Các công trình nghiên cứu, phát triển và ứng dụng

Công nghệ Thông tin và Truyền thông

https://www.ictmag.vn/cntt-tt

Tập 2024, Số 1, Tháng 6

Số đặc biệt CITA 2024

MỤC LỤC / TABLE OF CONTENTS

Hệ thống sonar chủ động sử dụng đồng thời nhiều đầu thu phát tín hiệu sóng âm / Active sonar systems using multiple acoustic transceivers ..... ***Nguyễn Văn Đức, Nguyễn Quốc Khương*** 1

Pathway-Capsule Neural Networks ..... ***Phuong-Nam Nguyen*** 9

Multi-task learning model for detecting and filtering internet violent images for children / Mô hình đa tác vụ nhận diện và lọc hình ảnh bạo lực cho trẻ em ..... ***Le Kim Hoang Trung, Nguyen Van Thanh Vinh, Phan Le Viet Hung, and Nguyen Huu Nhat Minh*** 18

Omni-Dimensional Adaptation for MobileNetV3 using Bayesian Hyperparameter Tuning / Sử dụng tối ưu Bayes để tối ưu siêu tham số cho mô hình Omni-Dimensional MobiletNetV3 ..... ***Vo Long, Phan Nhat Quang, Tran Van Dat, Dang Duc Long*** 25

A Lightweight Model to Skin Disease Recognition / Một mô hình nhẹ để nhận biết bệnh về da ..... ***To Huu Nguyen, Ma T. Hong Thu, Do Thanh Mai, Trang Phung T. Thu, An Dang, Duc-Quang Vu*** 33

Research and experiment with some activation functions in sleep stage classification using deep learning methods / Nghiên cứu và thử nghiệm một số hàm kích hoạt trong phân lớp giai đoạn giấc ngủ sử dụng các phương pháp học sâu ..... ***Tan Thuy Trang, Nguyen Thi Hong, Duong Phi Long*** 40

DaNangVMD: Vietnamese Speech Mispronunciation Detection / DaNangVMD: Nhận diện phát âm sai tiếng Việt ..... ***Nguyen Ket Doan, Tran Nguyen Anh, Vo Van Nam, Nguyen Tran Tien, Le Pham Tuyen, Nguyen Quoc Vuong and Nguyen Huu Nhat Minh*** 49

Hệ thống sonar chủ động sử dụng đồng thời nhiều đầu thu phát tín hiệu sóng âm

Nguyễn Văn Đức, Nguyễn Quốc Khương
Trường Điện Điện Tử, Đại Học Bách Khoa Hà Nội, Hà Nội

Tác giả liên hệ: Nguyễn Văn Đức, duc.nguyenvan1@hust.edu.vn
Ngày nhận bài: 21/11/2023, ngày sửa chữa: 24/4/2024, ngày duyệt đăng: 29/5/2024
Định danh DOI: 10.32913/mic-ict-research-vn.v2024.n1.1244

Tóm tắt: Bài báo đề xuất phương pháp thiết kế hệ thống quét tín hiệu siêu âm tạo ảnh 2D (gọi là ảnh thể hiện ở mặt phẳng 2 chiều), hoặc ảnh 3D (gọi là ảnh thể hiện trong không gian 3 chiều) của các vật thể dưới nước sử dụng công nghệ nhận dạng và định vị sóng siêu âm sonar và phương pháp xử lý tín hiệu siêu âm tại đầu thu được áp dụng cho hệ thống siêu âm sử dụng nhiều đầu thu tín hiệu sóng âm (hệ ăng ten mảng) đảm bảo độ phức tạp tính toán cũng như cấu hình phần cứng giảm, nhờ đó có thể thiết kế hệ định vị sóng âm và dựng ảnh 3D trên nền tảng các hệ thống nhúng hiện nay. Thuật toán đề xuất trong bài báo đã được ứng dụng thiết kế thành công hệ sonar chủ động và có kết quả quét sóng âm chính xác tương đương so với các thiết bị sonar chuyên dụng được kiểm định bởi đơn vị đo lường chuyên ngành. Bài báo trình bày các kết quả thử nghiệm và hiệu chuẩn được thực hiện ở một số vùng biển Việt Nam như vịnh Nha Trang, v.v.

Từ khóa: *định vị sóng âm - sonar, hệ ăng ten mảng*

Title: **Active sonar systems using multiple acoustic transceivers**

Abstract: This article proposes a method to design an active sonar system using multiple transceivers for scanning underwater objects. This method converts the multiple independently received signals in the spatial domain to the multiple received signals in the frequency domain. Thus, the computational complexity of the system is significantly reduced. Therefore, it is possible to implement an active sonar system based on embedded systems. The proposed method has been successfully applied to a real sonar testbed and has been tested in real conditions, such as in Halong and Nhatrang Bay. The implemented active sonar system offers scanned results comparable to specialized sonar devices.

Keywords: *active sonar systems, antenna array.*

I. GIỚI THIỆU

Công nghệ định vị bằng sóng âm SONAR (Sound Navigation And Ranging) là một công nghệ sử dụng sự lan truyền sóng âm trong môi trường nước để xác định và định vị các vật cản trong môi trường nước, hoặc đáy nước. Công nghệ định vị bằng sóng âm sonar được phân loại thành hai loại công nghệ cơ bản:

- Công nghệ sonar chủ động có hệ thống tự phát xung sóng và nghe tiếng vọng lại [1–3].

- Công nghệ sonar bị động (hay sonar thụ động) có hệ thống chỉ nghe âm thanh do tàu bè hay nguồn âm khác phát ra; tầm hoạt động nhỏ hơn so với sonar chủ động và ứng dụng cho do thám ngầm dưới biển [4]. Các công nghệ sonar chủ động và thụ động đều được thiết kế ứng dụng cho điều khiển lái các tàu ngầm và tàu tự hành dưới nước AUV (Autonomous Underwater Vehicles) [5, 6].

Bài báo này trình bày phương pháp xử lý tín hiệu sonar sử dụng công nghệ ăng ten mảng và xử lý tín hiệu song song, đồng thời đề xuất phương pháp chuyển phổ tín hiệu thu tách biệt trên miền không gian sang các phổ tín hiệu tách biệt trên miền tần số, sau đó xử lý tín hiệu đồng thời trên miền thời gian để giảm độ phức tạp của hệ thống. Phương pháp này ưu việt hơn so với các phương pháp hiện tại với các đánh giá như sau.

Đánh giá tổng quan tình trạng công nghệ hiện nay: Sáng chế US3500302 [7], "Sonar bathymetry system transmit-receive sequence programmer" của Hoa Kỳ, công bố ngày 13/01/1969 đưa ra phương pháp xác định độ sâu đáy nước và thực hiện trên mạch cứng, không đề cập đến phương pháp xử lý số liệu từ các đầu thu tín hiệu sóng âm song song, cũng như cách trộn tần tín hiệu và ghép tín hiệu trên miền thời gian để giảm chi phí phần cứng như mô tả của thuật toán đề xuất trong mục II.

Sáng chế US9268020 [8], "Sonar assembly for reduced interference" của Hoa Kỳ công bố ngày 10/02/2012 trình bày kiến trúc hệ thống và phương pháp xử lý tín hiệu siêu âm, tuy nhiên điểm hạn chế của sáng chế là không đề cập đến phương pháp ghép các mảng ăng ten thu (đầu thu tín hiệu sóng âm) để giảm độ phức tạp phần cứng và xử lý tín hiệu song song.

Sáng chế CA2922781A1 [9], "Methods and apparatuses for reconstructing a 3D sonar image", của Canada, Hoa Kỳ và cộng đồng các quốc gia Châu Âu, trình bày phương pháp tạo ảnh Siêu âm 3D, tuy nhiên sáng chế không đề cập đến phương pháp trộn tần để có thể xử lý tín hiệu song song.

Sáng chế US8305840 [10], "Downscan imaging sonar", của Hoa Kỳ công bố ngày 6.11.2012 trình bày phương pháp tạo ảnh siêu âm 2D, tuy nhiên tương tự như sáng chế CA2922781A1, bản mô tả sáng chế US8305840 không đề cập đến việc phương pháp xử lý tín hiệu song song và giảm thiểu độ phức tạp tính toán bằng kỹ thuật trộn tần và ghép tín hiệu.

Sáng chế US9019795 [11], "Method of object tracking using sonar imaging" của Hoa Kỳ công bố ngày 05/09/2012 trình bày phương pháp bám đuổi và dò tìm vật thể dưới nước. Sáng chế không đưa ra phương pháp xử lý tín hiệu từ các đầu thu tín hiệu sóng âm song song.

Sáng chế US20150369908A1 [12], "Integrated sonar devices and methods" của Hoa Kỳ công bố ngày 24/12/2015 trình bày phương pháp tạo ảnh siêu âm 2D, tuy nhiên sáng chế không đề cập đến việc trộn tần để có thể ghép nhiều đầu thu tín hiệu sóng âm song song thành một được dữ liệu nối tiếp, cho phép xử lý đồng thời nhiều đầu thu tín hiệu sóng âm sử dụng chung một bộ biến đổi A/D. Tình trạng kỹ thuật đã biết còn có nhược điểm là chưa đưa ra phương pháp trộn tần để có thể ghép nhiều đầu thu tín hiệu sóng âm song song thành một được dữ liệu nối tiếp, cho phép xử lý đồng thời nhiều đầu thu tín hiệu sóng âm sử dụng chung một bộ biến đổi A/D và chưa đưa ra phương pháp xử lý tín hiệu từ các đầu thu tín hiệu sóng âm song song để tạo ra các ảnh quét có độ phân giải cao.

Công bố khoa học của nhóm tác giả [13] trình bày phương pháp ước lượng góc sóng tới DOA cho các hệ thống sonar chủ động, tuy nhiên chưa đề cập đến việc sử dụng các hệ thống đa ăng ten thu phát, cách giảm thiểu độ phức tạp hệ thống, cũng như hiệu năng hệ thống trong thực tế.

Công bố khoa học của nhóm tác giả [14] trình bày các kết quả đánh giá độ mở búp sóng quét của hệ sonar quét sườn, với các kịch bản, thiết kế hệ thống là các kết quả được trình bày trong bài báo này. Công trình khoa học [15] trình bày ứng dụng của học máy sâu trong việc cải thiện các bức ảnh về hệ sonar quét sườn.

Mục đích của phương pháp đề xuất trong bài báo

Mục đích của nghiên cứu này là nhằm khắc phục nhược điểm đã biết nêu trên bằng cách đề xuất phương pháp thiết kế hệ thống siêu âm quét ảnh 3D chủ động sử dụng hệ thống mảng ăng ten phát và mảng ăng ten thu được sắp xếp miền không gian và xử lý tín hiệu trên miền tần số và thời gian nhằm đạt được chất lượng ảnh quét 3D chất lượng cao, đồng thời giảm đáng kể chi phí và độ phức tạp phần cứng của hệ thống.

Bài báo đề xuất phương pháp sử dụng nhiều đầu thu tín hiệu sóng âm với mục đích để thu được nhiều tín hiệu phản xạ từ các nguồn phản xạ khác nhau, sau đó sử dụng kỹ thuật trộn tần để dịch khoảng tần số của các nguồn tín hiệu không bị chồng lấn phổ tín hiệu, tiếp theo đó cộng dồn các tín hiệu để tạo thành một hoặc nhiều đường tín hiệu song song. Kỹ thuật này cho phép sử dụng số lượng đầu thu hiệu quả và giảm số bộ chuyển đổi A/D, do đó giảm thiểu chi phí phần cứng, đồng thời tăng độ chính xác xử lý dữ liệu thu và do vậy tăng độ chính xác bức ảnh quét sóng siêu âm.

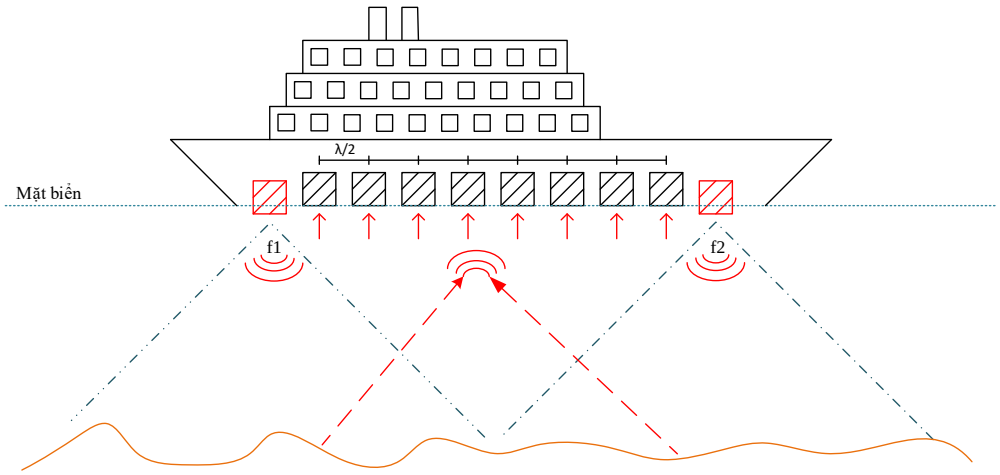
Điểm mới của phương pháp xử lý tín hiệu đề xuất trong bài báo là sử dụng nhiều đầu thu và phát với khoảng cách giữa các đầu thu và các đầu phát tín hiệu sóng âm phù hợp để có nhiều nguồn tín hiệu thu tín hiệu phản xạ cho việc tạo ảnh 3D có độ phân giải lớn. Tiếp theo, bài báo đề xuất là cách xử lý tín hiệu song song của đầu thu bằng phương pháp nhân tín hiệu thu được từ mỗi đầu thu với một thành phần tần số với mục đích dịch tần tín hiệu thu sang một khoảng tần số mong muốn. Nhờ đó mà các tín hiệu thu không bị chồng lấn phổ tại phía thu và bộ xử lý song song tại phía thu sẽ xử lý tín hiệu độc lập để tạo ra nguồn tín hiệu chính xác cung cấp cho việc tạo ảnh quét 3D của vật cản quét được bằng sóng âm dưới nước. Tại phía thu, thống sau khi đã nhân tín hiệu thu với các thành phần tần số khác nhau lại được cộng lại thành các nhóm tín hiệu trên miền thời gian. Các nhóm tín hiệu này được chuyển thành tín hiệu số qua bộ chuyển đổi A/D và được xử lý độc lập để tạo ảnh các bức ảnh quét 3D.

Phần còn lại của bài báo này được trình bày như sau: Mục II trình bày chi tiết về phương pháp thiết kế hệ thống. Mục III thảo luận về kết quả đo đạc, thực nghiệm. Mục IV kết luận bài báo.

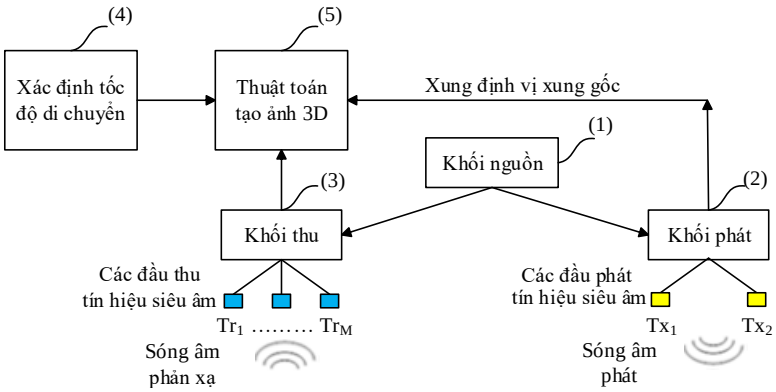
II. ĐỀ XUẤT PHƯƠNG PHÁP THIẾT KẾ HỆ THỐNG SONAR CHỦ ĐỘNG SỬ DỤNG NHIỀU ĐẦU THU PHÁT SÓNG ÂM

Bài báo đề xuất phương pháp thiết kế hệ thống quét ảnh các vật thể dưới nước bằng công nghệ sóng siêu âm chủ động và phương pháp xử lý tín hiệu phản xạ sóng siêu âm được mô tả như sau:

Hình 1 trình bày phương pháp sắp xếp các chấn tử đầu phát tín hiệu sóng âm và thu tín hiệu sóng âm cho hệ thống



Hình 1. Đề xuất sơ đồ bố trí các ăng ten thu phát sóng siêu âm sử dụng cho hệ sonar chủ động quét ảnh 2D hoặc 3D



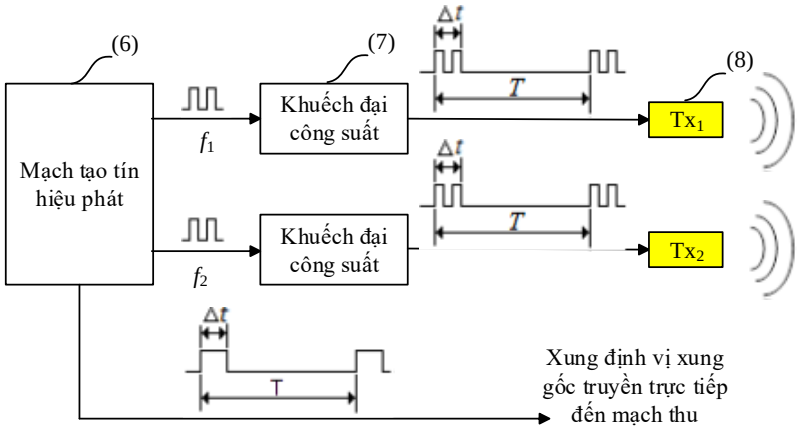
Hình 2. Đề xuất sơ đồ khối tổng quát hệ thống sonar chủ động

siêu âm chủ động 3D, trong đó hai chấn tử ở đầu hai bên mạn tàu phát tín hiệu sóng âm ở hai tần số và để tạo hai búp sóng của hai nửa mặt phẳng của một lát cắt cho một bức ảnh 2D. Việc quét liên tục các bức ảnh 2D theo hướng tàu chạy sẽ cho hình ảnh vật thể 3D ở dưới nước. Các đầu thu tín hiệu sóng âm (ăng ten thu) được ghép sau cho có thể thu được nhiều nhất tín hiệu phản xạ.

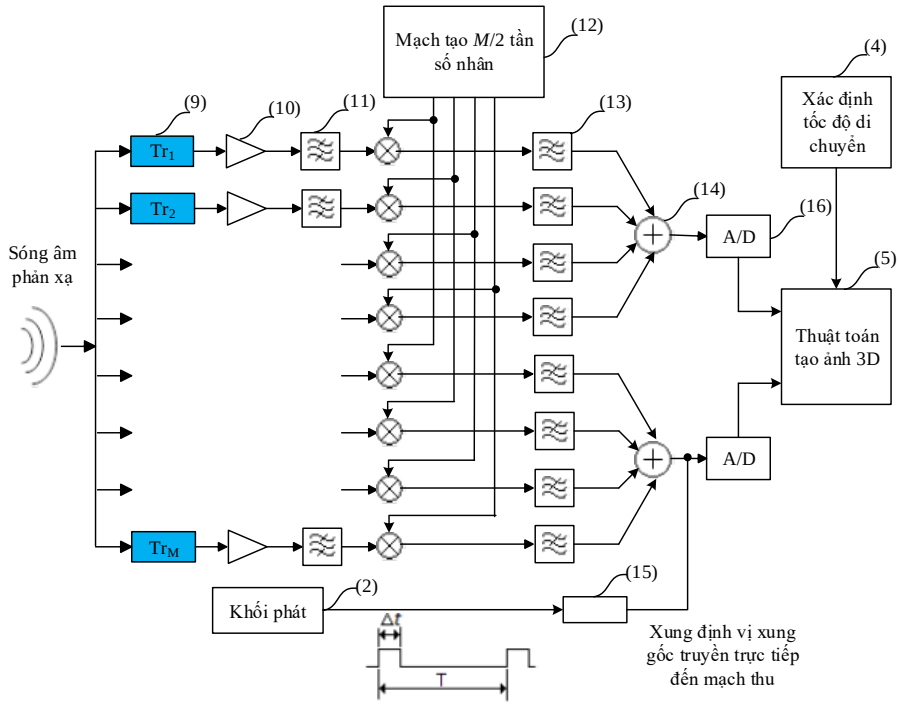
Hình 2 trình bày sơ đồ thiết kế hệ thống quét ảnh siêu âm 2D, bao gồm khối nguồn phát tín hiệu (1), khối phát tín hiệu sóng âm (2) và khối thu và xử lý tín hiệu (3). Thuật toán tạo ảnh 3D (5) nhận tín hiệu thu từ khối thu (3), cùng thông tin về vận tốc chuyển động của tàu gắn hệ thống quét siêu âm để tạo ra ma trận điểm ảnh 3D. Thông tin về vận tốc của tàu được xác định bằng khối (4).

Hình 3 trình bày sơ đồ khối của khối phát tín hiệu, bao gồm mạch tạo tín hiệu phát (6), bộ khuếch đại công suất (7) và bộ phát tín hiệu thủy âm (đầu phát tín hiệu sóng âm) (8). Mạch tạo tín hiệu phát sẽ tạo ra các xung quét siêu âm với chu kỳ lặp lại là T và bề rộng xung là Δt. Tại khung thời gian có xung siêu âm, hệ thống sẽ phát đi tín hiệu có dao động ở tần số siêu âm với tần số f_1 cho đầu phát tín hiệu sóng âm Tx_1 và f_2 cho đầu phát tín hiệu sóng âm Tx_2. Các xung này được gọi là các xung gốc siêu âm $S_1(t)$ và $S_2(t)$. Khối phát tín hiệu cũng đồng thời tạo 1 xung gốc $g(t)$ được sử dụng để xác định vị trí xung gốc trong quá trình giải mã tín hiệu ở phía thu. Xung gốc $g(t)$ có bề rộng cũng là Δt và lặp lại với chu kỳ T. Tuy nhiên xung gốc $g(t)$ không được nhân với thành phần tần số dao động do được dùng để truyền trực tiếp đến khối thu phục vụ cho việc giải mã tín hiệu phản xạ.

Hình 4 trình bày sơ đồ khối của khối thu và xử lý tín hiệu siêu âm, bao gồm bộ xác định tốc độ di chuyển tàu (4), thuật toán tạo ảnh 3D (5), bộ thu tín hiệu thủy âm (9), bộ khuếch đại (10), bộ lọc thông dải trước mạch nhân (11) , mạch tạo đa tần số nhân (12), mạch lọc thông dải sao bộ nhân trộn tần (13), bộ cộng tín hiệu (14), bộ phối



Hình 3. Đề xuất sơ đồ khối hệ phát tín hiệu sonar.



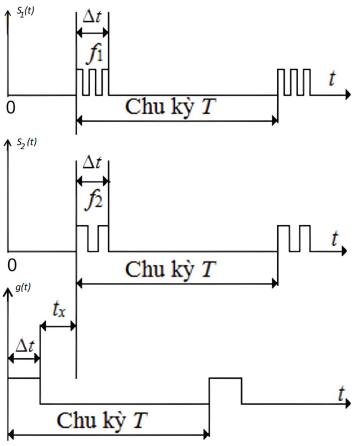
Hình 4. Đề xuất kiến trúc hệ sonar chủ động.

hợp trở kháng (15), bộ biến đổi A/D (16). Ở phía thu, hệ thống thu bao gồm M mạch thu chia thành 2 nhóm, mỗi nhóm gồm $M/2$ ăng ten thu (đầu thu tín hiệu sóng âm). Tín hiệu thu được từ các ăng ten thu sẽ được khuếch đại và lọc nhiễu, sau đó được nhân với một trong $M/2$ tần số $(f_{x,1}, f_{x,2}, \ldots, f_{x,M/2})$. Các tần số này được tạo bởi mạch tạo tần số nhân (khối (12) trong Hình 4). Tín hiệu sau bộ nhân được lọc thông thấp và được cộng vào theo từng nhóm $M/2$ đầu thu tín hiệu sóng âm để thành một tín hiệu đầu ra đưa đến các bộ A/D trước khi gửi đến bộ tạo ảnh 3D. Hai tín hiệu đầu ra từ 2 nhóm ăng ten thu sẽ được chuyển đổi A/D rồi đưa vào bộ tạo ảnh 3D (5).

Hình 5 trình bày giản đồ thời gian của tín hiệu phát, bao gồm xung phát siêu âm $S_1(t)$ với bề rộng là Δt, chu kỳ lặp lại xung là T, tần số dao động là f_1 . Tương tự như vậy $S_2(t)$ là xung gốc phát siêu âm, với bề rộng là Δt, chu kỳ lặp lại xung là T, với tần số dao động là f_2. Xung gốc $S(t)$ không được điều chế với tần số siêu âm. Xung gốc sonar bản chất không thể phát đi xa được mà cần điều chế với một tần số sóng mang trước khi gửi ra các transducer.

Để mô tả toán học, tín hiệu thu được tại mỗi đầu thu tín hiệu sóng âm (hydrophone) Tr_i được biểu diễn dưới dạng đơn giản nhất là một hàm sóng sin như sau:

$$x_i(t) = A_i \cos(2\pi f_m t + \phi_i) \quad (1)$$



Hình 5. Giản đồ thời gian của các tín hiệu phát xung sonar.

trong đó ϕ_i và f_i là góc pha của tín hiệu tới và tần số dao động của tín hiệu thu được tại đầu thu tín hiệu sóng âm Tr_i. Tín hiệu sóng âm thu được sau đó sẽ khi qua bộ khuếch đại tạp âm thấp tăng cường cường độ tín hiệu, và qua bộ lọc thông dải để loại bỏ nhiễu. Để có thể xử lý được đồng thời nhiều tín hiệu trên miền thời gian, các đầu thu song song trên miền không gian sẽ được đổi tần bằng cách nhân với một thành phần tần số f_k, được ký hiệu là $f_k(t) = \cos(2\pi f_k t)$. Khi nhân tín hiệu thu được với $f_k(t)$ ta thu được:

$$f_{y,i}(t) = x_i(t) f_k(t) = A_i \cos(2\pi f_i t + \phi_i) \cos(2\pi f_k t) \quad (2)$$

Sau khi triển khai giải tích ta thu được:

$$f_{y,i}(t) = \frac{A_i}{2}\{\cos[2\pi(f_i + f_k)t + \phi_i] + \cos[2\pi(f_i - f_k)t + \phi_i]\} \quad (3)$$

Tín hiệu thể hiện ở (3) có chứa thành phần tần số cao là tổng của hai tần số và thành phần tần số thấp là hiệu của 2 tần số. Bài báo đề xuất sử sử dụng một bộ lọc thông thấp để loại bỏ thành phần tần số cao. Như vậy tín hiệu thu được là thành phần hiệu của 2 tần số:

$$f'y, i(t) = \cos[2\pi(f_i - f_k)t + \phi_i] \quad (4)$$

Trên thực tế, tín hiệu thu không phải là dao động hàm số sin đơn giản như (1), mà nó là tổng các thành phần tín hiệu phát và các thành phần nhiễu. Để đơn giản chúng ta không xem xét đến thành phần nhiễu trong biểu diễn toán học. Giả thiết phía phát, phát hai thành phần tần số là f_1 và f_2, tức tạo ra hai dao động hàm số sin từ 2 đầu phát tín hiệu sóng âm phát. Hai tín hiệu này được biểu diễn như sau:

$$x_1(t) = \cos(2\pi f_1 t) \quad (5)$$

$$x_2(t) = \cos(2\pi f_2 t) \quad (6)$$

Các đầu thu tín hiệu sóng âm về cơ bản cũng sẽ thu được hai tần số được phát đi từ phía đầu phát tín hiệu sóng âm, sau khi trộn tần với thành phần tần số $f_k(t) = \cos(2\pi f_k t)$ và qua bộ lọc thông dải để nhận về hai nguồn tín hiệu ở hai thành phần tần số khác nhau, được biểu diễn dưới dạng

$$f'_{y,i}(t) = \cos[2\pi(f_1 - f_k)t + \phi_i] + \cos[2\pi(f_2 - f_k)t + \phi_i] \quad (7)$$

Trong trường hợp tổng quát, số lượng thành phần tần số trộn $f_k(t) = \cos(2\pi f_k t)$ phụ thuộc vào số lượng tần số phát $f_k(t)$, tức số đầu phát sóng siêu âm đặt phân tập về không gian, cũng như số lượng đầu thu sóng siêu âm Tr_i cũng được đặt phân tập về không gian, sao cho các đầu thu tín hiệu thu được tín hiêu phát mong muốn. Trường hợp cụ thể như thiết kế ở hình 1, hệ thống sử dụng hai đầu phát sóng âm đặt ở hai mạn tàu và 08 đầu thu sóng âm cũng được chia đều ở hai mạn tầu, với mỗi mạn tầu sử dụng 04 đầu thu sóng âm đặt phân tập về không gian. Như vậy để xử lý tín hiệu song song và đồng thời trên miền thời gian chúng ta cần 04 tần số trộn $f_k(t)$ phân biệt.

