VINBIGDATA

BÁO CÁO ĐỀ TÀI MÔN HỌC SÂU

NHẬN DIỆN CẢM XÚC KHUÔN MẶT

Học viên
Nguyễn Minh Dũng
Lê Công Pha
Nguyễn Duy Nhất
Võ Minh Tâm

Giảng viên hướng dẫn
TS. Đinh Viết Sang

Ngày 25 tháng 02 năm 2021

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục từ viết tắt

CNN	**C**onvolutional **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
FER-2013	Tập dữ liệu **F**acial **E**xpression **R**ecognition 2013
FER+	Tập dữ liệu **F**acial **E**xpression **R**ecognition Plus
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, ...

Trước đó, ở môn Máy học cơ bản, chúng tôi đã tiến hành thí nghiệm trên bộ dữ liệu FER-2013 với nhiều phương pháp rút trích đặc trưng và các mô hình học máy khác nhau. Kết quả cho thấy mô hình phân lớp SVM kết hợp với bộ rút trích đặc trưng CNN đơn giản cho kết quả tốt nhất trong số các thí nghiệm với độ chính xác 59.32%.

Ở môn học này, chúng tôi quyết định tiếp tục phát triển đề tài bằng việc áp dụng các mô hình và kỹ thuật học sâu cũng như tìm kiếm giải pháp xử lý vấn đề mất cân bằng dữ liệu.

Ngoài ra, chúng tôi đã tạo một web demo nhận diện cảm xúc khuôn mặt, mô phỏng một trợ lý ảo "mini", đáp lại bằng giọng nói trước cảm xúc của người dùng, trước hết để phục vụ cho môn học này, xa hơn là tạo tiền đề để nhóm tiếp tục phát triển ở môn Xử lý ngôn ngữ tự nhiên.

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 lí, 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 những người làm giáo dục đánh giá được mức độ quan tâm của người học, từ đó có thể cải tiến và nâng cao chất lượng bài giảng.

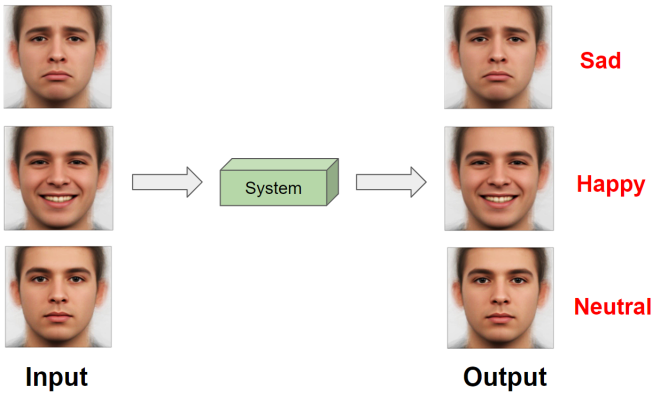
1.2 Giới thiệu bài toán

Input: Một tấm ảnh khuôn mặt người.

Output: Cảm xúc của khuôn mặt đó (vui, buồn, giận dữ, sợ hãi, ghê tởm, ngạc nhiên, 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 lí.



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

- Tìm hiểu, áp dụng một số mô hình học sâu cho bài toán nhận diện cảm xúc khuôn mặt.
- Xây dựng được chương trình demo.
- 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 và FER+.
- Tìm hiểu một số thuật toán mô hình học sâu có thể áp dụng vào bài toán nhận diện cảm xúc.
- Tìm hiểu và áp dụng mộ số mô hình học sâu nhằm cải thiện kết quả kiểm thử.
- 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

- Xác định đầu vào, đầu ra và ứng dụng của bài toán.
- Tìm hiểu một số thông tin cơ bản về bộ dữ liệu FER-2013 và FER+ (kích thước ảnh, số lượng mẫu, số lượng nhãn ...).
- Tìm hiểu ý tưởng, cách hoạt động của một số mô hình học sâu như ResNet50, EfficientNet, VGGFace, VGGFace2.
- 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 Accuracy.
- Xây dựng giao diện web để trực quan hóa kết quả.
- Viết báo cáo đề tài.

Chương 2

Cơ sở lý thuyết

Trong môn Học máy cơ bản, chúng tôi đã sử dụng các kỹ thuật truyền thống như SIFT, EigenFace, Bag of Visual Words và CNN để rút trích đặc trưng từ ảnh đầu vào. Sau đó, chúng tôi tiếp tục sử dụng các mô hình phân loại truyền thống như Logistic Regression, SVM, Random Forest dự đoán nhãn đầu ra. Ở môn học này, chúng tôi tập trung giải quyết vấn đề mất cân bằng dữ liệu với một số phương pháp như SMOTE, Random Over Sampling, ADASYN và class weight. Ngoài ra, chúng tôi còn sử dụng thêm các pretrained model như VGGFace, VGGFace2, EfficientNet,... để rút trích đặc trưng và phân loại các ảnh đầu vào.

