ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NHẬN DIỆN CẢM XÚC MẶT NGƯỜI SỬ DỤNG MẠNG HỌC SÂU CÓ CHÚ Ý

Hội đồng LVTN: Khoa học máy tính

Giảng viên hướng dẫn: TS. Trần Tuấn Anh

Giảng viên phản biện: TS. Nguyễn Hồ Mẫn Rạng

Sinh viên thực hiện: Phạm Quí Luận

TP. HỒ CHÍ MINH, THÁNG 12/2019

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn của TS. Trần Tuấn Anh. Nội dung nghiên cứu và các kết quả của nghiên cứu đều là trung thực và chưa từng được công bố trước đây. Các số liệu được sử dụng cho quá trình phân tích, nhận xét được chính tôi thu thập từ nhiều nguồn khác nhau và được ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, tôi cũng có sử dụng một số nhận xét, đánh giá và số liệu của các tác giả khác, cơ quan tổ chức khác. Tất cả đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kì sự gian lận nào, tôi xin hoàn toàn chịu trách nhiệm về nội dung của mình. Trường Đại học Bách Khoa Thành phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện luận văn này.

Lời cảm ơn

Đầu tiên, tôi xin cảm ơn thầy Trần Tuấn Anh vì sự quan tâm, giúp đỡ của thầy đối với tôi trong suốt những năm qua, trong học tập và cả công việc. Những năm cuối khóa, thật may mắn khi tôi được gặp và làm việc chung với thầy.

Cảm ơn Bách Khoa đã cho tôi những tháng ngày sinh viên đẹp tuyệt vời. Bốn năm học trôi qua, Bách Khoa cho tôi được gặp và làm quen với rất nhiều bạn bè tài giỏi, cho tôi được gặp những thầy cô rất khắt khe, nhưng luôn yêu thương sinh viên theo một cách nào đó. Bách Khoa cho tôi thấy tự hào vì mình là một trong một phần lịch sử của nó, còn cho tôi thấy khiêm nhường hơn qua môn Giải tích 2 hay bốn cái assignment của môn Nguyên lý ngôn ngữ lập trình. Có giáo viên đã nói với tôi: "Em sẽ chỉ có thể cảm nhận tình yêu em dành cho Bách Khoa khi em không còn ở đây nữa", giờ thì tôi mới thấm

thía. Cảm ơn tất cả vì đã cho tôi tham gia vào.

Sau cùng, tôi cảm ơn em, vì đã đến và đã đi.

Sài Gòn những ngày đầu tháng Một, tôi dành trọn vẹn con tim mình để viết những lời này 2

Tóm tắt luận văn

Phát hiện và phân tích cảm xúc từ các chuyển động trên khuôn mặt người là một bài toán đã được định nghĩa và phát triển trong rất nhiều năm vì những lợi ích mà nó mang lại. Trong quá trình phát triển, các bộ dữ liệu cũng với các phương pháp trở nên phức tạp dần và độ chính xác, độ khó cũng tăng dần. Trong luận văn này, tôi sẽ sử dụng một mạng học sâu có sử dụng cơ chế chú ý - Residual Masking Network để phân lớp cảm xúc dựa trên ảnh đầu vào trong môi trường phức tạp, bên cạnh đó tôi còn sử dụng phương pháp học kết hợp nhiều mô hình hiện đại để tăng cường độ chính xác. Ngoài ra, tôi còn xây dựng và phát triển một bộ dữ liệu có mặt người Việt Nam nhằm góp phần phát triển bài toán này ở đất nước ta. Kết quả thực nghiệm cho thấy hai phương pháp đề xuất đều có kết quả tốt hơn một số phương pháp hiện đại trong bài toán nhận diện cảm xúc cho ảnh đầu vào phức tạp và một số kết quả được báo cáo trong các nghiên cứu khoa học. Riêng phương pháp học kết hợp cho ra độ chính xác tốt nhất hiện nay - 76.82% trên tập dữ liệu FER2013.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Chương 1

Giới thiệu

Nét mặt hay còn gọi là biểu cảm khuôn mặt con người (facial expression) đóng một vai trò rất quan trọng trong giao tiếp xã hội. Một cuộc trò chuyện bình thường bao gồm các yếu tố ngôn ngữ và phi ngôn ngữ. Các yếu tố phi ngôn ngữ bao gồm giao tiếp bằng mắt, cử chỉ, nét mặt, và ngôn ngữ cơ thể, v.v. Một nụ cười cho thấy sự hạnh phúc, một nét mặt buồn bã cho thấy sự mất mát, một nét giận dữ cho thấy sự không vui và một nét mặt bất ngờ cho thấy một điều không đoán trước đã xảy ra. Theo C. Darwin và P. Prodger [7], nét mặt là một trong những tín hiệu phổ quát, tự nhiên và mạnh mẽ của con người để truyền tải ý định và trạng thái cảm xúc của họ. Trong lĩnh vực thị giác máy tính và học máy, rất nhiều nhà khoa học đã tiến hành nghiên cứu về các hệ thống phân tích nét mặt tự động bởi vì những ứng dụng thực tế quan trọng của nó trong các hệ thống tương tác người máy, qua đó, nhiều hệ thống nhận dạng nét mặt đã cố gắng mã hóa biểu cảm từ những thể hiện trên khuôn mặt.

Từ đầu thế kỷ 20, Ekman and Friesen [11] đã định nghĩa sáu nét mặt cơ bản dựa trên các nghiên cứu của họ, họ khẳng định rằng cảm xúc con người là phổ quát, nghĩa là con người nhận thức các biểu cảm là giống nhau bất kể họ đến từ nền văn hóa nào. Các nét mặt cơ bản là giận dữ, ghê tởm, sợ hãi, hạnh phúc, buồn bã, ngạc nhiên và trạng thái bình thường. Biểu cảm khinh bỉ được thêm vào sau này như một trong các nét mặt cơ bản. Tuy vậy, các nghiên cứu gần đây về khoa học thần kinh và tâm lý học đã lập luận rằng mô hình của sáu nét mặt cơ bản mang tính chất đặc thù văn hóa và không phổ quát[25]. Nhưng vì tính chất lịch sử, các tập dữ liệu đều được xây dựng dựa trên giả thuyết phổ quát của biểu cảm khuôn mặt. Hiện nay có một số nhà nghiên cứu Trung Quốc đã thực hiện thu thập

và phát triển những bộ dữ liệu nhận diện biểu cảm cho người Trung Quốc như Ma, Jialin và đồng nghiệp [35], góp phần đẩy mạnh triển khai ứng dụng bài toán này trong công nghiệp của Trung Quốc [24]. Vì đó tôi thực hiện xây dựng một bộ dữ liệu chứa ảnh người Việt Nam cho bài toán này, đồng thời thực nghiệm để so sánh với một số phương pháp hiện đại (xem Chương 6).

Hiện tay, các hệ thống nhận diện nét mặt có thể được chia ra làm hai loại chính dựa theo đầu vào hay là thiết kế của hệ thống ấy:

• Hệ thống dựa vào ảnh tĩnh.

• Hệ thống dựa vào chuỗi ảnh động.

Hệ thống dựa vào ảnh tĩnh là chỉ phân tích cảm xúc của riêng từng tấm ảnh dựa vào các thông tin sẵn có thông tấm ảnh đó, chủ yếu là trích xuất đặc trưng dựa vào vị trí tương đối của các thành phần có trong tấm ảnh ấy, xem Hình 4.4. Còn đối với hệ thống dựa vào chuỗi ảnh động, thì hệ thống xem xét cả thông tin về nhịp độ (temporal) của một chuỗi ảnh liên tục, nó không chỉ xem xét sự tương quan về không gian (spatial) trong từng tấm ảnh, và còn xem xét sự tương quan ấy giữa các bức ảnh với nhau. Cả hai hệ thống ấy đều được khoa học quan tâm và nghiên cứu đến. Trong luận văn này, tôi sẽ nghiên cứu phát triển hệ thống dựa vào ảnh tĩnh.

1.1 Đặt vấn đề

Nhận diện cảm xúc con người đóng một vai trò quan trọng trong các hệ thống tương tác người máy. Có nhiều phương thức có thể dùng để nhận dạng ra cảm xúc của một con người, từ giọng nói, biểu cảm, đến cử chỉ, hay thậm chí là chỉ số điện não đồ (Electroencephalography - EEG). Nhìn với con mắt khoa học, bằng việc phân tích cảm xúc con người chúng ta sẽ có thêm một yếu tố để phân tích và thấu hiểu con người. Do vậy mà nó có những tác động và ứng dụng rất lớn trong đời sống cũng như trong nghiên cứu của các lĩnh vực khác nhau.

Hình 1.1: Các biểu cảm cơ bản trên khuôn mặt người. bắt đầu từ bên trái cùng: Giận dữ, Ghê tởm, Sợ hãi, Hạnh phúc, Buồn bã và Ngạc nhiên. [54]

Một số nghiên cứu gần đây có liên quan đến biểu cảm và việc nhận thức chúng như nghiên cứu trẻ em bị mắc chứng rối loạn phổ tự kỷ [47, 19, 1, 4] , nghiên cứu người mắc chứng tâm thần phân liệt

(schizophrenia) [28], hay những ứng dụng trong đời sống như giám sát cảm xúc của tài xế để đảm bảo an toàn khi lái xe [26], hoặc trong lĩnh vực giáo dục như phân tích trạng thái nhận thức học tập [61]. Ngoài ra, các hệ thống tự động phát hiện và nhận diện biểu cảm khuôn mặt người còn có khả năng to lớn trong các lĩnh vực liên quan đến thương mại và quảng cáo. Khách hàng sẽ ít nhiều thể hiện biểu cảm của họ lên khuôn mặt khi quan sát và đánh giá món hàng, hay lướt qua một biển quảng cáo. Sự quan tâm của khách hàng, và các thông tin lúc đó cũng có thể hiện qua nét mặt [62]. Ngoài ra hệ thống này còn có nhiều ứng dụng trong việc quản lý và phân tích đám đông dành cho các cơ quan chức năng hay các nhà quản lý những nơi đông đúc như siêu thị, sân bay.

Tuy nhiên sự phát triển của các phương pháp nhận diện cảm xúc khuôn mặt hiện nay đang chia làm hai nhánh cùng với hai loại dữ liệu khác nhau: dữ liệu được điều khiển (lab-controlled) và dữ liệu thực tế phức tạp (in-the-wild). Các nghiên cứu gần đây dễ dàng đạt độ chính xác >90% đối với các tập dữ liệu được điều khiển như CK+ [33] hay >80% trong tập MMI [41]. Nhưng đối với các tập dữ liệu trong thực tế phức tạp thì độ chính xác lại giảm xuống < 80% như trong tập dữ liệu FER2013 [18], hay EmotiW [8], v.v. Vì thế trong nghiên cứu này tôi tập trung thực nghiệm, so sánh và đánh giá trên các tập dữ liệu thực tế phức tạp.

1.2 Phạm vi và mục tiêu

Như đã nói ở trên, điện toán tình cảm là một nhánh nghiên cứu rộng lớn, do đặc thù cảm xúc của con người được thể hiện qua nhiều mặt và sự phức tạp trên nhiều phương thức thể hiện của nó. Và do không có một quy ước chung nào, nên các tập dữ liệu, các nghiên cứu đều có những quy ước khác nhau và trong thực nghiệm đều có các cài đặt khác nhau. Dẫn đến việc rất khó có thể so sánh các phương pháp và nghiên cứu đề tài này một cách tổng quát và khách quan Nên tôi ràng buộc phạm vi và mục tiêu nghiên cứu của mình như sau.

Về lối tiếp cận, cùng với sự phát triển mạnh mẽ và những thành tựu to lớn đã gặt hái được của các kiến trúc mạng nơ-ron học sâu và các mô hình liên quan trong thời gian gần đây [31, 20, 50, 48, 30], tôi sẽ tập trung khảo sát, đánh giá, thử nghiệm và cải tiến đối với các mô hình học sâu và các phương pháp hiện đại.

Về mặt dữ liệu, các cơ sở dữ liệu biểu cảm mặt người công khai có sự khác nhau về nhiều mặt bao gồm môi trường thu thập, định dạng của đầu vào (một hoặc nhiều ảnh liên tục - chuỗi ảnh), phân phối của biểu cảm, số lượng và chất lượng của hình ảnh lẫn nhãn cho từng mẫu huấn luyện (Bảng 4.1). Trong phạm vi của đề tài luận văn này, tôi ràng buộc dữ liệu đầu vào của mô hình là dữ liệu thực tế phức tạp. Hai bộ dữ liệu tôi dùng để thực nghiệm và đánh giá so sánh được đặc tả ở Chương 4.

Về mặt đo lường và khảo thí, các nghiên cứu khác nhau thường được triển khai rất khác biệt nhau từ giai đoạn tiền xử lý, xây dựng kiến trúc mạng, cũng như là cách cài đặt quá trình huấn luyện và kiểm

thử trên các tập dữ liệu khác nhau. Mà tất cả các yếu tố đó đều có ảnh hưởng đến kết quả và hiệu suất. Do đó, việc so sánh đánh giá tác động của kiến trúc mạng và các yếu tố khác một cách cặn kẽ là không thể nếu chỉ dựa trên kết quả được báo cáo. Nên trong luận văn này, tôi sẽ hiện thực và huấn luyện lại một số mô hình mạng học sâu hiện đại để so sánh kết quả với mô hình đề xuất. Ngoài ra kết quả đạt được tôi cũng sẽ đem so sánh với các kết quả được báo cáo khoa học.

Về mục tiêu của đề tài, đối với dữ liệu thực tế phức tạp còn nhiều thử thách, trong luận văn này tôi tập trung cải thiện độ chính xác trên các tập dữ liệu này. Ngoài ra, vì còn có nhiều tranh luận về Universal Hypothesis - giả thuyết phổ quát. Ví dụ như nghiên cứu "Facial expressions of emotion are not culturally universal" [25]. Vì xuất thân là kĩ sư máy tính không có nhiều kinh nghiệm về y sinh học và tâm lý học, mặt khác không có nhiều điều kiện để ủng hộ hay phản bác các quan điểm khác nhau trong bài toán này. Tôi một mặt chấp nhận giả thuyết tồn tại 6 biểu cảm rời rạc, từ đó chấp nhận bài toán này là bài toán phân lớp. Thứ hai, tôi thực hiện làm một bộ dữ liệu về biểu cảm khuôn mặt người chứa hình ảnh người Việt Nam vì bị ảnh hưởng bởi việc làm của Giáo sư Guoying Zhao trong nghiên cứu được công bố ngày 11 tháng 6 năm 2019 [35], bà đã phát triển một bộ dữ liệu của người Trung Quốc. Từ đó dẫn đến nhiều sự phát triển và ứng dụng rộng rãi của bài toán này trong nền công nghiệp của Trung Quốc [24, 22, 23]. Bộ dữ liệu được hiện thực và trình bày trong Chương 4.

1.3 Bố cục luận văn

Luận văn này sẽ được chia thành 7 chương liền kề nhau theo thứ tự: Giới thiệu, Nghiên cứu liên quan, Cơ sở lý thuyết, Dữ liệu, Mô hình đề xuất, Thực nghiệm, và Tổng kết. Nội dung của từng chương trong luận văn này có thể tóm gọn như sau.

- Ở Chương 1, tôi sẽ giới thiệu chung về đề tài, mục tiêu, các vấn đề sẽ được giải quyết trong quá trình thực hiện luận văn.
- Tiếp đến, trong Chương 2, các nghiên cứu liên quan đến bài toán được trình bày mang tính tham khảo và dùng để so sánh với cải tiến trong luận văn.
- Chương 3 là chương tôi sẽ trình bày về các kiến thức liên quan đến phần mô hình đề xuất bao gồm kiến thức mạng học sâu và cơ chế chú ý.
- Dữ liệu được dùng để thực hiện trong xuyên suốt quá trình luận văn được trình bày ở Chương 4. Trong phần này sẽ trình bày về hai bộ dữ liệu FER2013 và bộ dữ liệu VEMO mà tôi đã tự thu thập và xây dựng.
- Trong Chương 5 tôi sẽ trình bày về mô hình Residual Masking Network là đóng góp chính của tôi trong luận văn này.

- Phần thực nghiệm bao gồm phương pháp, quá trình và các kết quả so sánh được trình bày trong

Chương 6.

- Cuối cùng, Chương 7 sẽ tóm tắt các đóng góp, ưu điểm và nhược điểm trong quá trình thực hiện luận văn. Đồng thời sẽ vạch ra một số hướng đi trong tương lai đối với kết quả hiện tại có được của luận văn.

Chương 2

Nghiên cứu liên quan

2.1 Phương pháp sử dụng Học sâu và Máy véc-tơ hỗ trợ tuyến tính

Đối với bài toán phân lớp có sử dụng các mô hình học sâu, chúng ta thường thấy chúng được sử dụng hàm Softmax [3] ở lớp cuối cùng để thực hiện bước phân lớp và cực tiểu hóa Cross Entropy Loss. Tuy nhiên trong nghiên cứu này, Yichuan Tang [54] đã dùng Linear Support Vector Machine để thay thế cho hàm kích hoạt Softmax, và hướng đến việc cho mô hình học tối thiểu hóa hàm lỗi dựa vào khoảng cách lề trong lý thuyết của Support Vector Machine.

2.1.1 Trình bày sơ lược

Nghiên cứu này của Yichuan Tang tập trung vào việc thử nghiệm so sánh mô hình học sâu trong việc sử dụng các phương pháp khác nhau để làm hàm kích hoạt, mà chính yếu là so sánh hiệu suất của hàm Softmax và lý thuyết Support Vector Machine. Việc thử nghiệm này đã đem đến một số thành tựu nhất định như là chiến thắng cuộc thi Nhận diện Biểu cảm Khuôn mặt người được tổ chức bởi Hội nghị Quốc tế về Học máy (International Conference on Machine Learning - ICML) vào năm 2013. Cuộc thi ấy được tổ chức trên Kaggle với hơn 120 đội tham gia trong thời gian đó. Ngoài ra tác giả còn thực hiện thử nghiệm so sánh trên các tập dữ liệu cổ điển như MNIST [31] hay CIFAR-10 [29].

Linear Support Vector Machine (còn được gọi là Máy Véc-tơ hỗ trợ tuyến tính) ban đầu được xây dựng cho bài toán phân lớp nhị phân - chỉ bao gồm hai lớp. Cho tập huấn luyện và nhãn tương ứng x_n, y_n, $n = 1, .., N$, $x_n \in \mathrm{R}^D$, $y_n \in \{-1, 1\}$. Việc học của thuật toán SVM bao gồm việc giải quyết bài

toán tối ưu hóa có ràng buộc như sau:

$$\min_{w,\xi} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \tag{2.1}$$

$$s.t.\ w^T x_n y_n \geq 1 - \xi_n,\ \forall n \tag{2.2}$$

$$\xi_n > 0,\ \forall n \tag{2.3}$$

Trong đó, ξ_n là biến bù (slack variable), biến này tồn tại nhằm mục đích trừng phạt những điểm dữ liệu vi phạm quy định lề của thuật toán. Kết hợp hai điều kiện của bài toán và hàm mục tiêu, ta có được hàm mục tiêu mới vô điều kiện như sau:

$$\min_w \frac{1}{2} w^T w + C \sum_{n=1} \max(1 - w^T x_n y_n, 0) \quad (2.4)$$

Bài toán bây giờ có nghĩa là lựa chọn một ma trận trọng số w nào đó để có thể tối thiểu hóa hàm mục tiêu 2.5, biểu thức này là dạng nguyên thủy của bài toán L1-SVM, với hàm lỗi *hinge loss*. Vì đặc trưng hàm này không khả vi, một biến thể khác phổ biến hơn được gọi là L2-SVM được biểu thức hóa như sau:

$$\min_w \frac{1}{2} w^T w + C \sum_{n=1} \max(1 - w^T x_n y_n, 0)^2 \quad (2.5)$$

Hàm L2-SVM là một hàm có thể khả vi. Ngoài ra, một ưu điểm khác vì L2-SVM là hàm bậc 2 nên nó trừng phạt lỗi nặng nề hơn L1-SVM - chỉ là một hàm tuyến tính, đối với các điểm dữ liệu vi phạm lề. Trong nghiên cứu còn đề cập tới phương pháp Kernel SVM, việc tối ưu hóa phương pháp này cần thực hiện ở dạng đối ngẫu. Mặc dù Kernel SVM cũng hoạt động khá tốt, nhưng việc tính toán ma trận kernel có thể tốn nhiều thời gian và bộ nhớ. Hơn nữa, việc mở rộng nó ra cho bài toán phân loại đa lớp thường không hiệu quả bằng phương pháp Multi-class SVM và phương pháp này cũng gặp phải vấn đề khi lượng dữ liệu rất lớn. Vì vậy, trong nghiên cứu của Y. Tang chỉ sử dụng linear SVMs với các mô hình học sâu bình thường.

Một lối tiếp cận cực kỳ đơn giản để chuyển bài toán phân lớp nhị phân sang bài toán yêu cầu phân lớp nhiều lớp được gọi là *one-vs-rest* được đề xuất bởi Vapnik và Vladimir [55]. Cho bài toán phân chia *K* lớp, *K* bộ phân lớp SVM sẽ được *học* độc lập với nhau, mỗi bộ phân lớp sẽ tương ứng với nhiệm vụ phân loại liệu rằng điểm dữ liệu ấy sẽ thuộc lớp *k* hay không thuộc lớp *k*, với $k \in [1, K]$.

Kí hiệu đầu ra của SVM thứ *k* như sau:

$$a_k(x) = w^T x \quad (2.6)$$

Lớp dự đoán lúc này sẽ là:

$$\text{argmax}_k \; a_k(x) \quad (2.7)$$

Mô hình được Y. Tang thử nghiệm được ghi chú là có kiến trúc cơ bản giống như được đề xuất trong các nghiên cứu của Zhong & Ghosh [64], Nagi và cộng sự [38]. Ngoài ra, Y. Tang đã thay thế hàm lỗi Hinge Loss cổ điển bằng L2-SVM, vì lý do hàm lỗi của L2-SVM là khả vi và trừng phạt lỗi rất nặng nề. Trọng

số của các lớp đầu tiên của mô hình được *học* qua việc truyền ngược đạo hàm từ lớp Support Vector Machine. Để làm được việc này, hàm mục tiêu của Support Vector Machine cần phải khả vi đối với hàm kích hoạt và lớp phía trước. Đặt Biểu thức 2.5 là $L(w)$, vào đầu vào x được thay bằng giá trị kích hoạt h của hàm liền trước, ta có đạo hàm:

$$\frac{\partial L(w)}{\partial h_n} = -C y_n w(1\{1 > w^T h_n y_n\}) \quad (2.8)$$

Trong đó $1\{\cdot\}$ là hàm đặc trưng (indicator function). Hàm đặc trưng có giá trị là 1 nếu như tham số của nó được thỏa mãn và có giá trị là 0 trong trường hợp ngược lại. Trong khi đó, đối với L2-SVM, chúng ta

có:

$$\frac{\partial L(w)}{\partial h_n} = -Cy_n w(\max(0,\ 1 - w^T h_n y_n)) \quad (2.9)$$

Từ đây, việc thực hiện giải thuật lan truyền ngược là hoàn toàn giống với các mô hình sử dụng hàm softmax. Y. Tang thấy rằng đa phần L2-SVM là tốt hơn L1-SVM và đã dùng L2-SVM trong các phần thực nghiệm.

2.1.2 Kết quả

Chiến thắng Cuộc thi Nhận diện biểu cảm khuôn mặt, năm 2013

Đây là một cuộc thi được tổ chức vào năm 2013 tại một hội thảo ở ICML, chủ trì bởi LISA của đại học Montreal. Dữ liệu huấn luyện của cuộc thi bao gồm 28,709 ảnh 48x48 rải rác khắp 7 sắc thái biểu cảm khác nhau, xem thêm ở mục mô tả tập dữ liệu này. Tập đánh giá và tập kiểm tra cùng chứa 3,589 hình ảnh và đây là một bài toán phân loại.