Tín hiệu sau khi trộn tần và qua bộ lọc thông dải để lựa chọn tần số mong muốn sẽ được đưa vào bộ ghép tín hiệu (hay bộ tổng) đưa ra một luồng tín hiệu duy nhất. Phương pháp này đã chuyển tín hiệu phân biệt từ miền không gian sang các tín hiệu phân biệt ở miền tần số, tạo tiền đề để xử lý tín hiệu đồng thời cho cả luồng tín hiệu thu phát song song mà không cần thêm chi phí phần cứng của hệ thống. Tín hiệu sau khi ghép sẽ được đưa đến bộ chuyển đổi A/D để chuyển tín hiêu từ dạng analog sang dạng số để xử lý tín hiệu tạo ảnh quét được trong không gian 2D hoặc 3D.

III. KẾT QUẢ THỰC NGHIỆM VÀ THẢO LUẬN

Các thử nghiệm trình bày bài báo này là các thử nghiệm tiến hành ngày 12 tháng 11 năm 2021 tại vịnh Nha Trang.

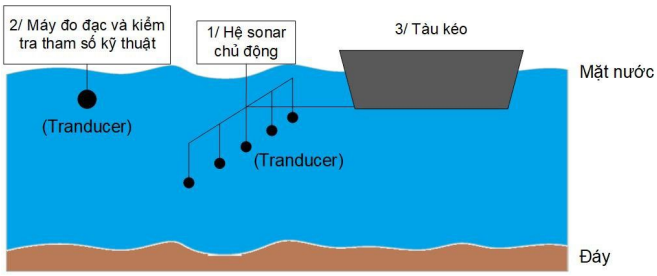
Bảng I
CÁC THAM SỐ HỆ THỐNG

Tham số	Giá trị
Chu kỳ phát xung sonar T	0.1s
Độ rộng một xung sonar Δt	0.4ms
Tần số lấy mẫu tại phía thu	192kHz
Số lượng transducer phát M	2
Số lượng hydrophong thu N	8
Khoảng cách thiết lập giữa hai Hydrophone liền kề L	$0.09m$
Tx_1 Tần số phát của transducer thứ nhất f_1	165kHz
Tx_2 Tần số phát của transducer thứ 2 f_2	200kHz

Bảng II
CÁC ĐIỀU KIỆN MÔI TRƯỜNG

Thời tiết	nắng
Nhiện độ	$25°C$
Gió	gió nhẹ
Sóng	sóng lăn tăn

Biển Nha Trang là biên nước nông. Một số tham số hệ thống được trình bày vắn tắt ở Bảng I. Các tham số về môi trường được trình bày ở Bảng II. Các điều kiện thử nghiệm về thời tiết, sóng gió là khá thuận lợi. Cụ thể, trời nắng nhẹ, gió nhẹ, sóng lăn tăn. Những điều kiện thử nghiệm như vật thường đem đến kết quả tốt nhất. Như chúng ta biết, việc lan truyền sóng âm dưới nước khá nhạy cảm với các điều kiện bên ngoài, đặc biệt là trong điều kiện biển nông. Trong các điều kiện khắc nghiệt như gió bão, các kết quả về truyền dẫn sóng âm thường là rất tồi và độ tin cậy thấp.



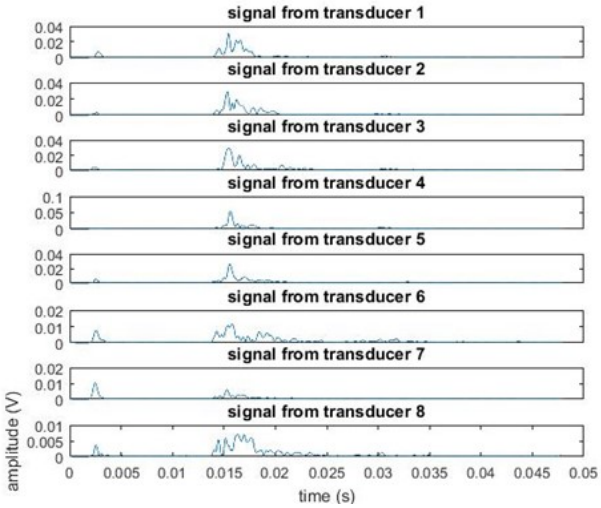
Hình 6. Kịch bản thử nghiệm hệ sonar chủ động.

Kịch bản thử nghiệm hệ thống được trình bày ở Hình 6, trong đó đó tàu thử nghiệm được sử dụng để gắn các trang bị kỹ thuật, với ký hiệu (1) là hệ sonar chủ động, (2) là máy đo đạc và kiểm tra tham số kỹ thuật, và 3) là tàu kéo hệ thống. Transducer là các đầu thu phát tín hiệu sóng âm.

Hình ảnh sản phẩm phần cứng của hệ thống sonar chủ động sử dụng phương pháp thiết kế đề xuất trong bài báo được trình bày ở Hình 7, trong đó hệ thống sử dụng hai búp sóng quét với hai tần số quét khác nhau, cụ thể là 165 Khz và 200 Khz. Ở phía thu, hệ thống sử dụng 08 đầu thu là các hydrophone có độ nhạy cao, với thiết kế sử dụng 04 đầu thu cho mỗi một mạn tầu (sườn tàu). Như vậy hệ

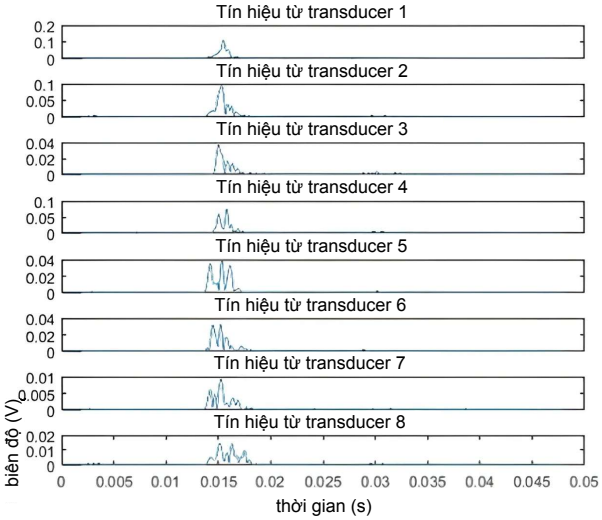


Hình 7. Hình ảnh hệ thống sonar được thiết kế trong thực tế.



Hình 8. Tín hiệu được từ transducer phát 165 KHz (sau khi khuếch đại và lọc nhiễu)

thống sử dụng 08 đầu thu dữ liệu song song và độc lập. Các transducer thu phát được đặt trên một phao kéo như trình bày ở Hình 10. Dữ liệu phản xạ đáy thu được ở tần số phát 165 kHz ghi nhận từ 08 đầu thu, sau khi khuếch đại và xử lý lọc nhiễu được trình bày ở Hình 8. Tương tự như vậy, tín hiệu phản xạ đáy thu được từ 08 đầu thu ở tần số 200 kHz được thể hiện ở Hình 9. Dưa vào tín hiệu thu được từ các đầu thu song song, hệ thống xử lý tín hiệu sonar thực hiện như sơ đồ kiến trúc hệ thống trình bày ở Hình 4 và sau đó hiển thị trên màn Hình quét tín hiệu như trình bày ở Hình 11. Trong Hình 11, vị trí đo là vịnh Nha Trang với kết quả đo đạc độ sâu dựa trên dữ liệu thu được từ 08 đầu thu là 16.5 m. Kết quả đo đạc của hệ thống đã được so sánh với các thiết bị chuyên dụng của quân đội với sai số khác biệt so với các thiết bị kiểm chuẩn chuyên dụng là tương đối nhỏ.



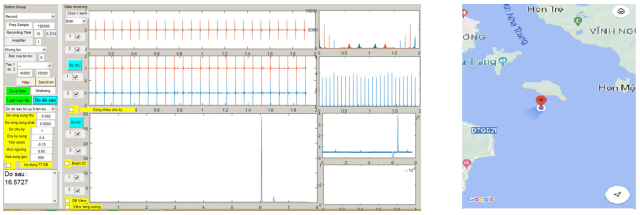
Hình 9. Tín hiệu được từ transducer phát 200 KHz (sau khi khuếch đại và lọc nhiễu).



Hình 10. Hình ảnh thực nghiệm hệ thống sonar tại Vịnh Nha Trang ngày 12 tháng 11 năm 2021.

IV. KẾT LUẬN

Bài báo trình bày phương pháp thiết kế hệ sonar chủ động với dạng quét sườn và ứng dụng việc tạo dựng các tín hiệu ảnh 2D hoặc 3D của các vật thể dưới nước. Bài báo cũng trình bày các sản phẩm thực tế được thiết kế chế tạo dựa trên phương pháp đề xuất và các kết quả thực nghiệm đo đạc ở vịnh Nha Trang. Việc áp dụng kiến trúc hệ thống quét tín hiệu siêu âm đề xuất trong bài báo cho phép ghép nối nhiều chấn tử phát sóng âm (đầu phát tín hiệu sóng âm) và các chấn tử thu sóng âm (đầu thu tín hiệu sóng âm) phục vụ cho việc quét tín hiệu ảnh của hệ thống siêu âm 3D chủ động với mức chi phí phần cứng thấp, đồng thời xử lý được nhiều nguồn tín hiệu thu để nâng cao chất lượng bức ảnh quét. Trong bài báo này chúng tôi tập trung về ứng dụng thuật toán cho hệ sonar chủ động dạng sonar



Hình 11. Kết quả đo đạc thực nghiệm hệ thống sonar tại Vịnh Nha Trang.

quét sườn. Tuy nhiên hiện nay một số công nghệ cho phép mở rộng phạm vi quét với độ phân giải tốt hơn so với hệ sonar quét sườn, đó là công nghệ sonar mặt mở [16, 17]. Ứng dụng giải pháp giảm độ phức tạp của hệ thống bằng cách chuyển phân tích tín hiệu từ miền không gian sang miền thời gian cho hệ sonar mặt mở là đối tượng nghiên cứu tiếp theo của chúng tôi.

ACKNOWLEDGEMENT

Một phần bài báo này đã được đăng trong đơn sáng chế đăng ký với Cục Sở Hữu Trí Tuệ, số đơn 1.2021.00616, 1.2021.00617 ngày 04/02/2021 [18, 19].

TÀI LIỆU THAM KHẢO

[1] I. G. Bryden, “An active sonar system using acvd beam-forming,” *Applied Acoustics*, vol. 27, no. 4, pp. 275–285, 1989.
[2] M. Caccia, G. Bruzzone, and G. Veruggio, “Active sonar-based bottom-following for unmanned underwater vehicles,” *Control Engineering Practice*, vol. 7, no. 4, pp. 459–468, 1999.
[3] E. J. Sullivan, *Active Sonar*, pp. 1737–1755. New York, NY: Springer New York, 2008.
[4] B. H. Maranda, *Passive Sonar*, pp. 1757–1781. New York, NY: Springer New York, 2008.
[5] G. Marani, S. K. Choi, and J. Yuh, “Underwater Autonomous Manipulation for Intervention Missions AUVs,” *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, 2009.
[6] D. W. Kim, “Tracking of REMUS Autonomous Underwater Vehicles with Actuator Saturations,” *Automatica*, vol. 58, no. C, pp. 15–21, 2015.
[7] J. G. Moss, “Sonar bathymetry system transmit-receive sequence programmer,” U.S. Patent US3500302A, 1969.
[8] C. Aaron, “Sonar assembly for reduced interference,” European Patent EP2626721B1, 2012.
[9] K. Shahzad F., “Methods and apparatuses for reconstructing a 3d sonar image,” Canadian Patent CA2922781A1, 2016.
[10] M. Brian T., “Downscan imaging sonar,” U. S. Patent US8305840B2, 2009.
[11] S. Martyn, “Method of object tracking using sonar imaging,” U. S. Patent US9019795B2, 2012.
[12] N. S. Matthew Jason Zimmerman, Alexander Yakubovskiy, “Integrated sonar devices and methods,” U. S. Patent US20150369908A1, 2015.
[13] S. Van Doan, T. C. Tran, and V. D. Nguyen, “Doa estimation of underwater acoustic signals using non-uniform linear arrays,” in *Industrial Networks and Intelligent Systems* (T. Q.

Duong and N.-S. Vo, eds.), (Cham), pp. 103–110, Springer International Publishing, 2019.
[14] V. D. Nguyen, N. M. Luu, Q. K. Nguyen, and T.-D. Nguyen, "Estimation of the acoustic transducer beam aperture by using the geometric backscattering model for side-scan sonar systems," *Sensors*, vol. 23, no. 4, 2023.
[15] Z. Yang, J. Zhao, Y. Yu, and C. Huang, "A sample augmentation method for side-scan sonar full-class images that can be used for detection and segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–11, 2024.
[16] D. Brown, D. Cook, and J. Fernandez, "Results from a small synthetic aperture sonar," in *OCEANS 2006*, pp. 1–6, 2006.
[17] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: A review of current status," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 3, pp. 207–224, 2009.
[18] Q. K. Nguyễn and V. Nguyễn, "Phương pháp thiết kế hệ thống quét ảnh các vật thể dưới nước sử dụng đồng thời nhiều đầu thu phát tín hiệu sóng âm," Bằng độc quyền sáng chế số 39938 do Cục sở hữu Trí tuệ cấp ngày 26.04.2024.
[19] Q. K. Nguyễn and V. Nguyễn, "Phương pháp quét ảnh không gian ba chiều sử dụng công nghệ nhận dạng và định vị sóng âm dưới nước," Bằng độc quyền sáng chế số 39939 do Cục sở hữu Trí tuệ cấp ngày 26.04.2024.



Nguyễn Văn Đức tốt nghiệp kỹ sư chuyên ngành kỹ thuật thông tin năm 1995, tốt nghiệp thạc sỹ kỹ thuật năm 1997 chuyên ngành điện tử viễn thông, Trường ĐHBK Hà Nội; tốt nghiệp tiến sỹ chuyên ngành kỹ thuật thông tin Trường ĐH tổng hợp Hannover, CHLB Đức năm 2003. Từ năm 1998 đến nay là giảng viên trường ĐHBK Hà Nội. Lĩnh vực nghiên cứu hiện nay là hệ thống thông tin thủy âm, hệ thống sonar, công nghệ MIMO, OFDM.
Email: duc.nguyenvan1@hust.edu.vn



Nguyễn Quôc Khương tốt nghiệp kỹ sư chuyên ngành kỹ thuật thông tin năm 1995, tốt nghiệp thạc sỹ kỹ thuật năm 1997 chuyên ngành điện tử viễn thông, Trường ĐHBK Hà Nội; tốt nghiệp tiến sỹ chuyên ngành kỹ thuật thông tin Trường ĐHBK Hà Nội năm 2011. Từ năm 1998 đến nay là giảng viên trường ĐHBK Hà Nội. Lĩnh vực nghiên cứu hiện nay là hệ thống thông tin thủy âm, hệ thống sonar, công nghệ MIMO, OFDM.
Email: khuong.nguyenquoc@hust.edu.vn

Pathway-Capsule Neural Networks

Phuong-Nam Nguyen

Faculty of Computer Science PHENIKAA University, Yen Nghia, Ha Dong Hanoi, 12116, Vietnam.

Correspondence: Phuong-Nam Nguyen, nam.nguyenphuong@phenikaa-uni.edu.vn

Received: 15/2/2024, revised: 07/5//2024, accepted: 11/5/2024

Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1265

Abstract: The progress made in Next-Generation Sequencing technology has resulted in many genetic databases, which offer the potential for more effective diagnostic tools. Recent literature highlights Artificial Intelligence (AI) as one of the most potent methods. Nonetheless, when it comes to analyzing genetic data, Machine Learning (ML) and Deep Learning (DL) face three significant challenges due to: (1) multi-modality analysis, (2) model explainability, and (3) extension to patient-specific biomarker identification. In this paper, we introduce Pathway-Capsule Neural Networks (*PCNN*) and its Bayesian version (*BayesPCNN*) to address the three limitations. We demonstrate the model using Big Data of $7,372$ samples with 57 protein expressions from the TCGA dataset. In the best setting, the proposed PCNN outperforms tree-based ML models and KNN with a significant gap of $10\% - 30\%$ in accuracy for Pan-Cancer detection; while only outperformed by kernel methods with under 2%. Another contribution of BayesPCNN is highlighted as Explainable AI (*XAI*), which reveals tumors with potential metastasis and patient-specific biomarkers.

Keywords: *deep learning, artificial intelligence, bioinformatics*

Tiêu đề: **Pathway-Capsule Neural Networks**

Tóm tắt: Tiến bộ trong Công nghệ Sequencing Thế hệ Tiếp theo đã dẫn đến nhiều cơ sở dữ liệu gen, mở ra tiềm năng cho các công cụ chẩn đoán hiệu quả hơn. Các báo cáo khoa học quốc tế đương đại nhấn mạnh rằng Trí tuệ Nhân tạo (AI) là một trong những phương pháp mạnh mẽ nhất. Tuy nhiên, khi đến việc phân tích dữ liệu gen, Học Máy (ML) và Học Sâu (DL) đối mặt với ba thách thức lớn do: (1) phân tích đa dạng, (2) khả năng giải thích mô hình, và (3) mở rộng để xác định chỉ số sinh học cụ thể cho từng bệnh nhân. Trong bài báo này, chúng tôi giới thiệu Pathway-Capsule Neural Networks (*PCNN*) và phiên bản Bayesian của nó (*BayesPCNN*) để giải quyết các hạn chế này. Chúng tôi thể hiện mô hình bằng Dữ liệu Lớn của $7,372$ mẫu với 57 biểu hiện protein từ tập dữ liệu TCGA. Trong cài đặt tối ưu, PCNN được đề xuất vượt trội hơn so với các mô hình ML dựa trên cây và KNN với khoảng cách đáng kể là $10\% - 30\%$ trong độ chính xác cho việc phát hiện Ung thư Phổ quát; trong khi chỉ bị vượt qua bởi các phương pháp nhân bản với dưới 2%. Một đóng góp khác của BayesPCNN được nhấn mạnh là Trí tuệ Nhân tạo Giải thích (*XAI*), làm sáng tỏ những khối u có tiềm năng di căn và chỉ số sinh học cụ thể cho từng bệnh nhân.

Từ khóa: *học sâu, trí tuệ nhân tạo, toán y sinh*

I. INTRODUCTION

The next-generation sequencing (NGS) technology advancement brings a large collection of genetic databases, which promises more powerful diagnostic tools. Artificial Intelligence (AI) is one of the most powerful approaches in the recent associated literature. However, Machine Learning (ML) and Deep Learning (DL) deal with a major challenge when analyzing genetic data. Specifically, most AI-driven solutions are black-box, which is not ideal for medical context as clinicians must understand the model inference to offer precise treatments for the *patient of target*.

In this paper, we propose a new neural architecture named *Pathway-Capsule Neural Networks* (PCNNs) to achieve three goals

1) Our proposed dual-encoder retrieves hierarchical information from multi-mode genetic data using a generalized loss module (Section III.2 and III.3).
2) The proposed AI model has two settings: (1) generic PCNN and (2) BayesPCNN, which is more accurate than most conventional ML models for genetic data. Both versions outperform four competitor models in the Pan-Cancer detection task while slightly outperformed by the Support Vector Machine (SVM) (Section IV).
3) BayesPCNN is an XAI model identifying secondary tumors and patient-specific biomarkers. (see Section IV.2.b and IV.2.c).

II. RELATED WORKS

1. Capsule Networks in Cancer Genetics

Capsule Networks, or CapsNets, are an Artificial Neural Network (ANN) that was proposed to address some of the limitations of Convolutional Neural Networks (CNNs) [1]. CapsNets solve the problem of preserving the hierarchical relationships between parts and wholes, and they have rotational and positional invariance[2]. They use a group of neurons to form a "capsule" that recognizes an object in the input data and outputs a vector. The vector's length represents the object's probability of being present in the input, and its orientation represents the object's properties, such as pose, deformation, and texture. In the context of cancer genetics data analysis, capsule neural networks are used to improve the accuracy, robustness, and the model interpretation[3–9].

$$\textbf{NNs}: \boldsymbol{X} \xrightarrow{f(\boldsymbol{\theta})} \boldsymbol{h}(\dim = 1) \longrightarrow \text{CLS}$$
$$\text{update}$$
$$\boldsymbol{\theta} \xleftarrow[\text{Stochastic Opt.}]{} \mathcal{D}(\boldsymbol{X}, \boldsymbol{Y})$$

$$\textbf{CapsNet}: \boldsymbol{X} \xrightarrow{f(\boldsymbol{\theta})} \boldsymbol{h}(\dim \geq 2) \longrightarrow \text{CLS}$$
$$\text{update}$$
$$\boldsymbol{\theta} \xleftarrow[\text{Dynamic Routing}]{} \mathcal{D}(\boldsymbol{X}, \boldsymbol{Y})$$

2. Challenges of AI models in Cancer Genomics

Genetic data analysis model architectures still need improvement despite many achievements. A recent review[10] shows tree-based and kernel methods are more commonly used than neural networks because Deep Neural Networks (DNN) are ineffective in learning from tabular data. DNNs are good at learning representations from data with physically structured domains, such as images or text sequences. However, tabular data does not have generic physical structures. Thus, DNN performance degrades, especially in low-limited data scenarios[11].

III. METHOD: PATHWAY-CAPSULE NEURAL NETWORK (PCNN)

1. Overview

We compare our proposed PCNN to conventional NN and CapsNets in Figure 1. Our model architecture has three distinct features:

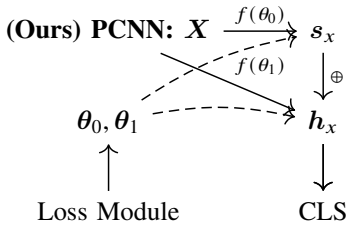


Figure 1: Pathway-Capsule Neural Networks (PCNN) Information Flow, compared to conventional NN and CapsNets.

1) We partition the model weight architecture as a capsule $\boldsymbol{s}_x$ and $\boldsymbol{h}_x$, which encodes the hierarchical structure on inputs.
2) The loss module is designed to update each cluster of model weight based on mutational signals and true labels.
3) The final classifier is produced by both secondary representation $\boldsymbol{s}_x$ and primary representation $\boldsymbol{h}_x$ though element-wise sum, which is inspired by residual connections in ResNet[12]. It allows us to use deeper model architecture with more efficient trainability.

2. Dual-Neural Encoder for Multi-Modality Analysis

We propose a novel neural architecture design for ANN models named Pathway-Capsule Neural Networks (PCNN), which leverages a dual-encoder for multi-modality analysis. Specifically, given two modes of data in cancer genetics, denoted as $D^1 = (\boldsymbol{X}_1, \boldsymbol{Y}_1)$ and $D^2 = (\boldsymbol{X}_2, \boldsymbol{Y}_2)$, in which $\boldsymbol{Y}_1$ is labels for cancer classes and $\boldsymbol{Y}_2$ is the mutational signals. The input data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be any genetic data ranging from DNA methylation, RNA/mRNA, or protein expression.

Construction of Pathway-Capsule ($g(.)$): We propose a capsule-like neural architecture to encode mutational signals for input pathways. Specifically, given the input data of k-genetic features $\boldsymbol{X}_1 = \boldsymbol{X}_2 = \{G_1, \ldots, G_k\}$, the Pathway-Capsule block $g(.)$ encodes the input signals as the following forward-pass in **Algorithm** 1. The output of the Pathway-Capsule block is the neural signal that represents mutational activation signals (binary $\in \{0, 1\}$). We denote the transformation of this neural block as $g(\boldsymbol{X}) = \boldsymbol{s}_x$.

Construction of Classifiers ($f(.)$): The final ANN classifier is parameterized by $(n_{\text{capsules}} = k, n_{\text{layers}} = l, \text{hidden dim} = d)$, in which n_{capsules} Pathway-Capsule block encodes mutational signals of k-genetic features with l layers of d hidden neurons in each block. We aggregate

the neural signals from all Pathway-Capsule blocks by concatenation, which is given as

$$\boldsymbol{H} = \text{Concat}(\boldsymbol{s}_x^i)_{i=1}^k. \quad (1)$$

We improve the trainability of the proposed model architecture by incorporating the original signals $\boldsymbol{X}_{(.)}$ into the embedding $\boldsymbol{H}$. Of note, residual connections[12] is already adopted in the Pathway-Capsule block for the same purpose depicted in the demonstrated forward pass. Here, we use the concatenation operator to construct such a representation, given as

$$\boldsymbol{H}_c = \text{Concat}(\boldsymbol{X}, \boldsymbol{H}). \quad (2)$$

The proposed representations $\boldsymbol{H}_c$ could have two-fold advantages. First, the input signal is accounted for in such an embedding so that it could ameliorate the model trainability. Second, the embeddings contain hierarchical information from multi-mode data, including (1) input genetic expressions and (2) mutational signals. The final classification head $f(.)$ is built on $\boldsymbol{H}_c$, which is given as

$$\hat{\boldsymbol{y}} = f(\boldsymbol{H}_c), \quad (3)$$

where $\hat{\boldsymbol{y}}$ is the predicted outcome. The forward pass of the proposed classifier is given as in **Algorithm** 2.

3. Loss Module

Our proposed PCNN model is different from Capsule Neural Networks and ANNs with auxiliary heads in [1, 3, 6, 8]. It encodes hierarchical information for multi-modality analysis and includes a loss module to learn class annotations and mutational signals. This is the key feature that sets it apart from existing models.

Here, we design the loss module consisting of two components, given as

$$\mathcal{L} = \mathcal{L}_{\text{Cls}} + \mathcal{L}_{\text{Mutation}} \quad (4)$$

Of note, the annotations for model feedback include $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, which is introduced in the earlier section. The classification loss is computed as

$$\mathcal{L}_{\text{Cls}}(\hat{\boldsymbol{y}}, \boldsymbol{Y}_1), \quad (5)$$

using cross-entropy criterion given as

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

where N is the number of samples, M is the number of classes, y_{ij} is the binary indicator (0 or 1) if class j is the true label for sample i p_{ij} is the predicted probability that sample i belongs to class j. The latter term in the final loss is computed between neural signals $\boldsymbol{s}_x^i$ and mutational signals recorded in each k-genetic feature. Specifically, we use the binary version of cross-entropy loss, yielding

$$\mathcal{L}_{\text{Mutation}} = \sum_{i=1}^{k} \mathcal{L}_{BCE}(\boldsymbol{s}_x^i, \boldsymbol{y}_2^i), \quad (6)$$

where $\boldsymbol{y}_2^i$ is the mutational-activation signals from $\boldsymbol{Y}_2$.

IV. RESULTS

1. Experimental Design

a) Case-study

Our proposed PCNN was demonstrated using RPPA and CNV, from DNA mutation to protein expressions. We used TCGA[13] with 7,372 patients, 31 cancer types, and $k = 57$ genetic features per observation. Mutational signals were created by CNV data with simple thresholding of $\boldsymbol{Y}_2 = 1$ when $y_{\text{CNV}} \neq 0$, otherwise 0.

b) Training Protocol

We optimize hyper-parameters using Sequential Model-based Optimization with Tree-Parzen Estimator[14]. This optimizer is cost-efficient for large search spaces[15, 16]. We optimize the number of capsule layers, neurons per layer, dropout rate, learning rate, and weight decay. We perform 100 trials using SMBO-TPE[14]. Each model candidate trains for 1000 epochs using the AdamW[17] optimizer. We conduct experiments using Python 3.7.0, numpy 1.21.5, scikit-learn 1.0.2, and PyTorch 1.11 on an Intel i9 processor, 16GB DDR4 memory, and GeForce GTX 1060 Mobile GPU.

2. Effectiveness Analysis

a) PCNN Enables More Accurate ML Classifiers for Pan-Cancer Detection

We compared our proposed PCNN model with five classical ML models and excluded ANNs for two reasons. *First*, PCNN is a more generalized version of ANNs. *Second*, the ML models have demonstrated superior performance in genetic data analysis on TCGA data[10]. We used the test accuracy score evaluated by a 5-Fold CV as the comparison metric and optimized all models using SMBO-TPE for a fair comparison.

We present two versions of PCNNs in Figure 2 - one based on a generic setting and the other on a Bayesian setting (by DeepEnsemble[18]). The generic PCNN model outperforms four competitor models, except SVM with radial kernel parameterized by $C = 5$. The gap between PCNN and decision tree and KNN models is large, with a difference of approximately 92.5% compared to 62.5% and 82.5%, respectively. PCNN also performs slightly better than other tree-based approaches by about 2.5% – 5% in test accuracy.

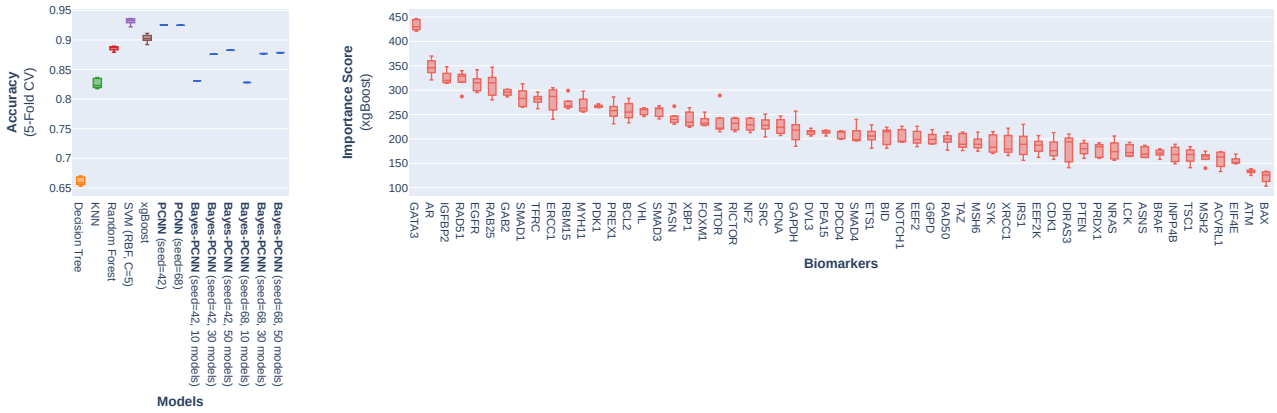


Figure 2: **(Left)** Our proposed generic PCNN and Bayes-PCNN outperform competed for tree-based classifiers, including decision tree, KNN, and xgBoost, while slightly outperformed by SVM using 5-Fold Cross-Validation (CV). **(Right)**: Proteomics-Biomarker scored by importance score from xgBoost models based on 5-Fold CV.