2.1 Mất cân bằng dữ liệu

Hầu hết các bộ dữ liệu trong thực tế thường mất cân bằng. Việc bổ sung thêm dữ liệu là không khả thi vì các vấn đề chi phí, nguồn dữ liệu, độ đồng thuận hay thậm chí là do bản chất của dữ liệu trong thực tế là mất cân bằng. Dữ liệu không cân bằng ảnh hưởng xấu đến chất lượng mô hình. Vì thế, việc nghiên cứu và áp dụng các phương pháp xử lý mất cân bằng dữ liệu là điều cần thiết.

2.1.1 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) [2] Mất cân bằng dữ liệu tạo ra sự thiên vị, trong đó mô hình học máy có xu hướng dự đoán lớp đa số mà bỏ qua lớp thiểu số. Trong kỹ thuật lấy mẫu quá mức cổ điển, dữ liệu thiểu số được sao chép từ tập hợp dữ liệu thiểu số. Mặc dù nó làm tăng số lượng dữ liệu, nhưng nó không cung cấp bất kỳ thông tin hoặc biến thể mới nào cho mô hình học máy.

SMOTE hoạt động bằng cách sử dụng thuật toán KNN để tạo dữ liệu tổng hợp, gồm các bước

- Bước 1: Với mỗi mẫu x trong tập thiểu số A, tìm k láng giềng gần nhất của x bằng cách tính khoảng cách Euclide giữa x và các mẫu còn lại trong A.

- Bước 2: Tỷ lệ lấy mẫu N được thiết lập theo tỷ lệ mất cân bằng. Với mỗi mẫu $x \in A$, chọn ngẫu nhiên N mẫu dữ liệu từ k láng giềng gần nhất và xây dựng tập A_1.

- Bước 3: Với mỗi $x_i \in A_1$, công thức sau được sử dụng để tạo một ví dụ mới: $x' = x + random(0,1) \times \mid x - x_i \mid$ trong đó random(0, 1) đại diện cho số ngẫu nhiên từ 0 đến 1.

Quá trình này được lặp lại cho đến khi thu đủ số mẫu cần tăng cường.

2.1.2 Random Over Sampling

Random Over Sampling là kỹ thuật lấy mẫu đơn giản, tăng số lượng mẫu trong lớp thiểu số bằng cách lấy mẫu ngẫu nhiên có hoàn lại chính các mẫu trong lớp thiểu số này. Điều này chỉ làm các lớp cân bằng về số lượng mẫu nhưng không cung cấp thêm thông tin gì cho việc huấn luyện mô hình. Do đó, mô hình dễ bị overfitting.

2.1.3 ADASYN

ADASYN (Adaptive Synthetic) [6] tương tự như SMOTE nhưng tạo ra số lượng mẫu khác nhau dựa trên sự ước lượng phân bố cục bộ của lớp cần lấy mẫu. Cụ thể, ở những nơi mà mật độ dữ liệu cao thì dữ liệu mới sẽ được tạo ra nhiều hơn. Điều này khá hữu ích khi các outliers thường tập trung ở nơi có mật độ thấp, và do đó giảm được phần nào ảnh hưởng của outliers lên mô hình.

2.1.4 Class weights

Đánh trọng số cho từng lớp trong mô hình phân loại là phương pháp đơn giản để giải quyết tình trạng mất cân bằng giữa các lớp. Trong đó, lớp có số lượng mẫu nhiều hơn sẽ có trọng số thấp hơn, và ngược lại, lớp có số mẫu ít sẽ có trọng số cao hơn. Nhờ vậy, mô hình vẫn có khả năng học tốt, giảm thiên vị cho lớp đa số trong trường hợp không thể tăng cường dữ liệu. Đây là kỹ thuật rất dễ sử dụng vì hầu hết các framework đều hỗ trợ. Trong Tensorflow hay Keras, chỉ cần tính trọng số cho các lớp ở dạng dictionary và truyền dictionary này vào tham số `class_weight` ở layer phân loại.

