HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

VÕ THỊ HỒNG NHUNG

PHÂN TÍCH BIỂU CẢM MẶT NGƯỜI DÙNG MẠNG NƠ RON TÍCH CHẬP

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

VÕ THỊ HỒNG NHUNG

PHÂN TÍCH BIỂU CẢM MẶT NGƯỜI DÙNG MẠNG NƠ RON TÍCH CHẬP

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC :
PGS.TS. Lê Hoàng Thái

TP. HỒ CHÍ MINH - NĂM 2022

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: "***Phân tích biểu cảm mặt người dùng mạng nơ ron tích chập***" là công trình nghiên cứu của chính tôi.

Những kết quả nghiên cứu được trình bày trong luận văn là công trình của riêng của tôi dưới sự hướng dẫn của **PGS.TS Lê Hoàng Thái**.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Võ Thị Hồng Nhung

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

Ban Giám hiệu, quý Thầy Cô Khoa Đào tạo sau đại học của Học viện Công nghệ Bưu chính Viễn thông đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi xin chân thành cảm ơn Thầy **PGS.TS Lê Hoàng Thái**, người thầy kính mến đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

MỤC LỤC

DANH SÁCH CÁC THUẬT NGỮ, CÁC CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
CNN	Convolutional Neural Networks	Mạng tích chập
ReLU	Rectified linear unit	Hàm kích hoạt
CK+	Cohn–Kanade dataset	Bộ dữ liệu chuẩn Quốc tế Cohn Kanade
DCNN	Deep Convolutional Neural Network	Mạng tích chập nhiều lớp
ANN	Artificial Neural Network	Mạng nơ ron nhân tạo
GD	Gradient Descent	Kỹ thuật tối ưu
SGD	Stochastic gradient descent	Kỹ thuật tối ưu Stochastic
LBP	Local binary patterns	Mẫu nhị phân cục bộ
JAFFE	Japanese Female facial Expression	Bộ dữ liệu JAFFE

DANH SÁCH CÁC BẢNG

DANH SÁCH HÌNH

I. MỞ ĐẦU

1. Lý do chọn đề tài

Phân loại biểu cảm là lĩnh vực đã được nghiên cứu trong nhiều năm qua với nhiều ứng dụng trong nhiều lĩnh vực khác nhau gắn liền với các hệ thống tương tác người máy. Trong máy học, phân loại biểu cảm là một bài toán khó, tuy nhiên, đối với con người, vấn đề này có thể giải quyết ngay lập tức. Các thách thức chính là: hình ảnh biểu cảm của cùng một người ở cùng một biểu cảm vẫn có thể khác nhau ở những điều kiện ánh sáng, môi trường và góc quay. Những biến đổi này càng lớn khi các đối tượng nghiên cứu càng đa dạng.

- Nhận biết cảm xúc từ nét mặt có một số lợi thế như:
  - Tiếp cận theo hướng tự nhiên nhất để xác định trạng thái cảm xúc của khuôn mặt.
  - Nhiều bộ dữ liệu có sẵn cho biểu hiện cảm xúc trên khuôn mặt.
  - Nhiều công cụ hỗ trợ xác định cảm xúc khuôn mặt có sẵn.
- Nhận biết cảm xúc từ nét mặt cũng có một số nhược điểm như:
  - Không thể cung cấp thông tin ngữ cảnh, do đó đôi khi kết quả bị sai lệch.
  - Kết quả phát hiện cảm xúc phụ thuộc vào chất lượng hình ảnh hoặc video.
  - Chuyển động liên quan đến cảm xúc khuôn mặt có thể được đối tượng cố tình làm giả như các diễn viên …

Vì thế, nhận biết biểu cảm vẫn là một thách thức với thị giác máy tính. Trong luận văn này, đưa ra một hướng tiếp cận đơn giản cho nhận biết biểu cảm khuôn mặt: kết hợp giữa Convolutional Neural Network (CNN) và các bước tiền xử lý đặc trưng. CNN sẽ đạt độ chính xác rất cao nếu học với bộ dữ liệu lớn. Tận dụng ưu điểm này, dự kiến đề xuất phương pháp áp dụng vài kỹ thuật tiền xử lý để chỉ rút trích các thành phần đặc trưng cho biểu cảm trên khuôn mặt và kết hợp với CNNs để thực hiện phân loại cảm xúc hiệu quả. Dự kiến sẽ thực nghiệm đánh giá trên 2 tập dữ liệu công khai lớn (CK+, JAFFE). Các thực nghiệm sẽ được thực hiện để đánh giá các ảnh hưởng

của tiền xử lý và một số ảnh hưởng của các yếu tố khác. Hy vọng xây dựng được hệ thống phân biệt cảm xúc có độ chính xác cao và đáp ứng các yêu cầu về thời gian thực.

2. Tổng quan về vấn đề nghiên cứu

2.1 Phân chia cảm xúc khuôn mặt

- Bảng dưới đây cho biết biểu cảm trên khuôn mặt thể hiện bảy cảm xúc chính của con người [1]:

Bảng 1. 1: Mô tả các cảm xúc cơ bản của con người

Cảm xúc	Biểu cảm khuôn mặt
Vui vẻ	Khóe môi hé mở,Má nâng cao
Buồn bã	Đôi mí mắt trên sụp xuống, mắt mất tập trung, mép kéo nhẹ xuống
Tức giận	Mắt nhìn chằm chằm, Mũi nỡ ra, Môi ép chặt
Sợ hãi	Lông mày nhướng lên, Miệng mở ra
Ghê tởm	Đôi môi được nâng cao lên, Mũi nhăn
Ngạc nhiên	Lông mày cong cao hơn Tròng trắng của mắt rõ hơn, miệng há
Bình thường	Không biểu hiện gì

2.2 Tình hình nghiên cứu

- Các hệ thống FER (facial Expression Recognition) có thể được chia thành hai loại chính dựa trên cách biểu diễn đặc trưng: FER dùng hình ảnh tĩnh và FER chuỗi động.
    - Trong các phương thức dựa trên ảnh tĩnh, biểu diễn đặc trưng được mã hóa chỉ với thông tin không gian từ hình ảnh đơn, trong khi các phương pháp dựa trên chuỗi hình ảnh xem xét mối quan hệ thời gian giữa các khung hình liền kề trong chuỗi biểu diễn đầu vào của khuôn mặt.
- Phần lớn các phương pháp truyền thống đã sử dụng các đặc trưng tìm bằng tay (hand-craft features) hoặc học nông (shallow learning) như : mẫu nhị phân cục bộ (Local Binary Pattern - LBP) [2], LBP trên ba mặt phẳng trực giao (LBP-TOP) [3], hệ số ma trận không âm (NMF) [4] và học thưa [4] cho FER.
- Tuy nhiên, kể từ năm 2013, các cuộc thi nhận biết cảm xúc như FER 2013 [5], và nhận biết cảm xúc trong tự nhiên (EmotiW) [6], đã thu thập dữ liệu huấn luyện tương đối đầy đủ từ các ngữ cảnh khác nhau trong thế giới thực, góp phần thúc đẩy quá trình chuyển đổi FER từ các ngữ cảnh trong phòng thí nghiệm sang các ngữ cảnh thực tế ngoài tự nhiên. Trong khi đó, do khả năng xử lý của bộ vi xử lý tăng đáng kể (ví dụ: Graphics Processing Unit - GPU) và kiến trúc mạng mới góp phần nâng cao tốc độ xử tính toán và độ chính xác trong bài toán xác định biểu cảm của khuôn mặt người.
- Các nghiên cứu trong các lĩnh vực FER đã bắt đầu chuyển sang các phương pháp học sâu, đạt được các kết quả vượt bật, độ chính xác tăng cao và vượt qua các kết quả nghiên cứu trước đó với độ cách biệt lớn [7].

2.3 Một số công trình nghiên cứu đã có

Tác giả Jie Cai [8] đã đề xuất một hàm lỗi mới Island Loss - IL để tăng cường khả năng phân tách các đặc trưng trích xuất bằng phương pháp học sâu. Đặc biệt, IL được thiết kế để giảm phương sai của các cá thể trong cùng một lớp đồng thời mở

rộng sự khác biệt giữa các lớp. Các tác giả thực nghiệm kết quả trên bốn cơ sở dữ liệu chuẩn đã chứng minh rằng CNN (Convolution Neural Network) với hàm lỗi được đề xuất (IL-CNN) vượt trội so với các mô hình CNN cơ bản với truyền thống với hàm lỗi softmax hoặc lỗi trung tâm (Center Loss [9]) và kết quả đạt được có thể so sánh với các phương pháp cho kết tốt nhất (state-of-the-art) trong bài toán xác định biểu cảm khuôn mặt. tác giả đã thực nghiệm trên bộ data CK+ [10], sử dụng ba khung hình cuối cùng tạo thành 981 ảnh, chia làm 10 phần (fold), dùng phương pháp kiểm tra chéo (cross-validation), sử dụng 8 phần cho huấn luyện, 1 phần cho xác thực (validation set) và 1 phần cho kiểm thử (test set) và cho độ chính xác đạt 94.35%. Phương pháp này thuộc phương pháp sử dụng ảnh tĩnh.

Tác giả Yuedong Chen [11] đã đề xuất một mô hình FER mới, được đặt tên là Facial Motion Prior Networks (FMPN). Các tác giả đã thêm một nhánh bổ sung để tạo ra một mặt nạ để tập trung vào các vùng cơ mặt di chuyển. Để học được mặt nạ vùng chuyển động trên khuôn mặt khi biểu cảm, tác giả đã sử dụng dụng sự khác biệt trung bình giữa khuôn mặt trung tính(không biểu cảm) và khuôn mặt biểu cảm tương ứng làm nhãn huấn luyện. Tiến hành thực nghiệm để chứng minh phương pháp của mình, các tác giả đã sử dụng tập CK+ [10] với 3 khung hình cuối được sử dụng, tạo thành 981 ảnh, chia làm 10 phần (fold), dùng phương pháp kiểm tra chéo (cross-validation) và độ chính xác (accuracy) để đánh giá mô hình, các tác giả đã đạt được độ chính xác 98.06%. Phương pháp này thuộc phương pháp sử dụng ảnh tĩnh.

Tác giả Debin Meng [12] và các cộng sự đề xuất mạng: Frame Attention Networks (FAN) để tự động làm nổi bật một số khung hình tách biệt trong một mạng đầu cuối. Mạng nhận vào là một video có số lượng hình ảnh khuôn mặt và biểu diễn lại dưới lại trong một không gian có số chiều cố định. Toàn bộ mạng lưới bao gồm hai phần. Tạo vector đặc trưng: sử dụng mạng CNN cho phần tạo vector đặc trưng (CNN). Học Trọng Số: Phần thứ hai dùng để học trọng số của mỗi khung hình, với mỗi khung hình sẽ có một trong số cho biết mức độ quan trọng của khung hình đó trong việc xác định biểu cảm khuôn mặt, tác giả đã thực nghiệm trên bộ dữ liệu CK+ [10] và sử dụng phương pháp kiểm tra chéo (cross-validation), sử dụng toàn bộ khung

hình có trong tập dữ liệu, chia làm 10 phần (fold), sử dụng độ chính xác (accuracy) để đánh giá mô hình, các tác giả đã đạt được 99.69%, phương pháp này thuộc loại sử dụng chuỗi hình ảnh.

3. Mục đích nghiên cứu

Nghiên cứu đề tài này nhằm mục đích tìm hiểu bài toán nhận biết cảm xúc từ nét mặt, từ đó xây dựng các hệ thống ứng dụng trong thực tiễn như: đánh giá cảm xúc nhân viên trong thời gian làm việc tại công ty, từ đó xác định hiệu quả công việc; hoặc xác định cảm xúc của lái xe đường dài: tạo báo động khi ở trạng thái buồn ngủ (tránh gây ra nguy hiểm).

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: tập trung tìm hiểu một số phương pháp CNN phổ biến hiện nay, xác định một trong bảy trạng thái cảm xúc cơ bản của con người dựa vào hình ảnh đơn nhập vào.

Phạm vi nghiên cứu: thực hiện trên tập dữ liệu chuẩn CK+ [10] và JAFFE, trên hai giới tính nam lẫn giới tính nữ, độ tuổi từ 18 - 45 tuổi, với nhiều chủng tộc người khác nhau. Đồng thời, cũng thử nghiệm trên một số ảnh chụp webcam để minh hoạ tính khả thi của hệ thống về mặt ứng dụng.

Đề xuất cách tiếp cận học sâu kết hợp với các kỹ thuật tiền xử lý như: chuẩn hóa hình ảnh và tăng cường mẫu học bằng các phép rotation, translation và scaling trên ảnh thật (synthetic training-samples generation), với hy vọng nâng cao độ chính xác trên các bộ dữ liệu thử nghiệm đã chọn. Tiến tới, xây dựng một hệ thống phân loại cảm xúc thoả các tiêu chí bên dưới:

- Hiệu suất cao và đáp ứng yêu cầu thời gian thực.
- Giảm tác động của môi trường và giải quyết vấn đề dữ liệu học quá ít (cải tiến khâu tiền xử lý).

- Phân tích đánh giá các Kết quả thử nghiệm để chỉ ra hiệu quả của đề xuất.

5. Phương pháp nghiên cứu

- Phương pháp chuyên gia:
  - Tổng hợp các kiến thức đã biết về các mô hình học sâu – cụ thể là mạng nơ ron tích chập, đưa ra nhận định mô hình nào phù hợp với việc xác định cảm xúc khuôn mặt người và có tốc độ cao.
- Phương pháp thực nghiệm:
  - Thực nghiệm trên tập dữ liệu về cảm xúc khuôn mặt người, đã được gắn nhãn để tìm ra một mô hình cho độ chính xác (accuracy) cao và tốc độ chạy thời gian thực khi xác định cảm xúc của khuôn mặt.
- Phương pháp tổng kết kinh nghiệm:
  - Nghiên cứu và xem xét lại những thành quả thực tiễn đã có của các tác giả đã thực hiện để rút ra kết luận: giúp xây dựng mô hình đạt độ chính xác cao.

6. Dự kiến nội dung của luận văn

Chương 1: Giới thiệu chung

Chương 2: Hệ thống nhận dạng biểu cảm khuôn mặt

Chương 3: Thử nghiệm và thảo luận

Chương 4: Ứng dụng

Chương 5: Kết luận và hướng phát triển

II. NỘI DUNG

CHƯƠNG 1: GIỚI THIỆU CHUNG

1.1 Mạng nơ ron nhân tạo

1.1.1 Giới thiệu mạng nơ ron nhân tạo

Mạng nơ ron nhân tạo (Artificial Neural Network ANN) là một chuỗi các giải thuật lập trình, mô phỏng dựa trên cách hoạt động của mạng lưới thần kinh trong não bộ các sinh vật sống. Mạng nơ ron nhân tạo được sử dụng để tìm ra mối quan hệ của một tập dữ liệu thông qua một thiết kế kiến trúc chứa nhiều tầng ẩn (hidden layer), mỗi tầng lại chứa nhiều nơ ron. Các nơ ron được kết nối với nhau và độ mạnh yếu của các liên kết được biểu hiện qua trọng số liên kết. [13]

Lập trình thông thường có thể làm được rất nhiều phần mềm lớn, như tính toán mô phỏng các vụ nổ hạt nhân trong siêu máy tính ở các phòng thí nghiệm, hoặc tái hiện các tế bào ở cấp độ phân tử để phân tích các thử nghiệm thuốc. Một siêu máy tính có thể tính toán được nhiều tỉ phép tính trên giây, tuy nhiên lập trình thông thường lại gặp khó khăn trong việc nhận ra các mẫu đơn giản, ví dụ như nhận diện mặt người, điều mà một bộ não sinh học xử lý nhanh và chính xác hơn nhiều.

Áp dụng với các kỹ thuật học sâu, mạng nơ ron nhân tạo hiện nay đang được áp dụng để giải quyết những vấn đề mà lập trình theo logic thông thường khó có thể giải quyết được. Do đó, mạng nơ ron nhân tạo đang nhanh chóng trở nên phổ biến, và là xu thế trên nhiều lĩnh vực.

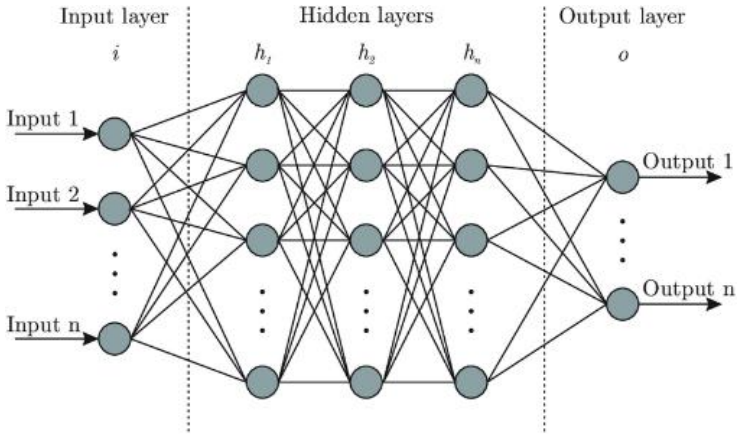
1.1.2 Kiến trúc mạng nơ ron nhân tạo

Một mạng Neural nhân tạo có cấu trúc như sau:

Tầng lớp đầu vào (Input Layer): giá trị các node chính là số lượng đặc trưng của dữ liệu đầu vào khi đưa vào mô hình. Chúng ta thấy giá trị đầu vào là n thuộc tính/đặc trưng. [14]

Tầng lớp ẩn (Hidden Layer): có số node ẩn thường không được xác định, thường do kinh nghiệm của người thiết kế hoặc qua quá trình thử nghiệm nhiều lần mà có được. Tuy nhiên thực tế nếu số lượng node ẩn quá nhiều thì mạng sẽ học chậm, còn nếu số node quá ít thì mạng sẽ không rút trích đủ các thông tin cần thiết trên các đặc trưng. Từ đó hiệu quả của mô hình sẽ không được chính xác. Số lượng các lớp ẩn ở đây có thể một hoặc nhiều lớp ẩn tùy thuộc vào tính chất cũng như độ phức tạp của dữ liệu.

Tầng đầu ra (Output layer): giá trị các số node chính là số lượng nhãn đầu ra mà chúng ta mong muốn. Ví dụ như trong tập dữ liệu của chúng ta có tổng cộng 5 nhãn, thì đầu ra của chúng ta tại lớp này chính là một lớp ẩn với 5 phần tử tương ứng với năm nhãn.



