VINBIGDATA

BÁO CÁO ĐỀ TÀI MÔN MÁY HỌC CƠ BẢN

Nhận diện cảm xúc trên khuôn mặt

Học viên
Nguyễn Minh Dũng
Lê Công Pha
Nguyễn Duy Nhất

Giảng viên hướng dẫn
TS. Đinh Viết Sang

Ngày 25 tháng 12 năm 2020

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục từ viết tắt

CNN	**C**onvolutional **N**eural **N**etwork
SIFT	**S**cale **I**nvariance **F**eature **T**ransform
DoG	**D**ifference **o**f **G**aussian
SVM	**S**upport **V**ector **M**achine
FER-2013	Tập dữ liệu **F**acial **E**xpression **R**ecognition 2013
tp	**t**rue **p**ositive
fp	**f**alse **p**ositive
tn	**t**rue **n**egative
fn	**f**alse **n**egative

Tóm tắt nội dung

Trong đề tài này chúng tôi nghiên cứu bài toán nhận diện cảm xúc khuôn mặt. Với đầu vào là ảnh của một khuôn mặt. Đầu ra của bài toán sẽ cho chúng ta biết khuôn mặt đó đang thể hiện cảm xúc gì. Đây là bài toán có nhiều ứng dụng trong nhiều lĩnh vực thực tế như chăm sóc khách hàng, rô-bốt trợ lí, xe tự hành, hệ thống camera giám sát, ...

Bằng việc thử nghiệm nhiều phương pháp rút trích đặc trưng như SIFT, EigenFace, CNN. Kết hợp với các mô hình phân lớp như Logistic regression, SVM, Random Forest. Chúng tôi đã tiến hành huấn luyện và đánh giá mô hình trên bộ dữ liệu FER-2013.

Ngoài ra, chúng tôi đã tạo một web demo nhận diện cảm xúc khuôn mặt.

Chương 1

Tổng quan

1.1 Đặt vấn đề

Cảm xúc khuôn mặt (facial expression) là một trong những tín hiệu mạnh mẽ, tự nhiên và phổ biến nhất để con người truyền tải cảm xúc hay ý định của họ. Đây là bài toán có nhiều ứng dụng trong thực tế như chăm sóc khách hàng, robot trợ lý, xe tự hành và camera giám sát.

Ngoài ra, nhận diện cảm xúc khuôn mặt còn có thể được ứng dụng trong các mô hình học trực tuyến, trong bối cảnh đại dịch COVID-19 ảnh hưởng nặng nề đến việc dạy và học trên toàn thế giới, nhất là các quốc gia phương Tây. Nhận diện được cảm xúc của học sinh trong quá trình giảng dạy trực tuyến sẽ giúp các nhà làm giáo dục đánh giá được mức độ quan tâm của người học, từ đó có thể cải tiến bài giảng của mình.

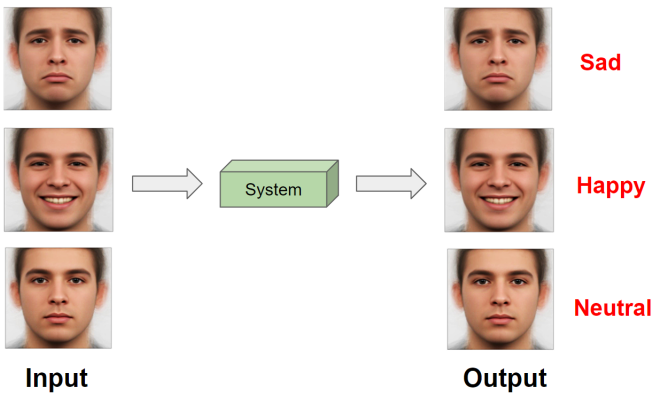
1.2 Giới thiệu bài toán

Đầu vào: Một tấm ảnh khuôn mặt người.

Đầu ra: Cảm xúc của khuôn mặt đó (vui, buồn, giận dữ, bình thường,...)

Ứng dụng: Đây là bài toán có nhiều ứng dụng trong thực tế như:

- Lĩnh vực chăm sóc khách hàng
- Rô-bốt trợ lý



HÌNH 1.1: Minh họa đầu vào và đầu ra của bài toán

- Xe tự hành
- Hệ thống camera giám sát

1.3 Mục tiêu đề tài

- Xây dựng được chương trình demo Nhận diện cảm xúc qua khuôn mặt
- So sánh, đánh giá được ưu nhược điểm của từng phương pháp

1.4 Phương pháp thực hiện

- Tìm hiểu bài toán Nhận diện cảm xúc qua khuôn mặt.
- Tìm hiểu tổng quan về bộ dữ liệu FER-2013.
- Tìm hiểu một số thuật toán rút trích đặc trưng và phân lớp có thể áp dụng vào bài toán Nhận diện cảm xúc.
- Xây dựng chương trình Nhận diện cảm xúc trên khuôn mặt.
- So sánh, đánh giá ưu nhược điểm của từng phương pháp.

1.5 Nội dung thực hiện

- Tìm hiểu đầu vào, đầu ra và ứng dụng của bài toán.
- Tìm hiểu một số thông tin cơ bản về bộ dữ liệu FER-2013 (kích thước ảnh, số lượng mẫu, số lượng nhãn, số lượng mẫu trong tập huấn luyện / kiểm tra, ...).
- Tìm hiểu và thử nghiệm một số phương pháp rút trích đặc trưng low-level và high-level.
- Tìm hiểu ý tưởng, cách hoạt động của một số thuật toán phân lớp: Logistic Regression, Random Forest, SVM.
- Xây dựng chương trình từ pipeline: Rút trích đặc trưng $\rightarrow$ Huấn luyện $\rightarrow$ Kiểm tra $\rightarrow$ So sánh, đánh giá trên tập dữ liệu FER-2013 sử dụng độ đo Precision, Recall, F1.
- Xây dựng giao diện web để trực quan hóa kết quả.
- Viết báo cáo đề tài

Chương 2

Cơ sở lý thuyết

2.1 Rút trích đặc trưng

Rút đặc trưng là quá trình thực hiện giảm số chiều dữ liệu, kết quả của quá trình này là những "đặc trưng", hay "dữ liệu" đã được rút gọn, mô tả tốt thông tin của dữ liệu ban đầu.

