Tiểu luận môn học

Khai phá dữ liệu

Đề tài 28: Nhận dạng cảm xúc khuôn mặt khách hàng từ camera, thống kê và biểu đồ hoá thông tin

Nhóm 8: Nguyễn Hải Hoàng, Thân Trọng Thành, Hà Nhật Minh
Khoa Công nghệ thông tin, Học viện Kĩ thuật quân sự
{hoangnh5901 , trongthanht3, minhk17khmt}@gmail.com

Ngày 15 tháng 10 năm 2021

Tóm tắt nội dung

Xây dựng hệ thống camera ghi lại các cám xúc của khách hàng, giúp thống kê lại dữ liệu cảm xúc của khách hàng khi vào cửa hàng hoặc khu vực. Một website để xem biểu đồ, thông tin, số lượng và thời gian thống kê được trên dữ liệu thu thập từ camera.

1 Dữ liệu và tiền xử lý dữ liệu

1.1 Tập dữ liệu

- Trong bài toán này, chúng ta sẽ sử dụng 2 bộ dữ liệu là **FER** và **FER+**, dữ liệu ảnh các khuôn mặt và được đánh nhãn cảm xúc cho từng khuôn mặt.
- Tải xuống dữ liệu:
  - **FER**
  - **FER+**

1.1.1 Tập dữ liệu FER

Dữ liệu bao gồm hình ảnh khuôn mặt, màu xám và kích thước 48x48 pixel. Dữ liệu được đánh nhãn bao gồm (0 = Giận dữ, 1 = Chán ghét, 2 = Sợ hãi, 3 = Hạnh phúc, 4 = Buồn, 5 = Ngạc nhiên, 6 = Trung lập). Dữ liệu được lưu trong tệp ***train.csv*** chứa hai cột, "emotion" và "pixel". Cột "emotion" chứa mã số từ 0 đến 6, bao gồm cả cảm xúc hiện diện trong hình ảnh. Cột "pixel" chứa một chuỗi pixel cho mỗi hình ảnh. Nội dung của chuỗi này một giá trị pixel được phân tách bằng dấu cách theo thứ tự chính của hàng.

emotion	pixels
0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171...
0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171...
2	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171...

Bảng 1: 3 bản ghi đầu tiên của tệp **train.csv**

1.1.2 Tập dữ liệu FER+

Bộ dữ liệu FER + cung cấp một tập hợp các nhãn mới cho tập dữ liệu FER bên trên. Trong FER +, mỗi hình ảnh đã được gắn nhãn bởi 10 người gắn thẻ có nguồn gốc từ cộng đồng, cung cấp độ chân thực nền chất lượng tốt hơn cho cảm xúc hình ảnh tĩnh so với nhãn FER ban đầu. Dữ liệu được lưu trong tệp các tệp ***label.csv*** nằm trong các thư mục ***FER2013Train***, ***FER2013Test***, ***FER2013Valid*** chứa các cột "Usage", "Image name", "neutral", "happiness", "surprise", "sadness", "anger", "disgust", "fear", "contempt", "unknown", "NF". Trong đó ta cần quan tâm đến cột **Image name** là tên của tệp ảnh và giá trị cảm xúc của ảnh trong các cột **neutral**, **happiness**, **surprise**, **sadness**, **anger**, **disgust**, **fear**, **contempt**

Usage	Image name	neutral	happiness	surprise	sadness	anger	disgust	fear	contempt	unknown	NF
Training	fer0000000.png	4	0	0	1	3	2	0	0	0	0
Training	fer0000001.png	6	0	1	1	0	0	0	0	2	0
Training	fer0000002.png	5	0	0	3	1	0	0	0	1	0
Training	fer0000003.png	4	0	0	4	1	0	0	0	1	0
Training	fer0000004.png	9	0	0	1	0	0	0	0	0	0
Training	fer0000005.png	6	0	0	1	0	0	1	1	1	0
Training	fer0000006.png	2	0	0	8	0	0	0	0	0	0

