TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

NHẬN DIỆN CẢM XÚC KHUÔN MẶT DÙNG MẠNG NƠ – RON TÍCH CHẬP CNN

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - TRUYỀN THÔNG

Sinh viên: **HOÀNG THỊ MINH HIẾU**

MSSV: 15141158

NGUYỄN THỊ HỒNG VY

MSSV: 15141335

TP. HỒ CHÍ MINH – 06/2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

NHẬN DIỆN CẢM XÚC KHUÔN MẶT DÙNG MẠNG NƠ – RON TÍCH CHẬP CNN

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

Sinh viên: **HOÀNG THỊ MINH HIẾU**
MSSV: 15141158
NGUYỄN THỊ HỒNG VY
MSSV: 15141335

Hướng dẫn: **ThS. TRƯƠNG QUANG PHÚC**

TP. HỒ CHÍ MINH – 06/2019

LỜI NÓI ĐẦU

Ngày nay, trí tuệ nhân tạo được sử dụng rộng rãi trong hầu hết các ngành công nghiệp. Đặc biệt là trong lĩnh vực tiếp thị khách hàng, trí tuệ nhân tạo phát triển mạnh mẽ, với khả năng xử lý thông tin nhanh chóng đưa thương hiệu đến gần với khách hàng. Máy học sẽ giúp trí tuệ nhân tạo đạt được mục tiêu đó bằng cách thu thập và phân tích dữ liệu trong thời gian ngắn.

Khi lượng khách hàng đến mua sắm ngày càng tăng, các trung tâm thương mại mọc lên nhiều hơn để đáp ứng nhu cầu tiêu dùng của khách hàng. Các doanh nghiệp rất khó nắm bắt chính xác tâm lý, nhu cầu người tiêu dùng đối với các sản phẩm được bày bán hoặc các sản phẩm dùng thử mà họ đã tung ra thị trường, liệu rằng những sản phẩm đó có đem đến sự hài lòng cho người tiêu dùng. Những cảm nhận về sản phẩm và dịch vụ sẽ biểu hiện rõ nét trên khuôn mặt của khách hàng và đó luôn là sự phản hồi chân thật nhất. Khi lắp đặt hệ thống nhận diện cảm xúc tương tác với khách hàng, việc thu thập thông tin phản hồi sẽ trở nên nhanh chóng. Các doanh nghiệp không cần phải gặp mặt trực tiếp, tham khảo ý kiến người tiêu dùng hoặc điều tra thị trường mà thay vào đó là xem đoạn video đã được lắp đặt tự động trong các khu mua sắm, dữ liệu từ hệ thống thu thập được có thể đánh giá chính xác mức độ hài lòng của khách hàng đối với sản phẩm và chất lượng phục vụ của nhân viên, từ đó cải thiện được mọi tương tác, tìm kiếm và gắn kết khách hàng với doanh nghiệp.

Để xây dựng hệ thống nhận diện khuôn mặt cần phải dựa vào kiến trúc học sâu sử dụng mạng Nơ – rơn tích chập CNN. Dữ liệu thu được từ webcam hay từ hình ảnh sẽ được định vị vùng khuôn mặt bằng phương pháp Haar Cascade từ thư viện OpenCV, sau đó chuyển vào mạng học sâu để xử lý và cuối cùng là trả về xác xuất của 7 loại cảm xúc. Kết quả thu được sẽ được lưu lại và dựa vào đó để kiểm tra, đánh giá mức độ hài lòng của khách hàng đối với sản phẩm từ các thương hiệu họ đang sử dụng.

LỜI CẢM ƠN

Để hoàn thành đồ án "Nhận diện cảm xúc khuôn mặt dùng mạng Nơ – ron tích chập CNN", nhóm tác giả xin chân thành cảm ơn sự hướng dẫn tận tình của thầy Trương Quang Phúc – Giảng viên khoa Điện – Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật thành phố Hồ Chí Minh.

Trong quá trình nghiên cứu, tìm hiểu và thực hiện đề tài không tránh khỏi những sai sót. Nhóm chúng tôi mong Thầy và các bạn góp ý để đề tài được hoàn thiện hơn và có thể ứng dụng nhiều hơn trong thực tế.

Một lần nữa chúng tôi xin chân thành cảm ơn!

ABSTRACT

In this thesis, the authors implemented facial emotion recognition using CNN convolution network. The goal that we want to achieve is extracting the areas of human face, following by processing the obtained image data and classify into 7 emotion labels (angry, surprised, scared, happy, neutral, sad, and disgust). Start with developing models, select optimal models with high accuracy and base on them to analyze emotions, display the results of identification on the design interface.

The popular data set on human face recognition used for model building is FER2013, obtains a model with an accuracy of 66%. Emotional analysis results on photos and videos returned on the interface are programmed from Python's Tkinter library.

The objective of this project is to develop a system which can identify emotions, apply at commercial centers and so as to evaluate customer satisfaction towards the product or service they are using. After studying, the topic of facial recognition has been completed relatively the set goal.

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG 1: TỔNG QUAN

1.1 GIỚI THIỆU

Ngày nay, sự phát triển của khoa học công nghệ đang dần thay đổi thế giới, khoa học công nghệ đóng vai trò quan trọng trong việc nâng cao năng suất lao động, hiệu quả sản xuất và thúc đẩy sự phát triển kinh tế ở mỗi quốc gia. Trong hầu hết các lĩnh vực: kinh tế, giáo dục, y tế, an ninh,..., khoa học công nghệ đã có những bước tiến mới. Từ những ứng dụng nghiên cứu phân tích gen trong y học và nông nghiệp; công nghệ quét giác mạc (Iris Scanner) có tính bảo mật cao trong an ninh mạng; hệ thống máy chủ (Server Farms) cung cấp dữ liệu đến các hệ thống vận chuyển hàng hóa thông minh như máy bay không người lái, xe tự hành, đường sắt trên không Hyperloop [1],..., đã góp phần đưa nền khoa học công nghệ thế giới sang bước tiến mới trong công nghiệp 4.0, đó là cuộc cách mạng kỹ thuật số sử dụng trí tuệ nhân tạo (Artificial Intelligence: AI) và máy học (Machine Learning: ML) cho các hệ thống.

AI dùng để nghiên cứu và phát triển những hệ thống máy tính có khả năng thực hiện những công việc vốn đòi hỏi bộ não con người như dịch thuật, nhận diện giọng nói hoặc biểu hiện cảm xúc theo hoàn cảnh. Tiềm năng của việc phát triển AI là rất lớn, chỉ trong vài phút nó có thể phân tích cùng một lúc hàng nghìn dữ liệu, và đưa ra những dự đoán hữu ích mà rất lâu con người mới có thể làm được. Việc xây dựng một hệ thống AI là cực kì phức tạp vì nó được xem như là một cỗ máy có thể bắt chước hành vi và tư duy của con người.

ML là tập hợp con của AI, một ngành học thuộc khoa học máy tính, giúp máy tính có khả năng tự học, khả năng nhận thức cơ bản của con người mà không phải lập trình một cách rõ ràng [2]. Hỗ trợ con người trong việc xử lý khối lượng thông tin khổng lồ. ML có thể tự tiếp thu kiến thức mới, tự cải thiện những lỗi sai và qua những tiếp xúc với con người. ML là một tính năng của AI được dùng để huấn luyện, hỗ trợ AI nhận biết các mẫu dữ liệu và dự đoán kết quả.

Cùng với sự phát triển về khoa học và công nghệ, máy móc đã dần thay thế con người trong nhiều công việc. Để có thể tương tác được với con người, máy

móc cần phải có các kỹ năng trao đổi thông tin với con người và một trong những kỹ năng đó là khả năng hiểu được cảm xúc. Công nghệ phân tích cảm xúc dựa trên AI và ML đã tạo ra các ứng dụng có khả năng xử lý ảnh theo chuỗi thời gian thực và bám sát đối tượng cần phân tích.

Ứng dụng nhận diện cảm xúc ngày càng được phát triển rộng rãi điển hình như ứng dụng camera gắn trong lớp học, có cài đặt phần mềm giúp giáo viên quản lý và kiểm tra sự thu hút trong bài giảng đối với các học viên. Các hệ thống kiểm tra độ chính xác thông tin, phần mềm điều khiển dựa vào cảm xúc hay các thiết bị hỗ trợ người tàn tật,... Từ đó, nhóm chúng tôi đã tìm hiểu về mạng Nơ - ron tích chập (Convolutional Neural Network: CNN) để phân tích cảm xúc đối tượng được nhận dạng.trực

1.2 MỤC TIÊU CỦA ĐỀ TÀI

Tìm hiểu kiến thức về thuật toán và phương pháp nhận diện cảm xúc từ khuôn mặt. Nghiên cứu và đánh giá các phương pháp nhận dạng mặt người với 7 cảm xúc cơ bản: vui (Happy), buồn (Sad), giận dữ (Angry), bình thường (Neutral), ngạc nhiên (Suprised), ghê tởm (Disgust) và sợ hãi (Scared).

Xây dựng hệ thống phát hiện khuôn mặt từ nguồn ảnh hoặc video theo thời gian thực trên webcam, phân tích cảm xúc trên khuôn mặt từ khung ảnh hiện tại và hiển thị kết quả trên giao diện thiết kế.

1.3 TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU

1.3.1 Ngoài nước

Hiện nay, công nghệ AI mang lại những hỗ trợ tối ưu cho các doanh nghiệp với nhiều ứng dụng trong thế giới thực, rất nhiều tập đoàn đã bước đầu thành công trong việc ứng dụng công nghệ học sâu vào các ứng dụng tiện ích.

Năm 2018, Google đã công bố các tính năng đặc biệt được bổ sung trong sản phẩm Google Photos, áp dụng hệ thống nhận diện hình ảnh. Ngoài tính năng đã có như lưu trữ, phân loại hình ảnh, Google Photos đã có thêm tính năng chuyển ảnh trắng đen sang ảnh màu, người dùng cũng có thể thay đổi màu sắc các chi tiết trong

ảnh theo sở thích chỉ với một vài thao tác đơn giản. Ngoài ra, còn có tính năng chuyển ảnh chụp sang định dạng PDF, tự động điều chỉnh độ tương phản, tự động xoay ảnh, chỉnh cân bằng sáng và chỉnh màu [3].

Tháng 07/2018, cổng thương mại điện tử Alibaba của Trung Quốc đã tạo ra ứng dụng AI Copywriter - một công cụ AI chuyên về học sâu, cho phép có thể phát sinh ra 20.000 mẫu quảng cáo mỗi giây. Người dùng có thể lựa chọn các ý tưởng để tạo ra quảng cáo của riêng mình, không phải mất nhiều thời gian cho việc viết lách mà thay vào đó là dành nhiều thời gian, công sức và trí tuệ hơn cho việc sáng tạo [4].

1.3.2 Trong nước

Lĩnh vực y học tại Việt Nam đã có những bước tiến lớn trong khoa học công nghệ, sau nhiều năm nghiên cứu và ứng dụng đã cho ra phần mềm MMS.Net. MMS.Net có nhiều tính năng nổi bật như sẵn sàng kết nối đến bất kỳ hệ thống nào trong y tế của quốc gia và quốc tế. MMS.Net có khả năng học máy, tự sinh các mô hình quản lý theo từng cấu hình cụ thể, tận dụng các giải thuật trong AI để giải quyết một số bài toán tối ưu. MMS.Net có nhiều phiên bản dành cho nhiều cơ sở khám chữa bệnh có quy mô khác nhau, từ trạm y tế đến phòng khám chuyên khoa và đa khoa. MMS.Net tích hợp với một số sản phẩm phần mềm khác thông qua việc trao đổi và chuyển giao công nghệ đã tích hợp hoàn thiện hệ thống quản lý thông tin bệnh nhân PHR và bệnh án điện tử EHR làm nền tảng cho việc phát triển một hệ sinh thái y tế thông minh trong giai đoạn 2020-2025 tại Việt Nam [5].

Các công ty trong nước cũng đang trên đà phát triển theo xu hướng công nghệ mới. Vào ngày 17/4/2019, Tập đoàn Vingroup chính thức công bố thành lập Viện Nghiên cứu Trí tuệ Nhân tạo AI - VinAI Research. Viện sẽ thực hiện các nghiên cứu khoa học đột phá trong lĩnh vực AI và máy học. Trước đó, vào tháng 10/2018 Vingroup đã ra mắt thành công 2 chiếc ô tô VinFast tại triển lãm Paris Motors Show. Đến tháng 12/2018, Tập đoàn Vingroup đã công bố 4 sản phẩm điện thoại thông minh Vinsmart, Vinsmart không sử dụng Android gốc mà tự tùy biến riêng cho mình một hệ điều hành dựa trên Android, điện thoại Vinsmart Active 1+ có

khả năng mở khóa nhận diện bằng khuôn mặt cực kì chuẩn xác, tương tác với người tiêu dùng [6].

1.4 PHƯƠNG PHÁP NGHIÊN CỨU

Nghiên cứu cơ sở lý thuyết về thuật toán nhận diện cảm xúc khuôn mặt sử dụng mạng Nơ - ron tích chập CNN. Mục đích của đề tài là lựa chọn thuật toán, phương pháp có độ chính xác tương đối để nhận diện cảm xúc và tăng độ chính xác cho mô hình.

Mạng Nơ - ron truyền thống (Neural Network: NN) hoạt động không thực sự hiệu quả với dữ liệu đầu vào là hình ảnh. Tuy nhiên, ngõ vào của mạng CNN là một hình ảnh chứ không có dạng vector như NN. Nếu đầu vào là một tập ảnh lớn thì NN không phải là lựa chọn hoàn hảo mà thay vào đó là CNN. Lớp tích chập trong CNN được dùng để phát hiện và trích xuất những đặc trưng xuất hiện trong ảnh. Lấy ảnh đầu vào nhân chập với bộ lọc kernel làm cho ảnh đầu ra có kích thước nhỏ hơn. Bộ lọc ở lớp chập càng sâu thì khả năng phát hiện các đặc trưng phức tạp càng lớn. Chính những lớp tích chập này làm CNN trở nên khác biệt so với mạng Nơ - ron truyền thống và hoạt động cực kỳ hiệu quả trong bài toán phân tích ảnh.

Nguồn dữ liệu thu thập là tập dữ liệu FER2013 lấy từ nguồn Kaggle. Mô hình trong mạng CNN có các tham số: dataset_path, test_size, train_size, validation_split, batch_size, epochs, num_classes, patience,... So sánh hệ số val_loss, val_acc từ các mô hình đã huấn luyện trước đó, dựa vào đó thay đổi các tham số batch_size, epochs và validation_split. Sau đó huấn luyện lại mô hình, tăng hệ số val_ acc và giảm hệ số val_loss nhằm tăng độ chính xác cho mô hình.

Dựa trên nền tảng lý thuyết đó để xây dựng một giao diện kiểm tra hệ thống, từ đó đưa ra nhận xét đánh giá các phương pháp đã tìm hiểu.

1.5 BỐ CỤC

Chương 1: Tổng quan

Trong chương này, nhóm tác giả nêu ra tình hình khoa học công nghệ hiện nay, xu hướng phát triển của AI. Đồng thời nói lên tính cấp thiết của đề tài, lý do

chọn đề tài. Từ đó, tiến hành nghiên cứu những phương pháp phù hợp với yêu cầu và mục tiêu đề tài.

Chương 2: Cở sở lý thuyết

Trình bày tổng quan về Deep Learning, mạng Nơ - ron tích chập và thuật toán CNN trong nhận diện cảm xúc trên khuôn mặt.

Chương 3: Thiết kế hệ thống

Nhóm tác giả trình bày về sơ đồ thiết kế hệ thống sử dụng mạng CNN. Nêu ra được cách xây dựng và huấn luyện mô hình CNN với những ràng buộc, yêu cầu cụ thể. Thiết kế giao diện nhận diện cảm xúc GUI từ thư viện Tkinter của Python.

Chương 4: Kết quả

Trình bày về kết quả nhận diện cảm xúc của một hoặc nhiều đối tượng từ ảnh và video theo thời gian thực, trong các trường hợp nhận diện đúng và không đúng phần được cho là khuôn mặt. So sánh kết quả, đánh giá ưu và nhược điểm của hệ thống.

Chương 5: Kết luận và hướng phát triển

Dựa theo kết quả có được từ chương 4, nhóm tác giả đưa ra kết luận về những vấn đề mà hệ thống đã đạt được và chưa đạt được. Từ những hạn chế trong đề tài, nhóm tác giả đề xuất hướng phát triển cải thiện hệ thống tối ưu hơn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 ẢNH SỐ

Ảnh số (digital image) là một tập hợp của nhiều điểm ảnh (pixel). Mỗi điểm ảnh biểu diễn một màu sắc nhất định (độ sáng với ảnh đen trắng) tại một điểm duy nhất, có thể xem một điểm ảnh giống như một chấm nhỏ trong một tấm ảnh màu [7].

Số điểm ảnh (pixel) xác định độ phân giải của ảnh. Ảnh càng có nhiều điểm ảnh thì độ phân giải càng cao, càng thể hiện rõ nét các đặc điểm của ảnh, ảnh trở nên thực và sắc nét hơn. Trong một ảnh, giá trị của một điểm ảnh (pixel) thường là từ 0-255 [8].

2.2 TIỀN XỬ LÝ

Thay đổi kích thước hình ảnh nghĩa là thay đổi chiều rộng, chiều cao hoặc cả chiều rộng lẫn chiều cao. Ngoài ra, tỷ lệ khung hình của hình ảnh gốc có thể được giữ nguyên trong hình ảnh được chỉnh kích thước.

Để thay đổi kích thước hình ảnh, OpenCV cung cấp hàm cv2.resize ().

