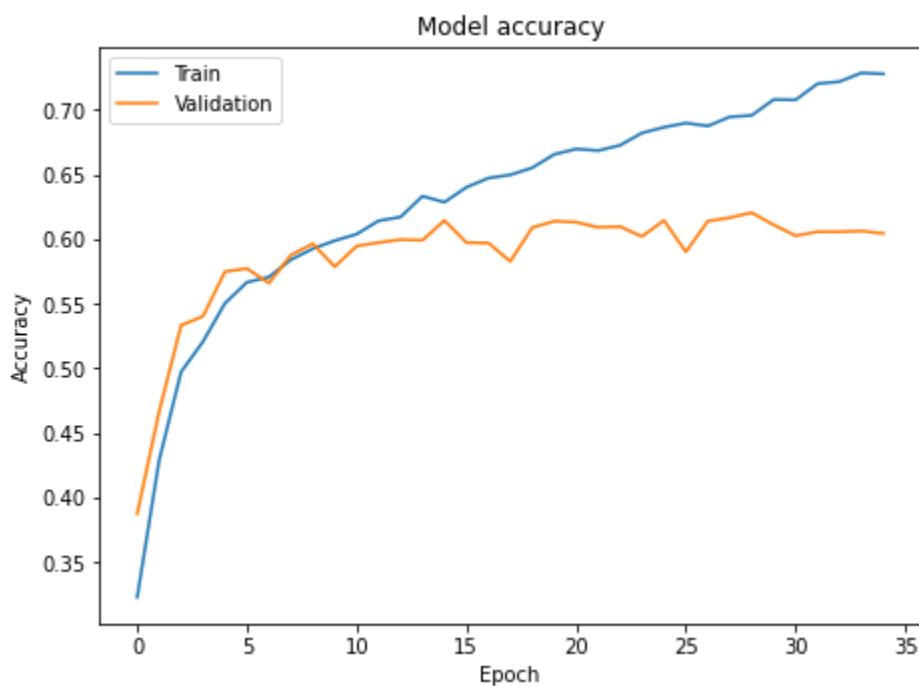
=====

Total params: 1,094,566
Trainable params: 1,094,566
Non-trainable params: 0

3.4.3.2 Kết quả đánh giá trên tập train, validation, test.



Hình 3.4.3.1: Loss trên training set và validation set

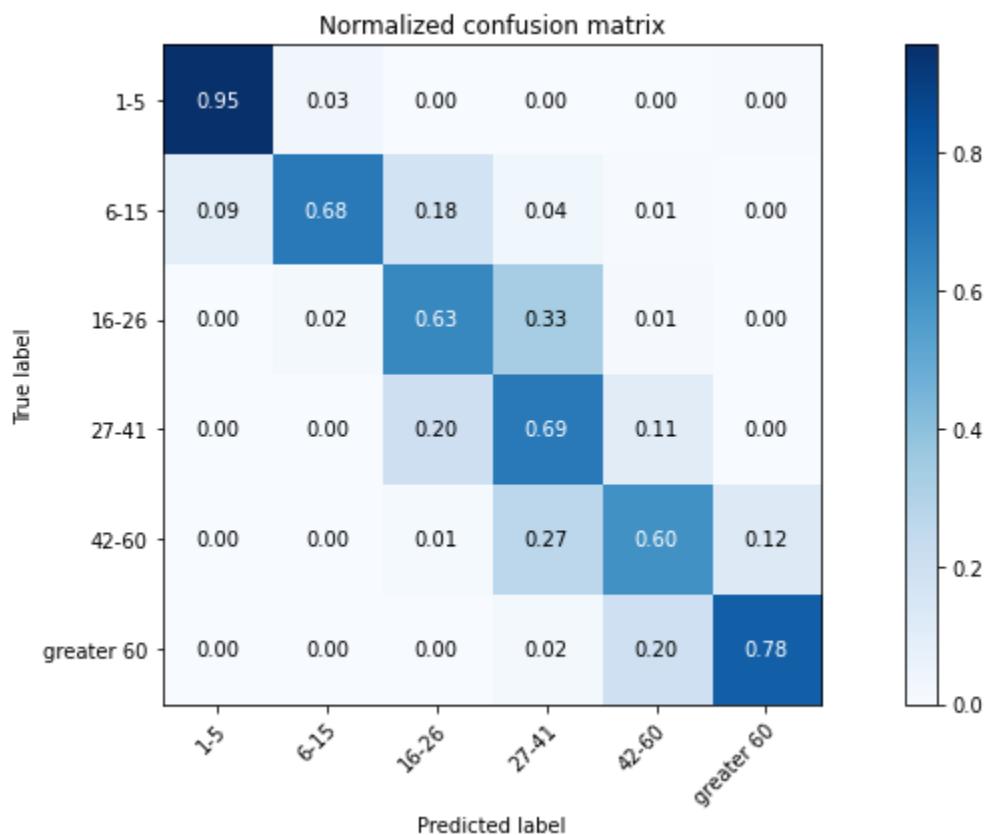


Hình 3.4.3.2: Độ chính xác trên training set và validation set

3.4.3.3 Đánh giá confusion matrix, recall, precision, f1-score

	precision	recall	f1-score	support
1-5	0.94	0.95	0.95	1482
6-15	0.81	0.68	0.74	941
16-26	0.68	0.63	0.65	3862
27-41	0.61	0.69	0.64	4664