2.1.1 Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [3] là một giải thuật phát hiện đặc trưng trong Thị giác máy tính. Các đặc trưng local như điểm, cạnh trong bức ảnh.
Giải thuật SIFT bao gồm các bước:

Scale-space extrema detection

SIFT tính DoG (Difference of Gaussians) trên từng pixel bằng cách lấy diff của Gaussian Blur với 2 sigma khác nhau. Sau khi tính được DoG của toàn ảnh, xét trên từng pixel so sánh với 8 neighbors và 9 pixels tương ứng của scale ảnh ngay trên và 9 pixels tương ứng ở scale dưới, nếu pixel đó là local extrema (lớn nhất) thì nó sẽ được coi như là 1 ứng viên keypoint ở scale đó.

Keypoint localization

Bước Scale-space extrema detection tạo ra rất nhiều ứng viên có thể là keypoint, một số trong đó không ổn định. Bước keypoint localization này sẽ thực hiện fit vào phần dữ liệu lân cận. Việc này giúp loại bỏ các điểm có độ tương phản thấp, hay nhạy cảm với nhiễu.

Orientation assignment

Tại bước này, mỗi keypoint sẽ đươc gán một hoặc nhiều hướng dựa trên hướng của các gradient. Đây là bước tối quan trọng trong việc đạt được tính bất biến với phép xoay ảnh, vì descriptor sẽ được biểu diễn theo các hướng này.

Keypoint descriptor

Các vector descriptor cho các keypoint sẽ được tính tại bước này sao cho các descriptor cho mỗi keypoint là rất khác nhau. Thao tác này được thực hiện với bức ảnh có scale gần nhất với scale của keypoint.

Một vùng lân cận kích thước 16x16 xung quanh keypoint được thiết lập, chia thành 16 khối 4x4. Mỗi khối 4x4 chứa histogram mô tả 8 hướng gradient, tổng cộng là 128 giá trị tạo nên một vector keypoint descriptor.

2.1.2 EigenFace

Eigenface là tập các vector riêng từ ma trận hiệp phương sai của phân bố xác suất của không gian vector biểu diễn ảnh khuôn mặt, giúp làm giảm số chiều bằng cách dùng tập các bức ảnh cơ sở để biểu diễn các bức ảnh gốc. Ta có thể thực hiện phân lớp bằng cách so sánh các khuôn mặt được biểu diễn bởi các ảnh cơ sở này.

2.1.3 Bag of Visual Words

Ý tưởng của phương pháp này cũng giống như Bag of Words khi phân tích nội dung các đoạn văn bản, khi các vector đặc trưng của bức ảnh sẽ đóng vai trò như các keyword trong văn bản. Tuy nhiên, khác với văn bản, không có một tiêu chuẩn rõ ràng để đếm tần suất xuất hiện của các vector đặc trưng có trong ảnh, vì không có vector đặc trưng nào giống nhau hoàn toàn.

Vậy nên, vấn đề ở đây chỉ là phải gom nhóm các vector đặc trưng vào các cụm phù hợp, với các cluster center được gọi là Visual Keyword. Lúc này, ta có thể biểu diễn các bức ảnh dưới dạng histogram tần suất xuất hiện của các Visual Keyword. Từ đây, ta có thể thực hiện phân lớp dựa trên các vector thể hiện các histogram này.

2.2 Mô hình phân lớp

2.2.1 Logistic Regression

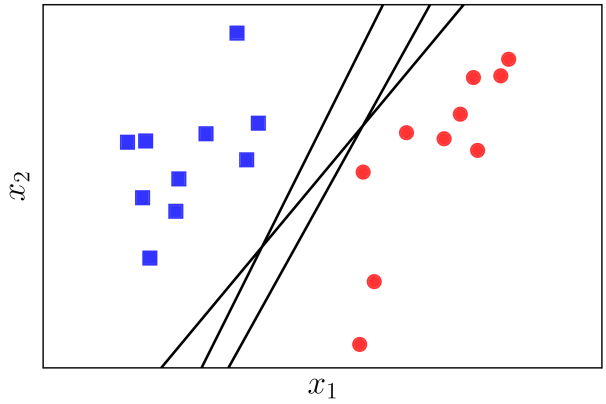
Định nghĩa

Logistic Regression được sử dụng nhiều cho các bài toán Classification. Logistic Regression cho đầu ra ở dạng xác suất (probability), là số thực và bị chặn trong đoạn $[0, 1]$.
Mô hình của Logistic Regression được thể hiện ở công thức 2.1

$$f(x) = \theta(w^T x) \tag{2.1}$$

Trong đó θ là logistic function.
Một số các hàm activation được sử dụng là sigmoid (2.2) và tanh (2.3)

$$f(s) = \frac{1}{1 + e^{-s}} \tag{2.2}$$



HÌNH 2.1: Các mặt phân cách hai lớp có thể phân tách tuyến tính [6]