Bảng 2: 7 bản ghi đầu tiên của tệp **label.csv** trong thư mục **FER2013Train**

1.2 Xử lý dữ liệu

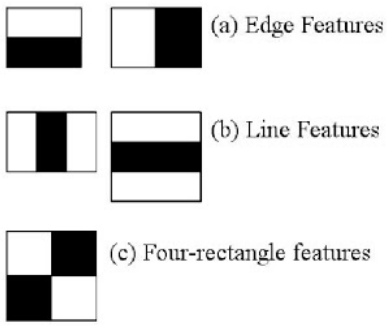
- Chúng ta sẽ trích xuất **pixels** thành ảnh từ bộ dữ liệu **FER** và kết hợp **emotion** được gán nhãn của bộ dữ liệu **FER+**
  - Trích xuất ảnh từ bộ dữ liệu **FER**:
    - Tiến hành đọc tệp **train.csv**, trích xuất cột **pixels**, với mỗi hàng, ta biến đổi chuỗi "70 80 82..." thành mảng các điểm ảnh [70,80,82,...], mảng này sẽ có độ dài là **2304** (48 * 48).
    - Biến đổi mảng **1 chiều** thành mảng **2 chiều** với kích thước là 48 * 48, lưu lại thành tệp với tên là "fer{**số thứ tự**}.png" (ví dụ **fer0000000.png**), vậy chúng ta sẽ có bức ảnh **xám** kích thước **48 * 48**.
  - Gán nhãn từ bộ dữ liệu **FER+**:
    - Sau khi trích xuất ảnh ra tệp, tên tệp sẽ có dạng **fer0000000.png**, **fer0000001.png**,...
    - Ta xẽ lần lượt xử lý trên tệp **label.csv** trong các thư mục ***FER2013Train***, ***FER2013Test***, ***FER2013Valid***. Mỗi thư mục tương ứng với các tập **Huấn luyện**, **Kiểm thử** và **Đánh giá**.
    - Tiến hành đọc tệp **label.csv** lấy ra tên ảnh tại cột **Image name** để truy xuất đến tệp ảnh tương ứng, , lấy ra cội liên quan đến **emotion** có giá trị cao nhất (**neutral**, **happiness**, **surprise**, **sadness**, **anger**, **disgust**, **fear**, **contempt**) rồi gán giá trị này làm nhãn cho ảnh tương ứng và sau đó biến đổi giá trị này qua **One hot encoding**.
    - Từ 3 tệp **label.csv** trong các mục trên ta sẽ lưu thành các biến chứa dữ liệu tương đương là **traning**, **test** và **validation**.

2 Phương pháp nhận dạng khuôn mặt và cảm xúc

Bài toán nhận diện cảm xúc có thể được chia ra làm hai bước cụ thể là “Nhận biết khuôn mặt” (Face detection) và “Nhận diện cảm xúc” (Facial expression recognition)

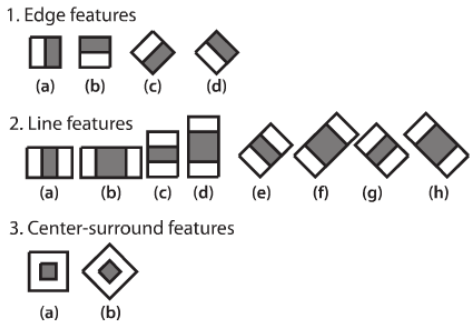
2.1 Nhận biết khuôn mặt

Trong bài viết này chúng tôi sử dụng *Haar feature-casesd cascade classifier* [3] cho bước nhận biết khuôn mặt. Module HC (Haar Cascaded) có thể sử dụng để nhận biết khuôn mặt được trong đa số các trường hợp ngoài thực tế với tốc độ xử lý nhanh, nhẹ và khả năng tích hợp dễ dàng cho mọi hệ thống. Cách hoạt động của HC tương đối giống với CNN (Convolutional Neural Networks) bằng cách sử dụng các bộ lọc để lấy ra các đặc trưng của ảnh. Các ví dụ bộ lọc được liệt kê ở hình 1 để bắt được các đặc trưng của ảnh.