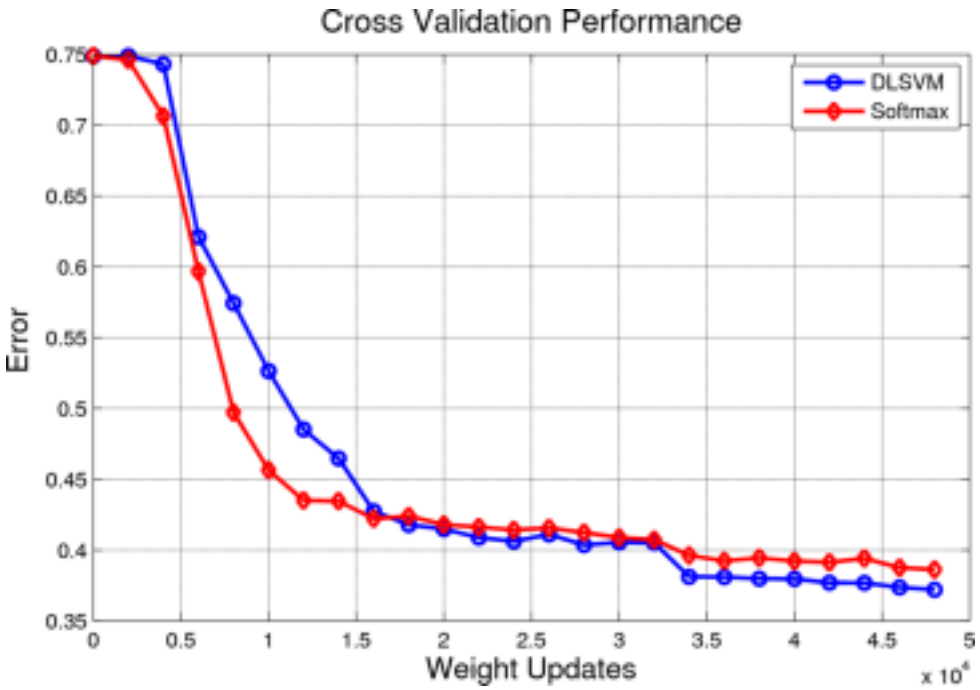
Y. Tang đã gửi nộp kết quả sử dụng mô hình của mình đạt kết quả 69.4% trên tập kiểm thử công khai (public test) và đạt được kết quả chính xác 71.2% trên tập kiểm thử chính (private test). Và kết quả này đã giành chiến thắng chung cuộc trong cuộc thi đó. Ngoài ra, do nhiễu, sai nhãn và các yếu tố khác, kết quả của con người được đo lường trên tập dữ liệu này vào khoảng giữa 65-68%.

Mô hình bao gồm một Mạng Nơ-ron tích chập đơn giản cùng với lớp SVM tuyến tính one-for-all ở cuối cùng. Giải thuật Stochastic gradient descent cùng với momemtum cũng được sử dụng trong quá trình huấn luyện. Bên cạnh đó, Y. Tang cũng lấy trung bình nhiều mô hình khác nhau để tăng khả năng tổng quát hóa. Tiền xử lý dữ liệu bao gồm: đầu tiên, trừ cho giá trị trung bình của mỗi ảnh và sau đó là cài đặt giá trị norm là 100. Mỗi pixel được chuẩn hóa bằng cách trừ cho giá trị trung bình của nó và chia giá trị đó cho độ lệch chuẩn của từng pixel. Việc chuẩn hóa được thực thi trên toàn bộ dữ liệu.

Thực nghiệm so sánh giữa Softmax và Support Vector Machine

Y.Tang đã so sánh khả năng của hai giải thuật này đối với mô hình học sâu. Cả hai mô hình đều được kiểm tra bằng giải thuật kiểm tra chéo với 8 cụm (8-fold validation), với một lớp phản chiếu ảnh (image mirroring), biến đổi ảnh tương tự, hai tầng bao gồm các bộ lọc tích chập và gộp tối đa, theo sau

là một lớp kết nối toàn bộ gồm 3,072 đơn vị đầu vào, tất cả chúng đều sử dụng hàm kích hoạt ReLU. Các tham số khác như giảm trọng số cũng được chọn khi sử dụng phương pháp kiểm tra chéo.

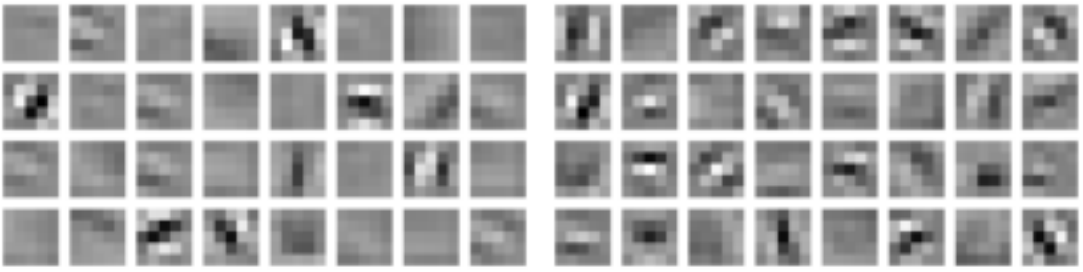


Hình 2.1: Đồ thị kết quả kiểm tra chéo của hai mô hình. Kết quả được lấy trung bình trên 8 cụm [54]

Chúng ta có thể nhìn vào đồ thị đánh giá (Hình 2.1) so sánh giữa việc sử dụng Softmax và L2-SVM như một hàm cập nhật trọng số, điểm lỗi của mô hình DLSVM giảm nhiều hơn ở nửa cuối của quá trình huấn luyện, tuy là không nhiều, nhưng rõ ràng là tốt hơn.

Y. Tang cũng phác họa các bộ lọc tích chập của hai mô hình để so sánh (xem Hình 2.2).

Ngoài ra, tác giả còn thực hiện thử nghiệm trên 2 tập dữ liệu là MNIST và CIFAR-10, nhưng nó đã vượt ra ngoài đề tài nghiên cứu nên tôi xin phép không nhắc đến ở đây.



(a) Các bộ lọc từ mạng nơ-ron tích chập với hàm softmax.

(b) Các bộ lọc từ mạng nơ-ron tích chập với hàm L2-SVM.

Hình 2.2: Các bộ lọc từ mạng nơ-ron tích

chập. 20

Bảng 2.1: So sánh hai mô hình dựa vào độ chính xác %. Điểm đánh giá trên tập huấn luyện là được trung bình trên 8 cụm kiểm tra chéo. Tập đánh giá là tập kiểm thử công khai của cuộc thi trong thời gian tổ chức. Tập kiểm tra là tập kiểm thử cuối cùng để quyết định người thắng chung cuộc.

Softmax

67.6%	
69.3%	DLSVM L2
70.1%	

Tập huấn luyện 68.9%

Tập đánh giá 69.4%

Tập kiểm tra 71.2%

2.1.3 Ưu điểm, nhược điểm và khó khăn

Ưu điểm

- Nghiên cứu này của Y. Tang là một đóng góp nền tảng cung cấp cái nhìn khách quan về việc huấn luyện mô hình học sâu.
- Lý do lớn nhất để tôi trình bày về nghiên cứu này là vì nó đã giành chiến thắng trong cuộc thi Nhận diện Biểu cảm Khuôn mặt, mà tập dữ liệu của cuộc thi ấy còn được sử dụng phổ biến đến bây giờ.

Nhược điểm và khó khăn

Tuy vậy, nghiên cứu này còn có một số hạn chế nhất định như:

- Thiếu đi sự phân tích kỹ lưỡng vào các đặc trưng của biểu cảm khuôn mặt người. Không có đề cập đến lý thuyết và cách thiết kế các lớp trích xuất đặc trưng sao cho phù hợp đối với bài toán.
- Nghiên cứu đa phần tập trung vào việc so sánh giữa việc sử dụng các cách cài đặt mà thiếu đi sự tập trung vào bài toán Nhận diện Biểu cảm.
- Việc trình bày về mô hình được sử dụng trong nghiên cứu vô cùng sơ khai, thiếu đi các chi tiết cụ thể về việc cài đặt các thông số.

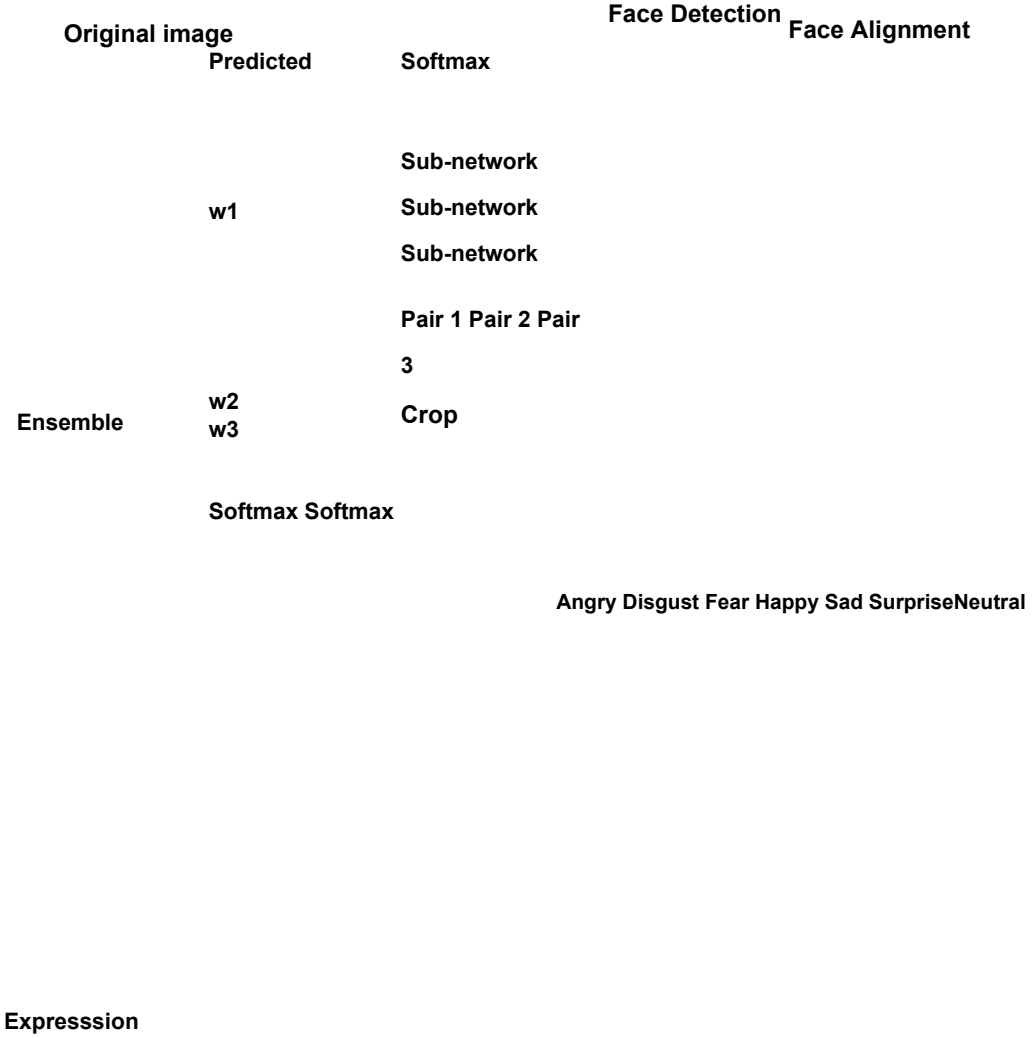
Một tranh luận khác [49]

Trong bài viết này các nhà nghiên cứu cho rằng sự khác nhau của Softmax và SVM là nhỏ và tùy theo quan điểm mỗi người. Nhưng điểm quan trong họ nhấn mạnh là Softmax không bao giờ thỏa mãn với kết quả, trong khi SVM chỉ cần thỏa điều kiện lề thì hàm mất mát sẽ bằng 0. Vì thế, trong luận văn này, tôi trình bày phương pháp này chỉ để mang tính tham khảo phương pháp đã chiến thắng trong cuộc thi dùng tập dữ liệu FER2013 mà tôi sẽ dùng thực nghiệm chủ yếu.

2.2 Phương pháp sử dụng chồng chất mạng tích chập đa khu vực

Y Fan và cộng sự [16] đã đề xuất một kiến trúc mạng có tên là Multi-Region Ensemble CNN (MRE-CNN) cho bài toán nhận diện biểu cảm khuôn mặt dựa vào phương pháp học kết hợp. Tổng quan về kiến

trúc của mô hình MRE-CNN được thể hiện qua Hình 2.3.



Hình 2.3: Tổng quan về lối tiếp cận của tác giả: Multi-Region Ensemble CNN (MRE-CNN) [16]

ig. 1. An overview of our approach: Multi-Region Ensemble CNN (MRE-CNN) fra2.2.1 Trình bày sơ lược

CNN approaches topped three slots in the 2014 ImageNet challenge f

Mô hình Multi-Region Ensemble Convolution Neural Network được tác giả tiến hành huấn luyện và đánh giá trên hai tập dữ liệu là AFEW 7.0 và RAF-DB.

ognition task, with the VGGNet [11] architecture achieving a remarkably Các điểm dự đoán của từng mạng con được tích hợp lại với nhau để sinh ra kết quả nhận diện cuối

e. With a review of previous CNNs, AlexNet [5] demonstrated the effecti

cùng. Bằng lối tiếp cận này, tác giả đã phân tích sự ảnh hưởng của các vùng cục bộ khác nhau trong

N by introducing convolutional layers followed by Max-pooling layers a

khuôn mặt đối với việc nhận diện biểu cảm

d Linear Units (ReLUs). AlexNet significantly outperformed the runner-

Tiền xử lý dữ liệu

-5 error rate of 15.3% in the 2012 ImageNet challenge. In our proposed fra22

e of the network structures is based on AlexNet and the other one VGG-16 istwork based on VGGNet [11].

The goal of automatic FER is to classify faces in static images or dynam

frames that have clear faces with an adaptive frame interval. To extract and align facboth from original images in RAF-DB and frames of videos in AFEW 7.0, we useC++ library, Dlib[3] face detector to locate the 68 facial landmarks. As shown in Fig3, based on the coordinates of localized landmarks, aligned and cropped whole-regi

Với một ảnh đầu vào, hệ thống sẽ thực hiện bước phát hiện khuôn mặt và cân chỉnh chúng trước,

and sub-regions of the face image can be generated in a uniform template with a affi

sau đó thực hiện làm giàu dữ liệu từ những ảnh này. Từ các bức ảnh trong tập dữ liệu RAF-DB, tác giả transformation. In this stage, we align and crop regions of the left eye, regions of t

sử dụng thư viện Dlib để định vị 68 vị trí mốc trên khuôn mặt. Từ tọa độ của của các dấu mốc ấy mà

nose, regions of the mouth, as well as the whole face. Then three pairs of images are

tiến hành bước cân chỉnh khuôn mặt. Bước làm giàu dữ liệu, Y Fan tiến hành cắt ảnh toàn bộ vùng mặt cũng như các vùng quan trọng khác như mắt, mũi, miệng. Sinh ra ba cặp ảnh và tất cả được điều chỉnh

resized into 224×224 pixels.

kích thước về thành 224 × 224. Quá trình được thể hiện qua hình 2.4. Bước làm giàu dữ liệu được thực

hiện cả ngoại tuyến lẫn trực tuyến. Số lượng mẫu huấn luyện sau khi làm giàu theo như báo cáo đã được tăng 15 lần.

Pair 1

Pair 2

Pair 3
Face image Face landmarks

Whole-region Sub-region

Hình 2.4: Bước tiền xử lý cắt ra các vùng mặt [16]

Nhận diện cảm xúc khuôn mặt bằng chồng chất mạng tích chập đa khu vực

Fig. 3. The processing of the cropped whole-region and sub-regions of the facial image. Phương pháp này được ông abc công bố trong nghiên cứu Multi-Region Ensemble Convolutional

Neural Network Tác giả sử dụng ba vùng đáng để ý trên khuôn mặt là mắt trái, mũi và miệng. Mỗi

3.2 Multi-Region Ensemble Convolutional Neural Network

vùng này sẽ đi kèm với ảnh toàn bộ khuôn mặt, tạo thành một cặp đầu vào cho mỗi mạng con trong MRE-CNN. Sau đó, dựa trên tổng điểm phân loại từ ba mạng con này, chúng ta sẽ có được dự đoán cuối Our framework is illustrated in Figure 1. We take three significant sub-regions of tcùng.

human face into account: the left-eye, the nose and the mouth. Each particular suMạng con trong MRE-CNN

region will be accompanied by its corresponding whole facial image, forming a douTheo Hình 2.5, Tác giả đã sử dụng 13 lớp convolutional và năm lớp max-pooling và nối các đầu ra

1 http://www.whdeng.cn/RAF/model1.html 2

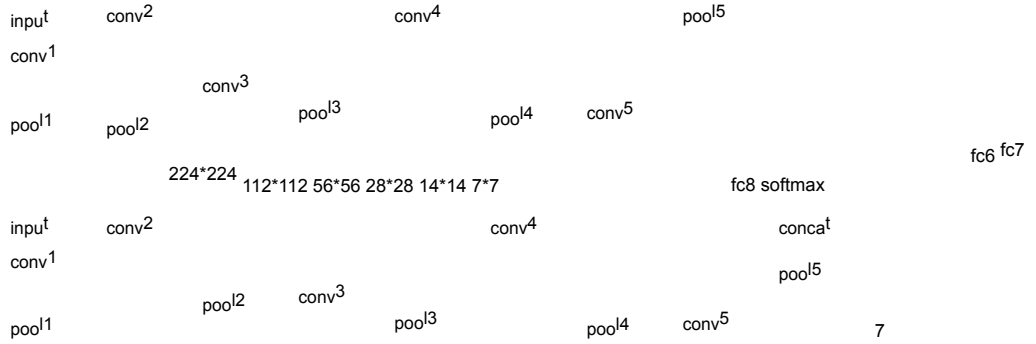
https://sites.google.com/site/emotiwchallenge/ 3 dlib.net

ở lớp pool5 trước khi đi qua các lớp fully-connected các bước cuối. Lớp softmax cuối cùng dùng để cho điểm số nhận dạng của mô hình. Kiến trúc mạng này là bắt nguồn từ mạng VGG-16 và để tăng độ tổng quát của mô hình, tác giả đã fine-tune mô hình đã được pre-trained để không phải huấn luyện lại toàn bộ mô hình cho tập dữ liệu này.

Để đánh giá kiến trúc đề xuất, tác giả cũng đã sử dụng kiến trúc AlexNet và không sử dụng bất kỳ mô hình nào đã được pre-trained trong quá trình huấn luyện. Đối với kiến trúc AlexNet, tác giả sử dụng

23

3.3 The Sub-networks in MRE-CNN Framework



trong MRE-CNN [16]

224*224 112*112 56*56 28*28 14*14 7*7 Hình 2.5: 4096 4096

Kiến trúc VGG-16 cho các sub-network

Fig. 4. The VGG-16 sub-network architecture in MRE-CNN framework.

năm lớp convolutional và ba lớp max-pooling, nhưng khác với kiến trúc gốc ở điểm chỉ thay đổi ở lớp cuối cùng từ một nghìn về còn bảy đầu ra.

As Figure 4 shows, we adopt 13 convolutional layers and 5 max pooling layer

Cuối cùng, tác giả của ba dự đoán từ ba mạng con bằng việc thực hiện phép toán tổng trọng số. concatenate the outputs from two pool5 layers before going through the first fullynected layer. The final softmax layer gives the prediction scores. When empl

2.2.2 Kết quả

VGG-16 [11], we fine-tune the pre-trained model with the training set of AFEand RAF-DB, respectively, in the following experiments.

Trong bước làm giàu dữ liệu, các kỹ thuật về lật ảnh, xoay, thêm nhiễm sử dụng Gaussian Noise được áp dụng. Các thông số cài đặt về quá trình huấn luyện của các mạng con VGG-16 cũng như AlexNet có thể xem thêm tại [16]. Khi thực hiện thí nghiệm trên tập RAF-DB, RAF-DB được chia thành dữ liệu huấn luyện và dữ liệu test dựa trên kỹ thuật kiểm tra chéo 5 cụm. Hơn nữa, tác giả còn đánh giá kết quả trên từng mạng con riêng biệt. Kết quả được thể hiện qua Bảng 2.2. Ta có thể thấy kết quả được đánh giá khi sử dụng kiến trúc VGG-16 là hiệu quả hơn so với sử dụng kiến trúc AlexNet.

Bảng 2.2: Kết quả trên tập RAF-DB [16]

Kiến trúc	Độ chính xác
Face+LeftEye (Single VGG-16 sub-network)	76.5%
Face+Nose (Single VGG-16 sub-network)	75.64%
MRE-CNN(VGG-16)	76.73%
MRE-CNN(AlexNet)	74.78%

2.2.3 Ưu điểm và nhược điểm

Y Fan và đồng nghiệp [16] đã đề xuất một kiến trúc mạng học kết hợp được các đặc trưng toàn cục và đặc trưng cục bộ trên các vùng khuôn mặt. Sử dụng kỹ thuật tổng trọng số để giúp tăng được độ chính

xác của cả mô hình. Tận dụng được những mô hình đã được huấn luyện trước để không phải huấn luyện lại từ đầu toàn bộ tham số của mô hình. Nhưng bên cạnh đó mô hình của Y Fan cũng có một số nhược điểm là việc sử dụng mô hình VGG-16 đã cũ. Thêm nữa, phần cân hỉnh khuôn mặt cũng đang bị phụ thuộc vào một công cụ khác là dlib, nếu như bước lấy các điểm mốc trên khuôn mặt thất bại thì sẽ dễ dàng dẫn tới phương pháp này cũng sẽ thất bại. Việc này không phù hợp các dữ liệu trong thực

tế phức tạp.

2.3 Phương pháp học kết hợp mạng tích chập đa mức.

2.3.1 Trình bày sơ lược

Hai-Duong Nguyen và cộng sự [40] đã thực hiện một nghiên cứu sử dụng mạng nơ-ron đa mức, thực nghiệm trên tập dữ liệu FER2013 . Kết quả nghiên cứu được đánh giá trên tập dữ liệu FER2013 với độ chính xác và những cải thiện đáng kể so với các kiến trúc trước đây.

Kiến trúc thẳng - Plain Networks

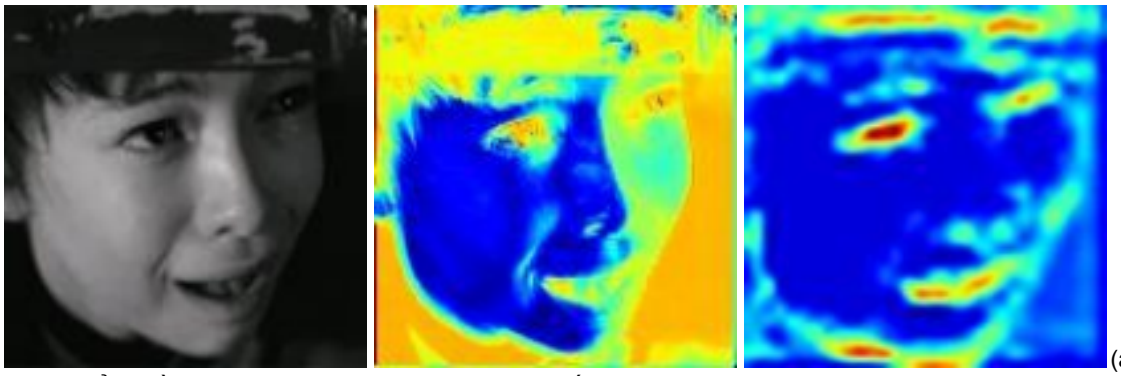
Kiến trúc thẳng được Hai-Duong Nguyen đề xuất được thiết kế bằng các lớp nơ-ron tích chập được xếp chồng nhau rất thông dụng trong các bài toán phân loại ảnh, được lấy cảm hứng từ mạng VGG [48], với 18 lớp được tổ chức thành 5 khối (xem Hình 2.6). Mỗi khối chứa các lớp tích chập và lớp thăm dò theo sau. Mô hình nhận đầu vào là ảnh thể xám có kích thước 48 × 48 và cuối mạng là lớp softmax bảy đầu ra để phân lớp ra bảy biểu cảm tương ứng.

a) Multi-level network for image-based facial expression recognition

Hình 2.6: Mạng nơ-ron đa mức nhận diện biểu cảm khuôn mặt. [40]. Kiến trúc này sử dụng các đặc trưng cấp trung và cấp cao trong việc phân lớp. Các thông tin đặc trưng này được trích xuất thông qua các khối nơ-ron giữa đến cuối mạng.