2.2 Mô hình phân lớp

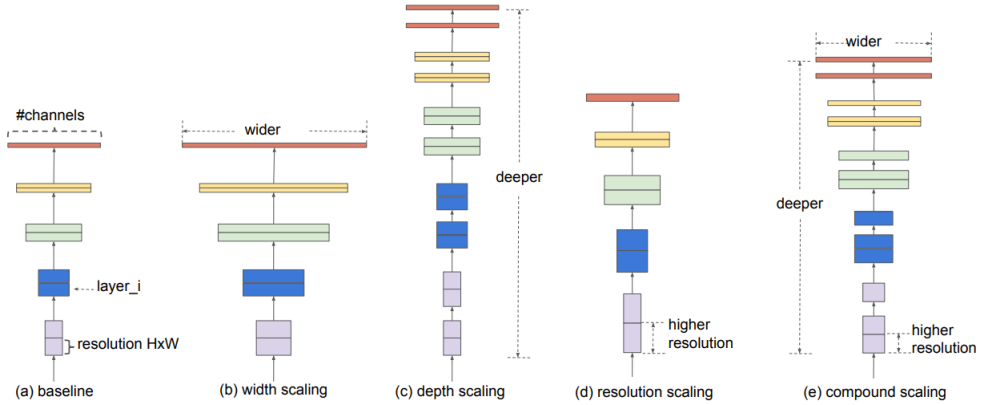
2.2.1 VGGFace

Mô hình VGGFace [9], được đề cập trong Bài báo Deep Face Recognition, 2015, mô tả phương pháp tạo nên một tập dữ liệu cực lớn. Tập dataset này được sử dụng để xây dựng các mạng CNN học sâu giải quyết các bài toán Nhận diện và xác thực khuôn mặt. Bộ phân lớp khuôn mặt sử dụng softmax activation function tại lớp output. Lớp này sau đó được lược đi, để đầu ra của mạng là một vector thể hiện đặc trưng khuôn mặt. Mô hình được train để khoảng cách Euclid giữa các vector của cùng định danh trở nên nhỏ lại, và khoảng cách giữ các vector khác định danh trở nên lớn hơn. Việc này được thực hiện dựa trên hàm mất mát triplet loss.

2.2.2 VGGFace2

VGGFace2 [3] là một tập dữ liệu thậm chí lớn hơn. Dù vậy, tên của nó được dùng để đề cập các mô hình được pretrained trên tập dữ liệu này để nhận diện khuôn mặt. Các mô hình này có backbone là ResNet-50 và SqueezeNet-ResNet-50.

Vector face embedding là một vector có độ dài là 2048. Độ dài của vector embedding được chuẩn hóa về 1 sử dụng L2 vector. Vector này được gọi là face descriptor, được tính dựa trên cosine similarity.



HÌNH 2.1: Model scaling. Ảnh (a) thể hiện kiến trúc một mô hình baseline. Ảnh (b), (c) và (d) thể hiện các cách scale up mô hình truyền thống thông qua width, depth hoặc resolution của mô hình. Ảnh (e) thể hiện phương pháp được đề xuất với việc kết hợp scale up cả 3 giá trị với tỉ lệ cố định [12]

2.2.3 EfficientNet

Phóng to (scale up) ConvNets là phương pháp được sử dụng rộng rãi để tăng độ chính xác của mô hình. Ví dụ, ResNet[7] có thể được scale up từ ResNet-18 thành ResNet-200 bằng cách sử dụng nhiều lớp hơn. Quá trình scale up ConvNets thường không được hiểu rõ ràng. Thông thường, người ta chọn scale up ngẫu nhiên depth hoặc width của các feature maps, hoặc chọn ngẫu nhiên resolution của ảnh đầu vào. Width scaling nghĩa là thêm các feature maps ở mỗi layer, depth scaling là tăng thêm các layers cho mạng, resolution scaling là tăng chất lượng ảnh đầu vào.(Ảnh 2.1). Theo tác giả, việc chỉ tăng một dimension riêng lẻ có giúp tăng độ chính xác, nhưng mô hình sẽ nhanh chóng bão hòa.

Phương pháp compound scaling là ý tưởng chính đằng sau EfficientNet [12]: cân bằng giữa việc tăng width, depth và resolution với một hằng số tỷ lệ. Hằng số tỷ lệ này được quyết định bởi hệ số α, β, γ.
depth: d = α^ϕ
width : $w = \beta^\phi$
resolution : $r = \gamma^\phi$
Với $\alpha \times \beta^2 \times \gamma^2 \approx 2$ và $alpha, beta, gamma \geq 1$.

Các hệ số này được tìm bằng cách cách sử dụng grid search trên một không gian có ràng buộc. Mô hình baseline EfficientNet-B0 có $\alpha = 1.2$, $\beta = 1.1$ và $\gamma = 1.15$. Các tác giả sử dụng các hệ số này như hằng số và scale up mô hình baseline với ϕ khác nhau, từ đó thu được các mô hình EfficientNet B1-B7.

EfficientNet lấy cảm hứng từ MobileNet khi tối ưu hóa mạng để tăng độ chính xác nhưng cũng tăng phạt nếu chi phí tính toán quá cao hoặc thời gian dự đoán lâu (FLOPS). Nếu ảnh đầu vào lớn hơn thì tăng resolution của ảnh -> mạng cần thêm layers (depth) để tăng receptive fields và thêm channels để nắm bắt được nhiều hơn các fine-grain patterns.

Kết quả là, EfficientNet-B7 cho kết quả state-of-the-art 84.3% top-1 accuracy trên ImageNet, trong khi nhỏ hơn 8.4 lần và suy luận nhanh hơn 6.2 lần so với mô hình mạng tốt nhất trước đó.

2.3 Độ đo

- **Accuracy**

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

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.1}$$

  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**

$$Precision = \frac{tp}{tp + fp} \tag{2.2}$$

  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**

$$Recall = \frac{tp}{tp + fn} \tag{2.3}$$

  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**

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

  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 tiền xử lý dữ liệu và phân lớp. Ảnh đầu vào sau khi tiền xử lí sẽ được đưa vào một mạng CNN để phân lớp. 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 $[0, 1]$.