Hình 1: a) các bộ lọc bắt các cạnh trong ảnh, b) bắt các đường thẳng trong ảnh, c) bộ lọc về đặc trưng 4-hình vuông dưới đây

hoặc các đặc trưng nằm gọn trong trung tâm một vùng. Trong hình 2 dưới đây:



Hình 2: các đặc trưng trong trung tâm một vùng

2.2 Nhận dạng cảm xúc

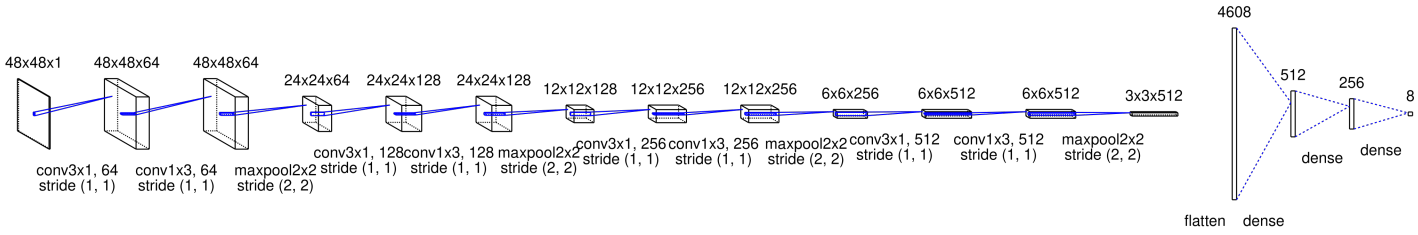
2.2.1 Mô hình DCNN

Nhận dạng cảm xúc về cơ bản là một bài toán phân loại ảnh. Do đó, một mô hình học sâu CNN hiện đại hoạt động tốt trong việc phân loại hình ảnh cũng nên hoạt động tốt trong việc nhận dạng nét mặt. Trong bài này, chúng tôi sẽ sử dụng một mạng VGG được tùy chỉnh tùy chỉnh để đạt hiệu suất tốt hơn trên tập dữ liệu FER+.

2.2.2 Kiến trúc mạng

Đầu vào cho mô hình nhận dạng cảm xúc là một ảnh xám được thay đổi kích thước về tỉ lệ 64x64. Đầu ra là 8 lớp cảm xúc: tự nhiên, vui vẻ, ngạc nhiên, buồn bã, tức giận, chán ghét, sợ hãi, khinh thường. Mô hình của chúng tôi được tinh chỉnh lại từ VGG13, thể hiện trong hình 3. Mô hình có 8 lớp convolution, xen kẽ với các lớp max pooling và dropout. Cụ thể hơn, nối tiếp lớp đầu vào, sẽ có 2 lớp convolution với 64 kernel có kích thước lần lượt là 3x1 và 1x3. Ngay sau đó là một lớp max pooling nối tiếp dropout với tỉ lệ 25%. Cấu trúc này được sử dụng lặp lại và chỉ thay đổi số lượng kernel. Sau toàn bộ các lớp convolution, chúng tôi thêm một lớp flatten và ngay sau đó là 2 lớp dense với mỗi lớp lần lượt là 512 node và 256 node, kèm theo mỗi lớp đó là một lớp dropout với tỉ lệ 25%. Ở lớp cuối cùng, chúng tôi sử dụng một lớp soft-max để tạo kết quả đầu ra.

Mặc dù tập dữ liệu FER+ có khoảng 35.000 ảnh, các lớp dropout sẽ giúp mô hình của chúng tôi tránh khỏi hiệu ứng over-fitting.



Hình 3: Mạng VGG13 đã được tùy chỉnh

2.2.3 Tiến hành học dữ liệu

Chúng tôi tiến hành học dữ liệu cho mô hình hoàn toàn từ đầu với tập dữ liệu FER+ với cùng lượng phân chia dữ liệu cho tập training, validation và testing. Nhờ lượng nhãn lớn được gắn với mỗi ảnh, chúng tôi có thể tạo ra một phân phối xác suất các lớp cho mỗi ảnh. Dưới đây, chúng tôi sẽ trình bày cách dữ liệu phân phối nhãn trong quá trình training của mạng.