$$tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \tag{2.3}$$

Hàm mất mát được thể hiện ở công thức 2.4

$$J(w) = -\sum_{i=1}^{N}(y_i log z_i + (1 - y_i) log(1 - z_i)) \tag{2.4}$$

2.2.2 Support Vector Machines

Định nghĩa

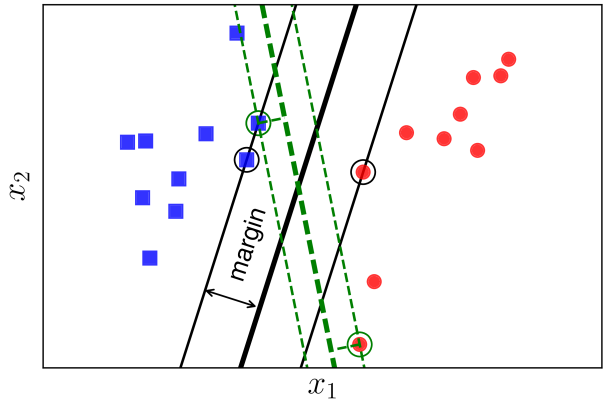
Support Vector Machines (SVM) (hiện tại mô hình sử dụng soft margin được đề xuất ở [1] được sử dụng nhiều) là một phương pháp phân lớp tuyến tính (linear classifier), với mục đích xác định một siêu phẳng (hyperplane) để phân tách hai lớp của dữ liệu.

Support vectors

Trong hình 2.2, các điểm xanh, đỏ phân bổ trên 2 đường biên được gọi là các support vector (nhiệm vụ của chúng là support để tìm ra siêu phẳng)

Margin

Margin được xem là khoảng cách từ 2 điểm dữ liệu đến siêu phẳng. SVM tìm cách tối ưu hóa giá trị margin này.
Về mặt hình học, margin được tính bằng công thức $\frac{2}{\|w\|}$ ở không gian nhiều chiều. Margin cực đại được giải bằng bài toán đối ngẫu Lagrange.



HÌNH 2.2: Margin của hai lớp là bằng nhau và lớn nhất có thể.
[7]

Soft margin

Trong những trường hợp dữ liệu nhiễu hoặc không hoàn toàn có thể phân tách tuyến tính thì không thể tìm được Hard Margin, ta cần tìm Soft margin. Từ đó tham số C được sử dụng với qui ước:

- $C = \infty$: Hard margin
- C lớn: sai lệch nhỏ, thu được margin nhỏ
- C nhỏ: sai lệch lớn, thu được margin lớn

2.2.3 Random Forest

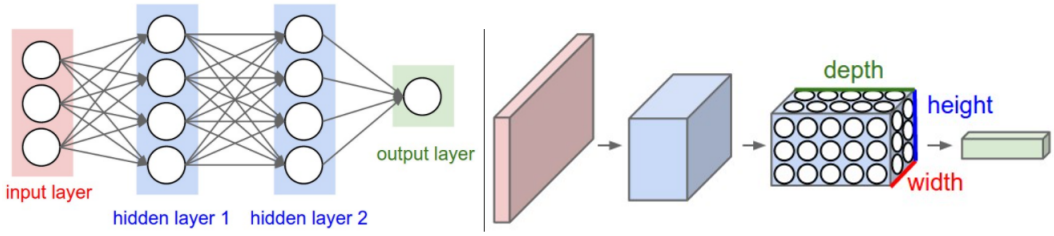
Định nghĩa

Random Decision Forests [4] hoặc Random Forest là một meta estimator được xây dựng từ 1 nhóm các cây quyết định. Dữ liệu để huấn luyện được chọn từ các sub-sample từ tập dữ liệu gốc ban đầu và sử dụng phương pháp trung bình để cải thiện độ chính xác dự đoán của mô hình và hạn chế over-fitting.

Xây dựng

Mỗi cây được xây dựng dựa trên thuật toán sau:

- Số mẫu là N, số variable là M.
- Chọn n mẫu có hoàn lại cho training. Dùng $N - n$ mẫu còn lại để tính error của cây.
- Ở mỗi node, chọn ngẫu nhiên m variable. Tính the best split dựa trên m variable này trên tập training set.
- Cây được xây dựng như việc tạo một cây quyết định bình thường.



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

HÌNH 2.3: Mạng nơ-ron thông thường (trái) và CNN (phải) [1]

2.3 Phân cụm dữ liệu

2.3.1 KMeans

Đối với phân cụm K-means, nhãn của từng điểm dữ liệu là không có. Thuật toán sẽ tìm cách phân dữ liệu thành những cụm mà trong đó các điểm ở cùng một cụm có sự tương đồng về tính chất. K-means hoạt động tốt với lượng mẫu lớn, được sử dụng trong nhiều ứng dụng của nhiều lĩnh vực.

2.4 Mạng Nơ-ron tích chập (Convolutional neural network)

Mạng nơ-ron thần kinh nhân tạo (ANN - Artifical Neural Networks) mô phỏng cấu trúc mạng nơ-ron sinh học của con người. Mạng được cấu thành bởi các đơn vị tính toán đơn giản được liên kết với nhau mà ở đó chức năng của mạng được quyết định bởi các liên kết giữa các nơ-ron.

Mạng Nơ-ron tích chập (CNN - Convolutional Neural Network) được phát triển dựa trên ý tưởng của mạng ANN. Kiến trúc mạng CNN được thể hiện ở hình 2.3). Ở layer đầu tiên có 3 chiều cao, rộng và sâu, là nơi thường được dùng để đưa ảnh RGB thô vào. Tiếp theo các nơ-ron ở các lớp ẩn không liên kết hoàn toàn với tất cả nơ-ron ở lớp tiếp theo đó như ở mạng ANN mà liên kết tới một khu vực nhỏ các nơ-ron. Cuối cùng, lớp đầu ra là môt vector chứa các giá trị xác suất.