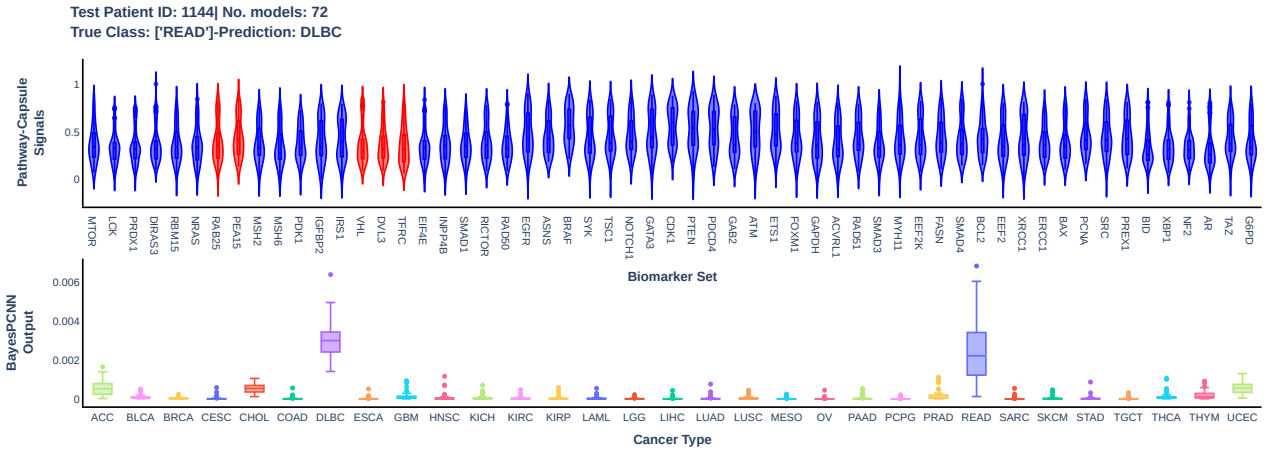


Figure 3: **Bayes-PCNN Improves Patient-specific Inference. (Top)** Neural signals recorded in Pathway-Capsule outputs, colored by mutational signal: red for mutation and blue for wild-type. **(Bottom)** Assembled predictions by Bayes-PCNN using 72 optimized models: Although the top-1 prediction is wrong, the second highest confidence is the right cancer class.

The predictive performance of PCNN is slightly lower than SVM by about 2.5%.

BayesPCNN, a variant of PCNN in the Bayesian setting, achieves lower accuracy than the generic version due to its aggregation of top-10, -30, and -50 to produce the mean of predicted values. While more classifiers result in higher accuracy, there is a trade-off between inference time and accuracy. However, BayesPCNN using ten to thirty models gains a 10% accuracy gain, which saturates at 50 models, whereas the additional 20 classifiers only gain $< 2\%$ accuracy.

b) Enhancing The Robustness of Model Inference with Bayes-PCNN

Although Bayes-PCNN induces a slight degradation in accuracy, the model setting has two-fold advantages. *First*, the model produces a more robust and reliable inference thanks to the nature of Bayesian learning. We show in Figure 3 that even if the classifier makes the wrong prediction in its top-1 configuration, the second highest confident prediction is the right cancer class. *Second*, we can associate top-2 and top-3 confidence prediction as potential metastasized diseases (secondary). The analyzed sample contains genetic signals that resemble other cancer classes.

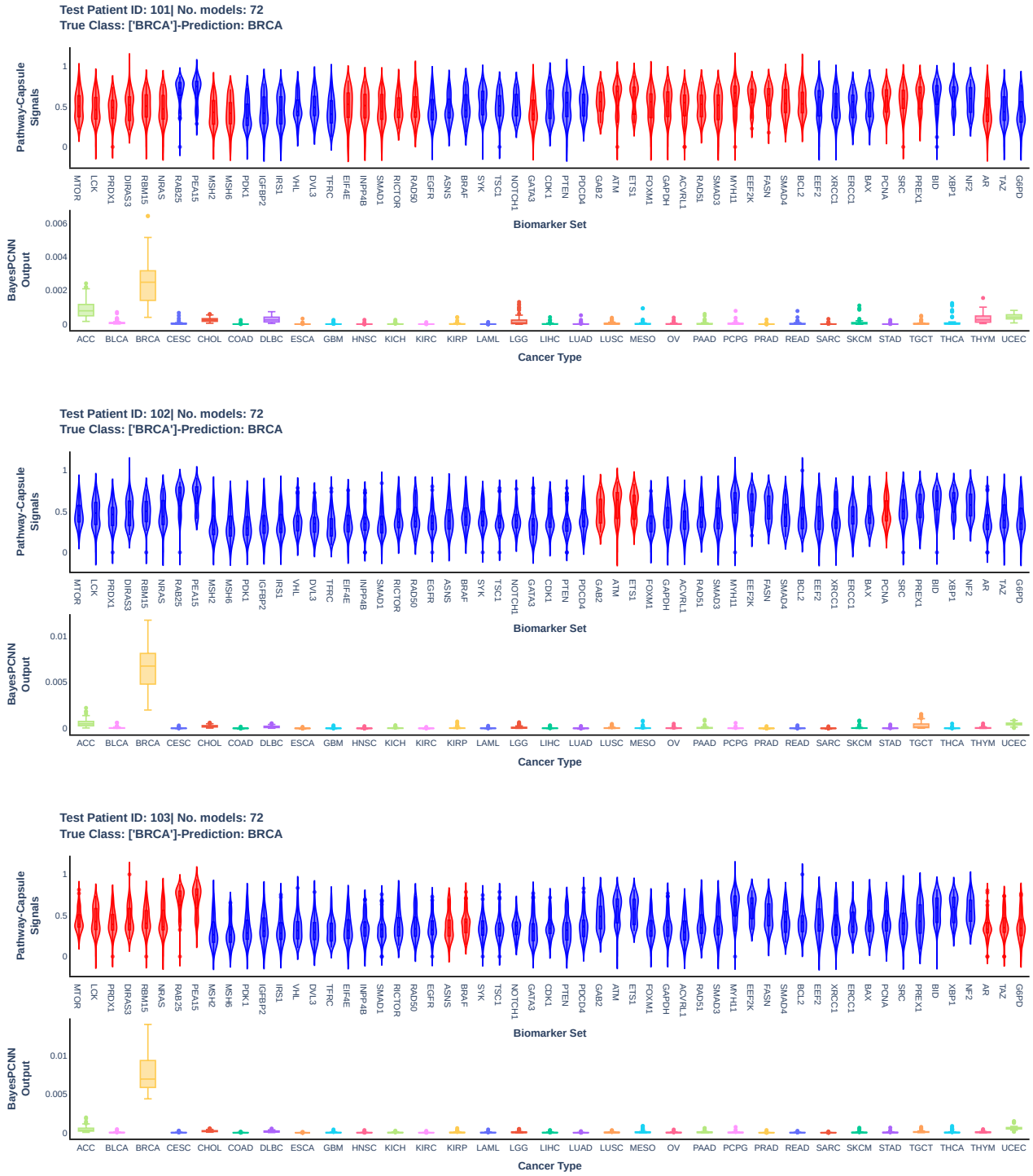


Figure 4: **Explainable AI (XAI) by Bayes-PCNN Demonstrated on Comparing Different Patients with *Breast Cancer*.** The top panel depicts neural signals recorded from the output of Pathway-Capsules blocks, colored by true mutational signals (red for mutant, blue for wild-type). The bottom panel depicts the confidence in model inference by cancer diseases. The predicted metastasized tumor has *MSH2*, *MSH6*, *EIF4E*, *INPP4B* and *SMAD1* mutations, that associates with high values in capsule signals

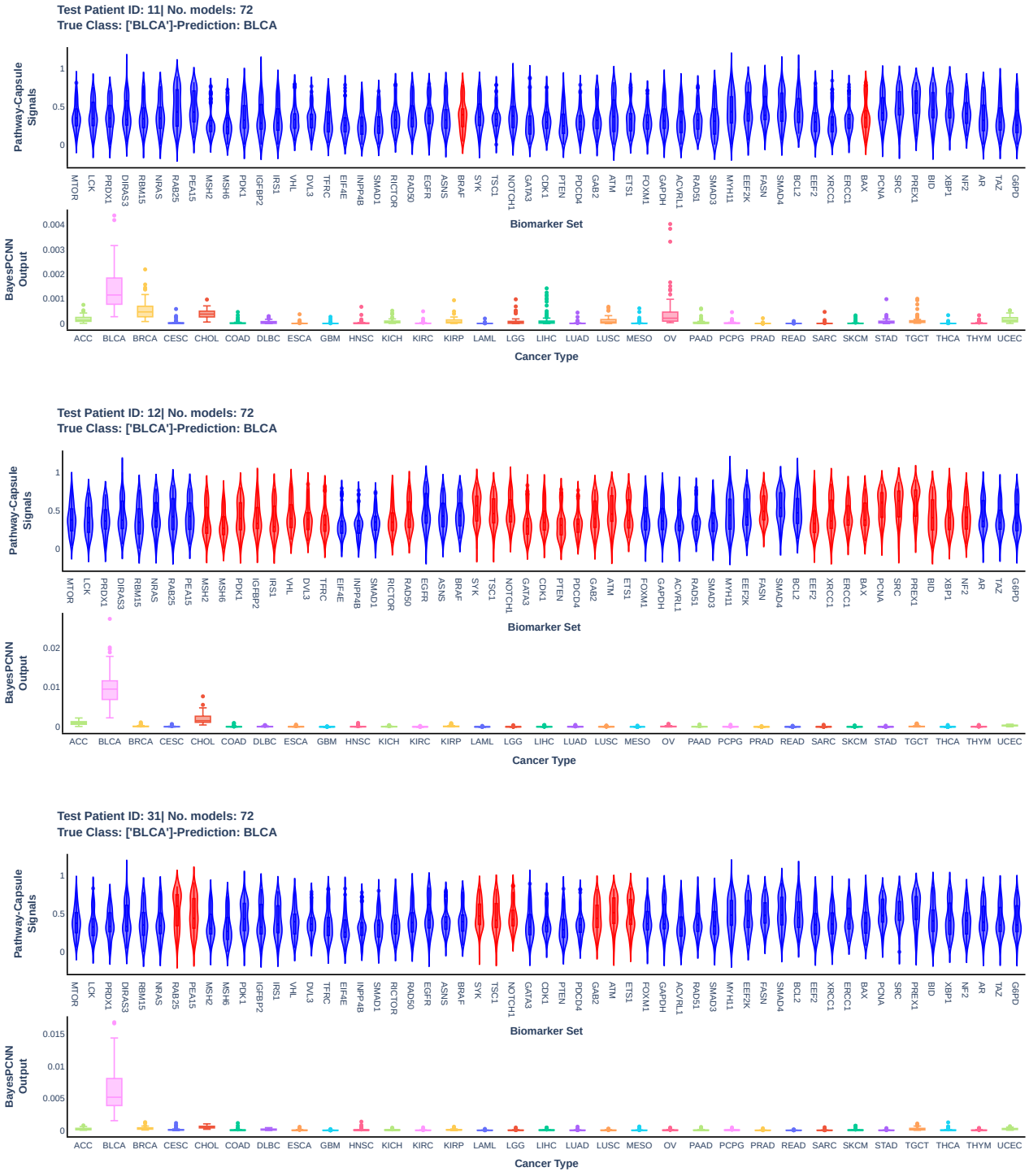


Figure 5: **Explainable AI (XAI) by Bayes-PCNN Demonstrated on Comparing Different Patients with *Bladder Cancer*.** The top panel depicts neural signals recorded from the output of Pathway-Capsules blocks, colored by true mutational signals (red for mutant, blue for wild-type). The bottom panel depicts the confidence in model inference by cancer diseases. The co-activation of *BRAF* and *BAX* could lead to potential metastasis to secondary at the breast, suggested by our BayesPCNN.

For instance, BayesPCNN suggests the diagnosed patient in Figure 3 could have potential disease metastasis from the rectum (READ) to Lymphoid Neoplasm Diffuse Large B-cell Lymphoma (DLBC) or adrenocortical carcinoma (ACC). The two discussed features are superior to the existing models, which are considered black-box.

c) PCNN Enables Explainable-AI (XAI) for Patient-specific Diagnostics

We used Pathway-Capsule blocks to understand model inference and identify clinically significant observations in breast cancer patients. Figure 4 shows the Pathway-Capsule output, colored according to mutational signals. Our XAI system identified patient-specific biomarkers, whereas existing scoring systems for feature importance only improve model accuracy.

Patient 101 may have potential metastasis to ACC, while the other patients are not at risk for secondary tumors. The predicted metastasized tumor has mutations in *MSH2*, *MSH6*, *EIF4E*, *INPP4B*, and *SMAD1*, which are associated with high values in capsule signals. *MSH2* and *MSH6* staining absence in tumors can be due to genetic mutations inherited from the germline[19], and the risk of having *MSH2* microsatellite instability (MSI) is higher in males with colon tumors, late-stage tumors, and poor tumor differentiation. Similarly, males with colon tumors and tumors at a late stage are more likely to have *MSH6* MSI. *SMAD1* promotes cell invasion and metastasis in different cancer types[20].

We can see three bladder tumor patients in Figure 5. Patient 12 has the most mutations, but our Bayes-PCNN model shows that patient 11 has a higher risk of metastasis due to the co-activation of *BRAF* and *BAX* genes. *BRAF* mutations[21] can cause carcinogenesis[22] and are linked to poor prognosis in 5-10% of metastatic colorectal cancers. Activation of the *BAX* driver gene triggers apoptosis, leading to the death of cancer cells.

We found that neural signals from Pathway-Capsule and mutational activation can predict disease metastasis risk. For personalized treatment, we identified potential proteomics biomarkers like *SMAD1* and *BAX*.

V. DISCUSSION

Why PCNN is More Generalized than Conventional ANN Design in the Context of Multi-Modality Analysis?

The proposed PCNN is superior to conventional machine learning models as it encodes information from multiple genomic data modes into the final latent vectors. Multi-modalities analysis is crucial in AI-driven genetic analyses, but the high-dimensional data can impact the effectiveness and efficiency of machine/deep learning models. Our proposed PCNN builds on recent proposals for the path towards General AI (GAI). For instance, GLOM[2] suggests representing part-whole hierarchies in neural networks, which is achieved in the proposed PCNN by the dual-encoder. Additionally, our proposal adopts the trainable cost module[23]. However, the intrinsic cost module still needs addressing. Here, we design this module to encode mutational signals from the fundamental mode of the DNA codes.

Some Limitations of The Model Architecture

We have identified some limitations with our proposed neural architecture design. Firstly, our model requires additional costs for data annotations of mutational signals, which can be significant in medical data. However, this issue can be addressed by high-end pre-training protocols such as self-supervised learning. *Secondly*, we are also concerned about the model's scalability, as our architecture consists of dense auxiliary capsules, which could lead to high latency in model training and inference.

VI. CONCLUSION

To this end, I have introduced a novel architecture PCNN with a dual-neural-encoder to retrieve information from multi-modalities of genetic data in Section III. Although the model architecture can be generalized on various data scenarios, I have demonstrated the proof of concept using Big Data, including mutational signals and protein expressions in Section IV.1.a. The numerical experiment yields promising results three-fold. *First*, the generic PCNN outperformed four in five classical ML competitor models (Section IV.2.a). *Second*, the BayesPCNN improves the robustness of model inference and suggests a potential secondary site of tumor based on data-driven inference (Section IV.2.b). *Finally*, the final XAI-driven analysis yields robust proteomics-biomarkers demonstrated in some cancer types (Section IV.2.c). Besides, I also highlighted the motivation and explained why our proposed PCNN guarantees the improvements in Section V with the model limitation is discussed in Section V. For further study, I suggest several directions, such as investigating the proposed PCNN and BayesPCNN in other learning tasks not restricted to genetic data. Besides, in-depth studies of model interpretation to discover new biomarkers are worth investigating. Finally, addressing the discussed limitation could improve the effectiveness and scalability of the proposed PCNN.

REFERENCES

[1] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[2] G. Hinton, "How to represent part-whole hierarchies in a neural network," *Neural Computation*, pp. 1–40, 2022.
[3] C. Peng, Y. Zheng, and D.-S. Huang, "Capsule network based modeling of multi-omics data for discovery of breast cancer-related genes," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 5, pp. 1605–1612, 2019.
[4] P. Zheng, Y. Qi, X. Li, Y. Liu, Y. Yao, and G. Huang, "A capsule network-based method for identifying transcription factors," *Frontiers in Microbiology*, vol. 13, p. 1048478, 2022.
[5] Y. Wang, L. Huang, S. Jiang, Y. Wang, J. Zou, H. Fu, and S. Yang, "Capsule networks showed excellent performance in the classification of herg blockers/nonblockers," *Frontiers in pharmacology*, vol. 10, p. 1631, 2020.
[6] L. Wang, X. Miao, R. Nie, Z. Zhang, J. Zhang, and J. Cai, "Multicapsnet: a general framework for data integration and interpretable classification," *Frontiers in genetics*, p. 2237, 2022.
[7] X. Luo, X. Kang, and A. Schönhuth, "Predicting the prevalence of complex genetic diseases from individual genotype profiles using capsule networks," *Nature Machine Intelligence*, vol. 5, no. 2, pp. 114–125, 2023.
[8] L. Wang, R. Nie, Z. Yu, R. Xin, C. Zheng, Z. Zhang, J. Zhang, and J. Cai, "An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell rna-sequencing data," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 693–703, 2020.
[9] G. Hadash, E. Kermany, B. Carmeli, O. Lavi, G. Kour, and A. Jacovi, "Estimate and replace: A novel approach to integrating deep neural networks with existing applications," *arXiv preprint arXiv:1804.09028*, 2018.
[10] J. Liñares-Blanco, A. Pazos, and C. Fernandez-Lozano, "Machine learning analysis of tcga cancer data," *PeerJ Computer Science*, vol. 7, p. e584, 2021.
[11] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[13] D. C. C. B. R. . J. M. A. . K. A. . P. T. . P. D. . W. Y. 53 and T. S. S. L. D. A. 68, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.
[14] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
[15] N. Nguyen and K.-C. Chen, "Quantum embedding search for quantum machine learning," *IEEE Access*, vol. 10, pp. 41 444–41 456, 2022.
[16] N. Nguyen and J. M. Chang, "Csnas: Contrastive self-supervised learning neural architecture search via sequential model-based optimization," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 4, pp. 609–624, 2021.
[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.
[19] C. D. Putnam, R. D. Kolodner *et al.*, "Dna mismatch repair: Mechanisms and cancer genetics," 2019.
[20] M.-J. Nokin, J. Bellier, F. Durieux, O. Peulen, G. Rademaker, M. Gabriel, C. Monseur, B. Charloteaux, L. Verbeke, S. van Laere *et al.*, "Methylglyoxal, a glycolysis metabolite, triggers metastasis through mek/erk/smad1 pathway activation in breast cancer," *Breast Cancer Research*, vol. 21, no. 1, pp. 1–19, 2019.
[21] T. Xu, J. Li, Z. Wang, X. Zhang, J. Zhou, Z. Lu, L. Shen, and X. Wang, "Real-world treatment and outcomes of patients with metastatic braf mutant colorectal cancer," *Cancer Medicine*, 2023.
[22] P.-P. Wang, C. Lin, J. Wang, G. A. Margonis, and B. Wu, "Braf mutations in colorectal liver metastases: Prognostic implications and potential therapeutic strategies," *Cancers*, vol. 14, no. 17, p. 4067, 2022.
[23] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, 2022.

APPENDIX

Implementation

```
class PathwayCapsule(nn.Module):
  def __init__(self, in_dim, hid_dim,
    n_cap_layers, dropout):
    super(PathwayCapsule, self).__init__()
    self.dropout = dropout
    self.fc_in = nn.Linear(in_dim, hid_dim)
    self.layers = nn.ModuleList([nn.Linear(
    hid_dim, hid_dim)
                  for _ in range(
    n_cap_layers)])
    self.fc_out = nn.Linear(hid_dim, 1)
  def forward(self,x):
    x = self.fc_in(x)
    for f in self.layers:
      x = f(x) + x
      x = nn.Dropout(self.dropout)(x)
      x = nn.ReLU()(x)
    x = self.fc_out(x)
    return nn.Sigmoid()(x)
```

Listing 1: **Forward-Pass of Pathway-Capsule**

```
class Model(nn.Module):
  def __init__(self, in_dim, hid_dim, n_caps
    , n_cap_layers, out_dim, dropout = 0.):
    super(Model, self).__init__()
    self.in_dim, self.n_caps = in_dim,
    n_caps
    self.dropout = dropout
    self.capsules = nn.ModuleList([
    PathwayCapsule(in_dim, hid_dim,
    n_cap_layers,dropout) for _ in range(
    n_caps)])
    self.cls_head = nn.Linear(in_dim +
    n_caps, out_dim, bias = False)
  def forward(self, x):
    x = x.float()
    sx = []
    for f in self.capsules:
      sx.append(f(x))
    H = torch.cat(sx, dim = 1)
    Hc = torch.cat([x,H], axis = 1)
    y = self.cls_head(Hc)
    return H, y
```

Listing 2: **Forward-Pass of PCNN.**



Phuong-Nam Nguyen is a current Ph.D. student in Electrical Engineering at the University of South Florida. He earned a Bachelor's degree in Mathematics and Education from the Hanoi University of Education and a Master's in Statistics from the University of South Florida. He is currently a lecturer and research fellow at Phenikaa University. His research interests are applying classical and quantum machine intelligence to biological problems, including oncology research, molecular dynamics modeling, large genome data analysis, and biomarker identification.
Email: nam.nguyenphuong@phenikaa-uni.edu.vn

Multi-task Learning Model for Detecting and Filtering Internet Violent Images for Children

Le Kim Hoang Trung[1], Nguyen Van Thanh Vinh[1], Phan Le Viet Hung[1], and Nguyen Huu Nhat Minh[1]
[1] The University of Danang, Vietnam - Korea University of Information and Communication Technology

Correspondence: Nguyen Huu Nhat Minh, nhnminh@vku.udn.vn
Received: 15/4/2024, revised: 14/5/2024, accepted: 26/5/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1261

Abstract: The Internet has emerged as an essential daily information access, but exposing children to inappropriate content can impair their early development. Existing content filtering methods exhibit limitations in accurately and efficiently detecting diverse inappropriate internet content. In this paper, we propose a multi-task learning model for detecting and filtering violent images to provide safer online experiences. The multi-task model is developed from the pre-trained lightweight base model such as MobileNetv2 to enable proper integration within web browser extensions. Pure training to detect violent images could raise false alarms in the classification results when the landscape or object images don't contain any human, hence we develop two joint learning tasks such as detecting humans and detecting violent images simultaneously. Our experiments demonstrate that the proposed multi-task approach with binary rule achieves 98.5% accuracy, outperforming the single-task model for detecting violent images by a margin. Thereafter, the multi-task model is also integrated into the web extension to detect and filter out violent images to prevent children from harmful content.

Keywords: *multi-task learning, violence detection, Web extension.*

Tiêu đề: **Mô hình đa nhiệm nhận diện và lọc hình ảnh bạo lực cho trẻ em**

Tóm tắt: Internet đã trở thành một phương tiện truy cập thông tin thiết yếu hàng ngày, nhưng việc trẻ em tiếp xúc với nội dung không phù hợp có thể gây hại cho sự phát triển ban đầu. Các phương pháp lọc nội dung hiện có gặp hạn chế trong việc phát hiện chính xác và hiệu quả các nội dung không phù hợp đa dạng trên Internet. Trong bài báo này, chúng tôi đề xuất một mô hình đa nhiệm để phát hiện và lọc hình ảnh bạo lực nhằm cung cấp trải nghiệm trực tuyến an toàn hơn. Mô hình đa nhiệm này được phát triển từ mô hình nền tảng nhẹ đã được huấn luyện trước như MobileNetv2 để có thể tích hợp hợp lý vào các tiện ích mở rộng của trình duyệt web. Việc huấn luyện để phát hiện hình ảnh bạo lực có thể gây ra cảnh báo sai trong kết quả phân loại khi các hình ảnh phong cảnh hoặc vật thể không chứa bất kỳ con người nào, do đó chúng tôi phát triển hai nhiệm vụ học liên kết như phát hiện con người và phát hiện hình ảnh bạo lực đồng thời. Kết quả thí nghiệm cho thấy đề xuất mô hình đa nhiệm với quy tắc nhị phân đạt độ chính xác 98.5%, vượt trội so với mô hình đơn tác vụ trong việc phát hiện hình ảnh bạo lực. Sau đó, mô hình này cũng được tích hợp vào tiện ích mở rộng của trình duyệt để phát hiện và lọc bỏ hình ảnh bạo lực nhằm ngăn chặn trẻ em tiếp xúc với nội dung có hại.

Từ khóa: *học đa nhiệm, nhận diện bạo lực, tiện ích mở rộng Web*

I. INTRODUCTION

As technology advances rapidly, Internet access plays an increasingly vital role in human life. For children, the Internet provides vast information access including educational, entertainment, and social communication resources. However, along with these benefits, there is a worrying proliferation of violent images circulating uncontrollably, posing a threat to children. Exposure to such content can negatively impact the early development of children. Hence, there is an urgent need for effective measures to protect children from exposure to inappropriate images on the Internet.

Statistics show that 71% of children aged 15-17 in developed countries have accessed websites with violent images [1]. Exposure to negative images and videos can lead to severe consequences such as depression, increased bullying tendencies, and potentially irreversible damage to brain structure and cognition capabilities, especially given their developing brains. Despite concerted efforts by governmental and non-governmental entities to address this

predicament, existing solutions still lack the capacity for automated detection and filtering of risky content. Due to these technical gaps, children still readily access images and videos that are violent. Given these disturbing vulnerabilities, this research puts forward an AI system capable of identifying and obscuring negative elements in visual media. Integrating the model into web browsers would obstruct the proliferation of harmful materials, thereby safeguarding future generations from the detrimental influences of the online world. The proposed approach leverages intelligent algorithms to construct a bulwark against age-inappropriate content, aiming to uphold the safety of children during their digital encounters. As they traverse the largely unregulated online landscape, young users require robust protection from materials that can severely impact their developmental trajectory. By preemptively barring such exposures, this system endeavors to supplement ongoing initiatives focused on fostering a benevolent internet environment tailored uniquely to children's needs.

To address this challenge, this research has endeavored to build a deep learning model that has the ability to detect feature extraction from violent images, with a focus on safeguarding the online experience for children. By leveraging technological advancements and drawing insights from existing research, this research paper aims to contribute to ongoing efforts to create a safe, healthy environment for children from violence at an early age. Through meticulous research and development, this study aims to enhance our understanding of methods for filtering inappropriate images and provide practical solutions to mitigate risks associated with children's exposure to harmful online content. Additionally, the learning model will be integrated into a web extension capable of effectively recognizing and filtering inappropriate images for children is intended to promote positive online experiences for future generations.

There have been efforts and solutions to limit the prevalence of violent and negative images and video content online, but identifying all such images is a difficult problem to solve. New images can be continuously posted every second, and it's impossible to know if they are safe for children. This paper aims to address the following questions such as:

- How to develop a deep learning-based image classification model with high accuracy in detecting violent images?

- If integrated into a browser extension, can the model automatically blur/block inappropriate images in real time?

Objectives and contributors:

The main objective of this research is to develop a multi-task learning model capable of classifying and identifying violent images on the Internet. The proposed model employs the pre-trained MobileNetv2 model as the shared layers, trained on two learning tasks with specific-task layers for detecting human as well as violent images. The learning model increases recognition capabilities and will be integrated into web browser extensions to automatically blur or block inappropriate images from children by training and collection of Existing image data from the internet.

In evaluating the effectiveness of the solution, the research will be contributed by the process of collecting image data on violent images and labeling the dataset. Additionally, the design and training of the neural network model through multiple times to improve performance. Moreover, the development of the extension and evaluation through multiple testing experiments also contributes significantly to the research process.

This initiative aspires to set new standards for protecting children from the hazards latent within the internet's vast unsupervised terrain. The envisioned automation of visible content moderation through an AI-based approach aims to alleviate the damages wrought by inadequate safeguards while upholding youthful innocence in the digital age.