Chúng tôi cũng tiến hành thay đổi kích thước ảnh đầu vào từ (48,48) thành (224,224,3) để phú hợp với đầu vào của mô hình CNN.

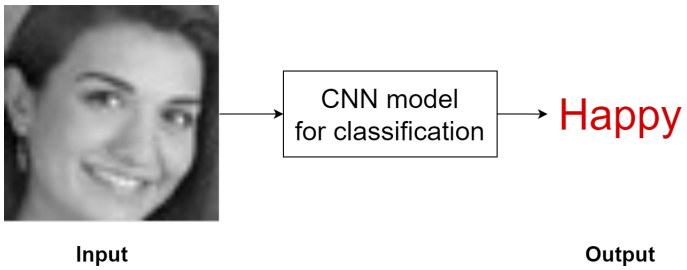
3.2 Làm sạch dữ liệu

FER-2013

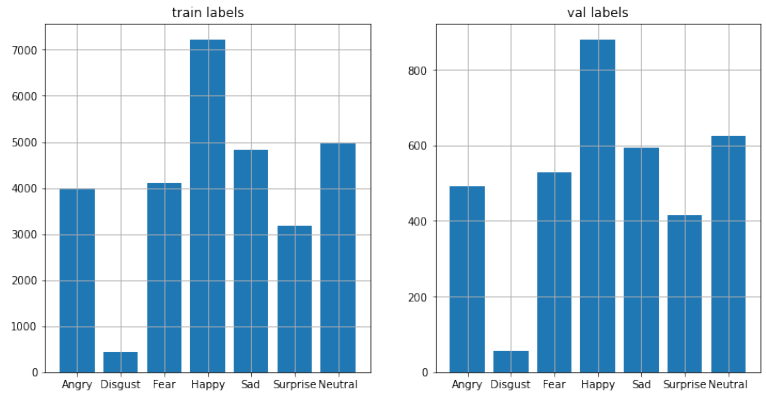
Hình 3.2 mô tả phân bố dữ liệu của các lớp của tập train và validation.

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. Do đó chúng tôi thực hiện việc cân bằng dữ liệu dựa trên các kỹ thuật RandomOverSampling, SMOTE, ADASYN và đặt trọng số khi huấn luyện (class weights).

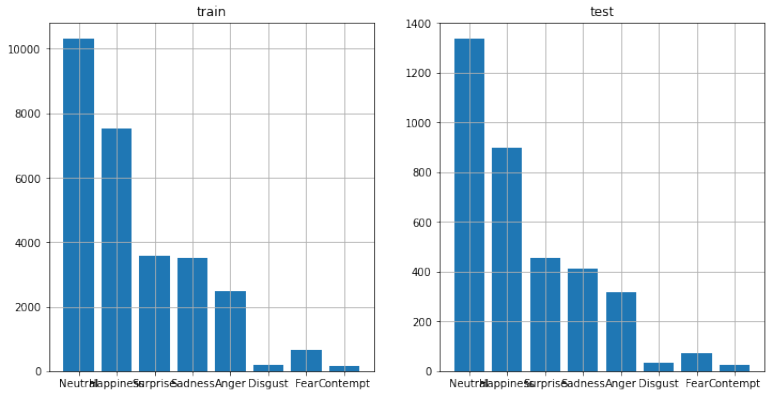
Đồng thời bộ dữ liệu này bao gồm một số ảnh không phải là khuôn mặt. Vì vậy chúng tôi đã tiến hành làm sạch dữ liệu bằng cách loại bỏ những ảnh không phải khuôn mặt.



HÌNH 3.1: Pipeline nhận diện cảm xúc khuôn mặt.



HÌNH 3.2: Phân bố dữ liệu train, valid trên bộ dữ liệu FER-2013



HÌNH 3.3: Phân bố dữ liệu train/test trên bộ dữ liệu FER+

FER+

Hình 3.3 mô tả phân bố dữ liệu của các lớp của tập train và validation.

Bộ dữ liệu FER+ bị mất cân bằng nên chúng tôi sử dụng phương pháp RandomOverSampling để cân bằng lại dữ liệu. Chúng tôi tiến hành huấn luyện mô hình để dự đoán 8 cảm xúc của bộ dữ liệu này: Neutral,Happiness, Surprise, Sadness, Anger, Disgust , Fear, Contempt (loại bỏ lớp unknown và NF)

3.3 Phương pháp thực hiện

Từ dữ liệu đầu vào đã được tiền xử lí, chúng tôi tiến hành huấn luyện với các mô hình CNN: ResNet50[7], VGGFace[9], VGGFace2[3] và EfficientNet[11]. Và tinh chỉnh mô hình với bộ phân lớp là 7 classes (đối với bộ FER-2013) và 8 classes (đối với bộ FER+). Đồng thời áp dụng phương pháp Ensemble learning để cải thiện độ chính xác cho mô hình.