2.5 Độ đo

- **Accuracy**

  Accuracy, hay độ chính xác, thể hiện tỷ lệ các dự báo đúng trên tổng số các dự báo.

[1] Ảnh lấy từ cs231n.github.io/convolutional-networks/

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.5}$$

Tuy nhiên hạn chế của nó là đo lường trên tất cả các nhãn mà không quan tâm đến độ chính xác trên từng nhãn. Với dữ liệu có sự mất cân bằng giữa các lớp thì độ đo này không phù hợp.

- **Precision**

$$Precision = \frac{tp}{tp + fp} \tag{2.6}$$

Precision thể hiện độ chính xác của việc dự đoán các mẫu positive, hữu ích trong các trường hợp khi dự đoán sai các mẫu positive ảnh hưởng nghiêm trọng đến mục tiêu bài toán.

- **Recall**

$$Recall = \frac{tp}{tp + fn} \tag{2.7}$$

Độ đo này thể hiện rằng bao nhiêu mẫu positive thực tế được xác định đúng. Metric này dùng để đánh giá 1 model khi mà việc dự đoán sai 1 mẫu positive thực tế là rất nguy hiểm.

- **F1**

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.8}$$

Độ đo này là trung bình điều hòa giữa 2 độ đo Precision và Recall. Trong phần lớn ứng dụng thực tế, dữ liệu có thể rất hay mất cân bằng. Chính vì vậy, F1 là một độ đo khá hữu ích để đánh giá mô hình học máy.

Chương 3

Phương pháp thực hiện

Phương pháp chúng tôi đề xuất gồm hai phần chính là rút trích đặc trưng và phân lớp dữ liệu. Ảnh đầu vào sau khi tiền xử lí sẽ được rút trích đặc trưng, sau đó những đặc trưng này qua một mô hình phân lớp để phân loại cảm xúc đầu ra. Hình 3.1 minh họa hệ thống nhận nhiện cảm xúc khuôn mặt chúng tôi đề xuất.

3.1 Tiền xử lý dữ liệu

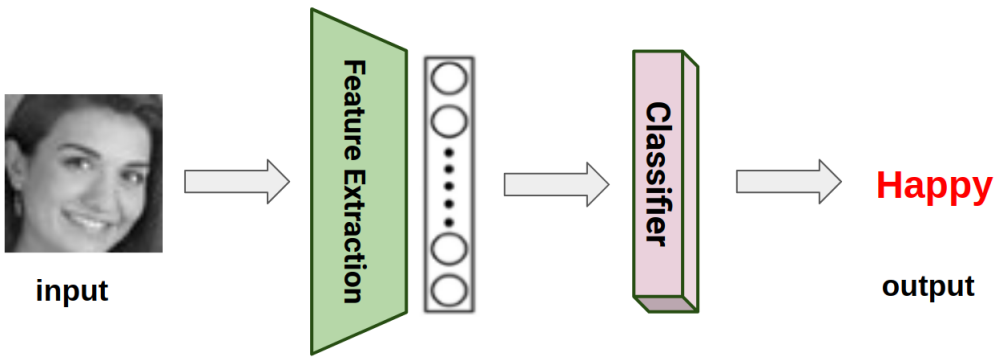
Bởi vì đầu vào là ảnh xám với giá trị các pixel từ 0 đến 255 nên khi ảnh chuyền qua hàm sigmoid hoặc ReLu giá trị trả ra luôn rất lớn. Để tránh bùng nổ gradient dẫn đến bước cập nhật trọng số "kinh khủng", các ảnh đầu vào sẽ được chuẩn hóa về miền giá trị $[0, 1]$.

3.2 Rút trích đặc trưng

3.2.1 SIFT và Bag of Visual Words

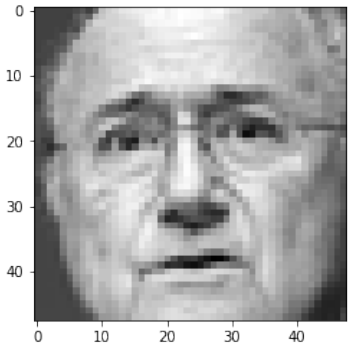
Ảnh đầu vào với kích thước $48x48$ sau khi dùng SIFT để rút trích đặc trưng thì ta thu được k keypoints và d descriptors (kích thước 128 chiều).

Ứng với mỗi ảnh ta có tập descriptors D tương ứng. Từ tập hợp tất cả các tập (n, D) lấy từ tập ảnh đầu vào, chúng tôi sử dụng KMeans để gom thành m nhóm, trọng tâm của mỗi nhóm là tâm của từng túi trong Bag of Visual Word.

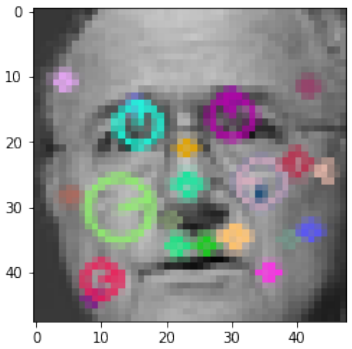


HÌNH 3.1: Pipeline nhận diện cảm xúc khuôn mặt.

Kết quả thu được một ma trận nxm n số lượng ảnh đầu vào, m số lượng thuộc tính, là tập đặc trưng đầu vào, Hình 3.2 thể hiện ảnh đầu vào và Hình 3.3 thể hiện 25 keypoint được rút trích đặc trưng bằng SIFT.