II. RELATED WORKS

In this section, we provide an overview of existing studies and research related to developing systems for detecting and filtering violent images on the Internet, particularly in relation to children. This section explores the approaches, techniques, challenges, and limitations identified by previous research. Additionally, it serves as a basis for analyzing these studies and how they inform the methodology of the current paper.

In existing research, there were different approaches to this topic. Research by David Alexander on the use and effectiveness of Artificial Intelligence on internet websites shows the future possibilities and promising results in applying the many potentials of Artificial Intelligence [2]. In addition, there is research on the use of AI in work using violent images to threaten children and also focuses on techniques using Deep learning in image recognition on social networking platforms [3]. Identifying violent images by dividing them into multiple cases the identification of violent images provides a higher level of accuracy and coverage than the research of Muhammad Ramzan [4].

The recent methods for detecting violent images include feature-based methods and neural network-based methods. Feature-based methods [5] focus on identifying specific image points or possibly sounds within images/videos related to violence. On the other hand, neural network-based methods concentrate on building and developing flexible neural networks to understand the characteristic relationships within image data. Neural network-based methods generally yield better results compared to feature-based

methods, but it is worth noting that they require a large amount of complex and accurate training data.

We also have approached many software available on the software market and researched the methods they have used such as NetNanny [6], CyberSister[7], MaxProtect[8], etc. most of which are still limited in flexibility, active and accessible web extension browsers. In the existing paper, we explore and investigate various methodologies of inappropriate image detection to gauge aspects of viability and application in real-world contexts. Specifically, we analyze methodologies including URL Blacklists/Whitelists. In URL lists maintaining dynamically updated forbidden URL datastores via web crawlers enables preemptive filtering by browser cross-referencing requested pages against known banned domains. Benefits include real-time detection and easy integration into existing platforms [9].

Image recognition algorithms are also used with array models instead of pixels, they give better results than modern algorithms that model pixels, whereas the adult Web page bag recognition is carried out using multi-instance learning based on the combination of classifying texts, images, and videos in Web pages, Both the speed and the accuracy for recognizing the Web adult content are increased, in contrast, to detect Web pages one-by-one and also called Hybrid Text/ Metadata Heuristic, in every content supplementary signals like text captions, webpage markup, and origin domains assist violent/explicit detection when used adjunctively with computer vision. This fusion approach counterbalances blind spots between modalities yet requires more complex orchestration [10]. There are also content filtering and detection techniques based on Chroma to detect image content, so this approach follows classic machine learning methods including RandomForest, NeuralNetwork. With "Adult & Safe" images classified and accuracy up to 88% [11]

Challenges and Limitations of Existing Methods:

We are always aware that detecting sensitive and violent content on the internet is very important to protect users, especially children, from harmful experiences online. However, current methods face many challenges and limitations that affect the research and development process. The first challenge encountered is the complexity of images and video content, the constant and diverse appearance of training data, and the ability to accurately identify all cases of violent images due to their complexity and diversity. For example, the image of two people rubbing each other's heads also makes it difficult for the model to know whether it is an act of love or violence.

The next challenge is that if you want the model to be accurate then the work that comes with it. It is the training data that must be suitable for each classification and requires a large amount of data, which causes difficulty in training and evaluating the model and slows it down significantly. The next step is to apply them to the web extension. This must ensure the model is small enough for processing speed to be almost real-time, ensuring accurate model recognition along with high processing speed. Therefore, the biggest challenge for this research is the proposed model for web extensions must be lightweight, high accuracy, and suitable for almost all devices.

III. DATASET CONSTRUCTION

1. Public Datasets

The "Public Datasets" section of this paper incorporates several publicly available datasets essential for training and evaluating human activity and violence detection systems. We utilized two public datasets as follows:

- **Real Life Violence Situations Dataset [12]**
  This dataset, introduced by Soliman et al., comprises 1000 violent and 1000 non-violent videos sourced from YouTube. The violent videos encompass real street fight scenarios in diverse environments and conditions, while the non-violence videos depict various human actions such as sports, eating, and walking. We extracted five random frames from those videos, however, this is limited by a number of scenarios and is not enough to produce a high-performance detection model in diverse practical scenarios.
- **Human Action Recognition (HAR) Dataset [13]**
  The HAR Dataset, available on Kaggle, provides a collection of $12,000$ images depicting various human actions. However, we have only selected $6,000$ images. This dataset is added to provide more images for normal situations without violence.

These datasets play a pivotal role in the development and assessment of the proposed system, ensuring its efficacy in detecting and filtering violent images on the internet, particularly with a focus on children. Therefore, we collectively construct a preliminary dataset including $10,000$ samples for violent and $3,000$ for normal human action.

2. Manual Collection

While public datasets served as a valuable starting point, we recognized that they had significant limitations in diversity and coverage of realistic scenarios. Labeled images from public datasets, though useful, did not fully capture the wide variability of violent images propagated through social media. To complement the public data, we undertook a manual collection process focused on gathering samples from major social platforms. This allowed us to obtain

more varied and representative examples reflective of actual online content.

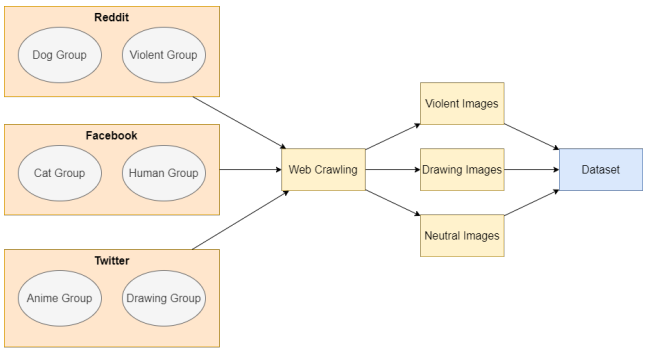


Figure 1. How web crawling work

We manually crawled additional samples from three public social media platforms such as Facebook, Reddit, and Twitter that had violent images or others as shown in Figure 1. We accessed existing online social network groups to collect more samples in a variety of realistic images from many kinds of groups. For instance, to collect more violent images we searched for the group titled "Fight Situations" and downloaded all shared images within those groups for around 2 months. All images from the "Fight Situations" Facebook group were annotated with the label “Violent". On the other hand, the normal cases include groups containing animals, nature, anime, and drawing pictures. As a result, we manually added $5,000$ normal, $2,000$ violent into our dataset containing $23,000$ images. While automatically crawling social media enabled the collection of a large volume of labeled image data, however, we detected that it contained significant noise in automatic labeling. For example, the "Fight Situations" group likely contained some nonviolent images that would be incorrectly labeled as “Violent”.

Ultimately, we accepted this trade-off because social media provided access to an exceptionally large and diverse range of image data. Our strategy for dataset collection made the learning model more robust compared to training only on limited public datasets. Hence, diverse data sources enabled us to develop a deep learning model with superior generalization ability for detecting violent images in practical internet images.

IV. PROPOSED METHOD

1. Multi-task learning approach

Multi-task learning [14] trains a model to perform multiple related learning tasks simultaneously. This technique provides several advantages over specialized single-task models. By sharing representations between learning tasks, the model learns generalized features that benefit all of the tasks. Training signals from the multiple objectives also regularize the model and prevent overfitting. Additionally, multi-task models require fewer parameters than multiple independent single-task models, improving efficiency.

To enhance our detection of violent images, we employ a multi-task learning approach for detecting human appearance as well as violence simultaneously. The learning model architecture shares base layers such as convolutional layers in MobileNet [14] that extract generalized visual features for both learning tasks. In that way, detecting human appearance as a learning task helps to improve violence detection in two aspects. *Firstly*, it increases the generalization of the detection model focus on human actions. *Secondly*, the output from human appearance detection helps to correct misclassified normal pictures due to our limited datasets. Furthermore, the proposed multi-task learning approach leverages the MobileNetv2 model [15], which has been pre-trained on the ImageNet dataset as the transfer learning approach[15]. The pre-trained MobileNetv2 was chosen due to the lightweight design that makes it well-suited for integration into web extensions for existing browsers (e.g., Chrome, Edge, Brave) to provide real-time filtering on the client side.

The model will be continued training on our collected dataset from public datasets and manual collection. After passing through the MobileNet layers for feature extraction, the features will be fed into separate task-specific layers for each class (violence, human). These layers will produce predictions on whether the image has violence or human figures as shown in Figure2. The reason we included the human appearance detection task is that it provides additional features related to the presence of humans in the images, which aids the model in detecting violent actions involving humans. Also, the output from human appearance detection helps to correct the wrong prediction from violent detection learning tasks. This often happens in realistic unseen natural scenes or normal animal pictures that do not exist in the training datasets.

Specifically, the predictions from the two output layers will be combined to determine the final action of blurring or not blurring the image. The predictions will be combined and represented as a binary string “ab":

- “a" indicates violence, “0" if the image is violent, else “1"
- “b" indicates human presence, “0" if the image contain human, else “1"

Based on the predefined **binary rule**, images predicted as “11" will be blurred, while images predicted as “00", “01", or “10" will remain unblurred. This multi-task approach allows the model to leverage inter-related signals between

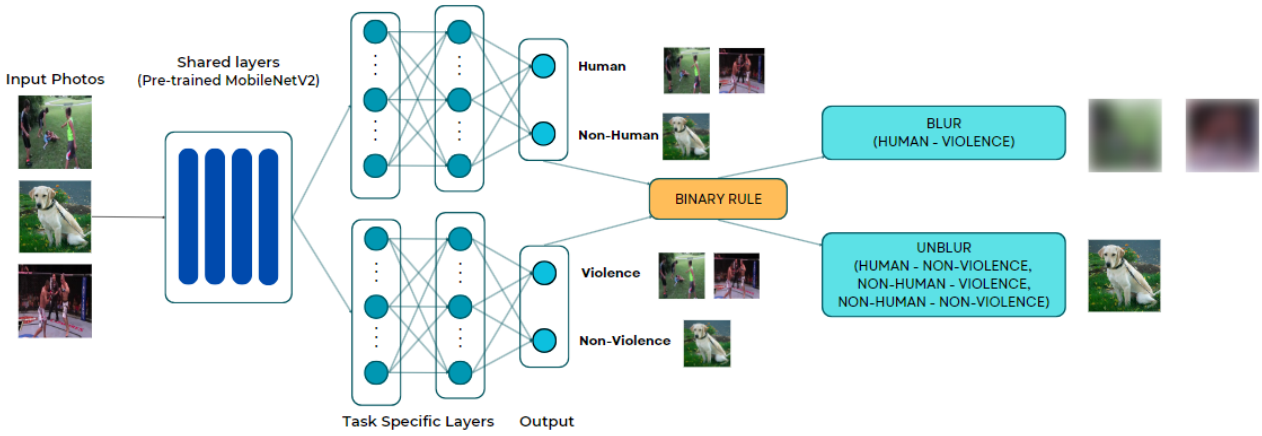


Figure 2. The proposed multi-task learning approach.

detecting violence, and humans in images to improve the performance of identifying images that should be blurred. Thereby, the proposed method ultimately boosts the robustness of the learning model as shown in our evaluation section.

V. RESULTS

We evaluated the performance of our proposed multi-task learning model on a dataset of 23,000 images. The dataset consisted of 10,000 images for violent or non-violent detection and 13,000 images for detecting human appearance. The non-human images include a variety of objects (such as drawings, animals, flowers, etc.).

In training our multi-task learning model, we utilized the **categorical cross-entropy** loss function for both the human detection and violence detection tasks. This choice of loss function was crucial in optimizing the model's performance across both tasks. The categorical cross-entropy effectively measures the divergence between the predicted probabilities and the true class labels, ensuring that the model learns to distinguish accurately between categories. By summing this loss function from each task, we ensured consistent and reliable training outcomes, which contributed to the high accuracy levels observed in our evaluations.

$$\mathcal{L}_{CE} = -\sum_{i=1}^{classes} y_i \log(\hat{y}_i) \quad (1)$$

where $\hat{y}$ is the predicted result and y_i is the true label.

The dataset was splitted into 70/30 for training/testing data. For the violence detection task, the proposed multi-task model achieved an accuracy of **97.6%** while the human detection task obtained an accuracy of **96%** as shown in Table I. To evaluate the performance of the multi-task model against single-task models, we trained separate models for violence detection using the same dataset.

The single-task violence detection model achieved an accuracy of 95.2% when using the additional collected data while obtaining 94% with two public datasets. Hence, this result demonstrates that the multi-task learning approach, by sharing representations between related tasks, improves performance over single-task models by 2.4%. Thereby, we adopted the binary rule to enhance the robustness of the proposed approach and obtain the highest accuracy performance i.e., 98.5%. Even though we did not count exactly in practice, the **binary rule** exhibits consistently improving the robustness of detecting realistic images on social networks.

To conclude, we showed that using multi-task learning to predict the relation between human presence and violent images enhanced the accuracy of detecting violent images, unlike models that use only one task. The model achieved promising results on the test set, indicating its potential for integration into web browsers as a client-side filtering extension.

VI. WEB EXTENSION INTEGRATION

To enable real-time filtering of violent images during web browsing, we developed a web extension integrating our trained multi-task learning model. The extension is implemented for all browsers with Chromium core by using Chrome Extensions API [16].

The extension loads the file of our model locally within the web browser for inference. For every web page that users visit, the extension will scan and multi-task learning model evaluates each appeared image and decides whether to blur it or not. If the decision is "blur", the extension dynamically modifies the image tagged as inappropriate by applying a blur effect, else does nothing. This requires fast interaction to guarantee a low delay for the users. Only those visible images to the users will be detected and blurred iteratively.

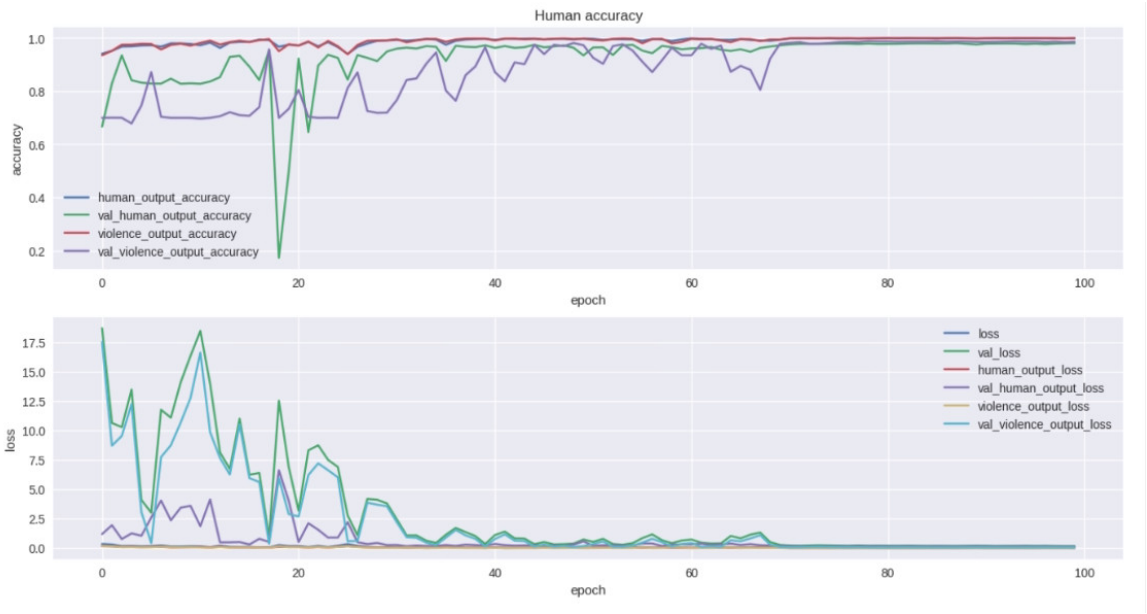


Figure 3. The accuracy performance and loss of training and validation for human appearance detection task and violent detection task.

Table I
TESTING ACCURACY OF LEARNING TASKS USING MULTI-TASK LEARNING AND SINGLE-TASK LEARNING APPROACHES. **SINGLE-TASK-PD** IS THE SINGLE-TASK MODEL USING ONLY TWO PUBLIC DATASETS WITHOUT ADDITIONAL DATA. AND THE PROPOSED **MULTI-TASK-B** APPROACH USES THE MULTI-TASK MODEL WITH THE ADDITIONAL BINARY RULE.

Learning Task	Single-Task	Single-Task-PD	Multi-Task	Multi-Task-B
Violence Detection	95.2%	94%	97.6%	**98.5%**
Human Detection	-	-	**96.4%**	**96.4%**

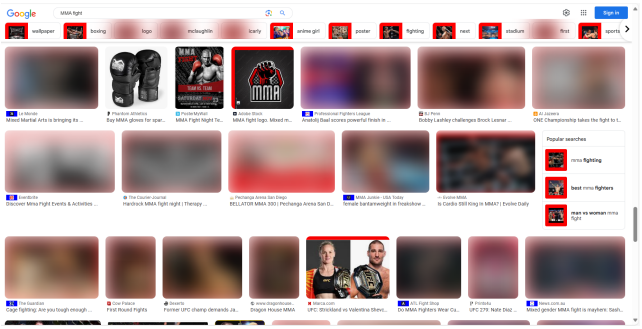


Figure 4. Extension automatically filter “violent" images

The extension necessitates efficient processing times to ensure a seamless user experience. The running time for the model to predict whether an image contains violent content is approximately **2 seconds**. However, due to the concurrent processing capability of the extension, this time is optimized. For the first image, the model takes 2 seconds, but subsequent images benefit from this concurrency. For instance, the second image is processed at 2.4 seconds, the third at 2.8 seconds, and so on. This overlapping processing mechanism ensures that while the first image sets the initial processing time, each subsequent image only adds an incremental 0.4 seconds to the total processing time, thus maintaining efficient and rapid content filtering.

In summary, the extension integrates efficient client-side filtering to block violent images, creating a safer browsing experience. Through minimal filtering, the extension creates a safer online space for underage users, while upholding unconstrained availability of content across the open internet.

VII. CONCLUSION

Through the research and development process for building a new tool for protecting children, we have developed a multi-task learning model capable of detecting and filtering images with violent characteristics regarding the appearance of humans in the images. We first collect sufficient data in many realistic cases to extend the public datasets. The proposed method improves the performance of models regarding diverse cases compared to the single-task model. Thereafter, we successfully applied the model with the binary rule for the web extension to provide flexible and safe management capabilities for children.

Although we have achieved promising results, we recognize that there are still potential research directions that could be further explored such as continuing to increase realistic training data. We advocate the necessity of combining the detection of sensitive images into our model to provide safer online environments for children.

ACKNOWLEDGEMENT

This work is conducted under the support of Vietnam - Korea University of Information and Communication.

REFERENCES

[1] Unicef-annual-report-2021, https://www.unicef.org/reports/unicef-annual-report-2021, Last accessed 21 Feb, 2024.

[2] Van Bruwaene, D., Huang, Q. & Inkpen, D. "A multi-platform dataset for detecting cyberbullying in social media", *Lang Resources & Evaluation*, 54, 851–874, 2020.

[3] Van Bruwaene, David, Qianjia Huang, and Diana Inkpen, "A multi-platform dataset for detecting cyberbullying in social media", *Language Resources and Evaluation*, 54, 851-874, 2020.

[4] Ramzan, Muhammad, Adnan Abid, Hikmat Ullah Khan, Shahid Mahmood Awan, Amina Ismail, Muzamil Ahmed, Mahwish Ilyas, and Ahsan Mahmood, "A review on state-of-the-art violence detection techniques", *IEEE Access*, 7: 107560-107575, 2019.

[5] Mumtaz, N., Ejaz, N., Habib, S., Mohsin, S. M., Tiwari, P., Band ,"An overview of violence detection techniques: current challenges and future directions", *Artificial intelligence review*, 56(5), 4641-4666, 2023

[6] Netnanny, https://www.netnanny.com/, Last accessed 21 Feb, 2024

[7] Cybersitter, https://www.cybersitter.com/, Last accessed 21 Feb, 2024

[8] Maxprotect, https://www.maxprotect.com/, Last accessed 21 Feb, 2024

[9] M. Hammami, Y. Chahir and L. Chen, "WebGuard: Web based adult content detection and filtering system," *In Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), Halifax, NS, Canada*, pp. 574-578, 2003.

[10] Hu, Weiming, Haiqiang Zuo, Ou Wu, Yunfei Chen, Zhongfei Zhang, and David Suter. "Recognition of adult images, videos, and web page bags", *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 7*, no. 1: 1-24, 2011

[11] Sharma, Preeti, Manoj Kumar, and Hitesh Sharma, "Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation," Multimedia Tools and Applications 82, no. 12,: 18117-18150, 2023.

[12] Soliman, Mohamed Mostafa, Mohamed Hussein Kamal, Mina Abd El-Massih Nashed, Youssef Mohamed Mostafa, Bassel Safwat Chawky, and Dina Khattab, "Violence recognition from videos using deep learning techniques," *In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 80-85, 2019.

[13] Human Action Recognition (HAR) Dataset. https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset, Last accessed 21 Feb, 2024

[14] Caruana, "Rich Multitask learning," *Machine learning*, 28 (1997): 41-75.

[15] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510-4520, 2018.

[16] Google Chrome extension API, https://developer.chrome.com/docs/extensions, Last accessed 21 Feb, 2024.



Le Kim Hoang Trung is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: trunglkh.21it@vku.udn.vn



Nguyen Van Thanh Vinh is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: vinhnvt.21it@vku.udn.vn



Phan Le Viet Hung is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: hungplv.21ad@vku.udn.vn



Nguyen Huu Nhat Minh (M'20) received Ph.D. degree in Computer Science and Engineering from Kyung Hee University, South Korea, in 2020. He continued Post-Doc with Federated Learning and Democratized Learning at Intelligent Networking lab, Kyung Hee University, South Korea. He is Deputy Head of Department of Science, Technology, and International Cooperation, and In charge of Research Program at Digital Science and Technology Institute, The University of Danang – Vietnam - Korea University of Information and Communication Technology, Vietnam. His research interests include wireless communications, federated learning, NLP, and computer vision.
Email: nhnminh@vku.udn.vn

Omni-Dimensional Adaptation for MobileNetV3 using Bayesian Hyperparameter Tuning

Vo Long, Phan Nhat Quang, Tran Van Dat, Dang Duc Long
VN-UK Institute for Research and Executive Education - the University of Danang, Danang, Vietnam
Correspondence: Dang Duc Long, long.dang@vnuk.edu.vn
Received: 29/2/2024, revised: 20/4/2024, accepted: 3/6/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1267

Abstract: This paper proposes enhancing MobileNetV3 with Omni-Dimensional Dynamic Convolution (OD-Conv) to overcome CNNs' limitation of static convolution kernels. OD-Conv introduces multi-dimensional attention to adjust convolution kernels across spatial, input channel, output channel, and number of kernels dimensions, improving feature representation. Bayesian Optimization optimizes hyperparameters efficiently. Experiments show Omni-MobileNetV3 outperforms MobileNetV3 on CIFAR-100, Tiny ImageNet, and medical image datasets, achieving up to 3% accuracy gain while maintaining efficiency. This dynamic convolution method combined with Bayesian tuning achieves state-of-the-art results in image classification

Keywords: *Omni-dimensional convolution, MobileNetV3, Bayesian optimization*

Tiêu đề: **Sử dụng tối ưu Bayes để tối ưu siêu tham số cho mô hình Omni-Dimensional MobiletNetV3**

Tóm tắt: Bài báo này đề xuất việc cải tiến MobileNetV3 với Omni-Dimensional Dynamic Convolution (OD-Conv) nhằm khắc phục hạn chế của các mạng nơ-ron tích chập (CNN) về các kernel tích chập tĩnh. OD-Conv giới thiệu sự kết hợp đa chiều để điều chỉnh các kernel tích chập trên các chiều không gian, kênh đầu vào, kênh đầu ra và số lượng kernel, cải thiện khả năng biểu diễn đặc trưng. Phương pháp Tối ưu hóa Bayes (Bayesian Optimization) được sử dụng để tối ưu hóa các siêu tham số một cách hiệu quả. Thí nghiệm cho thấy Omni-MobileNetV3 vượt trội hơn MobileNetV3 trên các bộ dữ liệu CIFAR-100, Tiny ImageNet và hình ảnh y tế, đạt được mức tăng độ chính xác lên tới 3% trong khi vẫn duy trì hiệu quả. Phương pháp tích chập động này kết hợp với tối ưu hóa Bayes đạt được kết quả tiên tiến nhất trong phân loại hình ảnh.

Từ khóa: *Omni-dimensional convolution, MobileNetV3, tối ưu Bayes*

I. INTRODUCTION

Convolutional Neural Networks (CNNs) have proven to be a valuable tool in various domains, including computer vision, natural language processing, and signal processing. The development of influential models, such as ResNet [1], Inception [2, 3], and MobileNet [4–6], has played a significant role in advancing the capabilities of computer vision systems.

At the core of CNNs lies the convolution layer, a fundamental operation that employs fixed, pre-defined kernels to extract features from input data. While undeniably effective, the static kernel does not perform well with the data outside of the training distribution[7, 8], which underscores the need for continuous innovation in CNN architectures.

To address this limitation, recent research has explored dynamic convolution techniques [9–11] aiming to learn the spatially-adaptive filters for each convolution operation, which improves the network's ability to adapt to complex patterns. However, existing dynamic convolution approaches often focus on specific dimensions of the kernel space, such as the spatial dimensions (height and width) or the input channel dimensions [9–11].

This paper proposes a novel approach, Omni-Dimensional Dynamic Convolution (ODConv), which introduces a multi-dimensional attention mechanism to dynamically adjust convolution kernels across all four dimensions of the kernel space:

- Spatial dimensions: Height and width of the kernel
- Input channel dimension: Number of channels in the input data

- Output channel dimension: Number of channels in the output feature map
- Number of kernels: Total number of kernels in the layer

However, optimizing such multi-dimensional dynamic convolution (ODConv) presents a significant challenge due to the vast hyperparameter space. Traditional hyperparameter tuning methods, like grid search or random search, can be inefficient and computationally expensive in such high-dimensional scenarios. This is where Bayesian optimization emerges as a crucial tool [12].

By leveraging past evaluations and incorporating prior knowledge, Bayesian optimization offers a more efficient and targeted approach to finding the optimal hyperparameter configuration for the ODConv layer. This can lead to significantly improved performance compared to traditional methods, unlocking the full potential of dynamic convolutions and paving the way for more powerful and adaptable CNN architectures. There are two main aspects of our contributions in this paper:

1) We apply ODConv to replace the ordinary convolutional block of MobileNetv3.
2) We explore the application of Bayesian optimization with ODConv for the new architecture's hyperparameter tuning.

II. RELATED WORK

1. Lightweight networks

In the realm of research, there has been a notable surge in the exploration of lightweight networks, emphasizing the reduction of complexity and parameters [1–4, 13]. Particularly, renowned examples, such as MobileNet, SqueezeNet [14], and ShuffleNet [15, 16], have garnered substantial attention, symbolizing the forefront of this research trend. SqueezeNet [14] introduces an innovative approach by advocating for the replacement of conventional 3x3 convolution kernels with 1x1 kernels in a "splicing" technique. This method strategically trims parameters within the current model architecture, resulting in a notable reduction in model size without compromising its functionality

2. Dynamic Weight Networks

Dynamic convolution techniques, like DyConv [9] and CondConv [10], are replacing static filters in CNNs. These methods dynamically adjust filters based on the input data, allowing the network to learn more complex features. Building on this, ODConv [17] uses a multi-dimensional attention mechanism to further enhance dynamic convolutions by adjusting filters across all their dimensions.

3. Hyperparameter Optimization

Hyperparameter optimization [18] is crucial for maximizing machine learning model performance. While traditional methods like random and grid search provide a starting point, they can be computationally inefficient. Advanced techniques such as Bayesian optimization [12] and evolutionary algorithms offer more sophisticated approaches for exploring the hyperparameter space. The comparison is illustrated in the Table I

Additionally, Neural Architecture Search (NAS) [19, 20] has the potential to improve model performance by automating architecture selection. However, NAS typically requires significant computational resources and may overfit certain datasets. Despite these challenges, these techniques collectively contribute to enhancing the effectiveness of machine learning models.

III. BACKGROUND

1. MobileNetV3

MobileNetV3 [6], a convolutional neural network (CNN), is tailored for mobile and embedded devices, balancing efficiency and accuracy. It builds upon MobileNetV1 [4] and MobileNetV2 [5], integrating hardware-aware architecture search (NAS) [6] for optimal performance across platforms. Notable features include Squeeze-and-Excitation [21] (SE) blocks for dynamic feature recalibration and hard-Swish activation for computational efficiency instead of ReLU6. Available in Large and Small versions, it caters to diverse use cases, prioritizing either accuracy or computational savings. MobileNetV3 facilitates robust deep-learning applications on resource-constrained devices, outlined in Figure 1, which illustrates its architecture: starting with a 1x1 convolutional layer, followed by depth-wise convolution with a 3x3 kernel size. Subsequent steps include SE excitation for enhanced feature representation, followed by a 1x1 convolutional layer to reduce parameters and computations, ultimately generating the output category.