Chương 4

Kết quả thực nghiệm

4.1 Bộ dữ liệu

4.1.1 FER-2013

Bộ dữ liệu Facial Expression Recognition 2013 (FER-2013)[5] 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

4.1.2 FER+

Bộ dữ liệu FER+ [1] chính là tập FER-2013 được gán nhãn lại với nhãn mỗi ảnh được quyết định bởi 10 người. Tập FER-2013 chỉ gán mỗi ảnh với duy nhất 1 lớp, còn tập FER+ sẽ gán cho mỗi ảnh phân bố xác suất của các lớp trên ảnh đó (như trong bảng **??**) . File nhãn mới từ tập FER+ còn bổ sung thêm 3 lớp "contempt", "unknown" và "NF" (Not a Face).
Trong bảng thì các số từ I đến X ứng với các giá trị neutral, happiness, surprise, sadness, anger, disgust, fear, contempt, unknown, NF.

Usage	Image name	I	II	III	IV	V	VI	VII	VIII	IX	X
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
...	...	...	...	...	...	...	...	...	...	...	...

BẢNG 4.1: Nội dung file nhãn dữ liệu của FER+

HÌNH 4.1: Một số ảnh của từng lớp

	FER-2013	FER+
Train	28,709	28,389
Val	3589	3553
Test	3589	3546

BẢNG 4.2: Bảng số lượng ảnh train/val/test của bộ dữ liệu FER-2013 và FER+

4.2 Các thực nghiệm và kết quả

4.2.1 Các thực nghiệm

Chúng tôi tiến hành thuấn luyện và tinh chỉnh bốn mô hình đã đề xuất ở trên cho bài toán nhận diện cảm xúc khuôn mặt trên tập dữ liệu FER-2013 và FER+. Bảng 4.2 thể hiện số lượng ảnh train/val/test.

Những thực nghiệm dưới đây chúng tôi tiến hành huấn luyện trên bộ dữ liệu FER-2013 theo tham số :
- Learning rate: 1e-3, 1e-4, 1e-5
- Batch size: 64
- Epoch: 40
- Framework: Keras

- **ResNet50 với phương pháp RandomOverSampling, SMOTE và class weights:**

  Trước tiên, chúng tôi huấn luyện mô hình Resnet50 từ đầu với với trọng số khởi tạo lấy từ pretrained ImageNet [10] và learning rate 1e-3. Sử dụng các phương pháp cân bằng dữ liệu như RandomOverSampling, SMOTE, class weight.

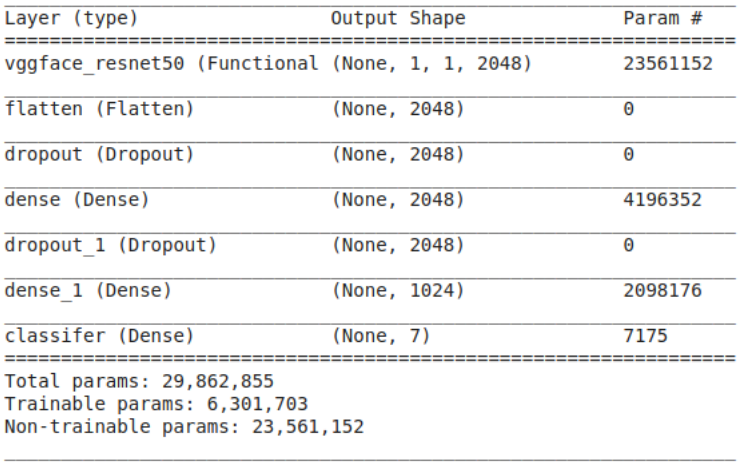
  Sau 40 epoches, kết quả đánh giá của ba mô hình đã huấn luyện được trên tập validation lần lượt là 0.5943, 0.5982 và 0.5690.

  Nhận xét: Kết quả đánh giá của ba mô hình trên tương đương nhau. Tuy nhiên, phương pháp SMOTE tạo ra những hình không phải là khuôn mặt nên dễ gây khó khăn trong quá trình huấn luyện. Vì vậy, chúng tôi quyết định chọn phương pháp RandomOverSampling để giải quyết sự mất cân bàng dữ liệu của tập dữ liệu.

- **VGGFace2 với RandomOverSampling:**

  Dựa theo notebook [4] chúng tôi tiến hành tinh chỉnh mô hình pretrained VGGFace2 bằng cách thay bộ phân lớp khác cho bài toán nhận diện cảm xúc (Hình 4.2)

  Đầu tiên chúng tôi đóng băng trọng số của những lớp CNN và huấn luyện những trọng số của những lớp fully connected với learning rate là 1e-3. Sau đó, chúng



Layer (type)	Output Shape	Param #
vggface_resnet50 (Functional	(None, 1, 1, 2048)	23561152
flatten (Flatten)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 2048)	4196352
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
classifer (Dense)	(None, 7)	7175

Total params: 29,862,855
Trainable params: 6,301,703
Non-trainable params: 23,561,152

HÌNH 4.2: Mô hình VGGFace2 sau khi fine-tune

tôi tiến hành tinh chỉnh toàn mô hình với learning rate 1e-4 và 1e-5. Hàm tối ưu được sử dụng là Adam. Kết quả đánh giá thu được trên tập validation là 0.6761.