Hình 1. 1: Minh họa về mạng neural nhân tạo

Ngoài ra chúng ta còn một số thông tin liên quan đến mạng trí tuệ nhân tạo như :

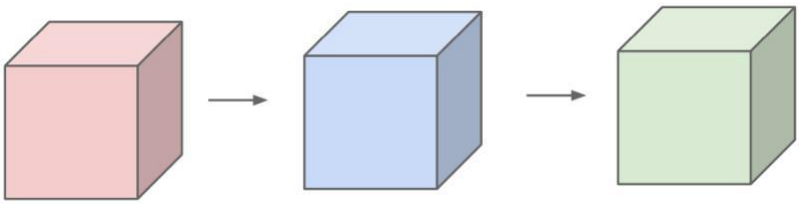
- Hàm tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

- Ngưỡng (còn gọi là độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm tổng.
- Hàm kích hoạt (Activation function): Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi neural. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

1.2 Mạng nơ ron tích chập (Convolutional Neural Networks)

1.2.1 Khái niệm về mạng nơ ron tích chập

Mạng nơ ron tích chập là một trong những mạng truyền thẳng đặc biệt. Mạng nơ ron tích chập là một mô hình học sâu phổ biến và tiên tiến nhất hiện nay. Hầu hết các hệ thống nhận diện và xử lý ảnh hiện nay đều sử dụng mạng nơ ron tích chập vì tốc độ xử lý nhanh và độ chính xác cao. Trong mạng nơ ron truyền thống, các tầng được coi là một chiều, thì trong mạng nơ ron tích chập, các tầng được coi là 3 chiều, gồm: chiều cao, chiều rộng và chiều sâu . Mạng nơ ron tích chập có hai khái niệm quan trọng: kết nối cục bộ và chia sẻ tham số. Những khái niệm này góp phần giảm số lượng trọng số cần được huấn luyện, do đó tăng nhanh được tốc độ tính toán. [14]

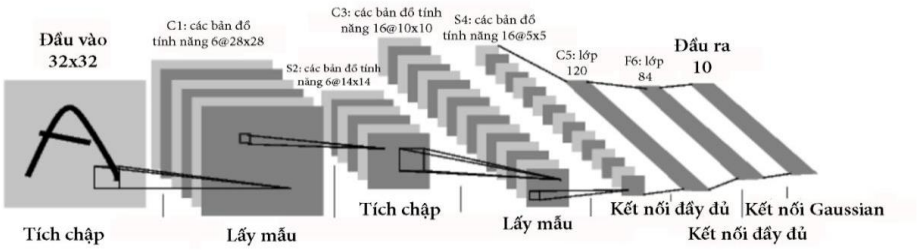


Hình 1. 2: Các tầng (layer) trong CNN là 3 chiều

Convolutional Neural Networks (CNN) là một trong những mô hình deep learning phổ biến nhất và có ảnh hưởng nhiều nhất trong cộng đồng thị giác máy tính (Computer Vision). CNN được dùng trong trong nhiều bài toán như nhân dạng ảnh, phân tích video, ảnh MRI, hoặc cho bài các bài của lĩnh vự xử lý ngôn ngữ tự nhiên,và hầu hết đều giải quyết tốt các bài toán này.

1.2.2 Mô hình mạng nơ ron tích chập

Một kiến trúc CNN bao gồm các lớp: convolution layer, pooling layer và fully connected layer. Ở giữa các lớp convolution và pooling thường có các hàm kích hoạt phi tuyến. Ảnh khi đưa vào mạng sẽ được lan truyền qua tầng convolution layer, giá trị tính được từ các tầng convolution sẽ đi qua một hàm kích hoạt, sau đó giá trị này sẽ được lan truyền qua pooling layer. Cuối cùng ảnh sẽ được lan truyền đến tầng fully connected layer và đi qua hàm kích hoạt Softmax, thường thì cuối cùng sẽ thu được một vector chứa xác suất phần trăm thuộc về các lớp đối với các bài toán phân loại. Ví dụ minh họa về một kiến trúc mạng nơ ron tích chập đầy đủ:



Hình 1. 3: Ví dụ minh họa về cấu trúc CNNs – LeNet – 5[15]

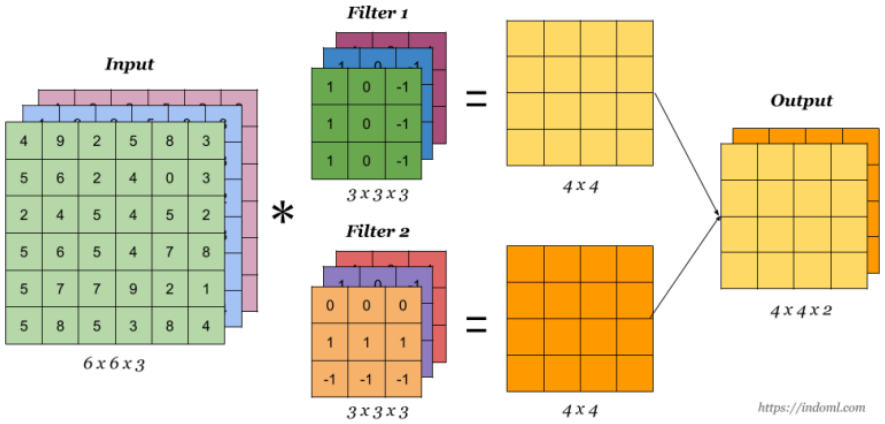
❖ **Convolution layer**

Convolution layer là lớp quan trọng nhất và cũng là lớp đầu tiên của của mô hình CNN. Lớp này có chức năng chính là phát hiện các đặc trưng có tính không gian hiệu quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, bộ filters, và receptive field, feature map. Conv layer nhận đầu vào là một ma trận 3 chiều và một bộ filters cần phải học. Bộ filters này sẽ trượt qua từng vị trí trên bức ảnh để tính tích chập (convolution) giữa bộ filter và phần tương ứng trên bức ảnh. Phần tương ứng này trên bức ảnh gọi là receptive field, tức là vùng mà một neuron có thể nhìn thấy để đưa ra quyết định, và mà trận cho ra bởi quá trình này được gọi là feature map.

Khi đưa ảnh vào mạng, bộ filter sẽ quét qua toàn bộ ảnh cho nên các đặc trưng cơ bản của ảnh như là góc, cạnh, màu sắc và texture sẽ được mạng phát hiện ra bất kể nó nằm ở vị trí nào trong ảnh. Do đó tầng convolution được xem như là một bộ

trích chọn đặc trưng (feature detector) vì nó có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh đầu vào. [16]

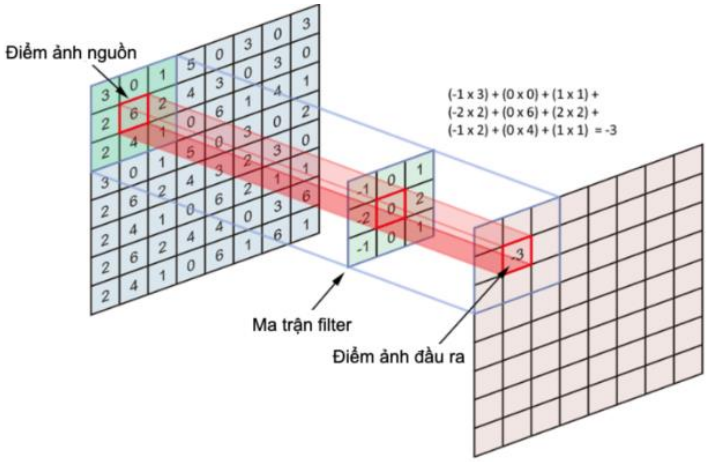
Khi áp dụng phép tính tích chập cho xử lý hình ảnh, người ta nhận thấy rằng kỹ thuật tích chập này sẽ giúp biến đổi các thông tin đầu vào thành các yếu tố đặc trưng (nó tương tự như bộ phát hiện nhằm phát hiện ra các đặc trưng như cạnh, hướng, ...). Hình 1.4 minh họa cho việc áp dụng phép tính tích chập trên ảnh và cho ra kết quả là một bản đồ đặc trưng - feature map. Cụ thể hơn, tích chập sẽ trích xuất đặc trưng của ảnh đầu vào qua các vùng ảnh nhỏ. Các vùng này được gọi là Local Receptive Field (LRF). Tích chập sẽ tính toán trên các LRF chồng lấp lên nhau. Độ chồng lắp này phụ thuộc vào hệ số trượt S (stride) của từng kiến trúc mạng cụ thể. Nếu sử dụng với hệ số trượt S = α, thì tương ứng LRF (bằng kích thước với kernel) sẽ dịch chuyển α đơn vị pixel sau mỗi lần tích chập.



Hình 1. 4: Minh họa cách thức tính chập của một ảnh RGB và ma trận kernel

Ảnh đầu vào sau khi thực hiện quá trình tích chập sẽ thu được bản đồ đặc trưng, số LRF ở ảnh đầu vào sẽ tương ứng với số neural ở feature map và kernel sẽ là trọng số liên kết mỗi LRF với một neural ở bản đồ đặc trưng. Lớp tích chập có thể chứa một hoặc nhiều feature map. Nếu lớp tích chập có K feature map, thì ta nói lớp conv này có độ sâu là k. Để hình dung rõ hơn về quá trình này, sau đây sẽ minh họa quá trình trích xuất đặc trưng từ ảnh đầu vào cụ thể như sau: thực hiện xử lý tính giá

trị đầu ra của một ảnh có kích thước W1×H1×D1 (W1 và H1 lần lượt là chiều rộng và chiều cao của ảnh và D1 là chiều sâu hay thực chất là giá trị tại 3 kênh màu tương ứng của ảnh RGB). khi đó, một Conv như một cửa sổ trượt (sliding window, còn được gọi là kernel, filter hay feature detector) với kích thước F ×F - giả sử trong trường ta sử dụng K filter. Trong quá trình xử lý, mỗi filter sẽ được tính toán với tất cả các LRF trong hình và S = α. Trong một số trường hợp để cân bằng giữa số bước di chuyển và kích thước của ảnh, người ta đã chèn thêm P pixel với một giá trị màu được gán (thông thường là 0) xung quanh viền của ảnh. sau cùng ta thu được ma trận đầu ra (feature map) với kích thước W2 ×H2 ×D2. [17] [18] [19]



Hình 1. 5: Mô phỏng quá trình tích chập trong CNN

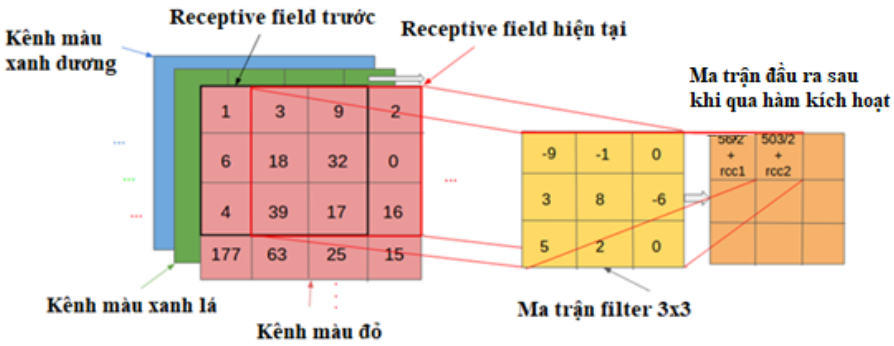
❖ **Các tham số của lớp tích chập – Convolutional Layer:**

Các tham số cơ bản của tầng convolution chính là kích thước filter, stride và padding. Trong đó quan trọng nhất chính là kích thước bộ filter, vì nó tỉ lệ thuận với số tham số cần học tại mỗi tầng convolution và là tham số quyết định receptive field của tầng này. Kích thước filter phổ biến thường dùng là 3x3.

Thông thường chúng ta nên chọn kích thước filter nhỏ, vì các lý do sau:

- Rút trích được các đặc trưng có tính cục bộ cao.
- Phát hiện được các đặc trưng nhỏ.
- Rút trích đa dạng đặc trưng, hữu ích cho các tầng sau.

- Kích thước ảnh giảm chậm, cho phép xây dựng một kiến trúc mạng sâu, học được nhiều hơn.
- Chia sẻ trọng số tốt.

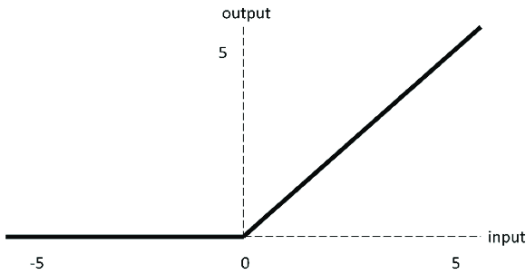


Hình 1. 6: Minh họa về bộ lọc filter

Ngoài ra, tham số stride cũng cần lưu ý bởi vì nó thể hiện số pixel cần phải dịch chuyển mỗi khi trượt bộ filter qua bức ảnh. Tham số padding cũng rất quan trọng bởi vì nó sẽ giúp giữ nguyên kích thước ma trận đầu ra của mỗi tầng convolution, do đó ta có thể xây dựng được một kiến trúc mạng với số tầng tùy ý.

❖ Hàm kích hoạt

Hàm kích hoạt là một hàm số nhận vào một giá trị đầu vào và kết quả là một giá trị có miền giá trị nằm trên một khoảng (hay nửa khoảng) nào đó. Một số các hàm kích hoạt phổ biến có thể kể đến đó là Sigmod, Tanh, Relu. Hàm kích hoạt rất quan trọng bởi vì nó sẽ tăng khả năng dự đoán của mạng neural và giúp mô hình học được các quan hệ phi tuyến phức tạp tiềm ẩn trong dữ liệu. Thông thường hàm kích hoạt sử dụng ở giữa các tầng convolution và pooling là hàm Relu. [17] [18] [19]



Hình 1. 7: Đồ thị hàm kích hoạt Relu

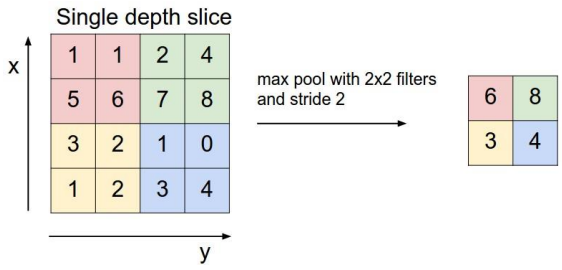
Hàm Relu có công thức toán học là $\boldsymbol{f(x) = \max(0, x)}$. Hàm Relu được ưa chuộng vì tính toán đơn giản, giúp hạn chế tình trạng vanishing gradient, và cũng cho kết quả tốt hơn. Relu cũng như những hàm kích hoạt khác, được đặt ngay sau tầng convolution, Relu sẽ gán những giá trị âm bằng 0 và giữ nguyên giá trị của đầu vào khi lớn hơn 0.

- **Lớp Pooling**

Lớp Pooling được sử dụng sau lớp Relu theo như mẫu thiết kết các lớp theo như trình bày của đại học Standford. Pooling giúp cho mạng giảm số lượng tham số, từ đó giúp đơn giản hóa quá trình tính toán của CNN và qua đó góp phần giải quyết vấn đề overfiting khi huấn luyện mạng.

Có nhiều toán tử pooling như Sum-pooling, Max-pooling, L2-pooling nhưng Max-pooling được sử dụng phổ biến nhất trong kiến trúc mạng CNN vì nó cho kết quả hơn so với những toán tử còn lại. Ngoài ra, Max-pooling còn giúp tạo ra tính bất biến dịch chuyển (translation invariance) cho đặc trưng. Cụ thể, dù đối tượng trong hình ảnh đầu vào có sự dịch chuyển nhỏ thì mạng vẫn có khả năng phân lớp chính xác được đối tượng. Đó là bởi vì max-pooling chọn ra neural có giá trị đầu ra tại mỗi vùng neural của lớp trước và tổng hợp thành lớp sau. Việc chọn neural có giá trị tín hiệu lớn nhất được xem như chọn ra đặc trưng tốt nhất để xử lý. Chính vì thế, khả năng phân lớp chính xác đối tượng dựa trên đặc trưng này vẫn không thay đổi và giúp cho CNN đạt được tính ổn định khi đối tượng di chuyển.

Trong lớp tích chập, có thể có nhiều feature map, tương ứng với mỗi feature map sẽ có một lớp max-pooling. Hình 1.8 là một ví dụ minh họa, với lớp đầu vào kích thước [28 × 28] giả sử ta thu được ba feature map kích thước [24 × 24] và ba lớp max-pooling kích thước [12×12], ta sử dụng kernel kích thước [5×5] và max-pooling lấy giá trị lớn nhất tại mỗi vùng [2×2] neural. [20]



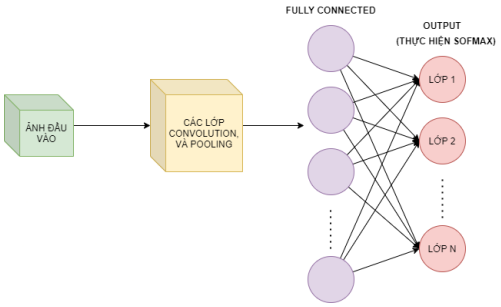
Hình 1. 8: Minh họa kỹ thuật Pooling trong mô hình CNN

❖ Lớp Normalization

Lớp Normalization (Norm) là lớp giúp chuẩn hóa dữ liệu đầu ra cho các lớp trong CNN trước khi được truyền đi tiếp. Trong những kiến trúc CNN lớn và phức tạp, lớp Norm sẽ chuẩn hóa các giá trị neural trước khi chúng được truyền đến hàm Relu. Hàm Relu tuy giúp rút ngắn thời gian huấn luyện, nhưng nếu không điều chỉnh trọng số phù hợp, hàm Relu sẽ rất dễ gặp phải vấn đề "dying Relu" khiến cho mạng trở nên chậm hơn khi huấn luyện. Lớp Norm lúc này sẽ chuẩn hóa và tạo ra các giá trị tích chập phù hợp để tránh cho Relu rơi vào giá trị 0. Tránh việc gradient xấp xỉ bằng 0 khiến cho tốc độ học của mạng trở nên rất chậm.

❖ Lớp đầy đủ - Fully connected layer

Tầng cuối cùng của mô hình CNN trong bài toán phân loại ảnh là tầng fully connected layer. Tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán. Ví dụ trong một bài toán phân lớp có 10 lớp, tầng fully connected layer sẽ chuyển ma trận đặc trưng của tầng trước thành vector có 10 chiều thể hiện xác suất của 10 lớp tương ứng. [8] [22]



Hình 1. 9: Minh họa Fully connected layer

1.3 Bài toán phân loại cảm xúc khuôn mặt

Nhận dạng cảm xúc trên khuôn mặt (Facial Emotion Recognition) là một công nghệ được sử dụng để phân tích cảm xúc theo các nguồn khác nhau, chẳng hạn như hình ảnh và video. Bài toán này thuộc về nhóm công nghệ gọi là "tính toán cảm xúc", một lĩnh vực nghiên cứu đa ngành về khả năng của máy tính để nhận biết và giải thích cảm xúc và trạng thái tình cảm của con người và nó thường được xây dựng dựa trên công nghệ Trí tuệ nhân tạo. [20] [21]

Hình 1. 10: Tổng quan hệ thống nhận diện cảm xúc

Dựa vào Hình 1.10, chúng ta thấy rằng một hệ thống nhận diện cảm xúc hoàn chỉnh sẽ có ba bước chính như sau:

- **Bước 1**: Phát hiện khuôn mặt – Face detection, mục tiêu của bước này sẽ xác định khuôn mặt ở trong bức hình, từ đó chúng ta cắt khuôn mặt ra qua bước hai để xác định các biểu diễn cảm xúc trên khuôn mặt

- **Bước 2**: Phát hiện biểu cảm - Facial Expression detection: Sau khi xác định được khuôn mặt của một người trong bức hình, chúng ta sẽ xác định ra các biểu trên khuôn mặt như ánh mắt, nụ cười, v.v.
- **Bước 3**: Phân loại cảm xúc - Expression Classification: Dựa vào các đặc trưng rúc trích ở bước 2 chúng ta sẽ xác định cảm xúc của người hiện tại trên bức hình.