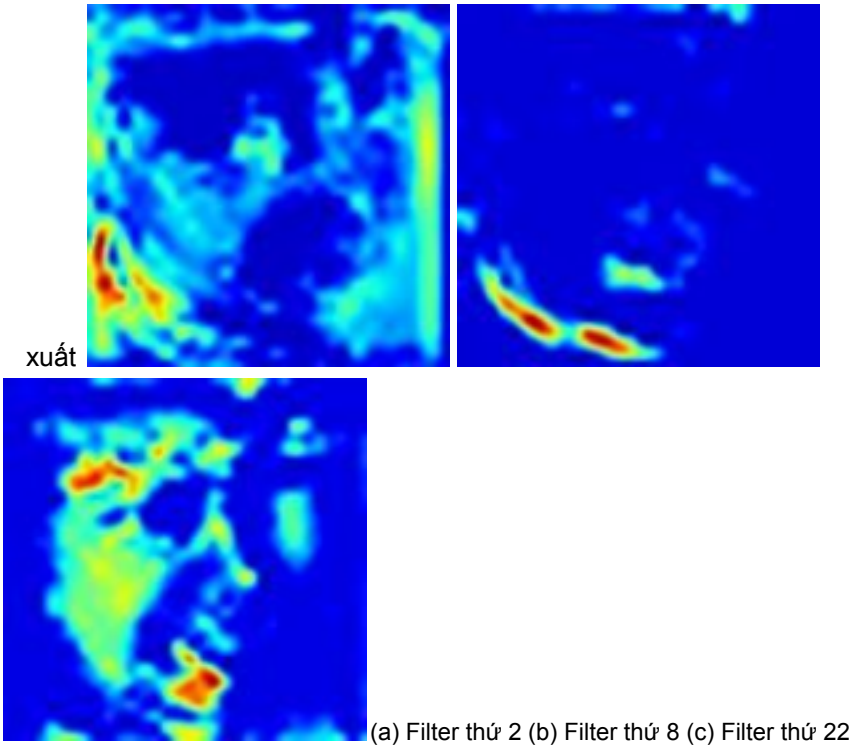
Hai-Duong Nguyen và cộng sự đã sử dụng kỹ thuật Gradient-weighted Class Activation Mapping (Grad-CAM)[43] để mô phỏng những vùng chứa thông tin hữu ích cho việc phân loại. Mặt khác, phương pháp này cũng giải thích cách một lớp nơ-ron tích chập đặt sự chú ý vào từng vị trí đặc biệt của đối tượng ở các lớp cụ thể trong mạng. Hình 2.7 thể hiện một Grad-CAM được sinh ra bởi bộ lọc thứ 24 trong khối thứ 2 của mạng. Tại mức này, mạng chỉ tập trung vào mắt, mũi, những bộ phận chiếm vai trò

quan trọng trong bài toán nhận diện cảm xúc. Trái lại, tại khối thứ nhất, mạng chỉ chú ý đến phông nền và những vùng thông tin vô ích khác. Bên cạnh đó, Hình 2.8 thể hiện mô phỏng Grad-CAM được sinh ra từ khối thứ ba của mạng. Mặt dù những bộ lọc đó là rất đáng giá để tạo nên các đặc trưng ở mức cao hơn nhưng cũng có thể thấy rằng không phải tất cả chúng đều đóng góp vào quá trình nhận diện. Chỉ có một vài bộ lọc ở mức giữa của mạng là có đóng góp cho quá trình nhận diện và phân loại. Từ quan điểm này, tác giả đã đề xuất mạng nơ-ron tích chập nhiều mức cho bài toán này.



(a) Ảnh đầu vào (b) Grad-CAM sau block thứ nhất (c) Grad-CAM sau block thứ ba

Hình 2.7: Một ví dụ về ảnh đầu vào và ảnh trực quan của Grad-CAM từ mạng 18 lớp được đề xuất



(a) Filter thứ 2 (b) Filter thứ 8 (c) Filter thứ 22

Hình 2.8: Các bộ lọc đơn giản, mức độ chú ý tăng dần từ màu xanh sang màu đỏ

2.3.2 Kết quả được báo cáo

Mô hình mà tác giả đề xuất được đánh giá trên tập dữ liệu FER2013. Chi tiết về tập dữ liệu này đã trình bày ở mục giới thiệu các tập dữ liệu. Kết quả được báo cáo được trình bày ở phần thực nghiệm.

2.3.3 Ưu điểm và nhược điểm

Trong nghiên cứu của Nguyen Hai-Duong và cộng sự [40], anh đã đề xuất các kiến trúc để giải quyết cho bài toán nhận diện cảm xúc trong trường hợp lấy mẫu bất kỳ, cụ thể là trên tập dữ liệu FER2013.

Ưu điểm của lối tiếp cận này là tác giả đã tận dụng được kiến trúc mạng có sẵn là VGG-Net để lấy

cảm hứng cho mô hình mạng và tác giả đã sử dụng những đặc trưng ở mức giữa để cùng đóng góp vào quá trình nhận diện nhằm cải thiện độ chính xác của mô hình nhận diện, sử dụng phương pháp GradCAM để trực quan hóa - thứ mà trong nghiên cứu này tôi sẽ sử dụng.

Nhược điểm của nghiên cứu này là thật khó khăn khi mà tác giả sử dụng một mạng nơ-ron cơ bản để so sánh với mạng nơ-ron đa mức chứ không hiện thực lại các mô hình hiện đại dưới cùng cấu hình, từ đó mà có thể so sánh sức mạnh của mạng nơ-ron đa mức và các mạng nơ-ron hiện đại một cách khách quan hơn. Từ đây mà trong nghiên cứu này, tôi sẽ hiện thực và huấn luyện lại các kiến trúc hiện đại, từ đó so sánh với kiến trúc đề xuất của tôi dưới cùng một cấu hình và điều kiện huấn luyện.

Chương 3

Cơ sở lý thuyết

3.1 Cảm xúc con người thông qua biểu thị nét mặt.

3.1.1 Biểu thị cảm xúc trên khuôn mặt người

Khuôn mặt con người là một thực thể phức tạp đóng vai trò trọng tâm để phân tích trong bài toán này. Khuôn mặt không chỉ là nơi phản ánh cảm xúc mà còn thể hiện cả yếu tố tinh thần của con người, những tương tác xã hội và cả các tín hiệu sinh lý học [19]. Trong đề tài này, chúng tôi chỉ đề cập đến nội dung quan trọng nhất mà khuôn mặt biểu lộ, đó chính là cảm xúc của con người. Cấu trúc và sự kết hợp giữa các cơ làm cho khuôn mặt trở thành thành phần chứa đựng nhiều cảm xúc nhất của con người. Chúng ta không thể biết được trạng thái của một người nào đó nếu như không nhìn vào khuôn mặt của họ. Giáo sư Albert Mehrabian của đại học California, Los Angeles đã kết luận trong nghiên cứu của ông về ba yếu tố cơ bản trong các cuộc giao tiếp là:

- Từ ngữ dùng trong giao tiếp.
- Cách truyền đạt và giọng điệu khi giao tiếp.
- Các hành vi phi ngôn ngữ khi giao tiếp.

Trong đó các hành vi phi ngôn ngữ chiếm đếm 55% hiệu quả của cuộc nói chuyện [15]. Các hành vi phi ngôn ngữ bao gồm: cử chỉ, nét mặt, tư thế và các chuyển động khác nhau của cơ thể. Trong các thành phần ấy, nét mặt hay còn gọi là biểu cảm trên khuôn mặt là một thành phần không thể thiếu góp phần đến sự hiệu quả của việc tương tác. Khuôn mặt của chúng ta là một thành phần phức tạp và có những khác biệt lớn so với những bộ phận còn lại của cơ thể. Cụ thể, đó là một trong số những hệ thống tín hiệu phức tạp có sẵn của chúng ta, bao gồm khoảng trên 40 cơ tự trị về cấu trúc và chức năng, mỗi cơ có thể được kích hoạt độc lập với những cơ còn lại [15]. Hệ thống cơ mặt là nơi duy nhất trên cơ thể chúng ta

mà các cơ, hoặc là gắn liền với xương và mô mặt (các cơ khác trên cơ thể người kết nối đến hai xương), hoặc là chỉ gắn với mô mặt như cơ bao quanh mắt hoặc môi. Rõ ràng, hoạt động của cơ mặt là biểu lộ cảm xúc - nó cho phép chúng ta chia sẻ thông tin xã hội với người khác thông qua giao tiếp bằng ngôn ngữ và giao tiếp bằng phi ngôn ngữ. Tất cả các cơ trên cơ thể của chúng ta được điều khiển bởi các dây thần kinh, giúp xác định đường đi vào não và tủy sống. Các dây thần kinh có kết nối hai chiều, có nghĩa là chúng có thể kích hoạt các cơ dựa trên sóng não, đồng thời truyền thông tin về trạng thái cơ hiện tại trở lại về não. Các dây thần kinh mặt xuất hiện từ sâu bên trong thân não đặt ở hộp sọ bên dưới tai, và phân nhánh đến tất cả các cơ như một cái cây. Hơn nữa, các dây thần kinh mặt cũng kết nối với những vùng vận động trẻ hơn (younger motor regions) trong não trước (neo-cortex) của chúng ta (não trước là vùng chỉ xuất hiện trong bộ não của động vật có vú), những vùng mà chủ yếu chịu trách nhiệm cho các chuyển động cơ mặt cần thiết để nói chuyện [15]. Thân não và vỏ não cực kỳ nhạy, phụ thuộc vào việc một biểu cảm trên khuôn mặt là có chủ ý hay không

chủ ý. Trong khi thân não kiểm soát các biểu cảm không có chủ ý và vô thức xảy ra một cách tự nhiên, thì vỏ não bao gồm các biểu cảm trên khuôn mặt có ý thức kiểm soát và có chủ ý. Đó là lí do vì sao một gương mặt với một nụ cười giả tạo không xuất hiện một cách tự nhiên, trong khi một nụ cười thành thật rất dễ phát sinh. Hay nói cách khác, việc tạo ra một nụ cười giả thậm chí không cảm thấy như ý muốn, vì rõ ràng nó không kích hoạt những dây thần kinh giống như khi chúng ta có một cụ cười đích thực.

Đến bây giờ chúng ta đã biết được rằng các dây thần kinh mặt kết nối phần lớn các cơ mặt với não. Vậy dây thần kinh mặt có liên quan đến cảm xúc và các hành vi cảm xúc như thế nào? Thực tế, các vùng giống nhau trong thân não (kích hoạt các biểu cảm khuôn mặt của chúng ta) kiểm soát quá trình xử lý và điều tiết cảm xúc. Những nghiên cứu trong hình ảnh thu được từ máy quét cộng hưởng từ (Magnetic Resonance Imaging - MRI) đã nhận diện được một khu vực cụ thể trong thân não rất nhạy cảm khi đối mặt với các mối đe dọa tiềm ẩn về thị giác và thính giác (ví dụ như một bóng lờ mờ xuất hiện trong đêm tối hay một tiếng hén lớn phát ra). Đó chính là hạch hạnh nhân (amygdala) trái và phải của não bộ. Thông thường, hạch hạnh nhân được liên kết với quá trình xử lý của các sự kiện đáng sợ, đe dọa trực tiếp đến chúng ta, hoặc những kích thích vui sướng phấn chấn của cơ thể. Bên cạnh việc xử lý sự sợ hãi và niềm vui, hạch hạnh nhân cũng đã được tìm thấy trong việc chịu chách nhiệm chung cho các chức năng tự trị liên quan đến kích thích cảm xúc. Hạch hạnh nhân điều khiển quá trình phóng thích của cortisol và các hormone gây căng thẳng khác vào máu, kiểm soát nhịp tim cũng như các hành vi có thể quan sát được như những thay đổi trong tư thế và các biểu cảm khuôn mặt [15].

Theo như giả thuyết phổ quát (universality hypothesis) tuyên bố rằng: việc nhận thức và biểu cảm cảm xúc trên khuôn mặt là giống hệt nhau bất kể nguồn gốc xuất thân và nền văn hóa của con người. Trong công trình đầu tiên nghiên cứu về các biểu cảm nét mặt và tầm quan trọng của chúng (được thực hiện bởi Charles Darwin vào năm 1872 [7]), ông đã tuyên bố rằng biểu cảm khuôn mặt là bẩm sinh, nghĩa là không cần học và có ý nghĩa tiến hóa để sinh tồn. Sau đó, Paul Ekman đã thực hiện một nghiên cứu để tìm hiểu xem con người, trên toàn thế giới, có biểu hiện cảm xúc trên khuôn mặt tương tự nhau hay không, và ông đã phát hiện ra được biểu hiện cảm xúc trên khuôn mặt người có tính phổ quát ở một

mức nhất định. Nhờ vào việc ông đã nghiên cứu một bộ tộc bị cách ly ở New Guinea và quan sát những biểu cảm trên nét mặt của họ, nhận ra rằng những biểu cảm đó cũng tương tự như thế giới văn minh xung quanh họ. Ngoài ra, khi ông cho họ xem những biểu cảm cảm xúc trên các bức ảnh của con người ở thế giới văn minh, những người ở bộ tộc này cũng có khả năng nhận diện được các cảm xúc đó [13]. Sau đó, Paul Ekman đã nghiên cứu rõ hơn và đưa ra kết luận rằng có một tập hợp các loại cảm xúc luôn được biểu đạt bằng cùng một kiểu nét mặt giống nhau, bất kể giới tính, tuổi tác, nền văn hóa hay lịch sử xã hội [10].

3.1.2 Hệ thống mã hóa cơ mặt

Hệ thống Mã hóa Hoạt động Cơ mặt (Facial Action Coding System - FACS) được phát triển bởi Ekman và Friesen[14] là hệ thống mô tả đầy đủ những hành vi trên gương mặt con người bằng một

tập hợp các đơn vị hoạt động được định nghĩa (Action Unit - AUs). Hệ thống làm việc dựa trên việc phát hiện một hoặc nhiều chuyển động cơ mặt gọi là AUs, sau đó tham khảo đến một mục lục để chỉ ra loại cảm xúc nào đang được biểu hiện. Có 46 đơn vị hoạt động khác nhau được mô tả trong [14]. Từ đó, Ekman và Rosenberg mô tả các biểu cảm khuôn mặt từ tổ hợp của các đơn vị hoạt động ấy.

Bảng 3.1: Bảng tổ hợp các đơn vị hoạt động được phân theo cảm xúc [14]

Cảm xúc	Đơn vị hoạt động (Action Units)	Mô tả
Hạnh phúc	6 + 12	Cheek Raiser, Lip Corner Puller.
Buồn bã	1 + 4 + 15	Inner Brow Raiser, Brow Lowerer, Lip Corner Depressor.
Bất ngờ	1 + 2 + 5 + 26	Inner Brow Raiser, Outer Brow Raiser, Upper Lid Raiser, Jaw Drop.
Sợ hãi	1 + 2 + 4 + 5 + 7 + 20 + 26	Inner Brow Raiser, Outer Brow Raiser, Brow Lowerer, Upper Lid Raiser, Lid Tightener, Lip Stretcher, Jaw Drop.
Giận dữ	4 + 5 + 7 + 23	Brow Lowerer, Upper Lid Raiser, Lid Tight ener, Lip Tightener.
Khinh bỉ	9 + 15 + 16	Nose Wrinkler, Lip Corner Depressor, Lower Lip Depressor.

Ngoài những cảm xúc cơ bản còn có khái niệm về vi cảm xúc, nó được định nghĩa là các biểu cảm trên khuôn mặt diễn ra rất nhanh và không có chủ ý [12] tức là khi một biểu cảm xảy ra sẽ không chịu bất cứ sự chi phối nào về những ý thức có chủ đích của con người. Các vi cảm xúc có khuynh hướng xuất hiện trong những tình huống căng thẳng, đặc biệt khi thứ gì đó quý giá mà chúng ta có được nó hoặc

làm mất nó đi, và các vi cảm xúc xảy ra khi che giấu cảm xúc có ý thức hoặc vô ý thức. Vì thế, các vi cảm xúc là một gợi ý để ta có thể phát hiện những kẻ nói dối, việc này có thể rất quan trọng trong điều tra tội phạm, an ninh sân bay hoặc kiểm tra tâm lý vì những vi cảm xúc này phát sinh một cách vô thức mà con người không thể điều khiển được, từ đó có thể phát hiện được cảm xúc thật mà đối phương đang cố che giấu.

Qua phần này, ta có thể biết được rằng cảm xúc xuất phát từ đâu và được điều khiển bởi bộ phận thần kinh nào. Chúng ta cũng có được cái nhìn toàn diện hơn về các biểu cảm cảm xúc trên khuôn mặt người. Khuôn mặt con người là một trong những phần quan trọng nhất trong việc phân tích biểu cảm của con người.

3.2 Mạng nơ-ron tích chập trong bài toán nhận ảnh

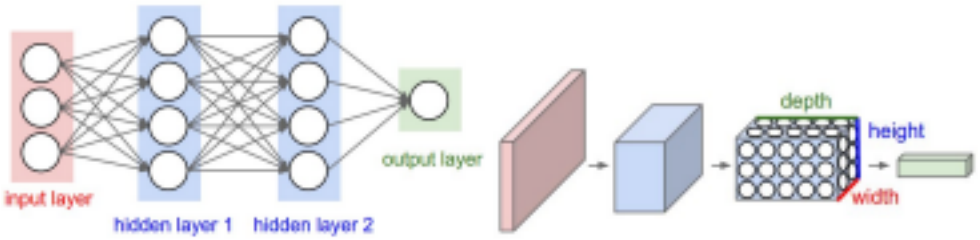
Mạng nơ-ron tích chập (CNNs/ConvNets) rất giống với mạng nơ-ron thông thường, chúng được tạo thành từ các nơ-ron có trọng số và độ chệch mà có thể học được. Mỗi nơ-ron nhận được một số đầu vào, thực hiện một phép tích vô hướng và tùy ý theo sau nó với một lớp phi tuyến tính. Toàn bộ mạng vẫn biểu thị một hàm khả vi duy nhất: từ các pixel hình ảnh thô đến một đầu ra mong muốn. Và chúng

vẫn có hàm mất mát (như SVM / Softmax) ở lớp cuối cùng (lớp kết nối đầy đủ) và tất cả các kỹ thuật được sử dụng để học Mạng thần kinh thông thường vẫn được áp dụng.

Vậy điều gì thay đổi? Kiến trúc Mạng nơ-ron tích chập đưa ra giả định rằng đầu vào là hình ảnh, cho phép chúng ta mã hóa các thuộc tính nhất định vào kiến trúc này. Những điều này sau đó làm cho mô hình được hiện thực hiệu quả hơn và đồng thời giảm đáng kể số lượng tham số trong mạng.

3.2.1 Tổng quan về mạng nơ-ron tích chập

Mạng nơ ron tích chập [6] lợi dụng thực tế là đầu vào bao gồm các hình ảnh và những ràng buộc của chúng để thiết kế ra kiến trúc theo một cách hợp lý hơn. Cụ thể là, không giống như một Mạng nơ-ron thông thường, các lớp của một ConvNet có tế bào thần kinh được sắp xếp theo 3 chiều: chiều rộng, chiều cao, chiều sâu. Chiều sâu ở đây đề cập đến chiều thứ ba của khối đặc trưng. Ví dụ hình ảnh đầu vào trong tập dữ liệu CIFAR-10 là một khối đầu vào có kích thước 32x32x3 (tương ứng với chiều rộng, chiều cao và chiều sâu). Trong mạng nơ-ron tích chập, các tế bào thần kinh trong một lớp sẽ chỉ được kết nối với một vùng nhỏ của lớp trước đó, thay vì tất cả các nơ-ron như cách của lớp kết nối đầy đủ. Hơn nữa, lớp đầu ra cuối cùng sẽ có kích thước 1x1x10, bởi vì đến cuối kiến trúc ConvNet, chúng ta cần một vectơ điểm số lớp, được sắp xếp theo chiều sâu. Hình 3.1 là một ví dụ



(a) Mạng nơ-ron 3 lớp (b) Một mạng nơ-ron tích chập sắp xếp các nơ-ron của nó theo 3 chiều.

Hình 3.1: So sánh giữa mạng nơ-ron thông thường và mạng nơ-ron tích chập. Một mạng ConvNet được tạo nên từ các lớp. Tất cả các lớp có một API đơn giản: Nó biến đổi một khối đầu vào 3 chiều thành một khối đầu ra 3 chiều bằng một hàm khả vi có thể có hoặc không có tham số.

3.2.2 Các lớp cơ bản trong Mạng Nơ-ron Tích chập.

Như được đặc tả từ trước, một mạng ConvNets [6] là một chuỗi của các lớp và mỗi lớp của ConvNet biến đổi một khối đặc trưng sang một khối đặc trưng khác thông qua một hàm khả vi. Có ba loại lớp chính để tạo nên một kiến trúc ConvNets: Lớp Tích chập (Convolutional Layer), Lớp Gộp (Pooling Layer) và một Lớp Kết nối Đầy đủ (Fully-Connected Layer). Chồng chất các lớp này chúng ta sẽ nhận được một kiến trúc ConvNet.

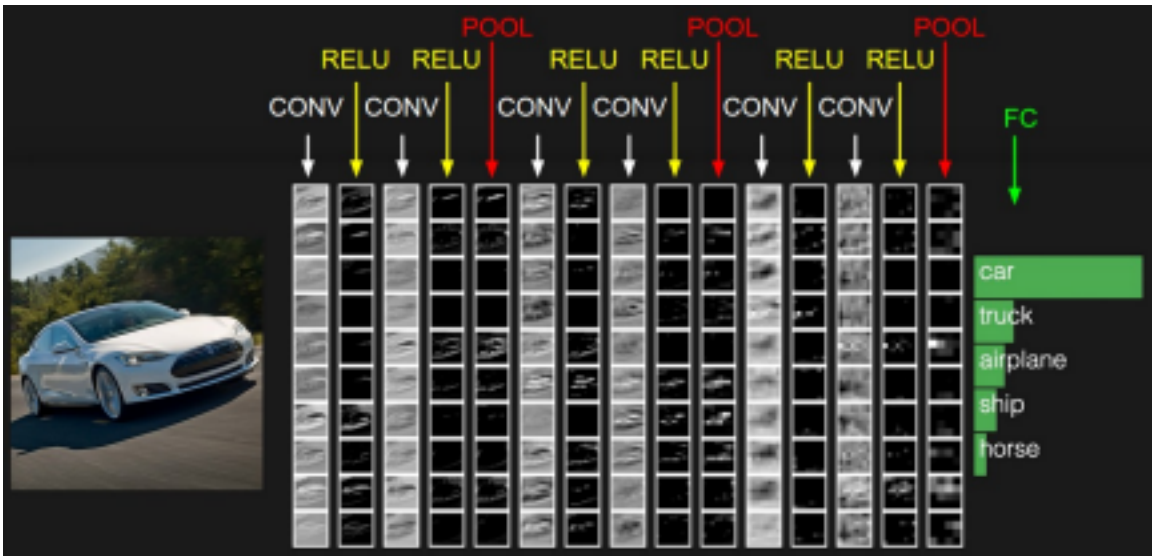
Để hiểu hơn về kiểu kiến trúc này, ta có thể xem một qua một kiến trúc mẫu đơn giản nhưng đầy

đủ [INPUT - CONV - RELU - POOL - FC], chi tiết như sau:

- INPUT [32x32x3] sẽ giữ các giá trị pixel thô của hình ảnh, trong trường hợp này là hình ảnh có chiều rộng 32, chiều cao 32 và với ba kênh màu R, G, B.
- Lớp CONV sẽ tính toán đầu ra của các nơ-ron được kết nối với các vùng cục bộ trong đầu vào, mỗi nơ-ron thực hiện một phép tích vô hướng giữa trọng số của chúng và một vùng nhỏ mà chúng được kết nối trong khối đầu vào. Điều này có thể dẫn đến khối đầu ra có kích cỡ [32x32x12] với giả định ta dùng 12 bộ lọc.
- Lớp RELU sẽ áp dụng chức năng kích hoạt cho từng phần tử, chẳng hạn như đặt ngưỡng $max(0, x)$. Điều này không thay đổi kích cỡ của khối đặc trưng ([32x32x12]).
- Lớp POOL sẽ thực hiện thao tác lấy mẫu dọc theo chiều không gian (chiều rộng, chiều cao), cho ra khối đặc trưng có kích thước [16x16x12].
- Lớp FC (tức là được kết nối đầy đủ) sẽ tính toán điểm số của lớp, cho đầu ra có kích thước [1x1x10], trong đó mỗi số trong 10 số tương ứng với điểm số của lớp, chẳng hạn như trong số 10 loại của tập dữ liệu CIFAR-10. Như với Mạng nơ-ron thông thường và như tên gọi của nó, mỗi nơ-ron trong lớp này sẽ được kết nối với tất cả các số trong tập trước.