Nhận xét: Mô hình được tinh chỉnh từ pretrained VGGFace2 cho kết quả tốt hơn huấn luyện mô hình ResNet50 từ đầu.

- **VGGFace2 với RandomOverSampling trên bộ dữ liệu đã được làm sạch:**

  Bộ dữ liệu FER-2013 86 ảnh không phải là khuôn mặt người. Để dễ huấn luyện mô hình, chúng tôi đã tiến hành làm sạch dữ liệu bằng cách loại bỏ những hình không phải khuôn mặt ra khỏi tập training và tiến hành tinh chỉnh tham số tương tự thực nghiệm trên với hai phương pháo cân bằng dữ liệu là RandomOverSampling và ADASYN

  Kết quả đánh giá thu được trên tập validation lần lượt là 0.7031 và 0.7051.

  Nhận xét: Bộ dữ liệu sau khi được làm sạch cho kết quả huấn luyện tốt hơn, tăng khoảng 3%. Kết quả đánh giá trên tập validation của hai phương pháp RandomOverSampling và ADASYN tương tự nhau. Vì thời gian thực thi ADASYN lâu nên chúng tôi quyết định vẫn giữ phương pháp cân bằng RandomOverSampling cho những thực nghiệm tiếp theo.

- **VGGFace1 với RandomOverSampling:**

  Với tập dữ liệu đã được làm sạch, chúng tôi tiến hành tinh chỉnh mô hình pretrained VGGFace bằng cách thay đổi bộ phân lớp và tinh chỉnh tương tự như thực nghiệm với VGGFace2. Để mô hình hội tụ nhanh, chúng tôi sử dụng hàm tối ưu SGD (learning_rate=1e-3, momentum=0.9, nesterov=True). Hình 4.3 mô tả kiến trúc mô hình sau khi fine-tune.

  Kết quả đánh giá sau hơn 30 epoches của tập validation là 0.7062

  Nhận xét: Kết quả accuracy trên tập validation xấp xỉ với mô hình VGGFace2.

- **Ensemble Learning:**

  Để tăng độ chính xác cho bài toán nhận diện cảm xúc khuôn mặt, chúng tôi kết hợp hai mô hình đã huấn luyện được là VGGFace và VGGFace2 theo công thức 4.1