Cho tập hợp N mẫu dữ liệu training $\mathbf{I}_i, i = 1, ..., N$. Với mẫu i^{th}, cho đầu ra của mạng sau lớp soft-max là $q_k^i, k = 1, ..., 8$, và phân phối nhãn của mẫu này là $p_k^i, k = 1, ..., 8$. Từ đó, ta có:

$$\sum_{k=1}^{8} q_k^i = 1; \qquad \sum_{k=1}^{8} p_k^i = 1.$$

Với mô hình này chúng tôi sử dùng cross-entropy (CEL) cho hàm mất mát (loss function)

2.2.4 Hàm mất mát

Chúng tôi sử dụng hàm mất mát cross-entropy hay còn được gọi là Softmax loss. Đây là hàm loss thông dụng nhất được sử dụng cho bài toán multi-class classification. Với việc coi phân phối nhãn là yêu cầu cuối cùng mà mô hình DCNN đạt được, biểu diễn qua phương trình:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{k=1}^{8} p_k^i \log q_k^i$$

2.3 Kết quả thực nghiệm

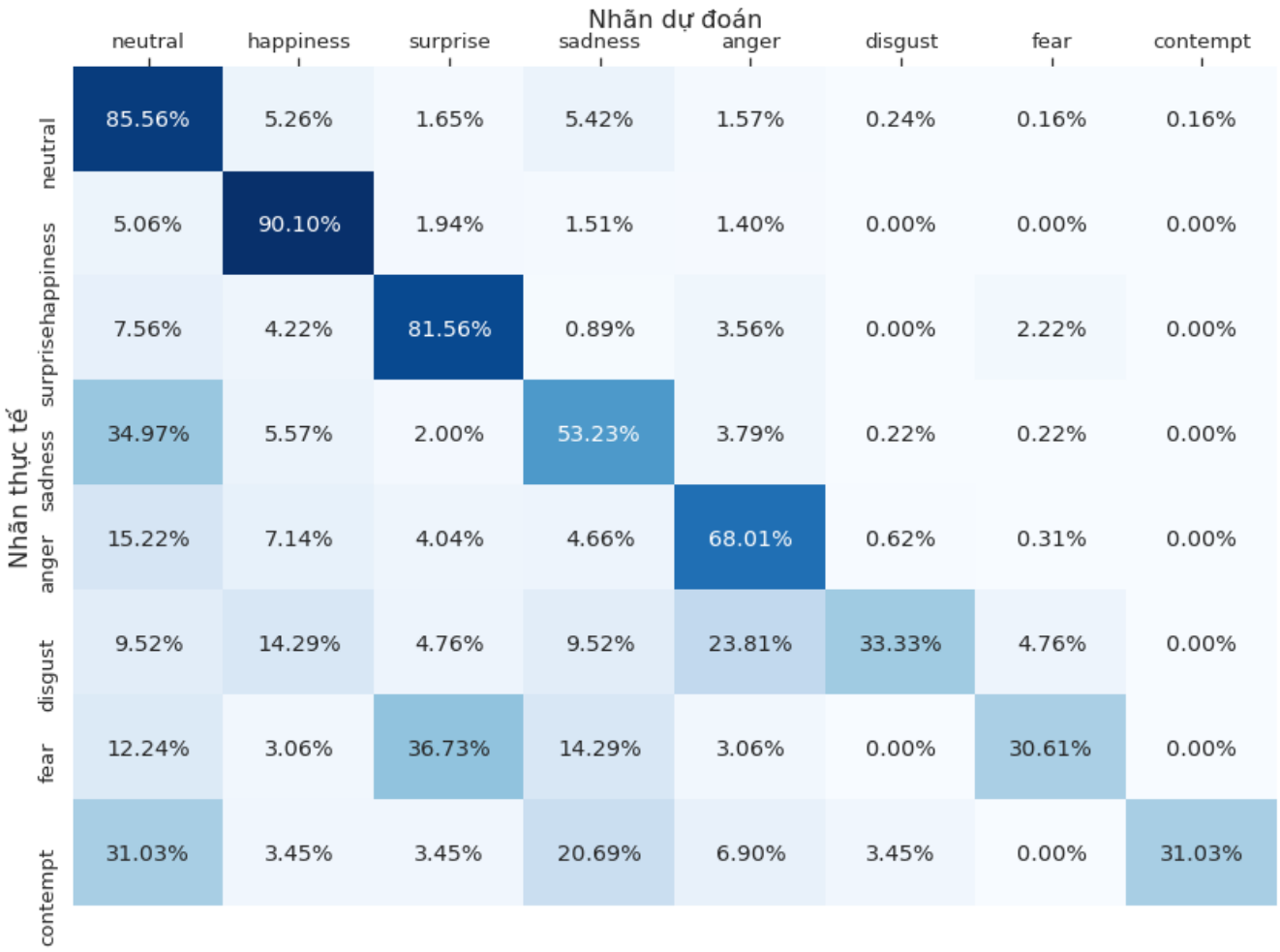
Như đã đề cập ở trên, mỗi ảnh sẽ được gắn 10 nhãn. Một lớp lọc phân phối nhãn được thiết kế để đặt về 0 các cảm xúc có số lần đánh nhãn nhỏ hơn 1 lần. Tần suất của nhãn cũng được chuẩn hóa để đảm bảo tổng phân phối bằng 1.