Theo cách này, ConvNets biến đổi lớp hình ảnh đầu vào theo từng lớp từ các giá trị pixel ban đầu thành điểm số của lớp cuối cùng. Lưu ý rằng một số lớp chứa tham số và và một số lớp khác thì không. Cụ thể, các lớp CONV / FC thực hiện các phép biến đổi bằng cách ngoài kích hoạt khối đặc trưng đầu vào, mà còn có sự tham gia của các tham số (trọng lượng và độ chệch của các nơ-ron) trong các phép biến đổi. Mặt khác, các lớp RELU / POOL sẽ thực hiện một chức năng cố định. Các tham số trong các lớp CONV / FC sẽ được đào tạo với giải thuật Gradient Descent để tiến tới mục đích giảm lỗi được tính bằng hàm mất mát, để phục vụ mục tiêu cuối cùng là đầu ra dự đoán của mô hình trùng khớp nhất với thực tế được gắn nhãn.

Nói ngắn gọn, một mạng ConvNet à một chồng chất các Lớp nhằm biến đổi một khối ảnh sang thành một khối đầu ra. Mạng này chứa một số lớp cơ bản như CONV/RELU/FC/POOL/... Mỗi lớp này nhận một đầu vào ba chiều và trả về đầu ra ba chiều bằng một hàm khả vi. Mặt khác, mỗi lớp có thể có tham số hoặc không.



Hình 3.2: Các đặc trưng mẫu của một kiến trúc ConvNets. Khối đầu vào chứa các pixel thô (trái) và khối đầu ra chứa điểm từng lớp (phải). Mỗi khối đặc trưng ở giữa được thể hiện bằng các cột. Kiến trúc mẫu này có tên là tiny VGG Net.

Vì luận văn chủ yếu là tận dụng những ưu điểm của loại mạng này nên tiếp theo sẽ trình bày kỹ càng về các siêu tham số cũng như cách kết nối của chúng trong từng lớp.

3.2.3 Lớp tích chập (Convolutional Layer - CONV)

Lớp tích chập là lớp rất quan trọng và làm hầu hết mọi công việc tính toán trong kiểu kiến trúc này. Nó nổi tiếng đến mức mặc dù trong kiến trúc ConvNets còn chứa nhiều lớp khác nhưng được đặt theo tên của nó - Mạng nơ-ron tích chập (hay còn gọi là Convolution Neural Network).

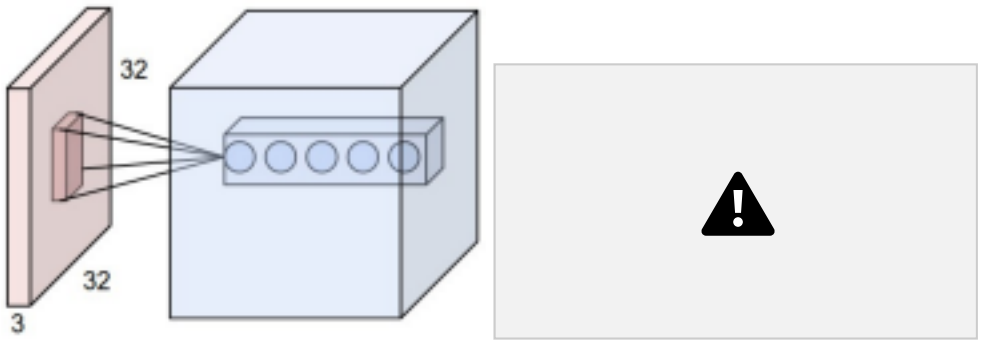
Tổng quan và trừu tượng bỏ qua khái niệm nơ-ron: Chúng ta lúc này có thể xem lớp CONV bao gồm một bộ các bộ lọc có thể học được. Mọi bộ lọc đều nhỏ về mặt không gian (dọc theo chiều rộng và chiều cao), nhưng kéo dài qua toàn bộ chiều sâu của khối đầu vào. Ví dụ: bộ lọc điển hình trên lớp đầu tiên của ConvNet có thể có kích thước 5x5x3 (tức là chiều rộng và chiều cao 5 pixel và chiều sâu là 3 vì hình ảnh có độ sâu 3, các kênh màu). Trong quá trình xử lý, ta trượt (chính xác hơn, tích chập) từng bộ lọc theo chiều rộng và chiều cao của khối đầu vào và tính toán tích vô hướng giữa bộ lọc và đầu vào tại tất cả các vị trí. Khi ta trượt bộ lọc qua chiều rộng và chiều cao của âm lượng đầu vào, ta sẽ tạo ra một đầu ra được gọi là feature map - đặc trưng, có 2 chiều cung cấp các phản hồi của bộ lọc đó ở mọi vị trí không gian. Theo trực giác, mạng sẽ học các bộ lọc kích hoạt khi chúng thấy một số hiện tượng trực quan như cạnh hoặc một vệt màu nào đó trên lớp đầu tiên, hoặc cuối cùng là toàn bộ mô hình tổ ong hoặc bánh xe trên các lớp cao hơn của mạng.

Kết nối cục bộ: Khi làm việc với các đầu vào nhiều chiều như ảnh, chúng ta dễ thấy việc kết nối các nơ-ron với tất cả các nơ-ron ở lớp liền trước là không thực tế. Thay vào đó, chúng ta sẽ kết nối mỗi nơ-ron chỉ với một vùng cục bộ của khối đầu vào. Phạm vi không gian của kết nối này là một siêu tham số được gọi là trường tiếp nhận (receptive field) của nơ-ron (tương đương đây là kích thước bộ lọc). Phạm vi của kết nối dọc theo trục độ sâu luôn bằng độ sâu của khối đầu vào. Điều quan trọng là

phải nhấn mạnh lại sự bất đối xứng này trong cách chúng ta xử lý các kích thước không gian (chiều rộng và chiều cao) và kích thước chiều sâu: Các bộ lọc xử lý cục bộ trong không gian (dọc theo chiều rộng và chiều cao), nhưng luôn luôn xử lý toàn cục đối với chiều sâu của khối đầu vào.

Ví dụ 1: Giả sử rằng khối đầu vào có kích thước [32x32x3], (ví dụ: hình ảnh RGB CIFAR-10). Nếu trường tiếp nhận (hoặc kích thước bộ lọc) là 5x5, thì mỗi nơ ron trong Lớp Conv sẽ có trọng số cho một vùng [5x5x3] trong khối đầu vào, với tổng số 5 * 5 * 3 = 75 trọng số (và +1 tham số cho độ chệch). Lưu ý rằng phạm vi kết nối dọc theo trục độ sâu phải là 3, vì đây là độ sâu của khối đầu vào.

Ví dụ 2: Giả sử một khối đầu vào có kích thước [16x16x20]. Sau đó, sử dụng trường tiếp nhận có kích thước là 3x3, mỗi nơ-ron trong Lớp Conv giờ đây sẽ có tổng cộng 3 * 3 * 20 = 180 kết nối với khối đầu vào. Lưu ý rằng, một lần nữa, kết nối là cục bộ trong không gian (ví dụ: 3x3), nhưng toàn cục theo độ sâu đầu vào (20).



Hình 3.3: Trái: Một khối đầu vào có màu đỏ và một khối nơ-ron trong lớp Convolution đầu tiên. Mỗi nơ-ron trong lớp tích chập chập chỉ được kết nối với một vùng cục bộ trong khối lượng đầu vào theo không gian. Lưu ý, có nhiều nơ-ron (5 trong ví dụ này) dọc theo độ sâu, tất cả đều nhìn vào cùng một khu vực trong đầu vào. Phải: Các nơ-ron vẫn phải tính toán tích vô hướng giữa bộ trọng số và khối đầu vào theo sau là phép toán phi tuyến tính, nhưng khả năng kết nối của chúng hiện bị hạn chế theo không gian cục bộ.

Về depth, stride và zero-padding: Đây là ba siêu tham số dùng để điều khiển kích cỡ của khối đầu ra và phần này chúng ta cũng sẽ thảo luận về cách kết nối của các nơ-ron trong Lớp Conv đối với khối đầu vào.

1. Đầu tiên, chiều sâu của khối đầu ra là một siêu tham số, nó tương ứng với số bộ lọc ta dùng, số

bộ lọc này cũng được học để phát hiện đặc trưng trong khối đầu vào thông qua quá trình huấn luyện.

2. Thứ 2, bước nhảy cũng là một siêu tham số, chỉ định độ trượt của bộ lọc trên khối đầu vào theo chiều không gian.

3. Thứ 3, zero-padding là một siêu tham số để chỉnh định số phần tử bằng 0 được đệm thêm theo chiều không gian, tham số này liên quan chặt chẽ và được dùng để điều chỉnh kích cỡ của khối đầu ra.

Theo đó, chúng ta có thể tính kích thước không gian của khối đầu ra qua một hàm nhận kích thước khối đầu vào (*W*), kích thước trường tiếp nhận của các nơ ron lớp Conv (*F*), stride (*S*) và zero-padding được sử dụng (*P*). Công thức ấy được tính toán bởi $(W - F + 2P)/S + 1$. Ví dụ: đối với đầu vào 7 × 7 và bộ lọc 3 × 3 với stride 1 và zero-padding 0, chúng ta sẽ nhận được đầu ra 5 × 5. Với stride 2 chúng ta sẽ có được đầu ra 3 × 3.

Hình 3.4: Mô phỏng sắp xếp không gian. Trong ví dụ này chỉ có một chiều không gian (trục x), một nơ-ron có kích thước trường tiếp nhận là $F = 3$, kích thước đầu vào là $W = 5$ và zero-padding $P = 1$. Trái: Chuỗi nơ-ron với stride $S = 1$, cho đầu ra có kích thước $(5 - 3 + 2)/1 + 1 = 5$. Phải: Với stride $S = 2$, cho đầu ra có kích thước $(5 - 3 + 2)/2 + 1 = 3$. Lưu ý rằng không thể sử dụng stride $S = 3$ vì nó sẽ không nằm gọn trong khối đầu vào. Về mặt toán học, điều này có thể được xác định vì $(5 - 3 + 2) = 4$ không chia hết cho 3. Trọng số của các nơ-ron nằm trong ví dụ này là [1, 0, −1] (như hình bên phải) và độ chệch của nó bằng không. Các trọng số này được chia sẻ trên tất cả các tế bào thần kinh màu vàng.

Về việc sử dụng zero-padding: Trong ví dụ trên bên trái, lưu ý rằng kích thước đầu vào là 5 và kích thước đầu ra là bằng nhau và bằng 5. Điều này đạt được vì các trường tiếp nhận có kích thước là 3 và ta đã sử dụng zero-padding để đệm thêm 2 phần tử 0 vào 2 đầu của đầu vào. Nếu không sử dụng zerp-padding, theo đó, khối đầu ra sẽ có kích thước không gian là 3, theo ví dụ trên bên phải. Nói chung, siêu tham số zero-padding được tính toán theo công thức $P = \frac{F-1}{2}$ khi stride $S = 1$ đảm bảo rằng khối đầu vào và khối đầu ra sẽ có cùng kích thước theo không gian. Việc sử dụng phần đệm bằng 0 theo cách này là rất phổ biến trong các kiến trúc ConvNet.

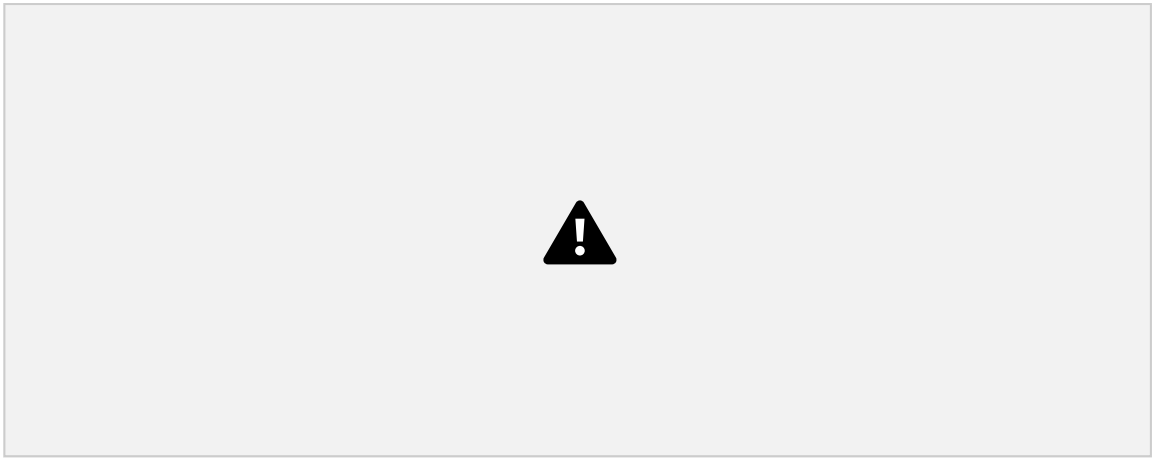
Ràng buộc của stride: Các siêu tham số có ảnh hưởng đến kích cỡ không gian của đầu ra là có ràng buộc lẫn nhau. Ví dụ: khi đầu vào có kích thước $W = 10$, zero-padding $P = 0$ và kích thước bộ lọc là $F = 3$, thì không thể sử dụng stride $S = 2$, vì $\frac{W-F+2P}{S} + 1 = \frac{10-3+0}{2} + 1 = 4.5$, là một số lẻ, chỉ ra rằng các nơ-ron không "khít" và đối xứng. Do đó, việc cài đặt siêu tham số này được coi là không hợp lệ, các thư viện lập trình hiện đại có thể ném ra ngoại lệ hoặc thực thi zero-padding phần còn lại hoặc cắt khối đầu vào để tiếp tục thực thi tính toán. Việc định cỡ các ConvNets một cách thích hợp là một vấn đề cần thiết và phải làm.

Ví dụ thực tế: kiến trúc của Krizhevsky và cộng sự [30] đã chiến thắng ImageNet challenge năm 2012 nhận hình ảnh đầu vào có kích thước [227 × 227 × 3]. Trên Lớp Convolutional đầu tiên, mô hình đã sử dụng các nơ-ron với kích thước trường tiếp nhận $F = 11$, stride $S = 4$ và zero-padding $P = 0$. Vì $\frac{227-11}{4} + 1 = 55$ và do lớp Conv có độ sâu $K = 96$, nên khối đầu ra của lớp Conv có kích thước [55×55×96]. Mỗi nơ-ron 55×55×96 trong tập này được kết nối với một vùng có kích thước [11×11×3] trong khối đầu vào. Hơn nữa, tất cả 96 nơ-ron trong mỗi cột độ sâu được kết nối với cùng một khu vực [11 × 11 × 3] của đầu vào, nhưng tất nhiên với các trọng số khác nhau.

Chia sẻ trọng số: đặc tính chia sẻ trọng số của lớp tích chập cũng là một đặc tính nổi bật và quan trọng. Sử dụng ví dụ trên, chúng ta thấy rằng có $55 * 55 * 96 = 290.400$ nơ-ron trong Lớp Conv

đầu tiên và mỗi loại có $11 * 11 * 3 = 363$ trọng số và 1 độ chệch. Kết hợp với nhau, ta có tối đa $290400 * 364 = 105.705.600$ tham số trên lớp đầu tiên của riêng ConvNet. Rõ ràng, con số này rất lớn.

Nhưng chúng ta có thể giảm đáng kể số lượng tham số bằng cách đưa ra một giả định: Nếu một đặc trưng hữu ích ở vị trí không gian (x, y), thì cũng sẽ hữu ích khi ở một vị trí khác $(x2, y2)$. Nói cách khác, ta có thể tưởng tượng một khối đặc trưng có kích cỡ [55*x*55*x*96] có 96 lát cắt có kích thước [55*x*55] xếp chồng lên nhau, chúng ta ràng buộc các nơ ron trong mỗi lát cắt sử dụng cùng bộ trọng số và độ chệch. Với cách cài đặt này, Lớp Conv đầu tiên trong ví dụ mẫu chỉ cần 96 bộ trọng số cho 96 lát cắt, với tổng số $96 * 11 * 11 * 3 = 34.848$ trọng số, hoặc 34.944 tham số (+96 số độ chệch). Ngoài ra, tất cả các nơ-ron trong mỗi lát cắt $55 * 55$ sẽ sử dụng cùng một bộ tham số. Thực tế trong quá trình truyền ngược, mỗi nơ ron sẽ tính toán đạo hàm cho các trọng số của nó, nhưng các đạo hàm này sẽ được cộng dồn trên mỗi lát cắt và chỉ cập nhật một lần cho một bộ trọng số trên mỗi lát cắt.



Hình 3.5: Các bộ lọc được học trong kiến trúc đề xuất bởi Krizhevsky và cộng sự [30]. Mỗi bộ lọc này có kích cỡ [11*x*11*x*3] và được chia sẻ cho 55 nơ-ron trong một lát cắt có cùng chiều sâu

Tổng kết, lớp tích chập:

• nhận một khối đầu vào kích cỡ $W_1 \times H_1 \times D_1$.

• cần 4 siêu tham số:

– Số bộ lọc K.

– Kích cỡ trường tiếp nhận F.

– Bước trượt stride S.

– Số lượng đệm 0, zero-padding P.

• cho ra khối đầu ra kích cỡ $W_2 \times H_2 \times D_2$, với:

– $W_2 = \frac{W_1 - F + 2*P}{S+1}$

– $H_2 = \frac{H_1 - F + 2*P}{S+1}$

– $D_2 = K$

• Với đặc tính chia sẻ trọng số, nó cần $F \cdot F \cdot D_1$ trọng số cho mỗi bộ lọc, vậy tổng cộng cần $(F \cdot F \cdot D_1) \cdot K$ trọng số và K độ chệch.

• Đối với khối đầu ra, lát cắt thứ d (kích cỡ $W_2 \times H_2$) là kết quả của việc thực hiện một phép tích chập đúng của bộ lọc thứ d trên khối đầu vào với bước trượt S và sau đó cộng với độ chệch thứ d.

3.2.4 Lớp Gộp (Pooling Layer - POOL)

Trong các kiến trúc ConvNets, ta thường thấy một lớp gộp được chèn vào giữa các lớp tích chập, khởi nguồn từ mạng LeNet [31]. Chức năng của nó là giảm kích thước không gian của khối đặc trưng nhằm giảm bớt lượng tham số và tính toán trong mạng, và do đó cũng kiểm soát tình trạng học quá

khớp. Lớp Pooling hoạt động độc lập trên mỗi lát cắt của khối đặc trưng và thay đổi kích thước theo không gian, sử dụng một hàm (như *max*) để tính toán cho từng vị trí không gian thuộc khối đặc trưng đầu ra. Hình thức phổ biến nhất là một lớp gộp là các bộ lọc có kích thước 2 × 2 với bước trượt stride $S = 2$ theo mỗi lát cắt trong khối đầu vào. Dễ thấy sau khi thực hiện tính toán, lớp gộp loại bỏ 75% lượng thông tin trong lớp đầu vào. Mỗi phép tính gộp sẽ lấy 2 × 2 số theo chiều không gian và trả ra số lớn nhất, trong khi vẫn duy trì chiều sâu của khối đặc trưng. Tổng quát hơn, lớp gộp được đặc tả như sau:

- nó chấp nhận một khối đầu vào có kích cỡ $W_1 \times H_1 \times D_1$.
- cần 2 siêu tham số:
  - kích cỡ gộp F.
  - bước trượt S.
- khối đầu ra có kích cỡ $W_2 \times H_2 \times D_2$, trong đó:
  - $W_2 = \frac{W_1 - F}{S} + 1$
  - $H_2 = \frac{H_1 - F}{S} + 1$
  - $D_2 = D_1$
- không có tham số.

Hình 3.6: Lớp gộp giảm kích thước không gian của khối đầu vào, độc lập theo chiều sâu. Trái: Trong ví dụ này, khối đầu vào có kích thước [224×224×64] được gộp với bộ lọc có các siêu tham số $F = 2$, $S = 2$, và khối đầu ra có kích thước [112×112×64]. Phải: Một ví dụ chi tiết về hoạt động của lớp Max-Pooling, nó lấy giá trị tối đa trong vùng gộp làm giá đầu ra.

3.3 Kiến trúc mã hóa - giải mã và cơ chế chú ý đối với bài toán nhận

ảnh

3.3.1 Kiến trúc mã hóa - giải mã

Kiến trúc mã hóa - giải mã theo tên gọi của nó, bao gồm 2 phần riêng biệt: phần mã hóa và phần giải mã. Trong bài toán nhận ảnh, ảnh đầu vào sẽ được đi qua phần mã hóa trước để tạo ra thể hiện tiềm ẩn sau đó phần thể hiện sẽ tiếp tục đi qua phần giải mã để cho ra đầu ra mong muốn. Một cách toán học hơn, cho ảnh đầu vào I, ta có thể gọi phần mã hóa là một hàm khả vi $g = E(I)$, và phần giải mã là một hàm khả vi khác $O = D(g)$, khi đó chồng chất hàm $O = D(E(I))$ được gọi là một bộ mã hóa giải mã, với O là đầu ra mong muốn.