1.4 Kết luận chương 1

Mạng nơ ron nhân tạo là một chuỗi các thuật toán được sử dụng để tìm ra mối quan hệ của một tập dữ liệu thông qua cơ chế vận hành của bộ não sinh học. Mạng nơ ron nhân tạo thường được huấn luyện qua một tập dữ liệu chuẩn cho trước, từ đó có thể đúc rút được kiến thức từ tập dữ liệu huấn luyện, và áp dụng với các tập dữ liệu khác với độ chính xác cao.

Các phương pháp sử dụng để huấn luyện mạng nơ ron nhân tạo ngày càng tối ưu hơn về mặt tính toán và phục vụ cho nhiều mục đích khác nhau. Hiện nay, kiến trúc mạng nơ ron ngày càng được hoàn thiện cho nhiều nhiệm vụ, trong đó mạng nơ ron tích chập được chú ý rất nhiều vì tính hiệu quả trong thị giác máy tính. Mạng nơ ron tích chập với các cải tiến góp phần giảm thời gian tính toán và tăng độ chính xác hứa hẹn sẽ là một trong những phương pháp được áp dụng rất nhiều vào thực tế trong tương lai.

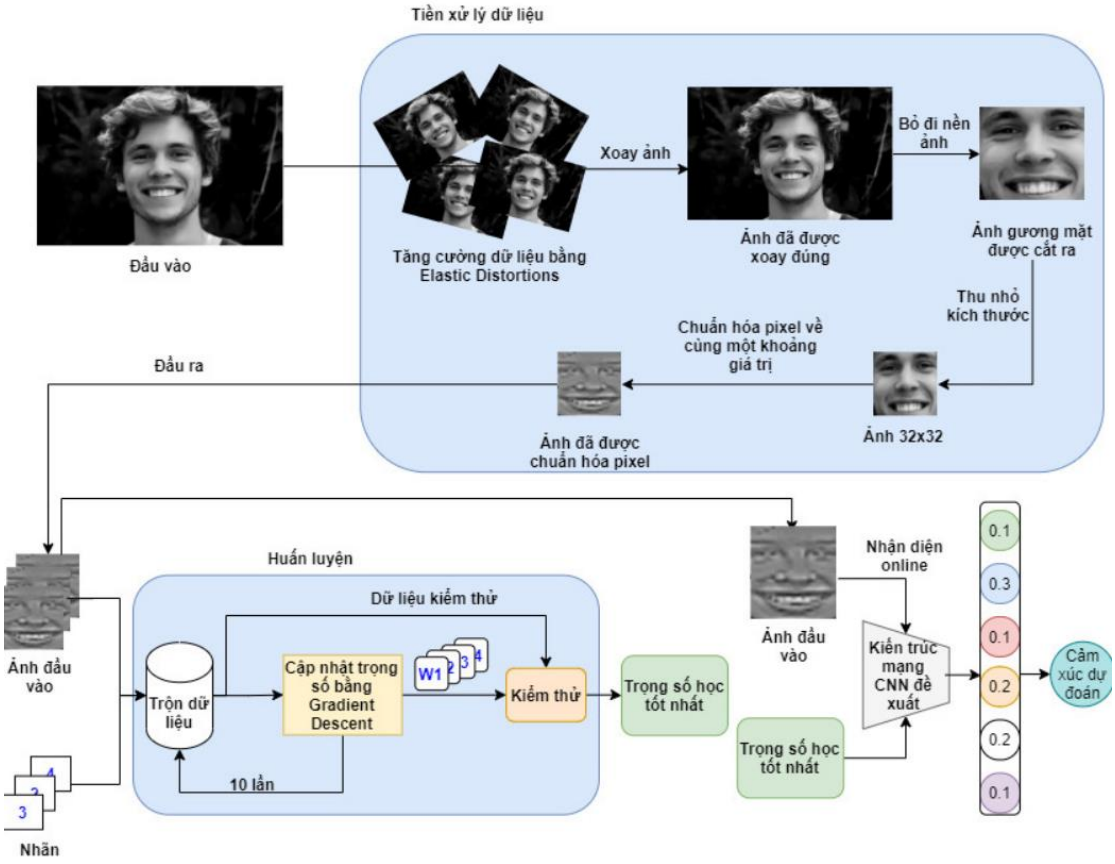
CHƯƠNG 2: HỆ THỐNG NHẬN DẠNG BIỂU CẢM KHUÔN MẶT

Ở chương này, học viên sẽ trình bày chi tiết hệ thống nhận dạng biểu cảm khuôn mặt dựa trên kiến trúc mạng tích chập Convolution Neural Network cũng như quy trình thí nghiệm trên hai bộ dữ liệu chuẩn cho bài toán này là CK+ và JAFFE. Quy trình thí nghiệm bao gồm hai bước là: bước huấn luyện và bước thử nghiệm. Học viên mô tả bài toán như sau:

Đầu vào: Bức ảnh của một người trong bộ dữ liệu CK+ hoặc JAFFE.

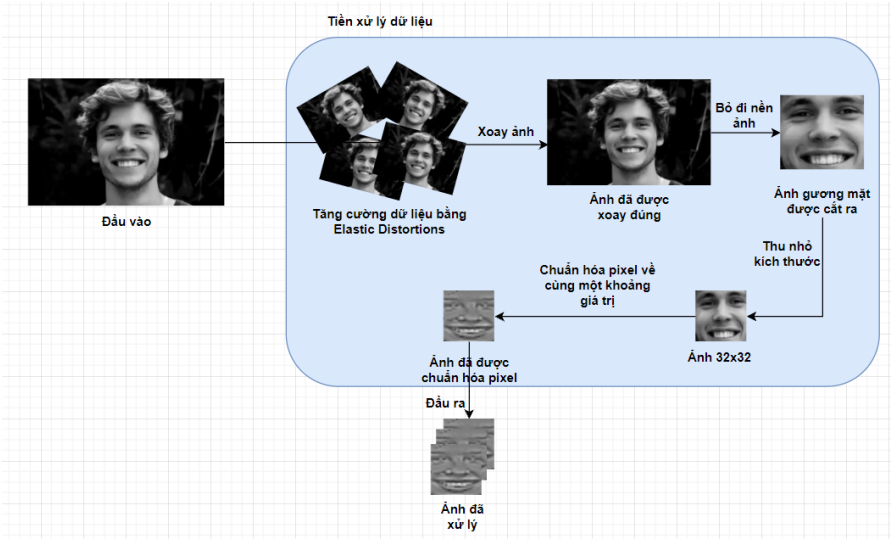
Đầu ra: Nhãn cảm xúc của người dùng theo các lớp như 0. angry, 1. disgust, 2. fear, 3. happy, 4 . neutral, 5. sad và 6. Surprise.

Pha huấn luyện bao gồm các kỹ thuật: dùng phương pháp tạo ảnh tổng hợp (Synthetic images generation) để tăng số lượng mẫu học; sử dụng một chuỗi các kỹ thuật tiền xử lý ảnh (pre-processing) như: tính toán lại góc nghiêng của ảnh trong các điều kiện chụp không lý tưởng, chuẩn hóa giá trị cường độ ảnh, loại bỏ nền xung quanh gương mặt, điều chỉnh lại kích thước ảnh gương mặt (down sampling). Sau đó sử dụng một kiến trúc mạng học sâu tích chập (Convolution neural network) để huấn luyện. Pha thử nghiệm giống với pha huấn luyện tuy nhiên phương pháp sinh ảnh sẽ không được áp dụng.



Hình 2. 1: Sơ đồ tổng quan phương pháp đề xuất

2.1 Tiền xử lý ảnh mặt người và tăng cường mẫu học



Hình 2. 2: Sơ đồ tổng quan các bước tiền xử lý dữ liệu được áp dụng

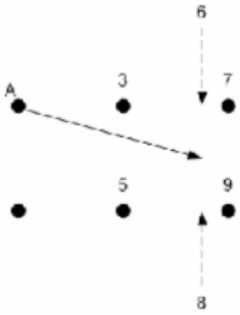
2.1.1 Tổng hợp tạo mẫu

Các thuật toán Học sâu hiện nay đôi khi cần phải được huấn luyện trên một bộ dữ liệu đủ nhiều, khi thiếu dữ liệu chúng ta sẽ phải gặp các vấn đề như: thuật toán hoạt động tốt trên tập huấn luyện nhưng mô hình lại dự đoán kém trên tập thử nghiệm (over-fitting); huấn luyện trở nên khó khăn, khó tìm ra được các trọng số tốt nhất cho mô hình, khó hội tụ. Trong những trường hợp thiếu dữ liệu, việc sử dụng một mạng Học sâu hoặc kiến trúc mạng quá phức tạp sẽ mang lại kết quả không khả quan, do đó ta có thể giải quyết bằng các cách: thiết kế một mạng không quá phức tạp để thử nghiệm lại; sử dụng kỹ thuật transfer learning để tận dụng các trọng số đã được huấn luyện trên một bộ dữ liệu lớn và tiếp tục huấn luyện trên dữ liệu riêng; thu thập thêm nhiều dữ liệu.

Các mạng CNN học sâu thường yêu cầu rất nhiều dữ liệu để mô hình có thể hội tụ nhanh chóng, tuy nhiên các bộ dữ liệu hiện tại về cảm xúc gương mặt thường gặp hạn chế về số mẫu, do đó việc sinh thêm dữ liệu bằng các kỹ thuật tăng cường là một giải pháp cần thiết để nâng cao độ chính xác của các mô hình học sâu. Việc tăng cường dữ liệu học thông thường là thực hiện các phép rotations, translations và skewing trên ảnh thật. Một trong những phương pháp đó là sử dụng Elastic Distortions, áp dụng biến đổi affine trên các vùng của ảnh, Tương ứng với ban đầu của mỗi pixel, tính toán vị trí mới cho mỗi pixel đó: tính các giá trị $\Delta x(x, y) = \alpha x$ và $\Delta y(x, y) = \alpha y$ cho mỗi pixel (x, y), giá trị color mới cho pixel (x, y) là color nội suy tại pixel $(x + \alpha x, y + \alpha y)$, trong đó Δ là giá trị khoảng shift và α là giá trị scale.

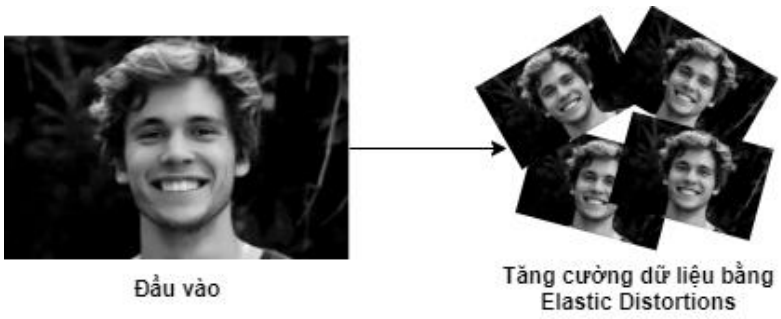
Hình 2.3 minh họa cách áp dụng một trường biến đổi để tính toán giá trị mới ở mỗi pixel. Trong ví dụ này, vị trí ở A sẽ được gán (0,0), và các vị trí 3, 5, 7, 9 là các mức xám của ảnh được chuyển đổi tại vị trí (1,0), (2,0), (1, -1) và (2, -1) một cách lần lượt. Các biến đổi của A được xác định là $\Delta x(0, 0) = 1.75$ và $\Delta y(0, 0) = -0.5$ như ví dụ mũi tên trong ảnh. Giá trị xám mới A ở ảnh mới được tính toán bằng cách đánh giá mức xám tại vị trí (1.75, -0.5) ở ảnh gốc. Một giải thuật đơn giản để tính mức xám

là "nội suy song tuyến tính" của các giá trị pixel trên ảnh gốc. Mặc dù các phép nội suy khác có thể được sử dụng (như phép lưỡng tính hoặc nội suy spline), nội suy song tuyến tính là phương pháp đơn giản nhất và hoạt động tốt.



Hình 2. 3: Ví dụ minh họa tính một giá trị mức xám mới ở A, tại vị trí (0,0)

Mô hình đề xuất sẽ sử dụng kỹ thuật Elastic Distortions để sinh thêm ảnh trong quá trình huấn luyện. Cụ thể, mô hình sử dụng Elastic Distortions với $\sigma = 3$ pixels and $\mu = 0$ để tạo ra các mẫu học với nhiễu ngẫu nhiên ở vùng mắt, với mỗi ảnh thật sẽ tạo ra 70 ảnh tăng cường. Vị trí của mắt thì tương ứng với ảnh thật nhưng bị làm nhiễu bằng Gaussian noise.



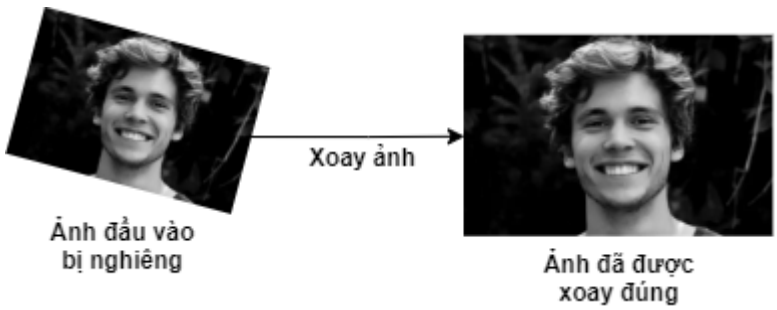
Hình 2. 4: Ví dụ cách áp dụng Elastic Distortions để sinh các ảnh gương mặt

2.1.2 Chỉnh sửa xoay (Rotation correction)

Các ảnh trong môi trường thật có thể khác nhau ở góc độ, ánh sáng, kích thước dù cùng 1 biểu cảm, những khác biệt đó có thể ảnh hưởng đến độ chính xác của hệ thống. Do đó, cần loại bỏ các ảnh hưởng của góc độ bằng cách căn chỉnh lại vùng

mặt theo phương ngang bằng các phép quay và phép chuyển đổi. Để xoay lại ảnh gương mặt, cần phải có hai thông tin hình ảnh khuôn mặt và trung tâm của hai mắt.

Với mô hình đề xuất, cần xác định vị trí của 2 mắt và điểm trung tâm giữa 2 mắt làm điểm trung tâm. Thực hiện phép xoay với để đường nối 2 mắt hợp với phương ngang 1 góc là 0 độ. Ảnh đưa vào có thể là ảnh thật hoặc ảnh tăng cường hiệu chỉnh xoay được thể hiện ở hình dưới [8] [22] [23]



Hình 2. 5: Minh họa quá trình xoay lại ảnh gương mặt

Đầu vào cho quá trình xoay lại ảnh có thể là ảnh gốc hoặc ảnh tăng cường. Việc điều chỉnh quay cho các hình ảnh tăng cường có thể không thực hiện được tốt với trục ngang bởi vì trung tâm mắt là các vị trí thực bị nhiễu bởi điểm ngẫu nhiên được sinh ra bởi Elastic Distortions. Do đó, nó sẽ tạo ra hình ảnh bị xáo trộn bởi phép quay và phép tăng cường, làm tăng số mẫu tăng cường trong dữ liệu huấn luyện. Hình 18 minh họa lại quá trình xoay lại ảnh gương mặt cho hợp với phương ngang.

2.1.3 Cắt ảnh gương mặt (Face cropping)

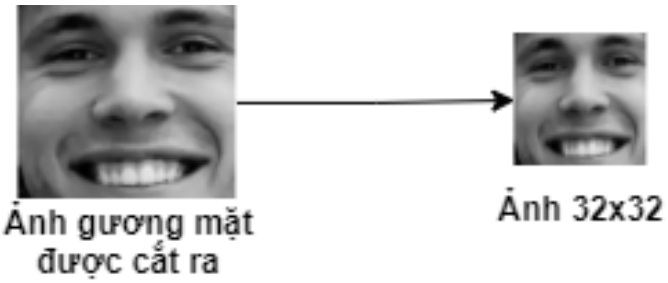
Các vùng ảnh nền không liên quan đến biểu cảm nhưng gây ảnh hưởng đến độ chính xác của mô hình, nên cần loại bỏ thông tin ảnh nền khỏi ảnh. Sau khi cắt ảnh, chỉ các giá trị pixel vùng mặt được giữ lại.

Hình 2. 6: Một ví dụ loại bỏ các nền xung quanh gương mặt

Với mô hình đề xuất, cần xác định khoảng cách giữa 2 mắt, sau đó thực hiện cắt ảnh gương mặt với các kích thước xác định như sau: chiều cao (height) vùng cắt có hệ số 4.5 gồm: 1.3 cho vùng trên mắt và 3.2 cho vùng dưới mắt, chiều rộng vùng cắt (width) có hệ số 2.4, các hệ số trên nhân với khoảng cách từ giữa 2 mắt đến mắt phải để có kích thước thật. Việc loại bỏ các phần nền, chỉ giữ lại ảnh gương mặt giúp mô hình chỉ tập trung khai thác các đặc trưng ngữ nghĩa trên vùng gương mặt, giúp khai thác tốt hơn các yếu tố về cảm xúc mà không ảnh hưởng bởi nhiễu. Hình 2.6 cho thấy minh họa của quá trình này.

2.1.4 Giảm kích thước ảnh gương mặt (Downsampling)

Kỹ thuật giảm kích thước ảnh thường rất được sử dụng để huấn luyện các mô hình học sâu. Lý do phổ biến là khi đưa dữ liệu ảnh vào các mạng học sâu để huấn luyện, chiều dài và chiều rộng của ảnh cần được đưa về một kích thước thống nhất (thông thường là nhỏ hơn ảnh ban đầu). Ngoài ra, kỹ thuật giảm kích thước ảnh còn có nhiều lợi ích khác như sau: làm cho dữ liệu có kích thước dễ quản lý hơn; Giảm kích thước của dữ liệu do đó cho phép xử lý dữ liệu nhanh hơn (hình ảnh); Giảm kích thước lưu trữ của dữ liệu. Ngoài ra còn có một số cách sử dụng khác của kỹ thuật này tùy thuộc vào cách sử dụng. Với mô hình đề xuất, sử dụng nội suy tuyến tính để giảm kích thước ảnh về 32x32. Hình 2.7 cho thấy một ví dụ giảm kích thước ảnh. [22] [23]

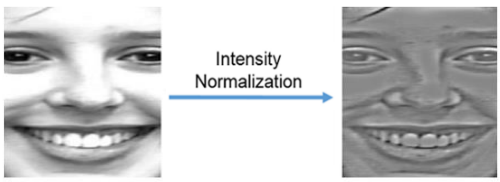


Hình 2. 7: Một ví dụ giảm kích thước ảnh

2.1.5 Chuẩn hóa cường độ

Các yếu tố về độ sáng (brightness) và độ tương phản (contrast) sẽ gây nhầm lẫn ở các ảnh của cùng lớp biểu cảm và cùng đối tượng. Do dó, cần giảm ảnh hưởng hai yếu tố này để giảm độ phức tạp cho bộ phân lớp bằng chuẩn hóa cường độ (Intensity normalization).