HÌNH 3.2: Ảnh đầu vào



HÌNH 3.3: 25 keypoints SIFT

3.2.2 EigenFace

Eigenface [5] là những ảnh mà sau khi thêm thông tin khuôn mặt trung bình thì sẽ tạo được một khuôn mặt mới. Minh họa bởi công thức 3.1

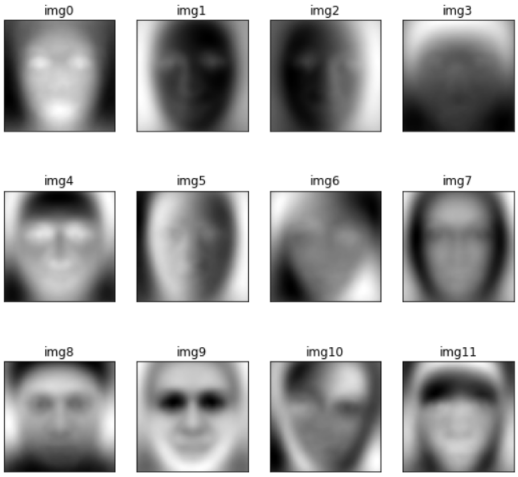
$$F_n = F_m + \sum_{i=1}^{n} \alpha_i F_i \tag{3.1}$$

Trong đó:
F_n : Khuôn mặt mới
F_m : Khuôn mặt trung bình
F_i : EigenFace thứ i
α_i : Hệ số của EigenFace thứ i

Các EigenFace được tính toán bằng cách sử dụng những thành phần chính của ảnh trong tập dữ liệu.Trong đề tài này, chúng tôi sử dụng phương pháp PCA để rút trích các thành phần chính. Chi tiết thực hiện như sau:

- mỗi ảnh đầu vào có kích thước $48x48$ được duỗi thành 1 vector 2304 chiều. Kết quả sau khi duỗi n ảnh đầu vào là một ma trận kích thước $(n, 2304)$.
- Tính toán vector trung bình của toàn bộ ảnh đầu vào. sau đó toàn bộ ảnh đầu vào sẽ trừ đi cho vector trung bình đó.
- Từ kết quả sau khi trừ cho vector trung bình. Chúng tôi sử dụng PCA với số lượng thành phần chính là 150 để rút trích đặc trưng, EigenFace.

Hình 3.4 minh họa 12 trong tổng số 150 eigenfaces.



HÌNH 3.4: Ví dụ 12 eigenfaces

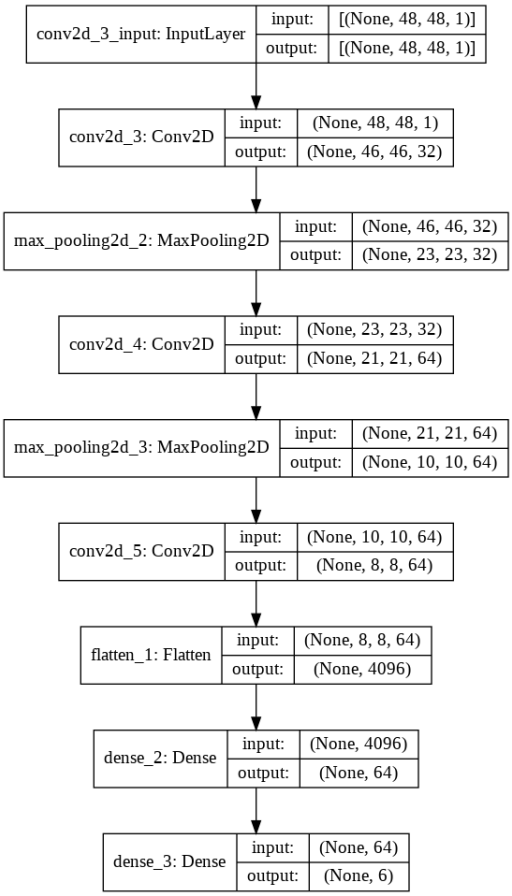
3.2.3 Đặc trưng học sâu

Chúng tôi sử dụng một mạng CNN với cấu trúc như hình 2.3 để rút trích những đặc trưng học sâu. Các đặc trưng này được lấy ra từ đầu ra của lớp Flaten với số chiều là 4096. Với đầu vào là một tensor $(n, 48, 48, 1)$ sau khi qua CNN cho đầu ra là đặc trưng $(n, 4096)$. Hình 3.6 minh họa mô hình CNN được sử dụng.

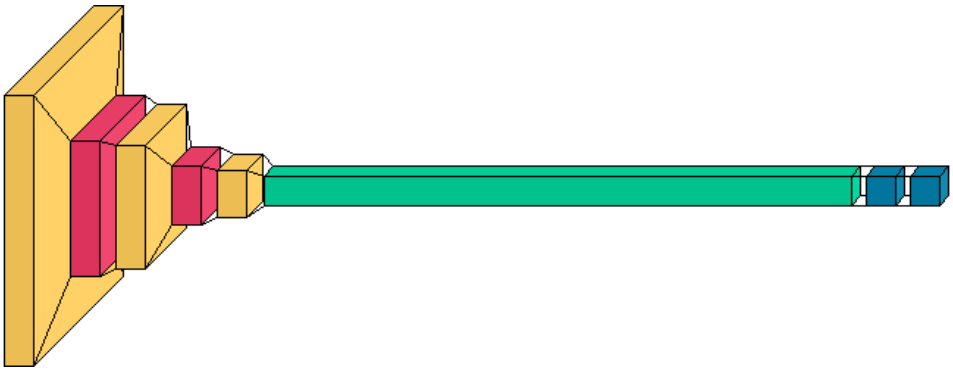
3.3 Phân lớp

Từ những đặc trưng được rút trích, chúng tôi tiến hành thử nghiệm những phương pháp phân lớp khác nhau để phân loại những đặc trưng đó. Từ đó nhận diện được cảm xúc khuôn mặt. Những phương pháp phân lớp được sử dụng:

- Logistic Regression
- Support Vector Machine
- Random Forest



HÌNH 3.5: Cấu trúc mô hình CNN được sử dụng



HÌNH 3.6: Minh họa mô hình CNN được sử dụng

Chương 4

Kết quả thực nghiệm

4.1 Bộ dữ liệu FER-2013

Bộ dữ liệu Facial Expression Recognition 2013 (FER-2013)[2] bao gồm 35,887 ảnh xám của khuôn mặt.