42-60	0.65	0.60	0.62	2690
greater 60	0.78	0.78	0.78	1533
accuracy			0.69	15172
macro avg	0.74	0.72	0.73	15172
weighted avg	0.70	0.69	0.69	15172

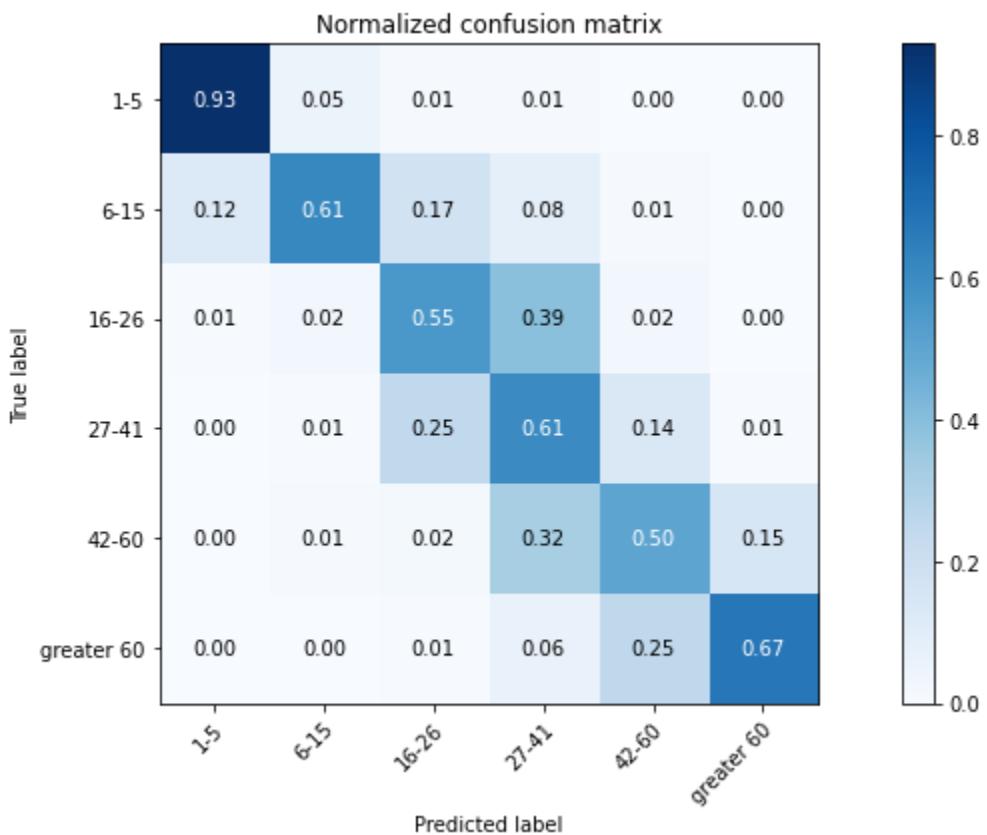
Hình 3.4.3.1: Recall, precision, f1-score của training set



Hình 3.4.3.2: Confusion matrix training set

	precision	recall	f1-score	support
1-5	0.90	0.93	0.91	375
6-15	0.72	0.61	0.66	235
16-26	0.60	0.55	0.58	946
27-41	0.53	0.61	0.57	1173
42-60	0.54	0.50	0.52	669
greater 60	0.71	0.67	0.69	396
accuracy			0.61	3794
macro avg	0.67	0.65	0.66	3794
weighted avg	0.62	0.61	0.61	3794