2. Omni-dimentional Dynamic Convolution (ODConv)

Omni-dimensional Dynamic Convolution (ODConv) [17] presents a novel perspective on dynamic convolutional kernels within Convolutional Neural Networks (CNNs), intending to bolster the representation capabilities of deep CNNs. In contrast to conventional static convolutional kernels, which learn a singular kernel within each convolutional layer, ODConv introduces a multi-dimensional attention mechanism. This mechanism enables the learning of varied attentions for convolutional kernels across all four dimensions of the kernel space simultaneously. This process is depicted in the Figure 2.

Table I
COMPARISON OF HYPERPARAMETER OPTIMIZATION METHODS.

Feature	Grid Search	Random Search	Bayesian Optimization
Methods	Exhaustive	Random Sampling	Statistical Modeling
Computational Cost	High	Low	Moderate
Efficiency	Low	Moderate	High
Ease of Implementation	Easy	Easy	Moderate

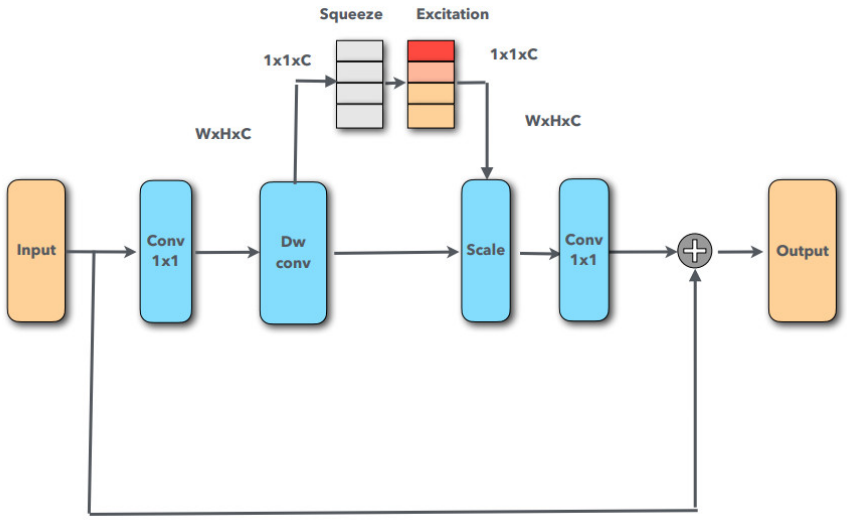


Figure 1. Structure of MobileNetV3 network model.

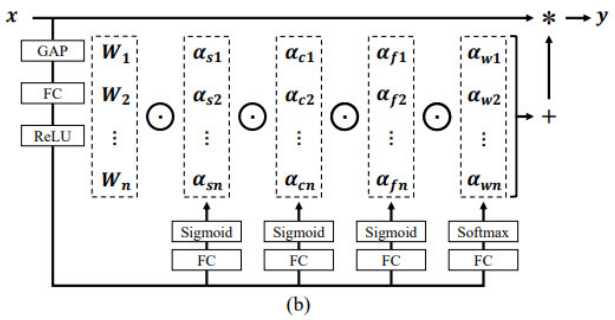


Figure 2. Schematic of Omni Dimensional Dynamic Convolution

a) Formulation for Omni-Dynamic Convolution

$$y = \left(\sum_{i=1}^{n} \alpha_{wi} \odot \alpha_{fi} \odot \alpha_{ci} \odot \alpha_{si} \odot \mathbf{W}_i \right) \mathbf{X} \qquad (1)$$

where $\alpha_{w_i} \in \mathbb{R}$ represents the attention scalar for the convolutional kernel $\mathbf{W}_i$; $\alpha_{s_i} \in \mathbb{R}^{k \times k}$, $\alpha_{c_i} \in \mathbb{R}^{c_{\text{in}}}$ and $\alpha_{f_i} \in \mathbb{R}^{c_{\text{out}}}$ are denoted as the attention for the spatial dimensions, the input channel dimensions, and the output channel dimensions for the kernel space of the convolution kernel $\mathbf{W}_i$ respectively. The operator $\odot$ indicates the element-wise multiplication across different dimensions of the kernel space.

b) Learning mechanism of ODConv

In ODConv [17], each convolutional kernel $\mathbf{W}_i$ is enhanced with four types of attention mechanisms (1) α_{si} allocates varying attention scalars to convolutional parameters (per filter) across $k \times k$ spatial locations. (2) α_{c_i} assigns diverse attention scalars to the c_{in} channels of each convolutional filter $\mathbf{W}_i$. (3) α_{fi} distributes distinct attention scalars to the c_{out} convolutional filters. (4) α_{wi} assigns an attention scalar to the entire convolutional kernel $\mathbf{W}_i$. The process of combining these four types of attention across n convolution is illustrated in the image above. ODConv learns four types of attention for convolutional kernels, which are shown to be complementary to each other. By progressively applying attention to the corresponding kernel, ODConv significantly strengthens the representation power of CNNs, leading to improved performance in tasks like image classification. Despite its enhanced feature learning ability, ODConv maintains efficiency by competing with or outperforming existing dynamic convolution counterparts with multiple kernels using just one single kernel. This efficiency is crucial in reducing the number of parameters and model size, making ODConv a practical and effective solution for various CNN architectures.

3. Bayesian Optimization

Bayesian Optimization [12] serves as a valuable method for optimizing objective functions that are expensive to assess, making it particularly well-suited for determining the optimal hyperparameters in deep learning tasks. It builds on a surrogate model for the objective and quantifies the uncertainty in that surrogate using the Bayesian machine learning technique, specifically the Gaussian Process, and then utilize an acquisition function defined from the surrogate model to decide where to sample in the next step.

a) Gaussian Process

Suppose there is a collection of vector points $x_1, \ldots, x_k \in \mathbb{R}^d$ and the function's values according to each point into the vector $[f(x_1), \ldots, f(x_k)]$. The vectors are drawn randomly by the prior probability distribution, The GP regression takes this distribution to be a multivariate normal with a specific mean vector and a covariance matrix.

$$f(x_{1:k}) \sim \mathcal{N}(\boldsymbol{\mu}(x_{1:k}), \boldsymbol{\Sigma}(x_{1:k}, x_{1:k})) \tag{2}$$

where $\boldsymbol{\mu}$ is the mean function that $\boldsymbol{\mu}(x_{1:k}) = [\mu(x_1), \ldots, \mu(x_k)]$ and $\boldsymbol{\Sigma}$ is the covariance matrix which:

$$\boldsymbol{\Sigma}(x_{1:k}, x_{1:k}) = \begin{bmatrix} \Sigma(x_1, x_1) & \cdots & \Sigma(x_1, x_k) \\ \vdots & \ddots & \vdots \\ \Sigma(x_k, x_1) & \cdots & \Sigma(x_k, x_k) \end{bmatrix} \tag{3}$$

b) Acquistion Functions

The acquisition function is a criterion that quantifies the utility of evaluating the objective function at a new point, given the current posterior distribution. It balances the trade-off between exploration (sampling points where the uncertainty is high) and exploitation (sampling points where the expected value is high). We choose *Expected Improvement* (EI), as a measure to identify the expected amount of improvement over the current best-observed value.

The formula for *Expected Improvement* is defined as:

$$\mathrm{EI}_n(x) = E_n[\max(f(x) - f_n^*, 0)] \tag{5}$$

The next point to sample is the one that maximizes the EI, which can be done using standard optimization methods.

$$x_{n+1} = \mathrm{argmax} EI_n(x) \tag{6}$$

IV. DATASET

In this study, we employ a diverse set of datasets to evaluate the performance of our proposed model across various computer vision tasks. The datasets used in our experiments include:



(a) (b) (c)

Figure 3. (a) Scabby apples leaves and healthy apples leaves, (b) Powdery mildew and healthy cherry leaves and (c) Rusted corn leaves and healthy corn leaves.

a) CIFAR-100 [22]

CIFAR-100 extends the complexity with 100 classes, organized into 20 superclasses. Split: The dataset consists of 50,000 training images and 10,000 test images.

b) Tiny ImageNet [23]

A subset of the extensive ImageNet dataset, Tiny ImageNet comprises 200 classes with 500 training, 50 validation, and 50 test images per class. Split: The training set contains 100,000 images, and there are 10,000 images each for validation and testing.

c) Plant Disease [24]

Focused on plant health assessment, this dataset includes 87K RGB images of leaves from three different plant species. It encompasses 38 classes relating to each state of disease of different type of species (apple, blueberry, corn, grape, orange, peach, pepper and strawberry) (Fig 3)

d) Covid-Chest Xray [25]

It has 3 different classes: Healthy chest, COVID, pneumonia. The images in this dataset have different heights and lengths so we have brought it to the same size of 448 pixels for both dimensions in each image. The total number of images in the data set is 6432 images, the training data set is 5144 images and the validation set is 1288 images (refer to Fig 4, 5, 6 for visual representation).

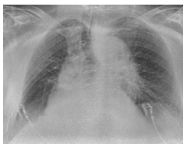


Figure 4. COVID 19



Figure 5. Normal



Figure 6. Pneumonia

V. METHODOLOGY

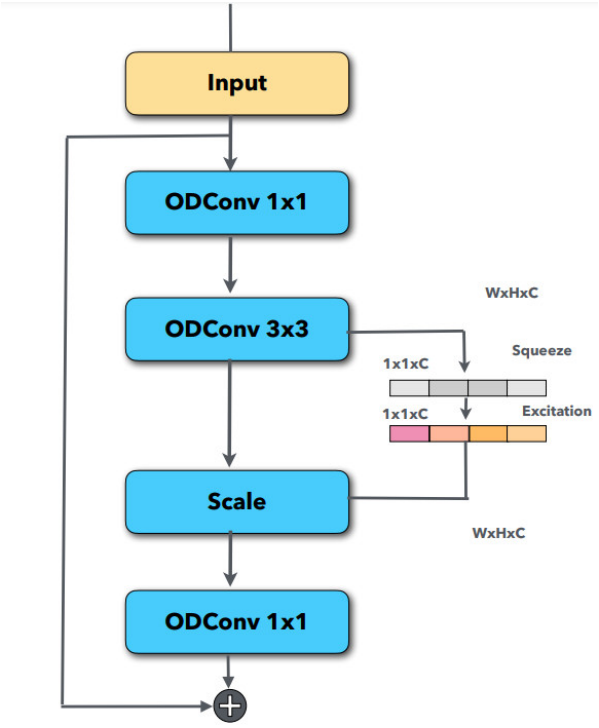


Figure 7. Structure of Omni-MobileNetV3, the combination of Omni-Dimentional Dynamic Convolution and MobileNetV3.

1. Omni-MobileNetV3 Architecture

We introduced a novel multi-dimensional attention mechanism, called "Omni-MobileNetV3", which dynamically adapted the convolutional kernels in each layer to the specific features present in the input. This allowed the network to learn more expressive filters that can better capture the relevant information for the task at hand.

We replaced the traditional convolutions in MobileNetV3 blocks with ODConv to create a hybrid architecture combining several advantages:

- **Improved Feature Extraction**: ODConv's multi-dimensional attention mechanism allowed the network to dynamically adjust convolutional kernels to the specific features in the input. This enabled the network to learn more expressive filters compared to static filters, potentially leading to improved accuracy.
- **Maintained Efficiency**: The core architecture of MobileNetV3, particularly its depthwise separable convolutions and bottleneck blocks, remained intact. This ensured that the computational efficiency and lightweight nature of the original model were preserved.
- **Increased Adaptability**: ODConv's dynamic nature granted the model an increased ability to adapt to diverse datasets and tasks. This flexibility opened doors for broader application of this hybrid architecture.

However, further exploration was needed to fully realize these benefits:

- **Balancing Complexity**: Finding the right balance between the inherent efficiency of MobileNetV3 and the added complexity of ODConv was crucial to ensure optimal performance on resource-constrained platforms.
- **Hyperparameter Tuning**: Optimizing both architectures required extensive hyperparameter tuning. This was essential to achieve the desired balance between accuracy and efficiency.

As discussed, the combination of MobileNetV3 and Omni-Dimensional Dynamic Convolution (ODConv), forming Omni-MobileNetV3, holds immense potential for efficient and accurate deep learning models. However, challenges remain in balancing complexity and finding optimal hyperparameters.

2. Bayesian Optimization

We want to enhance the performance of Omni-MobileNetV3 by fine-tuning its hyperparameters. Compared to methods like Grid Search and Random Search, Bayesian Optimization finds the optimal solution more quickly. This efficiency is due to its explore-exploit strategy, which is particularly advantageous for expensive evaluations. We focus on two hyperparameters: the number of attention kernels and the softmax temperature in the ODConv layers of each Inverted Residual block. We utilize Bayesian Optimization, which proves to be an efficient method that can search a large and complex hyperparameter space, to find the best setting for each block. The hyperparameter space of the number of kernels is from 1 to 4, and the range of softmax temperature is from 10 to 60. We begin with 5 instances generated by the Sobel Random Generator and then use the Gaussian Process and Expected Improvement algorithm to generate new combinations, from 10 to 15 instances. We evaluate the accuracy of each instance after 60 epochs of training.

VI. RESULT AND DISCUSSION

The detailed configuration of the Omni-MobileNetV3 model after applying Bayesian Optimization is in the Table II. The optimization resulted in the use of 4 3×3 kernels, 2 1×1 kernels, a temperature setting of 50, and 3 OD blocks. Note that, InvertedResidualOD is the ordinary Inverted Residual block used in MobileNetV3 but the traditional convolution layers are replaced with Omni-Convolution layers. We use the same structure as MobileNetV3-Small.

In this study, we employ the OmniMobileNetV3 model, training it for 100 epochs with optimized hyperparameters.

Table II
NUMBER OF PARAMETERS IN EACH FEATURE BLOCK OF THE MOBILENETV3 MODEL.

Block Index	Layer Type	Parameters
1	Convolution 3x3	464
2	InvertedResidualOD	3079
3	InvertedResidualOD	15985
4	InvertedResidualOD	20905
5	InvertedResidual	13736
6	InvertedResidual	57264
7	InvertedResidual	57264
8	InvertedResidual	21968
9	InvertedResidual	29800
10	InvertedResidual	91848
11	InvertedResidual	294096
12	InvertedResidual	294096
13	Omni-Convolution 1x1	124770
14	AvgPooling2D	-
15	Linear	590848
16	Prediction	-

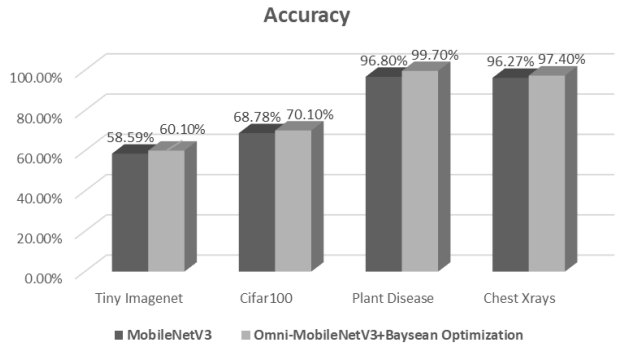


Figure 8. Compare the accuracy across four datasets

We use SGD with momentum as the optimizer, starting with a learning rate of 0.05 and momentum of 0.0004. The learning rate follows a cosine annealing schedule, and we implement temperature annealing to reduce the softmax temperature gradually for the first 50 epochs. Dataset resizing is standardized at 224×224 pixels, except for the Chest X-ray dataset, which is resized to 448×448 pixels. To improve generalization and robustness, we apply the mixup technique [26], generating synthetic data samples through linear interpolation between existing samples and their labels.

The analysis of the data table indicates that the Bayesian Optimization method led to an improvement in accuracy from 58.59% to 60.1%, a nearly 2% enhancement compared to the initial result in the Tiny-ImageNet dataset. Subsequently, we applied the identified hyperparameters from Tiny-Imagenet to the CIFAR-100 dataset to observe their impact. The accuracy increased from 68.78% to 70.1%, demonstrating a similar approximately 2% improvement. This suggests that the hyperparameters determined by the Bayesian Optimization method perform effectively across different benchmark datasets. We also extended the application of these hyperparameters to other datasets, namely Plant Diseases and Chest Xrays. Reapplying the identified hyperparameters resulted in a significant accuracy boost for the Plant Diseases dataset, rising from 96.8% to 99.7%, approximately a 3% increase. Likewise, the Chest Xrays dataset showed an accuracy improvement from 96.27% to 97.4%, roughly more than 1% compared to the original. While the accuracy improvements may look small, they are significant considering MobileNetV3 has the best performance in the class at present and is optimized carefully by NAS.

Table III
COMPARE THE PERFORMANCE OF TWO MODELS IN TINY-IMAGENET

Models	Params	Accuracy	F1-Score
MobileNetV3	**1.52M**	0.585	0.58
Omni-MobileNetV3	**1.82M**	0.596	0.59

Table IV
COMPARE THE PERFORMANCE OF TWO MODELS IN CIFAR-100

Models	Params	Accuracy	F1-Score
MobileNetV3	**1.52M**	0.687	0.69
Omni-MobileNetV3	**1.82M**	0.701	0.72
MobileNetV2	**2.36M**	0.66	0.67

Table V
COMPARE THE PERFORMANCE OF TWO MODELS IN CHEST XRAYS

Models	Params	Accuracy	F1-Score
MobileNetV3	**1.52M**	0.962	0.96
Omni-MobileNetV3	**1.82M**	0.974	0.98

Table VI
COMPARE THE PERFORMANCE OF TWO MODELS IN PLANT DISEASE

Models	Params	Accuracy	F1-Score
MobileNetV3	**1.52M**	0.968	0.96
Omni-MobileNetV3	**1.82M**	0.997	0.98

VII. ABLATION STUDY

We perform Bayesian Optimization to determine the optimal parameters for our model. Specifically, we are tuning the number of kernels in the Omni-Dimensional Dynamic Convolution with corresponding kernel sizes of 3 and 1, the temperature for flattening the attention to near-uniform to aid learning in the early epochs, and the number of OD Blocks. This search is conducted on the Tiny-ImageNet dataset over 60 epochs. The result is displayed in

the (Table VII). The configuration column provides details on the number of 3×3 kernels, the number of 1×1 kernels, the temperature within an OD-Block, and the number of OD-Blocks.

Table VII
TABLE OF PARAMETERS AND PERFORMANCE METRICS

Configuration	Parameters	GFLOPS	Accuracy
2/1/29/7	1.84 M	0.26	0.5786
3/3/53/1	1.85 M	0.44	0.5819
4/2/19/4	1.84 M	0.36	0.5819
1/4/39/11	3.25 M	**0.11**	0.5835
1/2/58/9	2.04 M	0.2	0.5792
1/1/26/8	1.84 M	0.23	0.5769
3/1/33/6	1.84 M	0.28	0.5847
2/1/30/6	1.82 M	0.28	0.5776
4/2/50/3	***1.82 M***	*0.38*	***0.5867***

VIII. CONCLUSION

We proposed a novel approach integrating ODConv into the MobileNetV3 architecture and created a new hybrid model called Omni-MobileNetV3. By enabling kernel adaptation across spatial, channel, filter and kernel dimensions independently, ODConv enhances the network's ability to capture diverse feature representations tailored to the input. To efficiently optimize its hyperparameters, we employed Bayesian Optimization which leverages past evaluations to guide the search process.

Experiments were conducted on four diverse image classification datasets - CIFAR-100, Tiny ImageNet, Plant Disease, and Chest X-rays. The results demonstrated that Omni-MobileNetV3 optimized with Bayesian Optimization consistently outperformed the baseline MobileNetV3 architecture. Accuracy improvements ranged from 1-3% depending on the dataset, with the largest gains seen on the more challenging Plant Disease and Chest X-rays medical imaging datasets.

A deeper analysis of the results indicated that the multi-dimensional attention mechanism incorporated by ODConv played a key role in dynamically enhancing the representational power of the convolutional kernels. Bayesian Optimization also proved effective in identifying hyperparameter configurations that balanced model complexity and predictive performance.

Overall, the findings validate that ODConv is an effective approach for dynamically shaping convolutional kernels based on input features. When combined with Bayesian tuning, it achieves state-of-the-art accuracy on image classification tasks. Future work includes exploring its integration with other architectures and applications beyond computer vision.

ACKNOWLEDGEMENT

This research is funded by Funds for Science and Technology of the VN-UK Institute for Research and Executive Education, the University of Danang under project number VNUK-2023-Student 03

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[9] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *Advances in neural information processing systems*, vol. 29, 2016.

[11] A. Diba, V. Sharma, L. V. Gool, and R. Stiefelhagen, "Dynamonet: Dynamic action and motion network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[12] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[13] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[14] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[15] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in

Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.
[17] C. Li, A. Zhou, and A. Yao, "Omni-dimensional dynamic convolution," *arXiv preprint arXiv:2209.07947*, 2022.
[18] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," 2020.
[19] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International conference on machine learning*. PMLR, 2018, pp. 4095–4104.
[20] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
[21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[22] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[23] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
[24] S. Bhattarai, "New plant diseases dataset," https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset/data.
[25] P. Patel, "Chest x-ray (covid-19 & pneumonia)," https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia.
[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.



Vo Long is a third year student, studying Data Science major, at VN-UK Institute for Research and Executive Education - the University of Danang, Danang. His current research interest is Computer Vision, Natural Language Process, and Foundation Models.
Email: long.vo210404@vnuk.edu.vn



Phan Nhat Quang is a third year student, studying Data Science major, at VN-UK Institute for Research and Executive Education - the University of Danang, Danang. His current research interest is Computer Vision, Natural Language Process, and Foundation Models.
Email: quang.phan210405@vnuk.edu.vn



Tran Van Dat is a second year student, studying Data Science major, at VN-UK Institute for Research and Executive Education - the University of Danang, Danang. His current research interest is Computer Vision, Natural Language Process, and Foundation Models.
Email: dat.tran220407@vnuk.edu.vn



Dang Duc Long Dr. Dang Duc Long is a lecturer at the VN-UK Institute for Research & Executive Education at the University of Danang, where he has worked since 2014. He previously served as the Head of the Research and Industrial Cooperation Office at the same institute from 2014 to 2021. Dr. Long obtained his Ph.D. in Biochemistry from the University of Massachusetts Lowell in 2003, after completing his M.Sc. in Biochemistry from the same university in 1999. His research interests encompass a diverse range of fields, including computational biology, molecular diagnostics, enzymology, and circular economy.
Email: long.dang@vnuk.edu.vn

A Lightweight Model to Skin Disease Recognition

To Huu Nguyen[1], Ma T. Hong Thu[2], Do Thanh Mai[3], Trang Phung T. Thu[3], An Dang[4], Duc-Quang Vu[5]
[1] Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Vietnam
[2] Tan Trao University, Tuyen Quang, Vietnam
[3] Thai Nguyen University, Thai Nguyen, Vietnam
[4] Faculty of Computer Science, Phenikaa University
[5] Thai Nguyen University of Education, Thai Nguyen, Vietnam

Correspondence: Duc-Quang Vu, quangvd@tnue.edu.vn
Received: 14/2/2024, revised: 16/5/2024, accepted: 27/5/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1265

Abstract: Skin disease has become increasingly prevalent, emerging as one of the most widespread conditions. It significantly affects human health and even causes skin cancer and death. Therefore, there are many methods have been proposed to solve this issue recently, especially deep learning-based methods. However, these state-of-the-art methods seem to only focus on how to achieve better performance and ignore the issue of inference time. Specifically, deep learning-based methods usually build very deep with a huge of model size and computational cost. As a result, this becomes very difficult to deploy these models on devices with no GPU support. In this study, we introduce a proficient and lightweight model designed to address this issue, leveraging the Mobilenet architecture. Our experimental findings demonstrate that the suggested network delivers comparable performance to contemporary cutting-edge techniques across diverse benchmark datasets, including HAM10000, International Skin Imaging Collaboration 2017, and International Skin Imaging Collaboration 2019. Notably, our approach utilizes merely 0.2 million parameters and 0.3 GFLOPs for image classification. This attribute holds substantial importance for deploying the model on edge devices lacking GPU support.

Keywords: *skin disease, skin lesion classification, lightweight network, MobileNet*

Tiêu đề: **Một mô hình nhẹ để nhận biết bệnh về da**

Tóm tắt: Bệnh về da ngày càng trở nên phổ biến, nổi lên như một trong những tình trạng phổ biến nhất. Nó ảnh hưởng đáng kể đến sức khỏe con người, thậm chí gây ung thư da và tử vong. Vì vậy, gần đây có rất nhiều phương pháp được đề xuất để giải quyết vấn đề này, đặc biệt là phương pháp dựa trên học sâu. Tuy nhiên, những phương pháp tiên tiến này dường như chỉ tập trung vào việc làm thế nào để đạt được hiệu suất tốt hơn mà bỏ qua vấn đề về thời gian suy luận. Cụ thể, các phương pháp dựa trên học sâu thường xây dựng rất sâu với kích thước mô hình và chi phí tính toán rất lớn. Do đó, việc triển khai các mô hình này trên các thiết bị không hỗ trợ GPU trở nên rất khó khăn. Trong nghiên cứu này, chúng tôi giới thiệu một mô hình gọn nhẹ và hiệu quả được thiết kế để giải quyết vấn đề này tận dụng kiến trúc Mobilenet. Các kết quả thử nghiệm của chúng tôi chứng minh rằng mạng được đề xuất mang lại hiệu suất tương đương với các kỹ thuật tiên tiến hiện đại trên các bộ dữ liệu chuẩn khác nhau, bao gồm HAM10000, International Skin Imaging Collaboration 2017 và International Skin Imaging Collaboration 2019. Đáng chú ý hơn, phương pháp của chúng tôi chỉ sử dụng 0,2 triệu thông số và 0,3 GFlops để phân loại hình ảnh. Điều này có tầm quan trọng đáng kể trong việc triển khai mô hình trên các thiết bị cạnh không có GPU hỗ trợ.

Từ khóa: *bệnh ngoài da, phân loại tổn thương da, mạng nhẹ, MobileNet*

I. INTRODUCTION

Skin cancer stands as the most prevalent among all cancer types, surpassing the cumulative total of other cancer categories combined [1]. The occurrence of skin cancer is particularly prominent among individuals of Caucasian descent, with an incidence rate of approximately 200 cases per 100,000 population. In Vietnam, although the prevalence of skin cancer is comparatively lower in comparison to Caucasians, there has been a noticeable rise in cases. This trend is particularly notable among men, with an incidence of about 3.2 cases per 100,000 population, and among women, with a rate of 3.1 cases per 100,000 individuals. The most lethal variant of skin

cancer, melanoma, is projected to escalate to nearly half a million cases by 2040, marking a 62 percent surge from 2018 [2]. Astonishingly, skin cancer claims a life every four minutes, prompting dermatologists to classify the surge in skin cancer occurrences as a worldwide pandemic.

Prompt intervention, particularly for melanoma, is pivotal for achieving favorable survival outcomes amidst the burgeoning instances [3]. The chief discernible trigger for skin cancer is prolonged exposure to ultraviolet radiation. Dwindling ozone levels contribute to escalated high-level ultraviolet radiation, consequently heightening the risk associated with natural sunlight exposure.

Dermatoscopy stands as a widely employed imaging methodology, utilizing immersion fluid to amplify light and enable visualization of the skin's surface [4]. Nonetheless, the precision of its diagnostic outcomes hinges greatly on the experience of the dermatologist [5, 6]. The dearth of specialized medical personnel, particularly in developing or economically disadvantaged nations, can profoundly impede the timely management of skin cancer cases. Consequently, the necessity for automated, remote diagnostic solutions is assuming heightened significance. The scarcity of specialized medical personnel, especially in developing or economically disadvantaged nations, can significantly hinder the prompt management of skin cancer cases. Consequently, the classification of skin lesions has emerged as a burgeoning realm of study, spurred by the increasing integration of deep learning techniques in the realm of medical image analysis.

The advent of deep learning models has brought about a transformative shift in conventional machine learning applications, offering unparalleled performance enhancements and the capacity to autonomously learn intricate information. Deep neural networks have proven successful across diverse domains, spanning image classification [7, 8], object detection [9], action recognition [10, 11], speech recognition [12, 13], and video prediction [14, 15], among others. There are various methods proposed in skin lesion classification (SLC) such as Zhang et al. [16], Datta et al. [17], TALT [18] and Huynh et al [19]. However, many of these methodologies lean towards utilizing exceedingly large models to attain peak performance, often neglecting secondary factors such as memory usage and runtime efficiency. For example, the ResNet-50 uses 45M parameters and 8GFLOPs, VGGNet uses 133M parameters and 8GFLOPs, etc. This seriously affects the process of deploying these models on embedded/mobile devices without GPU support. In this study, we introduce an efficacious lightweight network that prioritizes minimized memory and computational expenses. This aspect bears substantial significance, particularly when deploying the model on embedded devices devoid of GPU support. To be precise, we present an innovative framework rooted in MobileNet architecture [20] aimed at curbing the model's size and computational overhead in the context of SLC. The principal contributions of this study can be succinctly outlined as follows:

1) Our proposition involves the introduction of a streamlined and resource-efficient network for SLC, built upon the MobileNet architecture.
2) Empirical findings from our experiments conclusively demonstrate the remarkable reduction in both model size and computational expenditure achieved by the suggested network, standing in stark contrast to prevailing deep models at the forefront of the field.

II. PROPOSED MODEL

Table I
THE PROPOSED LIGHTWEIGHT NETWORK FOR SKIN LESION CLASSIFICATION BASED ON MOBILENET.