Một trong những kiến trúc mã hóa - giải mã nổi tiếng hiện nay và được lấy ý tưởng trong luận văn này là mạng U-net, vốn được phát triển để phân đoạn ảnh y tế, xem Hình 3.7,

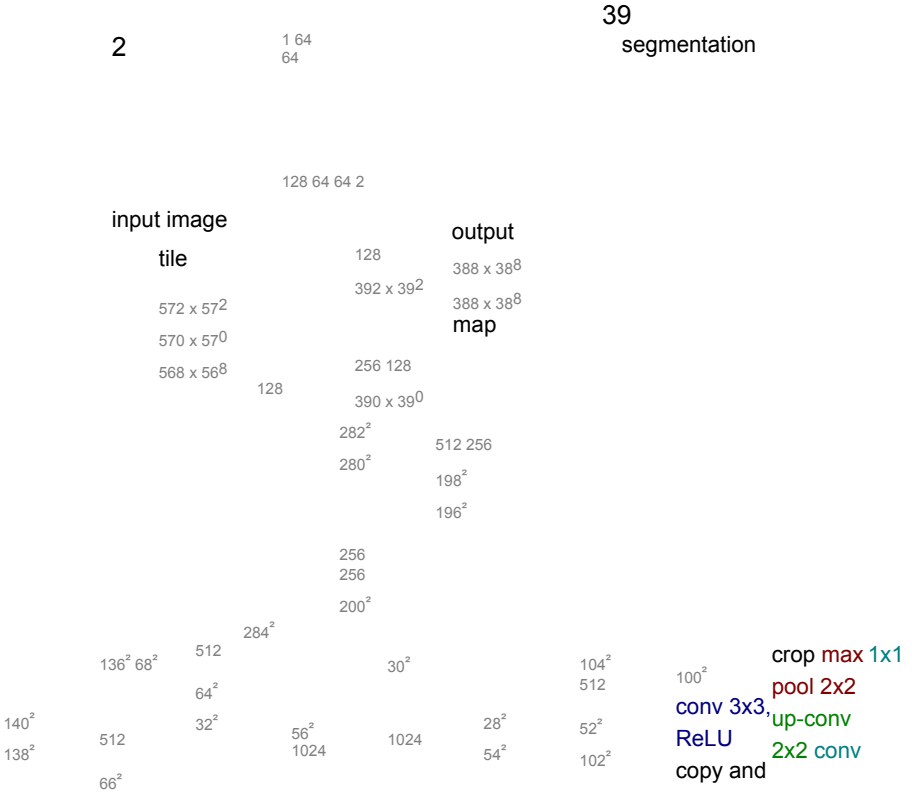


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue

Hình 3.7: Kiến trúc mạng chữ U [45] (ví dụ cho 32x32 pixel ở độ phân giải thấp nhất). Mỗi hình chữ nhật box corresponds to a multi-channel feature map. The number of channels is denoted màu xanh tương ứng với một khối đặc trưng. Hộp màu trắng bên khối giải mã là bản sao của khối đặc on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. trưng ở cùng bậc bên khối mã hóa. Các mũi tên biểu thị các hoạt động khác nhau như được chú thích.

Kiến trúc này được Olaf Ronneberger, Philipp Fischer, và Thomas Brox đề xuất và mô tả trong bài as input. First, this network can localize. Secondly, the training data in terms báo U-Net: Convolutional Networks for Biomedical Image Segmentation, xuất bản năm 2015, of patches is much larger than the number of training images. The resulting network won the EM segmentation challenge at ISBI 2012 by a large margin. cho bài toán phân đoạn ảnh y sinh, xem Hình 3.8[3] Obviously, the strategy in Ciresan et al. [1] has two drawbacks. First, it is quite slow because the network must be run separately for each patch, and there is a lot of redundancy due to overlapping patches. Secondly, there is a trade-off between localization accuracy and the use of context. Larger patches require more max-pooling layers that reduce the localization accuracy, while small patches allow the network to see only little context. More recent approaches [11,4] proposed a classifier output that takes into account the features from multiple layers. Good localization and the use of context are possible at the same time.

In this paper, we build upon a more elegant architecture, the so-called “fully convolutional network” [9]. We modify and extend this architecture such that it works with very few training images and yields more precise segmentations; see Figure 1. The main idea in [9] is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. Hence, these layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the upsampled

Hình 3.8: Mô hình nhận vào một ảnh được chụp bởi kính hiển vi điện tử (Electron Microscope) và trả về Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here phân đoạn của cấu trúc nơ-ron. [45] segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

output. A successive convolution layer can then learn to assemble a more precise output based on this information.

Kiến trúc mạng có thể được nhìn như bao gồm hai phần: mã hóa và giải mã, bao gồm các lớp convolutional liên tiếp để chuyển đổi các khối đặc trưng, và sau cùng là khối convolution với kernel size 1 × 1 làm nhiệm vụ phân lớp cho bài toán phân đoạn ảnh.

3.3.2 Cơ chế chú ý

Trong các thập kỷ gần đây khi các mạng học sâu với nơ-ron tích chập được phát triển, bắt đầu từ kiến trúc LeNet [31] đến các mạng kiểu Residual gần đầy [20] thì kiến trúc mạng bắt đầu sâu dần. Mạng VGGNet [48] cho thấy chồng chất các lớp Convolutional cho ra một kết quả tốt, sau đó, kiến trúc ResNet [20] ra đời với cùng tư tưởng chồng chất các lớp Convolutional nhưng có thêm các kết nối bỏ

qua (skip connection) để có thể build các mạng sâu đến rất sâu. Bên cạnh đó, GoogLeNet [50] cho thấy rằng chiều rộng của kiến trúc của là một yếu tố quan trọng để cải thiện hiệu quả của mô hình qua kiến trúc Inception. Một khía cạnh khác, luận văn này tập trung vào tính *chú ý* của một mô hình học sâu.

12/14/2019 attention_girls.jpg (950×534)

Cơ chế chú ý được giới thiệu lần đầu bởi Dzmitry Bahdanau, Kyunghyun Cho, và Yoshua Bengio trong bài báo Neural Machine Translation by Jointly Learning to Align and Translate để cải thiện kết quả dịch máy đối với những câu dài. Sự hiệu quả của cơ chế này trong bài toán Machine Translation đã kéo theo một loạt các nghiên cứu về cơ chế chú ý trong các bài toán khác nhau như bài toán đặt tiêu đề cho ảnh, phân loại, hay phân đoạn ảnh,.. Một ví dụ phổ biến về cơ chế chú ý cho bài toán đặt tiêu đề cho ảnh, xem Hình 3.9, hình ảnh với tiêu đề "Một cô bé đang ném chiếc đĩa nhựa", và các vùng được khoanh tròn là các vùng mà mô hình máy tính chú ý vào, rõ ràng là những vùng khoanh màu xanh lục mới đáng chú ý để cho ra một câu tiêu đề như vậy hơn là những vùng được khoanh tròn màu vàng.

Hình 3.9: Một cô bé đang ném một chiếc đĩa nhựa

Hay gần với bài toán mà luận văn đang hướng tới hơn, theo Bảng tổ hợp các đơn vị hoạt động được 41

file:///home/z/Desktop/thesis_images/attention_girls.jpg 1/1

phân theo cảm xúc được mô tả rõ ở Bảng 3.1,và ngay trong bài báo được khảo sát ở Phần 2.2, các nhà khoa học đã giả định rằng các vùng gần mắt và vùng miệng rõ ràng có đóng góp nhiều hơn đường cằm, vùng tai hay vùng tóc trong việc dự đoán cảm xúc của một người. Trong một số các nghiên cứu, các vùng được chọn được trích xuất ra cơ bản dựa vào các điểm mốc trên khuôn mặt, được phát hiện bởi thành quả nghiên cứu của Kazemi, Vahid, và Josephine Sullivan [27]. Sau đó dựa vào các điểm mốc này để trích xuất ra vùng mặt cần quan tâm, đưa vào mạng học sâu để trích xuất đặc trưng cần thiết và sau đó là thực hiện phân lớp. Các nghiên cứu này đa phần là đạt kết quả tốt trên một số tập dữ liệu. Từ đây cho thấy rõ ràng đối với bài toán nhận diện cảm xúc con người dựa

trên khuôn mặt có sự đóng góp khác nhau giữa các vùng khác nhau trên mặt người.

Một trong những nghiên cứu về mạng học sâu có sử dụng cơ chế chú ý cho bài toán phân loại ảnh là nghiên cứu của Wang, Fei và cộng sự [57], xem Hình 3.11, có thể thấy trong toàn kiến trúc có các mô-đun chú ý xếp chồng lên nhau, trong mỗi mô-đun chú ý ấy khối đặc trưng được tách theo 2 nhánh khác nhau, xem Hình ..., sau đó đầu ra H của hai khối ấy được kết hợp lại theo Công thức 3.1,

$$H_{i,c}(x) = (1 + M_{i,c}(x)) * F_{i,c}(x) \quad (3.1)$$

trong đó $M(x)$ bao gồm các giá trị trong đoạn [0, 1], vậy với $M(x)$ xấp xỉ 0 thì $H(x)$ bằng với $F(x)$. Cách làm này gọi là học chú ý thặng dư, tránh đi việc chú ý nhầm làm mất đi các đặc trưng quan trọng.

Figure 2: Example architecture of the proposed network for ImageNet. We use three hyper-parameters for the design of

Hình 3.10: Kiến trúc được giới thiệu trong bài báo [57], có 3 siêu tham số cần quan tâm: là *p, t* và *r*. *p*

Attention Module: *p, t* and *r*. The hyper-parameter *p* denotes the number of pre-processing Residual Units before splitting kí hiệu cho số Đơn vị trích xuất đặc trưng trước khi đi vào Trunk branch và Mask Branch, *t* kí hiệu cho

into trunk branch and mask branch. *t* denotes the number of Residual Units in trunk branch. *r* denotes the number of Residual Units between adjacent pooling layer in the mask branch. In our experiments, we use the following hyper-parameters setting:

số Đơn vị trích xuất đặc trưng nằm ở Trunk branch và *r* là số Đơn vị trích xuất đặc trưng giữa các lớp *{p* = 1, *t* = 2, *r* = 1*}*. The number of channels in the soft mask Residual Unit and corresponding trunk branches is the

Pooling kề nhau trong Mask Branch

same.

3.1. Attention Residual Learning

However, naive stacking Attention Modules leads to the obvious performance drop. First, dot production with mask range from zero to one repeatedly will degrade the value of features in deep layers. Second, soft mask can potentially break good property of trunk branch, for example, the iden

branch and forward to top layers to weaken mask branch's feature selection ability. Stacked Attention Modules can gradually refine the feature maps. As show in Fig.1, fea tures become much clearer as depth going deeper. By using attention residual learning, increasing depth of the proposed Residual Attention Network can improve performance con

tical mapping of Residual Unit.

sistently. As shown in the experiment section, the depth of 42

We propose attention residual learning to ease the above problems. Similar to ideas in residual learning, if soft mask unit can be constructed as identical mapping, the perfor mances should be no worse than its counterpart without at tention. Thus we modify output H of Attention Module as

$$H_{i,c}(x) = (1 + M_{i,c}(x)) * F_{i,c}(x) \quad (3)$$

Residual Attention Network is increased up to 452 whose performance surpasses ResNet-1001 by a large margin on CIFAR dataset.

3.2. Soft Mask Branch

Following previous attention mechanism idea in

Soft Mask Branch Trunk Branch ◆◆ $f_3(x)$

Table 1: The testnetwork with diff

Layer
Conv1
Max pooling

Residual Unit

up sample up sample down sample

◆◆

down sample

convolution
Activation Funct$f_1(x)$
$f_2(x)$

1 + ◆ ◆ % ◆(◆)

receptive field

Attention Module

◆

branch.

Residual Unit Attention Module

Figure 3: The receptive field comparison between mask

Hình 3.11: So sánh vùng nhận thức giữa Mask branch và Trunk branch branch and trunk

Việc kết nối kiến trúc chú ý như vậy dẫn đến một tính chất rất hay, tính chống nhiễu. Tạm bỏ qua

range to [0, 1] after two consecutive 1 × 1 capture information from different scales. convolution lay tính thặng dư, ta xét công thức dẫn The full module is illustrated in Fig.2. ra kết quả của Mô-đun H được tính theo Công thức 3.2

ers. We also added skip connections between bottom-up and top-down parts to

Residual Unit Attention Module

$$H_{i,c}(x) = M_{i,c}(x) * T_{i,c}(x) \quad (3.2)$$

The bottom-up top-down structure has been applied to image segmentation and human pose

estimation. However,

Residual Unit

trong đó i chạy trên tất cả vị trí không gian và $c \in \{1..C\}$ chỉ định số kênh. Lúc này, vai trò của của

the difference between our structure and the previous one

Average pooling

Mô-đun Chú ý H không những phát huy tác dụng theo chiều xuôi, mà còn trong chiều ngược cập nhật

lies in its intention. Our mask branch aims at improving

FC,Softmax

trọng số trong lúc huấn luyện mô hình. Giả định φ và θ lần lượt là bộ trọng số của Trunk branch và trunk

branch features rather than solving a complex prob

Mask branch, ta xem xét Công thức 3.3, params×FLOPs×Trunk de

lem directly. Experiment in Sec.4.1 is conducted to verify

above arguments. Table 2: Residua

$$\frac{\partial \varphi}{\partial M(x,\theta)T(x,\varphi)} = M(x,\theta)\frac{\partial T(x,\varphi)}{\partial \varphi} \quad (3.3)$$

3.3. Spatial Attention and Channel Attention

for ImageNet. AWe make the size

Ta dễ dàng thấy được, trọng số cập nhật cho Trunk branch cũng được nhân với đầu ra của Mask

In our work, attention provided by mask branch changes branch 7×7 to be

branch, từ đó các trọng số sai sẽ tránh bị cập nhật, việc này có tác dụng rất lớn trong việc chống nhiễu

adaptably with trunk branch features. However, constrains to attention can still be added to mask branch by changing

thường gặp trong thực tế.

normalization step in activation function before soft mask output. We use three types of activation functions corre 3.4 Phương pháp nhận diện khuôn mặt

sponding to mixed attention, channel attention and spatial attention. Mixed attention f_1 without additional restriction use simple sigmoid for each channel and spatial position.

map size. Thus 3,branch with input The Attention MUnit [11] with thsame as ResNet [1

Phát hiện mặt người là bài toán cơ bản được xây dựng từ nhiều năm nay, có nhiều phương pháp được

Channel attention f_2 performs $L2$ normalization within all The experimen

đưa ra như sử dụng các phương thức truyền thống tới các mạng học sâu hiện đại. Vào tháng 8, 2017, với

channels for each spatial position to remove spatial infor attention has the

bản OpenCV 3.3 được phát hành, một phiên bản phát hiện khuôn mặt bằng mạng Single Shot Detector

mation. Spatial attention f_3 performs normalization within mally focus on on

được cài đặt bằng Caffe đã được sẵn sàng sử dụng ở dưới mô-đun dnn [58]. Vì phần này tôi tuyệt đối

feature map from each channel and then sigmoid to get soft tham khảo và sử

dụng lại chức năng của OpenCV nên không trình bày thêm ở đây. mask related to spatial information only.

$$f_1(x_{i,c}) = \frac{1}{1 + exp(-x_{i,c})} \quad (4)$$

$$f_2(x_{i,c}) = x_{i,c}$$

$$kx_{ik} \quad (5)$$

1

attention [3] or spconstrain on soft However, as supption change adaptistraint leads to the

4. Experiment

Chương 4

Dữ liệu

4.1 Tổng quan về dữ liệu của bài toán

Cùng với sự phát triển mạnh mẽ của lĩnh vực điện toán cảm xúc trong các thập kỷ gần đây cùng với sự phát triển chóng mặt về sức mạnh phần cứng, các bộ dữ liệu phục vụ cho việc phân lớp biểu cảm cũng phát triển một cách mạnh mẽ (xem Hình 4.1). Một bộ cơ sở dữ liệu nhận dạng biểu cảm khuôn mặt là một tập hợp bao gồm các hình ảnh hoặc video clip chứa mặt người được gắn nhãn cảm xúc tương ứng với trạng thái lúc đó. Tập các ảnh được gắn nhãn dùng cho đào tạo, kiểm thử và kiểm tra các thuật toán để phát triển các hệ thống nhận dạng biểu cảm. Gắn nhãn cảm xúc có thể được thực hiện trong các nhãn cảm xúc rời rạc hoặc trên liên tục. Hầu hết các cơ sở dữ liệu thường dựa trên lý thuyết cảm xúc cơ bản của Paul Ekman và Armindo Freitas-Magalhaes giả định sự tồn tại của sáu cảm xúc cơ bản rời rạc (giận dữ, sợ hãi, ghê tởm, bất ngờ, vui, buồn). Tuy nhiên, một số cơ sở dữ liệu bao gồm việc gắn thẻ cảm xúc theo thang hóa trị kích thích liên tục. Và một số cơ sở dữ liệu bao gồm kích hoạt AU dựa trên FACS. Trong luận văn này tôi coi giả định của Ekman là đúng, có sự tồn tại của 6 cảm xúc cơ bản rời rạc. Thông tin về một số bộ dữ liệu có sẵn được trình bày như ở Bảng 4.1, có thể thấy sự đa dạng về cả phương pháp thu cập, kích cỡ và phân phối biểu cảm của các bộ dữ liệu này.

2007-2009 2011

Zhao et al. [1]

(LBP, SVM) Shan et al. [2] CK+, Zhong et al. [3] (Sparse
(LBP, AdaBoost) MMI learning)

2013 Y. Tang [4] (SVM)

Small, Lab

Ng,Hong-Wei et al. [5] controlled.

(Transfer Learning) Large scale, In-the-wild

2015-2016

Fan et al. [6] (CNN-RNN) FER2013

Regions Learning) Emotiw EmotioNet

Y. Fan et al. [7] (Multi HD-Nguyen [8] (Multi Level AffectNet

Learning) **2017-2019**

[1] Guoying Zhao et al. "Facial expression recognition from near-infrared videos". In:Image VisionComput.29 (2011).
[2] C. Shan, et al., "Facial expression recognition based on local binary patterns: A comprehensive study," IVC, 2009.
[3] L. Zhong et al., "Learning active facial patches for expression analysis," CVPR, 2012.
[4] Y. Tang, "Deep learning using linear support vector machines," ICML, 2013.
[5] Ng, Hong-Wei, et al. "Deep learning for emotion recognition on small datasets using transfer learning." ICML, 2015.
[6] Fan, et al. "Video-based emotion recognition using cnn-rnn and c3d hybrid networks," IMCL, 2016.
[7] Y. Fan el al. "Multi-region ensemble convolutional neuralnetwork for facial expression recognition", ICANN 2018.
[8] HD Nguyen et al. "Facial expression recognition using a multi-level convolutional neuralnetwork", ICPRAI 2019.

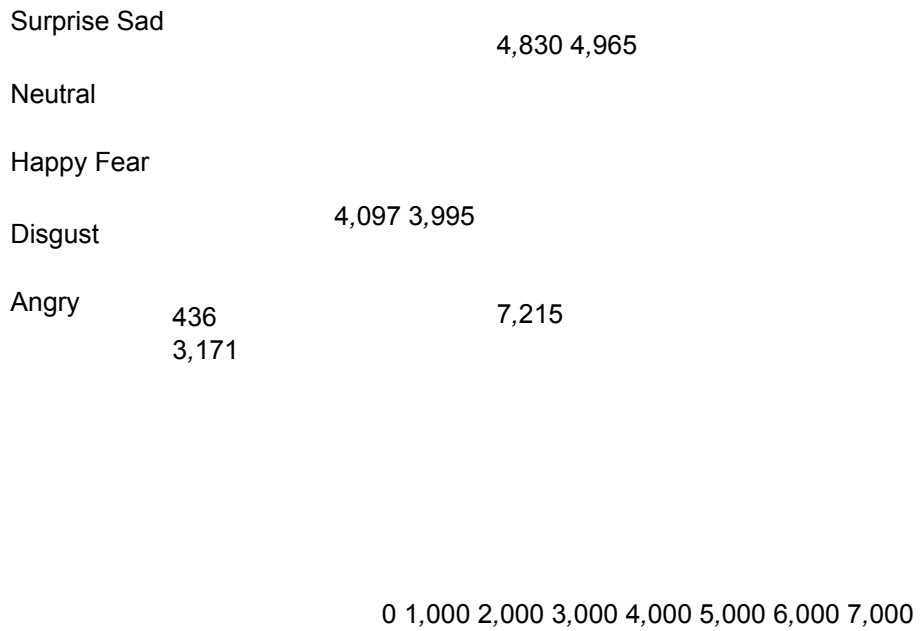
Hình 4.1: Mức độ phát triển về cả phương pháp và các tập dữ liệu trong bài tóan nhận diện biểu cảm.

Tên CSDL	Số mẫu	Cách thu thập	Phân phối biểu cảm
CK+ [33]	593 chuỗi ảnh	Lab	6 cảm xúc cơ bản + bình thường, khinh bỉ
JAFFE [34]	213 ảnh	Lab	6 cảm xúc cơ bản + bình thường
MMI [41]	740 ảnh, 2,900 videos	Lab	6 cảm xúc cơ bản + bình thường
FER-2013 [18]	35,887 ảnh	Web	6 cảm xúc cơ bản + bình thường
AFEW 7.0 [8]	1,809 videos	Phim	6 cảm xúc cơ bản + bình thường
SFEW 2.0 [9]	1,766 ảnh	Phim	6 cảm xúc cơ bản + bình thường
Oulu-Casia [63]	2,880 chuỗi ảnh	Lab	6 cảm xúc cơ bản
Emotio-Net [2]	1,000,000 ảnh	Web	23 biểu cảm
AffectNet [37]	450,000 ảnh	Web	6 cảm xúc cơ bản + bình thường

Bảng 4.1: Thông tin tổng quan một số Cơ sở dữ liệu (CSDL) có sẵn. [32]

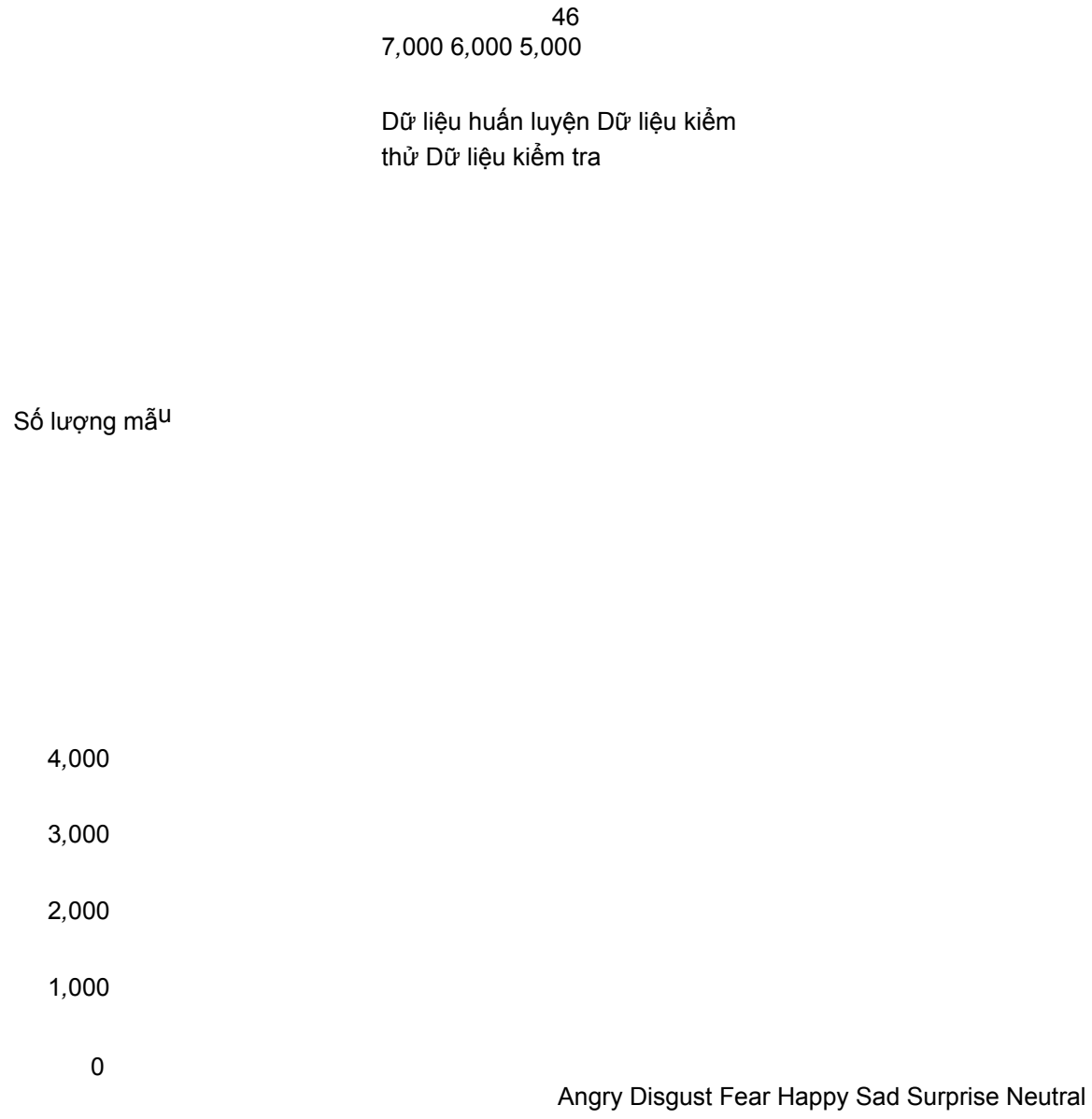
4.2 Tập dữ liệu FER2013

Bộ dữ liệu chính được sử dụng để huấn luyện, đánh giá và so sánh các mô hình là tập dữ liệu của Cuộc thi Nhận diện Biểu cảm Mặt người - Challenges in Representation Learning: Facial Expression Recognition Challenge [18] được tổ chức năm 2013 trên Kaggle. Tập dữ liệu này được chuẩn bị bởi Pierre-Luc Carrier and Aaron Courville, như một phần dự án nghiên cứu của họ.