Chuẩn hóa cường độ là thực hiện chuẩn hóa về một khoản giá trị bằng phương pháp cân bằng đối lập (contrastive equalization) tại mỗi pixel: 1) xác định kernel xung quanh mỗi pixel, tính độ lệch chuẩn σ và phương sai μ tại kernel đó; 2) giá trị pixel mới được tính toán bởi công thức công thức $x' = \frac{x - \mu_{nhgx}}{\sigma_{nhgx}}$ với x' là giá trị mới và x là giá trị pixel ban đầu. Hình 2.8 cho thấy ví dụ về quá trình chuẩn hóa giá trị pixel trong ảnh. [22] [23]



Hình 2. 8: Một ví dụ chuẩn hóa các giá trị pixel trong ảnh [13]

2.2 Mạng nơ ron tích chập cho phân lớp cảm xúc

2.2.1 Kiến trúc mạng nơ-ron tích chập (Convolutional Neural Network)

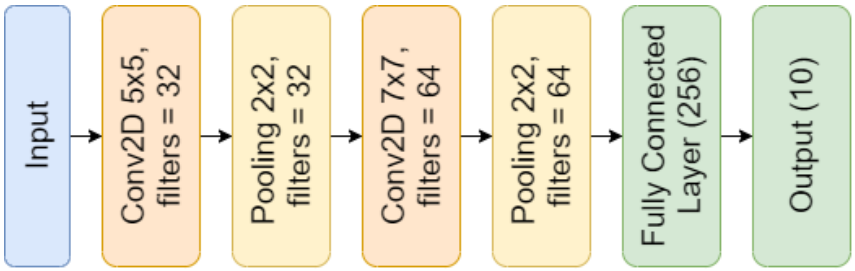
Mô hình đề xuất có kiến trúc mạng CNN như sau:

Input (32, 32)
Lớp Convolution 1: kích thước kernel (5, 5); kích thước đầu ra (28, 28, 32)
Lớp Pooling 1: kích thước kernel (2, 2); kích thước đầu ra (14, 14, 32)
Lớp Convolution 2: kích thước kernel (7, 7); kích thước đầu ra (8, 8, 64)
Lớp Pooling 2: kích thước kernel (2, 2); kích thước đầu ra (4, 4, 64)
Lớp kết nối đầy đủ: 256 hidden node
Lớp đầu ra: kết quả của hàm softmax, trả về phân bố xác suất của 6 hoặc 7 lớp cảm xúc

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Conv2D_1 (Conv2D)            (None, 28, 28, 32)        832
_________________________________________________________________
Activation_1 (Activation)    (None, 28, 28, 32)        0
_________________________________________________________________
MaxPooling2D_1 (MaxPooling2D (None, 14, 14, 32)        0
_________________________________________________________________
Conv2D_2 (Conv2D)            (None, 8, 8, 64)          100416
_________________________________________________________________
Activation_2 (Activation)    (None, 8, 8, 64)          0
_________________________________________________________________
MaxPooling2D_2 (MaxPooling2D (None, 4, 4, 64)          0
_________________________________________________________________
Flatten_1 (Flatten)          (None, 1024)              0
_________________________________________________________________
Dense_1 (Dense)              (None, 256)               262400
_________________________________________________________________
Activation_3 (Activation)    (None, 256)               0
_________________________________________________________________
Dense_2 (Dense)              (None, 7)                 1799
_________________________________________________________________
Activation_4 (Activation)    (None, 7)                 0
=================================================================
Total params: 365,447
Trainable params: 365,447
Non-trainable params: 0
```

Hình 2. 9: Thông số chi tiết mô hình CNN trong thí nghiệm của học viên

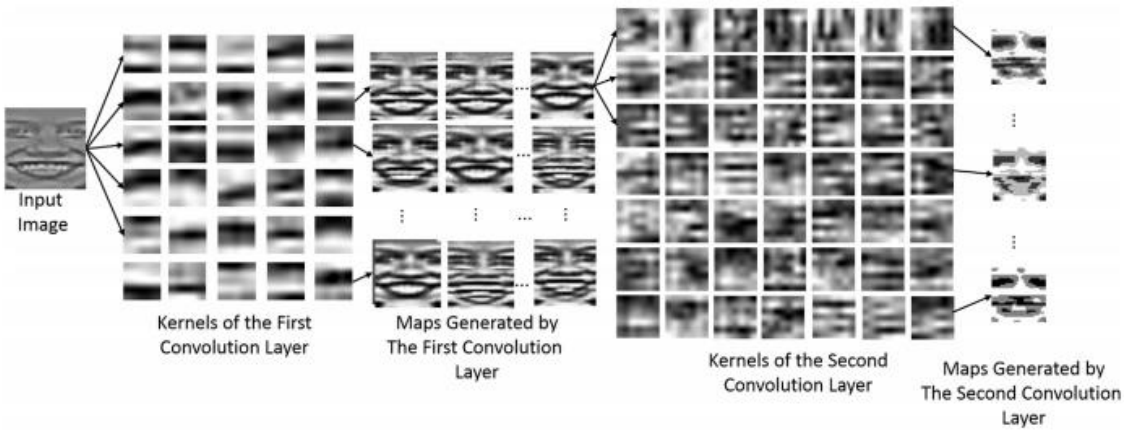
Trong quá trình huấn luyện, bộ tối ưu hóa được sử dụng là Stochastic Gradient Descent (SGD) và hàm kích hoạt ở mỗi lớp tích chập là Relu. Kiến trúc đề xuất được minh họa ở Hình 2.10.



Hình 2. 10: Minh họa kiến trúc CNN trong mô hình đề xuất

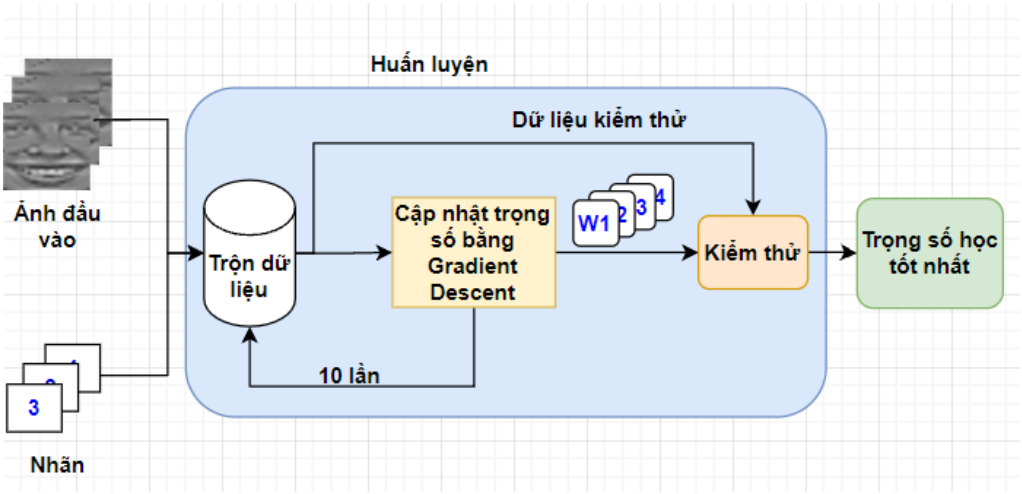
Ý nghĩa của các layer của mô hình là như sau:

- Lớp convolution 1 và pooling thứ 1: rút trích các đặc điểm cơ bản như các cạnh, điểm cuối, góc và hình dạng nói chung. Các đặc trưng rút trích được chủ yếu là các hình dạng, góc, các cạnh của mắt, chân mày và môi.
- Lớp convolution 2 và pooling thứ 2: rút trích các đặc trưng thấp hơn của khuôn mặt có ảnh hưởng lớn đến biểu cảm. Các đặc trưng rút trích được chủ yếu là các vùng gần mắt, miệng và mũi.
- Kết hợp các lớp trên thu được các yếu tố bất biến của biểu cảm trên khuôn mặt.
- Lớp ẩn cuối (kết nối đầy đủ): nhận một bộ các feature đã học và output độ tin cậy của mỗi lớp biểu cảm.



Hình 2. 11: Ví dụ minh họa các đặc trưng ảnh trích xuất được qua từng lớp tích chập Convolutional layer [13]

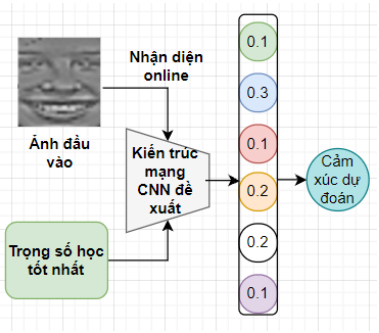
2.2.2 Huấn luyện



Hình 2. 12: Mô hình tổng quan quá trình huấn luyện và kiểm thử mô hình huấn luyện trên hai bộ dữ liệu

Trong phương pháp đề xuất, dữ liệu đã qua tiền xử lý và nhãn tương ứng sẽ được đưa vào kiến trúc mạng học sâu để tiến hành huấn luyện. Trước khi huấn luyện, học viên tiến hành trộn dữ liệu (Data shuffle), mục đích của việc này giúp dữ liệu không theo một thứ tự có trước nào, nên khi đưa một tập ảnh vào kiến trúc mạng thì mạng sẽ học được nhiều các trường hợp cảm xúc hơn. Nếu không trộn dữ liệu, các mẫu dữ liệu có cảm xúc giống nhau đưa vào mạng khiến mạng không học được các mẫu dữ liệu có cảm xúc khác, điều này ảnh hưởng cực kỳ nặng đến độ chính xác của mô hình.

2.2.3 Kiểm thử



Hình 2. 13: Mô hình tổng quan quá trình kiểm thử dữ liệu trên bộ dữ liệu kiểm tra

Sau quá trình huấn luyện sẽ thu được bộ trọng số tốt nhất. Bộ trọng số này được dùng để kiểm thử độ chính xác của mô hình trên tập dữ liệu thử nghiệm. Các ảnh thử nghiệm cũng được đưa qua hệ thống tiền xử lý giống như huấn luyện (trừ sử dụng Elastic Disortions để tăng cường dữ liệu), sau đó đầu ra của quá trình tiền xử lý là đầu vào của mạng học sâu. Các đặc trưng ảnh ngày càng tăng về độ sâu để học đặc trưng ngữ nghĩa cấp cao, đến các lớp kết nối đầy đủ (fully connected layer) sẽ được trải thẳng ra thành một dạng lớp ẩn gồm nhiều node, cuối cùng đi qua một lớp Softmax gồm 6 node chính là phân bố xác suất của 6 lớp cảm xúc đầu ra. Lúc này, xác suất dự đoán của cảm xúc nào cao nhất thì đầu ra cuối cùng sẽ là cảm xúc đó.

2.2.4 Mạng Deep Convolutional Neural Network (DCNN)

Ở phần trên, học viên đã trình bày một mô hình kiến trúc mạng mạng học Convolutional Neural Network cơ bản với sự kết hợp giữa các lớp tích chập Convolutional layer, Lớp gộp Maxpooling và 2 lớp đầy đủ Fully connected layer để tiến hành chạy thí nghiệm trên bộ dữ liệu. Các nghiên cứu gần đây cho thấy rằng chúng ta có thể áp dụng nhiều lớp tích chập với các kích thước bộ lọc khác nhau để rút trích các đặc trưng trong ảnh. Tuy nhiên khi tăng độ sâu của mô hình CNN sẽ dẫn đến trường hợp overfitting trong quá trình huấn luyện mô hình. Do đó, học viên cũng áp dụng các phương pháp giảm overfitting trong quá trình huấn luyện như kỹ thuật DropOut hay kỹ thuật BatchNormalization để nhằm mục đích tăng hiệu quả dự đoán các nhãn trên hai bộ dữ liệu thực nghiệm. Học viên gọi mô hình này là mô hình Deep Convolutional Neural Network – DCNN bởi vì học viên thử nghiệm nhiều lớp tích chập hơn để rút trích các đặc trưng khác nhau của bức ảnh. Tổng quan mô hình DCNN được trình bày như sau:

Input (32, 32)
Lớp Convolution 1: kích thước kernel (5, 5); kích thước đầu ra (32, 32, 64)
Lớp Batch Normalization 1; kích thước đầu ra (32, 32, 64)
Lớp Convolution 2: kích thước kernel (5, 5); kích thước đầu ra (32, 32, 64)
Lớp Batch Normalization 2; kích thước đầu ra (32, 32, 64)
Lớp Pooling 2: kích thước kernel (2, 2); kích thước đầu ra (16, 16, 64)
Lớp Dropout 2 với tỷ lệ 0.25
Lớp Convolution 3: kích thước kernel (3, 3); kích thước đầu ra (16, 16, 128)
Lớp Batch Normalization 3; kích thước đầu ra (16, 16, 128)
Lớp Convolution 4: kích thước kernel (3, 3); kích thước đầu ra (16, 16, 128)
Lớp Pooling 4: kích thước kernel (2, 2); kích thước đầu ra (8, 8, 128)
Lớp Dropout 4 với tỷ lệ 0.25
Lớp Convolution 5: kích thước kernel (3, 3); kích thước đầu ra (8, 8, 256)
Lớp Batch Normalization 5; kích thước đầu ra (8, 8, 256)
Lớp Convolution 6: kích thước kernel (3, 3); kích thước đầu ra (8, 8, 128)
Lớp Pooling 6: kích thước kernel (2, 2); kích thước đầu ra (4, 4, 128)
Lớp Dropout 6 với tỷ lệ 0.25
Lớp Fully Connected 1: Đầu ra (128)
Lớp Batch Normalization 7; kích thước đầu ra (128)
Lớp Dropout 7 với tỷ lệ 0.25
Lớp Output; đầu ra (7)