Chúng tôi tiến hành học dữ liệu trên mạng VGG13 tùy chỉnh 5 lần, và kết quả được thể hiện trong bảng 3:

Lần học					Độ chính xác
1	2	3	4	5	
78.33%	78.83%	79.21%	77.91%	79.32%	**78.72 ± 0.28** %

Bảng 3: Kết quả thử nghiệm

Hình 4 thể hiện confusion matrix từ kết quả tốt nhất của mô hình. Chúng ta có thể thấy rằng mô hình thể hiện rất tốt ở hầu hết các cảm xúc ngoại trừ **disgust**, **fear** hay **contempt**. Điều này tương đối dễ hiểu vì chúng ta có khá ít dữ liệu ở các phân lớp này trong bộ dữ liệu FER+, dẫn đến việc mô hình sẽ thể hiện khả năng học và phân lớp yếu hơn nhiều ở các lớp này.



3 Lưu trữ dữ liệu cảm xúc của khách hàng

3.1 Thông tin lưu trữ

- Dữ liệu của khách hàng sẽ được lưu làm 3 loại chính gồm: Thời gian, cảm xúc và hình ảnh khách hàng
  - **Thời gian**: Hệ thống sẽ lưu lại thời gian khi khách hàng, đối tượng được ghi lại vào hệ thống từ đó giúp truy xuất được dữ liệu theo các mốc thời gian 1 cách dễ dàng hơn.
  - **Cảm xúc**: Khi hệ thống đã bắt được đối tượng cần tìm thì cảm xúc của khách hàng (vui, buồn, sợ, ...) sẽ được ghi lại nhờ mô hình thiết kế như trên và dùng bộ dữ liệu Fer để trả về nơi lưu trữ dữ liệu cảm xúc của khách hàng ở mốc thời gian đó.
  - **Hình ảnh khách hàng**: Hệ thống dùng camera lưu lại hình ảnh của khách hàng cần nhận dạng từ đó giúp việc chuyển hóa hay mô tả dữ liệu them trực quan và sinh động.

3.2 Tiền xử lý dữ liệu lưu trữ

- Khi hai hoặc nhiều người mua hàng trong cùng một khung hình thì khi đó vấn đề đặt ra là làm sao để biết giữa 2 khuôn mặt trên 2 khung hình trước và sau của camera là cùng một người để tối ưu hoá việc lưu trữ dữ liệu, chúng ta sẽ sử dụng thuật toán **FaceNet**.

3.2.1 FaceNet

- **FaceNet** là dựa trên việc nhúng mỗi ảnh vào không gian Euclide bằng cách sử dụng mạng CNN. Mạng được huấn luyện sao cho khoảng cách bình phương L2 trong không gian nhúng tương ứng với khoảng cách giữa các khuôn mặt: Khuôn mặt của cùng một người có khoảng cách nhỏ và khuôn mặt của những người khác nhau có khoảng cách lớn. Khi đó, mỗi khuôn mặt được đại diện bởi một vector đặc trưng 128 chiều.