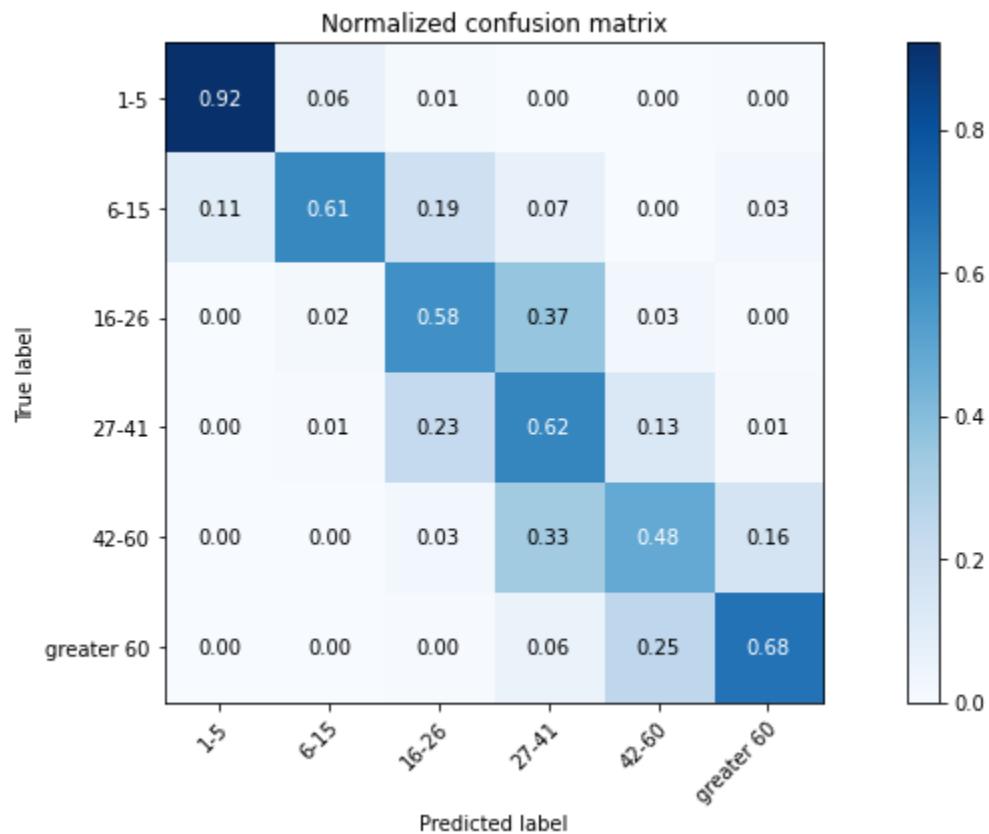
Hình 3.4.3.3: Recall, precision, f1-score của validation set



Hình 3.4.3.4: Confusion matrix validation set

	precision	recall	f1-score	support
1-5	0.93	0.92	0.92	506
6-15	0.72	0.61	0.66	289
16-26	0.62	0.58	0.60	1198
27-41	0.54	0.62	0.58	1461
42-60	0.54	0.48	0.51	820
greater 60	0.67	0.68	0.67	468
accuracy			0.62	4742
macro avg	0.67	0.65	0.66	4742
weighted avg	0.63	0.62	0.62	4742