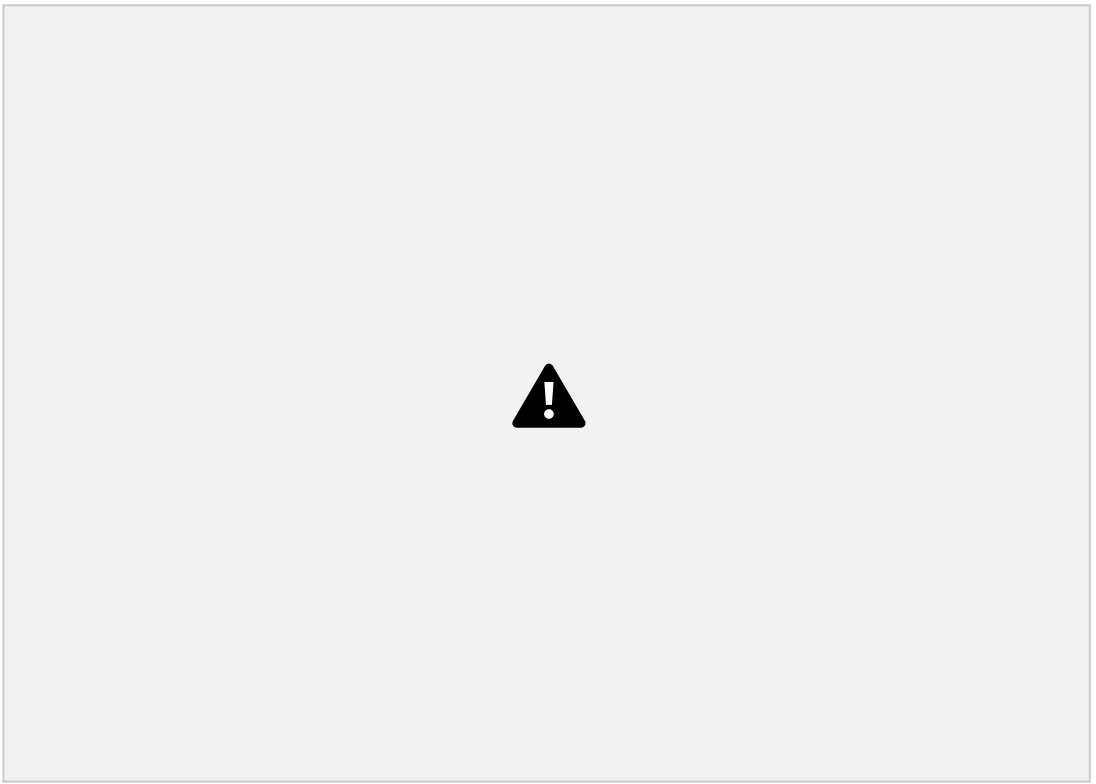


Hình 4.2: Phân phối các lớp của tập dữ liệu FER2013 trong tập huấn luyện [18].

Tập dữ liệu này bao gồm 35,887 bức ảnh xám cỡ 48 × 48: Trong đó 28,709 ảnh được dùng cho việc huấn luyện, 3,859 ảnh được dùng để kiểm thử (Public Test) và 3,589 ảnh còn lại được dùng để thực hiện kiểm tra (Private Test). Mỗi tấm ảnh chứa một gương mặt thuộc một trong bảy lớp (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral), xem Biểu đồ 4.3. Một trong những thách thức lớn nhất của tập dữ liệu này là sự mất cân bằng dữ liệu. Chi tiết được thể hiện trong Hình 4.2. Thêm vào đó, tập dữ liệu này còn chứa một số mẫu bị sai bao gồm các hình ảnh không chứa khuôn mặt, gương mặt không đầy đủ, thậm chí là đánh nhãn bị sai.



Hình 4.3: Tương quan phân phối dữ liệu giữa tập huấn luyện, kiểm thử và kiểm tra trong bộ dữ liệu FER2013. Có thể thấy sự mất cân bằng ở lớp Disgust trong bộ dữ liệu này [18].



Hình 4.4: Một số hình ảnh trong tập dữ liệu. Rất nhiều biến thể về độ sáng, tuổi tác, góc độ, cường độ biểu cảm, và sự xuất hiện trong môi trường thực tế [54]

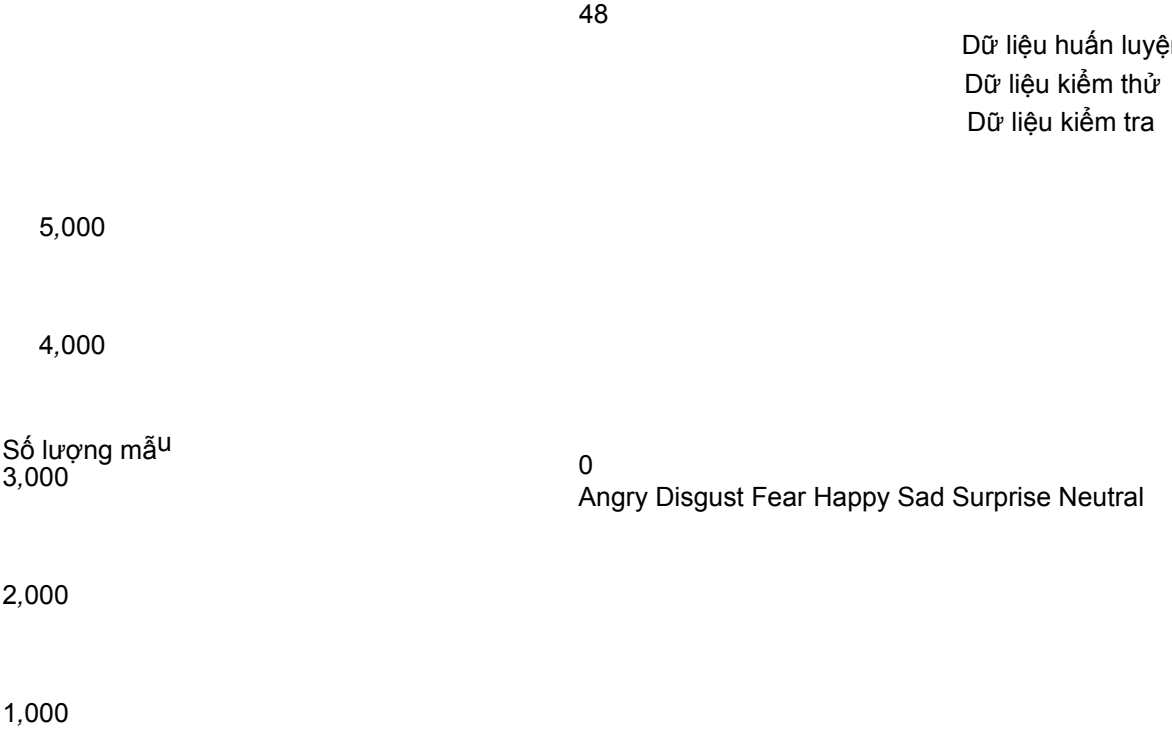
4.3 Tập dữ liệu VEMO

Dữ liệu được thu thập từ trên các trang ảnh phổ biến như Google Image, Flickr với các từ khóa như

"sinh viên Việt Nam", "con người Việt Nam", "trẻ em Việt Nam", "học sinh trong lớp", "giảng đường", "đám cưới", "show truyền hình",.. Ngoài ra dữ liệu còn được trích xuất từ 92 đoạn phim được thu thập trên các kênh phim và Youtube. Quá trình trích xuất hình ảnh từ video được chia làm 2 giai đoạn:

- Trích xuất ảnh khuôn mặt từ trong video: bước này được hoàn thành bởi mô-đun phát hiện khuôn mặt V&J [56]. Cứ 1 frame được trích xuất thì 5 frame liên tiếp sau đó bị bỏ đi, để tránh trùng lặp dữ liệu. Sau bước này ta thu thập được khoảng 3,3 triệu ảnh.
- Bước gắn nhãn dữ liệu: bước này được hoàn thành bằng tay bởi 3 người Việt Nam độ tuổi từ 18 đến 23 tuổi. Sau bước này ta có 6470 ảnh người Việt Nam được đánh nhãn.

Tập dữ liệu dùng thêm 30 nghìn ảnh của người trên toàn thế giới được đánh nhãn bởi các chuyên gia đánh nhãn [37]. Tổng cộng tập dữ liệu có 36470 ảnh, chia thành 3 tập huấn luyện - kiểm thử - kiểm tra theo tỉ lệ 0.7 - 0.15 - 0.15. Phân phối của 3 tập dữ liệu được thể hiện như Biểu đồ 4.5



Hình 4.5: Tương quan phân phối dữ liệu giữa tập huấn luyện, kiểm thử và kiểm tra trong bộ dữ liệu VEMO.

Hình 4.6: Một số hình ảnh ví dụ trong tập dữ liệu VEMO. Hình ảnh được xếp theo 7 cột tương ứng với 7 nhãn biểu cảm: Giận dữ, Ghê tởm, Sợ hãi, Hạnh phúc, Bình thường, Buồn bã, Bất ngờ. Tập dữ liệu bao gồm các ảnh màu có sự đa dạng về độ sáng, tuổi tác, góc độ, cường độ biểu cảm, và sự xuất hiện trong môi trường thực tế.

Chương 5

Phân lớp biểu cảm bằng Residual Masking Network

5.1 Giới thiệu

Một hệ thống nhận diện biểu cảm mặt người nhận một ảnh đầu vào *I* và lần lược thực hiện các bước: phát hiện (các) khuôn mặt *F* trong ảnh *I*, phân lớp biểu cảm *E* cho (từng) khuôn mặt *F* trong ảnh *I*, thực hiện phân tích dữ liệu dựa trên kết quả dự đoán *E*. Tầm vực của luận văn này là phát triển và đóng góp vào phần phân lớp biểu cảm, mô-đun đứng giữa và giữ vai trò cốt lõi trong hệ thống (xem Hình 5.1). Kết quả sau đó được kết hợp với các phương pháp nhận diện khuôn mặt có sẵn, các phương pháp hậu xử lý để có kết quả sẵn sàng cho bước phân tích sau đó. Mô-đun nhận diện cảm xúc được phát triển cơ bản sẽ nhận ảnh đầu vào là khuôn mặt *F* đã được cắt ra và trả về đầu ra *E* là một trong bảy nhãn cơ bản: Giận dữ, Ghê tởm, Sợ hãi, Hạnh phúc, Buồn bã, Bất ngờ và Bình thường, ngoài ra sẽ trả về giá trị Không xác định nếu mô-đun không xác định được giá trị biểu cảm của ảnh đầu vào.

Hình 5.1: Hệ thống nhận diện biểu cảm.

5.2 Suy luận biểu cảm từ cấu hình cơ mặt

Các nghiên cứu khoa học trong lĩnh vực phát hiện cảm xúc dựa vào cấu hình của các cơ mặt hay nét mặt giả định là chúng ta có thể suy luận được cảm xúc của con người dựa vào những cấu hình ấy, rằng người ta thường cười khi hạnh phúc, nhăn mặt khi họ buồn, cau có khi giận dữ,.. Các cấu hình cơ mặt người còn tiết lộ ra nhiều thông tin hơn việc chỉ đơn thuần là cảm xúc, nhưng trong một số cấu hình nhất định nó được nhận là thuộc vào một lớp cảm xúc nhất định theo bảng FACS.

5.2.1 Phân tích một số ví dụ dựa trên Bảng FACS

Biểu cảm giận dữ là một phản ứng cảm xúc mạnh và là một cảm xúc nguy hiểm cần được chú ý vì có thể gây kích thích bạo lực. Cảm xúc giận dữ bắt nguồn từ nhiều nguyên nhân khác nhau. Khi thất vọng, một người có thể cảm thấy tức giận đối với chướng ngại trên con đường đi đến thành công của họ. Chúng ta cũng sẽ thấy tức giận khi một ai đó muốn làm tổn hại đến chúng ta. Bên cạnh đó, bạo lực về thể xác, các lời nói gây đe dọa cũng gây ra cảm xúc tức giận. Giận dữ có một tác động đáng kể đến toàn bộ cơ thể. Nó làm gia tăng huyết áp và các cơ trong cơ thể căng lên. Trên gương mặt sẽ có những biểu hiện sau, xem Hình 5.2:

- Lông mày hạ xuống và chụm lại gần nhau, có những nếp nhăn dọc giữa hai lông

mày • Mi mắt căng và thẳng

- Đôi mắt căng chặt và tập trung vào nguồn gây ra tức giận. Đồng tử bị thu hẹp và tập trung vào nguồn gây ra tức giận
- Môi được đóng chặt hoặc mở nhẹ (chuẩn bị la hét)

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.2: Hình ảnh mặt người giận dữ trong điều kiện lab-controlled và điều kiện tự nhiên

Biểu cảm ghê tởm là một cảm xúc mạnh mẽ thể hiện sự không đồng ý, ác cảm hoặc không tán thành. Ghê tởm có liên quan chặt chẽ đến cảm nhận của các giác quan, đặc biệt là vị giác và thị giác, khi thu thập dữ liệu trong môi trường lab-controlled, các diễn viên sẽ được cho ăn chanh hoặc cho nhìn thấy các thức ăn bị hư để sản sinh ra cảm xúc này. Cảm xúc này được chủ yếu biểu thị thông qua miệng và vùng mũi, đặc biệt là vùng sóng mũi nhăn lại (xem Hình 5.3), các đặc trưng về cấu hình khuôn mặt được mô tả như sau:

- Môi trên được nâng lên.
- Có những nếp nhăn trên mũi.
- Má cũng được nâng lên.
- Mí mắt được nâng lên nhưng không chặt. Có những nếp nhăn dưới mắt.
- Lông mày được kéo xuống.

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.3: Hình ảnh mặt người ghê tởm trong điều kiện lab-controlled và điều kiện tự nhiên

Biểu cảm sợ hãi xuất hiện trong các tình huống căng thẳng hoặc nguy hiểm. Một người cảm thấy sợ hãi từ một sự việc xấu có thể xảy ra trong tương lai hay nhìn thấy tình huống bạo lực. Khi có cảm giác sợ hãi, cơ thể con người sẽ chuẩn bị để trốn thoát hoặc chống cự lại các mối đe dọa có thể xảy đến. Khi đó, nhịp tim và huyết áp tăng, mắt mở ra và con ngươi rộng ra, vì vậy mắt có thể hấp thụ lượng ánh sáng tối đa. Trong trường hợp sự sợ hãi là cực kỳ lớn có thể ảnh hưởng đến các nhóm cơ và gây tê liệt. Các dấu hiệu trên khuôn mặt cho biểu cảm sợ hãi được miêu tả như sau (xem Hình 5.4):

- Lông mày được nhấc lên và kéo vào trong
- Có những nếp nhăn trên trán
- Mí mắt trên được nâng lên
- Miệng mở và đôi môi căng lên theo cường độ của cảm xúc.

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.4: Hình ảnh mặt người sợ hãi trong điều kiện lab-controlled và điều kiện tự nhiên

Biểu cảm hạnh phúc là một trạng thái cảm xúc tích cực của con người, thường xuất hiện với một

nụ cười trên khuôn mặt. Cảm xúc hạnh phúc được biểu hiện khi chúng ta cảm thấy thỏa mãn hay đạt được thứ gì đó. Các đặc điểm điển hình của cảm xúc hạnh phúc trên khuôn mặt là (xem Hình 5.5):

- Các góc môi được kéo rộng ra và nâng lên
- Miệng có thể mở và có thể nhìn thấy răng
- Má có thể nâng lên
- Các nếp nhăn dưới mí mắt dưới có thể xuất hiện
- Các nếp nhăn xuất hiện bên ngoài khóe mắt

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.5: Hình ảnh mặt người hạnh phúc trong điều kiện lab-controlled và điều kiện tự nhiên

Biểu cảm buồn bã là một cảm xúc liên quan đến nỗi đau, hoặc đặc trưng bởi các cảm giác bất lợi, mất mát, tuyệt vọng, đau buồn, bất lực, thất vọng và buồn phiền, là trạng thái cảm xúc xuất hiện khi con người cảm thấy đau khổ. Nguồn gốc của nỗi buồn thường là việc mất một cái gì đó. Khi mang cảm xúc này, các cơ mặt mất đi sự căng thẳng và có thể có các đặc điểm điển hình sau (xem Hình 5.6):

- Các phần bên trong của lông mày được kéo xuống
- Hai bên khóe môi kéo xuống, và đôi môi run rẩy
- Ánh mắt đăm chiêu và không tập trung

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.6: Hình ảnh mặt người buồn bã trong điều kiện lab-controlled và điều kiện tự nhiên

Biểu cảm ngạc nhiên là cảm xúc đột nhiên xuất hiện. Nó đến mà không cần suy nghĩ và chỉ tồn tại

trong một thời gian ngắn. Cảm xúc ngạc nhiên có cả yếu tố tích cực và tiêu cực và không thể lường trước được. Nếu có thời gian để suy nghĩ về tình huống, phản ứng sẽ không phải là ngạc nhiên. Cảm xúc bất ngờ thường dẫn đến một cảm xúc khác: hạnh phúc hoặc buồn bã. Các đặc điểm điển hình của sự ngạc nhiên là nhướng mày, điều này cũng gây ra nếp nhăn trên trán, đôi mắt mở rộng (xem Hình 5.7).

- Lông mày được nhấc lên và kéo vào trong
- Nếp nhăn ngang xuất hiện trên trán
- Đôi mắt mở to
- Miệng có thể mở rộng mà hàm dưới kéo xuống

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.7: Hình ảnh mặt người bất ngờ trong điều kiện lab-controlled và điều kiện tự nhiên

Trạng thái bình thường là trạng thái con người không bộc lộ cảm xúc (thực ra có thể có cảm xúc trong lòng nhưng được kiểm soát hay kiềm chế mà không bộc lộ ra cơ mặt, tuy rằng vẫn có thể có những

thay đổi trên cơ mặt nhưng rất nhỏ với cường độ không đáng kể thì lúc này bài toán trở về dạng micro expression). Các cơ mặt trong trạng thái bình thường được thư giãn và không có biến đổi gì đáng kể (xem Hình 5.8)

(a) Điều kiện lab-controlled (b) Điều kiện tự nhiên

Hình 5.8: Hình ảnh mặt người ở trạng thái bình thường trong điều kiện lab-controlled và điều kiện tự nhiên

5.2.2 Nhận xét và kết luận

Qua những ví dụ trên, và trong cả các nghiên cứu gần đây, chúng ta dễ dàng thấy rằng cảm xúc được bộc lộ ra bên ngoài cơ mặt thông qua chỉ một số vùng đặc trưng, đặc biệt là vùng miệng và vùng mắt. Trong các nghiên cứu gần đây, các thông tin vùng miệng và mắt được trích xuất thông qua các *điểm mốc trên khuôn mặt* - được biết đến như *facial landmarks* hay *facial keypoints*. Các điểm mốc này đã được Vahid Kazemi và Josephine Sullivan nghiên cứu và công bố khoa học trong bài báo "One Millisecond Face Alignment with an Ensemble of Regression Trees" [27], phương pháp này nhận vào một ảnh và trả ra vị trí các điểm mốc trên khuôn mặt, xem Hình 5.9. Từ các điểm mốc này mà các nhà nghiên cứu biểu cảm có thể trích xuất ra tọa độ của vùng miệng và vùng mắt để tiến hành các phương pháp học máy hay thậm chí là học sâu để quan sát cấu hình hay sự thay đổi của cơ mặt, từ đó rút trích đặc trưng làm cơ sở cho bộ phân lớp biểu cảm.

Teknikringen 14, Stockholm, Sweden
{vahidk,sullivan}@csc.kth.se

ract

roblem of Face Alignment for
w an ensemble of regression
he face's landmark positions
of pixel intensities, achieving
ith high quality predictions.
rk based on gradient boosting
gression trees that optimizes
nd naturally handles missing
show how using appropriate
of image data helps with ef
rent regularization strategies
t overfitting are also investi
e the effect of the quantity of
of the predictions and explore
using synthesized data.

Figure 1. Selected results on the HELEN dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a millisecond.

Hình 5.9: Một số kết quả trích xuất điểm mốc khuôn mặt trong nghiên cứu của [27]

pixel intensi

new algorithm that performs

The first revolves around the indexing of

Nhưng phương pháp trích xuất các điểm mốc này đồng thời cũng có điểm yếu là không thể hiện tốt

and achieves accuracy supe

shape. The ex

ties relative to the current estimate of the

đối với ảnh cỡ nhỏ, ví dụ ảnh trong tập dữ liệu FER2013 kích cỡ chỉ khoảng 48×48, nếu muốn nhận diện

-the-art methods on standard

representation of a face image

tracted features in the vector

điểm mốc thì phải điều chỉnh về các kích cỡ lớn hơn mà kết quả cũng không thực sự tốt. Ngoài ra, đối

r previous methods is a con

deformation and nui

can greatly vary due to both shape

với các khuôn mặt không hướng thẳng về phía trước thì kết quả nhận dạng điểm mốc cũng không tốt

ssential components of prior d then

illumination conditions. This makes

incorporating them in

accurate shape estimation using these

sance factors such as changes in

features

(xem Hình 5.10), dẫn đến các vec-tơ đặc trưng không mang nhiều ý nghĩa gây ra nhiễu và các dự đoán

o a cascade of high capacity

reliable features to

difficult. The dilemma is that we need

sai cho bộ phân lớp. Một điểm yếu đáng kể khác liên quan đến phần chi phí là chi phí gắn nhãn cho các

gradient boosting.

other hand we need

accurately predict the shape, and on the

điểm mốc trong khuôn mặt người là không nhỏ.

, 2], that face alignment can ression

functions. In our case cascade efficiently

estimates mate and the intensities of a

lative to this initial estimate. amount of

research over the significant progress for

face 6, 18, 3, 6, 19]. In particular,

regression functions two key veral of the

successful algo e elements now.

an accurate estimate of the shape to

(a) Các điểm mốc khuôn mặt được dự đoán tốt

parameters.

extract reliable features. Previous work [4, 9, 5, 8] as well as this work, use an it erative approach (the cascade) to deal with this problem. Instead of regressing the shape parameters based on fea tures extracted in the global coordinate system of the image, the image is transformed to a normalized coordinate system based on a current estimate of the shape, and then the fea tures are extracted to predict an update vector for the shape

This process is usually repeated several times until convergence.