```
Model: "DCNN"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 64)        1664
_________________________________________________________________
batchnorm_37 (BatchNormaliza (None, 32, 32, 64)        256
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        102464
_________________________________________________________________
batchnorm_1 (BatchNormalizat (None, 32, 32, 64)        256
_________________________________________________________________
maxpool2d_1 (MaxPooling2D)   (None, 16, 16, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
batchnorm_2 (BatchNormalizat (None, 16, 16, 128)       512
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 128)       147584
_________________________________________________________________
maxpool2d_2 (MaxPooling2D)   (None, 8, 8, 128)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 256)         295168
_________________________________________________________________
batchnorm_7 (BatchNormalizat (None, 8, 8, 256)         1024
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 8, 8, 256)         590080
_________________________________________________________________
maxpool2d_3 (MaxPooling2D)   (None, 4, 4, 256)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               524416
_________________________________________________________________
batchnorm_8 (BatchNormalizat (None, 128)               512
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
out_layer (Dense)            (None, 7)                 903
=================================================================
Total params: 1,738,695
Trainable params: 1,737,415
Non-trainable params: 1,280
_________________________________________________________________
```

Hình 2. 14: Chi tiết đầu vào và các thông số của mô hình DCNN được sử dụng

- **Nhận xét sự so sánh giữa hai mô hình DCNN và CNN**

Nhìn vào chi tiết thông số và kiến trúc mô hình ở Hình 2.9 và Hình 2.14 tương ứng của hai mô hình CNN và mô hình DCNN. Chúng ta có thể nhận thấy rằng mô hình DCNN có tổng số lượng tham số mô hình cao hơn khoảng gần 3 lần so với mô hình CNN. Cụ thể với mô hình DCNN có tổng tham số là 1,738,695 tham số, trong khi số lượng này ở mô hình CNN chỉ là 365,447 tham số. Bởi vì ở mô hình DCNN, chúng ta sử dụng nhiều lớp tích chập Convolutional layer với các bộ lọc và kích thước khác nhau. Dẫn đến việc mô hình có nhiều tham số cần phải học trong quá trình huấn luyện hơn.

Chính vì số lượng tham số nhiều nên là thời gian để huấn luyện một mô hình DCNN cũng nhiều hơn so với mô hình CNN. Cụ thể đối với mỗi epoch thì mô hình DCNN cần khoảng 12s để hoàn thành một epoch, trong khi đó với mô hình CNN thì chỉ cần khoảng 3s để hoàn thiện trên một epoch. Còn về độ chính xác và sự hiệu quả của mô hình được học viên trình bày ở chương 3 so sánh hiệu quả giữa hai mô hình.

2.3 Kết luận của chương 2

Trong chương này, học viên đã trình bày tổng quan hệ thống nhận diện khuôn mặt và các bước tiền xử lý hình ảnh được áp dụng trong quá trình thí nghiệm nhằm mục đích nâng cao hiệu quả dự đoán trên cả hai bộ dữ liệu. Mục 2.1.1 là kỹ thuật tăng cường ảnh, từ mục 2.1.2 đến mục 2.1.5 là các kỹ thuật tiền xư lý.

Bên cạnh đó, học viên cũng đã trình bày chi tiết hai mô hình mạng học sâu CNN và mô hình học sâu DCNN bao gồm chi tiết các thông số, kiến trúc và số lượng tham số của mỗi mô hình. Đây là hai mô hình chính được sử dụng để chạy thí nghiệm trên hai bộ dữ liệu thực nghiệm và so sánh.

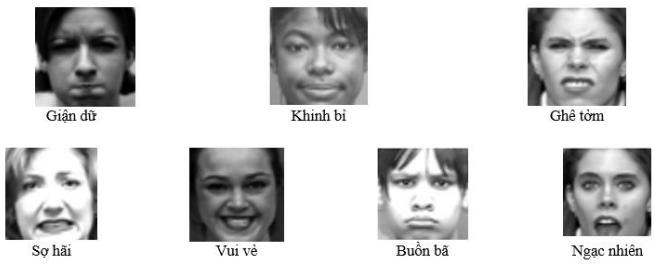
CHƯƠNG 3: THỬ NGHIỆM VÀ THẢO LUẬN

3.1 Cơ sở dữ liệu

3.1.1 Dữ liệu Cohn-Kanade mở rộng (CK+)

Dữ liệu CK+ bao gồm các ảnh gương mặt (phần lớn là các ảnh đơn sắc) với các loại cảm xúc: buồn bã (sadness), bất ngờ (surprise), hạnh phúc (happiness), sợ hãi (fear), giận dữ (anger), khinh thường (contempt), ghê tởm (disgust). Mỗi ảnh trong tập dữ liệu có kích thước 48x48 và có tổng cộng 981 ảnh trong bộ dữ liệu. [18]

Cơ sở dữ liệu CK+ được tạo từ 210 đối tượng với độ tuổi từ 18-50 tuổi. Các hình ảnh được chụp từ một máy ảnh đặc trực tiếp ở phía trước đối tượng. Các đối tượng được hướng dẫn để thực hiện một loạt biểu cảm. Mỗi trình tự được bắt đầu bằng biểu cảm trung lập và kết thúc với biểu hiện đặc trưng. Gồm các đối tượng, người mỹ gốc phi chiếm 81%, người châu á hoặc nam mỹ là 13%, đối tượng khác là 6%.

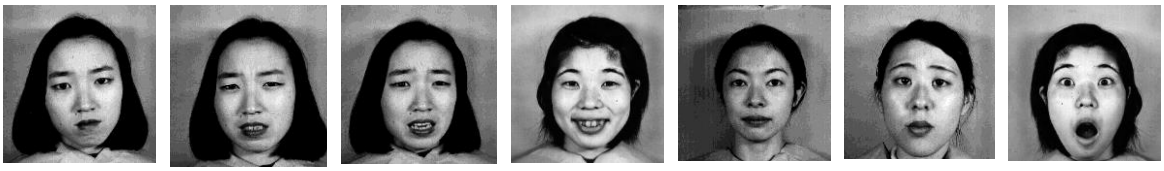


Hình 3. 1: Hình ảnh trong tập dữ liệu CK+

3.1.2 The Japanese Female Facial Expression (JAFFE) Dataset

Dữ liệu JAFFE bao gồm ảnh của 10 phụ nữ Nhật Bản với 6 cảm xúc cơ bản (Giận dữ, khinh thường, sợ hãi, hạnh phúc, buồn bã, bất ngờ) và 1 nhãn bình thường. Dữ liệu bao gồm 213 ảnh đơn sắc với kích thước 256x256 với độ chính xác 8 bit cho

các giá trị thang độ xám. Trong cơ sở dữ liệu này, có khoảng 4 hình ảnh ở mỗi một trong sáu biểu cảm cơ bản và một hình ảnh của biểu cảm trung tính từ mỗi đối tượng.



Angry Disgust Fear happy Neutral Sad Surrise

Hình 3. 2: Hình ảnh trong tập dữ liệu JAFFE

3.2 Môi trường thử nghiệm

Để huấn luyện mô hình và cài đặt thuật toán này "Phân tích biểu cảm mặt người dùng mạng nơ ron tích chập", học viên sử dụng các công cụ, thư viện và ngôn ngữ lập trình như sau:

- Lập trình bằng Python: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết ngắn hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép chúng ta tập trung vào những giải pháp chứ không phải cú pháp.
- Thư viện máy học Tensorflow – keras: Keras được coi là một thư viện mức độ cao của TensorFlow, Microsoft (CNTK), hoặc Theano. Keras có cú pháp đơn giản hơn TensorFlow rất nhiều. Các ưu điểm của thư viện này là: (1) Keras ưu tiên trải nghiệm của người lập trình, (2) Keras hỗ trợ huấn luyện trên nhiều GPU phân tán, (3) Keras đã được sử dụng rộng rãi trong doanh nghiệp và cộng đồng nghiên cứu.
- Thư viên Xử lý ảnh – OpenCV: OpenCV là tên viết tắt của Open Source computer vision library. Đây là một thư viện mã nguồn mở phục vụ cho hướng nghiên cứu xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực. OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt

động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các mô hình máy học khác nhau.

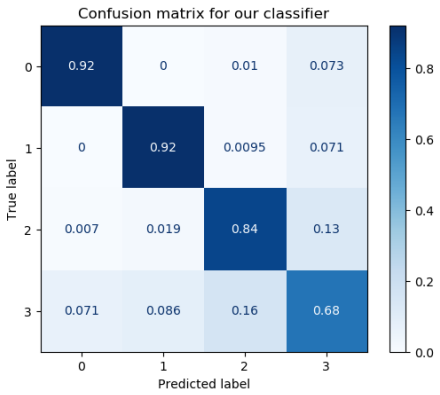
Cấu hình thử nghiệm cho các nghiên cứu được thực hiện trên máy tính cá nhân Window 10 Enterprise LTSC, Intel Core i7 Tiger Lake -1180H 4.6 GHz với NVIDA GeForce RTX 3050 Ti 4GB có khả năng có 2.3 Gb bộ nhớ trong GPU.

3.3 Cài đặt thử nghiệm và độ đo đánh giá

Đối với phương pháp mô hình mạng tích chập CNN, học viên sử dụng thư viện Tensorflow Keras trên ngôn ngữ lập trình Python để cài đặt. Giá trị dropout em sử dụng là 0.2 và hàm kích hoạt là Relu. Số lượng node của lớp đầy đủ là 128 với hàm kích hoạt là Relu. Gía trị dropout được sử dụng tại lớp này là 0.25 để giảm. Hàm tối ưu được sử dụng là SGD với giá trị học là 0.01 và tốc đô momentum là 0.95. Hàm mất mát được sử dụng là hàm Cross-entropy.

Để đánh giá hiệu quả của các phương pháp, học viên tiến hành các mô hình thử nghiệm đề xuất và các mô hình học máy cơ bản sử dụng ba độ đo là độ chính xác – Accuracy, độ chính xác - Precision, độ phủ - Recall và chỉ số F1 – F1 score giữa tập dự đoán và tập dữ liệu được gán nhãn. Các độ đo được tính bằng các công thức sau đây:

- Accuracy – Độ chính xác: Cách đơn giản nhất để đánh giá một mô hình phân lớp đó là sử dụng độ chính xác (Accuracy). Ý tưởng đơn giản là tỷ lệ giữa các mẫu dự đoán đúng trên tổng số mẫu của dữ liệu kiểm thử.
- Ma trận Confusion: Cách tính sử dụng accuracy chỉ cho chúng ta biết được bao nhiêu phần trăm dữ liệu được dự đoán đúng mà không chỉ ra các dữ liệu được dự đoán đúng/sai như thế nào. Do đó chúng ta cần một phương pháp đánh giá tốt hơn gọi là ma trận Confusion (Confusion matrix). Một cách tổng quát, ma trận này sẽ thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc về một lớp, và được được dự đoán là rơi vào một lớp.



Hình 3. 3: Ví dụ về ma trận confusion

Như hình ở trên, confusion matrix được thể hiện bằng nhiều màu sắc để dễ nhận biết. Một mô hình tốt sẽ cho một confusion matrix có các phần tử trên đường chéo chính có giá trị lớn, các phần tử còn lại có giá trị nhỏ. Một cách đơn giản đó là đường chéo chính càng màu đậm thì mô hình càng tốt.

- **True/False Positive/Negative:** Cách đánh giá này thường được áp dụng cho các bài toán phân lớp có hai lớp dữ liệu. Cụ thể hơn, trong hai lớp dữ liệu này có một lớp nghiêm trọng hơn lớp kia và cần được dự đoán chính xác.

  True/False Positive/Negative được định nghĩa như sau:

  - True postitive (TP): các điểm dữ liệu dự đoán là 1 và có nhãn là 1.
  - True negative (TN): các điểm dữ liệu dự đoán là 0 và có nhãn là 0.
  - False positive (FP): các điểm dữ liệu dự đoán là 1 nhưng có nhãn là 0.
  - False negative (FN): các điểm dữ liệu dự đoán là 0 nhưng có nhãn là 1.

Trong đó, Precision và Recall: Đối với các bộ dữ liệu không cần bằng (số dữ liệu các lớp lệch nhau nhiều) thì ta cần một phép đo mới đánh giá rõ hơn, đó là Precision – Recall.

- Precision hoặc Positive Predictive Value (PPV): Tỉ lệ dương tính đoán đúng.

$$PPV = \frac{TP}{TP + FP}$$

- Recall hoặc True Positive Rate (TPR): Độ nhạy, tỷ lệ dương tính thực.

$$TPR = \frac{TP}{TP + FN}$$

- Precision được định nghĩa là tỉ lệ số điểm TP trong số các điểm được phân loại là positive (TP + FP).
- Recall được định nghĩa là tỉ lệ số điểm TP trong số các điểm thực sự là positive (TP + FN).
- Precision cao đồng nghĩa với việc độ chính xác của các điểm tìm được là cao. Recall cao đồng nghĩa với việc TPR cao, có nghĩa là tỉ lệ bỏ sót của các điểm thực sự positive là thấp.
- Khi đó, ta có thể xác định: PPV chính là tỉ lệ báo động nhầm, TPR chính là tỉ lệ bỏ sót.

❖ F1-score: F1-score là giá trị trung bình của precision và recall.

$$\frac{2}{F_1} = \frac{1}{precision} + \frac{1}{recall} \; hay \; F_1 = 2.\frac{precision.recall}{precision + recall}$$

F1-score có giá trị nằm trong nửa khoảng $(\mathbf{0}, \mathbf{1}]$. F1-score càng cao đồng nghĩa với bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 thì F1-score = 1. Khi cả recall và precision đều thấp thì F1-score tiến về 0.

3.4 Số liệu

3.4.1 Thử nghiệm bộ dữ liệu CK+ gốc

- *BestEpoch = 35 46 55 38 21 34 40 41*
- *Accuracy = 0.73 0.82 0.79 0.8 0.75 0.73 0.82 0.67*

Learning Rate = 0.01

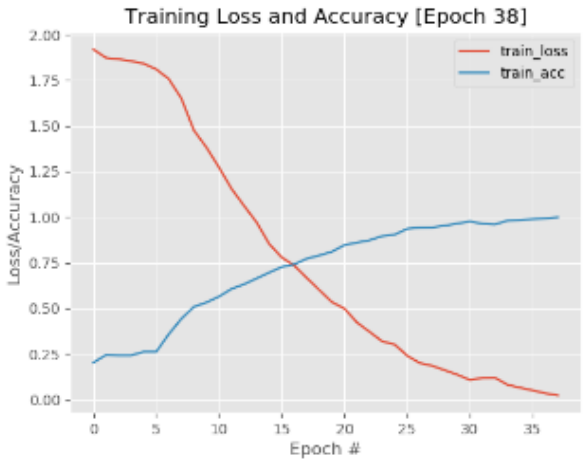
Momentum = 0.95

ClassLabes = ['ANGRY' 'DISGUST' 'FEAR' 'HAPPY' 'NEUTRAL' 'SAD' 'SUPPRISE']

TrainSupport = 1467: [165 206 107 238 357 115 279]

Train Loss = 0.02

Train Accuracy = 1.00



Hình 3. 4: Epoch tốt nhất khi chạy bộ dữ liệu gốc CK+

Dựa trên kết quả thử nghiệm và hình 3.4 khi chạy bộ dữ liệu thử nghiệm bộ dữ liệu gốc CK+ vơi 8 fold mỗi fold là 100 epochs thì Number Epochs = 38, trong fold 4 la tốt nhất.

3.4.2 Thử nghiệm bộ dữ liệu CK+ khi tăng cường dữ liệu học

- BestEpoch = 17 13 26 27 36 33 10 56
- Accuracy = 0.86 0.88 0.89 0.82 0.86 0.87 0.78 0.92

Learning Rate = 0.01

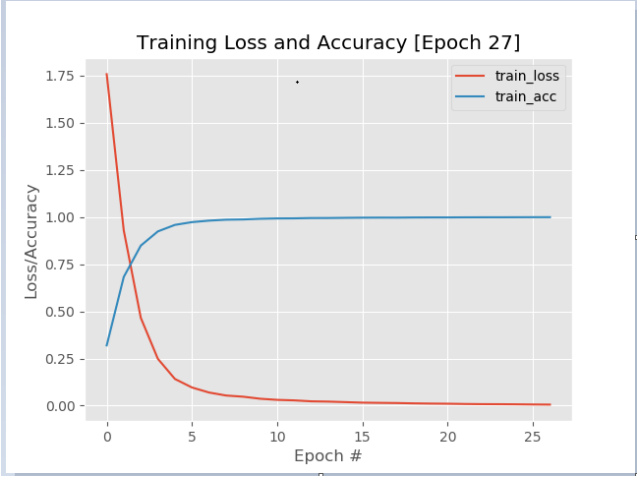
Momentum = 0.95

ClassLabes = ['ANGRY' 'DISGUST' 'FEAR' 'HAPPY' 'NEUTRAL' 'SAD' 'SUPPRISE']

TrainSupport = 89029: [9585 12567 5325 14697 23212 5964 17679]

Train Loss = 0.01

Train Accuracy = 1.00



Hình 3. 5: Epoch tốt nhất khi chạy bộ dữ liệu đã tăng cường CK+