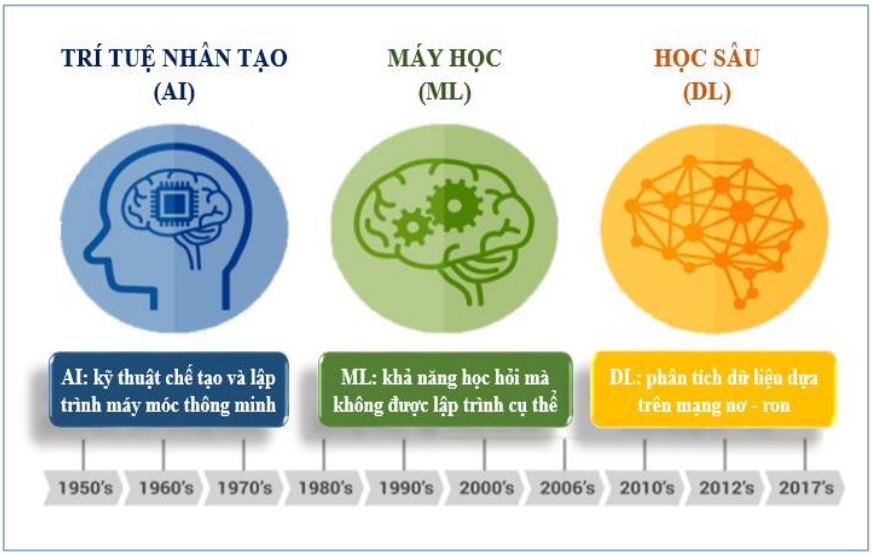
i) Ảnh gốc (ii) Ảnh sau khi thay đổi kích thước

Hình 2.1: Thay đổi kích thước ảnh

Trong hình 2.1, ảnh ngõ vào có kích thước ảnh là 480 x 361 pixel, ảnh ngõ ra thay đổi kích thước là 600x300 pixel, độ tương phản của ảnh vẫn giữ nguyên. Ảnh giữ nguyên đặc trưng nếu thay đổi kích thước vừa phải (không quá chênh lệch với ảnh đầu vào), ngược lại ảnh sẽ bị mờ do các pixel dãn ra hoặc các pixel chồng lên nhau.

2.3 TỔNG QUAN VỀ DEEP LEARNING

AI là một kỹ thuật do con người lập trình tạo ra với mục tiêu giúp cho máy móc có trí thông minh và khả năng nhận biết, tương tác với hành động cụ thể như con người. ML là một nhánh nghiên cứu về kỹ năng học máy của AI, bao gồm các kỹ thuật và mô hình tiên tiến hơn, cho phép máy tính có khả năng tự học dựa trên dữ liệu đưa vào mà không cần phải lập trình cụ thể [9].

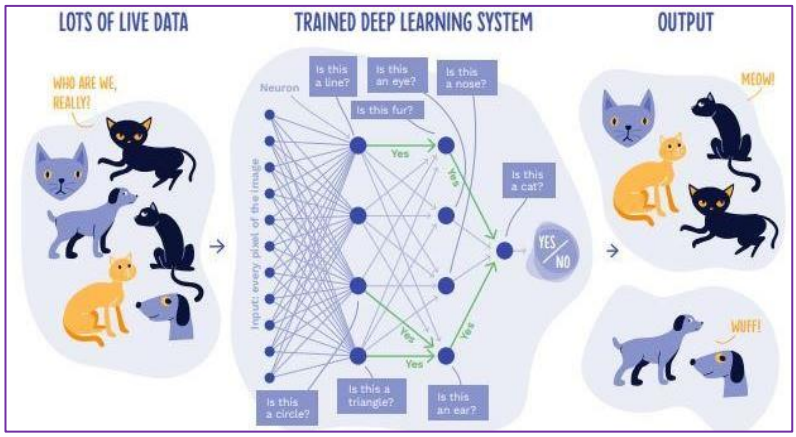


Hình 2.2: Mối liên hệ giữa AI, ML và DL [10]

Trong hình 2.2, cho thấy mối liên hệ giữa AI, ML và DL. Từ năm 1950, AI được mô phỏng và chỉ được hiện thực hóa khi ML (tập hợp con của AI) được phát triển. Khi khả năng tính toán của máy tính ngày càng nhanh và cần xử lý các nguồn thông tin lớn hơn, ML đã được cải tiến và cho ra đời DL.

DL là một phương pháp của ML, cho phép chúng ta huấn luyện một AI có thể dự đoán được các đầu ra dựa vào một tập các dữ liệu đầu vào. Khác với ML - chương trình chạy trên một mạng thần kinh nhân tạo, có khả năng tự học hỏi từ nguồn dữ liệu lớn được đưa vào mà không cần phải lập trình cụ thể, thì DL sử dụng nhiều lớp thần kinh nhân tạo để phân tích dữ liệu về nhiều chi tiết khác nhau (mỗi lớp xác định đặc điểm cụ thể của đối tượng phân tích).

DL sử dụng mạng Nơ-ron để bắt chước trí thông minh của con người. Khi ta xây dựng một mạng lưới Nơ-ron lớn và nạp vào càng nhiều dữ liệu thì hệ thống sẽ vận hành càng nhanh, cho ra kết quả chính xác hơn, đòi hỏi cần phải có nhiều tài nguyên để tính toán trong quá trình huấn luyện [9]. DL liên quan đến các thuật toán lấy cảm hứng từ cấu trúc và hoạt động của bộ não động vật gọi là mạng thần kinh nhân tạo (Artificial Neural Networks: ANN). ANN là một hệ thống các chương trình và cấu trúc dữ liệu mô phỏng cách vận hành của não người [9].



Hình 2.3: Deep Learning một mạng lưới Nơ - ron phát hiện mèo trong ảnh [11]

Trong hình 2.3, là một mạng lưới Nơ - ron phát hiện mèo trong ảnh. Để dạy máy tính nhận diện được hình ảnh con mèo thì nguồn dữ liệu đầu vào sẽ bao gồm hàng ngàn bức ảnh mèo và không phải mèo. Các đặc điểm của một bức ảnh mèo bao gồm: móng vuốt, râu, đuôi, chân, tai, được đưa vào mỗi lớp trong mạng (mỗi lớp sẽ có khả năng xác định một đặc điểm cụ thể của con mèo). Theo thời gian ANN đọc hết các hình ảnh, các lớp node sẽ dần nhận ra đặc điểm như râu, móng vuốt, đuôi, chân,... Xác định lớp nào quan trọng và không quan trọng. Nó sẽ ghi nhớ những dữ liệu này và sắp xếp theo mức độ quan trọng. Nó cũng sẽ nhận ra rằng móng vuốt không chỉ mèo mới có, nhưng nếu móng đi chung với bàn chân và ria mép thì đây chính là con mèo. Quy trình này diễn ra trong thời gian dài và lặp đi lặp lại nhiều lần, lần sau tốt hơn lần trước [12].

ANN bao gồm 3 phần chính là lớp ngõ vào (input layer), lớp trung gian (hidden layer) và lớp ngõ ra (output layer). Các nơ-ron thần kinh là các node (node

là đơn vị thần kinh trong mạng thần kinh nhân tạo - một chiếc máy tính trong mạng thần kinh có thể xem như là 1 node), chúng được kết nối với nhau trong một mạng lưới Nơ-ron lớn. Những tầng Nơ-ron rời rạc kết nối với những Nơ-ron khác. Mỗi tầng lại trích chọn ra những tính năng cụ thể để học, ví dụ như là đường cong hoặc cạnh trong nhận dạng hình ảnh. Chiều sâu của ANN được tạo ra bằng cách sử dụng nhiều lớp kết nối với nhau hay gọi tắt là Neural Network (NN) [11].

NN là phương pháp được sử dụng trong Deep Learning, dùng các phương pháp cơ bản để học, nhưng tùy theo mục đích sử dụng mà sẽ có nhiều loại mạng Nơ - ron khác nhau. Trong xử lý ảnh thường sử dụng mạng Nơ - ron tích chập (Convolutional Neural Networks: CNN) hay trong lĩnh vực máy dịch thường sử dụng mạng Nơ - ron hồi quy (Recurrent Neural Network : RNN).

Mạng RNN được ứng dụng nhiều trong các bài toán về lĩnh vực NLP (xử lý ngôn ngữ tự nhiên) sử dụng một bộ nhớ để lưu lại thông tin từ những bước tính toán xử lý trước đó, nhưng RNN không thể nhớ được những bước ở xa do bị mất mát đạo hàm, RNN chủ yếu biểu diễn dữ liệu thời gian. Về mặt lý thuyết thì RNN có thể xử lý và lưu trữ thông tin của một chuỗi dữ liệu với độ dài bất kỳ. Tuy nhiên, trong thực tế thì RNN chỉ tỏ ra hiệu quả với chuỗi dữ liệu có độ dài không quá lớn [13].

Mạng CNN được sử dụng nhiều trong các bài toán nhận dạng các đối tượng trong ảnh, các mạng CNN được thiết kế theo nguyên tắc chung là sử dụng nhiều lớp tích chập (convolutional layer) chồng lên nhau, giảm dần kích thước ngõ ra ở mỗi tầng và tăng số lượng ánh xạ đặc trưng (feature map). Thông qua việc nhân chập, CNN làm cho dữ liệu lớn nhỏ dần trong khi lưu giữ thông tin.

2.4 CONVOLUTIONAL NEURAL NETWORK (CNN)

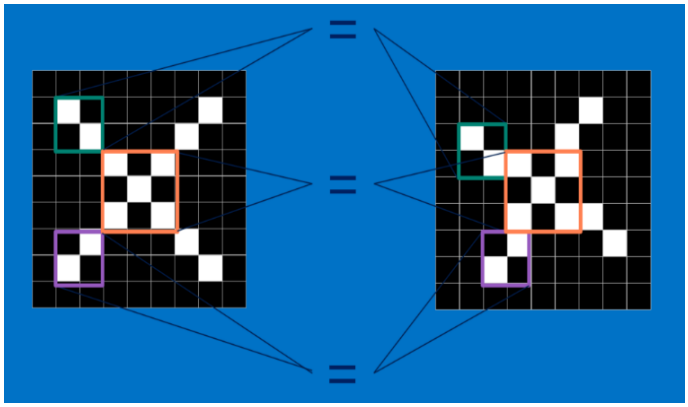
Mạng Nơ-ron tích chập (Convolutional Neural Network: CNN) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao. CNN trong nhận dạng cảm xúc khuôn

mặt cũng là một trong những giải pháp trong thị giác máy tính, giao tiếp giữa người và máy trong xu thế hiện nay.

CNN là một kiểu mạng ANN truyền thẳng, trong đó kiến trúc chính gồm nhiều thành phần được ghép nối với nhau theo cấu trúc nhiều tầng đó là: Convolution, Pooling, ReLU và Fully connected [14].

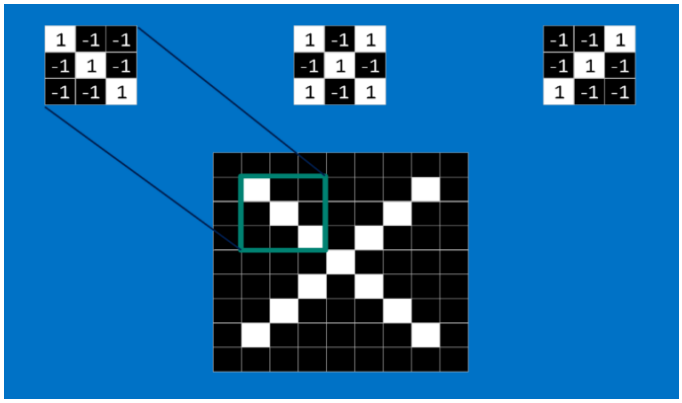
2.4.1 Đặc trưng (Feature)

CNN so sánh hình ảnh theo từng mảng. Các mảng mà nó tìm được gọi là tính năng (feature). Bằng cách tìm các mức thô của các feature khớp nhau ở cùng một vị trí trong hai hình ảnh, CNN tìm ra sự tương đồng tốt hơn nhiều so với việc khớp toàn bộ bức ảnh [15].



Hình 2.4: Feature [15]

Mỗi feature giống như một hình ảnh mini, chứa một mảng giá trị hai chiều. Các feature khớp với các khía cạnh chung của hình ảnh.



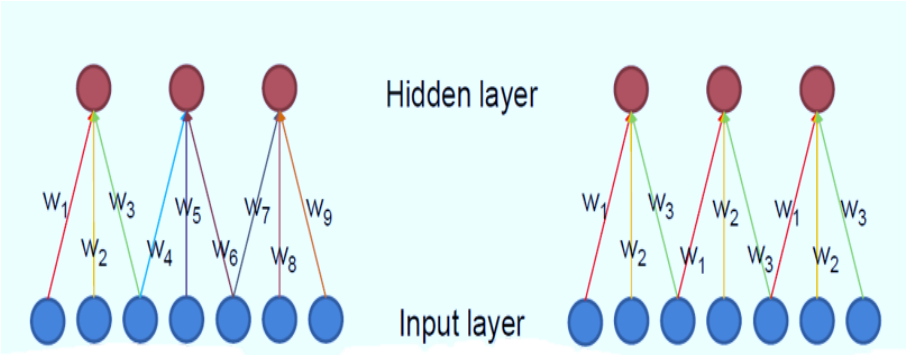
Hình 2.5: Feature của ảnh X [15]

Trong hình 2.5, feature của ảnh chứa tất cả những đặc điểm quan trọng của hầu hết các hình ảnh X, bao gồm các đường chéo và hình chữ thập. Những feature này có thể khớp với phần cạnh (đường chéo ở các góc) hoặc phần trung tâm của một hình ảnh X bất kì.

2.4.2 Chia sẻ trọng số (Shared weights and bias)

Tất cả các Nơ - ron trong lớp trung gian (hidden layer) chia sẻ cùng một trọng số hóa: vectơ trọng lượng và độ lệch (shared weight and bias) tạo thành một "Feature map".

Chia sẻ trọng số là một trong những nguyên tắt quan trọng, làm giảm tối đa số lượng trọng số giúp việc huấn luyện mô hình trở nên hiệu quả hơn trong mạng CNN [16].



i) Không chia sẻ trọng số ii) Chia sẻ trọng số

Hình 2.6: Kỹ thuật chia sẻ trọng số trong CNN [17]

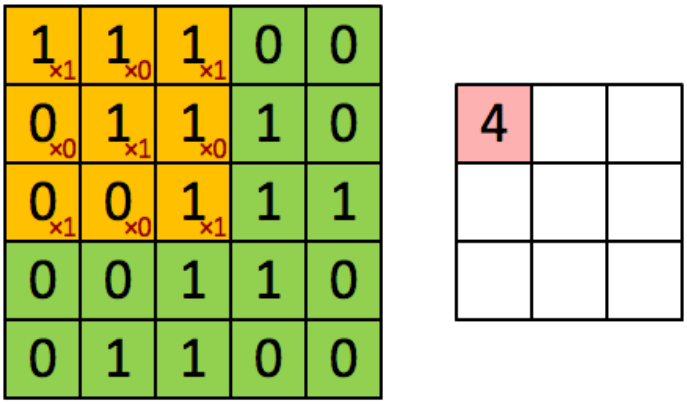
Trong hình 2.6, là kỹ thuật chia sẻ trọng số trong CNN. Các vùng bộ lọc đi qua trên ma trận đầu vào không tách biệt hoàn toàn, mà chia sẻ ít nhiều một phần diện tích (phụ thuộc vào cửa sổ trượt Stride). Các vùng khác nhau trên ảnh dùng chung một trọng số, từ đây số lượng trọng số sẽ được giảm xuống.

Ví dụ như bộ phát hiện cạnh hoạt động tốt trên một vùng của ảnh đầu vào thì cũng có thể hoạt động tốt trên các vùng còn lại. Nếu một tập trọng số có thể phát hiện cạnh ở góc trên bên trái của ảnh đầu vào thì chúng cũng có thể được dùng để phát hiện cạnh ở góc phải bên dưới của ảnh đó. Vì thế không cần thiết phải sử dụng hai bộ phát hiện cạnh khác nhau cho hai vùng khác nhau của bức ảnh [18].

2.4.3 Các thành phần cơ bản của mạng CNN

2.4.3.1 Convolution

Lớp Convolution (Conv) là lớp quan trọng nhất trong cấu trúc của CNN. Conv dựa trên lý thuyết xử lý tín hiệu số, việc lấy tích chập sẽ trích xuất được những thông tin quan trọng từ dữ liệu.



Hình 2.7: Phép toán tích chập [19]

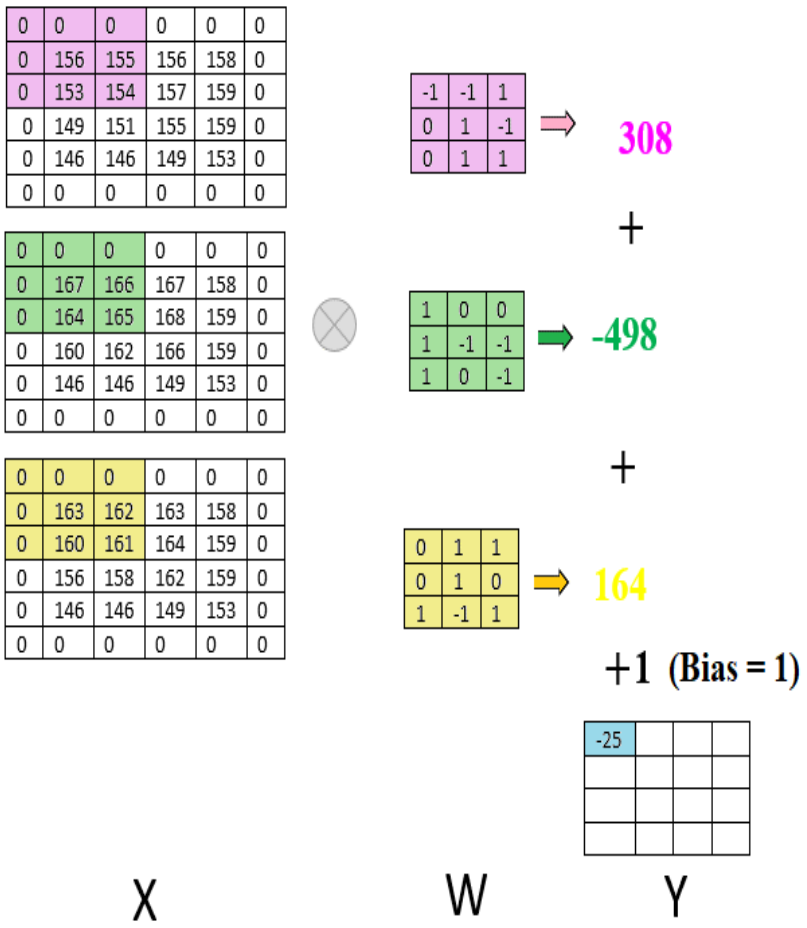
Hình 2.7 mô tả lý thuyết và cách thức lớp Conv hoạt động trên một dữ liệu đầu vào. Phép tính chập được thực hiện bằng cách dịch chuyển một cửa sổ mà ta gọi là kernel trên ma trận đầu vào, trong đó kết quả mỗi lần dịch chuyển được tính bằng tổng tích chập (tích của các giá trị trong 2 ma trận tại vị trí tương ứng) [19].