The second considers how to combat the difficulty of the

1

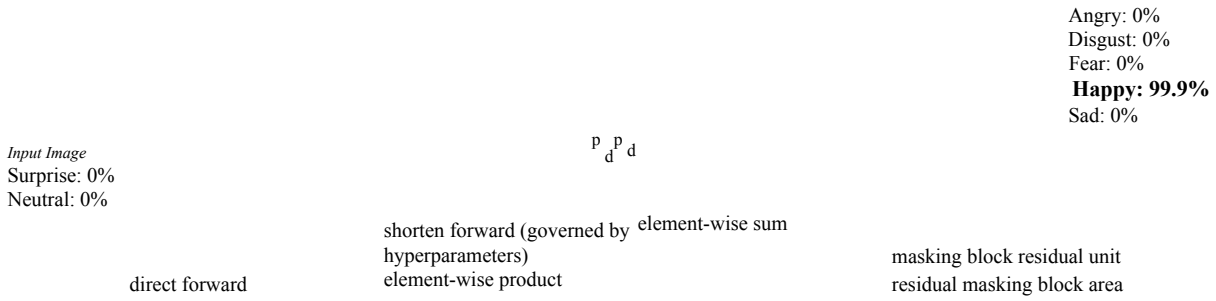
(b) Các điểm mốc khuôn mặt được dự đoán không tốt

Hình 5.10: Các điểm mốc được đánh dấu tốt và

không tốt. 58

Trong những năm gần đây, với sự phát triển của các mạng học sâu, ngoài sự phát triển về độ sâu của mô hình, như VGGNet [48], Resnet [20], Densenet [21],.. hay về độ rộng như GoogLeNet [51], Inception V3 [52],.. thì có một khía cạnh khác được phát triển không kém, đó là học có chú ý như được mô tả ở Phần 3.3.2, mà từ đó sẽ làm nền tảng cho luận văn này, phần học có chú ý sẽ được mô tả ở phần tiếp theo.

5.3 Kiến trúc mạng Residual Masking Network depth size



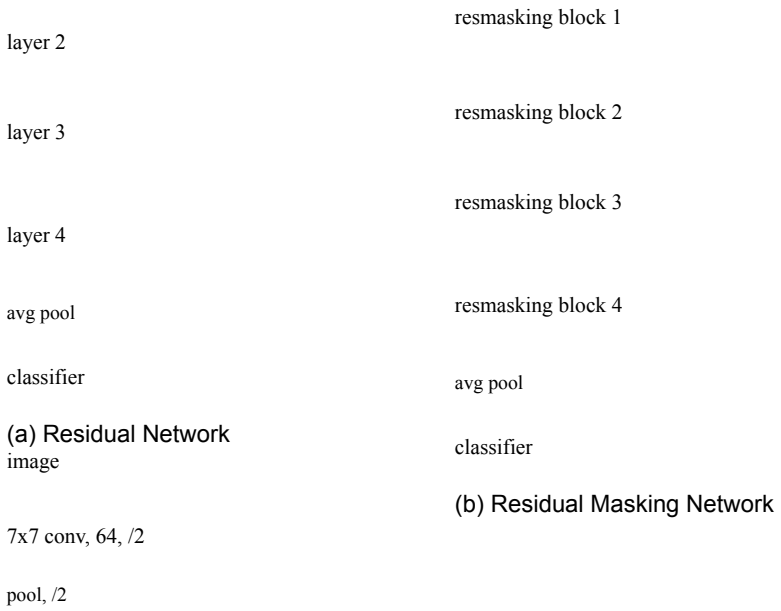
Hình 5.11: Tổng quan kiến trúc Residual Masking Network

Kiến trúc Residual Masking Network tổng quát như sau, xem Hình 5.11:

- Một khối trích xuất đặc trưng được tạo từ chồng chất các khối Residual Masking Block.
- Một lớp phân lớp bằng lớp kết nối đầy đủ và Softmax.
- Kiến trúc được huấn luyện end-to-end.

Các khối Residual Masking Block chồng chất theo số lượng được điều khiển bởi siêu tham số depth_size, trong luận văn này, depth_size sẽ được cấu mình mặc định là 4. Hình ảnh so sánh độ tương tự giữa Deep Residual Learning Network và Residual Masking Network được thể hiện ở Ảnh 5.12

Khối Residual Masking Block được tạo thành từ hai phần: một lớp trích xuất đặt trưng *T* và một khối Masking *M* để *sàng lọc* đặc trưng. Thành phần của *M* và *T* có thể được tạo nên từ các khối Learning Unit của các mạng hiện đại, trong luận văn này, tôi sử dụng Residual Unit trong mạng ResNet [20] để làm đơn vị trích xuất đặc trưng, và sử dụng tư tưởng top-down bottom-up, kiến trúc mã hóa - giải mã đã gặt hái được nhiều thành công trong các lĩnh vực khác nhau để thiết kế khối Masking Block sàn lọc đặc trưng (xem Hình 5.12). Cả kiến trúc được huấn luyện end-to-end.



Hình 5.12: So sánh độ tương tự giữa Residual Network và Residual Masking Network. Residual Masking Network cũng bao gồm các lớp Convolution, Pooling ở đầu mạng, Average Pooling và lớp Linear ở cuối mạng. Ở giữa là 4 khối trích xuất đặc trưng.

5.4 Khối học tập *T*

Khối học tập (hay là learning unit) là khối sẽ trực tiếp tham gia học tập, trích xuất đặc trưng để phân lớp. Có thể được thay thế bằng các khối học tập trong bất kỳ kiến trúc nào có thể làm được điều tương tự.

Trong luận văn này, tôi sử dụng các khối học tập cơ bản Residual Unit Block được định nghĩa trong mạng Deep Residual Network, để hình thành nên khối học tập *T*. Một khối học tập *T* cơ bản sẽ được tạo thành từ các lớp cơ bản trong mạng tích chập: Convolution, Batchnorm, ReLU. Một điểm mạnh và vượt trội của khối Residual Unit là nó có phép cộng identity giúp mô hình cải thiện hiệu quả và nổi tiếng tới bây giờ.

Tính toán đầu ra cho khối học tập *T*

Cho một khối đặc trưng đầu vào *x*, khối học tập *T* sẽ tính toán đầu ra *t* theo Công thức 5.1. Một khối học tập *T* được thể hiện ở Hình 5.15a.

$$t = T(x) \quad (5.1)$$

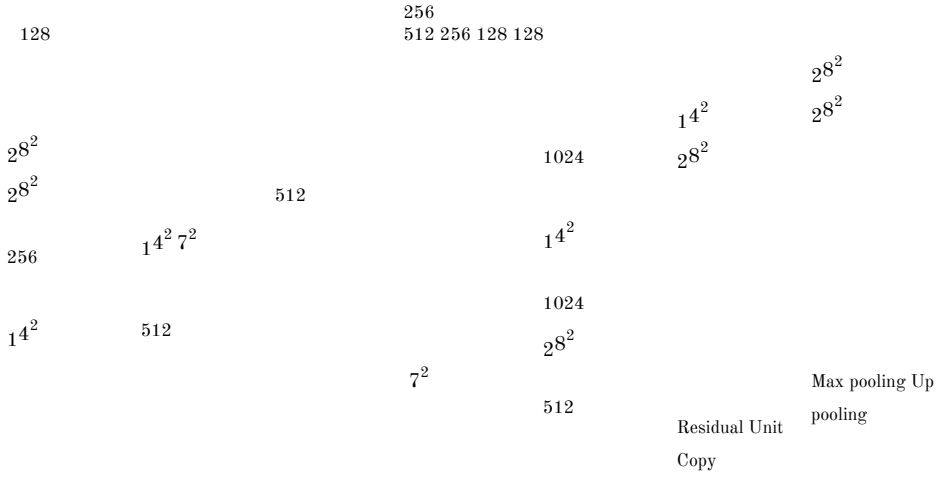
5.5 Khối Masking Block *M*

Khối Masking Block lấy cảm hứng từ cơ chế chú ý từ dưới lên - từ trên xuống (top down - bottom up feedforward attention) hay cơ chế mã hóa - giải mã đã đạt được nhiều thành công trong bài toán ước lượng tư thế con người, và bài toán phân đoạn. Cấu trúc Masking cơ bản sẽ trả về các điểm xác

suất/trọng số/mức quan trọng (soft weights) của đặc trưng ngay tại vị trí đó, xem Hình 5.13 và Hình 5.14. Bộ trọng lượng mềm này sẽ kết hợp với các đặc trưng để sàn lọc - hay tạo thành cơ chế chú ý.

Các thành phần trong khối Masking Block

Như vậy khối Masking Block bao gồm 2 thành phần cơ bản: Bộ mã hóa (Encoder) và Bộ giải mã (Decoder). Bộ Encoder, với khối đặc trưng đầu vào, sẽ lần lượt học thêm các đặc trưng ở nhiều mức độ không gian, bằng các lớp Pooling sẽ được thêm vào để nhanh chóng giảm chiều không gian của khối đặc trưng. Đồng thời các khối Convolution sẽ nhanh chóng tăng độ sâu của khối đặc trưng để tăng khả năng thể hiện tiềm ẩn - latent representation. Sau khi đạt tới độ phân giải thấp nhất (thể hiện tiềm ẩn đạt mức cao nhất), khối đặc trưng toàn cục sẽ được mở rộng ra bằng khối Decoder đối xứng với khối Encoder. Kích cỡ không gian sẽ được tăng dần thông qua lớp tích chập chuyển vị để đảm bảo đầu ra có kích cỡ tương đương với đầu vào. Sau đó, một hàm chuẩn hóa sẽ chịu trách nhiệm chuẩn hóa đầu ra nằm trong đoạn [0, 1].



Hình 5.13: Khối Masking Block với độ sâu bằng 3 (*d* = 3)

Kết nối bỏ dỡ (skip connection)

Khối Masking Block có các kết nối bỏ dở (skip connection) được lấy cảm hứng từ mạng U-net, nhằm kết hợp các đặc trưng trong khối Encoder và khối Decoder giúp cho nó tăng khả năng localisation, trong bài toán này là tăng khả năng đánh trọng số cho các đặc trưng.

Tính toán đầu ra cho khối Masking Block *M*

Cho một khối đặc trưng đầu vào *t*, khối Masking Block *M* sẽ tính toán đầu ra *m* theo Công thức 5.2, trong đó *σ* là một hàm chuẩn hóa giá trị. Hai ví dụ của khối Masking Block *M* ở độ sâu ba và một được thể hiện ở Hình 5.13 và Hình 5.14.

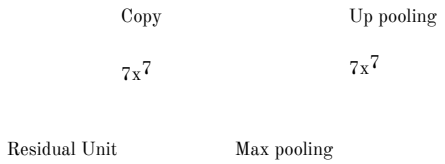
$$m = \sigma(M(t)) \quad (5.2)$$

512

512 1024 512

$7x^7$

$7x^7$



Hình 5.14: Khối Masking Block với độ sâu bằng 1 (d = 1)

Chuẩn hóa giá trị về đoạn [0 − 1]

Việc chuẩn hóa giá trị về đoạn [0 − 1] được sử dụng trong luận văn này gồm hai hàm Sigmoid và Softmax. Được lấy cảm hứng từ các thảo luận [46] và thực nghiệm trong bài toán phân đoạn ảnh. Công thức của hàm Sigmoid và hàm Softmax lần lượt được thể hiện trong Công thức 5.3 và Công thức 5.4,

$$S(x) = \frac{1}{1 + e^{-x}} \quad (5.3)$$

$$a_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \ \forall i = 1, 2, \ldots, C \quad (5.4)$$

Đối với một khối đặc trưng cho trước có kích cỡ $C \times W \times H$ hàm sigmoid nhận vào từng phần tử và chuẩn hóa giá trị của chúng về đoạn [0 − 1] mà không bị phụ thuộc hay ảnh hưởng bởi các giá trị khác, cách làm này còn được biết đến như mixed attention [57]. Đối với Softmax, một dãy giá trị chạy theo chiều kênh màu được làm đầu vào, vào đầu ra là một dãy giá trị có cùng kích cỡ có giá trị thuộc đoạn [0 − 1] và tổng của dãy phải bằng 1. Cả hai cách chuẩn hóa này sẽ được tiến hành thực nghiệm và lựa chọn dựa trên kết quả.

5.6 Kết hợp kết quả *m* và *t*

Sau khi tính toán ra kết quả m và t bằng cách sử dụng lần lượt hai khối học tập T và khối Masking Block M. Chúng ta kết hợp hai kết quả này lại bằng Công thức 5.5,

$$b_{i,c} = (1 + m_{i,c}) * t_{i,c} \quad (5.5)$$

Với i là các chỉ mục theo không gian, c là chỉ mục theo độ sâu của khối đặc trưng. Công thức được lấy cảm hứng từ thuật toán học Residual Attention như trong nghiên cứu Residual Attention Network [57] hay trong nghiên cứu Bottleneck Attention Module [42], việc cộng lại kết qủa của khối đặc trưng tránh cho việc nhân trọng số nhầm làm mất đi các đặc trưng hữu ích. Tuy vậy, tôi vẫn tiến hành thực nghiệm trên phiên bản Naive Attention - nghĩa là không có cộng thặng dư, lúc đấy khối đầu ra sẽ được tính như Công thức 5.6, để có kết luận đúng đắn về việc góp phần cải thiện hiệu quả của mô hình.

$$b_{i,c} = m_{i,c} * t_{i,c} \quad (5.6)$$

Khối Masking Block có nhiệm vụ tạo ra một mặt nạ trọng số cho khối đặc trưng $T(x)$ trước đó, việc nối đuôi khối Masking M ngay sau khối trích xuất đặc trưng T có thể có ưu điểm hơn Mạng Residual Attention Network ở chỗ trọng số được tạo ra thích ứng với chính đầu vào của nó, có thể giả định vì điều này mà nó tập trung tốt hơn.

Bảng 5.1: Cấu hình chi tiết của kiến trúc Residual Masking Network cho bài toán nhận diện biểu cảm khuôn mặt. Giữa các khối Residual Masking Block sẽ có các lớp Max Pooling để giảm kích cỡ không gian từ 56 × 56 → 28 × 28 → 14 × 14 → 7 × 7. Cái khối trích xuất đặc trưng được xây dựng có cùng số kênh ở mỗi giai đoạn giống như mạng Residual Network [20]

Kích cỡ đầu ra $C \times W \times H$
64 × 112 × 112
64 × 56 × 56
64 × 56 × 56
128 × 28 × 28
256 × 14 × 14
512 × 7 × 7
512 × 1 × 1
7

Tên lớp Cấu hình, thành phần tạo thành Conv1 7 × 7, stride 2

MaxPooling 3 × 3, stride 2 Residual Masking Block 1 (BasicBlock) * 3, Masking Block Depth 4

Residual Masking Block 2 (BasicBlock) * 4, Masking Block Depth 3 Residual Masking Block 3

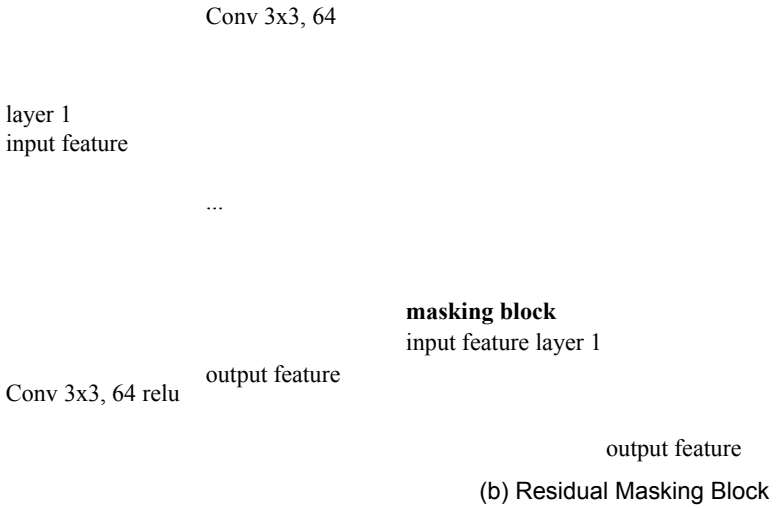
(BasicBlock) * 6, Masking Block Depth 2 Residual Masking Block 4 (BasicBlock) * 3, Masking Block

Depth 1

Average Pooling 7 × 7, stride 1 FC, Softmax

params $\times 10^6$ 142.9

trunk depth 59



Hình 5.15: So sánh Residual Layer và Residual Masking Block.

5.7 Học kết hợp với các kiến trúc khác

Một trong những hướng phát triển yêu thích của giải pháp học máy hiện nay là phương pháp học kết hợp - ensemble learning. Phương pháp này là một phương pháp tìm cách kết hợp nhiều phương pháp lại với nhau nhằm tạo ra một phương pháp có tính tối ưu hơn các phương pháp con, từ đó tăng độ chính xác chung. Riêng trong bài toán nhận diện cảm xúc khuôn mặt, các giải pháp học kết hợp hiện đang đạt kết quả cao nhất.

Vì những điểm đó, tôi thực hiện một phương pháp học kết hợp đơn giản bằng cách trung bình tổng xác suất không trọng số - Sum avarage. Kết hợp mô hình tôi đề xuất ở trên - Residual Masking Network và sáu kiến trúc khác tôi tự hiện thực và huấn luyện lại, chi tiết như sau:

- Resnet18 [20]
- Resnet50 [20]
- Resnet101 [20]
- Cbam Resnet50 [59]

- EfficientNet_b2b [53]
- ResMaskingNet không dùng drop out.
- ResMaskingNet dùng drop out.

Ngoại trừ kiến trúc Residual Masking Network, những mô hình còn lại đều được tinh chỉnh (fine tuning) lại từ bộ trọng số được huấn luyện trước trên tập dữ liệu ImageNet. Tất cả 7 mô hình đều

được huấn luyện trên tập FER2013 dưới cùng một điều kiện giống nhau.

Bước kết hợp

Vì số lượng mô hình học kết hợp là khá nhiều và điều kiện máy móc không cho phép, nên tôi kết hợp kết quả của chúng ngoại tuyến (offline). Kết quả xác suất đầu ra của các mô hình sẽ được ghi xuống một tập tin txt. Sau đó bước kết hợp sẽ được thực hiện trên tập tin text. Dãy xác suất đầu ra sẽ được trực tiếp lấy trung bình trên 7 mô hình, kết quả cuối cùng được lấy thông qua phép argmax.

Bước tìm kiếm

Bước tìm kiếm các mô hình kết hợp được thực hiện bằng giải thuật tìm kiếm tham lam với tiêu chí tốt nhất trên tập dữ liệu kiểm thử - tập validation. Cách làm này tương tự như cách tìm kiếm trong nghiên cứu của Pramerdorfer và cộng sự [44].

Chương 6

Thực nghiệm

6.1 Tổng quan về phương pháp thực nghiệm

6.1.1 Môi trường thực nghiệm

Môi trường thực nghiệm được thực hiện bằng máy tính cá nhân đối với các mạng cỡ nhỏ và thuê ở các trang web cho thuê máy chủ như vast.ai, paperspace.com, AWS, Google Cloud đối với các mạng cỡ lớn.

Cấu hình chung đối với các máy tính được dùng để huấn luyện các mô hình cơ bản như sau (cấu hình tối thiểu/cấu hình tối đa):

- Hệ điều hành Ubuntu 18.04 64 bit.
- CPU Intel$_R$ Core^{TM} i7-8750H, xung nhịp 2.20GHz × 12 nhân.
- RAM 16GB/480GB.
- GPU 1050ti/1080ti.
- GPU RAM 4GB/12GB.
- Dung lượng đĩa 100GB (HDD/SSD).

6.1.2 Tiêu chí đánh giá

Đối với bài toán này, hai phương pháp thường được dùng để đánh giá hiệu quả của mô hình là độ chính xác và ma trận bối rối.

Độ chính xác - Accuracy

Độ chính xác là một độ đo đơn giản thường được sử dụng trong các bài toán phân lớp. Nó chỉ thể hiện bằng một số nên rất dễ dàng để đánh giá. Cách tính độ chính xác được thể hiện qua Công thức 6.1,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.1)$$

Trong đó,

- True Positive (TP): Phần dữ liệu thuộc lớp A và được dự đoán thuộc lớp A. • True Negative (TN): Phần dữ liệu không thuộc lớp A và được dự đoán không thuộc lớp A. • False Positive (FP): Phần dữ liệu không thuộc lớp A và mô hình dự đoán thuộc lớp A. • False Negative (FN): Phần dữ liệu thuộc lớp A và mô hình dự đoán thuộc lớp A.

Nhược điểm của độ chính xác là nó không phản ánh toàn diện được sức mạnh của các bộ phân

lớp đối với các bộ dữ liệu mất cân bằng - một vấn đề đặc trưng của bài toán phân lớp biểu cảm. Lúc này có một phương pháp đánh giá tuyệt vời hơn, đó là Confusion Matrix.

Ma trận bối rối - Confusion Matrix

Trong lĩnh vực học máy và cụ thể là bài toán phân loại thống kê, ma trận bối rối, hay ma trận nhầm lẫn, hay ma trận lỗi, là một độ đo được bố trí ra bảng cho phép trực quan hóa hiệu suất của mô hình đối với các lớp. Mỗi hàng của ma trận thể hiện kết quả dự đoán của mô hình trong khi mỗi cột thể hiện giá trị thực tế (groundtruth) của điểm dữ liệu. Tên Confusion Matrix xuất phát từ thực tế là nó giúp dễ dàng xem liệu hệ thống có gây nhầm lẫn hai lớp hay không (nghĩa là thường gắn nhãn sai là lớp khác). Kết quả của phương pháp này là một ma trận vuông với kích thước mỗi chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng thứ *i*, cột thứ *j* là số lượng điểm lẽ ra thuộc vào class *i* nhưng lại được dự đoán là thuộc vào class *j*. Trong một số tài liệu có thể viết ngược lại, thì đó là ma trận chuyển vị của ma trận này. Ma trận bối rối chưa chuẩn hóa sẽ thể hiện số điểm dữ liệu vào mỗi ô của ma trận. Ma trận được chuẩn hóa sẽ thể hiện số phần trăm dữ liệu thuộc class *i* nhưng lại được dự đoán thuộc class *j*.

Ma trận bối rối được thể hiện ở các Phần 6.4.2 và 6.5.2

6.1.3 Khung thức thực nghiệm

Khung thức thực nghiệm được hiện thực bằng ngôn ngữ Python và framework Pytorch.

Ngôn ngữ python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Hiện nay được sử dụng rất rộng rãi trong lĩnh vực Khoa học dữ liệu và Học Máy.

Pytorch

PyTorch là một thư viện máy học mã nguồn mở dựa trên thư viện Torch, được sử dụng cho các ứng dụng như thị giác máy tính và xử lý ngôn ngữ tự nhiên. Nó chủ yếu được phát triển bởi phòng thí nghiệm AI Research của Facebook (FAIR). Đây là phần mềm miễn phí và nguồn mở được phát hành theo giấy phép BSD. Pytorch và Python đều cùng sẽ được sử dụng để phát triển khung thức thực nghiệm trong luận văn này.

Cấu trúc khung thức

Ngõ vào của khung thức là tập tin *main.py*, là nơi gọi các mô-đun cần thiết trong từng gói *models*,

utils, *trainers*,.. những thông tin/cấu hình cần thiết cho thực nghiệm nằm trong thư mục *configs*, những kết quả của quá trình thực nghiệm vào thư mục *saved*. Cấu trúc của khung thức được thể hiện qua Hình 6.1

Hình 6.1: Khung thức hiện thực thực nghiệm.

Các cấu hình cần thiết được sử dụng cho việc huấn luyện được mô tả bằng một dictionary, lưu trữ trong tập tin json, được đọc đầu tiên trong tập tin main, sau đó các mô-đun khác sẽ được lần lượt được tải lên theo yêu cầu cấu hình. Một cấu hình mẫu có thể hiện như sau:

{ "data_path": "saved/data/fer2013", "image_size": 224, "in_channels": 3, "num_classes":
7, "arch": "resattnet56", "lr": 0.0001, "weighted_loss": 0,

"momentum": 0.9, .. }

và các thông tin khác nữa, ưu điểm của việc sử dụng tập tin cấu hình là chúng ta tránh được việc hard-code hoặc quản lý quá nhiều tham số, ngoài ra còn có thể tái huấn luyện để sinh lại kết quả một cách dễ dàng.