3.2.2 Triplet loss

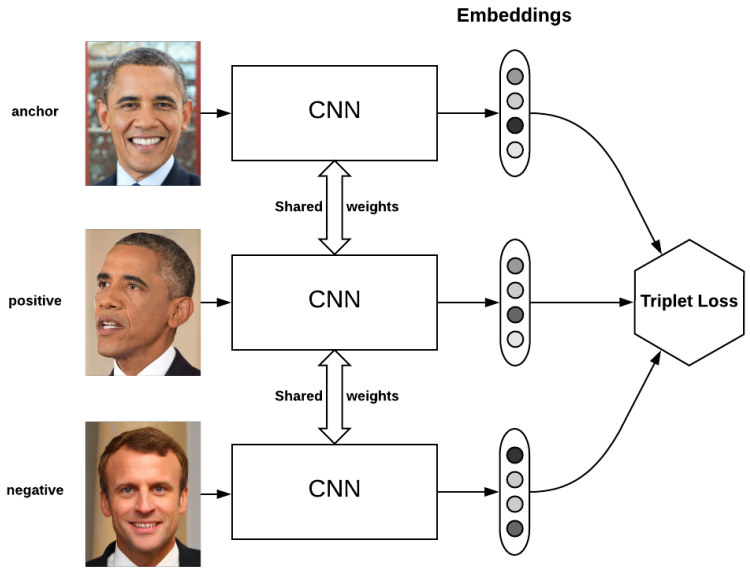
- **Triplet loss**: Trong facenet, quá trình encoding của mạng convolutional neural network đã giúp ta mã hóa bức ảnh về 128 chiều. Sau đó những véc tơ này sẽ làm đầu vào cho hàm loss function đánh giá khoảng cách giữa các véc tơ.
- Để áp dụng triple loss, chúng ta cần lấy ra 3 bức ảnh trong đó có một bức ảnh là anchor. Kí hiệu ảnh **Anchor**, **Positive**, **Negative** lần lượt là $\mathbf{A}, \mathbf{P}, \mathbf{N}$.
- Mục tiêu của hàm loss function là tối thiểu hóa khoảng cách giữa 2 ảnh khi chúng là negative và tối đa hóa khoảng cách khi chúng là positive. Như vậy chúng ta cần lựa chọn các bộ 3 ảnh sao cho:
- Ảnh **Anchor** và **Positive** khác nhau nhất: cần lựa chọn để khoảng cách $d(\mathbf{A}, \mathbf{P})$ lớn. Điều này cũng tương tự như bạn lựa chọn một ảnh của mình hồi nhỏ so với hiện tại để thuật toán học khó hơn. Nhưng nếu nhận biết được thì nó sẽ thông minh hơn.
- Ảnh **Anchor** và **Negative** giống nhau nhất: cần lựa chọn để khoảng cách $d(\mathbf{A}, \mathbf{N})$ nhỏ. Điều này tương tự như việc thuật toán phân biệt được ảnh của một người anh em giống bạn với bạn.
- Triplot loss function luôn lấy 3 bức ảnh làm input và trong mọi trường hợp ta kì vọng:

$$d(\mathbf{A}, \mathbf{P}) < d(\mathbf{A}, \mathbf{N})$$

- Như vậy hàm loss function sẽ là:

$$\mathcal{L}(\mathbf{A}, \mathbf{P}, \mathbf{N}) = \sum_{i=0}^{n} ||f(\mathbf{A}_i) - f(\mathbf{P}_i)||_2^2 - ||f(\mathbf{A}_i) - f(\mathbf{N}_i)||_2^2 + \alpha$$

n là số lượng các bộ 3 hình ảnh được đưa vào huấn luyện.
α để làm cho khoảng cách giữa $d(\mathbf{A}, \mathbf{P})$ và $d(\mathbf{A}, \mathbf{N})$ lớn hơn,
chúng ta sẽ cộng thêm vào vế trái một hệ số không âm rất nhỏ.