Dựa trên kết quả thử nghiệm và hình 3.5 khi chạy bộ dữ liệu tăng cường CK+ vơi 8 fold mỗi fold là 100 thì Number Epochs = 27, trong fold 4 là tốt nhất.

3.4.3 Thử nghiệm bộ dữ liệu JAFFE gốc

- BestEpoch = 34 36 41 24 27
- Accuracy = 0.63 0.37 0.5 0.6 0.5

Learning Rate = 0.01

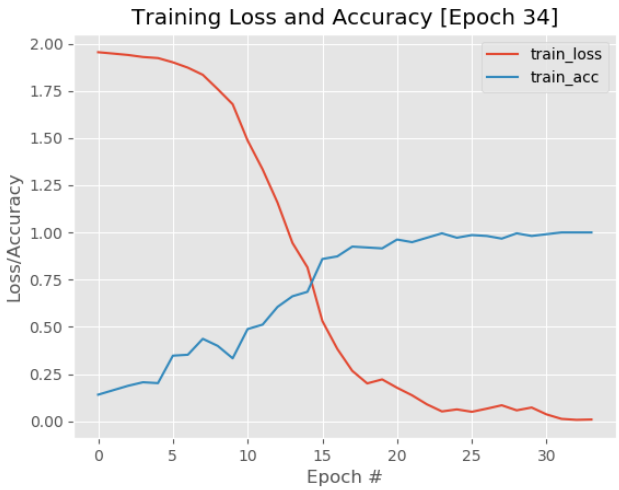
Momentum = 0.95

ClassLabes = ['ANGRY' 'DISGUST' 'FEAR' 'HAPPY' 'NEUTRAL' 'SAD' 'SUPPRISE']

TrainSupport = 213: [30 29 32 31 30 31 30]

Train Loss = 0.01

Train Accuracy = 1.00



Hình 3. 6: Epoch tốt nhất khi chạy bộ dữ liệu gốc JAFFE

Dựa trên kết quả thử nghiệm và hình 3.6 khi chạy bộ dữ liệu JAFFE gốc chia làm 5 fold thì Number Epochs = 34, trong fold 1 là tốt nhất.

3.4.4 Thử nghiệm bộ dữ liệu JAFFE tăng cường

- BestEpoch = 23 80 66 53 51
- Accuracy = 0.6 0.59 0.65 0.69 0.71

Learning Rate = 0.01

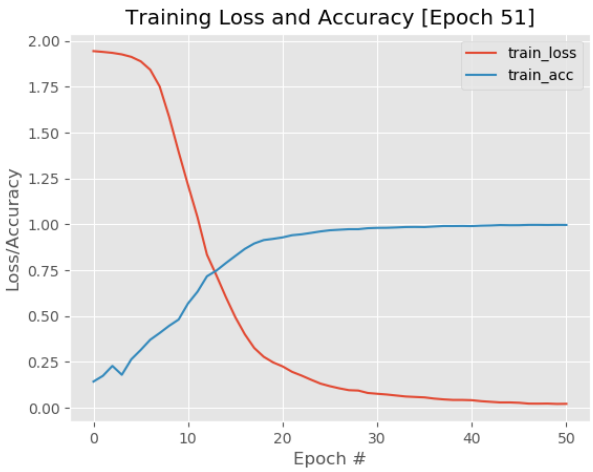
Momentum = 0.95

ClassLabes = ['ANGRY' 'DISGUST' 'FEAR' 'HAPPY' 'NEUTRAL' 'SAD' 'SUPPRISE']

TrainSupport = 15122: [2129 2059 2272 2201 2130 2201 2130]

Train Loss = 0.02

Train Accuracy = 1.00



Hình 3. 7: Epoch tốt nhất khi chạy bộ dữ liệu tăng cường JAFFE

Dựa trên kết quả thử nghiệm và hình 3.7 khi chạy bộ dữ liệu JAFFE tăng cường ảnh làm 5 fold thì Number Epochs = 51, trong fold 5 là tốt nhất.

3.5 Kết quả thử nghiệm

Trong phần này, học viên sẽ phân tích và đánh giá hiệu quả của các phương pháp tiền xử lý hình ảnh so với việc không áp dụng tiền xử lý trên hai bộ dữ liệu chuẩn là CK+ và JAFFE. Kết quả được báo cáo dưới dạng đánh giá chéo K-fold cross validation. Kết quả chi tiết các độ đo như độ chính xác – Accuracy, độ chính xác – Precision, độ phủ - Recall và chỉ số F1-score trên hai bộ dữ liệu. Đầu tiên học viên sẽ trình bày kết quả chi tiết trên bộ dữ liệu CK+, sau đó học viên sẽ trình bày kết quả trên bộ JAFFE. Kết quả thí nghiệm được báo cáo dựa trên chạy thí nghiệm đánh giá chéo (cross validation), cụ thể là 8 folds cho bộ dữ liệu CK+ và 5 folds cho bộ dữ liệu JAFFE. Kết quả cuối cùng là giá trị trung bình mỗi lần chạy.

Nhìn vào Bảng 3.1, chúng ta sẽ thấy kết quả chính của mô hình CNN khi kết hợp các phương pháp tiền xử lý khác nhau trên bộ dữ liệu CK+ tương ứng cho mỗi nhãn cảm xúc. Dựa vào bảng kết quả, chúng ta thấy rằng có sự khác biệt về sự hiệu quả dựa các nhãn với nhau. Chúng ta dễ dàng nhìn thấy được sự hiệu quả của mô hình này đối với các nhãn cảm xúc “Disgust”, “Happy”, “Surprise” hay nhãn

"Neutral" với chỉ số F1-score lần lượt là 0.88, 0.89, 0.93 và 0.81. Trong khi đó, các nhãn còn lại có kết quả tương đối thấp là nhãn "Fear", nhãn "Sad" với giá trị F1 score là 0.45 và 0.46. Một trong những nguyên chính dẫn đến kết quả này là số lượng dữ liệu của các nhãn này trong bộ dữ liệu tương đối ít. Điều này dẫn đến việc mô hình không học được tốt nên hiệu quả các nhãn này thấp bởi vì mô hình CNN cần một lượng lớn dữ liệu để học hiệu quả. Số lượng dữ liệu cho từng nhãn cảm xúc ít nhất trong bộ dữ liệu CK+ tương ứng là nhãn "Fear", "Sad", "Angry", đó chính là lý do tại sao kết quả trên ba nhãn này đạt kết quả thấp nhất trong toàn bộ nhãn cảm xúc. Để tiến hành kiểm tra kết quả tại sao mô hình CNN lại cho kết quả thấp trên các nhãn này, bảng kết quả ma trận nhầm lẫn trung bình được mô tả ở Bảng 3.2.

Bảng 3. 1: Kết quả chi tiết của mô hình CNN trên bộ dữ liệu CK+ cho từng nhãn cảm xúc

Nhãn cảm xúc	**Precision**	**Recall**	**F1-score**
Angry	0.67	0.57	0.61
Disgust	0.89	0.88	**0.88**
Fear	0.48	0.43	0.45
Happy	0.89	0.88	**0.89**
Neutral	0.77	0.87	0.81
Sad	0.57	0.42	0.46
Surprise	0.92	0.94	**0.93**
Accuracy		**0.80**	
Weighted F1	0.79	0.80	0.80

Bảng 3. 2: Kết quả nhầm lẫn giữa các nhãn cảm xúc của bộ dữ liệu CK+ khi huấn luyện sử dụng mô hình CNN

	Angry	Disgust	Fear	Happy	Neutral	Sad	Suprise
Angry	**9.6**	0.9	0.5	0.6	**4.0**	0.9	0.4
Disgust	0.5	19.8	0.0	0.8	0.6	0.0	0.5
Fear	1.2	0.0	**3.1**	0.9	1.9	1.6	0.6
Happy	0.5	1.1	0.4	22.8	0.6	0.0	0.5
Neutral	1.8	0.2	0.9	0.1	35.8	1.4	0.8
Sad	0.9	0.2	1.0	0.2	**3.1**	**4.8**	0.2
Suprise	0.0	0.0	0.8	0.1	1.0	0.1	29.1

Dựa theo tỷ lệ trung bình nhẫm lẫn giữa các các nhãn cảm xúc của bộ CK+, chúng ta thấy được rằng tỷ lệ nhãn "Angry" bị dự đoán nhãn thành nhãn "Neutral" chiếm tỷ lệ cao so với các nhãn khác với giá trị trung bình 4.0, cao hơn rất nhiều so với sự nhầm lẫn các nhãn còn lại. Trong khi đó nhãn "Fear" lại có tỷ lệ dự đoán sai hầu như đối với các nhãn cảm xúc còn lại trừ nhãn "Disgust" với các giá trị nhẫm lẫn tương đối gần nhau. Ngược lại thì nhãn "Sad" lại bị dự đoán nhầm lẫn thành nhãn "Neutral" với tỷ lệ 3.1, tỷ lệ này cao rất nhiều so với các nhãn còn lại. Từ đó có thể thấy rằng hầu hết các nhãn có tỷ lệ dự đoán sai hầu hết đều dự đoán thành nhãn "Neutral" trong bộ dữ liệu CK+ này. Chúng ta có thể thấy rằng nhãn "Angry" và nhãn "Sad" có xu hướng nhầm lẫn qua "Neutral" do con người thường không bộc lộ cảm xúc, nên trạng thái nó gần nhau. Do đó mô hình đạt kết quả thấp trên hai nhãn này.

Tiếp theo, học viên sẽ đánh giá kết quả trên bộ dữ liệu JAFFE, bảng 3.3 và bảng 3.4 trình bày kết quả chi tiết trên từng nhãn cảm xúc và ma trận nhầm lẫn trung bình của mô hình CNN.

Bảng 3. 3: Kết quả chi tiết của mô hình CNN trên bộ dữ liệu JAFFE cho từng nhãn cảm xúc

Nhãn cảm xúc	Precision	Recall	F1-score
Angry	0.59	0.53	0.56
Disgust	0.65	0.59	**0.62**
Fear	0.38	0.38	0.38
Happy	0.76	0.56	**0.64**
Neutral	0.55	0.47	0.52
Sad	0.48	0.43	0.45
Surprise	0.59	0.53	0.55
Accuracy	0.49		
Weighted F1	0.52	0.54	0.53

Nhìn một cách tổng quan chúng ta thấy rằng bộ dữ liệu JAFFE này cho kết quả thấp hơn hẳn so với bộ dữ liệu CK+ trên hầu hết các nhãn cảm xúc. Trong đó nhãn có kết quả tốt nhất trên bộ dữ liệu này là nhãn "Happy" với độ chính xác là 0.76, độ phủ là 0.56 và độ đo F1 là 0.64. Tiếp theo là nhãn cảm xúc "Disgust' với độ chính xác là 0.65, độ phủ là 0.59 và độ chính xác là 0.62. Trong khi đó nhãn thấp nhất vẫn là nhãn "Fear" và nhãn "Sad" với kết quả lần lượt là 0.38 cho cả ba độ đo và độ chính xác là 0.48, 0.43 và 0.45 tương ứng cho từng nhãn. Tuy nhiên, khác với bộ dữ liệu CK+ có tỷ lệ chênh lệnh nhãn khác nhau thì bộ dữ liệu JAFFE này lại có sự đồng đều trong các nhãn. Do đó, khả năng kết quả các nhãn này thấp bởi vì mô hình CNN thí nghiệm hiện tại chưa có khả năng phân biệt rõ các khuôn mặt cảm xúc trong bộ dữ liệu. Còn xét về độ đo weighted F1 score thì mô hình CNN đạt kết quả là độ chính xác là 0.52, độ phủ là 0.54 và giá trị F1 cuối cùng là 0.53. Kết quả này vẫn còn tương đối khá thấp với bộ dữ liệu CK+. Tiếp theo học viên sẽ trình bày kết quả của mô hình CNN trên ma trận nhầm lẫn của bộ JAFFE ở bảng 3.4.

Bảng 3. 4: Kết quả nhầm lẫn giữa các nhãn cảm xúc của bộ dữ liệu JAFFE khi huấn luyện sử dụng mô hình CNN

	Angry	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**	**Suprise**
Angry	**3.2**	**1.2**	0.4	0.4	0.0	0.8	0.0
Disgust	1.0	3.4	0.6	0.2	0.0	0.6	0.0
Fear	0.0	0.6	**1.6**	0.4	**1.2**	0.4	0.0
Happy	0.2	0.2	0.4	**3.4**	0.6	0.6	2.2
Neutral	0.4	0.4	0.4	0.6	2.8	0.6	0.8
Sad	0.6	0.6	0.4	0.0	**1.6**	**2.2**	0.8
Suprise	0.4	0.0	0.6	0.4	0.4	1.0	3.2

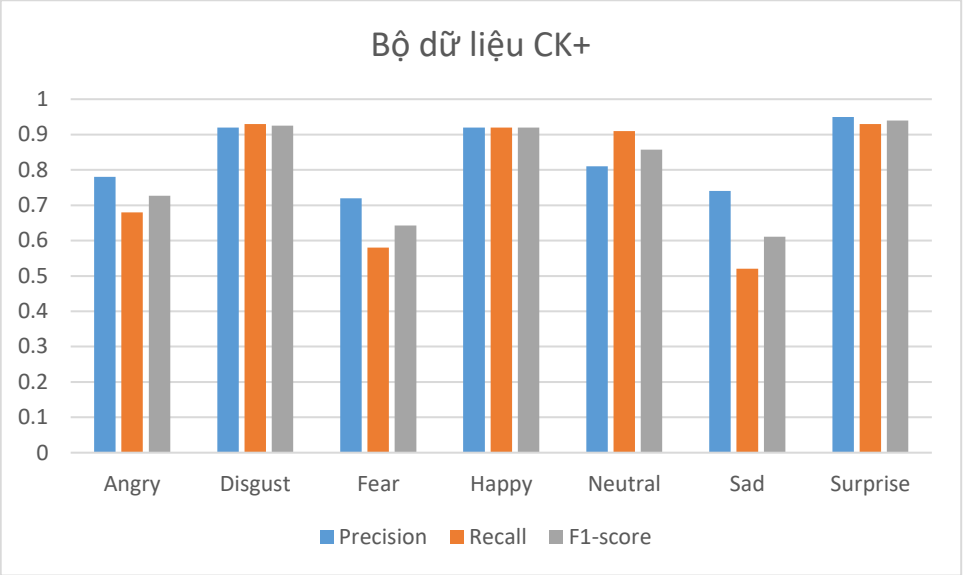
Dựa vào bảng 3.4, chúng ta thấy rằng sự nhầm lẫn giữa các nhãn trong bộ dữ liệu này không có sự chện lệch giữa các nhãn với nhau như bộ CK+, chúng ta có thể thấy như nhãn "Angry" bị dữ đoán sai nhiều nhất thành nhãn "Disgust" với tỷ lệ 1.2, tiếp theo bị nhầm lẫn với nhãn "Sad" là 0.8 và hai nhãn "Fear" và "Happy" là 0.4. Ở chiều ngược lại thì nhãn "Disgust" cũng bị dự đoán nhầm thành nhãn "Angry" với tỷ lệ nhiều nhất là 1.0. Còn đối với nhãn "Fear" thì hầu hết bị dự đoán nhầm thành nhãn cảm xúc "Neutral" với tỷ lệ là 1.2.

Phần tiếp theo sau đây, học viên sẽ đánh giá và so sánh hiệu quả của phương pháp tăng cường dữ liệu trên cả hai bộ dữ liệu để đánh giá xem hiệu quả. Tương tự như kết quả ở trên, học viên sẽ đánh giá lần lượt trên cả hai bộ dữ liệu và tất cả thí nghiệm được chạy theo đánh giá chéo (cross validation). Đầu tiên, học viên sẽ báo cáo kết quả tổng quan của mô hình CNN khi chạy trên dữ liệu thực tế được thu thập của hai bộ dữ liệu CK+ và JAFFE. Sau đó, học viên sẽ phân tích chi tiết từng nhãn cảm xúc cho từng bộ dữ liệu.

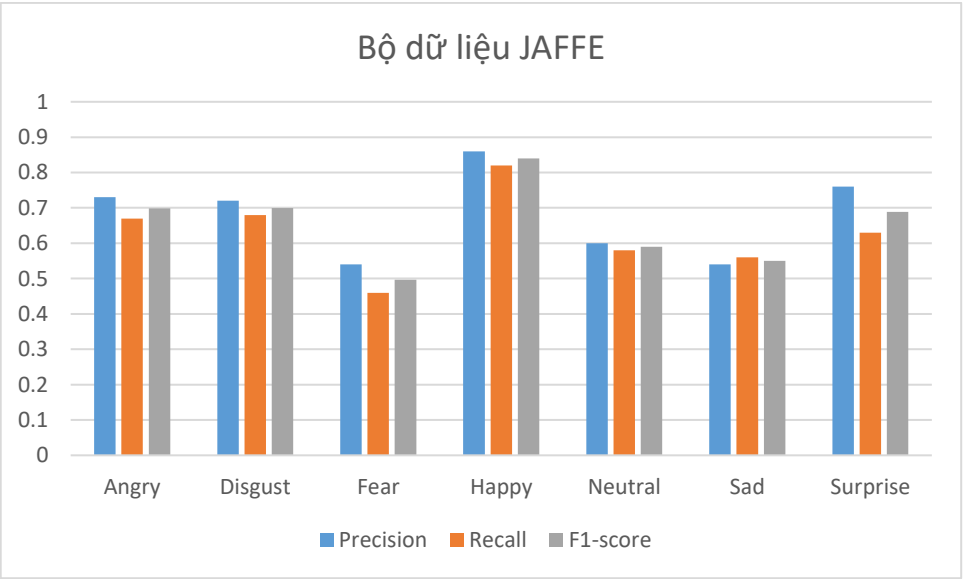
Bảng 3. 5: Kết quả khi áp dụng kỹ thuật tăng cường dữ liệu trên cả hai bộ dữ liệu CK+ và bộ dữ liệu JAFFE sử dụng mô hình CNN

	Bộ dữ liệu	**Accuracy**	**Precision**	**Recall**	**F1-score**
Tăng cường dữ liệu	CK+	0.86	0.84	0.86	0.85
	JAFFE	0.62	0.68	0.60	0.63
Không tăng cương dữ liệu	CK+	0.79	0.80	0.79	0.80
	JAFFE	0.49	0.52	0.54	0.53

Nhìn vào Bảng 3.5, chúng ta dễ dàng nhận thấy được sự hiệu quả của phương pháp tăng cường dữ liệu trên bộ cả hai bộ dữ liệu này, đặc biệt là bộ dữ liệu JAFFE. Cụ thể đối với bộ dữ liệu CK+, mô hình CNN kết hợp với phương pháp tăng cường dữ liệu đạt kết quả độ chính xác – Accuracy là 0.86, độ chính xác – Precision là 0.84, độ phủ là 0.86 và chỉ số F1 là 0.85. Kết quả này cao hơn mô hình CNN khi không áp dụng kỹ thuật tăng cường dữ liệu là +0.7 đối với độ đo Accuracy, độ chính xác – precision là +0.4 và độ phủ là +0.7 và độ đo F1 là +0.5. Trong khi đó thì đối với bộ dữ liệu JAFFE, thì mô hình CNN và kỹ thuật tăng cường dữ liệu đều cho kết quả tốt hơn khi không áp dụng. Cụ thể kỹ thuật tăng cường dữ liệu giúp mô hình đạt được độ chính xác - Accuracy là 0.60, độ chính xác 0.68, độ phủ là 0.60 và chỉ số F1 đạt được là 0.63. Kết quả này cao hơn kết quả gốc lần lượt là +1.3, +1.4, +0.7 và +1.0 trên lần lượt các độ đo. Điều này chứng tỏ rằng phương pháp tăng cường dữ liệu này giúp mô hình đạt hiệu quả tốt hơn khi áp dụng kiến trúc mạng tích chập CNN trên cả hai bộ dữ liệu. Tiếp theo học viên sẽ phân tích hiệu quả chi tiết trên từng nhãn cảm xúc khi tăng cường dữ liệu trên cả hai bộ dữ liệu.



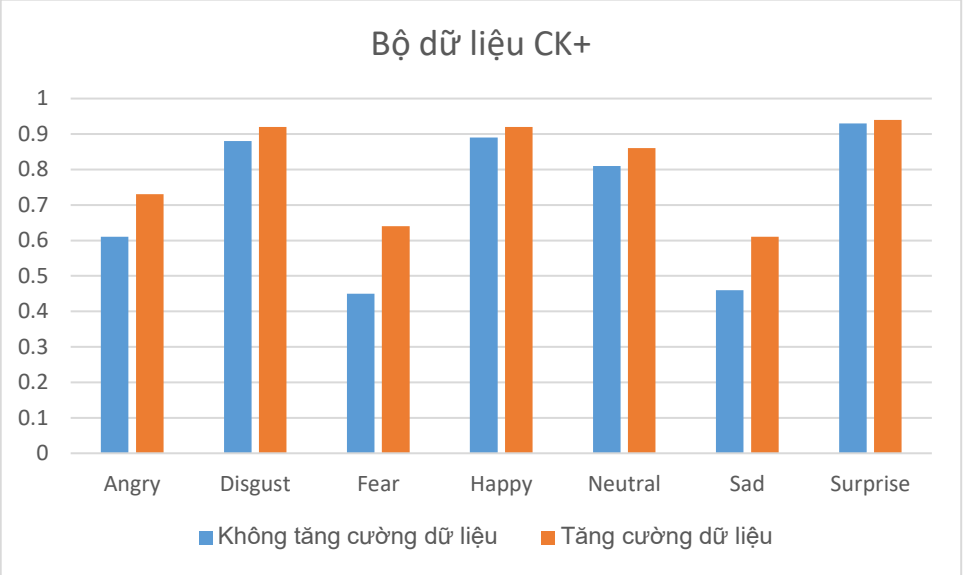
Hình 3. 8: Kết quả chi tiết các độ đo cho từng nhãn cảm xúc khi tăng cường dữ liệu sử dụng mô hình CNN trên bộ dữ liệu CK+



Hình 3. 9: Kết quả chi tiết các độ đo cho từng nhãn cảm xúc khi tăng cường dữ liệu sử dụng mô hình CNN trên bộ dữ liệu JAFFE

Hình 3.8 và Hình 3.9 trình bày kết quả chi tiết 3 độ đo theo bộ là chính xác, độ phủ và chỉ số F1 của mô hình CNN khi tăng cường dữ liệu. Đối với bộ dữ liệu CK+, chúng ta thấy có các nhãn đạt kết quả trên 0.9 như nhãn "Disgust", "Happy" và

nhãn “Surprise”. Các nhãn “Fear” và “Sad” vẫn là hai nhãn có kết quả thấp nhất. Tương tự như bộ dữ liệu JAFFE, chúng ta thấy nhãn “Happy” đạt kết quả tốt nhất với kết quả độ chính xác là 0.86, độ phủ là 0.82 và độ đo F1 là 0.84. Tiếp theo, học viên sẽ so sánh kết quả độ đo F1 khi sử dụng mô hình CNN trước và sau khi tăng cường dữ liệu.

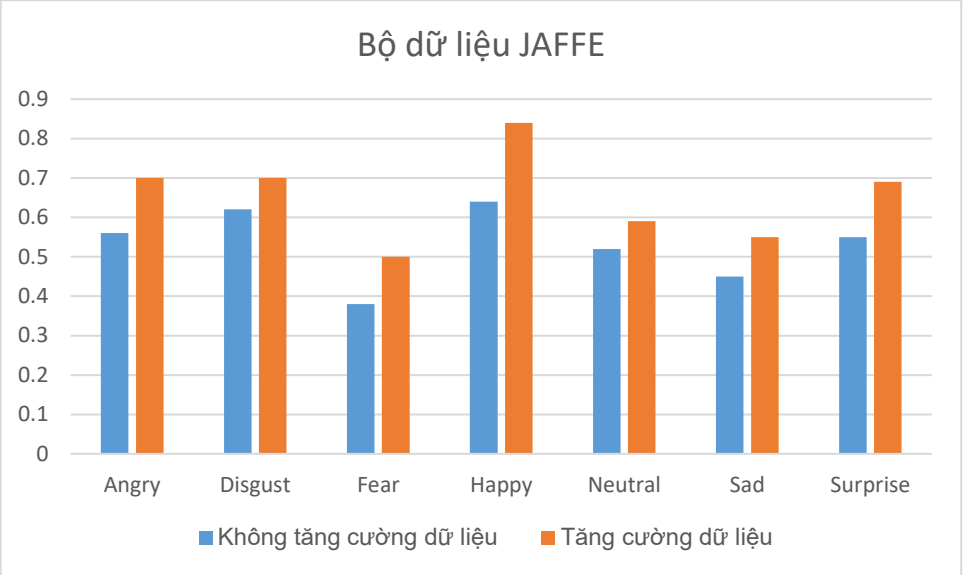


Hình 3. 10: Kết quả chi tiết các độ đo cho từng nhãn cảm xúc khi tăng cường dữ liệu và không tăng cường dữ liệu sử dụng mô hình CNN trên bộ dữ liệu CK+

Hình 3.10 trình bày kết quả chi tiết trên từng nhãn cảm xúc của mô hình CNN trên bộ dữ liệu CK+ trước khi và sau khi áp dụng phương pháp tăng cường dữ liệu. Nhìn một cách tổng quan chúng ta thấy được hầu hết phương pháp tăng cường giữ liệu đều giúp mô hình nâng cao hiệu quả trên tất cả các nhãn cảm xúc, trong đó các nhãn như “Fear” hay “Sad” cho thấy sự hiệu quả vượt trội, cụ thể nhãn “Fear” tăng +0.19, còn nhãn “Sad” tăng +0.15. Trong khi các nhãn đạt hiệu quả tốt khi sử dụng nguyên gốc dữ liệu như nhãn “Surprise” hay nhãn “Happy” vẫn có sự hiệu quả, tuy nhiên hiệu quả chênh lệch không đáng kể.