Các tập tin định nghĩa mô hình

Các tập tin định nghĩa mô hình được hiện thực sử dụng mô-đun torch.nn và các hàm thông dụng của torch. Tất cả chúng đều được thiết kế trong một lớp con của lớp nn.Module. Cách làm này giúp các mô hình dễ dàng được thừa kế và phát triển tiếp. Một ví dụ về việc định nghĩa mô hình mạng AlexNet được thể hiện dưới đây,

```
1 class AlexNet ( nn . Module ) :
2
3 def __init__ ( self , in_channels =3 , num_classes =1000) :
4 super ( AlexNet , self ) . __init__ ()
```

```
self . features = nn . Sequential (
nn . Conv2d ( in_channels , 64 , kernel_size =11 , stride =4 , padding =2) , nn . ReLU ( inplace = True ) ,
nn . MaxPool2d ( kernel_size =3 , stride =2) ,
nn . Conv2d (64 , 192 , kernel_size =5 , padding =2) ,
nn . ReLU ( inplace = True ) ,
nn . MaxPool2d ( kernel_size =3 , stride =2) ,
nn . Conv2d (192 , 384 , kernel_size =3 , padding =1) ,
nn . ReLU ( inplace = True ) ,
nn . Conv2d (384 , 256 , kernel_size =3 , padding =1) ,
nn . ReLU ( inplace = True ) ,
nn . Conv2d (256 , 256 , kernel_size =3 , padding =1) ,
nn . ReLU ( inplace = True ) ,
nn . MaxPool2d ( kernel_size =3 , stride =2) ,
)
self . avgpool = nn . AdaptiveAvgPool2d ((6 , 6) )
self . classifier = nn . Sequential (
nn . Dropout () ,
nn . Linear (256 * 6 * 6, 4096) ,
nn . ReLU ( inplace = True ) ,
nn . Dropout () ,
nn . Linear (4096 , 4096) ,
nn . ReLU ( inplace = True ) ,
# TODO : strictly set to 1000 to load pretrained
# nn. Linear (4096 , num_classes ),
nn . Linear (4096 , 1000) ,
)

def forward ( self , x) :
```



```
x = self . features ( x)
x = self . avgpool (x )
x = torch . flatten (x , 1)
x = self . classifier (x)
return x
```

Thủ tục huấn luyện

Các thủ tục huấn luyện được hiện thực và đặt trong gói trainers, mỗi trainer cơ bản sẽ có một cách huấn luyện khác nhau. Các thông số của trainer được cài đặt từ thông tin đọc được của tập tin cấu hình. Cấu trúc của một đối tượng phục vụ thủ tục huấn luyện cơ bản như sau:

```
class FER2013Trainer ( Trainer ):
def __init__ ( self , model , train_set , val_set , test_set , configs ): pass

def train ( self ):
pass

def _train ( self ) :
pass

def _val ( self ):
pass

def _calc_acc_on_private_test ( self ):
pass

def _update_training_state ( self ) :
```

```
18 pass
19
20 def _logging ( self ):
21 pass
22
23 def _another_utils ( self ) :
24 pass
```

Ngoài ra thư mục saved để chứa các kết quả trong quá trình huấn luyện như điểm lỗi, các tập tin trực quan hóa, và các bộ trọng số. Gói utils chứa các mô-đun định nghĩa các tiện ích liên quan đến quá trình huấn luyện như các bộ optimizer tự hiện thực, các hàm mất mát tự định nghĩa, các cách xử lý dữ liệu,..

6.2 Xử lý dữ liệu

Phát hiện khuôn mặt

Đối với dữ liệu ảnh chưa được thực hiện bước phát hiện khuôn mặt thì khuôn mặt được phát hiện bởi phương pháp phát hiện khuôn mặt được cung cấp bởi OpenCV 3.4.

Các phép biến đổi làm giàu dữ liệu

Để công bằng trong việc so sánh giữa các mô hình thì tôi chỉ sử dụng 2 phép làm giàu dữ liệu trong xuyên suốt quá trình huấn luyện và so sánh hiệu năng giữa các mô hình:

- Lật ảnh theo trục dọc (Flip Left Right).
- Xoay ảnh từ -30 đến 30 độ.

Phép làm giàu dữ liệu được hiện thực bởi mô-đun imgaug đặt trong tập tin utils/augmenters/augment.py như sau

```
1 import imgaug
2 imgaug . seed (1234)
3
4 from imgaug import augmenters as iaa
5
6 seg = iaa . Sequential ([
7 iaa . Fliplr (p =0.5) ,
8 iaa . Affine ( rotate =( -30 , 30) )
9 ])
```

Lớp định nghĩa dữ liệu

Lớp định nghĩa dữ liệu được thừa kế từ lớp Dataset trong gói torch.utils.data. Đối tượng khi được khởi tạo sẽ tải toàn bộ dữ liệu lên RAM. Hàm __getitem__ được hiện thực để lấy dư liệu ra, trình tự như sau:

- Lấy ảnh và nhãn gốc từ danh sách dữ liệu.
- Sửa kích cỡ ảnh về 224×224.
- Chồng ảnh lên 3 lần tạo ra 3 kênh.
- Thực hiện làm giàu nếu như đang trong quá trình huấn luyện.
- Chuyển ảnh về Tensor.


```
class FER2013 ( Dataset ):
def __init__ ( self , stage , data , configs ):
self . _stage = stage
self . _configs = configs
self . _image_size = ( configs ['image_size '] , configs ['image_size ']) # 224 x 224 self . _data = data

self . _transform = transforms . Compose ([
transforms . ToPILImage () ,
transforms . ToTensor () ,
])

def __len__ ( self ):
return len ( self . _pixels )

def __getitem__ ( self , idx ):
image , targer = self . _data [ idx ]
image = cv2 . resize ( image , self . _image_size )
image = np . dstack ([ image ] * 3)

if self . _stage == 'train ':
image = seg ( image = image ) # augment

image = self . _transform ( image )
return image , target
```

6.3 Cài đặt huấn luyện

Để đảm bảo tính công bằng cho các kiến trúc và cấu hình mạng khác nhau, cấu hình của quá trình huấn được cài đặt cứng đối với các cài đặt cơ bản. Từ đó có thể dễ dàng tái sao chép lại quá trình cũng như kết quả trước đó. Nội dung của các cấu hình cơ bản được dùng chung các tất cả các kiến trúc như sau:

- Cỡ ảnh 224×224×3.
- Số channels của đầu vào ảnh: 3
- Số lớp của bộ phân lớp sau cùng: 7

• Tốc độ học (learning rate): 0.0001

• Mometum: 0.9

• Weight decay: 0.001

• Batch size: 48

• Num workers: 8

• Số lượng epoch tối đa: 50

• Cứ hơn 8 epoch liên tiếp không giảm được validation accuracy thì ngưng huấn luyện. • Cứ hơn 2 epoch liên tiếp không giảm được validation accuracy thì giảm tốc độ học xuống 10 lần. • Nhân ngẫu nhiên là 1234 cho tất cả các mô-đun cần bộ sinh số ngẫu nhiên.

• Mô hình được chọn là mô hình có validation accuracy tốt nhất.

6.4 Đánh giá kết quả trên tập dữ liệu FER2013

6.4.1 So sánh kết quả với các mạng hiện đại

Để mang tính công bằng trong việc so sánh hiệu suất của kiến trúc được đề xuất, tôi hiện thực và huấn luyện lại các kiến trúc mạng phân lớp hiện đại dưới cùng một cấu hình và khung thức. Các mô hình được huấn luyện lại được trình bày trong phần phụ lục. Kết quả được thể hiện ở Bảng 6.1,

Bảng 6.1: Kết quả so sánh với các mạng hiện đại được huấn luyện lại dưới cùng cấu hình trên tập FER2013

Tên Kiến trúc/Mạng	params×10^6	Độ chính xác (%)
VGG19 [44]	139.5	70.8
Efficientnet_b2b [53]	7.7	70.8
Googlenet [51]	5.6	71.97
Resnext50_32x4d [60]	25.0	72.22
Inception_v3 [52]	25.1	72.72
Resnet18 [20]	11.2	72.9
Resnet50_pretrained_vgg_face [20]	23.5	72.91
Densenet121 [21]	6.9	73.16
Resnet152 [20]	58.1	73.22
Cbam_resnet50 [59]	28.5	73.39
Bam_resnet50 [42]	23.8	73.14

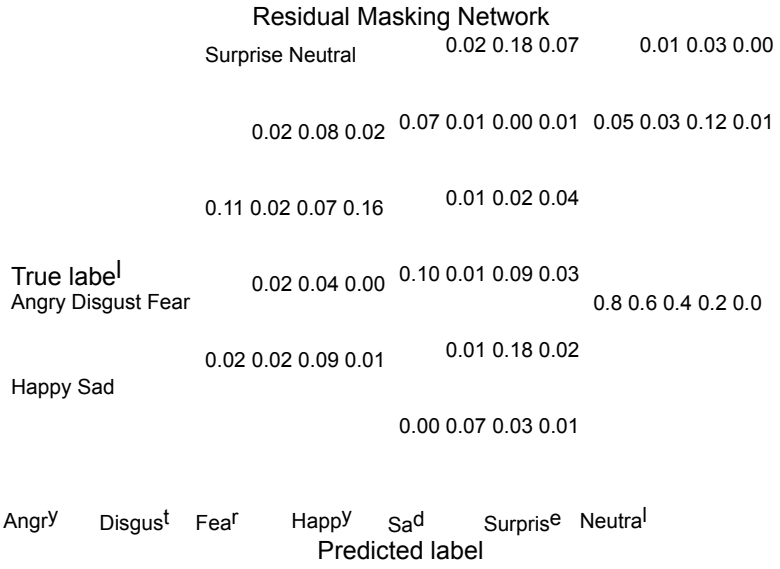
(Our) ResMaskingNet	142.9	74.14

Bảng 6.2: So sánh với các kết quả được báo cáo khoa học

Tên Kiến trúc/Mạng	Độ chính xác (%)
Human Accuracy [18]	65 ±5
Deep-Emotion [36]	70.8
DL-SVM [54]	71.16
CNN–SIFT [5]	73.4
Ensemble MLCNNs [39]	74.09
Ensemble 8 CNNs [44]	75.2
CNNs and BOVW + local SVM [17]	75.42
(Our) ResMaskingNet	74.14
(Our) ResMaskingNet + 6 CNNs	76.82

6.4.2 Ma trận bối rối của các mô hình

Trong phần này sẽ hiển thị các ma trận bối rối của các mô hình và mạng Residual Masking Network trên tập dữ liệu FER2013.



Hình 6.2: Ma trận bối rối của mạng Residual Masking Network trên tập dữ liệu FER2013

6.5 Đánh giá kết quả trên tập dữ liệu VEMO

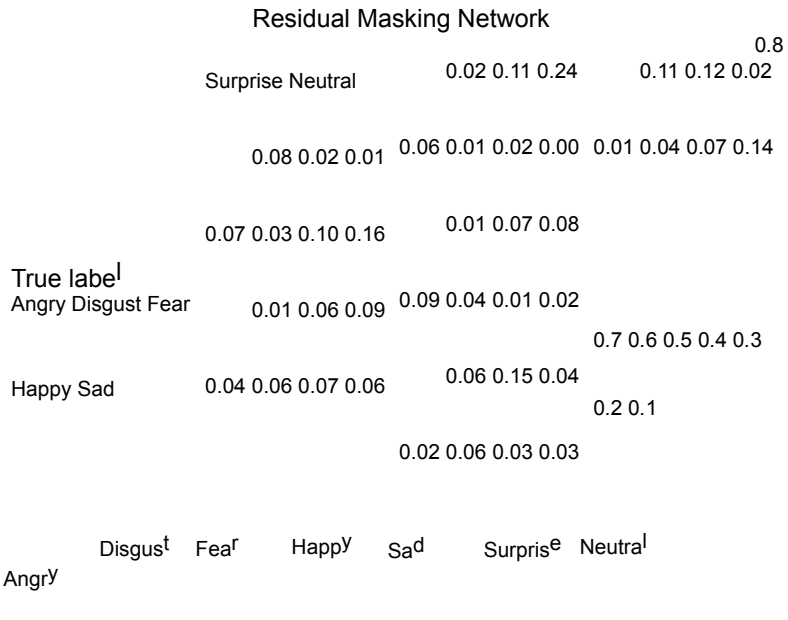
6.5.1 So sánh kết quả với các mạng hiện đại

Bảng 6.3: Kết quả so sánh với các mạng hiện đại được huấn luyện lại dưới cùng cấu hình trên tập VEMO

Tên Kiến trúc/Mạng	Độ chính xác (%)
Resnet18	63.94
Resnet34	64.84
ResAttNet56	60.82
ResMaskingNet	65.949

6.5.2 Ma trận bối rối của Residual Masking Network

Ma trận bối rối của Mạng Residual Masking Network trên tập kiểm tra của tập dữ liệu VEMO



Hình 6.3: Ma trận bối rối của mạng Residual Masking Network trên tập dữ liệu VEMO 75

6.6 Một số hình trực quan từ mạng học sâu

Mạng học sâu, hay cụ thể hơn khối trích xuất đặc trưng bằng mạng tích chập hiện đại ngày nay được nổi tiếng biết đến như một hộp đen (blackbox) vì chúng ta không thể diễn giải được ý nghĩa của tất cả các con số (trọng số). Điều này rất khó làm quen so với tất cả những giải thuật trước đây trong lịch sử của khoa học máy tính, tất cả mọi con số đều mang một ý nghĩa cho riêng nó góp phần vào việc tính toán và hoàn thiện giải thuật.

Vấn đề giải thích, hay debug một mạng học sâu, hay gọi là *hiểu* tại sao lại có những dự đoán tốt hay không tốt dẫn đến một nhu cầu trực quan hóa những lớp dữ liệu chảy trong khối trích xuất đặc trưng.

Cùng với sự phát triển của các mạng học sâu, các phương pháp trực quan hóa lần lượt ra đời theo. Trong nghiên cứu này, tôi dùng phương pháp Gradient-weighted Class Activation Mapping (GradCAM) và phương pháp gộp trung bình các khối đặc trưng để suy diễn những gì mà mạng học sâu học được.

6.6.1 Trực quan bằng GradCAM

Gradient-weighted Class Activation Mapping (Grad-CAM) là một phương pháp nổi tiếng và được sử dụng rộng rãi để trực quan hóa các đặc trưng của mạng học sâu. Grad-CAM sử dụng đạo hàm riêng của từng lớp chảy về lớp tích chập cuối cùng của mạng để từ đó sản sinh ra một lớp ánh xạ đánh dấu tầm quan trọng của một khu vực lên ảnh góp phần vào kết quả dự đoán.

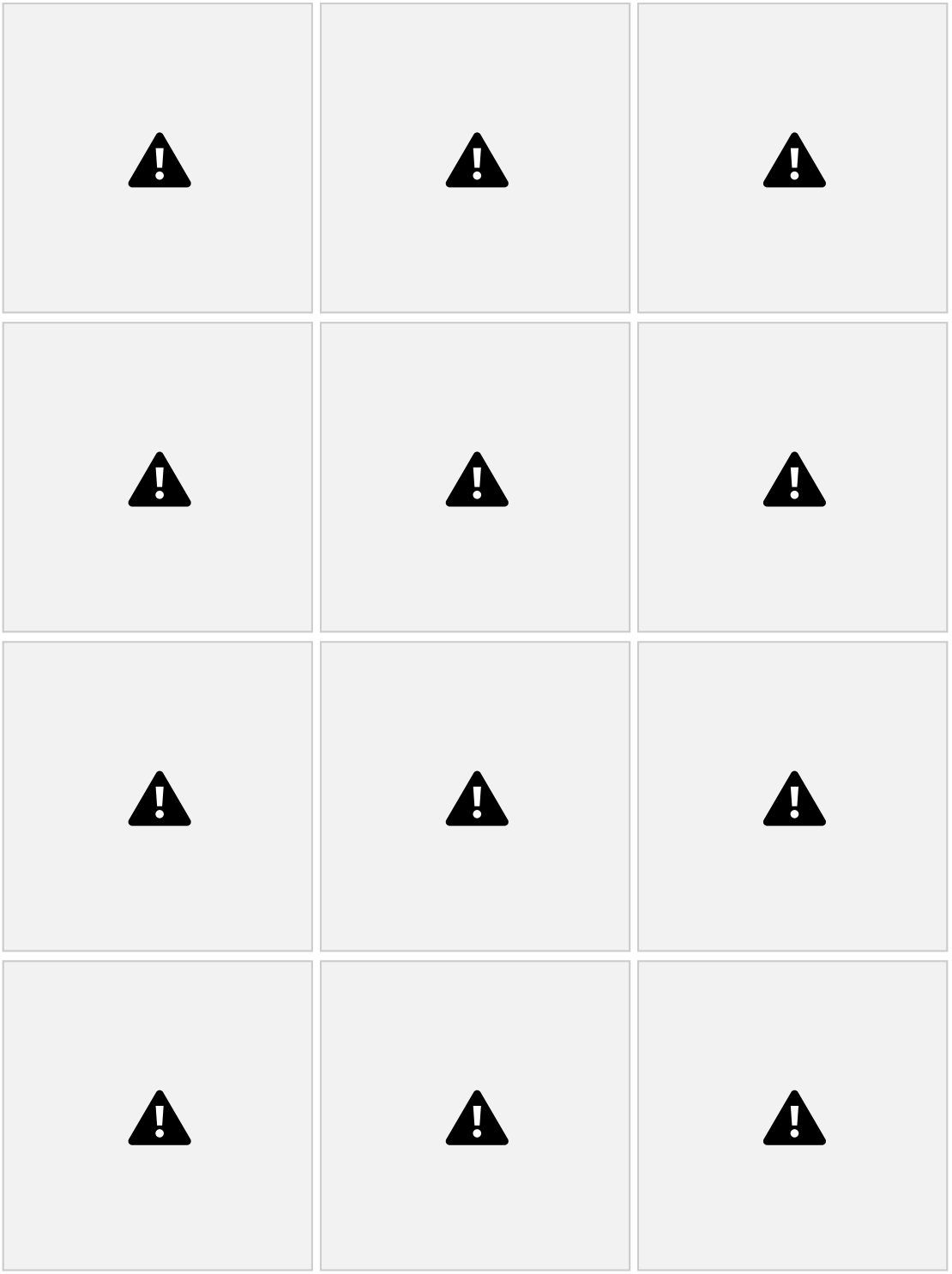
Đơn giản hơn, chúng ta lấy khối đặc trưng cuối cùng của lớp tích chập, sau đó đánh trọng số tất cả các kênh của khối đặc trưng ấy với đạo hàm riêng được trả về cho từng kênh. Điều hay của phương pháp này là nó không cần thiết phải huấn luyện lại mô hình hay thay đổi kiến trúc mạng.

Chúng ta xem qua Công thức 6.2 sau:

$$S^c = \frac{1}{Z} X_i X_j X_k \frac{\partial y^c}{\partial A^k_{ij} A^k_{ij}} \quad (6.2)$$

Điểm S^c là kết quả của phép Gộp trung bình xuyên qua hai chiều không gian i và j với đạo hàm riêng của kết quả dự đoán đối với đặc trưng A^k_{ij}. Sau đó chúng ta thực hiện nhân kết quả có được với giá trị của đặc trưng trong kênh k. Cuối cùng, phép Gộp trung bình được thực hiện xuyên qua chiều kênh k. Do đó, đầu ra là một ánh xạ trọng số kích cỡ $i \times j \times 1$.

Một số hình ảnh trực quan bằng phương pháp này được thể hiện tại Hình 6.4



Hình 6.4: Một số hình ảnh trực quan bằng GradCAM

6.6.2 Trực quan bằng gộp theo chiều kênh

Phương pháp này được thực hiện bằng cách lấy Gộp trung bình khối đặc trưng sau lớp tích chập theo chiều kênh mà không quan tâm tới kết quả dự đoán, cũng không nhân trọng số. Thứ tự của ba hình

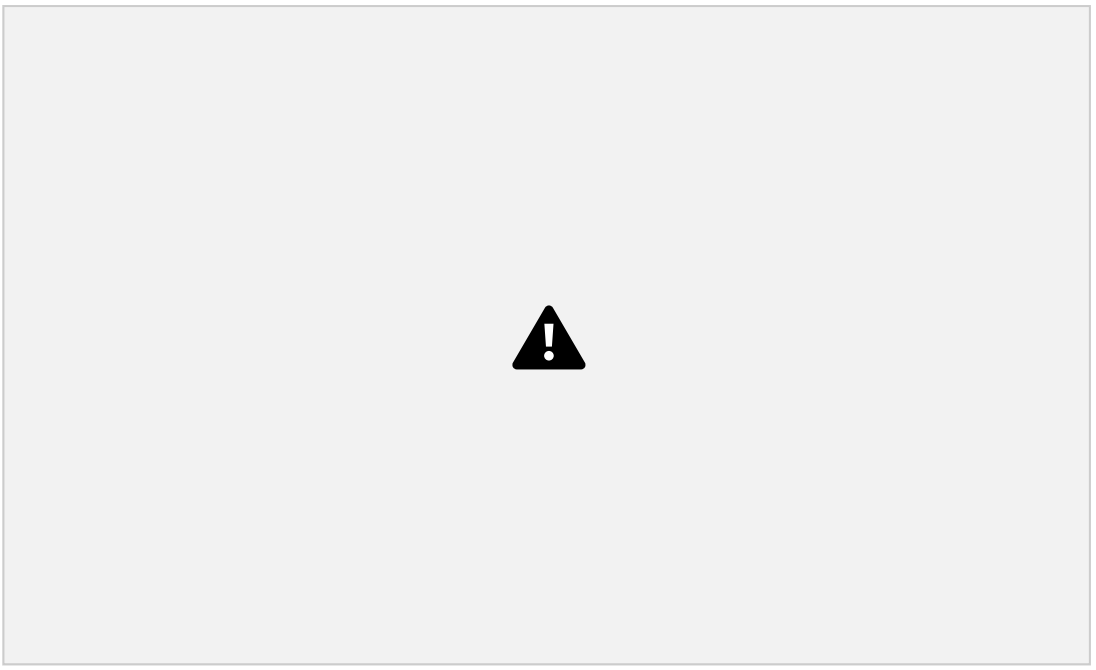
được thể hiện dưới đây là Ảnh gốc → Ảnh của activations trước khi kết hợp với khối Masking → Ảnh của activations sau khi kết hợp với khối Masking. Mặc dù có sự thay đổi giữa hai khối đặc trưng trước và sau khi kết hợp với khối Masking, đã kiểm chứng bằng phép trừ và lấy tuyệt đối, nhưng khi thực hiện gộp, chúng không thể hiện rõ ràng mà chúng ta có thể thấy được.

Tuy nhiên ở đây chúng ta có thể kết luận là chúng đã tập trung vào vùng mặt người, có chú ý vào các bộ phận có ảnh hưởng trực tiếp đến cảm xúc như được miêu tả trong FACS. Một số hình ảnh trực quan bằng phương pháp gộp theo chiều kênh được thể hiện ở Hình 6.5

Hình 6.5: Một số hình ảnh trực quan bằng cách thực hiện Avarage Pooling dọc theo chiều channel của các activations trong khối Residual Masking Block thứ 3. Thứ tự của từng ảnh là Ảnh gốc → Ảnh của activations trước khi kết hợp với khối Masking → Ảnh của activations sau khi kết hợp với khối Masking

6.7 Một số kết quả dự đoán trong thực tế

Ngoài các hình ảnh trong các tập dữ liệu nghiên cứu thì tôi cũng thực hiện chạy thử mô hình trên một số hình ảnh thực tế để kiểm chứng hiệu quả của mô hình, chúng được thể hiện qua các Hình 6.6, 6.7, 6.8, 6.9, 6.10, 6.12.



Hình 6.6: Hình ảnh hai nhân vật Dũng và Hà Lan đang chạy xe Honda với cảm giác hạnh phúc trong phim Mắt Biếc.