Ảnh màu có 3 kênh R, G, B, khi biểu diễn ảnh dưới dạng tensor 3 chiều thì đối với ma trận kernel cũng được biểu diễn là 1 tensor 3 chiều có kích thước k*k*3 (k: số lẻ).

Trong hình 2.8, ma trận kernel có cùng độ sâu (depth) với ảnh, thực hiện dịch chuyển kernel trên ảnh màu cũng tương tự như trên ảnh xám.

Biểu diễn ma trận có 2 chỉ số hàng và cột: i và j, khi biểu diễn ở dạng tensor 3 chiều ma trận có thêm chỉ số độ sâu k. Nên chỉ số mỗi phần tử trong tensor là x_{ijk}, ngõ ra là y_{ij}.

Ví dụ: Tính giá trị y_{11} trong hình 2.8



Hình 2.8: Thực hiện tích chập kernel trên ảnh đối với ảnh màu [20]

Từ hình 2.8, ta có:

y_{11}= [(x_{111}* w_{111} + x_{121}* w_{121} + x_{131}* w_{131} + x_{211}* w_{211} +x_{221}* w_{221} + x_{231}* w_{231} + x_{311}* w_{311} + x_{321}* w_{321} + x_{331}* w_{331}) + (x_{112}* w_{112} + x_{122}* w_{122} + x_{132}* w_{132} + x_{212}* w_{212} +x_{222}* w_{222} + x_{232}* w_{232} + x_{312}* w_{312} + x_{322}* w_{322} + x_{332}* w_{332}) + (x_{113}* w_{113} + x_{123}* w_{123} + x_{133}* w_{133} + x_{213}* w_{213} +x_{223}* w_{223} + x_{233}* w_{233} + x_{313}* w_{313} + x_{323}* w_{323} + x_{333}* w_{333})] + Bias = 308 + (-498) + 164 +1 = -25

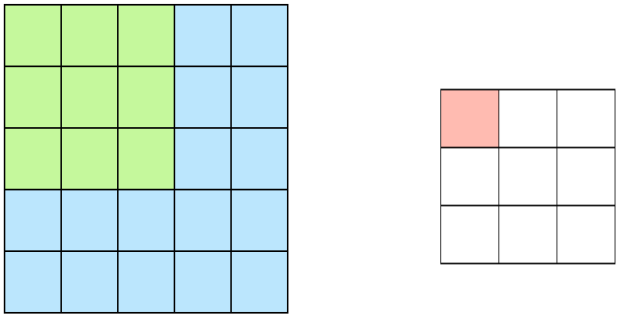
Trong đó, hệ số Bias được cộng vào sau bước tính tổng các phần tử của phép tính element-wise (áp dụng cho từng thành phần của ma trận đầu vào, sau đó các thành phần này được sắp xếp lại đúng theo thứ tự để được một ma trận có kích thước bằng với ma trận ngõ vào) [20].

Mỗi kernel dịch chuyển trên ảnh màu có kích thước k*k*D (D =3) và có 1 hệ số Bias, nên tổng tham số (parameter) của 1 kernel là k*k*D + 1. Mà lớp tích chập (Conv) áp dụng n kernel nên tổng số parameter trong lớp này là:

$$\text{Total Params (Conv)} = n * (k*k*D + 1) \quad (2.8)$$

Ngõ ra của lớp Conv đầu tiên sẽ thành ngõ vào của lớp Conv tiếp theo. Càng qua nhiều lớp Conv thì kích thước độ rộng (W) và độ cao (H) của ảnh càng giảm nhưng độ sâu (D) càng tăng [20].

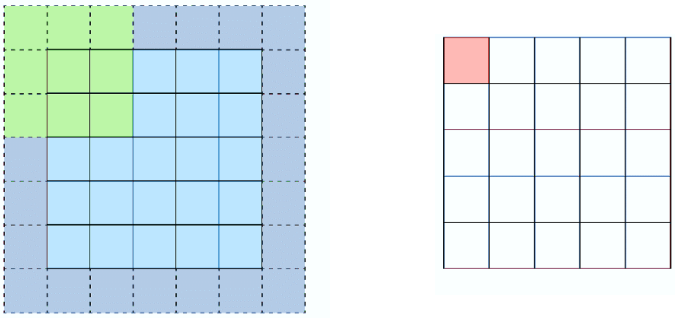
2.4.3.2 Cửa sổ trượt (Stride) và Padding



i) Stride 1 ii) Feature map

Hình 2.9: Stride bằng 1 [19]

Stride là khoảng cách giữa 2 kernel khi quét. Với stride = 1, kernel sẽ quét 2 ô cạnh nhau. Với stride =2, kernel sẽ quét ô số 1 và ô số 3, bỏ qua ô số 2. Điều này sẽ tránh được việc lặp lại giá trị ở các ô bị quét.



i) Stride 1 với padding ii) Feature map

Hình 2.10: Stride với padding [19]

Khi giá trị stride và kích thước của kernel càng lớn thì kích thước của feature map càng nhỏ. Sử dụng padding để giữ nguyên kích cỡ của feature map so với ban đầu. Khi padding = 1, ma trận ảnh sẽ được thêm 1 ô bọc xung quanh các cạnh của ngõ vào, phần bọc này càng dày thì giá trị của padding sẽ càng tăng [19].

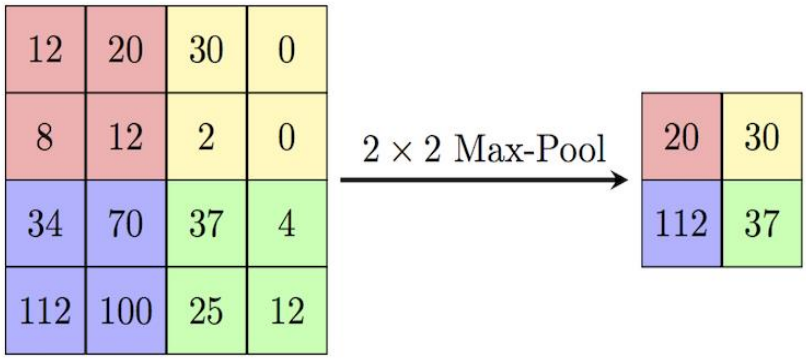
2.4.3.3 Pooling

Lớp Pooling (hay còn gọi Subsampling hoặc Downsample) thường được dùng giữa các lớp Conv, dùng để giảm kích thước ma trận nhưng vẫn làm nổi bật lên được đặc trưng có trong ma trận đầu vào. Trong CNN, toán tử Pooling được thực hiện độc lập trên mỗi kênh màu của ma trận ảnh đầu vào.

Trong một số mô hình người ta dùng lớp Conv với Stride > 1 để giảm kích thước dữ liệu thay cho lớp Pooling [20].

Có 2 loại Pooling phổ biến là: Max Pooling và Average Pooling.

a) Max pooling

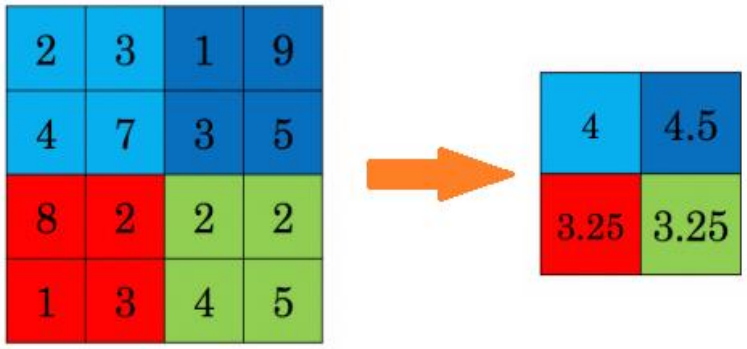


Hình 2.11: Max Pooling [19]

Sau khi qua lớp Pooling, ảnh sẽ có khoảng một phần tư số điểm ảnh so với lúc bắt đầu. Tuy nhiên, Max Pooling được sử dụng nhiều là do nó giữ lại chi tiết quan trọng. Đó là giá trị lớn nhất từ mỗi cửa sổ và bảo toàn tính khớp của mỗi feature bên trong cửa sổ.

b) Average Pooling

Pooling với bộ lọc 2x2 và stride 2. Trong hình 2.12, Average Pooling lấy trung bình cộng 4 số và trả về kết quả.



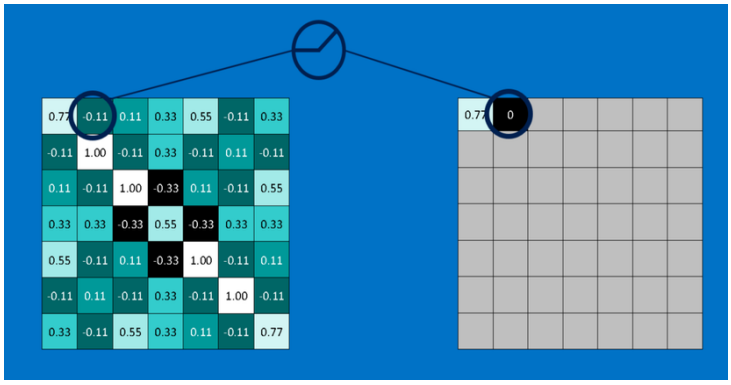
Hình 2.12: Average Pooling [20]

Tất cả giá trị ở ngõ ra đều được tính trung bình và lấy xấp xỉ nên Average Pooling không thể trích xuất các tính năng tốt, nó sẽ không chính xác đối với việc phát hiện đối tượng.

2.4.3.4 ReLU

Hàm ReLu (Rectified Linear Unit) được sử dụng để biến đổi mạng Nơ-ron tổng hợp từ phép biến đổi tuyến tính đưa về dưới dạng một hàm tuyến tính cho mỗi mạng Nơ-ron (do khi qua lớp Conv, quá trình nhận chập sẽ là một phép biến đổi tuyến tính), hàm ReLu cho kết quả tốt ở nhiều khía cạnh.

Thuật toán của ReLU: trong những trường hợp có số âm thì hoán đổi số âm với 0.

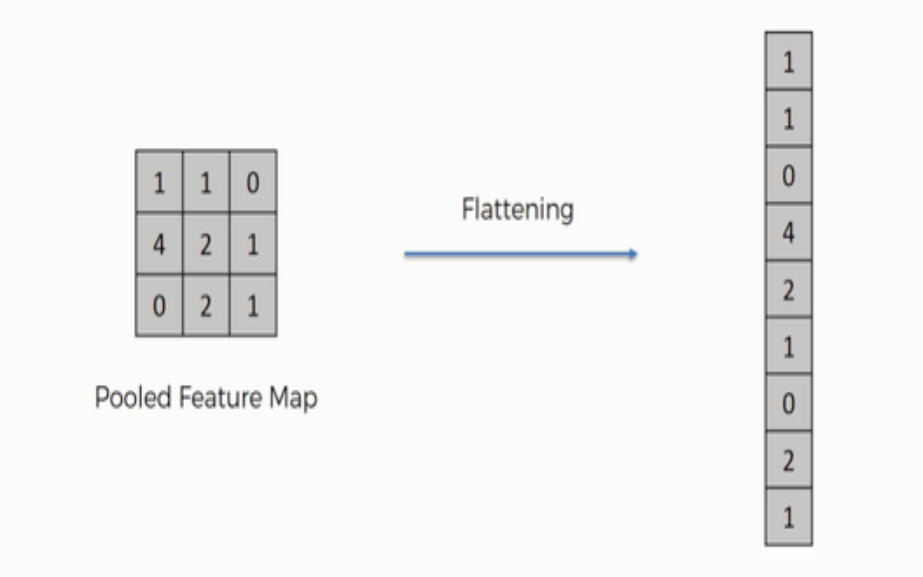


Hình 2.13: ReLU [17]

Trong hình 2.13, đầu ra của một lớp ReLu có kích thước giống với đầu vào, ngoại trừ tất cả các giá trị âm được loại bỏ. Mục đích của lớp ReLu là đưa ra một mức ngưỡng, ở đây là 0. Để loại bỏ các giá trị âm không cần thiết, có thể sẽ ảnh hưởng cho việc tính toán ở các lớp (layer) sau đó [17].

2.4.3.5 Fully Connected (FC)

Các lớp được kết nối đầy đủ (Fully Connected layers) là cách kết nối các Nơ - ron ở hai tầng với nhau trong đó tầng sau kết nối đầy đủ với các Nơ - ron ở tầng trước nó. Đây cũng là dạng kết nối thường thấy ở ANN, trong CNN lớp này thường được sử dụng ở các tầng phía cuối của kiến trúc mạng.



Hình 2.14: Fully Connected Layer [20]

Trong hình 2.14, sau khi ảnh được truyền qua nhiều lớp Conv và lớp Pooling thì mô hình đã học được tương đối các đặc điểm của ảnh (ví dụ: mắt, mũi, khung mặt,…). Tensor ngõ ra của lớp cuối cùng có kích thước H*W*D, sẽ được chuyển về 1 vector kích thước (H*W*D). Các lớp Fully Connected sẽ kết hợp đặc điểm ảnh để đưa tới ngõ ra của mô hình huấn luyện [18].

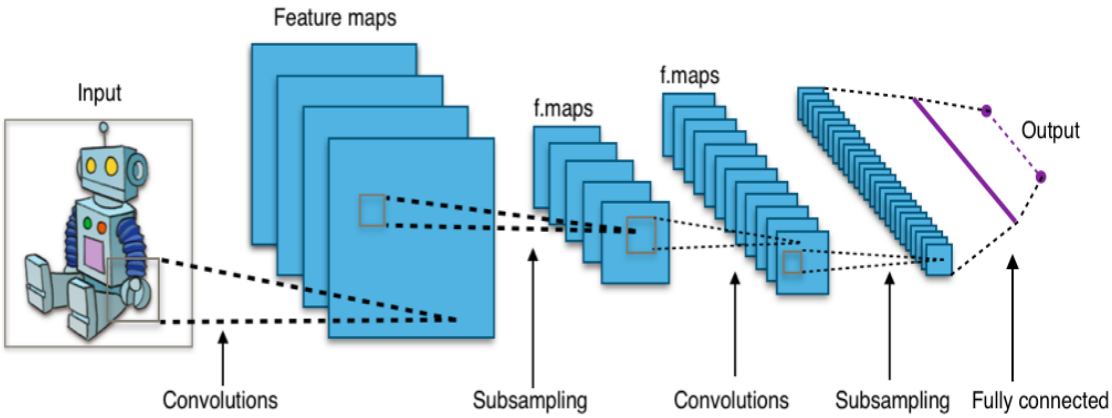
Có 2 lớp Fully Connected, một lớp để tập hợp các feature mà ta đã tìm ra, chuyển đổi dữ liệu từ 3D, hoặc 2D thành 1D, tức chỉ còn là 1 vector. Còn một lớp là ngõ ra (output), số Nơ - ron của lớp này phụ thuộc vào số lớp ở ngõ ra cần tìm.

2.4.4 Kiến trúc mạng CNN

2.4.4.1 CNN layers

Các mạng CNN đều được thiết kế theo nguyên tắc chung:

- Sử dụng nhiều lớp Conv chồng lên nhau
- Giảm dần kích thước ngõ ra mỗi tầng
- Tăng dần số lượng feature map [19].



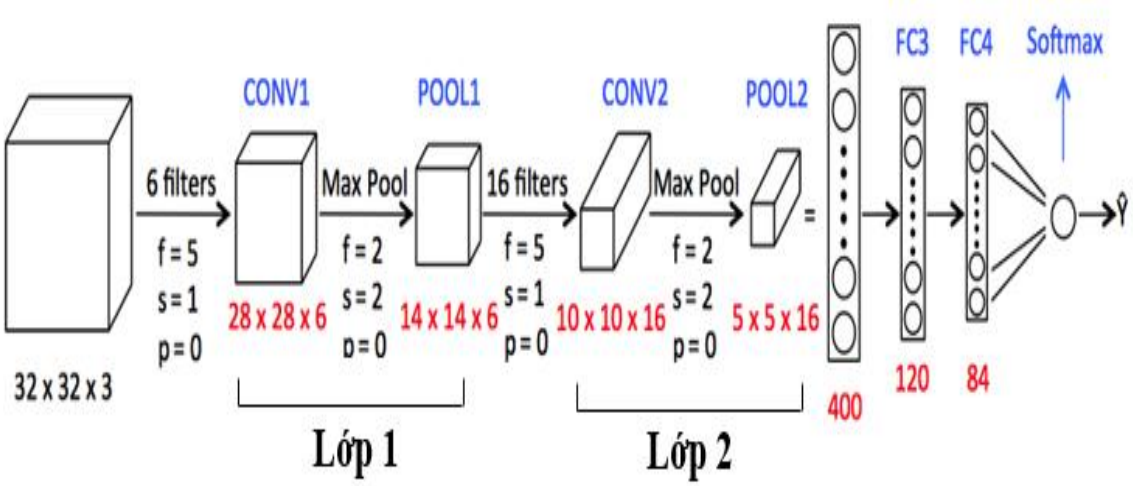
Hình 2.15: CNN layers [19]

Lớp Conv là một tập các feature map và mỗi feature map là một bản scan của ngõ vào ban đầu, nhưng được trích xuất ra các feature hoặc đặc tính cụ thể. Việc scan ngõ vào sẽ dựa vào bộ lọc chập (convolution filter) hay kernel.