Type/Stride	**Kernel size × No. of filters**	**Output size**
Input layer		(224,224,3)
Conv2D / s=(2,2)	(3,3) × 8	(112,112,8)
Depthwise / s=(1,1)	(3,3)	(112,112,8)
Pointwise / s=(1,1)	(1,1) × 16	(112,112,16)
Depthwise / s=(2,2)	(3,3)	(56,56,16)
Pointwise / s=(1,1)	(1,1) × 32	(56,56,32)
Depthwise / s=(1,1)	(3,3)	(56,56,32)
Pointwise / s=(1,1)	(1,1) × 32	(56,56,32)
Depthwise / s=(2,2)	(3,3)	(28,28,32)
Pointwise / s=(1,1)	(1,1) × 64	(28,28,64)
Depthwise / s=(1,1)	(3,3)	(28,28,64)
Pointwise / s=(1,1)	(1,1) × 64	(28,28,64)
Depthwise / s=(2,2)	(3,3)	(14,14,64)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(1,1)	(3,3)	(14,14,128)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(1,1)	(3,3)	(14,14,128)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(1,1)	(3,3)	(14,14,128)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(1,1)	(3,3)	(14,14,128)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(1,1)	(3,3)	(14,14,128)
Pointwise / s=(1,1)	(1,1)×128	(14,14,128)
Depthwise / s=(2,2)	(3,3)	(7,7,128)
Pointwise / s=(1,1)	(1,1)×256	(7,7,256)
Global AVE Pooling		(256,)
Output layer	K	(K,)

In this part, we first provide a concise overview of the problem setup. Subsequently, we introduce our proposed

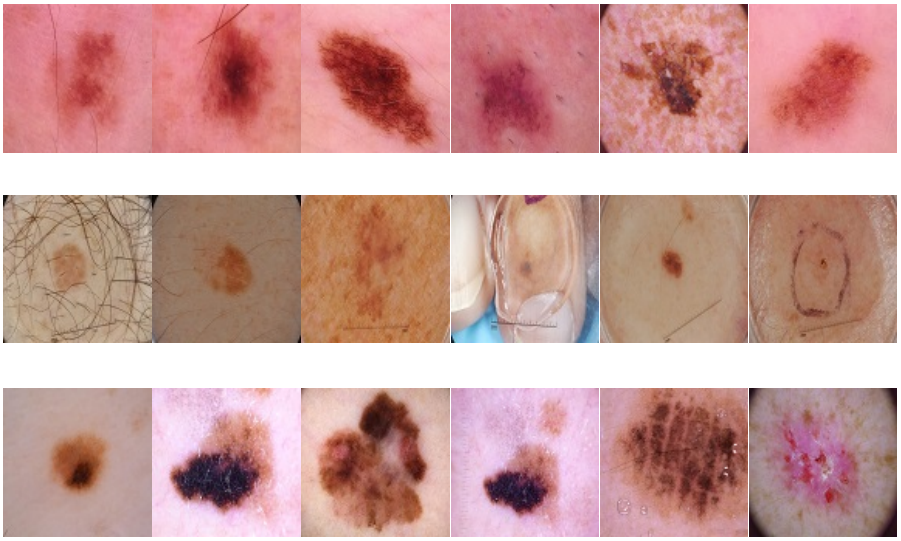


Figure 1. Several example images on three datasets. In which, the top of the figure is the samples in HAM10000, the middle is the samples in ISIC 2017 and the bottom is the samples in ISIC 2019.

lightweight network designed for skin disease recognition, utilizing the MobileNet backbone as its foundation.

1. Problem Setup

Given a dataset denoted by $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$. In which, x_i denotes the RGB training image and y_i is its corresponding label with $i = 1, 2, ...N$ and N is the total of samples in $\mathcal{D}$. Note that $x_i \in \mathbb{R}^{H,W,C}$ where H, W, C as the height, width, and channel dimension of the image, respectively, and $y_i \in \{1, 2, ..., K\}$ where K is the total of classes. We denote $p_i = F_\theta(x_i)$ where F_θ is the proposed network and p_i is the distribution probability vector of the network F_θ, θ indicates the trainable set of parameters. We aim to minimize the distance between the output prediction $\hat{y}_i$ from F_θ and the ground-truth label y_i i.e., $\hat{y}_i = \arg\max(p_i) \approx y_i$. To do that, we utilize the standard cross-entropy (CE) to perform this task as follows:

$$CE(y_i, \hat{y}_i) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{1}$$

2. Lightweight Network

Conventional Convolution. is computed over the entire depth (channel). Therefore, the total parameters of the network increase significantly depending on the depth of the previous layer. For example, we assume that the input of a standard convolution is $h \times w \times c$ we need a set of $k \times k \times c$ parameters to perform the convolution operator and generate an output size of $h' \times w' \times 1$. Therefore, we need c' different filters to create the output size of $h' \times w' \times c'$. From that, the total parameters used for a normal convolution will be $c' \times k \times k \times c$

The number of parameters is negligible if the depth of the input channel is small (usually in the first layers). However, when the depth increases gradually toward the final layers of the Convolutional Neural Network (CNN), the number of parameters of the model will increase significantly. This increase in parameters creates cumbersome models that take up a huge of memory and affect computation speed.

Depthwise Convolution. Inspired by MobileNet, we adopt Depthwise Convolution to reduce the computational cost by performing convolution on each slice. Specifically, each channel will apply a different filter and not share parameters at all. To produce a pixel point on the output, regular convolution uses $k \times k \times c$ operations meanwhile Depthwise Convolution only utilizes $k \times k$.

Pointwise Convolution. is adopted to change the depth channel of the output from c to c' by using 1 Convolution. The detail of our proposed framework is described in Table I. In which, s denotes the stride of the convolution operator, Global AVE Pooling is the Global Average Pooling layer, Depthwise is the Depthwise convolution operator and Pointwise is the pointwise convolution operator.

III. EXPERIMENT

In this section, we commence by introducing several standard datasets commonly employed in skin disease recognition. Following that, we delve into a detailed explanation of the implementation of our proposed network. Concluding this part, we provide the results of various experiments,

Table II
THE COMPARISON BETWEEN OUR PROPOSED LIGHTWEIGHT NETWORK AND RECENT MODELS ON THE HAM10000 DATASET INCORPORATES THE ACCURACY VALUES ACROSS SEVEN DISTINCT CLASSES.

Model	AKIEC	BCC	BKL	DF	MEL	NV	VASC
DenseNet [21]	0.975	0.993	0.960	0.851	0.963	0.975	0.993
VGGNet [22]	0.949	0.977	0.93	0.847	0.925	0.954	0.972
ResNet [7]	0.980	0.997	0.948	0.973	**0.961**	0.974	0.995
Inception-v4 [23]	**0.993**	0.997	0.970	0.973	0.965	0.984	**1.000**
Inception-v4 [23] + SA	0.981	**0.998**	**0.982**	**0.982**	0.974	**0.984**	1.000
Ours	0.970	0.987	0.936	0.962	0.951	0.964	0.985

Table III
THE TOP-1 ACCURACY OF THE PROPOSED NETWORK AND COMPARE IT TO RECENT APPROACHES ON THE ISIC 2019 DATASET.

Method	Method in [24]			Ours		
	AUC	Sensitivity	Specificity	AUC	Sensitivity	Specificity
AK	91.4	48.4	96.5	91.9	54.6	96.1
BCC	94.9	72.1	94.0	94.1	73.2	93.6
BKL	90.4	39.4	98.5	91.4	42.6	98.0
DF	97.9	57.8	99.2	97.7	58.2	98.1
MEL	92.8	59.4	96.2	92.4	60.1	95.8
NV	96.0	71.0	97.5	96.5	71.6	97.1
VASC	95.6	64.4	99.1	95.1	64.6	98.5
SCC	93.8	43.9	98.6	92.2	48.3	98.3
UNK	77.5	0.3	99.9	78.2	38.5	97.8

conduct a thorough analysis, and compare the performance of our framework against state-of-the-art approaches.

1. Dataset

The experiments are performed on three separate datasets including the HAM dataset [25], the ISIC 2017 dataset, and the ISIC 2019 dataset.

The HAM10000 dataset [25] consists of 10,015 dermatoscopy images, each sized 450×600. It encompasses seven distinct diagnostic categories: AK, DF, BCC, MEL, VASC, BKL, and NV. Before their inclusion in the predictive model, all samples are reshaped to dimensions of $224 \times 224 \times 3$. Within the dataset, 9,187 images are allocated for the training set, and the testing set comprises 828 images.

The ISIC 2017 dataset comprises 2,600 images, each with dimensions of 767×1022. Within the training dataset, a total of 2,000 images representing three distinct diseases: benign tumors, seborrheic keratosis, and melanoma. Complementing this, the testing dataset encompasses 600 images for evaluation purposes.

The ISIC 2019 dataset represents the most expansive compilation of publicly available, quality-assured dermatoscopy images designed for research endeavors. This all-encompassing dataset encompasses a total of 25,331 images meticulously categorized into nine distinct classes. An illustrative example featuring three datasets is presented in Fig. 1, where the images from the HAM10000 dataset are showcased at the top, those from the ISIC 2017 dataset are displayed in the middle, and the example images from the ISIC 2019 dataset are depicted at the bottom.

2. Impementation Detail

Within the paper, we undertake both over-sampling and under-sampling techniques to attain an equitably distributed number of images across each class. Subsequently, the images undergo normalization by scaling each pixel through division by 255, thereby constraining pixel values within the range of 0 to 1. Our chosen batch size is set at 128. Employing the Stochastic Gradient Descent (SGD) optimizer, we initialize the learning rate to 0.01, which is subject to a 10-fold reduction if, over 10 consecutive epochs, the model fails to exhibit performance enhancements. Notably, the data augmentation strategies adopted are rooted in the RandAugment methodology [26]. All evaluations of test results are conducted using the dedicated test dataset. All experiments have been conducted in the PC with dual CPU E5-2673 v3 2.4Ghz, 64GB of RAM, and a single GPU A4000.

3. Performance and Comparison

First, we compare the proposed network with other recent methods such as [7, 21–23]. The results have shown in Table II, we found that the proposed framework archives competitive performance on all classes including AKIEC, BKL, DF, MEL, and VASC. Specifically, we achieve 97.0% in terms of accuracy on the class AKIEC outperforms [22] using VGG network, 98.7% in terms of accuracy on the class BCC outperforms [22] using VGG network, 96.2% in terms of accuracy on the class DF outperforms [21] using DenseNet model and so on. Although our network is not the best model, it is the smallest network i.e., using fewer parameters and computational costs compared to very deep networks such as VGG Net, DenseNet, ResNet, and SENet.

Subsequently, we assess the effectiveness of the proposed network by conducting a comparative analysis with established methodologies, including [7, 16, 27, 28], using the ISIC 2017 dataset. The results presented in Table IV demonstrate the superior performance of our proposed method across various metrics. In particular, the proposed network archives 92.5%, 87.8%, 90.2%, and 70.2% in terms of AUC, accuracy, sensitivity, and specificity, respectively. Notably, our network secures a top-2 position, trailing only behind Irv2+SA [17], in terms of sensitivity and accuracy.

Table IV
THE TOP-1 ACCURACY OF THE PROPOSED NETWORK AND COMPARED TO RECENT MODELS ON THE ISIC 2017 DATASET.

Model	AUC	Accuracy	Sensitivity	Specificity
ResNet-50 [7]	0.948	0.842	0.867	0.837
RAN50 [27]	0.942	0.862	0.878	0.859
SENet50 [28]	0.952	0.863	0.856	0.865
ARL-CNN50 [16]	**0.958**	0.868	0.878	**0.867**
Irv2+SA [17]	0.935	**0.898**	**0.945**	0.711
Ours	0.925	0.878	0.902	0.702

We extend the experiment on the ISIC 2019 dataset and compare our framework with the method in [24]. Table III shows that the proposed framework gives better results than [24] in most classes in terms of standard measures such as AUC, sensitivity, and specificity. Especially, our proposed framework achieves 91.9%, 91.4%, 96.5%, and 78.2% on AK, BKL, NV, and UNK, respectively outperforms the method in [24] by a margin of 0.5%–1%.

Finally, we conduct a comparison of the model size and computational cost, i.e., FLOPs, of our framework with other recent methods for SLC. The data in Table V indicates that our network achieves the smallest size and the lowest FLOPs when compared to various methods such as [17, 21, 23]. Specifically, our model utilizes only 0.2 million parameters and 0.3 GFLOPs for SLC, surpassing VGGNet by more than 600 times in terms of model size and 26 times in terms of computational cost. In comparison to the Irv2+SA method [17], our framework reduces the model size by 120 times and FLOPs by 20 times. These findings underscore that our framework significantly alleviates the memory requirements and inference time of the model, facilitating easy deployment on embedded and/or mobile devices without the necessity for GPU support.

Table V
THE DETAIL ABOUT THE NUMBERS OF PARAMETERS AND COMPUTATIONAL COST (GFLOPS) AMONG OUR PROPOSED FRAMEWORK AND RECENT APPROACHES. NOTE THAT 1 GFLOPS EQUALS 10^9 FLOPS.

Method	Architecture	Model size	FLOPs
Inception-v4 [23]	InceptionNet	7M	2G
DenseNet [21]	DenseNet	8M	3G
VGGNet [22]	VGGNet	133M	8G
ResNet-50 [7]	ResNet	45M	8G
SENet50 [28]	SENet	1M	0.8G
Irv2+SA [17]	Inception ResNet v2	24M	6G
Ours	**MobileNet**	**0.2M**	**0.3G**

IV. CONCLUSION

This paper presents the introduction of a lightweight network for skin lesion classification, leveraging MobileNet as its foundation. Rather than employing conventional convolution layers, our approach utilizes depthwise and pointwise convolutions to effectively reduce both the model size and computational cost of the network. Experimental results across diverse datasets demonstrate that our proposed framework attains competitive performance compared to recent methods, all the while utilizing only 0.2M parameters and 0.3G FLOPs. Moving forward, our research endeavors will focus on further analysis and the development of new efficient and lightweight networks to enhance performance, while concurrently minimizing memory size and inference time.

REFERENCES

[1] T. S. C. Foundation, "Skin cancer facts and statistics," https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/, access: 2023-06-30.
[2] M. UK, "2020 melanoma skin cancer report," https://www.melanomauk.org.uk/2020-melanoma-skin-cancer-report, access: 2023-06-30.
[3] R. K. Voss, T. N. Woods, K. D. Cromwell, K. C. Nelson, and J. N. Cormier, "Improving outcomes in patients with melanoma: strategies to ensure an early diagnosis," *Patient related outcome measures*, pp. 229–242, 2015.
[4] H. Kittler, H. Pehamberger, K. Wolff, and M. Binder, "Diagnostic accuracy of dermoscopy," *The lancet oncology*, vol. 3, no. 3, pp. 159–165, 2002.

[5] T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, C. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, S. Fröhling *et al.*, "A convolutional neural network trained with dermoscopic images performed on par with 145 dermatologists in a clinical melanoma image classification task," *European Journal of Cancer*, vol. 111, pp. 148–154, 2019.

[6] T. J. Brinker, A. Hekler, A. H. Enk, J. Klode, A. Hauschild, C. Berking, B. Schilling, S. Haferkamp, D. Schadendorf, T. Holland-Letz *et al.*, "Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task," *European Journal of Cancer*, vol. 113, pp. 47–54, 2019.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] C. Nhan Duong, K. Luu, K. Gia Quach, and T. D. Bui, "Longitudinal face modeling via temporal deep restricted boltzmann machines," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5772–5780.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[10] D.-Q. Vu, N. Le, and J.-C. Wang, "Teaching yourself: A self-knowledge distillation approach to action recognition," *IEEE Access*, vol. 9, pp. 105 711–105 723, 2021.

[11] Q. V. Duc, T. Phung, M. Nguyen, B. Y. Nguyen, and T. H. Nguyen, "Self-knowledge distillation: an efficient approach for falling detection," in *International Conference on Artificial Intelligence and Big Data in Digital Era*. Springer, 2021, pp. 369–380.

[12] H. M. Tan, D.-Q. Vu, C.-T. Lee, Y.-H. Li, and J.-C. Wang, "Selective mutual learning: An efficient approach for single channel speech separation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3678–3682.

[13] H. M. Tan, D.-Q. Vu, and J.-C. Wang, "Selinet: A lightweight model for single channel speech separation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] T. Phung, V. T. Nguyen, T. H. T. Ma, and Q. V. Duc, "A (2+ 1) d attention convolutional neural network for video prediction," in *International Conference on Artificial Intelligence and Big Data in Digital Era*. Springer, 2021, pp. 395–406.

[15] D.-Q. Vu and T. P. T. Thu, "Simultaneous context and motion learning in video prediction," *Signal, Image and Video Processing*, pp. 1–10, 2023.

[16] J. Zhang, Y. Xie, Y. Xia, and C. Shen, "Attention residual learning for skin lesion classification," *IEEE transactions on medical imaging*, vol. 38, no. 9, pp. 2092–2103, 2019.

[17] S. K. Datta, M. A. Shaikh, S. N. Srihari, and M. Gao, "Soft attention improves skin cancer classification performance," in *Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data: 4th International Workshop, iMIMIC 2021, and 1st International Workshop, TDA4MedicalData 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 4*. Springer, 2021, pp. 13–23.

[18] D. M. Nguyen, T. T. Nguyen, H. Vu, Q. Pham, M.-D. Nguyen, B. T. Nguyen, and D. Sonntag, "Tatl: Task agnostic transfer learning for skin attributes detection," *Medical Image Analysis*, vol. 78, p. 102359, 2022.

[19] A. T. Huynh, V.-D. Hoang, S. Vu, T. T. Le, and H. D. Nguyen, "Skin cancer classification using different backbones of convolutional neural networks," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2022, pp. 160–172.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[24] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, "Skin lesion classification using ensembles of multi-resolution efficientnets with meta data," *MethodsX*, vol. 7, p. 100864, 2020.

[25] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[26] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[27] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.



To Huu Nguyen received the Bachelor's Education of Information Technology at Thai Nguyen University of Education in 2003 and Master's degree in Computer Science at Thainguyen University in 2008. He received the PhD degree at the Academy of Science and Technology - Vietnam Academy of Sciences in 2021. He worked as a lecturer at the Faculty of Information Technology, School of Information and Communication Technology, Thai Nguyen University from 2004. Now, he is a researcher at the Institute of Information Technology, Academic Institute of Science and Technology, Vietnam.

Email: thnguyen@ictu.edu.vn



Ma T. Hong Thu received a master's degree in Computer Science in 2015 at the Thai Nguyen University of Information and Communications Technology. Research interests include: Machine learning and deep learning.
Email: thutq7@gmail.com



Do Thanh Mai received the Bachelor of Education in Information Technology at Thai Nguyen University of Education in 2003 and the Master's degree in Computer Science at Thai Nguyen University in 2008. She worked as a lecturer of Information Technology in the Basic Sciences Department at the School of Foreign Languages (SFL-TNU). Office address: School of Foreign Languages, Thai Nguyen University, Thai Nguyen, Vietnam.
Email: dothanhmai.sfl@tnu.edu.vn



Trang Phung T. Thu was born in Bac Ninh, Vietnam in 1991. She received a B.S. degree in education in information technology from the Thai Nguyen University of Education, Vietnam, in 2013 and an M.S. degree from the Thai Nguyen University of Information and Communication Technology (ICTU) in 2015.

Her research interests include machine learning, deep learning, computer vision, speech processing, and bioinformatics.
Email: phungthutrang.sfl@tnu.edu.vn



An Dang received her Ph.D. in Computer Science and Information Engineering from National Central University, Taiwan in 2022. She specializes in deep learning applications for image processing and audio/speech signal analysis.
Email: an.dangthithuy@phenikaa-uni.edu.vn



Duc-Quang Vu was born in Nam Dinh, Vietnam in 1991. He received a B.S. degree in education in information technology from the Thai Nguyen University of Education, Vietnam, in 2013 and an M.S. degree in Information Systems, from the University of Engineering and Technology (UET), Vietnam National University, Hanoi (VNU) in 2016. He received the Ph.D. degree in the Department of Computer Science & Information Engineering, National Central University, Taiwan in 2022 and a postdoc in 2023. His research interests include machine learning, deep learning, computer vision, speech processing, and bioinformatics.
Email: quangvd@tnue.edu.vn

Research and Experiment with Some Activation Functions in Sleep Stage Classification Using Deep Learning Methods

Tan Thuy Trang, Nguyen Thi Hong, Duong Phi Long
Faculty of Information Systems, University of Information Technology, Vietnam National University Ho Chi Minh City, Vietnam.

Correspondence: Duong Phi Long, longdp@uit.edu.vn
Received: 14/1/2024, revised: 7/5/2024, accepted: 28/5/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1269

Abstract: Sleep disorders are becoming increasingly complex in today's society. Analyzing sleep structure provides valuable information about sleep types, sleep cycles, and variations in different stages. However, manual sleep stage classification is challenging and time-consuming, limiting its clinical use. Deep learning methods have demonstrated promise for automated sleep stage classification, but they face challenges such as class imbalance and the need for interoperability. This paper investigates three deep learning methods, TinySleepNet, TS-TCC, and CA-TCC, using the Expanded Sleep-EDF dataset. The results indicate that TinySleepNet performs well overall, achieving an accuracy of around 80%, while TS-TCC and CA-TCC also perform relatively well with accuracies of around 70%. All three methods classify the Wake stage effectively, with accuracies reaching up to 90%, but struggle with the N1 stage, where accuracy falls below 60%. We propose methods to address these challenges, including the use of GELU, PReLU, LeakyReLU, and SinLU activation functions, which can handle negative values as alternatives to the original ReLU function, and the result can increase from 0.2-1%. Additionally, we have developed a sleep stage classification tool based on these methods.

Keywords: *sleep stage classification, TinySleepNet, TS-TCC, CA-TCC.*

Tiêu đề: **Nghiên cứu và thử nghiệm một số hàm kích hoạt trong phân lớp giai đoạn giấc ngủ sử dụng các phương pháp học sâu**

Tóm tắt: Rối loạn giấc ngủ ngày càng trở nên phức tạp trong xã hội ngày nay. Phân tích cấu trúc giấc ngủ cung cấp thông tin quý giá về các loại giấc ngủ, chu kỳ giấc ngủ và sự biến đổi trong các giai đoạn khác nhau. Tuy nhiên, việc phân loại giai đoạn giấc ngủ thủ công gặp nhiều khó khăn và tốn nhiều thời gian, hạn chế việc áp dụng trong lâm sàng. Các phương pháp học sâu đã cho thấy triển vọng trong việc phân loại giai đoạn giấc ngủ tự động, nhưng vẫn gặp phải những thách thức như mất cân bằng giữa các lớp và nhu cầu về khả năng liên kết dữ liệu. Bài báo này nghiên cứu ba phương pháp học sâu, TinySleepNet, TS-TCC và CA-TCC, sử dụng bộ dữ liệu Expanded Sleep-EDF. Kết quả cho thấy TinySleepNet hoạt động tốt nhất với độ chính xác khoảng 80%, trong khi TS-TCC và CA-TCC cũng hoạt động tương đối tốt với độ chính xác khoảng 70%. Cả ba phương pháp đều phân loại giai đoạn Wake hiệu quả với độ chính xác lên đến 90%, nhưng gặp khó khăn với giai đoạn N1, khi độ chính xác chỉ dưới 60%. Chúng tôi đề xuất các phương pháp để giải quyết những thách thức này, bao gồm sử dụng các hàm kích hoạt GELU, PReLU, LeakyReLU và SinLU, có khả năng xử lý các giá trị âm, thay thế cho hàm ReLU gốc và kết quả có thể tăng từ 0,2-1. Ngoài ra, chúng tôi đã phát triển một công cụ phân loại giai đoạn giấc ngủ dựa trên các phương pháp này.

Từ khóa: *phân lớp giai đoạn giấc ngủ, TinySleepNet, TS-TCC, CA-TCC.*

I. INTRODUCTION

Sleep is a basic human need and has a profound impact on every aspect of daily life. During sleep, the body regenerates and restores energy. Lack of sleep or poor quality sleep can cause fatigue, impaired concentration, weakened immune function, and increase the risk of cardiovascular disease, diabetes, hypertension, and obesity [1]. Sleep also has a significant impact on mood and emotions. Lack of sleep or poor quality sleep can cause anxiety, depression, mental stress, and difficulty concentrating, affecting memory and learning. Research shows that adequate, good-quality sleep improves concentration, memory, and problem-solving abilities, stimulating work performance

and creativity. Conversely, poor sleep can affect weight and overall health, increase hunger, reduce concentration and reaction time, increase the risk of accidents, and reduce work performance [2].

Understanding sleep structure is vital for assessing sleep quality, identifying issues, and developing treatment strategies. Good sleep is crucial for health and cognitive function, while sleep disorders are increasingly common and need prompt treatment to minimize their impact on health and quality of life [2]. A night's sleep consists of 4 to 5 cycles, each lasting about 90 to 110 minutes and progressing through the stages: N1, N2, N3, N2, and REM [3]. The Wake stage is when a person is alert and can interact with the environment. The N1 stage, also known as Light Sleep, represents 5% of total sleep time and is characterized by theta waves with low voltage on EEG recordings. This stage marks the transition from wakefulness to sleep, with more than 50% of alpha waves being replaced with low-amplitude mixed-frequency (LAMF) activity. Muscle tone is present in the skeletal muscle, and breathing is regular. The N2 stage, or Deeper Sleep, accounts for 45% of total sleep and is marked by a drop in heart rate and body temperature. Sleep spindles and K complexes, indicative of neuronal firing in various brain regions, are observed in EEG recordings. This stage is also when teeth grinding, or bruxism, occurs. The N3 stage, or Deepest Non-REM Sleep, makes up 25% of total sleep and is characterized by delta waves, which have the lowest frequency and highest amplitude. This stage is the most restorative, with mental fog often experienced upon awakening. Finally, REM sleep accounts for 25% of total sleep and is associated with dreaming. While the brain is highly active during this stage, similar to wakefulness, the skeletal muscles are atonic and without movement. This stage is also characterized by irregular breathing and increased brain oxygen use. [4].

In recent years, many studies have utilized neural network models for the classification of sleep stages in various medical conditions. These studies have primarily focused on the analysis of sleep cycle patterns and spectral power to accurately classify sleep stages. Additionally, some research has incorporated heart rate and respiration data to aid in the diagnosis of sleep-related breathing disorders [6], [7]. These studies focus on utilizing deep learning methods that employ machine learning techniques to extract features for sleep stage classification. Depending on the specific methods and approaches, the choice of signal channels and neural network models for analysis varies. For example, the paper "Automatic sleep stage classification based on two-channel EOG and one-channel EMG" [8] discusses sleep stage classification using Electrooculogram (EOG) and Electromyogram (EMG) signals. In contrast, the EOGNET method introduced in 2021 only uses EOG (Electrooculogram) signals [9]. Other methods like DeepSleepNet [10], SleepEEGNet [11], and AttnSleep [12] utilize Electroencephalogram (EEG) signals. EEG signals are commonly used in sleep stage classification tasks because they are highly interpretable, providing essential information about the patient's sleep state [13]. Therefore, in this paper, we survey studies that employ EEG signals for sleep stage classification, including TinySleepNet [14], TS-TCC [15], and CA-TCC [16]. We conducted experiments on the Sleep EDF Expanded dataset [17] and evaluated these methods based on criteria such as F1-score and Accuracy. These results assess the potential of machine learning applications in sleep stage classification, contributing to further research and the enhancement of sleep quality in the future.

II. RESEARCH METHODOLOGY

Classification of sleep stages using deep learning based on raw EEG signals is a research topic that has attracted significant attention from the scientific community. Approaches to this problem involve utilizing deep neural networks to extract features from EEG signals and classify sleep stages based on various models, including CNNs (convolutional neural networks), and RNNs (recurrent neural networks), and their combinations. Furthermore, data augmentation techniques have been applied to improve model generalization, including weak and strong data augmentation. Adversarial learning models have also been employed to extract features from EEG signals, such as the "Temporal Contrasting" and "Contextual Contrasting" modules.