- Kích thước mỗi ảnh: 48x48
- Kích thước tập ảnh train: 28,709
- Kích thước tập ảnh validation: 3,589
- Kích thước tập ảnh test: 3,589
- Số lượng lớp: 7(0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral) minh họa ở Hình 4.1

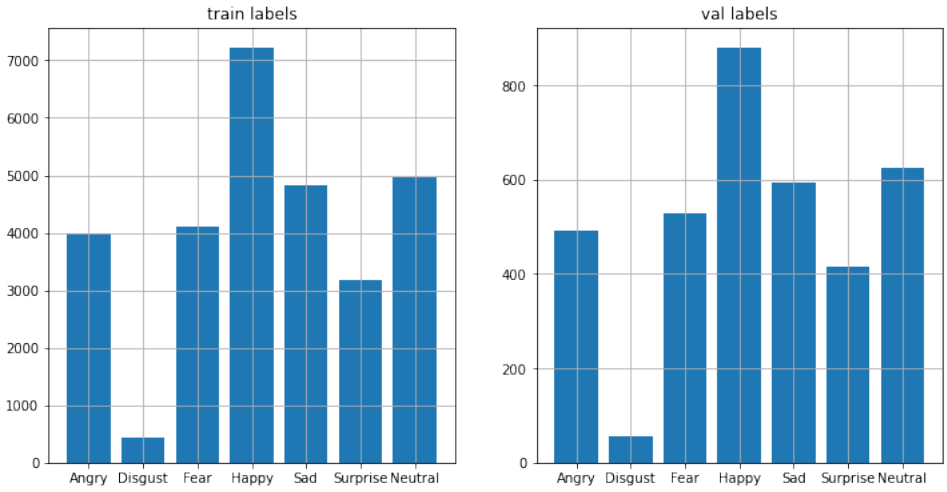
4.2 Các thực nghiệm và kết quả

4.2.1 Tiền xử lí bộ dữ liệu

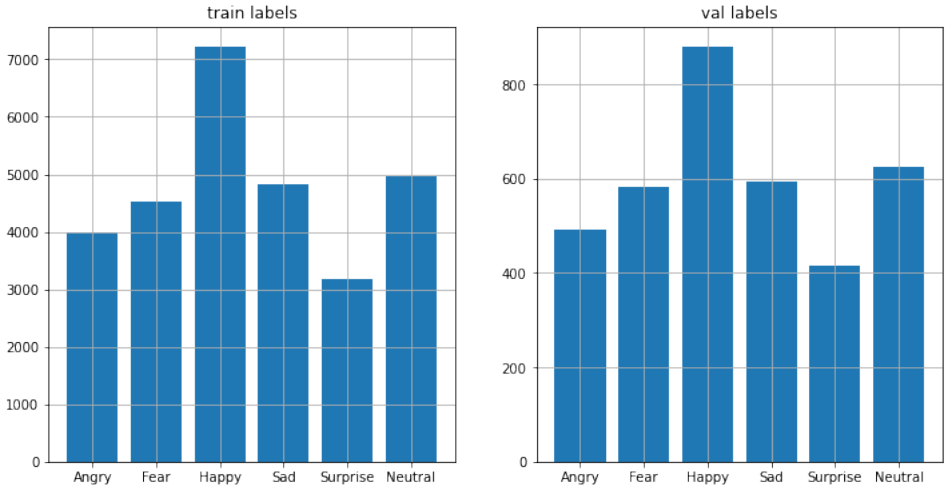
Hình 4.2 mô tả phân bố dữ liệu của các lớp của tập train và validation.

Từ phân bố trên có thể thấy dữ liệu phân bố của bộ dữ liệu không đồng đều. Lớp cao nhất (Happy) có 8989 mẫu, trong khi đó lớp Disgust chỉ có 547 mẫu. Bên cạnh đó, biểu cảm của Disgust với Fear rất giống nhau, dễ gây nhầm lẫn ngay cả với người. Vì những lí do trên chúng tôi quyết định gộp 2 lớp trên thành lớp Fear để giải quyết vấn đề dữ liệu mất cân bằng. Bài toán lúc này là phân lớp 6 cảm xúc khuôn mặt (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)

HÌNH 4.1: Một số ảnh của từng lớp



HÌNH 4.2: Phân bố dữ liệu trên tập train, validation



HÌNH 4.3: Phân bố dữ liệu sau khi gộp Disgust vào Fear

n clusters	KMeans	MiniBatchKMeans
150	3328.66s	5.18s

BẢNG 4.1: Thời gian gom nhóm K-means và Mini-Batch K-Means

4.2.2 Các thực nghiệm

Tập dữ liệu huấn luyện của chúng tôi bao gồm tập train và val của bộ dữ liệu FER-2013.

Tổng số dữ liệu huấn luyện: 32298

Tổng số dữ liệu kiểm tra: 3589

Những thực nghiệm đầu tiên chúng tôi sử dụng đặc trưng SIFT kết hợp với thuật toán phân lớp Logistic regression, SVM, Random forest.