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vggface_vgg16 (Functional)   (None, 7, 7, 512)         14714688
_________________________________________________________________
global_average_pooling2d_3 ( (None, 512)               0
_________________________________________________________________
batch_normalization_3 (Batch (None, 512)               2048
_________________________________________________________________
dropout_7 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_6 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_8 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_7 (Dense)              (None, 128)               65664
_________________________________________________________________
classifer (Dense)            (None, 7)                 903
=================================================================
Total params: 15,045,959
Trainable params: 330,247
Non-trainable params: 14,715,712
_________________________________________________________________
```

HÌNH 4.3: Mô hình VGGFace1 sau khi fine-tune

TN	Mô hình	Sử dụng	Dataset	Val Acc
1	Resnet50	RandomOverSampler	FER-2013	0.5943
2	Resnet50	SMOTE	FER-2013	0.5982
3	Resnet50	Disgust_Aug+Classweight	FER-2013	0.5690
4	VGGFace2	RandomOverSampler	FER-2013	0.6761
5	VGGFace2	RandomOverSampler	cleaned FER-2013	0.7031
6	VGGFace2	ADASYN	cleaned FER-2013	0.7051
7	VGGFace1	RandomOverSampler	cleaned FER-2013	0.7062
8	EfficientNetB4	RandomOverSampler	cleaned FER-2013	0.5234
9	EfficientNetB3	RAndomOverSampler	cleaned FER-2013	0.5017
10	VGGFace1+2	RandomOverSampler	cleaned FER-2013	**0.7283**

BẢNG 4.3: Kết quả thực nghiệm trên bộ dữ liệu FER-2013

$$predict_{ensemble} = \frac{(predict_{vggface} * w_0 + predict_{vggface2} * w_1}{2} \tag{4.1}$$

Trong đó: $predict_{vggface}$ là xác xuất phân lớp của mô hình VGGFace. $predict_{vggface2}$ là xác xuất phân lớp của mô hình VGGFace2. $predict_{ensemble}$ là xác xuất phân lớp cuối cùng. $w_0 = W_1$: trọng số của mô hình.

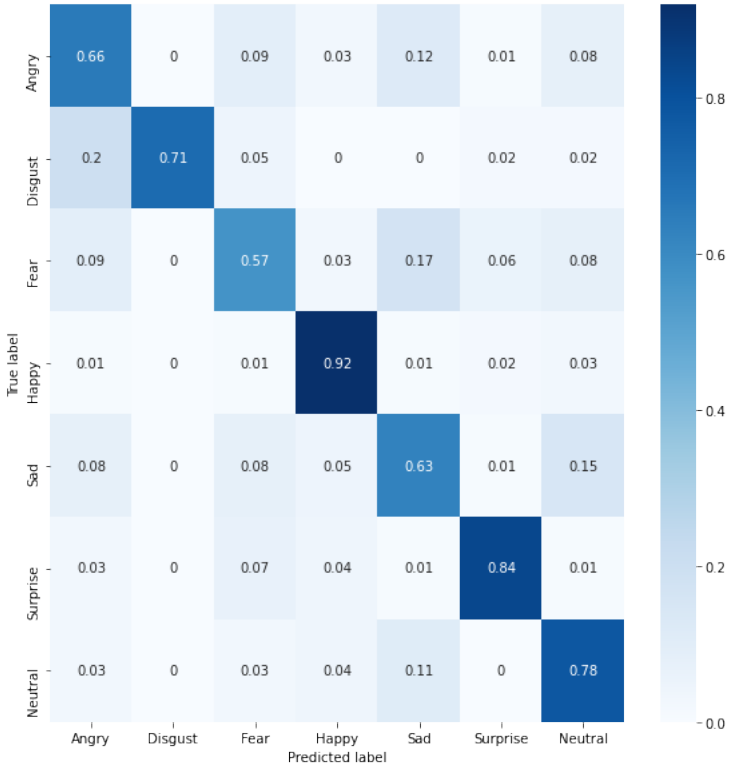
Kết quả Accuracy đánh giá trên tập validation được 0.7283, cao nhất so với các thực nghiệm mà chúng tôi đã tiến hành.

Kết quả đánh giá trên tập test: 0.747

Hình 4.4 thể hiện confusion matrix của các lớp được dự đoán từ mô hình này.

Bảng 4.3 thể hiện kết quả thực nghiệm của chúng tôi trên bộ dữ liệu FER-2013

Tiếp theo, chúng tôi tiến hành tinh chỉnh mô hình ensemble ở trên cho bộ dữ liệu FER+



HÌNH 4.4: Confusion Matrix của mô hình ensemble trên bộ dữ liệu FER-2013

- Finetune VGGFave và VGGFace2: Từ các thực nghiệm trên bộ FER-2013, chúng tôi nhận thấy mô hình VGGFace và VGGFace2 cho kết quả tốt nhất. Vì vậy, chúng tôi tiến hành tinh chỉnh lại từ pretrained VGGFace và VGGFace2 trên bộ dữ liệu FER+ với tham số tương đương.

  Kết quả đánh giá trên tập validation của hai mô hình trên lần lượt là 0.8477, 0.8377.

  Nhận xét: Mô hình VGGFace cho kết quả tốt hơn VGGFace2 trên bộ dữ liệu FER+.

- Ensemble VGGFave và VGGFace2:

  Thực nghiệm cuối cùng chúng tôi kết hợp hai mô hình trên với trọng số bằng nhau theo công thức 4.1