CNN là thuật toán có kiến trúc bao gồm nhiều tầng có chức năng khác nhau, trong đó tầng chính hoạt động thông qua cơ chế Conv. Trong suốt quá trình huấn luyện, CNN sẽ tự động học được các thông số cho các bộ lọc (filter), tương ứng là các đặc trưng theo từng cấp độ khác nhau [19].

2.4.4.2 Cấu trúc CNN

Ảnh đầu vào qua hàng loạt các lớp Conv và lớp Max Pooling (thường Pooling sẽ theo sau một lớp Conv), cuối cùng là 2 lớp Fully Connected. Càng về cuối của CNN, kích thước ngõ ra của ảnh càng giảm xuống trong khi số lượng feature map càng tăng lên.



Hình 2.16: Một ví dụ điển hình về ConvNet [21]

Trong hình 2.16, CNN có hai cụm (lớp 1 và lớp 2), mỗi cụm chứa một lớp Conv và một lớp Max Pooling. Hai lớp liên kết đầy đủ (FC layers) ở cuối cùng, theo sau bởi một lớp Softmax. Trong trường hợp này, ảnh có kích thước ngõ vào là 32x32 pixel với ba kênh màu R, G, B, sau khi qua bộ lọc, ngõ vào được giữ lại các giá trị pixel thô.

Kích thước của lớp tiếp theo sẽ được xác định bằng công thức:

$$\frac{n + 2p - f}{s} + 1 \qquad 2.16$$

Với: f: filter size, s: stride, p: padding

Lớp Conv sẽ tính toán đầu ra của các Nơ - ron được kết nối với các vùng cục bộ ở ngõ vào. Số lượng tham số trong các lớp tích chập (Conv) không cao. Lớp Pooling sẽ thực hiện thao tác lấy mẫu dọc theo kích thước không gian (chiều rộng, chiều cao).

Lớp Pooling không chứa trọng số hay tham số nhưng lại giúp giảm kích thước của ảnh. Đa phần các tham số tập trung ở lớp FC. Từ trái sang phải, kích thước các hàm kích hoạt ReLu có xu hướng giảm, kích thước các hàm ReLu giảm quá nhanh sẽ ảnh hưởng tới hiệu năng của cả CNN nên cần phải thiết lập giá trị các tham số (p, s, f) [20].

2.5 THƯ VIỆN CHÍNH SỬ DỤNG TRONG HỆ THỐNG

2.5.1 OpenCV

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và ML, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

Thư viện OpenCV bao gồm một số tính năng nổi bật như: bộ công cụ hỗ trợ 2D và 3D, nhận diện khuôn mặt, nhận diện cử chỉ, nhận dạng chuyển động, đối tượng, hành vi, tương tác giữa con người và máy tính, điều khiển Robot và hỗ trợ thực tế tăng cường [22].

Trong hệ thống, OpenCV được dùng để xử lý ảnh và nhận diện khuôn mặt qua trích xuất ROI - khuôn mặt trong ảnh hoặc video từ webcam.

2.5.2 Tensorflow

Thư viện Tensorflow là thư viện mã nguồn mở (open source) do Google phát triển, hỗ trợ mạnh mẽ các phép toán học để tính toán trong ML và DL. Tất cả các kiểu dữ liệu khi đưa vào trong Tensorflow thì đều gọi là tensor. Vậy nên, Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các tensor.

Chương trình Tensorflow có 2 phần chính: phần thứ nhất là xây dựng mô hình tính toán và phần thứ hai là chạy mô hình vừa mới xây dựng.

Tính năng quan trọng của TensorFlow là hệ thống nút nhiều lớp, cho phép huấn luyện các neural networks trên bộ dữ liệu lớn một cách nhanh chóng, hỗ trợ khả năng nhận diện giọng nói và định vị vật thể trong ảnh của Google.

Tuy nhiên, Tensorflow là một framework (tính toán dựa trên các tensor) tương đối khó sử dụng. Vì vậy, sự ra đời của Keras (một framework DL) thật sự hữu ích. Keras là một thư viện tương đối dễ sử dụng. Keras cung cấp các hàm số cần thiết với cú pháp đơn giản. Có bốn modules chính trong Keras là Keras models, Keras layers, Keras losses và Keras optimizers [23].

2.5.3 Keras

Keras là một thư viện được phát triển vào năm 2015 bởi một kỹ sư nghiên cứu DL tại Google. Keras là một API mạng thần kinh cấp cao, được viết bằng Python và có khả năng chạy trên TensorFlow, CNTK hoặc Theano. Nó được phát triển với trọng tâm là cho phép thử nghiệm nhanh, có thể đi từ ý tưởng đến kết quả với độ trễ ít nhất.

Khi huấn luyện mô hình, Keras có thể kết hợp các mô-đun độc lập để tạo ra mô hình mới. Các mô-đun độc lập bao gồm: các lớp mạng, hàm loss, optimizers, khởi tạo initialization, hàm activation và regularization. Ngoài ra, Keras không có tệp cấu hình mô hình riêng trong một định dạng khai báo. Các mô hình được mô tả bằng mã Python, nhỏ gọn, dễ gỡ lỗi hơn và cho phép dễ dàng mở rộng [24].

Keras có một số ưu điểm như: dễ sử dụng, xây dựng mô hình nhanh, hỗ trợ cho CNN, RNN hoặc cả hai, và có thể chạy trên cả CPU và GPU.

Chương III. THIẾT KẾ HỆ THỐNG

3.1 YÊU CẦU BÀI TOÁN

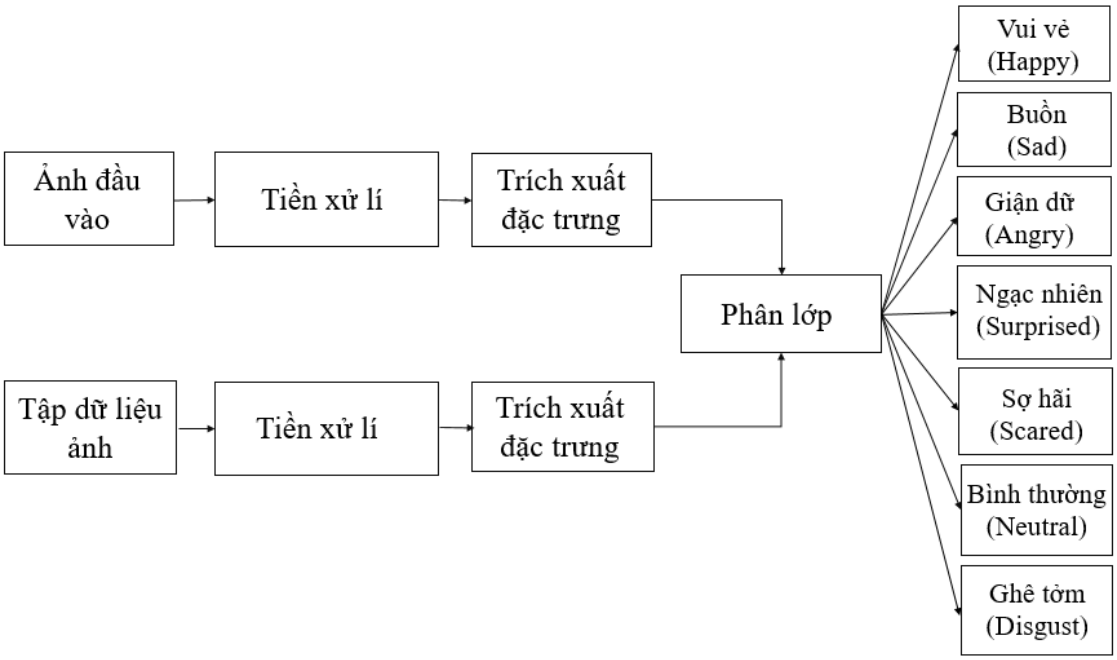
Con người rất giỏi trong việc nhận diện khuôn mặt và các hình mẫu phức tạp. Ngay cả khi thời gian trôi qua vẫn không ảnh hưởng đến khả năng này, khi công nghệ nhận dạng đối tượng phát triển, con người đã biến máy tính trở thành công cụ mạnh mẽ và thông minh trong việc nhận diện khuôn mặt, đưa ra những thông tin cần thiết. Mục tiêu là nghiên cứu xây dựng các ứng dụng phân tích cảm xúc khuôn mặt nhằm đánh giá về sản phẩm công nghệ của người dùng.

Bài toán nhận diện cảm xúc trên khuôn mặt là một kỹ thuật của máy tính tự động nhận dạng và xác định biểu hiện nét mặt của một người, từ một ảnh bất kỳ (ảnh kỹ thuật số) hoặc một khung hình video (webcam theo thời gian thực). Kỹ thuật này nhận biết các đặc trưng khuôn mặt, bỏ qua môi trường xung quanh như cây cối, tòa nhà, cơ thể,..., và sau đó là phân tích cảm xúc đối tượng. Một trong những cách để thực hiện điều này là so sánh các đặc tính của khuôn mặt với những hình ảnh trong cơ sở dữ liệu đã tạo hoặc có sẵn (trong cơ sở dữ liệu đó có chứa các hình ảnh phân loại theo từng nhãn cảm xúc riêng biệt) và trả về kết quả.

Để thực hiện hệ thống, nhóm tác giả đã sử dụng nhận dạng khuôn mặt người bằng trích xuất đặc trưng Haar Cascade, sau đó là phân tích cảm xúc đối tượng dựa trên kiến trúc mô hình CNN. Mô hình sau khi huấn luyện từ bộ dữ liệu FER2013, tiến hành lựa chọn mô hình tối ưu nhất cho hệ thống, mô hình này phải đạt độ chính xác hơn 70%. Cuối cùng là nhận dạng cảm xúc đối tượng người dùng thông qua hình ảnh và video trực tiếp từ webcam. Hiển thị kết quả có được lên giao diện thiết kế.

3.2 TỔNG QUAN HỆ THỐNG NHẬN DIỆN CẢM XÚC

Hệ thống gồm 4 phần: thu thập dữ liệu hình ảnh, phần tiền xử lý, phần trích xuất đặc trưng và phần phân tích cảm xúc.



Hình 3.1: Tổng quan hệ thống

Phần thu thập dữ liệu hình ảnh và phần tiền xử lý: tập dữ liệu thu thập được là các hình ảnh chứa khuôn mặt có các biểu hiện khác nhau, sau đó tiền xử lý các hình ảnh. Hệ thống sử dụng cơ sở dữ liệu FER2013 gồm 35887 hình ảnh thang độ xám 48 x 48 pixel được cắt xén trước vùng chứa khuôn mặt và phân loại thành 7 lớp cảm xúc: giận dữ (angry), ghê tởm (disgust), sợ hãi (scared), vui vẻ (happy), buồn (sad), ngạc nhiên (surprised) và bình thường (neutral).

Phần trích xuất đặc trưng là dựa vào tập dữ liệu, cho hình ảnh qua các lớp của mô hình CNN Mini_Xception.

Phần phân tích cảm xúc là nguồn ảnh vector ngõ ra qua lớp softmax chia thành 7 lớp cảm xúc.

Tiến hành đánh giá mô hình đã được lưu lại sau mỗi epoch, lựa chọn mô hình có độ chính xác cao nhất và tiến hành phân tích cảm xúc: kiểm tra kết quả trên hình ảnh hoặc video theo thời gian thực trên giao diện.

3.3 TẬP DỮ LIỆU FER2013

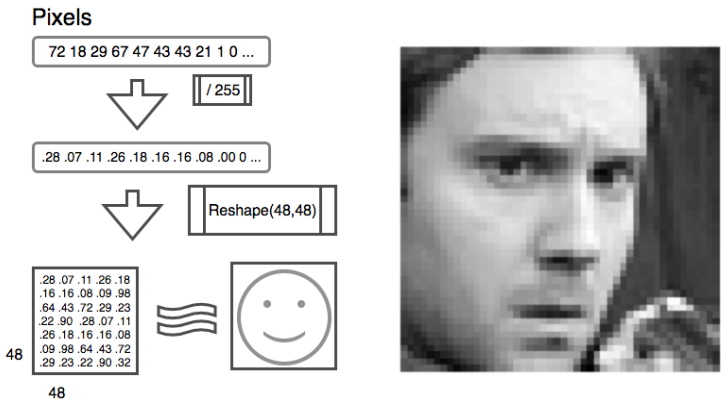
Tập dữ liệu nguồn mở **FER2013.csv**, được tạo ra cho một dự án bởi Pierre-Luc Carrier và Aaron Courville, được chia sẻ công khai trong cuộc thi Kaggle (2013) [25].

1	emotion	pixels	Usage
2	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84	Training
3	0	151 150 147 155 148 133 111 140 170 174 182 154 153 164 173 178 1	Training
4	2	231 212 156 164 174 138 161 173 182 200 106 38 39 74 138 161 164 1	Training
5	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 19 43 52 13 26 40 59 65	Training
6	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84 115 127 137 142 151 156	Training
7	2	55 55 55 55 55 54 60 68 54 85 151 163 170 179 181 185 188 188 191 1	Training
8	4	20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 108 118 130 139 134 13	Training
9	3	77 78 79 79 78 75 60 55 47 48 58 73 77 79 57 50 37 44 56 70 80 82 87	Training
10	3	85 84 90 121 101 102 133 153 153 169 177 189 195 199 205 207 209 2	Training

Hình 3.2: Dữ liệu trong tập FER2013.csv

Bộ dữ liệu này bao gồm 35.887 ảnh xám: hình ảnh khuôn mặt kích thước 48x48 pixel từ nhiều góc độ khác nhau. Hình ảnh được phân loại thành một trong bảy lớp thể hiện cảm xúc khuôn mặt khác nhau, tất cả được gán nhãn từ 0 – 7 (0 = Giận dữ, 1 = Ghê tởm, 2 = Sợ hãi, 3 = Vui vẻ, 4 = Buồn, 5 = Ngạc nhiên, 6 = Bình thường). Gồm 8.989 ảnh ‘Happy’, 6.077 ảnh ‘Sad’, 6.198 ảnh ‘Neutral’, 4002 ảnh ‘Suprised’, 5121 ảnh ‘Scared’, 547 ảnh ‘Disgust’ và 4593 ảnh ‘Angry’ [25].

Tập dữ liệu FER2013 định dạng tệp csv, sau khi qua chương trình đọc tệp csv, các chuỗi pixel của mỗi hàng được chuyển đổi thành hình ảnh có kích thước 48x48 pixel, giá trị trả về là khuôn mặt và nhãn cảm xúc.



Hình 3.3: Biến đổi chuỗi pixel sang ảnh 48x48 pixel trong FER2013.csv [25]

Hình 3.3, ảnh trong tệp FER2013.csv sau khi được chuyển từ chuỗi pixel sang ảnh kích thước 48 x48 pixel. Ngoài số lớp hình ảnh (từ 0 đến 7), các hình ảnh này được chia thành ba bộ dữ liệu khác nhau là trainning_set, validation_set và test_set. Có khoảng 29.000 hình ảnh cho training_set, 4.000 hình ảnh cho validation_set và 4.000 hình ảnh cho test_set.

Training_set: tập dữ liệu máy dùng để học và rút trích được những đặc điểm quan trọng để ghi nhớ lại, xây dựng mô hình bằng cách điều chỉnh trọng số (weights) trong mạng Nơ - ron.

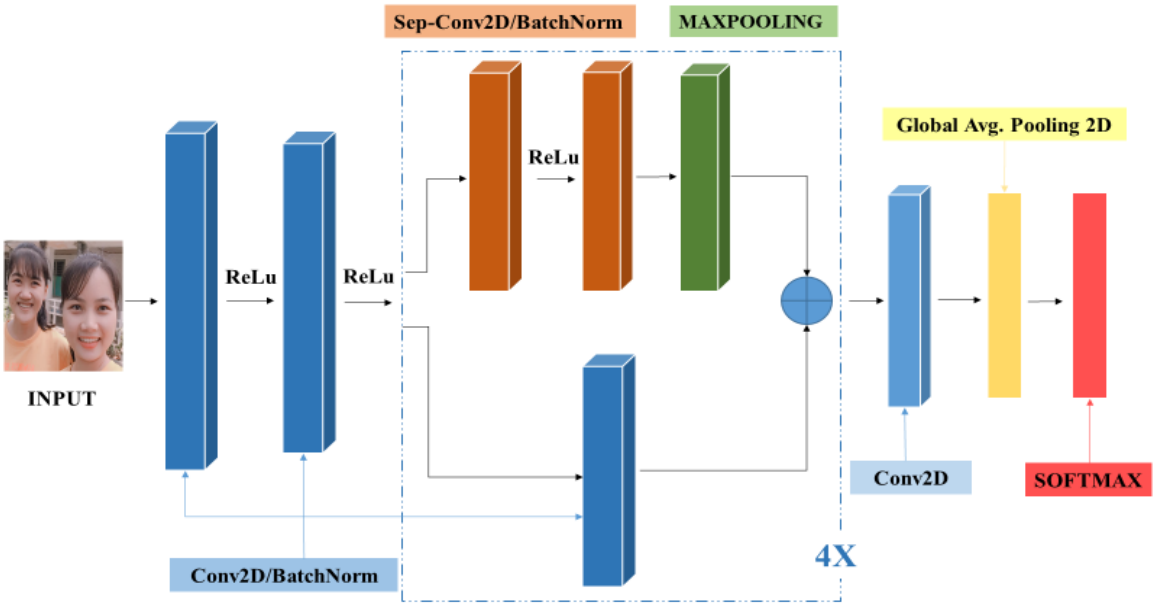
Validation_set: dùng để kiểm tra mạng Nơ - ron sau khi được huấn luyện, quyết định chọn một mô hình giữa các mô hình khác nhau. Bao gồm dữ liệu đầu ra, đầu vào (có cùng phạm vi với các mẫu của training_set) nhưng chúng không được sử dụng trong quá trình huấn luyện.

Test_set: dùng để đánh giá hiệu suất của mô hình và được thực hiện sau khi huấn luyện.