Trái ngược lại thì đối với bộ dữ liệu JAFFE, Hình 3.11 phương pháp tăng cường dữ liệu lại cho hiệu quả rõ rệt trên hầu hết các nhãn cảm xúc, đặc biệt nhất là

các nhãn “Happy”, “Angry” và “Surprise”. Bảng 3.6 trình bày chi tiết khoảng tăng cường hiệu quả của các nhãn cảm xúc trên cả hai bộ dữ liệu.



Hình 3. 11: Kết quả chi tiết các độ đo cho từng nhãn cảm xúc khi tăng cường dữ liệu và không tăng cường dữ liệu sử dụng mô hình CNN trên bộ dữ liệu JAFFE

Bảng 3. 6: Kết quả chi tiết độ đo F1 cho từng nhãn cảm xúc khi tăng cường dữ liệu và không tăng cường dữ liệu sử dụng mô hình CNN trên hai bộ dữ liệu

Nhãn	**Bộ dữ liệu JAFFE**			**Bộ dữ liệu CK+**		
	Không tăng cường	**Tăng cường**	**Độ lệch**	**Không tăng cường**	**Tăng cường**	**Độ lệch**
Angry	0.56	0.70	+0.14	0.61	0.73	+0.12
Disgust	0.62	0.70	+0.08	0.88	0.92	+0.04
Fear	0.38	0.50	+0.12	0.45	0.64	+0.19
Happy	0.64	0.84	+0.20	0.89	0.92	+0.03
Neutral	0.52	0.59	+0.07	0.81	0.86	+0.05
Sad	0.45	0.55	+0.10	0.46	0.61	+0.15
Surprise	0.55	0.69	+0.14	0.93	0.94	+0.01

3.6 Điều chỉnh tiền xử lý

Trong phần này học viên sẽ đánh giá các phương pháp tiền xử lý khác nhau trên cả hai bộ dữ liệu là CK+ và JAFFE. Các bước tiền xử lý được so sánh đánh giá như sau:

- **RotateCorrection (rc):** Phương pháp này sẽ xoay ảnh chụp gương mặt bị nghiêng thẳng đứng trở lại. Cụ thể: xác định vị trí của 2 mắt và điểm trung tâm giữa 2 mắt làm điểm trung tâm. Thực hiện phép xoay để đường nối 2 mắt hợp với phương ngang 1 góc là 0 độ.
- **ImageCropping (ic):** Phương pháp này sẽ thực hiện lấy đúng các phần pixel trên ảnh gương mặt và loại bỏ nền xung quanh. Cụ thể: xác định khoảng cách giữa 2 mắt. Thực hiện cắt với các kích thước xác định như sau: chiều cao vùng cắt có hệ số 4.5 gồm 1.3 cho vùng trên mắt và 3.2 cho vùng dưới mắt, chiều rộng vùng cắt ngang có hệ số 2.4, các hệ số trên nhân với khoảng cách từ giữa 2 mắt đến mắt phải để có kích thước thật.
- **DownSampling (ds):** Phương pháp dùng để giảm kích thước ảnh về kích thước phù hợp với đầu vào mạng CNN. Sử dụng linear interpolation để giảm kích thước ảnh thành 32x32.
- **ConvertColor (cc):** Đây là phương pháp chuyển tất cả màu sắc các hình ảnh về cùng một mức màu. Trong thí nghiệm của đề tài, học viên chuyển tất cả ảnh về ảnh mức độ xám.
- **IntensityNormalization (im):** Các yếu tố về brightness và contrast gây sai khác ở các ảnh của cùng lớp biểu cảm và cùng đối tượng. Do dó, cần giảm ảnh hưởng brightness và contrast để giảm độ phức tạp cho bộ phân lớp bằng intensity normalization. Intensity normalization là thực hiện chuẩn hóa về intensity bằng phương pháp contrastive equalization tại mỗi pixel: xác định kernel xung quanh mỗi pixel, tính σ và μ tại kernel đó. Giá trị mới theo công

thức $x' = \frac{x - \mu_{nhgx}}{\sigma_{nhgx}}$ với x' là giá trị mới và x là giá trị ban đầu. Với mô hình đề xuất, sử dụng contrastive equalization với kernel 7x7.

- **ImageToArrayPreprocessor (iap):** một hàm chuyển đổi để chuyển đổi ảnh số thành ma trận numpy, dùng để tính toán, huấn luyện mô hình.
- **FaceDetector (fd):** bộ phát hiện gương mặt bằng mô hình pretrained Haarcascades, trong đó đầu vào là một ảnh mức xám (Grayscale image).
- **EyesDetector (ed):** Phát hiện vị trí mắt trên gương mặt bằng mô hình pretrained haarcascade_eye, nhận ảnh đầu vào là ảnh mức xám.
- **DataGenerator (dg):** Bộ tăng cường dữ liệu học bằng elastic distortions.

Ở phần này, học viên sẽ nhận xét kết quả từng phương pháp tiền xử trên cả hai bộ dữ liệu. Kết quả được báo cáo là độ đo F1 chạy trên phương pháp đánh giá chéo (cross validation). Đầu tiên chúng ta thấy được sự khác biệt giữa các nhãn cảm xúc của từng phương pháp tiền xử lý khác nhau. Đối với nhãn "Angry", phương pháp "rc" khi áp dụng đơn lẻ đạt kết quả cao nhất với độ đo F1 là 0.60, kết quả này tương tự như nhãn "Disgust", nhãn "Fear" và nhãn "Happy" khi phương pháp tiền xử lý rc đạt hiệu quả tốt nhất với kết quả lần lượt là 0.90, 0.46 và 0.91. Ngược lại thì đối với hai nhãn là "Sad" và "Suprise" thì bước tiền xử lý "ds" và "iap" đều cho kết tốt nhất nhất tương ứng với hai nhãn này. Kết quả là lần lượt 0.45 và 0.92 cho nhãn "Sad" và "Surprise". Còn đối với nhãn "Neutral" thì cả hai phương pháp tiền xử lý là "ic" và "dg" đều cho kết quả tốt nhất với 0.81. Do đó chúng ta thấy rằng mỗi phương pháp tiền xử lý đều cho hiệu quả khác nhau với từng nhãn trong dữ liệu. Cũng dựa vào bảng 3.1, khi kết hợp tất cả các bước tiền xử lý lại với nhau, chúng ta có thể thấy đối với bộ dữ liệu này, sẽ có một số nhãn chúng ta đạt hiệu quả tốt hơn khi áp dụng riêng lẻ các phương pháp tiền xử lý. Cụ thể như khi áp dụng tất cả các phương pháp tiền xử lý, chúng ta đạt kết quả cao hơn tại hầu hết các nhãn từ nhãn "Fear", trong đó chúng ta đạt 0.63 đối với nhãn Angry, 0.91 với nhãn Disgust, 0.92 với nhãn Happy, 0.81 với nhãn "Neutral", 0.46 với nhãn Sad và 0.93 với nhãn Surprise. Còn lại nhãn "Fear" thì thấp hơn với 0.43.

Bảng 3. 7: Kết quả chi tiết các phương pháp tiền xử lý khác nhau trên bộ dữ liệu CK+

Tiền xử lý	Nhãn cảm xúc khuôn mặt						
	Angry	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**	**Surprise**
rc	0.60	0.90	**0.46**	0.91	0.80	0.35	0.89
ic	0.57	0.88	0.45	0.90	0.81	0.42	0.91
ds	0.58	0.88	0.43	0.89	0.80	0.45	0.91
cc	0.56	0.85	0.41	0.88	0.79	0.43	0.91
im	0.52	0.89	0.45	0.87	0.79	0.39	0.91
iap	0.54	0.86	**0.46**	0.87	0.79	0.42	0.92
fd	0.55	0.87	0.42	0.87	0.80	0.39	0.90
ed	0.55	0.89	0.40	0.89	0.79	0.38	0.91
dg	0.59	0.87	0.44	0.90	0.81	0.37	0.91
All	**0.63**	**0.91**	0.43	**0.92**	**0.81**	**0.46**	**0.93**

Tương tự như vậy, bảng 3.8 trình bày kết quả của các tiền phương pháp xử lý đầu vào trên bộ dữ liệu JAFFE. Tương tự như bộ dữ liệu CK+, thì mỗi phương pháp tiền xử lý trên bộ dữ liệu đều cho các hiệu quả khác nhau trên từng nhãn cảm xúc. Tuy nhiên khác với bộ CK+ thì các phương pháp tiền xử lý khác lại hiệu quả với các nhãn như phương pháp "im" đạt hiệu quả tốt nhất trên nhãn "Disgust" với giá trị F1 là 0.63, phương pháp "cc" đạt hiệu quả tốt nhất với độ chính xác là 0.65, trong khi đó phương pháp "ic" đạt kết quả kết quả tốt nhất trên nhãn "Surprise". Các kết quả từng phương pháp này còn cho kết quả tốt hơn kết hợp tất cả các phương pháp tiền xử lý với nhau. Tuy nhiên, chúng ta vẫn thấy rằng kết hợp các phương pháp lại vẫn cho kết quả tốt hơn trên một số nhãn như "Angry" với giá trị là 0.56, nhãn "Fear" với độ đo F1 là 0.38, nhãn "Neutral" với giá trị 0.52 và nhãn "Sad" 0.45. Nhìn chung, khi kết hợp các tiền xử lý cùng nhau, đều cho kết quả tốt hơn.

Bảng 3. 8: Kết quả chi tiết các phương pháp tiền xử lý khác nhau trên bộ dữ liệu JAFFE

Tiền xử lý	Nhãn cảm xúc khuôn mặt						
	Angry	**Disgust**	**Fear**	**Happy**	**Neural**	**Sad**	**Surprise**
rc	0.53	0.50	0.33	0.63	0.37	0.28	0.51
ic	0.46	0.58	0.29	0.62	0.32	0.35	**0.62**
ds	0.40	0.35	0.30	0.54	0.50	0.41	0.61
cc	0.52	0.57	0.25	**0.65**	0.48	0.47	0.58
im	0.53	**0.63**	0.28	0.62	0.41	0.44	0.52
iap	0.55	0.45	0.23	0.56	0.37	0.30	0.54
fd	0.52	0.43	0.29	0.57	0.35	0.40	0.51
ed	0.49	0.44	0.30	0.58	0.47	0.29	0.57
dg	0.44	0.54	0.27	0.55	0.46	0.29	0.49
All	**0.56**	0.62	**0.38**	0.64	**0.52**	**0.45**	0.55

3.7 So sánh kết quả mô hình CNN và DCNN

3.7.1 Tăng số lượng lớp tích chập – Convolution layer

Kỹ thuật tích chập Convolution là một tập hợp của các lớp hoạt động để rút trích thông tin của bước ảnh trước khi đưa kiến trúc mạng nơ-ron để xác định nhãn của nó. Các lớp tích chập – Convolution layer được sử dụng để giúp máy tính xác định các tính năng có thể bị bỏ sót khi chỉ đơn giản là làm phẳng một hình ảnh thành các giá trị pixel của nó. Mọi lớp bộ lọc đều được thiết kế để nắm bắt các thông tin khác nhau trong bức ảnh. Ví dụ: lớp đầu tiên của bộ lọc nắm bắt các mẫu như cạnh, góc, chấm. Các lớp tiếp theo kết hợp các mẫu đó để tạo ra các mẫu lớn hơn (như kết hợp các cạnh để tạo hình vuông, hình tròn). Đó là lý do tại sao học viên tăng kích thước bộ lọc trong các lớp tiếp theo để thu được nhiều kết hợp nhất có thể.

3.7.2 Áp dụng kỹ thuật dropout và batch normalization

Dropout là kỹ thuật được sử dụng để ngăn chặn quá trình overfitting trong mô hình học sâu. Kỹ thuật này được thêm vào trong kiến trúc mô hình loại bỏ ngẫu nhiên một số node neurons của kiến trúc mạng. Khi một số node bị tắt thì các trọng số và sự kết nối đến và đi đến đó cũng bị ngắt theo. Điều này được thực hiện để nâng cao khả năng học hỏi của mô hình và giúp mô hình có hiệu suất tốt hơn.

Batch normalization là một lớp cho phép mọi lớp của mạng thực hiện việc học một cách độc lập hơn. Nó được sử dụng để chuẩn hóa đầu ra của các lớp trước đó. Các kích hoạt mở rộng quy mô lớp đầu vào trong quá trình chuẩn hóa. Sử dụng phương pháp học chuẩn hóa hàng loạt trở nên hiệu quả và nó cũng có thể được sử dụng như quá trình chính quy hóa để tránh trang bị quá mức cho mô hình. Lớp được thêm vào mô hình tuần tự để chuẩn hóa đầu vào hoặc đầu ra. Lớp này có thể được sử dụng tại một số điểm giữa các lớp của mô hình. Nó thường được đặt ngay sau khi xác định mô hình tuần tự và sau các lớp tích chập và gộp.

3.7.3 Mô hình

Dựa trên mô tả ở trên, trong phần này học viên sẽ mô tả kiến trúc bổ sung mô hình CNN ở chương 2 bằng việc bổ sung thêm các lớp tích chập - covolution layer với các bộ lọc có kích thước khác nhau. Bên cạnh đó, học viên cũng bổ sung thêm các kỹ thuật giảm overfitting như Dropout, Batch Normalization để tăng độ chính xác. Học viên gọi mô hình này là mô hình Deep Convolution Neural Network – DCNN để so sánh với mô hình CNN ở chương 2.

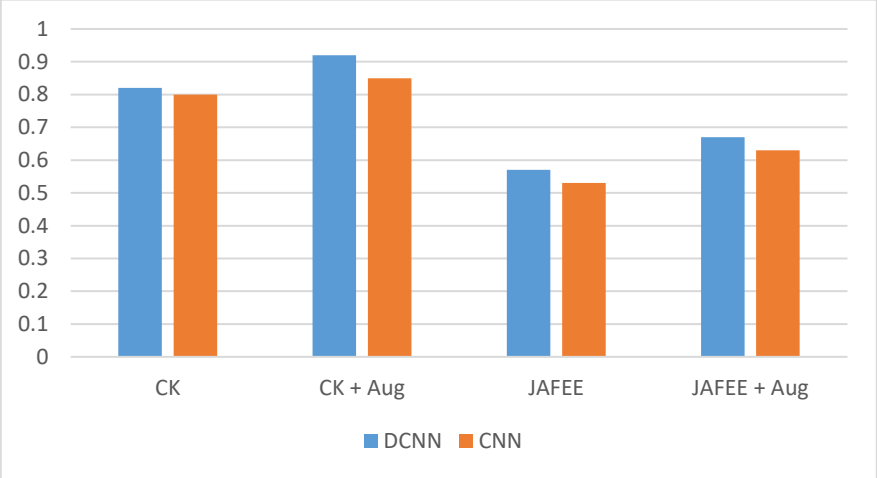
- **Kết quả thí nghiệm**

Ở phần này, học viên sẽ so sánh kết quả thí nghiệm mô hình cải tiến DCNN và mô hình CNN được trình bày ở chương 2 trên cả hai bộ dữ liệu CK+ và JAFFE gốc và khi tăng cương dữ liệu. Kết quả được báo cáo trên các độ đo như độ chính

xác, độ phủ và chỉ số F1-score. Bảng 3.9 và Hình 3.8 trình bày kết quả so sánh giữa hai mô hình DCNN và mô hình CNN trên cả hai bộ dữ liệu trước và sau khi tăng cường dữ liệu.

Bảng 3. 9: Kết quả các độ đo DCNN trên hai bộ dữ liệu gốc và sau khi tăng cường dữ liệu

Bộ dữ liệu	Loại dữ liệu	Precision	Recall	F1-score
CK+	Dữ liệu gốc	0.83	0.82	**0.82**
	Dữ liệu tăng cường	0.92	0.92	**0.92**
JAFFE	Dữ liệu gốc	0.59	0.55	**0.57**
	Dữ liệu tăng cường	0.71	0.63	**0.67**

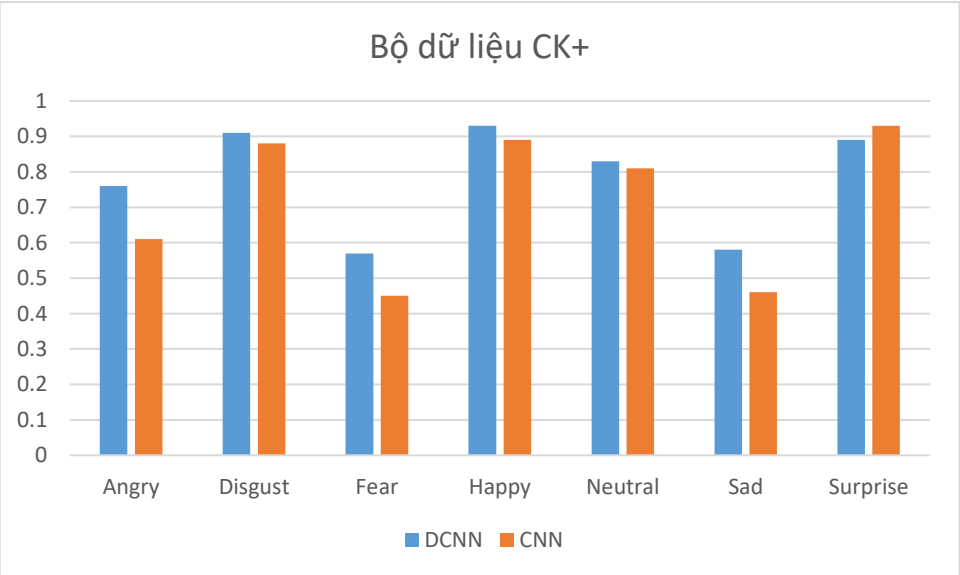


Hình 3. 12: Kết quả độ đo F1 giữa mô hình DCNN và CNN trên hai bộ dữ liệu gốc và tăng cường dữ liệu

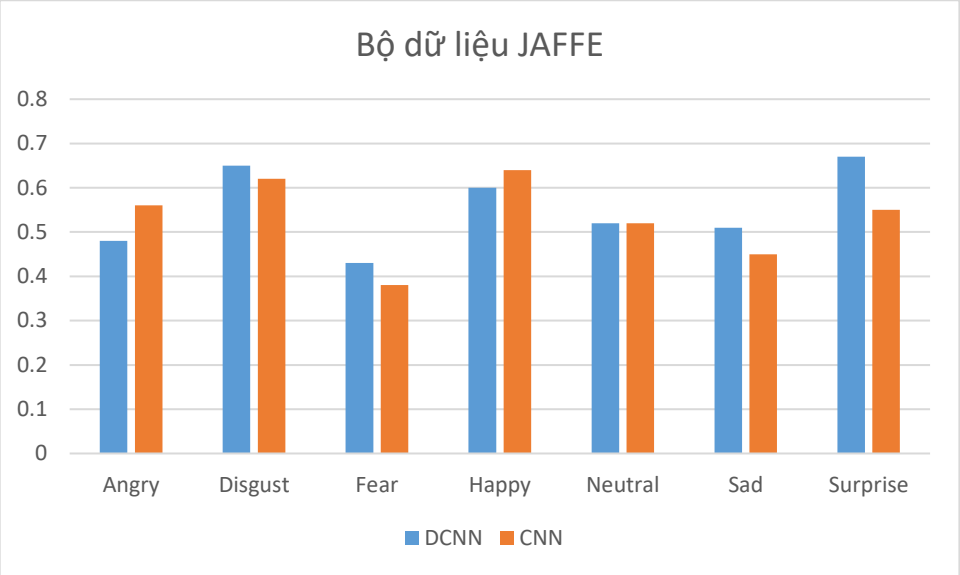
Nhìn vào bảng số liệu 3.9 và hình 3.12, chúng ta thấy mô hình cải tiến DCNN với thêm các lớp tích chập để rút nhiều thông tin hơn từ bức ảnh và áp dụng 2 kỹ thuật giảm overfiting trong quá trình huấn luyện mô hình đều cho kết quả tốt hơn mô hình CNN. Cụ thể đối với bộ dữ liệu gốc CK+, mô hình DCNN đạt độ chính xác 0.83, độ phủ là 0.82 còn giá trị F1 là 0.82, so với mô hình CNN thì mô hình cải tiến này