- **Sử dụng đặc trưng SIFT tạo BoW với 68 cụm và phân lớp bằng Logistic regession:** Sau khi rút trích đặc trưng ta thu được một ma trận $X(32298, 68)$. Ma trận X được phân lớp bằng Logistic Regression với tham số C = 1 và C = 0.00358 (dùng GridSearchCV). Kết quả accuracy trên tập huấn luyện/kiểm tra tương ứng là 0.2796/0.2784 và 0.2779/0.2744.

  Nhận xét: Mô hình chưa hội tụ $\rightarrow$ tăng thêm thuộc tính cho X

- **Sử dụng đặc trưng SIFT tạo BoW với 150 cụm và phân lớp bằng Logistic regession:** Ma trận X được phân lớp bằng Logistic Regression với tham số C=1 và C=0.00278 (dùng GridSearchCV). Kết quả accuracy trên tập huấn luyện/kiểm tra tương ứng là 0.3102/0.3098 và 0.2984/0.299.

  Nhận xét: Mô hình cho kết quả tốt hơn so với 68 thuộc tính, tuy nhiên mô hình vẫn chưa hội tụ và thời gian gom cụm tương đối lâu (3328.66s) $\rightarrow$ thay đổi thuật toán phân lớp thành SVM và sử dụng MiniBatchKMeans thay cho Kmeans truyền thống.

- **Sử dụng đặc trưng SIFT tạo BoW với 150 cụm (sử dụng MiniBatchKmeans) và phân lớp bằng Logistic regession:** Ma trận X được phân lớp bằng Logistic Regression với tham số C=1 và C=0.000077 (dùng GridSearchCV). Kết quả accuracy trên tập huấn luyện/kiểm tra tương ứng là 0.2507/0.3098 và 0.251/0.2521.

  Nhận xét: Thời gian gom cụm của MiniBatchKmeans nhanh hơn 832 lần so với Kmeans (5.18s/3328.66s), kết quả thể hiện ở bảng 4.1. Tuy nhiên độ chính xác thấp hơn Kmeans $\rightarrow$ sử dụng Kmeans truyền thống để gom cụm.

- **Sử dụng đặc trưng SIFT tạo BoW với 150 cụm và phân lớp bằng SVM:** Ma trận X được phân lớp bằng SVM với tham số C=30 và C=1. Kết quả accuracy trên tập huấn luyện/kiểm tra tương ứng là 0.9978/0.3098 và 0.6823/0.2521.

Nhận xét: Kết quả accuracy trên tập kiểm tra cao với tham số C=30 cao hơn so với C=1 một khoảng 0.0046. Và cao hơn so với sử dụng Logistic Regression một khoảng 0.0577. Tuy nhiên mô hình bị overfitting (accuracy trên tập huấn luyện 0.9978) → Thay đổi thuật toán phân lớp thành Random forest.

- **Sử dụng đặc trưng SIFT tạo BoW với 150 cụm và phân lớp bằng Random Forest:** Ma trận X được phân lớp bằng Random forest với chiều sâu là 20 và 47 (dùng GridSearch). Kết quả accuracy trên tập huấn luyện/kiểm tra tương ứng là 0.9031/0.3491 và 0.9977/0.3691.

  Nhận xét: Kết quả accuracy trên tập kiểm tra với chiều sâu 47cao hơn so với SVM một khoảng 0.0124.Tuy nhiên hiện tượng overfitting vẫn xảy ra → Đặc trưng SIFT không phù hợp, thử nghiệm trên đặc trưng EigenFace.

- **Sử dụng đặc trưng EigenFace với số lượng thành phần chính bằng 150 và phân lớp bằng SVM:** Ma trận X đầu vào được rút trích đặc trưng EigenFace, sau đó qua mô hình phân lớp SVM với tham số C=1 và C=7 (dùng GridSearchCV) và RandomForest với chiều sâu bằng 20. Kết quả accuracy trên tập huấn luyện/kiểm tra cao nhất trên SVM tham số C=7 là 0.9868/0.433

  Nhận xét: Đặc trưng EigenFace tốt hơn đặc trưng SIFT trong bài toán nhận diện cảm xúc này. Tuy nhiên độ chính xác không cao. → Sử dụng thêm mô hình CNN.

- **Phân lớp sử dụng mô hình CNN:** Chúng tôi sử dụng mô hình CNN như đã đề cập ở mục 3.2.3 để huấn luyện mô hình với tập train, tập validation và tập test dùng để đánh giá accuracy.

- Số ảnh tập train: 28,709

- Số ảnh tập validation: 3,589

- Số ảnh tập test: 3,589

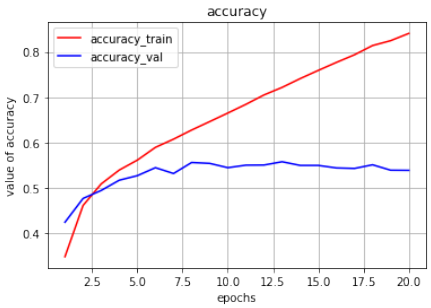
  Tham số huấn luyện mô hình:

  - Learning rate: 1e-3
  - Batch size: 64
  - Epoch: 20
  - Optimizer: SGD, Adam
  - Framework: Keras

  Kết quả huấn luyện mô hình được thể hiện ở hình 4.4 và 4.5:

- **Phân lớp sử dụng mô hình CNN kết hợp với mô hình phân lớp SVM:** Từ mô hình CNN đã huấn luyện ở thực nghiệm trên, chúng tôi bỏ hai lớp cuối cùng để sử dụng mô hình như một bộ rút trích đặc trưng học sâu. Với đầu vào là một tensor kích thước $(n, 48, 48, 1)$, đặc trưng rút

HÌNH 4.4: Độ lỗi trong quá trình huấn luyện



HÌNH 4.5: Độ chính xác trong quá trình huấn luyện

trích được có kích thước $(n, 4096)$. Đặc trưng này được qua mô hình phân lớp SVM với tham số C=1.