1. TinySleepNet

TinySleepNet is a deep learning model that was proposed in the paper "TinySleepNet: An Efficient Deep Learning Model for Sleep Stage Scoring based on Raw Single-Channel EEG" by Supratak et al. (2020) [14]. TinySleepNet is a deep learning model designed for classifying sleep stages using single-channel EEG signals. It integrates a convolutional neural network (CNN) to extract spatial features and long short-term memory (LSTM) to process temporal data, enabling the model to capture the continuity of sleep stages. This model is an improved version of the DeepSleepNet method [10], with the main difference being that it uses only one CNN branch instead of two branches like the DeepSleepNet [10] architecture. As a result, TinySleepNet has fewer model parameters than existing models, requiring less training data and computational resources. The first part of the network consists of four convolutional layers, two max-pooling layers, and a dropout layer. These layers

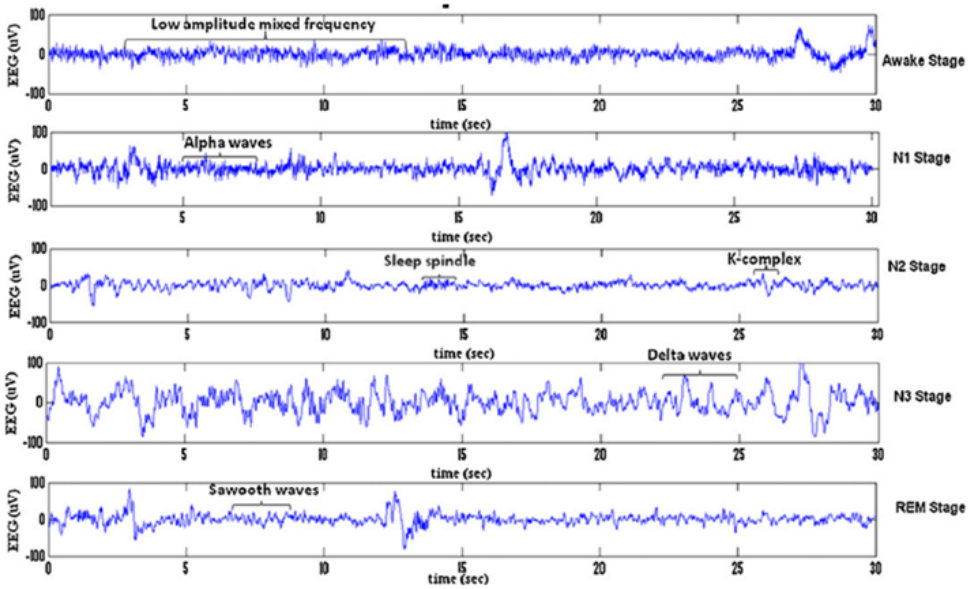


Figure 1. EEG signals for different sleep stages [5].

are used to extract time-invariant features from the original EEG signal. The convolutional layer converts each segment of the EEG signal into a feature vector, denoted as (a_i). In the preprocessing stage, EEG signals are normalized and segmented into smaller segments. The model is trained using the Cross-Entropy loss function and optimized with the Adam optimizer. This is done through a convolutional network, denoted as CNN, where are the learnable parameters of the network. The size of (a_i) varies depending on the sampling rate of the input EEG signal. Figure 2 describes the overall structure of TinySleepNet.

2. TS - TCC

This paragraph introduces a representation learning method for time series data called Temporal and Contextual Contrasting (TS-TCC), which was proposed in the paper "Time-Series Representation Learning via Temporal and Contextual Contrasting" by Eldele et al. (2021) [15]. TS-TCC uses Strong Augmentation and Weak Augmentation to create two different perspectives of the input data. Then, it uses a Temporal Contrasting module to explore the temporal characteristics of the data by simulating an autoregressive model. Autoregressive models use historical information and contextual contrast modules to optimize their predictive ability. Experimental results show that a simple model trained on features learned by TS-TCC can perform almost as well as a model trained with supervision. In addition, TS-TCC is highly effective when there are only a limited number of labeled examples and can also be used for

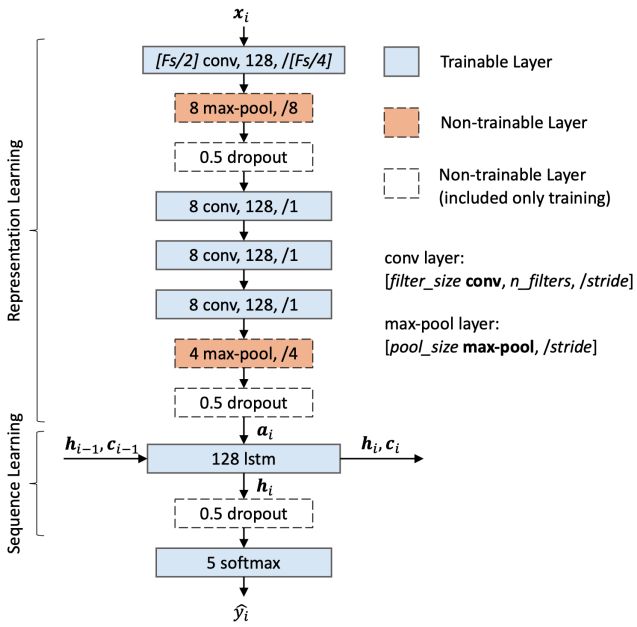


Figure 2. Overview structure of TinySleepNet [14].

transfer learning. For example, by using only 10% labeled data, TS-TCC can achieve performance similar to that of a supervised model. Figure 3 represents the overall structure of TS-TCC [15].

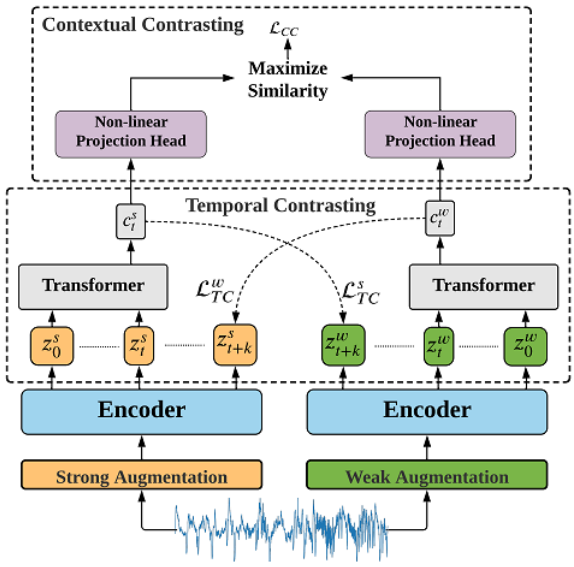


Figure 3. Structure of TS-TCC [15]

3. CA-TCC

In their 2022 paper titled "Contrastive Representation Learning for Semi-supervised Time-Series Classification", Eldele et al. proposed a second variant of TS-TCC, Class-Aware TS-TCC (CA-TCC), which is specifically aimed at semi-supervised learning [16]. This variant uses a combination of labeled and unlabeled data to improve the representation learned from TS-TCC. In the first stage of the semi-supervised learning process, the TS-TCC model is trained on unlabelled data. In the second stage, the pre-trained encoder is fine-tuned using labeled data. In the third stage, the unlabeled data is assigned pseudo labels by using a retuned TS-TCC encoder. Finally, in the fourth stage, the semi-supervised model is trained using the labeled and pseudo-labeled data. Figure 4 represents the four semi-supervised training stages of CA-TCC.

III. EXPERIMENT AND EVALUATE

1. Dataset

In this research, we utilized PHYSIONET's openly available polysomnography signal data source, Expanded Sleep-EDF (ES-EDF) [17]. The dataset encompasses recordings of polysomnographic signals (PSG) that have been annotated with sleep stages. The dataset features a range of signal types, such as electroencephalography (EEG) signals, electrooculography (EOG) signals, and electromyography (EMG) signals. The dataset is organized into two file formats: PSG.EDF files, which contain recordings of polysomnographic signals, and Hypnogram.EDF files, which contain labels for the sleep stages of each recording.

These Hypnogram.EDF files, which are of immense value to our research. These files supply the sleep stage labels, effectively providing a roadmap of the various sleep stages. The stages represented include wakefulness (W), transition to sleep (N1), deep sleep (N2, N3, N4), and Rapid Eye Movement (REM) sleep.

Additionally, the Hypnogram.EDF files include annotations for movement time (Movement(M)), unknown events (UNKNOWN), and timestamps for the signals obtained during each stage of sleep corresponding to the PSG file [4]. Each EDF and Hypnogram file has a header that identifies the patient (anonymized by gender and age), details about the recording (specifically the period recorded), and characteristics of the signal.

The data used for all three methods TinySleepNet, TS-TCC, and CA-TCC are labeled data, after extracting the EEG signal channel FPZ-CZ [18], the data is stored as NPZ file to facilitate data processing and analysis by converting waveform data into numbers, in which x has 3000 attributes and y has 5 labels which are 0, 1, 2, 3, 4 corresponding to wake stage W, sleep stage N1, sleep stage N2, sleep stage N3, sleep stage REM. Also at this step, remove the epochs labeled Movement(M) and UNKNOWN and combine phase 4 into phase 3 because they have similar characteristics [19]. We proceed to divide the data into 2 sets of train and test with a ratio of 8:2, then continue to divide the train set into a train set and a validation set with a ratio of 8:2 based on the criterion of balancing the ratio of stages in the group. Accordingly, the training set has 98 NPZ files, the validation set has 24 NPZ files and the test set has 31 files. Each file will have a duration of 86400s after removing the Movement and unknown stages. The remaining time will be divided into epochs of 30 seconds, and sleep scoring will be performed on each epoch. Additionally, for TS-TCC and CA-TCC, there is an additional step of converting data into PT files.

Table I
STATISTICS ON THE NUMBER OF EPOCHS AND FILES OF EACH DATA SET

Data set	Number of files	Number of epochs
Training set	98	158383
Validation set	24	34390
Testing set	31	37096

After training the TinySleepNet, TS-TCC, and CA-TCC methods using the training set specified in Table I, we performed an experimental evaluation by re-testing with the testing set specified in Table I. The testing set includes 31 samples, which were evaluated using criteria including

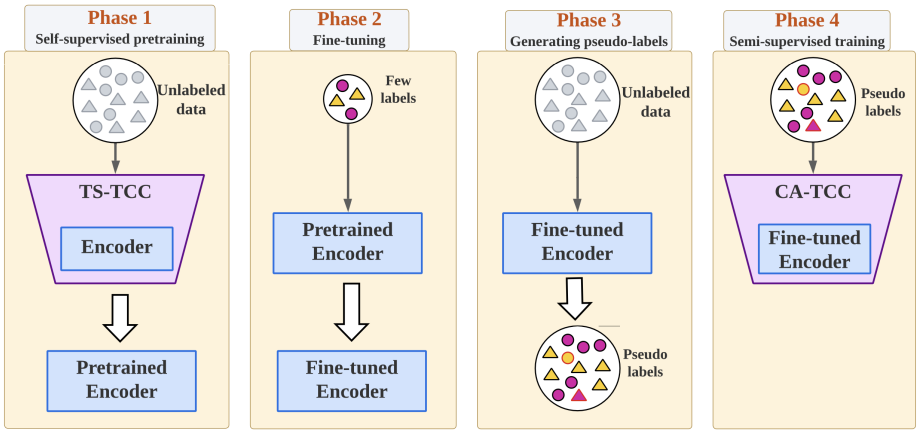


Figure 4. 4 stages of semi-supervised training of CA-TCC [16]

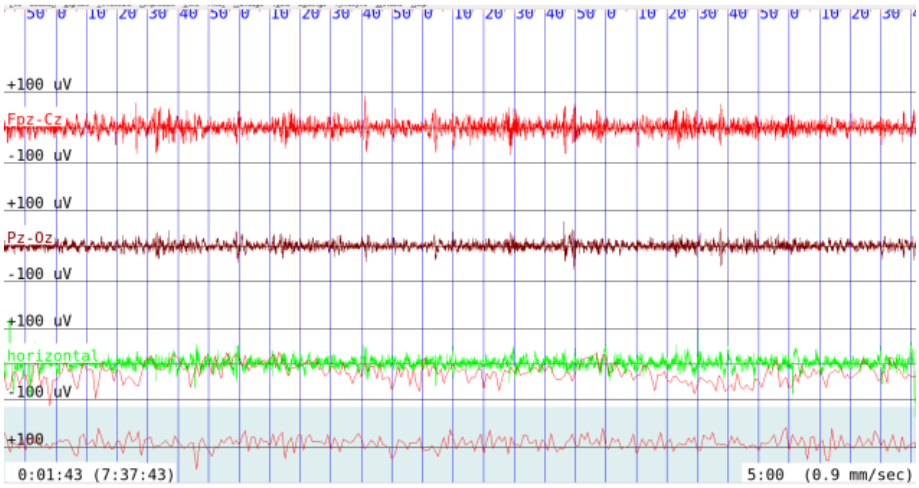


Figure 5. Image illustrating the signal of an EDF file

Table II
STATISTICS ON THE NUMBER OF EPOCHS OF EACH PERIOD IN EACH DATA SET

Stage	W	N1	N2	N3	REM
Training set	55304	16979	54873	10613	20614
Validation set	13260	3418	11808	2289	3615
Testing set	10647	4543	14259	2426	5221

accuracy and F1-score. The results are shown in Tables III and IV.

2. Evaluation criteria

In this article, we applied two criteria to evaluate the effectiveness of sleep stage classification models, specifically Accuracy and Weighted-averaged F1-score (Macro-Avg.F1). Macro-Avg.F1 is a popular criterion for evaluating the performance of models on imbalanced datasets.

We customized the training parameters in comparison to the original paper to align with our data and research approach.

3. Evaluate

After using the methods to train the model, experiment with testing the models on the previously divided testing set. The Accuracy and Macro-Avg.F1 results of the methods are shown in Table III.

Table III
TABLE OF ACCURACY AND MACRO-AVG.F1 RESULTS IN EACH METHOD

Methods	Accuracy	Macro-Avg.F1
TinySleepNet	**81.88**%	**77.64**%
TS-TCC	75.37%	68.84%
CA-TCC	75.52%	69.16%

The results have demonstrated, that the methods achieve

relative efficiency on the dataset. CA-TCC achieved an Accuracy of 75.52% and Macro-Avg. F1 is 69.16%, TS-TCC achieved an Accuracy of 75.37%, and Macro-Avg.F1 is 68.54%. The TinySleepNet method has a high compatibility performance with an Accuracy of 81.88% and Macro-Avg.F1 of 77.64%, showing the most accurate classification ability among the methods.

Table IV
COMPARISON OF ACCURACY AND F1-SCORE FOR EACH STAGE IN EACH METHOD

Methods	Metric	Stages				
		W	N1	N2	N3	REM
CA-TCC	Accuracy	91.45	38.15	78.35	74.28	60.33
	F1-score	87.50	36.28	80.55	72.18	65.26
TS-TCC	Accuracy	91.43	35.26	80.24	75.19	61.98
	F1-score	87.79	36.25	81.61	71.26	66.44
TinySleepNet	Accuracy	91.85	54.61	85.16	77.63	78.52
	F1-score	92.08	53.20	84.88	76.82	81.09

The results in Table IV show that the three deep learning methods all performed pretty well in classifying each sleep stage. However, there were some differences in performance between the methods and between the sleep stages:

- **Stage W:** The TinySleepNet method achieved the highest results with an accuracy of 91.85% and an F1-score of 92.08%. The TS-TCC and CA-TCC methods also yielded good results with accuracies above 80%.
- **Stage N1:** The CA-TCC and TS-TCC methods yield low results: accuracy and F1-score of only around 30%. The TinySleepNet method yields average results, with accuracy and F1-score above 50%. This indicates that the methods still face many difficulties in classifying stage N1.
- **Stage N2:** Stage 2 is a stage with quite good performance when the accuracy is from 75% and the F1-score of the methods is all above 80%.
- **Stage N3:** TinySleepNet shows relatively good performance in classifying stage N3, with an accuracy and F1-score of around 75%. TS-TCC and CA-TCC also achieve fairly good performance in classifying this stage, with accuracy and F1-score of around 70%. These methods have relatively good performance in detecting stage N3.
- **REM Stage:** TinySleepNet still maintains relatively good performance in classifying the REM stage, with an accuracy of 78.52% and an F1-score above 80%. Meanwhile, TS-TCC and CA-TCC face more difficulties and have lower performance, with accuracy and F1-score only reaching around 60%. These two methods can classify the REM stage at an average level and need further improvement to achieve higher accuracy.

The evaluation results indicate that the effectiveness of classification methods depends on each specific stage and exhibits significant variability.

Classification methods' efficacy varies across sleep stages. Classifying N1 is challenging due to its resemblance with other stages and short duration. W and N2 stages are less complex to classify due to distinct features. However, N3 and REM stages, despite their distinct characteristics, pose classification difficulties.

Regarding neural network architecture, different models perform differently. TinySleepNet, with its simple structure, effectively classifies stages like W, N2, and N3. TS-TCC and CA-TCC, having complex architectures, excel in classifying challenging stages like N1 and REM.

Performance improvement requires collecting more data for all stages, especially N1 and REM. Also, improving training data quality by removing noise and filtering inaccurate data, researching new methods for learning complex features, and handling noise in data is essential. Further research is needed on larger and more diverse datasets as performance may vary.

4. Testing improvement

In EDF to NPZ conversion, data is stored as a 3000-value array, each value representing an epoch of EEG signals from the FPZ-CZ channel, which can be positive or negative. The activation function, often using Rectified Linear Unit (ReLU), is crucial for deep learning performance, impacting learning capacity, stability, and computational load. While ReLU reduces costs and speeds up training, it can cause inactive neurons if input is negative, potentially affecting network performance. We conducted a survey and experiment on several different activation functions on the hidden layer to replace ReLU in the process of using methods to train the model such as LeakyReLU [20], PReLU [21], GELU [22], SinLU [23]

The Leaky ReLU function is defined by the formula $f(x) = max(\alpha x, x)$, where α is a small positive constant, typically set between 0.01 and 0.3. Unlike ReLU, which outputs 0 for negative inputs, LeakyReLU produces a small negative value. This slope helps alleviate the "dying ReLU" problem that ReLU can experience when the input is too negative. [20]

The PReLU function is also defined by the formula $f(x) = max(\alpha x, x)$. However, unlike Leaky ReLU, the α parameter is not fixed but is instead learned during training. This allows PReLU to learn hidden features and create more complex decision boundaries than ReLU. As a result, PReLU can be more effective at learning and representing complex data. [21]

The GELU function can be approximated by the following formula: $GELU(x) = 0.5x(1 + Tanh((2/\pi)(x + 0.044715x^3)))$. GELU is constructed based on the Gaussian cumulative distribution function, which allows it to have continuous properties and the ability to approximate non-linear functions while avoiding the dead state problem that occurs in ReLU. ReLU has a derivative of 0 for negative or zero values, while GELU has a non-zero derivative in this case.[22].

The SinLU function is defined by the formula $SinLU(x) = (x+asinbx)\cdot\sigma(x)$. It is a continuous activation function that helps to maintain the continuity of the SinLU derivative and prevents the vanishing gradient problem. Additionally, SinLU has a buffer zone on the negative side of the X-axis, similar to the GELU function. When the input is within this buffer zone, SinLU allows for a non-zero derivative, which helps to stabilize the training of the model. [23]

Table V
COMPARISON TABLE OF RESULTS
BETWEEN ACTIVATION FUNCTIONS

	ReLU	PReLU	LeakyReLU	GELU	SinLU
Accuracy					
TinySleepNet	81.88	81.18	80.33	**82.04**	80.90
TS-TCC	75.37	**75.57**	75.24	74.44	73.43
CA-TCC	75.52	**75.69**	75.23	74.39	74.04
Macro-Avg.F1					
TinySleepNet	**77.64**	75.85	76.24	76.91	75.07
TS-TCC	68.84	**69.43**	68.8	68.54	67.96
CA-TCC	69.3	**69.5**	69.2	68.52	68.47

Based on the data in Table V, the performance of TinySleepNet, TS-TCC, and CA-TCC can be compared when using different activation functions and key evaluation metrics such as Accuracy and Macro-Avg.F1:TinySleepNet achieves the best results with all activation functions, with the highest Accuracy score of 82.04% when using the GELU activation function. TS-TCC and CA-TCC have lower accuracy, with TS-TCC achieving the highest accuracy of 75.57% when using the PReLU activation function, and CA-TCC achieving the highest accuracy of 75.69% also with the PReLU activation function.

In terms of Macro-Avg.F1, TinySleepNet also achieves the best results with the highest score of 77.64% when using the ReLU activation function. TS-TCC and CA-TCC have lower scores, with TS-TCC achieving the highest score of 69.43% when using the PReLU activation function, and CA-TCC achieving the highest score of 69.5% also with the PReLU activation function.

The results show that TinySleepNet performs better than TS-TCC and CA-TCC in both evaluation metrics. ReLU can help the TinySleepNet model learn faster and avoid the issue of vanishing gradients, which can explain why it achieves the highest accuracy. GELU is a smooth non-linear activation function that can help improve the learning ability of neural networks. GELU can help the TinySleepNet model capture more complex patterns, which may explain why it achieves the highest Macro-Avg.F1 score. The TS-TCC and CA-TCC models perform best when using PReLU, as this activation function effectively utilizes negative input values by allowing the gradient to be learned during training. The slope for negative values is learned during the training process. This allows the neural network to learn the most appropriate slope for each neuron, enhancing the model's ability to capture complex patterns, and providing the highest accuracy TS-TCC and CA-TCC.

5. Sleep stages classification tool

We have developed a tool for classifying sleep stages based on 3 methods: TinySleepNet, TS TCC, and CA TCC. For each method, we select the model with the best performance to incorporate into the tool. This tool is built on the Flask framework, providing a simple interface to process EDF files and predict sleep stages using classification models trained with TinySleepNet, TS-TCC, and CA-TCC methods. Uploading raw EDF file (a standard format for sleep data) alongside the file containing its correct sleep stage labels. Then, data processing and classification are performed, feeding it into each pre-trained sleep stage classification model. The tool then presents the sleep stage predictions generated by each model. To make the comparison even more intuitive, it utilizes chart.js to create compelling statistical charts. These charts visually represent the accuracy level of each model, allowing you to see at a glance which one reigns supreme for your specific data. Figure 6 shows the screen after analyzing an EDF file containing 20 epochs. Tool links: https://link.uit.edu.vn/sleepstage

IV. CONCLUSION

The performance of TinySleepNet, TS-TCC, and CA-TCC models was evaluated using various activation functions and metrics such as Accuracy and Macro-Avg.F1. TinySleepNet emerged as the best overall performer, achieving the highest Accuracy of 82.04% with the GELU activation function, and the highest Macro-Avg.F1 of 77.64% with the ReLU activation function. TS-TCC and CA-TCC, while performing well, showed lower scores with their best performance using the PReLU activation function (Accuracy of 75.57% and 75.69%, respectively, and Macro-Avg.F1 of 69.43% and 69.5%, respectively).

The efficacy of classification methods varies significantly across different sleep stages. The N1 stage remains



Figure 6. Classification results and comparison from sleep stage classification tool

particularly challenging to classify, with accuracy below 60% due to its short duration and similarity to other stages. In contrast, W and N2 stages achieve higher accuracy rates (80%-90%) due to their distinct features. N3 and REM stages, despite having clear characteristics, still pose classification challenges, with accuracy rates ranging from 60% to 80%.

ReLU activation helps TinySleepNet in avoiding vanishing gradients, resulting in the highest Macro-Avg.F1 score, while GELU activation enhances its ability to capture complex patterns, achieving the highest Accuracy. PReLU activation is most effective for TS-TCC and CA-TCC, facilitating the learning of appropriate slopes for negative input values and improving pattern recognition. However, all three models face difficulties in accurately classifying the N1 stage.

To address these challenges, we propose the use of a Fuzzy Ensemble using the Gompertz function to combine the three methods to improve the reliability of the sleep stage classification. By integrating the strengths of multiple models, the ensemble approach can potentially achieve higher accuracy and better generalization performance compared to using a single model alone. The Gompertz function will allow me to capture the relationships between the EEG features and the sleep stages, which is crucial given the complexity of the underlying physiological processes. This integrated approach should make the overall sleep staging system more reliable and less prone to errors, which is important for real-world applications in sleep medicine and cognitive neuroscience research.

In summary, while exploring different activation functions can improve model performance, the increase is not significant. Future research should focus on enhancing data preprocessing and classification for the N1 stage, developing user-friendly tools with customizable functions, and investigating comprehensive systems suitable for combining classification results. These advances will enhance sleep stage classification using EEG signals from the FPZ-CZ channel and open new research directions. This paper presents the results of the author's graduation thesis research, which explored effective methods for sleep stage classification through the development of research content, experiments, and tools.

ACKNOWLEDGEMENT

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

REFERENCES

[1] A. J. Scott, T. L. Webb, M. Martyn-St James, G. Rowse, and S. Weich, "Improving sleep quality leads to better mental health: A meta-analysis of randomised controlled trials," *Sleep medicine reviews*, vol. 60, p. 101556, 2021.

[2] F. Mendonça, S. S. Mostafa, F. Morgado-Dias, A. G. Ravelo-Garcia, and T. Penzel, "A review of approaches for sleep quality analysis," *Ieee Access*, vol. 7, pp. 24 527–24 546, 2019.

[3] I. Feinberg and T. Floyd, "Systematic trends across the night in human sleep cycles," *Psychophysiology*, vol. 16, no. 3, pp. 283–291, 1979.

[4] A. K. Patel, V. Reddy, and J. F. Araujo, "Physiology, sleep stages," in *StatPearls [Internet]*. StatPearls Publishing, 2022.

[5] W. Al-salman, Y. Li, and P. Wen, "Detection of eeg k-complexes using fractal dimension of time frequency images technique coupled with undirected graph features," *Frontiers in Neuroinformatics*, vol. 13, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:192519080

[6] J.-S. Wang, G.-R. Shih, and W.-C. Chiang, "Sleep stage classification of sleep apnea patients using decision-tree-based support vector machines based on ecg parameters," in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*. IEEE, 2012, pp. 285–288.

[7] E. Rajbhandari, A. Alsadoon, P. Prasad, I. Seher, T. Q. V. Nguyen, and D. T. H. Pham, "A novel solution of enhanced loss function using deep learning in sleep stage classification: predict and diagnose patients with sleep disorders," *Multimedia Tools and Applications*, vol. 80, pp. 11 607–11 630, 2021.

[8] Y. Li, Z. Xu, Y. Zhang, Z. Cao, and H. Chen, "Automatic sleep stage classification based on a two-channel electrooculogram and one-channel electromyogram," *Physiological Measurement*, vol. 43, no. 7, p. 07NT02, 2022.

[9] J. Fan, C. Sun, M. Long, C. Chen, and W. Chen, "Eognet: A novel deep learning model for sleep stage classification based on single-channel eog signal," *Frontiers in Neuroscience*, vol. 15, p. 573194, 2021.

[10] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[11] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PloS one*, vol. 14, no. 5, p. e0216456, 2019.

[12] E. Eldele, Z. Chen, C. Liu, M. Wu, C.-K. Kwoh, X. Li, and C. Guan, "An attention-based deep learning approach for sleep stage classification with single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 809–818, 2021.

[13] Y.-L. Hsu, Y.-T. Yang, J.-S. Wang, and C.-Y. Hsu, "Automatic sleep stage recurrent neural classifier using energy features of eeg signals," *Neurocomputing*, vol. 104, pp. 105–114, 2013.

[14] A. Supratak and Y. Guo, "Tinysleepnet: An efficient deep learning model for sleep stage scoring based on raw single-channel eeg," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 641–644.

[15] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," *arXiv preprint arXiv:2106.14112*, 2021.

[16] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, X. Li, and C. Guan, "Self-supervised contrastive representation learning for semi-supervised time-series classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 15 604–15 618, 2023.

[17] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberyé, "Sleep-edf database expanded," *physionet. org*, 2018.

[18] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of biomedical engineering*, vol. 44, pp. 1587–1597, 2016.

[19] R. B. Berry, R. Brooks, C. Gamaldo, S. M. Harding, R. M. Lloyd, S. F. Quan, M. T. Troester, and B. V. Vaughn, "Aasm scoring manual updates for 2017 (version 2.4)," pp. 665–666, 2017.

[20] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky relu," in *2020 IEEE Symposium on Computers and communications (ISCC)*. IEEE, 2020, pp. 1–7.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[22] M. Lee, "Gelu activation function in deep learning: A comprehensive mathematical analysis and performance," *arXiv preprint arXiv:2305.12073*, 2023.

[23] A. Paul, R. Bandyopadhyay, J. H. Yoon, Z. W. Geem, and R. Sarkar, "Sinlu: Sinu-sigmoidal linear unit," *Mathematics*, vol. 10, no. 3, p. 337, 2022.



Tan Thuy Trang received an engineering degree in Information Systems from the University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam in 2023.
Email: 19522384@gm.uit.edu.vn



Nguyen Thi Hong received an engineering degree in Information Systems from the University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam in 2023.
Email: 19521550@gm.uit.edu.vn



Duong Phi Long has been working as a Lecturer in the Faculty of Information Systems of the University of Information Technology - Vietnam National University Ho Chi Minh City since 2012. He received her Master of Science degree from the University of Information Technology - Vietnam National University Ho Chi Minh City. His area of interest is Machine Learning, Deep Learning, Computer Vision, and Bioinformatics
Email: longdp@gm.uit.edu.vn

DaNangVMD: Vietnamese Speech Mispronunciation Detection

Nguyen Ket Doan[1], Tran Nguyen Anh[1], Vo Van Nam[1], Nguyen Tran Tien[1], Le Pham Tuyen[2], Nguyen Quoc Vuong[3] and Nguyen Huu Nhat Minh[1]

[1] The University of Danang, Vietnam - Korea University of Information and Communication Technology, Vietnam

[2] Industrial University of Ho Chi Minh City, Ho Chi Minh, Vietnam

[3] Dong A University, Danang, Vietnam

Correspondence: Nguyen Huu Nhat Minh, nhnminh@vku.udn.vn

Received: 04/2/2024, revised: 19/5/2024, accepted: 27/5/2024

Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1271

Abstract: Automatic Speech Recognition, also known as ASR, has grown exponentially over the past decade and is used to recognize and translate human speech into readable text automatically. However, Vietnamese Speech Recognition faces critical challenges such as frequent mispronunciations as well as a huge variant in Vietnamese speech. In this work, we dive into the difficult challenge of Mispronunciation Detection (MD) in the Vietnamese language. As such a tonal language, Vietnamese is not only based on consonants and vowels but also on variations in pitch or tone during pronunciation. In this paper, we propose DaNangVMD model for detecting mispronunciations in Vietnamese speech based on the audio speech and canonical transcript. By leveraging multi-head attention-based multimodal representation from the embeddings of the phonetic encoder and linguistic encoder, DaNangVMD aims to provide a robust solution for accurate mispronunciation detection and diagnosis. Throughout the extensive evaluation, the proposed DaNangVMD exhibits superior performances rather than that of the PAPL baseline models by 15% in F1 score and 13% in accuracy.