3.4 XÂY DỰNG MÔ HÌNH CNN SỬ DỤNG CNN VÀ KERAS

3.4.1 Xây dựng mạng lưới Mini_Xception

Trong đề tài này, hệ thống sử dụng mô hình CNN: Mini_Xception.



Hình 3.4: Mô hình huấn luyện CNN: kiến trúc Mini_Xception

Kiến trúc Mini_Xception là kiến trúc tương đối nhỏ, được đề xuất bởi Octavio Arriaga [26], trung tâm của khối kiến trúc này được lặp lại 4 lần trong thiết kế. Khác với kiến trúc CNN thông thường sử dụng các lớp Fully-Connected ở lớp cuối (nơi lưu hầu hết các tham số), kiến trúc Mini_Xception bỏ hoàn toàn lớp Fully-Connected ở lớp cuối, giảm lượng tham số sau mỗi lớp tích chập (Convolution), thay vào đó là sử dụng Global Average Pooling sau lớp tích chập cuối. Số lượng các feature map tương ứng với số lượng các lớp ngõ ra, các feature map này được giảm đi khi qua lớp đầu ra Softmax. Lớp Softmax sẽ dự đoán xác suất cho mỗi lớp cảm xúc.

3.4.2 Xây dựng chương trình mô hình

3.4.2.1 Các lớp trong mô hình

Mô hình Mini_Xception có kích thước ngõ vào giảm dần qua các lớp tích chập Convolution (Conv2D) và Separable Convolution (Sep-Conv2D), nhưng tăng số chiều sâu.

Bảng 3.1: Số lượng tham số của các lớp trong mạng

Layer	Kích thước ngõ ra	Parameters
Dữ liệu ngõ vào	48x48x1	0
Conv2D_1/BatchNorm_1	46x46x8	72/32
ReLu_1	46x46x8	0
Conv2D_2/BatchNorm_2	44x44x8	576/32
ReLu_2	44x44x8	0
Sep – Conv2D_1/BatchNorm_4	44x44x16	200/64
ReLu_3	44x44x16	0
Sep – Conv2D_2/BatchNorm_5	44x44x16	400/64
MaxPool 2D_1	22x22x16	0
Conv2D_3/BatchNorm_3	22x22x16	128/64
Add_1(BatchNorm3+MaxPooling1)	22x22x16	0
Sep – Conv2D_3/BatchNorm_7	22x22x32	656/128
ReLu_4	22x22x32	0
Sep – Conv2D_4/BatchNorm_8	22x22x32	1312/128
MaxPooling2D_2	11x11x32	0
Conv2D_4/BatchNorm_6	11x11x32	512/128

Add_2(BatchNorm6+MaxPooling2)	11x11x32	0
Sep-Conv2D_5/BatchNorm_10	11x11x64	2336/256
ReLu_5	11x11x64	0
Sep-Conv2D_6/BatchNorm_11	11x11x64	4672/256
MaxPooling2D_3	6x6x64	0
Conv2D_5/BatchNorm_9	6x6x64	2048/256
Add_3(BatchNorm9+MaxPooling3)	6x6x64	0
Sep - Con2D_7/BatchNorm_13	6x6x128	8768/512
ReLu_6	6x6x128	0
Sep - Conv2D_8/BatchNorm_14	6x6x128	17536/512
MaxPooling2D_4	3x3x128	0
Conv2D_6/BatchNorm_12	3x3x128	8192/512
Add_4(BatchNorm12+MaxPooling4)	3x3x128	0
Conv2D_7	3x3x7	8071
Global Avg Pooling2D	7	0
Softmax	7	0

Trong bảng 3.1, số lượng tham số tại từng lớp có sự khác nhau. Cụ thể là:

Lớp MaxPooling2D và lớp ReLu không có tham số.

Đa phần tham số tập trung ở lớp Sep – Conv2D và Conv2D.

Sử dụng BatchNormalization sau mỗi lớp Conv2D và Sep-Conv2D để cân bằng trọng số, chuẩn hóa đầu ra của mỗi lớp thành các lô nhỏ (mini-batchs), giúp tăng tốc độ huấn luyện mô hình và giảm đi số lượng chu kỳ huấn luyện (epoch) [27].

- Lớp tích chập Conv2D_1 và Conv2D_2:

Đầu vào là hình ảnh xám có kích thước 48x48x1 đi qua 2 lớp tích chập Conv2D_1 và Conv2D_2 với 8 feature maps, kích thước bộ lọc là 3x3 và Stride là 1. Kích thước ảnh thay đổi thành 46x46x8 qua lớp Conv2D_1 và 44x44x8 qua lớp Conv2D_2.

Qua mỗi lớp tích chập Conv2D_1 và Conv2D_2, mô hình áp dụng hàm kích hoạt ReLu chuyển đổi các giá trị hiện tại sang miền giá trị khác. Đầu ra của ReLU có kích thước giống với đầu vào, ngoại trừ tất cả các giá trị âm được loại bỏ.

- Lớp Sep-Conv2D_1 và Sep-Conv2D_2:

Ngõ ra của lớp Conv2D_2 sau khi qua hàm kích hoạt ReLu sẽ đưa tới ngõ vào của lớp Sep-Conv2D_1. Lớp này có kích thước bộ lọc là 3x3, feature maps là 16, ma trận ảnh khi qua lớp này được phân tích thành 2 vector, kích thước bộ lọc được chia thành 3x1 và 1x3. Với số nhân ít hơn, độ phức tạp tính toán giảm xuống và mạng có thể chạy nhanh hơn, lớp này đặc biệt có tác dụng trong phát hiện (detect) các cạnh về chiều dọc hoặc ngang [28]. Kích thước ngõ ra của ảnh là 44x44x16. Áp dụng hàm kích hoạt ReLu ở ngõ ra của lớp.

Tiếp theo là lớp Sep-Conv2D _2, tương tự như lớp Sep-Conv2D_1 ngõ ra của ảnh là 44x44x16. Sau đó, Mini_Xception áp dụng lớp Max Pooling_1 với kích thước bộ lọc 3×3 và Stride là 2. Kích thước hình ảnh thu được sẽ giảm xuống thành 22x22x16.

- Lớp Conv2D_3:

Ngõ ra của lớp Conv2D_2 sau khi áp dụng hàm kích hoạt ReLu được đưa tới ngõ vào của lớp này, với kích thước bộ lọc là 1x1, feature maps là 16 và Stride là 2. Kích thước ngõ ra là 22x22x16.

Sau đó, ngõ ra của lớp Max_Pooling_1 và ngõ ra của Conv2D_3 sẽ được kết hợp lại (lớp Add_1) và đưa tới ngõ vào của hai lớp Sep-Conv2D và Conv2D tiếp theo.

- Lớp Sep-Conv2D _3 và Sep-Conv2D_4:

Ngõ ra của lớp Add_1 đưa tới ngõ vào của lớp Sep-Conv2D_3. Lớp này có kích thước bộ lọc là 3x3, feature maps là 32. Kích thước ngõ ra của ảnh là 22x22x32. Áp dụng hàm kích hoạt ReLu ở ngõ ra của lớp.

Tiếp theo là lớp Sep-Conv2D_4, tương tự như lớp Sep-Conv2D_3 ngõ ra của ảnh là 22x22x32. Áp dụng lớp Max Pooling_2 với kích thước bộ lọc 3×3 và Stride là 2. Kích thước hình ảnh thu được sẽ giảm xuống thành 11x11x32.

- Lớp Conv2D_4:

Ngõ vào là lớp Add_1, lớp này có kích thước bộ lọc là 1x1, feature maps là 32 và Stride là 2. Kích thước ngõ ra là 11x11x32.

Sau đó, ngõ ra của lớp Max_Pooling_2 và ngõ ra của Conv2D_4 sẽ được kết hợp lại (lớp Add_2) và đưa tới ngõ vào của hai lớp Sep-Conv2D và Conv2D tiếp theo.

- Lớp Sep-Conv2D_5 và Sep-Conv2D_6:

Ngõ ra của lớp Add_2 đưa tới ngõ vào của lớp Sep-Conv2D_5. Lớp này có kích thước bộ lọc là 3x3, feature maps là 64. Kích thước ngõ ra của ảnh là 11x11x64. Áp dụng hàm kích hoạt ReLu ở ngõ ra của lớp.

Tiếp theo là lớp Sep-Conv2D_6, tương tự như lớp Sep-Conv2D_5 ngõ ra của ảnh là 11x11x64. Áp dụng lớp Max Pooling_3 với kích thước bộ lọc 3×3 và Stride là 2. Kích thước hình ảnh thu được sẽ giảm xuống thành 6x6x64.

- Lớp Conv2D _5:

Ngõ vào là lớp Add_2, lớp này có kích thước bộ lọc là 1x1, feature maps là 64 và Stride là 2. Kích thước ngõ ra là 6x6x64.

Sau đó, ngõ ra của lớp Max_Pooling_3 và ngõ ra của Conv2D_5 sẽ được kết hợp lại (lớp Add_3) và đưa tới ngõ vào của hai lớp Sep-Conv2D và Conv2D tiếp theo.

- Lớp Sep-Conv2D _7 và Sep-Conv2D_8:

Ngõ ra của lớp Add_3 đưa tới ngõ vào của lớp Sep-Conv2D_7. Lớp này có kích thước bộ lọc là 3x3, feature maps là 128. Kích thước ngõ ra của ảnh là 6x6x128. Áp dụng hàm kích hoạt ReLu ở ngõ ra của lớp.

Tiếp theo là lớp Sep-Conv2D_8, tương tự như lớp Sep-Conv2D_7 ngõ ra của ảnh là 6x6x128. Áp dụng lớp Max Pooling_4 với kích thước bộ lọc 3×3 và Stride là 2. Kích thước hình ảnh thu được sẽ giảm xuống thành 3x3x128.

- Lớp Conv2D_6:

Ngõ vào là lớp Add_3, lớp này có kích thước bộ lọc là 1x1, feature maps là 128 và Stride là 2. Kích thước ngõ ra là 3x3x128.

Sau đó, ngõ ra của lớp Max_Pooling_4 và ngõ ra của Conv2D_6 sẽ được kết hợp lại (lớp Add_4) và đưa tới ngõ vào lớp Conv2D cuối cùng.

- Lớp Conv2D cuối cùng:

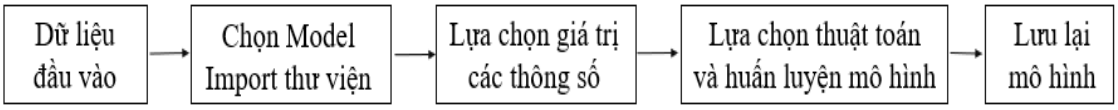
Ngõ vào là lớp Add_4, lớp này có kích thước bộ lọc là 3x3, feature maps bằng với số lớp ngõ ra (num_classes) là 7. Kích thước ngõ ra là 3x3x7 .

Tiếp theo là lớp Global Average Pooling, ngõ ra của lớp Conv2D cuối cùng được làm phẳng về 7 đơn vị.

- Lớp Softmax:

Cuối cùng, một lớp đầu ra softmax được hình thành với 7 đơn vị tương ứng với 7 nhãn cảm xúc ngõ ra.

3.4.2.2 Quá trình huấn luyện mô hình



Hình 3.5: Quá trình huấn luyện mô hình

Quá trình huấn luyện mô hình được chia thành 5 phần: phần dữ liệu đầu vào, phần chọn model và import thư viện, phần lựa chọn giá trị các thông số, phần lựa chọn thuật toán và huấn luyện mô hình, cuối cùng là lưu lại mô hình sau mỗi chu kỳ huấn luyện.

Dữ liệu đầu vào: trong dữ liệu huấn luyện FER2013 là các hình ảnh xám có kích thước 48x48 pixels, do đó chọn shape cho input layer là (48, 48, 1).

Chọn Model và Import thư viện:

Khởi tạo Model trong Keras sử dụng Sequential: `from keras.models import Sequential`. Sequential cho phép xây dựng một mô hình từng lớp, mỗi lớp có trọng số tương ứng với lớp theo sau nó.