Hình 3.4.3.5: Recall, precision, f1-score của test set

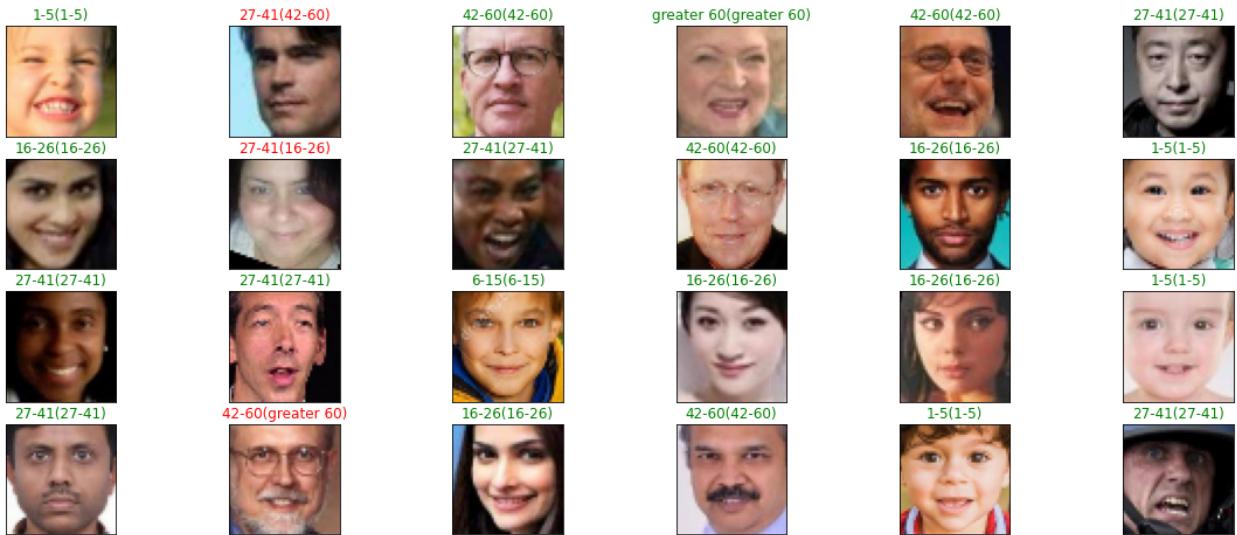


Hình 3.4.3.6: Confusion matrix test set

3.4.3.4 Kết luận

Kết luận: Độ chính xác của test set là 62%, chỉ có recall của lớp “42-60” là 48% (nhỏ hơn 50%), các chỉ số recall, precision, f1-score của các lớp khác đều trên 50%.

Random test trên test set



3.5 Xây dựng hệ thống

3.5.1 Giao diện người dùng



- Xây dựng giao diện sử dụng thư viện Tkinter python, gồm các thành phần:
 - (1) Hiển thị đường dẫn file ảnh/video tải lên hệ thống.

- (2) Hiển thị loại file đầu vào hệ thống hỗ trợ gồm:
 - + Image: dự đoán cho ảnh
 - + Vieo: dự đoán cho video
 - + Camera: dự đoán cho webcam
- (3) Browse: click để chọn file ảnh/video.
- (4) PLAY: click để bắt đầu chạy dự đoán
- (5) STOP: dừng xử lý
- Hiển thị kết quả dự đoán gồm:
 - + Capture khuôn mặt, vẽ kết quả dự đoán trên khung hình sử dụng OpenCV
- Các kết quả dự đoán gồm:
 - + Giới tính: male (nam) hoặc female (nữ).
 - + Tuổi: nằm trong 9 nhóm tuổi đã phân nhóm ở mục III.3 pre-processing
 - + Cảm xúc: gồm 7 giá trị trong tập dataset FER-2013 là Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral.

3.5.2 Back-end xử lý dự đoán

- Training model: sử dụng Keras để xây dựng và dự đoán từng ảnh trong video hoặc webcam.
- Dự đoán: Video hoặc dữ liệu từ web cam sẽ được đọc từng frame ảnh, mỗi frame ảnh này sẽ được đưa qua model tìm mặt người, nếu có sẽ cho ra vị trí của mặt. Sau đó gương mặt sẽ đi qua 3 model là: giới tính, tuổi, cảm xúc. Sau đó sẽ dùng OpenCV in kết quả lên frame ảnh và render ra GUI.

4. Giới hạn của dự án

Độ chính xác dùng trên dữ liệu thực tế chưa tốt, mặc dù trên test dataset là khá tốt, do đó cần có dữ liệu thực tế để train model, tốt nhất là lấy dữ liệu từ camera muốn lấy video.

TÀI LIỆU THAM KHẢO

1. Michael A.Nielsen. “Neural Networks and Deep Learning”, Determination Press, 2015.
2. CS231n Convolutional Neural Networks for Visual Recognition
<https://cs231n.github.io/convolutional-networks>.
3. An intuitive guide to Convolutional Neural Networks:
<https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050>.
4. FER-2013 dataset: <https://www.kaggle.com/datasets/deadskull7/fer2013>
5. UTKFace dataset: <https://www.kaggle.com/datasets/jangedoo/utkface-new>