cao hơn +0.02. Còn đối với bộ dữ liệu CK+ khi tăng cường dữ liệu thì mô hình DCNN cao hơn mô hình CNN +0.07 về độ đo F1. Trong khi đó với bộ dữ liệu JAFFE, mô hình DCNN cũng chứng minh sự hiệu quả khi đối với dữ liệu gốc, mô hình này cao hơn mô hình CNN +0.03, còn trên dữ liệu tăng cường thì mô hình +0.04. Nhìn một cách tổng quan, chúng ta thấy mô hình DCNN đều cho kết quả tốt hơn mô hình CNN. Hình 3.13 và Hình 3.14 trình bày kết quả chi tiết các độ đo của mô hình DCNN và mô hình CNN trên cả hai bộ dữ liệu CK+ và bộ dữ liệu JAFFE.

Nhìn vào hình 3.13, chúng ta thấy rằng mô hình DCNN giúp tăng hiệu quả hơn các mô hình CNN hầu hết trên tất cả các nhãn trừ nhãn Surprise. Đặc biệt là đối với hai nhãn có tỷ lệ nhãn ít và độ chính xác thấp như "Fear" và nhãn "Sad". Mô hình DCNN giúp tăng hiệu quả +0.12 cho cả hai nhãn này so với mô hình CNN. Từ đó cho thấy rằng mình bổ sung thêm các lớp tích chập và thêm các lớp giảm overfiting trong mô hình sẽ giúp tăng hiệu quả trên các nhãn ít dữ liệu và hiệu quả thấp. Tương tự như vậy đối với bộ dữ liệu JAFFE, mô hình DCNN cũng cao hơn một số nhãn cảm xúc ngoài trừ nhãn "Angry" và nhãn "Happy".



Hình 3. 13: Kết quả các độ đo của mô hình DCNN và mô hình CNN trên bộ dữ liệu gốc CK+



Hình 3. 14: Kết quả các độ đo của mô hình DCNN và mô hình CNN trên bộ dữ liệu gốc JAFFE

3.8 Kết luận của chương 3

Trong chương 3 này, học viên đã trình bày chi tiết hai bộ dữ liệu được sử dụng để thực nghiệm trong đề tài này là bộ dữ liệu CK+ và bộ dữ liệu JAFFE. Sau đó, học viên đã nhận xét và so sánh sự hiệu quả của hai mô hình CNN và DCNN trên cả hai bộ dữ liệu theo các độ đo độ chính xác – accuracy, độ phủ - recall, độ chính xác - precision và độ đo F1. Ngoài ra học viên còn phân tích xem tỷ lệ nhầm lẫn giữa các nhãn trong hai bộ dữ liệu để kiểm tra xem là nhãn nào hay bị dự đoán sai nhất. Cuối cùng học viên so sánh kết quả của hai mô hình DCNN và mô hình CNN trên hai bộ dữ liệu. Kết quả thí nghiệm cho thấy rằng mô hình DCNN cho kết quả cao hơn so với mô hình CNN trên một số nhãn ít dữ liệu.

CHƯƠNG 4: ỨNG DỤNG

4.1 Ứng dụng phát hiện cảm xúc khuôn mặt

Để xây dựng ứng dụng phát hiện cảm xúc khuôn mặt trên webcam của màn hình laptop, em sử dụng ngôn ngữ Python kết hợp với thư viện OpenCV để viết viết chương trình minh họa hỗ trợ cho việc xử lý dữ liệu đầu vào từ webcam. Quá trình xử lý qua 5 bước như sau:

Bước 1: Ảnh đầu vào được chuyển thành đa cấp xám.

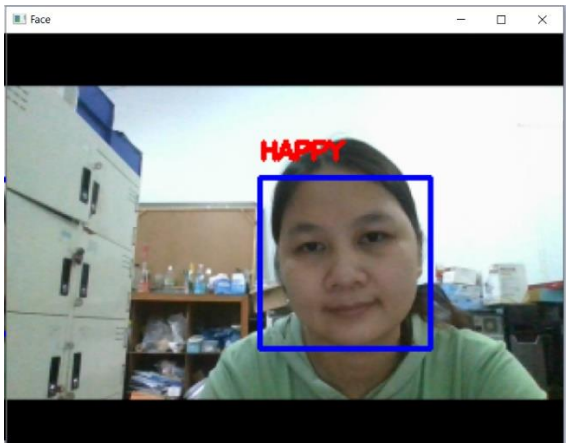
Bước 2: Dùng haar cascade của thư viện OpenCV để tìm kiếm vùng mặt người trên ảnh đầu vào, sau khi cắt được vùng khuôn mặt thì sẽ được chuyển đến bước 3.

Bước 3: Vùng ảnh mặt người được chuyển đổi về kích thước 32x32 bởi vì đây là kích thước đầu vào của việc huấn luyện các mô hình CNN trên các bộ dữ liệu.

Bước 4: Ảnh sau khi đã được chuẩn hóa thành kích thước 32x32 đa cấp xám chuyển đổi về miền [0, 1] sau đó đưa vào mô hình CNN đã được huấn luyện sẵn để tiến hành dự đoán nhãn cảm xúc.

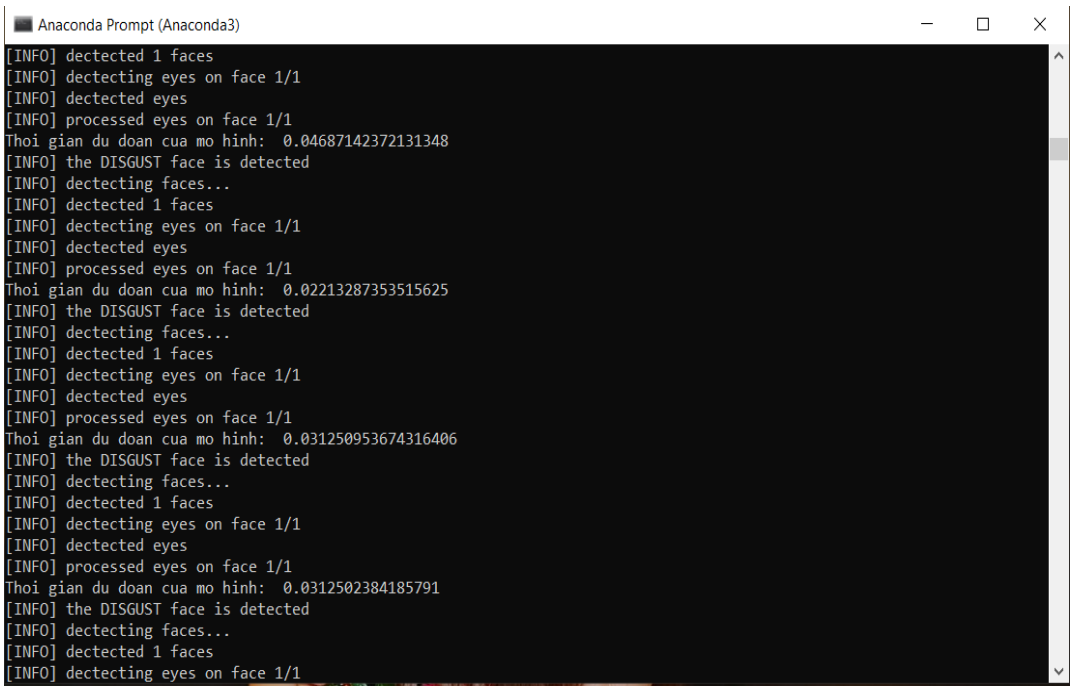
Bước 5: Đầu ra của CNN là xác suất của các cảm xúc, chọn cảm xúc có xác xuất cao nhất làm kết quả cuối cùng.

Kết quả chạy thử nghiệm thực tế cho thấy rằng mô hình dự đoán khá nhạy với nhãn cảm xúc “Happy” và khó xác định được với nhãn “Angry” bởi vì có thể do biểu diễn cảm xúc khuôn mặt của học viên không giống với các dữ liệu trong bộ huấn luyện. Ở trong phần này học viên lựa chọn mô hình CNN được huấn luyện trên bộ CK+, JAFFE bởi vì mô hình này đạt kết quả tương đối tốt và số lượng tham số ít hơn mô hình DCNN. Điều này sẽ đảm bảo việc xử lý khi sử dụng Webcam trực tiếp từ máy tính với cấu hình của laptop đang sử dụng. Dưới đây là hình ảnh minh họa chạy thực tế của học viên.



Hình 4. 1: Kết quả dự đoán mô hình CNN trên thử nghiệm thực tế đối vỡi nhãn "Happy"

Thời gian dự đoán mô hình CNN trên thử nghiệm thực tế khi nhận diện một khung hình để xác định cảm xúc, tính từ lúc webcam nhận hình đưa vào mô hình đến lúc ra kết quả mất trung bình 0.03s. Hình 4.2 là thời gian chạy thực nghiệm khi nhận dạng cảm xúc khuôn mặt qua webcam.



Hình 4. 2: Thời gian dự đoán mô hình CNN trên thử nghiệm thực tế

4.2 Kết luận chương 4

Trong chương 4 này, học viên đã trình bày một ứng dụng minh họa sử dụng trên webcam của máy tính để phát hiện và phân loại cảm xúc khuôn mặt. Với sự hỗ trợ của thư viện OpenCV trong việc xác định vùng chứa khuôn mặt, học viên lấy kết quả này đưa qua mô hình CNN đã được huấn luyện sẵn để tiến hành dự đoán nhãn cảm xúc rồi xuất lên màn hình.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả nghiên cứu của luận văn

Trong luận văn này, học viên đã tìm hiểu, khảo sát các kỹ thuật Học máy, Học sâu cũng như các kỹ thuật tiền xử lý. Dựa trên các tìm hiểu, học viên kiểm chứng một số phương pháp tiền xử lý ảnh và tăng cường mẫu học và mạng CNN sâu cải tiến cho bài toán nhận diện cảm xúc gương mặt đạt hiệu quả thời gian thực, hai bộ dữ liệu dùng để huấn luyện và đánh giá là JAFFE và CK+. Trong đó, các kết quả thực nghiệm ở Bảng 3.5 cho thấy việc áp dụng kỹ thuật tăng cường dữ liệu có ảnh hưởng tích cực rõ rệt thì độ chính xác thu được cao hơn khi huấn luyện với tập ảnh gốc ở cả hai bộ dữ liệu. Phương pháp tiền xử lý cũng chứng minh sự hiệu quả, trong đó các kết quả cho thấy việc áp dụng tất cả các kỹ thuật tiền xử lý được trình bày ở phần 2.1 sẽ hiệu quả hơn so với thực hiện riêng lẽ từng phương pháp.

Các thử nghiệm trình bày ở Bảng 3.9 cũng cho thấy kiến trúc mạng DCNN sâu đề xuất bởi học viên cho kết quả tốt trên cả hai bộ dữ liệu CK+ và JAFFE, tuy nhiên học viên sử dụng kiến trúc mạng CNN đơn giản ở Chương 2 để xây dựng hệ thống demo phát hiện cảm xúc để phù hợp với yêu cầu thời gian thực với cấu hình máy laptop cá nhân trong môi trường thực tế.

Học viên cũng đã xây dựng một ứng dụng minh họa phát hiện cảm xúc khuôn mặt chạy trong môi trường thực tế với thời gian thực.

5.2 Những hạn chế trong luận văn

Bên cạnh những gì học viên đã đạt được trong quá trình làm luận văn cũng tồn tại những hạn chế mà cần phải được nghiên cứu và phát triển trong tương lai. Học viên hiện tại chỉ kiểm tra và đánh giá trên bộ dữ liệu chuẩn đã được công bố cho nghiên cứu là bộ dữ liệu CK+ và JAFFE. Kết quả thử nghiệm trên bộ dữ liệu JAFFE chưa đạt kết quả như mong đợi, còn thấp khá nhiều so với bộ dữ liệu CK+.

5.3 Hướng phát triển

Thử nghiệm trên nhiều bộ dữ liệu để đạt kết quả cao hơn như Fer 2013, VGG 16, Resnet … với hệ thống máy có cấu hình mạnh hơn.

Xây dựng được hệ thống hỗ trợ phân biệt cảm xúc bệnh nhân có độ chính xác cao và đáp ứng các yêu cầu về thời gian thực, phục vụ cho việc khám chữa bệnh tại Bệnh Viện Đa Khoa Tây Ninh để nâng cao trải nghiệm người dùng.

TÀI LIỆU THAM KHẢO

[1] Jabon, Maria, et al. "Facial expression analysis for predicting unsafe driving behavior." IEEE Pervasive Computing 10.4 (2010): 84-95.

[2] Kapoor, Ashish, Winslow Burleson, and Rosalind W. Picard. "Automatic prediction of frustration." *International journal of human-computer studies* 65.8 (2007): 724-736.

[3] Lankes, M.; Riegler, S.; Weiss, A.; Mirlacher, T.; Pirker, M.; Tscheligi, M. Facial expressions as game input with different emotional feedbaCK+ conditions. In Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, Yokohama, Japan, 3–5 December 2008; pp. 253–256

[4] Li, Shan, Weihong Deng, and JunPing Du. "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[5] Huang, Yunxin, et al. "Facial expression recognition: A survey." *Symmetry* 11.10 (2019): 1189.

[6] Li, Shan, and Weihong Deng. "Deep facial expression recognition: A survey." *IEEE Transactions on Affective Computing* (2020).

[7] Barsoum, Emad, et al. "Training deep networks for facial expression recognition with crowd-sourced label distribution." *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. 2016.

[8] Y. Chen, J. Wang, Z. Shi và S. Chen, "Facial Motion Prior Networks for Facial Expression Recognition," arXiv, 2019.

[9] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. OReilly và Y. Tong, "Island Loss for Learning Discriminative Features in Facial Expression Recognition," Automatic

[10] Y. Wen, K. Zhang, Z. Li và Y. Qiao, “A Discriminative Feature Learning Approach for Deep Face Recognition,” ECCV 2016: Computer Vision – ECCV 2017, tập 9911, pp. 499-515, 2017.

[11] Lucey, Patri CK+, et al. "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression." 2010 ieee computer society conference on computer vision and pattern recognition-workshops. IEEE, 2010.

[12] D. Meng, X. Peng, K. Wang và Y. Qiao, “FRAME ATTENTION NETWORKS FOR FACIAL EXPRESSION RECOGNITION IN VIDEOS,” arxiv, 2019.

[13] Lopes, André Teixeira, et al. "Facial expression recognition with convolutional neural networks: coping with few data and the training sample order." Pattern recognition 61 (2017): 610-628.

[14] Hou, Qiqi, et al. "Facial landmark detection via cascade multi-channel convolutional neural network." 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015.

[15] Xiao, Shengtao, Shuicheng Yan, and Ashraf A. Kassim. "Facial landmark detection via progressive initialization." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.

[16] Altenberger, Felix, and Claus Lenz. "A non-technical survey on deep convolutional neural network architectures." *arXiv preprint arXiv:1803.02129* (2018).

[17] *Cao, Changyu et al. “A convolutional neural network face recognition algorithm based on data augmentation.” (2019).*

[18] Wang Q., Xiong D., Alfalou A., Brosseau C. Optical image authentication scheme using dual polarization decoding configuration. Opt. Lasers Eng. 2019;112:151–161. doi: 10.1016/j.optlaseng.2018.09.008.

[19] Vinay A., Hebbar D., Shekhar V.S., Murthy K.B., Natarajan S. Two novel detector-descriptor based approaches for face recognition using sift and surf. Procedia Comput. Sci. 2015;70:185–197

[20] Du G., Su F., Cai A. MIPPR 2009: Pattern Recognition and Computer Vision. Volume 7496. SPIE; Bellingham, WA, USA: 2009. Face recognition using SURF features; p. 749628. International Society for Optics and Photonics.

[21] Napoléon T., Alfalou A. Pose invariant face recognition: 3D model from single photo. Opt. Lasers Eng. 2017;89:150–161. doi: 10.1016/j.optlaseng.2016.06.019

[22] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid kpca and svm with ga model for intrusion detection," Applied Soft Computing, vol. 18, pp. 178–184, 2014.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar và I. Matthews, "The extended cohnkanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, San Francisco, CA, pp. 94-101, 2019.

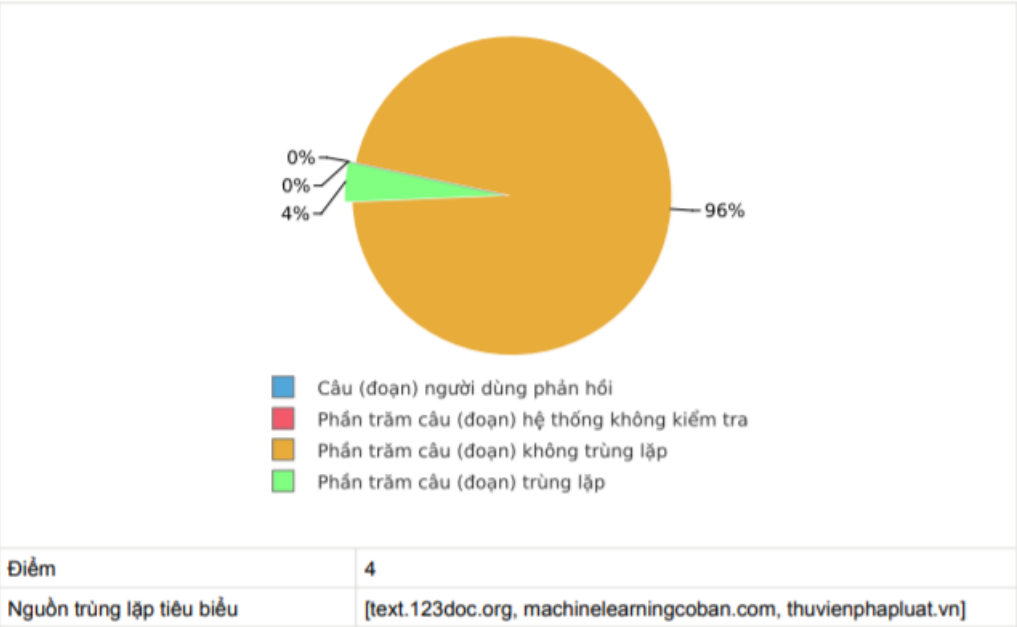
Hệ thống hỗ trợ nâng cao chất lượng tài liệu

KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU

THÔNG TIN TÀI LIỆU

Tác giả	VÕ THỊ HỒNG NHUNG
Tên tài liệu	Phân tích biểu cảm mặt người dùng mạng nơ ron tích chập
Thời gian kiểm tra	09-12-2021, 04:34:08
Thời gian tạo báo cáo	09-12-2021, 04:36:20

KẾT QUẢ KIỂM TRA TRÙNG LẶP



Điểm	4
Nguồn trùng lặp tiêu biểu	[text.123doc.org, machinelearningcoban.com, thuvienphapluat.vn]

Học Viên **Người hướng dẫn Khoa học**

Võ Thị Hồng Nhung **PGS.TS. Lê Hoàng Thái**

BẢNG CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng **4%** toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Tp. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học Viên Thực Hiện Luận Văn

Võ Thị Hồng Nhung