Khai báo hàm Flatten(): "`from keras.layers import Flatten`", để biến mỗi điểm dữ liệu ở dạng mảng nhiều chiều thành mảng một chiều. Tiếp theo sử dụng hàm train_test_split() trong Keras để chia tách dữ liệu tham số test_size:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(faces,emotions,test_size=0.2,shuffle=True).
```

Tập dữ liệu được chia ra 20% là test_set và 80% còn lại là training_set.

Lựa chọn giá trị các thông số:

Epochs: số lần mạng Nơ - ron chạy qua toàn bộ tập dữ liệu. Số lượng epochs càng nhiều thì thuật toán huấn luyện càng lâu và dễ dẫn đến hiện tượng overfitting (mô hình hiện tại thực hiện tốt với dữ liệu trong training_set nhưng dự đoán không tốt với validation_set). Do vậy nên nó thường được kết hợp với kỹ thuật early stopping để thu được một mô hình tốt nhất trong quá trình huấn luyện, tránh hiện tượng overfitting [29].

Batch_size: số lượng dữ liệu được đưa vào trong một batch. Batch được dùng để cải thiện việc tính toán trong CNN bằng cách chia nhỏ các hình ảnh đầu vào thành nhóm nhỏ hơn. Số lượng batch_size được lựa chọn tuỳ thuộc vào dung lượng RAM hay GPU trên máy, batch_size càng lớn càng cần thêm bộ nhớ, thời gian huấn luyện nhanh hơn.

Num_classes: số lượng lớp ngõ ra, có 7 biểu hiện cảm xúc nên giá trị tham số num_classes = 7.

Validation_split: phần trăm dữ liệu chọn cho tập validation_set.

Patience: số lượng epochs không cải thiện sau khi giảm learning - rate.

Lựa chọn thuật toán và huấn luyện mô hình:

Regularization là một kỹ thuật giúp giảm đi số mô hình cần huấn luyện trong cross-validation, Regularization sẽ thay đổi mô hình tránh hiện tượng overfiting, trong khi vẫn giữ được tính tổng quát của nó (tính mô tả được nhiều dữ liệu trong tập training_set và test_set) [30]. Trong chương trình huấn luyện mô hình, hệ thống sử dụng 'regularization = l2(l2_regularization)' và 'EarlyStopping'.

Ngõ ra của lớp Global Average Pooling được đưa tới hàm Compile(). Compile sẽ biên tập lại toàn bộ model đã xây dựng trước đó. Tiếp theo lựa chọn thuật toán huấn luyện thông qua tham số 'optimier', function loss của model sử dụng mặc định hoặc tự xây dựng thông qua tham số 'loss', chọn metrics hiển thị khi model được training:

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

- Thuật toán lựa chọn huấn luyện trong mô hình là Adam Optimier, vì nó yêu cầu ít bộ nhớ, tính toán hiệu quả và cho kết quả nhanh.
- Cross-entropy sử dụng để so sánh khoảng cách giữa các giá trị đầu ra của softmax và biểu diễn các lớp đầu ra. Cross-entropy là một hàm loss và giá trị của nó có thể được cực tiểu hoá (minimized), giúp cho một neural networks đánh giá được xác suất của phép dự đoán một mẫu dữ liệu tương ứng với một lớp ngõ ra.
- Metrics là thước đo để đánh giá accuracy của model.

Lưu lại mô hình:

Cuối cùng là lưu lại mô hình: dùng lệnh "Callbacks" để khi mô hình lớn, quá trình huấn luyện (training) gặp sự cố, chương trình sẽ lưu lại mô hình và tiếp tục chạy lại. Trong Callbacks có các hàm: EarlyStopping, ReduceLROnPlateau và ModelCheckpoint. EarlyStopping sẽ ngừng quá trình huấn luyện khi không cải thiện mô hình, ReduceLROnPlateau giảm learning - rate mỗi khi 'metrics' ko được cải thiện và ModelCheckpoint sẽ lưu lại mô hình sau mỗi chu kỳ huấn luyện.

Dùng hàm model.fit() để đưa dữ liệu vào quá trình huấn luyện tìm các tham số mô hình, dùng hàm model.evaluate() đánh giá độ chính xác của mô hình và hàm

H.history() lưu lại các giá trị tham số acc, loss, val_acc và val_loss trong quá trình huấn luyện. Kết hợp Matplotlib để vẽ đồ thị.

Bảng 3.2: Tổng hợp số lượng params cho cả model

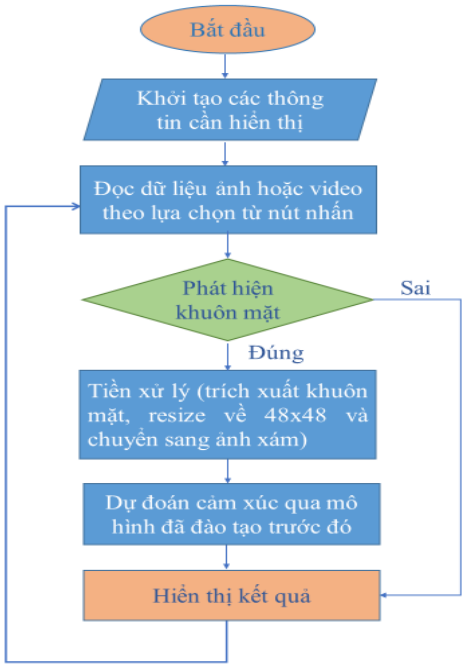
Total params	58,423
Trainable params	56,951
Non-trainable params	1,472

Tổng số Params thu được cuối mô hình là 58,423. Trong đó số lượng trọng số được cập nhật và không được cập nhật trong quá trình huấn luyện là 56,951 và 1,472.

3.5 THIẾT KẾ GIAO DIỆN HIỂN THỊ ẢNH VÀ VIDEO TỪ TKINTER

Hệ thống được xây dựng trên nền tảng OpenCV, Tkinter và Keras dùng ngôn ngữ Python để lập trình thiết kế giao diện (GUI). Bao gồm: Label hiển thị ảnh và cập nhật kết quả, nút Button điều khiển, thời gian thực và mô hình đã huấn luyện trước đó.

Lưu đồ hoạt động của hệ thống:



Hình 3.6: Lưu đồ của chương trình thiết kế giao diện

Tkinter là một gói trong Python có chứa module Tk hỗ trợ cho việc lập trình GUI. Thông qua Tkinter để xây dựng một giao diện cho mô hình nhận dạng cảm xúc trên khuôn mặt. Ứng dụng có chức năng cho phép người dùng tải ảnh bất kỳ hoặc video từ webcam trên máy tính. Bên cạnh đó, ảnh của đối tượng nhận dạng từ webcam sẽ được chụp lại và lưu vào bộ dữ liệu mới.

Ban đầu, khuôn mặt trong ảnh hoặc video từ webcam sẽ được nhận dạng bằng đặc trưng Haar Cascade. Khung dữ liệu thu được chuyển sang phần tiền xử lý, các đặc trưng chi tiết của khuôn mặt sẽ được trích xuất (ROI) và phần thừa được loại bỏ. Ảnh được thay đổi kích thước lưu vào ROI (kích thước 48x48 pixel).

Sau đó, thông qua mô hình CNN đã huấn luyện, hệ thống sẽ dự đoán cảm xúc và trả về kết quả trên khung giao diện.

Chương IV. KẾT QUẢ VÀ KẾT LUẬN

4.1 GIAO DIỆN CHƯƠNG TRÌNH

Hình 4.1: Giao diện chương trình

Trong hình 4.1, giao diện thực hiện nhận diện cảm xúc trên khuôn mặt của đối tượng thông qua ảnh offline hoặc video từ webcam, dưới đây là chức năng của các thành phần trong giao diện:

Nút Open_Image: cho phép chọn ảnh từ thư mục máy tính, hiển thị và nhận diện cảm xúc đối tượng trong ảnh. Trả về kết quả trên khung Label.

Nút Webcam: cho phép hiển thị video từ webcam lên khung Label, trả về kết quả dự đoán cảm xúc trên khuôn mặt theo thời gian thực. Đồng thời chụp lại ảnh từ webcam và lưu vào bộ dữ liệu mới.

Nút Close: cho phép ngưng hiển thị video và tắt webcam, sau khi nhấn nút Close, chương trình sẽ tiếp tục thực hiện khi chuyển sang nút Open_Image.

Nút Quit: cho phép thoát chương trình.

Ảnh và video sau khi được tải lên sẽ hiển thị ở khung label (ảnh nền trung tâm). Phần trăm độ chính xác theo từng nhãn cảm xúc của khuôn mặt được nhận dạng sẽ được hiển thị bên trái khung Label, lần lượt là giận dữ (Angry), ghê tởm (Disgust), sợ hãi (Scared), vui vẻ (Happy), buồn (Sad), ngạc nhiên (Suprised) và bình thường (Neutral).

Đồ thị mô hình huấn luyện 66% hiển thị ở bên phải khung Label.

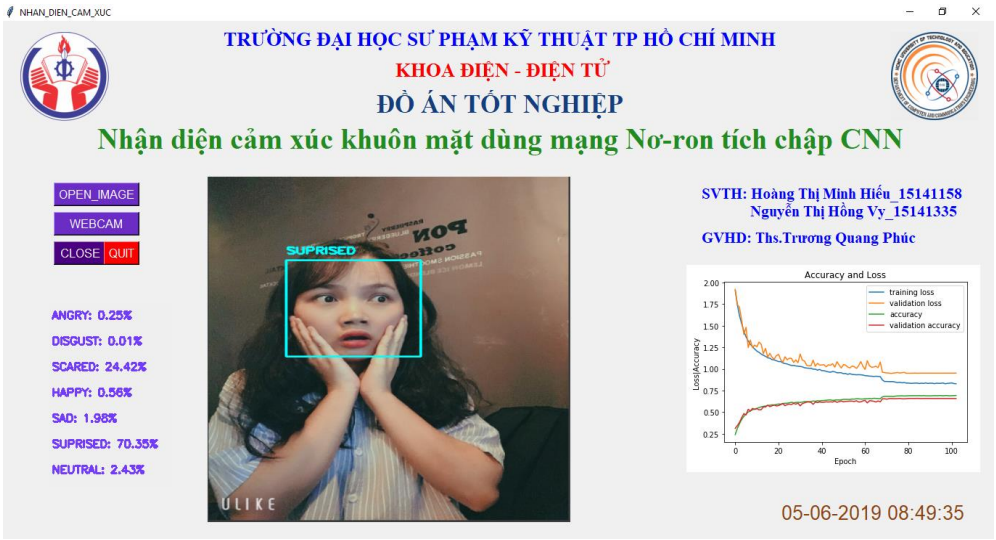
4.2 KẾT QUẢ

4.2.1 Kết quả thực nghiệm trên ảnh

Một số hình ảnh của nhóm tác giả khi đưa vào hệ thống và dưới đây là các kết quả thu được:

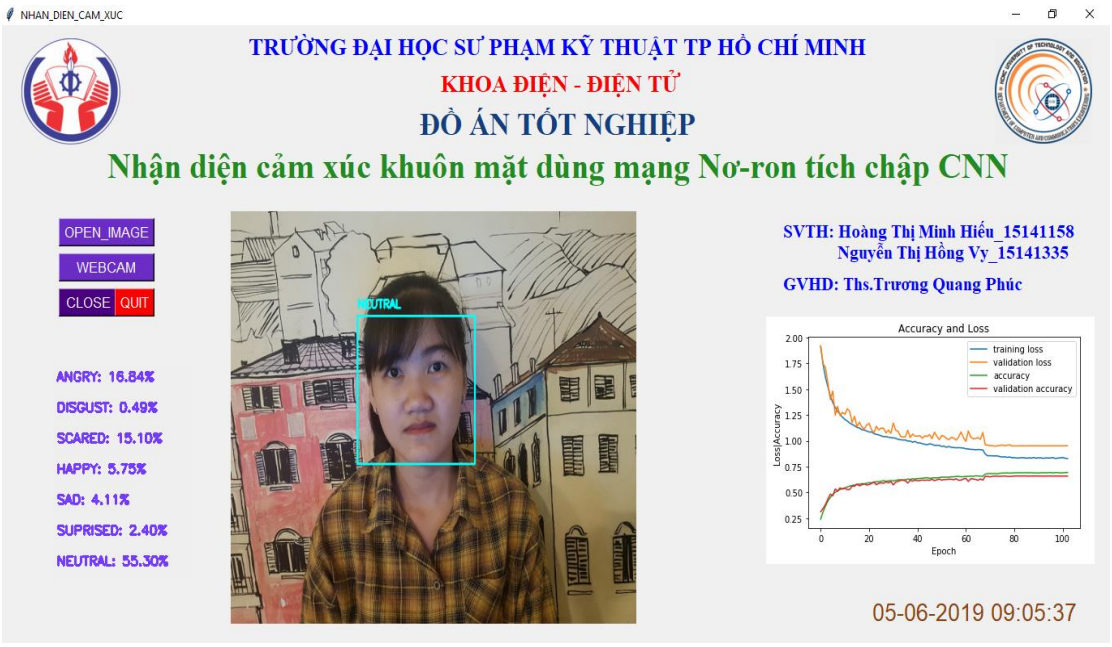
Hình 4.2: Kết quả nhận diện cảm xúc vui trên ảnh

Hình 4.2, cho thấy kết quả nhận diện cảm xúc vui trên ảnh, với độ chính xác cho lớp cảm xúc Happy là 99.20%.



Hình 4.3: Kết quả nhận diện cảm xúc ngạc nhiên trên ảnh

Hình 4.3, cho thấy kết quả nhận diện cảm xúc ngạc nhiên trên ảnh, với độ chính xác cho lớp cảm xúc Suprised là 70.35%.

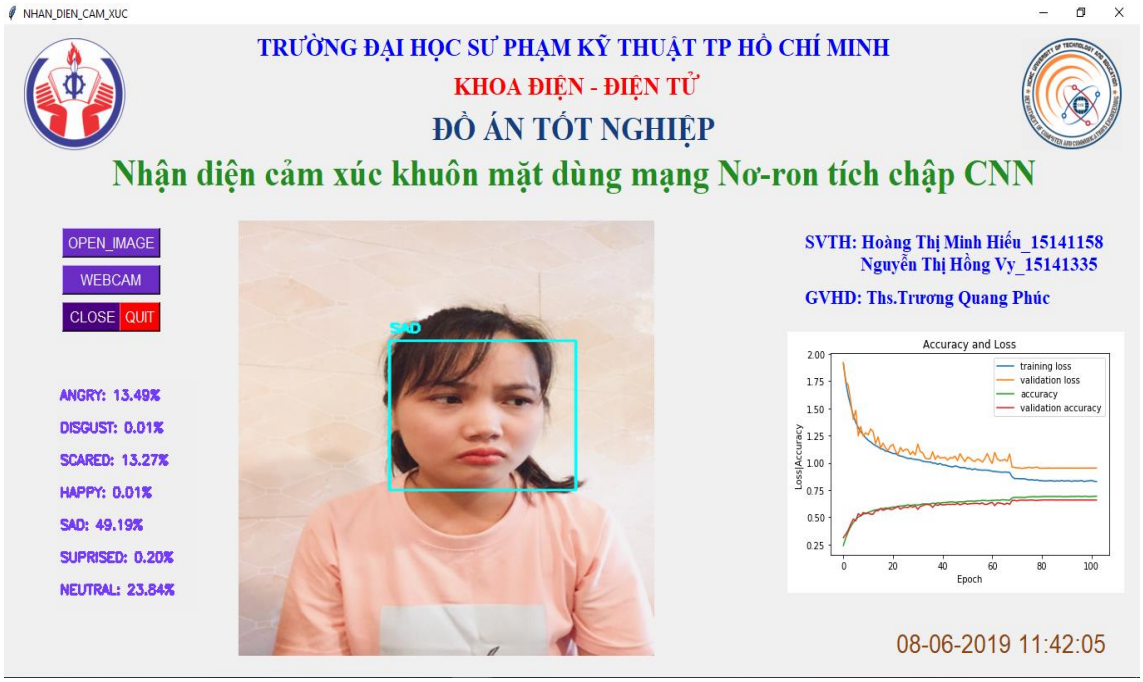


Hình 4.4: Kết quả nhận diện cảm xúc bình thường trên ảnh

Hình 4.4, kết quả nhận diện cảm xúc bình thường trên ảnh, với độ chính xác cho lớp cảm xúc Neutral là 55.30%.

Hình 4.5: Kết quả nhận diện cảm xúc sợ hãi trên ảnh

Hình 4.5, kết quả nhận diện cảm xúc sợ hãi trên ảnh, với độ chính xác cho lớp cảm xúc Scared là 52.87%.



Hình 4.6: Kết quả nhận diện cảm xúc buồn trên ảnh

Hình 4.6, kết quả nhận diện cảm xúc buồn trên ảnh, với độ chính xác cho nhãn cảm xúc Sad là 49.19%.

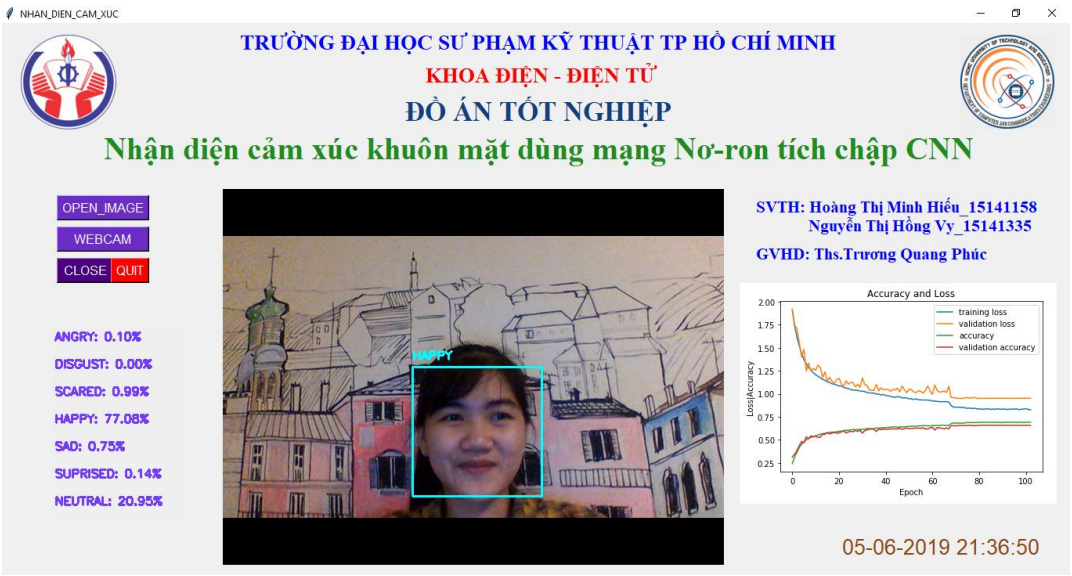
Hình 4.7: Kết quả nhận diện cảm xúc giận dữ trên ảnh

Hình 4.7, kết quả nhận diện cảm xúc giận dữ trên ảnh, với độ chính xác cho lớp cảm xúc Angry là 28.40%.

Hình 4.8: Kết quả nhận diện cảm xúc ghê tởm trên ảnh

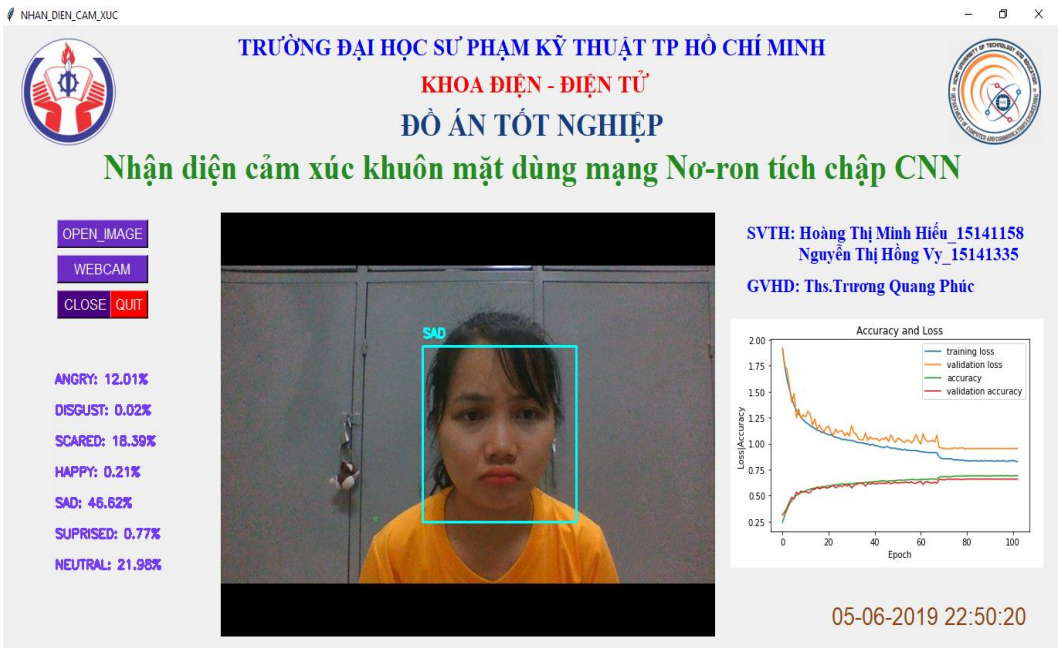
Hình 4.8, kết quả nhận diện cảm xúc ghê tởm trên ảnh, với độ chính xác cho lớp cảm xúc Disgust là 37.14%.