Hình 4: Mạng CNN với hàm Triplet loss

3.2.3 Pretrain model Facenet

- Hầu hết chúng ta khi xây dựng một thuật toán nhận diện khuôn mặt sẽ không cần phải train lại mô hình facenet mà tận dụng lại các mô hình pretrain sẵn có. Sẽ không cần phải tốn thời gian và công sức nếu không có đủ tài nguyên và dữ liệu. Những mô hình pretrain được huấn luyện trên các dữ liệu lên tới hàng triệu ảnh. Do đó có khả năng mã hóa rất tốt các bức ảnh trên không gian 128 chiều. Việc còn lại của chúng ta là sử dụng lại mô hình, tính toán embedding véc tơ và huấn luyện embedding véc tơ bằng một classifier đơn giản để phân loại classes.
- Ở bài toán này, sử dụng thư viện **face_recognition** đã được tích hợp modal **FaceNet** có sẵn để embedding từ đầu vào là khuôn mặt.

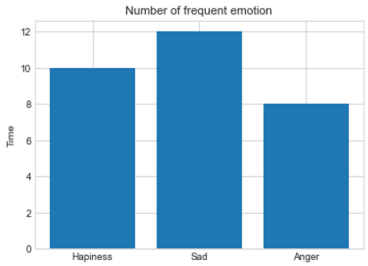
- Sau khi phân biệt được các khách hàng nằm trong một khung hình, chúng ta sẽ chỉ lưu trữ những cảm xúc khác **neutral**(bình thường), vì khuôn mặt **neutral** sẽ xuất hiện rất nhiều lần và không có ý nghĩa để chúng ta thống kê, lượng dữ liệu khi có **neutral** sẽ quá lớn, dẫn đến những cảm xúc khác cần quan tâm như (**happiness**, **anger**, ...) sẽ có số lượng bản ghi quá nhỏ.

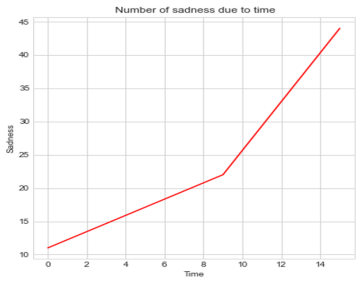
3.3 Nền tảng lưu trữ

- Với 1 dữ liệu lớn và đa dạng từ hệ thống thì cần có 1 nơi lưu trữ phù hợp. MongoDB sẽ là nền tảng tốt để lưu trữ dữ liệu khi hệ thống sẽ lưu vào Server rồi sau đó truy xuất nó trên chính Server đó.
  - MongoDB là một chương trình cơ sở dữ liệu hướng tài liệu đa nền tảng có sẵn nguồn. Được phân loại là một chương trình cơ sở dữ liệu NoSQL, MongoDB sử dụng các tài liệu giống JSON với các lược đồ tùy chọn. MongoDB được phát triển bởi MongoDB Inc. và được cấp phép theo Giấy phép Công cộng phía Máy chủ

3.4 Biểu đồ hoá dữ liệu

- Ở đây sẽ có 1 website hay 1 page cơ bản để biểu diễn dữ liệu từ nền tảng lưu trữ. Website có thể truy cập cũng như truy vấn toàn bộ dữ liệu từ nơi lưu trữ để có thể mô tả 1 cách đầy đủ nhất những thông tin cần thiết mà bài toán đề ra.
  - Công nghệ xây dựng website: ReactJS
- Biểu diện dạng biểu đồ
  - Khi website đã có thông tin cũng như dữ liệu của hệ thống thì đây sẽ là lúc biểu diễn dữ liệu trên website đó thông qua các dạng biểu đồ khác nhau nhằm giúp người xem dễ sử dụng cũng như trực quan hóa được dữ liệu.
  - Một số biểu đồ:





Tài liệu

[1] Yusuke Uchida, ConvNet Drawer.

[2] Kaggle, Challenges in Representation Learning: Facial Expression Recognition Challenge.

[3] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2001

[4] Lucas-Kanade Optical Flow in OpenCV.

[5] Facial Emotion Recognition with Apache MXNet.

[6] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, Zhengyou Zhang, Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution.

[7] François Chollet, Keras: the Python deep learning API.

[8] Microsoft, FER+.

[9] Sebastián Ramírez, FastAPI.

[10] MongoDB Inc, MongoDB.

[11] Face Recognition, face_recognition.