  Kết quả đánh giá trên tập validation: 0.857.

  Kết quả đánh giá trên tập test: 0.8432.

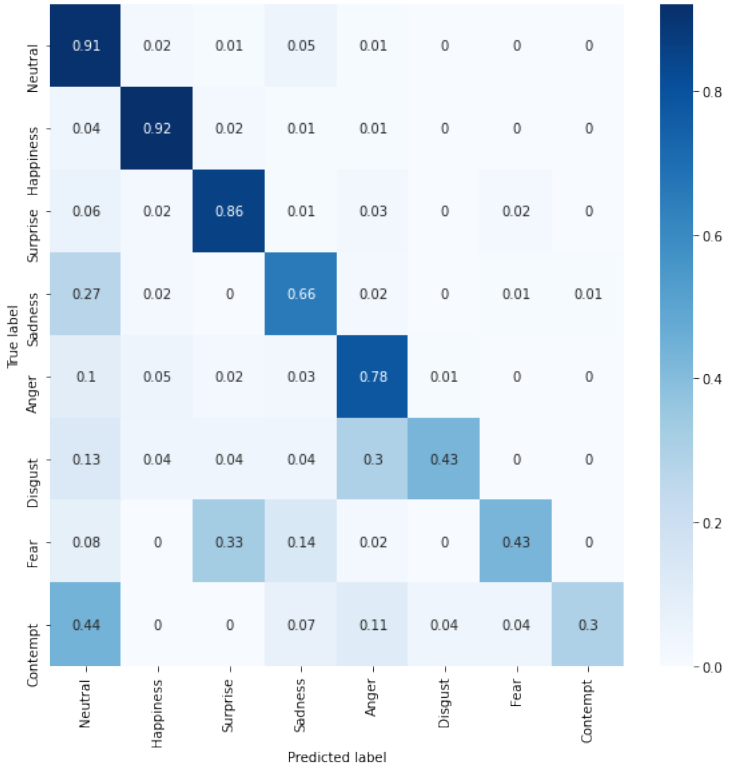
  Nhận xét: Mô hình ensemble learning cho kết quả tốt nhất.

  Hình 4.5 thể hiện confusion matrix của các lớp được dự đoán từ mô hình này.

  Bảng 4.4 thể hiện kết quả thực nghiệm của chúng tôi trên bộ dữ liệu FER-2013

4.3 Demo

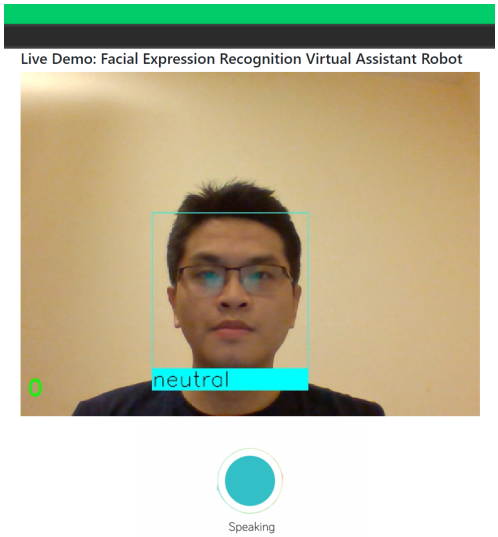
Chúng tôi thực hiện một bản demo sử dụng framework Flask sử dụng mô hình VGGFace được finetune trên bộ dữ liệu FER+.



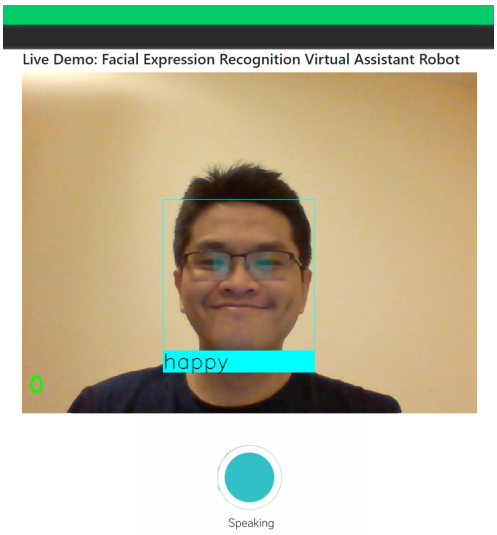
HÌNH 4.5: Confusion Matrix của mô hình ensemble trên bộ dữ liệu FER+

TN	Mô hình	Sử dụng	Dataset	Val Acc
1	VGGFace1	RandomOverSampler	cleaned FER-2013	0.8477
2	VGGFace2	RandomOverSampler	cleaned FER-2013	0.8365
3	VGGFace1+2	RandomOverSampler	cleaned FER-2013	**0.857**

BẢNG 4.4: Kết quả thực nghiệm trên bộ dữ liệu FER+



HÌNH 4.6: Dự đoán đúng biểu cảm Neutral



HÌNH 4.7: Dự đoán đúng biểu cảm Happy

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:

- Xử lý việc mất cân bằng dữ liệu bằng các kỹ thuật Random Over Sampling, SMOTE và ADASYN.
- Sử dụng mô hình pretrain trên dữ liệu khuôn mặt như VGGFace, VGGFace2 cho kết quả tốt hơn mô hình EfficientNet, ResNet50.
- 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 mô hình đã được huấn luyện từ trước cùng với các phương pháp giúp cân bằng giữ liệu, chúng tôi đưa ra một số hướng phát triển như sau:

- Thử các phương pháp như facial landmark alignment, attentional CNN.
- Tự thu thập dữ liệu và thử nghiệm trên các bộ dữ liệu lớn hơn như AffectNet [8].

Bibliography

[1] Emad Barsoum et al. "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution". In: *ACM International Conference on Multimodal Interaction (ICMI)*. 2016.

[2] Kevin W. Bowyer et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *CoRR* abs/1106.1813 (2011). arXiv: 1106.1813. URL: http://arxiv.org/abs/1106.1813.

[3] Qiong Cao et al. "Vggface2: A dataset for recognising faces across pose and age". In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE. 2018, pp. 67–74.

[4] *Challenges in representation learning: facial expression recognition challenge*. 2021. URL: https://www.kaggle.com/kilean/emotion-detection-accuracy70.

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

[6] Haibo He et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 1322–1328. DOI: 10.1109/IJCNN.2008.4633969.

[7] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[8] Ali Mollahosseini, Behzad Hassani, and Mohammad H. Mahoor. "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild". In: *CoRR* abs/1708.03985 (2017). arXiv: 1708.03985. URL: http://arxiv.org/abs/1708.03985.

[9] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. "Deep Face Recognition". In: *British Machine Vision Conference*. 2015.

[10] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[11] Mingxing Tan and Quoc V Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *arXiv preprint arXiv:1905.11946* (2019).

[12] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946.