4.2.2 Kết quả thực nghiệm trên video



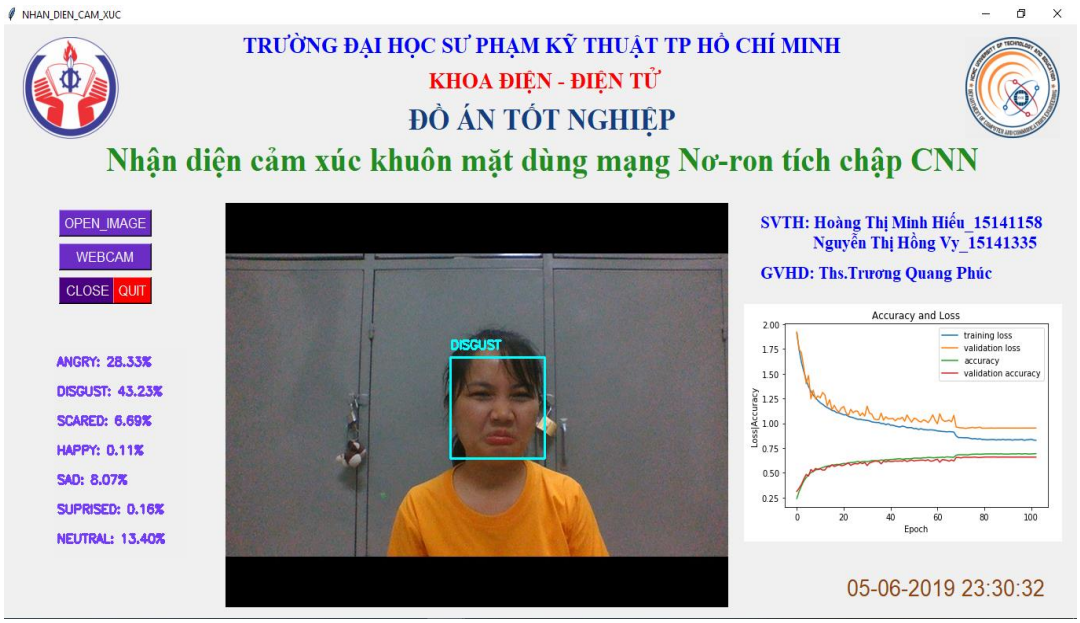
Hình 4.9: Kết quả nhận diện cảm xúc vui trên video

Trong hình 4.9, kết quả nhận diện cảm xúc vui trên video, với độ chính xác cho lớp cảm xúc Happy là 77.08%.



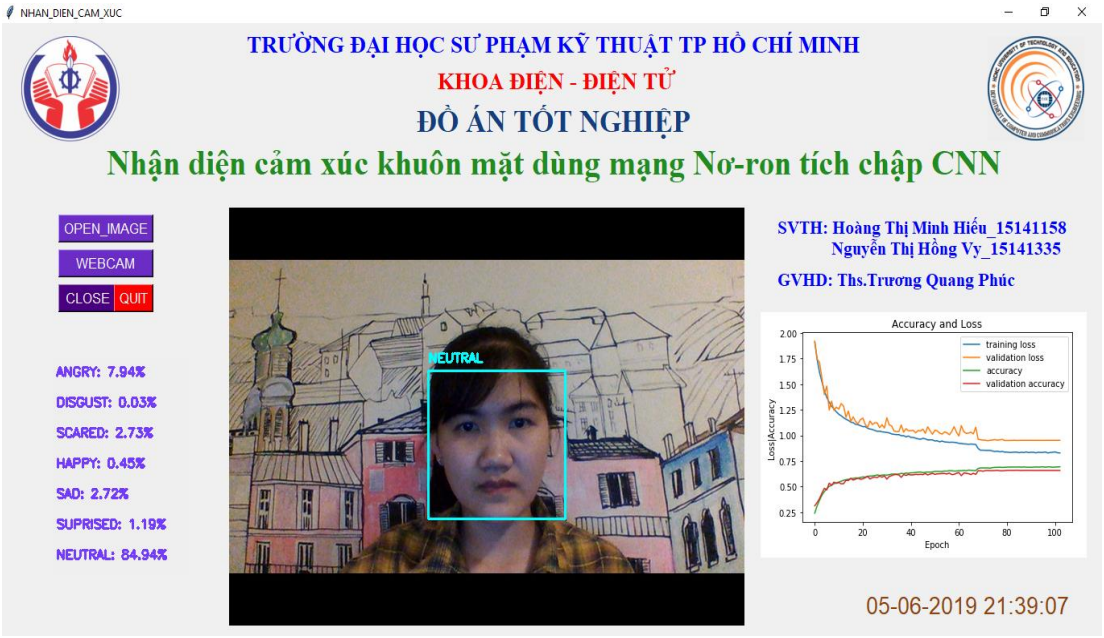
Hình 4.10: Kết quả nhận diện cảm xúc buồn trên video

Trong hình 4.10, kết quả nhận diện cảm xúc buồn trên video, với độ chính xác cho lớp cảm xúc Sad là 46.62%.



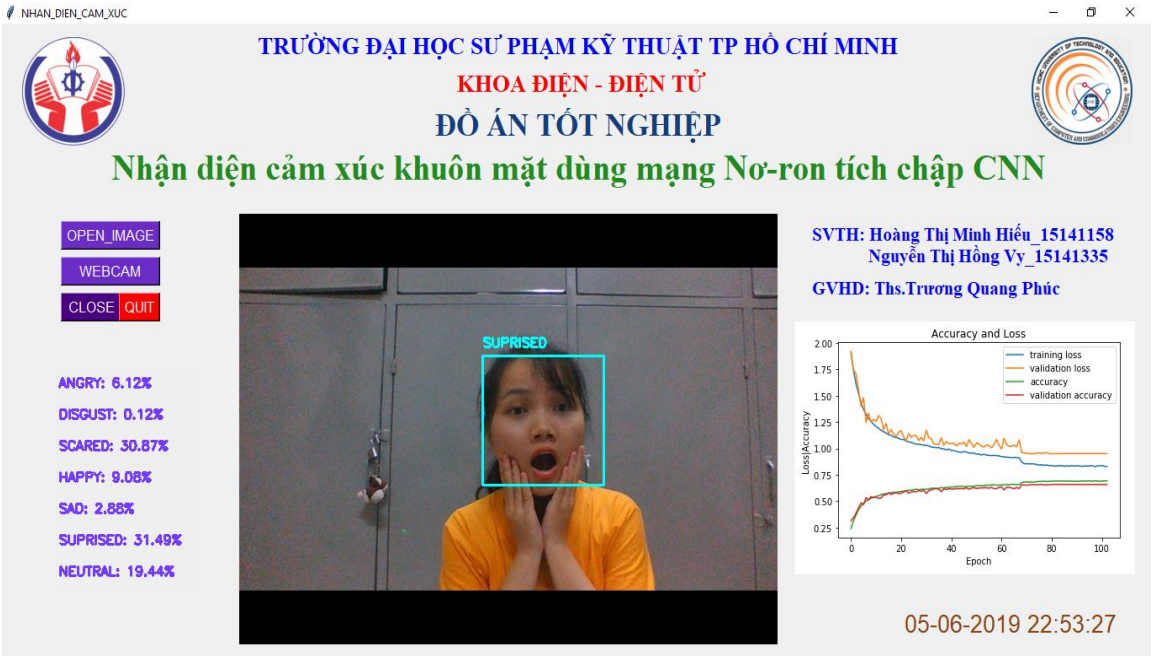
Hình 4.11: Kết quả nhận diện cảm xúc ghê tởm trên video

Trong hình 4.10, kết quả nhận diện cảm xúc ghê tởm trên video, với độ chính xác cho lớp cảm xúc Disgust là 43.23%.



Hình 4.12: Kết quả nhận diện cảm xúc bình thường trên video

Trong hình 4.12, kết quả nhận diện cảm xúc bình thường trên video, với độ chính xác cho lớp cảm xúc Neutral là 84.94%.

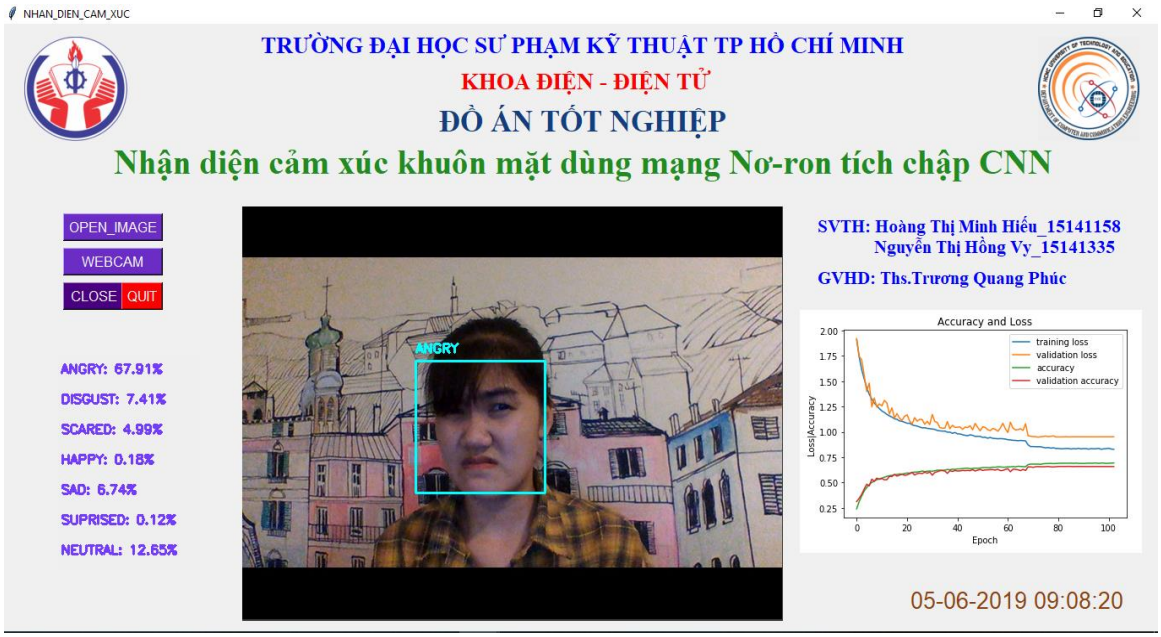


Hình 4.13: Kết quả nhận diện cảm xúc ngạc nhiên trên video

Trong hình 4.13, kết quả nhận diện cảm xúc ngạc nhiên trên video, với độ chính xác cho lớp cảm xúc Suprised là 31.49%.

Hình 4.14: Kết quả nhận diện cảm xúc sợ hãi trên video

Trong hình 4.14, kết quả nhận diện cảm xúc sợ hãi trên video, với độ chính xác cho lớp cảm xúc Scared là 61.28%.

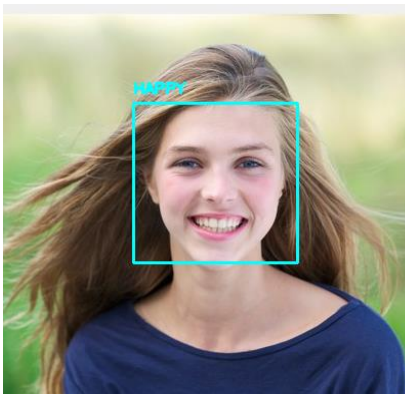


Hình 4.15: Kết quả nhận diện cảm xúc giận dữ trên video

Trong hình 4.15, kết quả nhận diện cảm xúc giận dữ trên video, với độ chính xác cho lớp cảm xúc Angry là 67.91%.

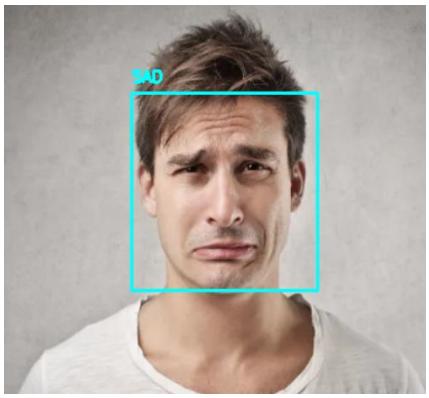
4.3 THỐNG KÊ KẾT QUẢ THỰC NGHIỆM TRÊN ẢNH BẤT KỲ

4.3.1 Kết quả thực nghiệm trên ảnh đúng

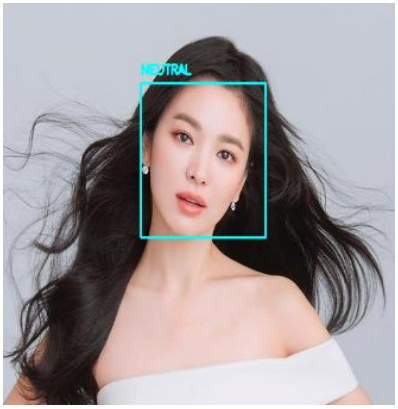


a) emotion_1

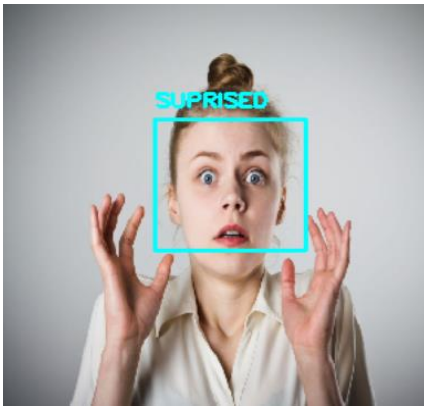
b) emotion_2



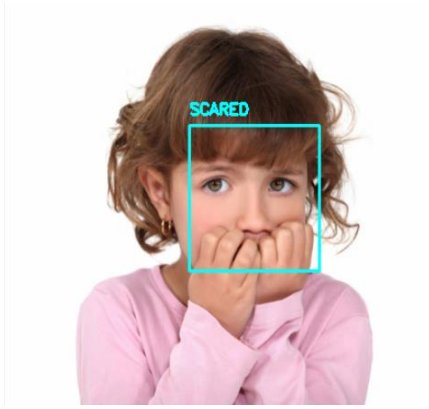
c) emotion_3



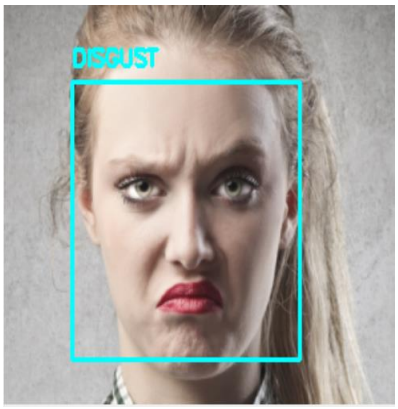
d) emotion_4



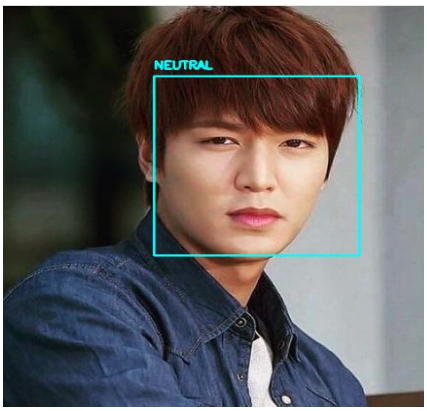
e) emotion_5



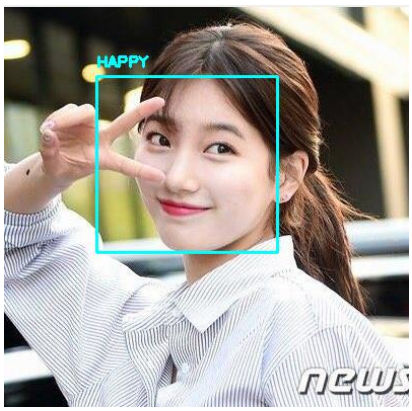
f) emotion_6



g) emotion_7



h) emotion_8



i) emotion_9

Hình 4.16: Một số kết quả dự đoán cảm xúc đúng

Trong hình 4.16, là kết quả nhận dạng ảnh bất kỳ, phần trăm trả về cho từng nhãn cảm xúc tương đối đúng với biểu hiện của đối tượng trong ảnh.

Bảng 4.1: Tỉ lệ cảm xúc các ảnh nhận diện đúng

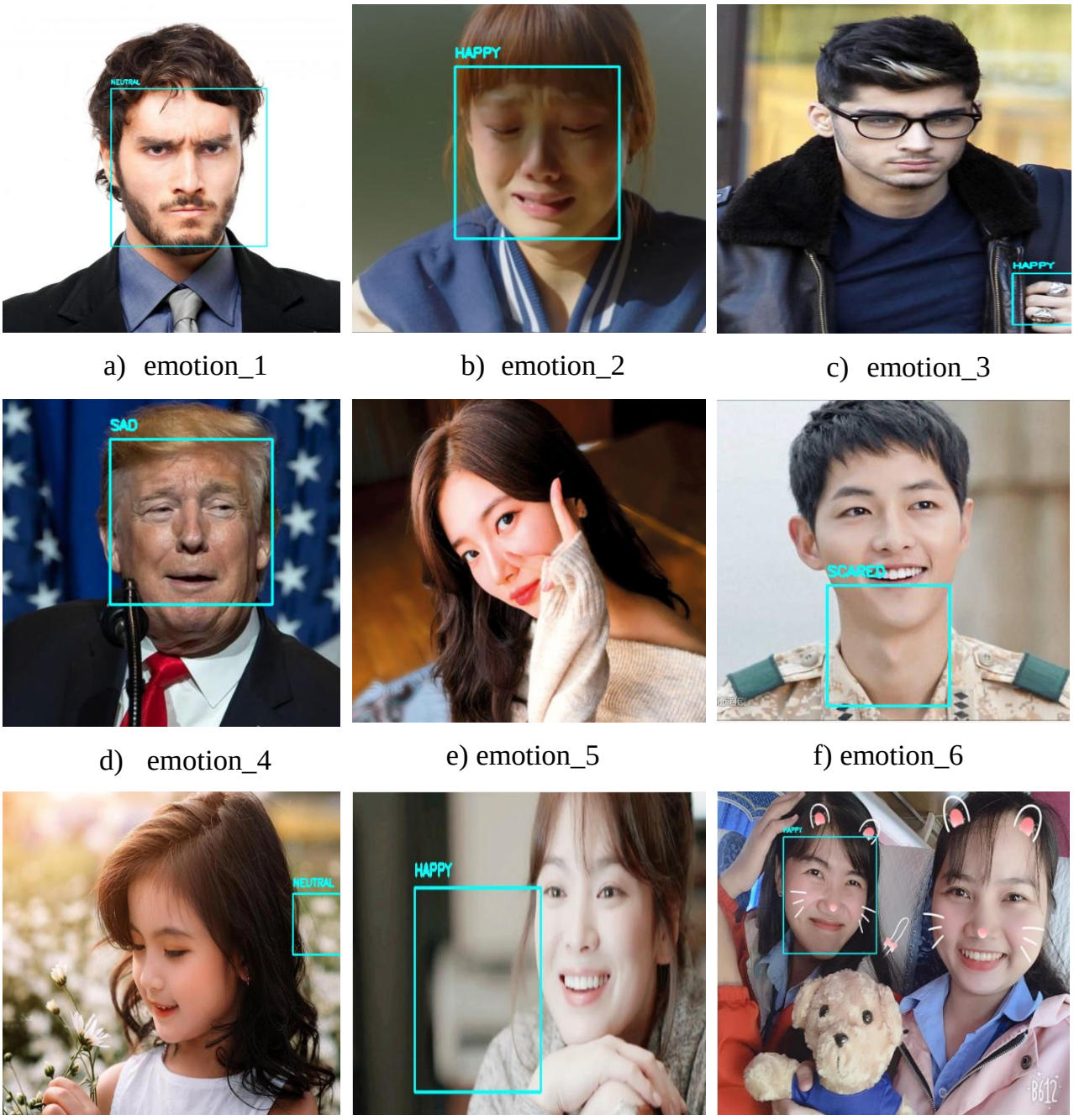
Ảnh	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
Giận dữ	0.13	34.29	10.91	0.26	0.07	0.73	24.8	1.06	0.00
Ghê tởm	0.00	0.09	0.02	0.00	0.00	0.00	69.17	0.01	0.00
Sợ hãi	0.02	28.99	28.56	0.58	22.12	92.75	2.15	0.93	0.00
Vui vẻ	97.19	11.25	0.31	4.17	0.01	0.05	0.01	0.05	99.73
Buồn	0.02	11.04	46.32	1.60	0.25	1.08	2.15	1.03	0.00
Ngạc nhiên	0.06	4.37	3.18	0.32	77.54	4.56	1.2	0.01	0.17
Bình thường	2.58	9.97	10.69	93.06	0.01	0.84	0.52	96.91	0.10

Trong bảng 4.1, đối với các cảm xúc vui vẻ, ngạc nhiên, bình thường và sợ hãi, hệ thống nhận diện với tỉ lệ khá tốt (> 77%). Tuy nhiên đối với cảm xúc buồn và giận dữ thì tỉ lệ dự đoán thấp hơn. Riêng cảm xúc ghê tởm, hệ thống có tỉ lệ dự đoán thấp nhất.

Kết quả nhận dạng đúng các trường hợp như ảnh có vùng khuôn mặt sáng, chi tiết đặc trưng của khuôn mặt không bị che khuất (tóc, bóng râm,...), ảnh có độ phân giải cao và có hướng mặt nhìn thẳng, hướng nhìn nghiêng 45^0 và hướng nhìn từ trên xuống.

Nhưng vẫn có trường hợp hệ thống trả về kết quả nhận diện cảm xúc sai và không nhận diện được đúng vị trí khuôn mặt đối tượng.

4.3.2 Kết quả thực nghiệm trên ảnh sai



a) emotion_1 b) emotion_2 c) emotion_3

d) emotion_4 e) emotion_5 f) emotion_6

g) emotion_7 h) emotion_8 i) emotion_9

Hình 4.17: Một số kết quả dự đoán cảm xúc sai

Trong hình 4.17, là kết quả các trường hợp nhận dạng sai đối tượng, phần trăm trả về cho từng nhãn cảm xúc tương đối sai với biểu hiện của đối tượng trong ảnh.

Bảng 4.2: Tỉ lệ cảm xúc của ảnh nhận diện sai

Ảnh	a	b	c	d	e	f	g	h	i
Giận dữ	12.82	0.53	1.14	3.84	0.00	17.75	8.84	3.85	1.97
Ghê tởm	0.01	0.00	0.43	0.04	0.00	0.19	0.58	0.01	0.01
Sợ hãi	12.48	3.62	0.19	10.21	0.00	23.41	19.23	3.79	3.97
Vui vẻ	0.22	69.15	96.68	2.78	0.00	21.12	15.81	83.62	52.81
Buồn	11.66	20.51	0.22	76.65	0.00	8.28	11.08	1.82	1.96
Ngạc nhiên	0.48	0.04	0.03	0.27	0.00	6.66	8.27	3.62	0.42
Bình thường	62.33	6.15	1.31	7.21	0.00	22.60	36.19	3.29	38.87

Hệ thống xảy ra hạn chế trong các trường hợp như đối tượng trên ảnh không thể hiện rõ cảm xúc, ảnh đầu vào trong điều kiện thiếu ánh sáng, ảnh có độ phân giải thấp và có các chi tiết không phải là đặc trưng khuôn mặt (bóng râm, màu sắc môi trường xung quanh), khuôn mặt đối tượng bị che khuất, hướng mặt nghiêng quá 45^0. Dẫn đến nhận diện sai vùng được cho là khuôn mặt, nhận diện sai cảm xúc và không nhận diện được vùng khuôn mặt nên không hiển thị được cảm xúc của đối tượng. Ngoài ra, hệ thống không thể nhận diện cảm xúc tất cả các đối tượng có trong ảnh.

4.4 KẾT QUẢ HUẤN LUYỆN MÔ HÌNH MINI_XCEPTION TRÊN TẬP DỮ LIỆU FER2013

4.4.1 Kết quả huấn luyện (training) mô hình

Từ thiết kế mô hình Mini_Xception, mô hình huấn luyện sử dụng tập dữ liệu FER2013 bao gồm một tập hình ảnh với kích thước 48x48 pixels định dạng ảnh xám. Quá trình huấn luyện qua 100 chu kỳ dữ liệu (100 epoch) với tổng thời gian

huấn luyện khoảng 8 giờ, thực hiện trên phần cứng laptop DELL (DESKTOP-LVFF0SH) Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70GHz. Bộ nhớ RAM 4.00GB (3.87 GB usable) 64-bit.

Dựa vào thực nghiệm, khi thay đổi giá trị của epochs, batch_size, validation_split nhiều lần, dưới đây là các mô hình sau khi huấn luyện được với độ chính xác gần 65%.

Batch_size = 32, Epoch = 100, Validation_split = 0.4. Độ chính xác 63%

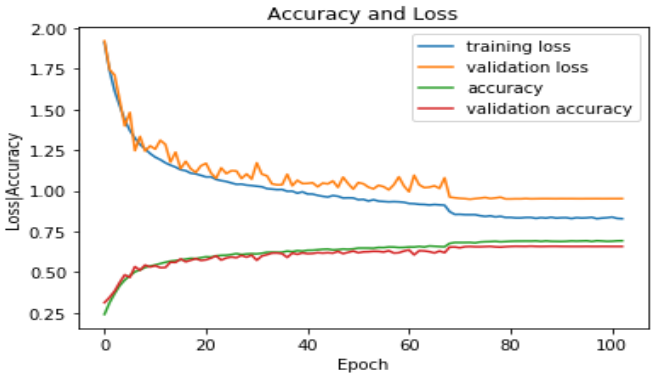
a)

Batch_size = 128, Epoch = 100, Validation_split = 0.4. Độ chính xác 65%

b)

Hình 4.18: Các trường hợp huấn luyện mô hình với độ chính xác thấp hơn 65%

Trong hình 4.18, các thông số loss và val_loss giảm từ 2.0 xuống 0.9 dừng lại ở mức 0.9, giá trị val_loss chênh lệch xảy ra hiện tượng overfitting, mô hình huấn luyện đúng với tập training_set nhưng dự đoán không chính xác với tập validation_set. Các val_acc sau mỗi Epoch tiến tới gần 0.63 (hình a) và 0.65 (hình b) rồi đạt trạng thái bão hòa. Độ chính xác cho các mô hình này tương đối thấp khoảng gần 63%-65%.



Hình 4.19: Đồ thị quá trình huấn luyện mô hình đạt 66%

Trong hình 4.19, là đồ thị kết quả mô hình sau khi thay đổi batch_size: 32, epochs: 110 và validation_split: 0.2.

Lần huấn luyện thứ 102 mô hình cho ra kết quả cao nhất (66%), kết quả đã cải thiện sau rất nhiều lần chỉ dừng lại ở 63-65%. Khi huấn luyện tới epochs là 110 mô hình dừng lại, nhưng trên đồ thị mô hình chưa bão hòa (do các đường trên mô hình chưa thẳng).

4.4.2 Đánh giá mô hình

Trong quá trình huấn luyện mô hình, các thông số sẽ được cập nhật liên tục sao cho sai số ở ngõ ra đạt đến mức thấp nhất. Sau khi có kết quả thì mô hình cho ra được biểu đồ về sự thay đổi của các giá trị như loss, acc, val_loss, val_acc.

Khi huấn luyện tới chu kì dữ liệu cuối cùng thì mô hình dừng lại, độ chính xác của quá trình huấn luyện tăng dần theo chu kỳ dữ liệu. Trong các mô hình đã huấn luyện, giá trị val_acc hầu như không thay đổi ở các chu kỳ huấn luyện tiếp theo.

4.5 ĐÁNH GIÁ HỆ THỐNG

4.5.1 Những khó khăn trong việc nhận diện cảm xúc trên khuôn mặt người

Trong cùng một bức ảnh có nhiều khuôn mặt khác nhau và nhiều cảm xúc khác nhau. Hướng khuôn mặt nhìn thẳng, nhìn nghiêng hoặc nhìn từ trên xuống.

Các chi tiết không phải đặc trưng của khuôn mặt người: mắt kính, nón, phụ kiện, bóng râm và môi trường xunh quanh. Màu sắc của môi trường xung quanh hay màu sắc quần áo của đối tượng.

Mặt người bị che khuất bởi các đối tượng có trong ảnh. Đặc biệt, những vùng không phải là khuôn mặt có độ sáng tương đối tốt hoặc có những đặc điểm giống khuôn mặt.