Keywords: *mispronunciation detection, multimodal embeddings, Vietnamese speech recognition.*

Tiêu đề: **DaNangVMD: Nhận diện phát âm sai tiếng Việt**

Tóm tắt: Nhận diện giọng nói tự động, còn được gọi là ASR, đã phát triển mạnh mẽ trong thập kỷ qua và được sử dụng để nhận diện và dịch giọng nói của con người thành văn bản đọc được một cách tự động. Tuy nhiên, nhận diện giọng nói tiếng Việt gặp phải những thách thức lớn như lỗi phát âm thường xuyên cũng như sự biến đổi lớn trong giọng nói tiếng Việt. Trong công trình này, chúng tôi giải quyết thách thức của việc phát hiện lỗi phát âm (MD) trong tiếng Việt. Là một ngôn ngữ có thanh điệu, tiếng Việt không chỉ dựa trên các phụ âm và nguyên âm mà còn phụ thuộc vào sự thay đổi trong cao độ hoặc thanh điệu trong quá trình phát âm. Trong bài báo này, chúng tôi đề xuất mô hình DaNangVMD để phát hiện lỗi phát âm trong giọng nói tiếng Việt dựa trên âm thanh giọng nói và văn bản đúng. Bằng cách tận dụng biểu diễn đa mô hình dựa trên sự chú ý đa đầu từ các mã hóa của bộ mã hóa ngữ âm và bộ mã hóa ngôn ngữ, DaNangVMD nhằm cung cấp một giải pháp mạnh mẽ cho việc phát hiện và chẩn đoán lỗi phát âm chính xác. Qua đánh giá mở rộng, DaNangVMD đề xuất cho thấy hiệu suất vượt trội so với các mô hình PAPL với mức tăng 15% trong điểm F1 và 13% trong độ chính xác.

Từ khóa: *nhận diện phát âm sai, biểu diễn đa phương thức, nhận diện giọng nói tiếng Việt*

I. INTRODUCTION

Nowadays, organizations, individuals, and businesses increasingly embrace AI as a supportive tool, voice input assumes a pivotal role in the operations. Voice input facilitates tasks such as searching, and popular social media platforms utilize automatic speech recognition to generate automatic subtitles for videos posted on platforms like YouTube, Facebook, TikTok, among others. However, there are still many errors in recognition due to the system not yet fully optimized for Vietnamese speech. Furthermore, there more and more foreigners working in Vietnam and preschool children who have a huge demand to study Vietnamese language. However, current language learning applications only focus on English pronunciation and do not focus on Vietnamese pronunciation.

Mispronunciation Detection (MD) is an crucial application of phoneme recognition which is a core of Computer-Aided Pronunciation Training (CAPT). This module plays

an important role in speech recognition for tonal languages such as Vietnamese and has recently attracted significant attention from researchers and many methods have been proposed to address these issues. However, up to now, results have not been effective due to the complex and variable nature of the target being recognized as human speech. A multitude of factors, including linguistic differences, syntactical, phonetic structures, pronunciation, and intonation contribute to this complexity. In the initial approach, a promising ASR end-to-end structure for the mispronunciation detection task, called CNN-RNN-CTC [1] was proposed with the phoneme representation by associating raw acoustic features with the corresponding pronounced phoneme sequences, facilitated by the utilization of the Connectionist Temporal Classification (CTC) loss function [1], showed superior performance, surpassing the outcomes of Kun Li et al [2]. However, researchers were not utilizing the pre-existing prior texts from canonical sequences.

In the MD task, because the canonical phoneme sequences are available, Linguistic data from the canonical text can be helpful to integrate into models to improve the performance. SED-MDD [3] was proposed by leveraging the attention mechanism to combine acoustic features and corresponding canonical character sequences. Besides, Kaiqi et al [4] proposed the model by combining the acoustic feature with the encoded phoneme features to improve the linguistic embedding, but the low-level features of the acoustic encoder might be sensitive to noise and affect the model performance. Thus, the Acoustic-Phonetic-Linguistic (APL) [5] approach has dramatically improved the performance of MD and achieved state-of-the-art results by including the phonetic embedding to the model. Moreover, due to the tonal nature of the Vietnamese speech, the APL architecture alone is inadequate for addressing its complexity and variety. Identifying tone is also a challenge for MD tasks in Vietnamese. In fact, the pitch feature, which represents the high and low frequency, is measured in the time and frequency domain which can provide useful features to identify the tone. An APL-based architecture, called PAPL leverages the pitch feature, extracts pitch by utilizing the Kaldi toolkit [6], and achieves promising results for VMD. In this work, we propose the DaNangVMD architecture for Vietnamese Speech mispronunciation detection by customizing the APL [5], and PAPL [6] architecture by utilizing Phonetic and linguistic embeddings for a better and more robust MD solution. Different from APL, PAPL we introduce Phoneme type and multi-head attention across modals data (i.e., speech and transcript), and eliminate unnecessary components in the models to produce the smaller model with a better performance in VMD by a large margin.

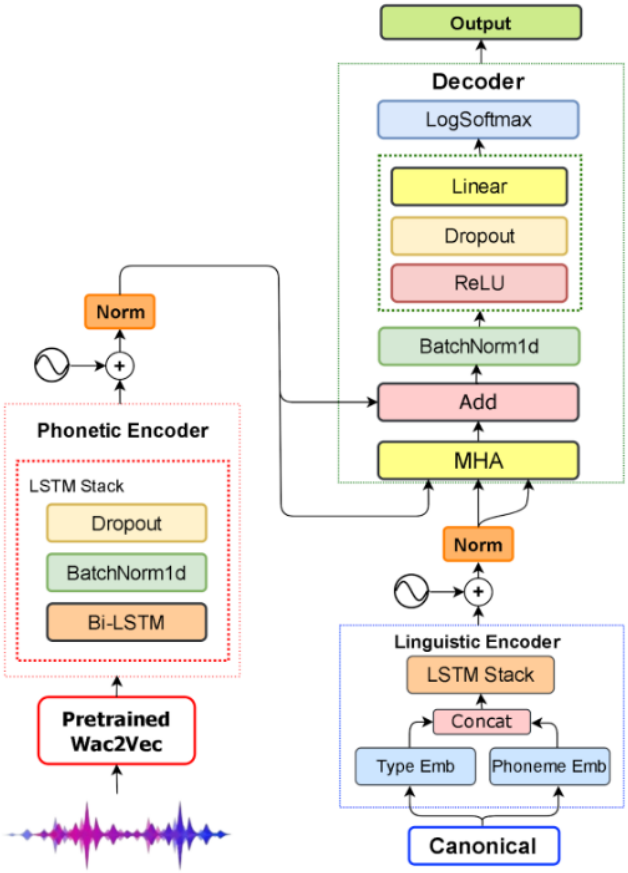


Figure 1. The proposed DaNangVMD for Vietnamese Speech mispronunciation detection

II. THE PROPOSED DANANGVMD ARCHITECTURE

1. Overview of Vietnamese phonemes

An overview of Vietnamese syllables is illustrated in Figure 2, where the hierarchical structure [7] is proposed. Each syllable of Vietnamese is divided into 3 parts: Initial, Rhyme, and Tone. Rhyme is broken down into Medial, Nucleus, and Ending. There are six tones in Vietnamese: Mid-Level Tone (no tone mark), Low Falling Tone (‘), High Rising Tone (´), Low Rising Tone (?), High Broken Tone (~), and Heavy Tone (.). This study maps each Vietnamese syllable to a phoneme, encompassing 53 phonemes. This mapping preserves the hierarchical structure of syllables, facilitating the detection of various types of mispronunciations. The example of syllable conversion to phoneme is presented in Figure 3. Because the phoneme can keep the original hierarchical structure of syllables, this mapping can allow us to detect all types of mispronunciation. In

the DaNangVMD, we will incorporate the phoneme type to better represent the canonical transcript to preserve the basic structures of Vietnamese speech.

2. DaNangVMD model

a) 2.2.1. Baseline model

We have chosen the recent PAPL architecture [4] which has been considered the state-of-the-art for Vietnamese MD as the baseline model for detecting and diagnosing mispronunciations. The PAPL approach integrates pitch, acoustic, phonetic, and linguistic features into the common embeddings for the end-to-end training with a CTC loss function. This enables the model to effectively learn and extract patterns, enhancing its performance in addressing Vietnamese mispronunciations. While conducting the experiments, we discovered that extracted acoustic and pitch features are sensitive to noise and have the potential to yield a negative impact on the overall output, especially in the small training dataset. On the other hand, the phoneme and tone features are already represented in the pre-trained model Wav2Vec2 for Vietnamese speech recognition using a large speech dataset [11]. This pre-trained model brings an incredible result in extracting phonetic embedding even when the data includes noises. Therefore, we decide to eliminate the acoustic and pitch encoder from PAPL architecture and retain the phonetic encoder for learning the relationships between audio frames. Additionally, we customize the linguistic encoder with the extra information from the given transcript and the decoder for phoneme recognition. The sections below will discuss all the components appearing in our proposed model.

b) 2.2.2 Detailed architecture

Phonetic Encoder

In the phonetic encoder, we utilize Wav2Vec2.0 [9] one of the most powerful ASR models for extracting the phonetic embedding which was fine-tuned on a large Vietnamese Speech dataset [11]. The phonetic embedding is fed into an encoder designed with a Bi-LSTM layer, a BatchNorm layer, and a Dropout layer. Additionally, we also leverage the Positional Encoding [14] - this technique provides the position of each audio frame, and this information enhances the output of the LSTM stack. Finally, we normalize this result by the Norm layer and achieve a matrix $H_q = [h_1^q, ..., h_T^q]$. This process is illustrated in Figure 1. We detected the redundancy of CNN layer of the phonetic encoder due to their already existed in Wav2Vec2.0 and produce low performance. As the results, we can reduce the model size significantly.

Linguistic Encoder

In the MD task, the language encoder is also a necessary component in identifying incorrectly pronounced phonemes. The input of this encoder will be canonical phoneme sequences (i.e., transcript) embedded into one hot embedding vector. Additionally, utilizing the genre of canonical phonemes such as Initial, Medial, Nucleus, Ending, and Tone is also crucial to enhance the robustness of canonical representation. It helps the model grasp the structure of phoneme sequences and prior information on identifying phonemes, as shown in Figure 1. Canonical phoneme embedding and phoneme type embedding will be concatenated and passed through a Bi-LSTM. The output of the Bi-LSTM $H_k = H_v = [h_1^{k,v}, ..., h_N^{k,v}]$ will be used as keys and values in the cross attention of the decoder. The N symbol represents the number of phonemes in the sentence.

Decoder

For the decoder, we implement Multi-Head cross Attention 1 (MHA) instead of the single-head attention mechanism utilized in the baseline model. The vector H_q from the Phonetic decoder and H_k, H_v from the Linguistic decoder are transmitted to the Multi-Head Attention as follows:

$$\begin{aligned} &\textbf{MHA}(H_q, H_k, H_v) = \textbf{Concat}(head_1, ..., head_h)W^O \\ &\text{and } head_i = \textbf{Attention}(H_q W_j^{H_q}, H_k W_j^{H_k}, H_k W_j^{H_k}) \end{aligned} \quad (1)$$

This mechanism enables the model to match the relevant linguistic features corresponding to specific segments of the extracted phonetic features. Thus, multiple cross-relations were built upon MHA to strengthen the correlation between phonetics and linguistic features which implicitly represented in the attention score and helped to boost pronunciation recognition performance. In the mispronounced cases, the phonetic feature needs to be emphasized to provide additional information for the decoder, hence the output of MHA is added to H_q. Thereby enhancing the model's capacity to attend to mispronounced phonemes and phonetic features. Subsequently, the combined output undergoes BatchNorm, Relu, Linear, Dropout, and finally, the LogSoftmax layer to compute probabilities based on audio frames. This entire process is depicted in Figure 1 to illustrate the stages of our model architecture.

III. DATASETS

In this work, we conduct the training and testing for the MD task of Vietnamese speech with two MD datasets [13] for detecting mispronunciation. The first dataset [10] contains augmented recordings of adults speaking pairs of

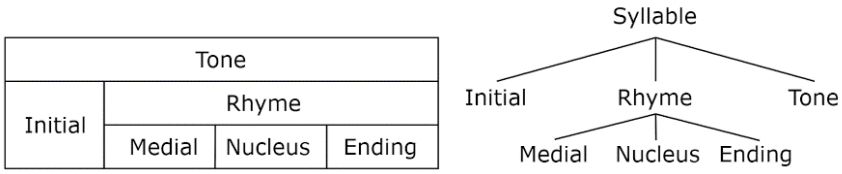


Figure 2. The hierarchical structure of Vietnamese syllables

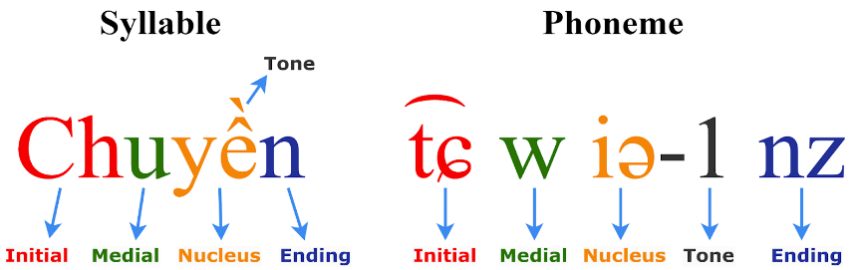


Figure 3. Example of a Vietnamese phoneme presentation.

single-syllable Vietnamese words (released by MachinaX). The second dataset [4] features recordings of children aged five to seven, either speaking or reading Vietnamese sentences in passages or dialogues (released by SoICT-HUST and the Vietnam Psycho-Pedagogical Association). For the first dataset, it is created by a set of pairs of single-syllable words. Therefore, a pair may or may not have a meaning but be easier to pronounce. Each pair will be generated into five other variations, corresponding to the overall six tones in Vietnamese: mid-level, high-rising, low-falling, high-rising-glottal, low-falling-rising, low-falling-glottal. These pairs of single-syllable words will later be recorded by a native speaker. For the second dataset, a group of native children who come from both kindergarten and primary schools participated in creating this dataset. The primary school children were recorded reading sample sentences that were collected from various Vietnamese schoolbooks. There are 20 annotators who were trained to evaluate the entire dataset and mark any instances of incorrect or defective pronunciation. There are a total of 4983 audio, of which 4263 are recordings of native children's voices, the rest are adult voices as shown in Table I.

Children aged from five to seven whose pronunciation varied from very poor and adults deliberately mispronouncing pairs of single-syllable Vietnamese words (such as tone, initial, etc.) as the testing set as described in Table II.

Table I
DETAILS OF THE COLLECTED DATASET [6]

Properties	Children	Adult
# of Recording	4263	746
# of Hours	5.68	0.36

Table II
DETAILS OF THE VIETNAMESE DATASET USED IN THE EXPERIMENTS

Properties	Training	Testing
# of Recording	4101	880

IV. EXPERIMENTS

1. Experimental Setups

In this work, we validate nine models for evaluation and incrementally find out based on the ablation study the best architecture for DaNangVMD:

- **PAPL Baseline:** Pitch - Acoustic - Phonetic - Linguistic model [6].
- **PPL**: Using Pitch features (Kaldi library [15]), Phonetic embeddings, and Linguistic embeddings (canonical phoneme sequences) as input and not using Acoustic features.
- **APL**: Taking Acoustic features (mel spectrogram), Phonetic embeddings, and Linguistic embeddings as input and not using Pitch features.
- **Customized PAPL (C-PAPL)**: Similar to the baseline PAPL model [6], but without using CNN in the Phonetic Encoder.
- **C-PL**: Use only Phonetic embeddings and Linguistic embeddings as input and don't use CNN in the Phonetic Encoder.
- **C-PL1**: Similar to the C-PL model, but using additional Phoneme Type Embedding in the Linguistic

Encoder.

- **C-PL2**: Similar to the C-PL model, but using an LSTM Layer (LSTM → Norm → Dropout) instead of LSTM in the Linguistic Encoder.
- **C-PL3**: Similar to the C-PL model, but using Multi-Head Attention (MHA) in Decoder.
- **DaNangVMD (Proposed architecture)**: Taking Phonetic embeddings and Linguistic embeddings as input; without using CNN in the Phonetic Encoder; using additional Phoneme Type Embedding and an LSTM Layer in the Linguistic Encoder; using Multi-Head cross Attention (MHA) in Decoder as shown in Figure 1.

The audio files were uniformly sampled at a rate of 16,000Hz, and we derived mel-spectrogram, pitch, and phonetic embeddings using a frameshift of 20*ms* and a frame length of 25*ms*. For model training, we employed the AdamW optimizer, setting a learning rate of 0.00005 with a maximum of 100 epochs.

2. Phoneme Recognition

To align the ASR predictions with human annotations, we utilized the Needleman-Wunsch algorithm [14]. We conducted our evaluation metrics using the formula 2, where “I" represents insertions, “D” represents deletions, “S” represents substitutions, and “N” represents the total number of phonetic units as follows:

$$\begin{aligned} Correctness &= \frac{N-S-D}{N}, \\ Accuracy &= \frac{N-S-D-I}{N} \end{aligned} \tag{2}$$

As a result, Table III shows the outcomes of our phoneme recognition performance from all models. The accuracy for phone recognition of PAPL baseline architecture stands at 79.46%. However, with PPL, the accuracy rises to 80.95%, APL achieves 81.41%, and C-PL sees an improvement to 82.87%. Therefore, this ablation study indicates that we can enhance the model accuracy by omitting Acoustic and Pitch features and excluding CNN in the Phonetic Encoder enhances the model accuracy. This improvement could be attributed to the good quality pre-trained Wav2Vec2.0 model in extracting Acoustic and Pitch features instead of training from the scratch acoustic, pitch encoder from a small mispronunciation dataset. In addition, since the Wav2Vec2.0 model already incorporated CNN layers, hence utilizing CNN in the Phonetic Encoder becomes redundant. Based on this observation, we validated this C-PAPL model without CNN layers in the Phonetic encoder. Consequently, we produced the **C-PL** model, resulting in a better accuracy of 82.87% and a correctness of 84.86%. These findings demonstrate the substantial improvement of the **C-PL** architecture over the baseline PAPL for more precise phoneme recognition as well as substantially reducing the model size.

Building upon the **C-PL** model, we iteratively refined its variations. **C-PL1** exhibited an accuracy of 83.55%, **C-PL2** reached 84.46%, and notably, **C-PL3** with multi-head cross attention achieved an impressive 90.68% accuracy, signifying a substantial improvement. This highlights the effectiveness of incorporating additional Phoneme Type Embedding, employing an LSTM Layer in the Linguistic Encoder, and Multi-Head Attention (MHA) in the Decoder in the architecture. Based on all of the aforementioned observations and enhancements, we built the **DaNangVMD** model, resulting in an accuracy of 92.3% and correctness of 93.53%. This marks a noteworthy increase of 12.84% and 11.39% compared to the original PAPL baseline, respectively. DaNangVM exhibits superior performance compared to the other baselines, particularly in timbre recognition. Moreover, **DaNangVMD** achieves a significant reduction in the number of model parameters, diminishing from 64.6M to 10.4M. This substantial reduction substantially improves the speed of phoneme recognition in the Vietnamese language and training time.

3. Mispronunciation Detection Result

The hierarchical evaluation structure is used to measure the MD system performance. With correct pronunciations of phones, *TA (True Acceptance)* means the recognized phone sequence is matched with the reference text, and *FR (False Rejection)* means the recognized phone sequence differs from the reference. With mispronounced phones, *TR (True Rejection)* means the mispronounced phones are detected, and *FA (False Acceptance)* means failing to recognize the mispronounced phones.

$$\begin{aligned} Recall &= \frac{TR}{FA+TR}, \; Precision = \frac{TR}{FR+TR} \\ F1Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{3}$$

As shown in Table III, the PAPL baseline model achieved the highest Recall of 87.95% but the Precision score was very low at only 29.72%, leading to a low F1 score of only 44.43%. We analyzed the PAPL model results which have a high Recall but low Precision score due to the heavy dependence on the canonical labels and could not detect the mispronounced phoneme. On the other hand, the proposed DaNangVMD model overcomes the above problems and balances the recall and precision performance by leveraging the refined Linguistic Encoder, and Multi-Head Attention in the Decoder of the model. The refined structure reduces

Table III
EXPERIMENTAL RESULTS OF VMD MODELS FOR VIETNAMESE MD TASK. NOTE THAT, WE DIDN'T COUNT THE NUMBER PARAMETERS OF THE PRE-TRAINED VIETNAMESE WAVE2VEC2.0 ENCODER FOR ALL OF THE MODELS

VMD Model	**Params**	**Phoneme Recognition performance (%)**		**Mispronunciation Detection and Diagnosis (%)**		
		Accuracy	**Correctness**	**Recall**	**Precision**	**F1**
PAPL *(Baseline)*	64.6M	79.46	80.91	**87.95**	29.72	44.43
PPL *(without Acoustic)*	46.2M	80.95	82.59	86.79	31.76	46.51
APL *(without Pitch)*	61M	81.41	82.89	85.37	32.26	46.83
C-PAPL *(without CNN in Phonetic)*	29.2M	81.38	82.8	86.25	32.03	46.71
C-PL	8M	82.87	84.96	68.39	31.48	43.12
C-PL1 *(Using Phoneme Type)*	8.6M	83.55	85.34	82.5	35.66	49.8
C-PL2 *(Using LSTM Layer)*	8M	84.46	86.13	83.75	37.39	51.69
C-PL3 *(MHA in Decoder)*	9.6M	90.68	92.9	32.68	47.41	38.69
DaNangVMD *(Proposed)*	10.4M	**92.3**	**93.53**	55.54	**63.73**	**59.35**

the over-dependency on the canonical via MHA and adds a skip connection after MHA from phoneme representation. Hence, the acoustic features impact more on the predictive capabilities of the model. As the results, DaNangVMD outperforms PAPL baseline 14,92% and 34,01% in terms of F1 and Precision, respectively. These results highlight the outstanding performance of our DaNangVMD model, particularly in the realm of mispronounced phoneme recognition within the Vietnamese language.

V. CONCLUSIONS

In this paper, we propose the DaNangVMD architecture designed for detecting mispronunciations in Vietnamese speech. Our findings in removing the CNN layers in the phonetic encoder, integrating Phoneme type embeddings in the Phonetic Encoder, and utilizing MHA in the Decoding, prove significant enhancement of phoneme recognition in the context of the Vietnamese language. The experimental results demonstrate the effectiveness of the proposed DaNangVMD system, by significantly improving Recall, Precision, and F1 scores by 55.54%, 63.73%, and 59.35%, respectively. The overall accuracy reached an impressive 92.3%, representing a substantial advancement compared to the baseline model. In future works, we involve further development of the DaNangVMD system into a more robust and effective speech recognition tool by leveraging diverse and extensive Vietnamese datasets, encompassing regional and ethnic minority voices.

ACKNOWLEDGEMENT

This work is under the support of Vietnam - Korea University of Information and Communication and the dataset is provided by Association for Vietnamese Language and Speech Processing.

REFERENCES

[1] Wai-Kim Leun. "CNN-RNN-CTC BASED END-TO-END MISPRONUNCIATION DETECTION AND DIAGNOSIS." *Department of Systems Engineering and Engineering Management*, https://www1.se.cuhk.edu.hk/ hccl/publications/pub/ICASSP2019-mdd.pdf. Accessed 22 November 2023.

[2] Kun Li, and Helen Meng. "Mispronunciation Detection and Diagnosis in L2 English Speech Using Multi-Distribution Deep Neural Networks." *Human-Computer Communications Laboratory Department of System Engineering and Engineering Management The Chinese University of Hong Kong, Hong Kong SAR*, China, 16 June 2023, https://www1.se.cuhk.edu.hk/ hccl/publications/pub/2014 PID3298385 LK.pdf. Accessed 22 November 2023.

[3] Yiqing Feng, et al. "SED-MDD: Towards Sentence Dependent End-To-End Mispronunciation Detection and Diagnosis." 16 June 2023, https://ieeexplore.ieee.org/document/9052975. Accessed 24 November 2023.

[4] Kaiqi Fu. Jones Lin, Dengfeng Ke, Yanlu Xie, Jinsong Zhang, Binghuai Lin "A Full Text-Dependent End to End Mispronunciation Detection and Diagnosis with Easy Data Augmentation Techniques." arXiv, 17 April 2021, https://arxiv.org/abs/2104.08428. Accessed 24 November 2023.

[5] Wenxuan Ye, et al. "An Approach to Mispronunciation Detection and Diagnosis with Acoustic, Phonetic and Linguistic (APL) Embeddings." arXiv, 14 October 2021, https://arxiv.org/abs/2110.07274. Accessed 24 November 2023.

[6] Huu Tuong Tu, et al. "Mispronunciation detection and diagnosis model for tonal language, applied to Vietnamese." [Dublin, Ireland], 20-24 8 2023, https://www.isca-speech.org/archive/pdfs/interspeech 2023/huu23 interspeech.pdf.

[7] Thi Thu Trang Nguyen. HMM-based Vietnamese Text-To-Speech: Prosodic Phrasing Modeling, *Corpus Design System Design, and Evaluation*, 22 January 2016, https://theses.hal.science/tel-01260884/document. Accessed 24 November 2023.

[8] Maxprotect, https://www.maxprotect.com/. Last accessed 21 Feb. 2024

[9] M. Hammami, Y. Chahir and L. Chen, "WebGuard: Web based adult content detection and filtering system," *In Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, Halifax, NS, Canada, 2003, pp. 574-578.

[10] Hu, Weiming, Haiqiang Zuo, Ou Wu, Yunfei Chen, Zhongfei Zhang, and David Suter. "Recognition of adult images, videos, and web page bags." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 7, no. 1 (2011): 1-24.

[11] Sharma, Preeti, Manoj Kumar, and Hitesh Sharma. “Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation." *Multimedia Tools and Applications* 82, no. 12 (2023): 18117-18150.
[12] Soliman, Mohamed Mostafa, Mohamed Hussein Kamal, Mina Abd El-Massih Nashed, Youssef Mohamed Mostafa, Bassel Safwat Chawky, and Dina Khattab. “Violence recognition from videos using deep learning techniques." *In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 80-85. IEEE, 2019.
[13] Human Action Recognition (HAR) Dataset. https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset. Last accessed 21 Feb. 2024
[14] Caruana, Rich. “Multitask learning." Machine learning 28 (1997): 41-75.
[15] M. Ravanelli, T. Parcollet, Y. Bengio, "The PyTorch-Kaldi Speech Recognition Toolkit", https://arxiv.org/abs/1811.07453. Accessed 19 Nov 2018.



Nguyen Ket Doan is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: nkdoan.20it7@vku.udn.vn



Tran Nguyen Anh is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: anhtn.21it@vku.udn.vn



Vo Van Nam is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: namvv.21it@vku.udn.vn



Nguyen Tran Tien is pursuing the B.Eng. degree in Information Technology from the University of Danang, Vietnam - Korea University of Information and Communication Technology. His research interests include software development, machine learning and deep learning.
Email: nttien.20it6@vku.udn.vn



Le Pham Tuyen received the B.S. degree in computer science from Ho Chi Minh City University of Technology, Vietnam, in 2013, and the Ph.D. degree in computer science and engineering from Kyung Hee University, South Korea, in 2019. He is currently an Principal Research Engineer at AgileSoDA Company and visiting lecturer at Industrial University of Ho Chi Minh City. His current research interests include machine learning, reinforcement learning, combinatorial optimization, and robotics.
Email: tuyen_01036033@iuh.edu.vn



Nguyen Quoc Vuong receive master degree computer science from the University of Da Nang, in 2011. His research interests include software development, machine learing and deep learning.
Email: voung@donga.edu.vn



Nguyen Huu Nhat Minh (M’20) received Ph.D. degree in Computer Science and Engineering from Kyung Hee University, South Korea, in 2020. He continued Post-Doc with Federated Learning and Democratized Learning at Intelligent Networking lab, Kyung Hee University, South Korea. He is Deputy Head of Department of Science, Technology, and International Cooperation, and In charge of Research Program at Digital Science and Technology Institute, The University of Danang – Vietnam - Korea University of Information and Communication Technology, Vietnam. He received the best KHU Ph.D. thesis award in engineering in 2020. He had publications in premier ACM/IEEE journals and conferences. His research interests include wireless communications, federated learning, NLP, and computer vision.
Email: nhnminh@vku.udn.vn