Nhận xét: Đặc trưng được rút trích từ CNN cho kết quả tốt hơn các đặc trưng cấp thấp như SIFT và EigenFace. Kết quả đánh giá trên tập huấn luyện của thực nghiệm sử dụng CNN để rút trích đặc trưng và dùng SVM để phân lớp cho kết quả tốt nhất 59.32%

4.2.3 Trực quan hóa kết quả chạy trên tập test

Ban đầu tập dữ liệu FER-2013 là các file csv chứa các giá trị intenssity của các pixel. Sau đó chúng tôi đã chuyển dữ liệu trên thành ảnh và cho mô hình dự đoán.

HÌNH 4.6: Dự đoán đúng biểu cảm Fear

HÌNH 4.7: Dự đoán sai, biểu cảm Sad bị nhầm thành Fear

Với những trường hợp khuôn mặt không bị che khuất (ví dụ hình 4.6), thì mô hình có thể nhận dạng đúng trong đa số các trường hợp . Còn đối với

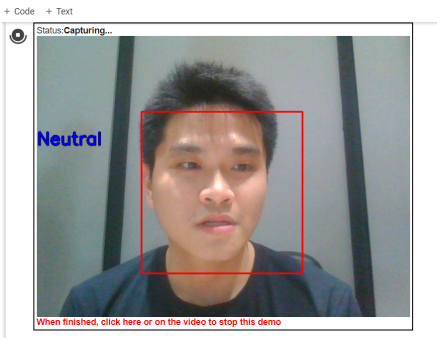
TN	Phương pháp	Thời gian (s)	Train Acc	Test Acc
1	SIFT68 + LogisticRegession_C1	2	0.2796	0.2784
2	SIFT68 + LogisticRegession_C00358	1.69	0.2779	0.2744
3	SIFT150 + LogisticRegession_C1	2.36	0.3102	0.2984
4	SIFT150 + LogisticRegession_C00278	2.24	0.3098	0.299
5	SIFT150mini + LogisticRegession_C1	4.54	0.2507	0.251
6	SIFT150mini + LogisticRegession_C000077	3.8	0.2577	0.2521
7	SIFT150 + SVM_C30	3525.13	0.9978	0.3567
8	SIFT150 + SVM_C1	318.18	0.6823	0.3121
9	SIFT150 + RandomForest_depth20	5.39	0.9031	0.3491
10	SIFT150 + RandomForest_depth47	8.71	0.9977	0.3691
11	EigenFace + SVM_C1	767.73	0.7809	0.4762
12	EigenFace + SVM_C7	2032	0.9868	0.4834
13	EigenFace + RandomForest_depth20	49.2	0.9982	0.433
14	CNN+SGD optimizer	-	-	0.5249
15	CNN+Adam optimizer	67	-	0.5561
16	pretrainCNN+SVM	5105.5	-	**0.5932**

BẢNG 4.2: Kết quả thực nghiệm trên một số đặc trưng và mô hình phân lớp - LEAVE TEST SET ALONE

những ảnh chứa khuôn mặt bị che khuất (ví dụ hình 4.7), ngay cả mắt thường cũng có thể không phân biệt được,thì mô hình thường đưa ra dự đoán sai .

4.3 Demo

Chúng tôi thực hiện một bản demo, nhận hình ảnh trực tiếp từ camera và cho kết quả. Bên dưới là một số hình ảnh được dự đoán từ mô hình của thực nghiệm thứ 14, "pretrainCNN+SVM".



HÌNH 4.8: Dự đoán đúng biểu cảm Neutral

HÌNH 4.9: Dự đoán đúng biểu cảm Surprise

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Từ những mục tiêu đã đề ra, trong đề tài này chúng tôi đã thực hiện được những việc và kết luận như sau:

- Tìm hiểu được và áp dụng một số phương pháp rút trích đặc trưng như SIFT, EigenFace và CNN. Cũng như thử nghiệm được một số phương pháp phân lớp như Logistic regression, SVM, Random Forest.
- Trong bài toán nhận diện cảm xúc khuôn mặt này, các đặc như SIFT hay EigenFace không tốt bằng đặc trưng được rút trích bằng CNN và phương pháp phân lớp SVM là tốt hơn so với hai phương pháp còn lại.
- Xây dựng được một demo nhận diện cảm xúc khuôn mặt.

5.2 Hướng phát triển

Thông qua một số thử nghiệm kết hợp các phương pháp rút trích đặc trưng như SIFT, Eigenface với các mô hình phân lớp như Logistic Regression, SVM, Random Forest, hay các thử nghiệm sử dụng CNN. Chúng tôi đưa ra một số hướng phát triển như sau:

- Tiến hành thử nghiệm các phương pháp state-of-the-art hiện nay nhằm đạt độ chính xác cao.
- Thu thập dữ liệu và thử nghiệm mô hình với các bộ dataset khác.

Bibliography

[1] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018.

[2] *Facial Expression Recognition Kernel Description*. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data. Accessed: 2010-09-30.

[3] David G-Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* (2004).

[4] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[5] Matthew Turk and Alex Pentland. "Eigenfaces for recognition". In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[6] Tiep Vu. *Bài 19: Support VVector Machine*. 2020. URL: https://machinelearningcoban.com/assets/19_svm/svm1.png (visited on 12/25/2019).

[7] Tiep Vu. *Bài 19: Support VVector Machine*. 2020. URL: https://machinelearningcoban.com/assets/19_svm/svm5.png (visited on 12/26/2019).