Sự phức tạp của cảm xúc trên khuôn mặt người quá lớn (gần như giống nhau giữa các cảm xúc), và sự thay đổi linh hoạt cảm xúc của con người nên hệ thống chưa thể nhận dạng đúng hoàn toàn cảm xúc khuôn mặt người.

Bộ dữ liệu FER2013 dùng cho huấn luyện mô hình có các hạn chế do nguồn ảnh ngõ vào là các hình ảnh nhỏ (48x48 pixel) nên đầu vào mô hình nhỏ. Ảnh bị thay đổi kích thước từ có độ phân giải cao thành độ phân giải thấp, vì thế nên mất nhiều chi tiết ảnh có độ phân giải cao, làm giảm hiệu suất mô hình huấn luyện. Ngoài ra trong bộ dữ liệu này có chứa các ảnh không phải mặt người (ảnh hoạt hình, cây lá, cửa kính, tóc,....) làm ảnh hưởng đến sự phân lớp các đặc trưng cảm xúc khi huấn luyện mô hình.

4.5.2 Đánh giá hệ thống

Nhận dạng được cảm xúc trên khuôn mặt nhưng chưa hoàn toàn đúng.

Tỷ lệ xác định chính xác chưa cao: kết quả nhận diện trả về có số lượng khuôn mặt được hiển thị cảm xúc ít hơn số lượng người thật sự có trong ảnh.

Nhận dạng sai cảm xúc của đối tượng.

Xác định sai vùng được cho là khuôn mặt dẫn đến không nhận diện được cảm xúc trong ảnh hoặc trường hợp nhận diện được cảm xúc nhưng trả về kết quả hiển thị trên vùng không phải là khuôn mặt.

Chương V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Sau quá trình thực hiện, nhóm tác giả đã hoàn thành được mục tiêu đề ra. Nhận diện cảm xúc trên khuôn mặt dùng mạng Nơ-ron tích chập CNN, với mô hình kiến trúc Mini_Xception đạt độ chính xác là 66% thấp hơn so với yêu cầu bài toán đặt ra (70%) nhưng vẫn nhận diện chính xác cảm xúc trên khuôn mặt người ở mức tương đối.

Nhận diện được cảm xúc của đối tượng trong điều kiện vùng khuôn mặt sáng, khuôn mặt không bị che khuất, hướng khuôn mặt nhìn thẳng, nhìn nghiêng 45^0 hoặc nhìn từ trên xuống và trường hợp đối tượng nhận dạng ở phạm vi gần (gần 2m). Khi có nhiều khuôn mặt khác nhau trong ảnh hoặc từ nguồn webcam, hệ thống sẽ trả về kết quả cho khuôn mặt gần nhất, có vùng khuôn mặt sáng và ảnh có chất lượng tốt. Tỷ lệ nhận dạng chính xác các cảm xúc vui vẻ, ngạc nhiên và bình thường khá cao (thường cao hơn 80%), cảm xúc buồn và giận dữ tỷ lệ dự đoán thấp, riêng với cảm xúc ghê tởm tỷ lệ dự đoán là thấp nhất.

Tuy nhiên, còn có một số hạn chế mà hệ thống chưa khắc phục được. Trong các trường hợp đối tượng trên ảnh không thể hiện rõ cảm xúc, ảnh trong điều kiện thiếu ánh sáng, ảnh có độ phân giải thấp và có các chi tiết không phải là đặc trưng khuôn mặt sẽ dẫn đến nhận diện sai vùng được cho là khuôn mặt, nhận diện sai cảm xúc của đối tượng. Ngoài ra, hệ thống không thể nhận diện cảm xúc của nhiều người cùng lúc.

Mặt dù số lượng hình ảnh trong tập dữ liệu FER2013 rất lớn, nhưng đó có nhiều hình ảnh không phải là ảnh người, chẳng hạn như: ảnh hoạt hình, ảnh cửa hoặc ảnh mặt người bị lệch. Điều đó làm cho việc nhận diện cảm xúc trên khuôn mặt trở nên khó khăn hơn. Các cảm xúc thường không được bộc lộ một cách rõ ràng trên khuôn mặt người, khiến việc nhận diện cảm xúc cũng không được chính xác hoàn toàn.

5.2 HƯỚNG PHÁT TRIỂN

Từ những hạn chế của hệ thống, nhóm tác giả sẽ tập trung vào việc nâng cao chất lượng của mô hình, bằng cách bổ sung thêm hình ảnh khuôn mặt người châu Á (dữ liệu ảnh trong các bộ dữ liệu FER2013 chủ yếu là khuôn mặt của người châu Âu) và loại bỏ những hình ảnh không phải khuôn mặt. Cân bằng tỉ lệ dữ liệu hình ảnh khuôn mặt giữa các nhãn cảm xúc. Ngoài ra, hỗ trợ phân biệt giới tính cùng lúc với cảm xúc.

Xây dựng phần cứng hệ thống vì đây là đề tài có tính ứng dụng cao, có thể áp dụng vào nhiều vấn đề trong thực tế, đặc biệt là liên quan đến việc phản hồi của khách hàng.

TÀI LIỆU THAM KHẢO

[1] S. Nguyen, “Medium: Thế kỉ 21: Khoa học công nghệ và những bước tiến vượt thời gian,” 31 01 2018. [Online]. Available: https://bit.ly/2J3fOBy.

[2] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Arthur_Samuel.

[3] Thành Luân, Anh Quân, “Công nghệ,” Thanh Niên, 09 05 2018. [Online]. Available: https://thanhnien.vn/cong-nghe/google-photos-them-tinh-nang-to-mau-cho-anh-den-trang-960793.html.

[4] N. Trang, “VnEconomy,” 08 07 2018. [Online]. Available: http://vneconomy.vn/cong-cu-tri-tue-nhan-tao-cua-alibaba-viet-20000-quang-cao-moi-giay-20180706094721953.htm.

[5] T. Lam, “Invest TV,” 17 12 2018. [Online]. Available: http://investtv.vn/phan-mem-mms-net-dat-giai-vang-tai-trien-lam-quoc-te-siif-2018-n4872.html.

[6] “Trang thông tin điện tử tập đoàn Vingroup,” VINGROUP, 17 04 2019. [Online]. Available: http://vingroup.net/vi-vn/tin-tuc-su-kien/tin-tuc-hoat-dong/vingroup-thanh-lap-vien-nghien-cuu-tri-tue-nhan-tao-ai-3362.aspx.

[7] H.T.Nguyễn, Giáo trình xử lý ảnh, TP Hồ Chí Minh: Đại học Quốc Gia, 2014.

[8] H.Tạ, "Lập trình MATLAB," Matlabthayhai, 2018. [Online]. Available: http://www.matlabthayhai.info/2015/11/bai-1-gioi-thieu-khai-quat-ve-anh-so.html.

[9] "Bitcoin Vietnam News," [Online]. Available: https://bitcoinvietnamnews.com/deep-learning-la-gi. [Accessed 7 5 2019].

[10] "VIA News," [Online]. Available: https://www.viatech.com/en/2018/05/history-of-artificial-intelligence/

[11] "FPT-Aptech," [Online]. Available: http://aptech.fpt.edu.vn/chitiet.php?id=5337

[12] P. Thanh, "iDesign," 11 08 2018. [Online]. Available: https://idesign.vn/graphic-design/deep-learning-trong-tri-tue-nhan-tao-ai-257065.html.

[13] D. Duong, "TechBlog," [Online]. Available: https://techblog.vn/recurrent-neural-networkphan-1-tong-quan-va-ung-dung.

[14] "TIENDV'S BLOG," 25 12 2016. [Online]. Available: https://tiendv.wordpress.com/2016/12/25/convolutional-neural-networks/. [Accessed 13 3 2019].

[15] T. Nguyen, "VIBLO," 27 9 2016. [Online]. Available: https://viblo.asia/p/mang-no-ron-tich-chap-p1-DZrGNNjPGVB. [Accessed 13 3 2019].

[16] N. T. Toan, "TechBlog," [Online]. Available: https://techblog.vn/su-dung-cnn-trong-bai-toan-nhan-dang-mat-nguoi-phan-1.

[17] ChhoeungYean, "SCRIBD," 02 03 2019. [Online]. Available: https://vi.scribd.com/document/408384865/ChhoeungYean-Luan-Van-docx.

[18] "Dlapplications: dl_ap," 17 07 2018. [Online]. Available: https://dlapplications.github.io/2018-07-17-cnn-introduction/.

[19] N. P. Luong, "VIBLO," 28 11 2017. [Online]. Available: https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM.

[20] N. Tuan, "Deep Learning co ban," 30 3 2019. [Online]. Available: https://nttuan8.com/bai-6-convolutional-neural-network/. [Accessed 4 4 2019].

[21] "Machine Learning bites," 24 02 2018. [Online]. Available: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524

[22] T. V. Dev, "TVD," 24 06 2018. [Online]. Available: https://teamvietdev.com/opencv-la-gi-ung-dung-opencv-trong-the-gioi-thuc/.

[23] H. D. Thoi, "VIBLO," 15 09 2018. [Online]. Available: https://viblo.asia/p/deep-learning-qua-kho-dung-lo-da-co-keras-LzD5dBqoZjY.

[24] "Keras," [Online]. Available: https://keras.io/.

[25] J. Flores, “Training a TensorFlow model to recognize emotions,” 24 5 2018. [Online]. Available: https://medium.com/@jsflo.dev/training-a-tensorflow-model-to-recognize-emotions-a20c3bcd6468. [Đã truy cập 4 4 2019].

[26] P. G. P. M. V. Octavio Arriaga, “Real-time Convolutional Neural Networks for Emotion and Gender Classification,” 2017.

[27] N. N. Vịnh, “VIBLO,” 28 10 2017. [Online]. Available: https://viblo.asia/p/vanishing-exploding-gradients-problems-in-deep-neural-networks-part-2-ORNZqPEeK0n#_batch-normalization-9.

[28] W. Chi-Feng, “Medium,” 14 04 2018. [Online]. Available: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728.

[29] P. V. Toàn, “[Handbook CV with DL - Phần 3] Bài toán phân loại hình ảnh - Image Classification với Keras,” [Online]. Available: https://techblog.vn/handbook-cv-with-dl-phan-3-bai-toan-phan-loai-hinh-anh-image-classification-voi-keras.

[30] “MachineLearningcoban,” 04 03 2017. [Online]. Available: https://machinelearningcoban.com/2017/03/04/overfitting/#-regularization.