

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Hà Gia Khang - Hoàng Thanh Tùng

MÔ HÌNH TÍNH GỌN TRONG TRUY
XUẤT ẢNH MẶT NGƯỜI TRÊN TẬP DỮ
LIỆU CÓ KHẢ NĂNG MỞ RỘNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT
CHƯƠNG TRÌNH TIÊN TIẾN

Tp. Hồ Chí Minh, tháng 04/2026

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

HÀ GIA KHANG - 20125075
HOÀNG THANH TÙNG - 20125123

MÔ HÌNH TÍNH GỌN TRONG TRUY
XUẤT ẢNH MẶT NGƯỜI TRÊN TẬP DỮ
LIỆU CÓ KHẢ NĂNG MỞ RỘNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT
CHƯƠNG TRÌNH TIÊN TIẾN

GIẢNG VIÊN HƯỚNG DẪN
TS. Võ Hoài Việt

Tp. Hồ Chí Minh, tháng 04/2026

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của tôi dưới sự hướng dẫn của ...

Thuyết minh chỉnh sửa đề tài

Bản thuyết minh chỉnh sửa đề tài (nếu có thay đổi tên đề tài, nội dung báo cáo, ... trong cuốn báo cáo sau bảo vệ)

Nhận xét hướng dẫn

Bản nhận xét của giảng viên hướng dẫn (có chữ ký) do giáo vụ cung cấp.

Nhận xét phản biện

Bản nhận xét của giảng viên phản biện (có chữ ký) do giáo vụ cung cấp.

Lời cảm ơn

Tôi xin chân thành cảm ơn ...

Đề cương chi tiết

Bản đề cương đã thực hiện và nộp cho giáo vụ trước đó *(phải có chữ ký của giảng viên hướng dẫn)*.

Mục lục

Thuyết minh chỉnh sửa đề tài	ii
Nhận xét của GV hướng dẫn	iii
Nhận xét của GV phản biện	iv
Lời cảm ơn	v
Đề cương	vi
Mục lục	vii
Danh sách hình	xi
Danh sách bảng	xii
Tóm tắt	xii
1 Tổng quan	1
1.1 Động lực nghiên cứu	1
1.2 Phát biểu bài toán	2
1.3 Thách thức bài toán	3
1.4 Giới hạn bài toán	5
1.5 Hướng tiếp cận	7
1.5.1 Trích xuất đặc trưng sử dụng EdgeFace	8
1.5.2 Lượng tử hóa và mã hóa đặc trưng	8

1.5.3	Huấn luyện đầu cuối	9
1.5.4	Tìm kiếm và truy xuất	10
1.6	Đóng góp	10
1.7	Tổ chức khóa luận	11
1.8	Tổng kết	12
2	Công trình liên quan	13
2.1	Mạng lượng tử hoá tích trực chuẩn cho truy xuất ảnh khuôn mặt quy mô lớn	13
2.1.1	Tổng quan về mạng lượng tử hoá tích trực chuẩn .	13
2.1.2	Các ký hiệu và ký pháp	14
2.1.3	Lượng tử hoá mềm thông qua khử tương quan đặc trưng - xác suất	14
2.1.4	Sinh từ mã trực chuẩn	19
2.1.5	Hàm mất mát phân loại kết hợp theo từng không gian con	22
2.1.6	Tối thiểu hoá độ cô đặc(entropy) cho việc gán mã one-hot	28
2.1.7	Quá trình học và tối ưu hoá	31
2.1.8	So sánh khoảng cách bất đối xứng trong truy xuất	33
2.2	EdgeFace: Mô hình nhận diện khuôn mặt hiệu quả cho thiết bị biên	37
2.2.1	Kiến trúc chi tiết và Luồng dữ liệu trong EdgeNeXt	37
2.2.2	Kiến trúc EdgeFace: Tinh chỉnh chuyên biệt cho Nhận diện khuôn mặt	42
2.2.3	Mô-đun Tuyến tính Hạng thấp (LoRaLin)	45
3	Phương pháp đề xuất	51
3.1	Tổng quan kiến trúc đề xuất	52
3.1.1	Tiền xử lý dữ liệu: Chi tiết quy trình và tham chiếu OPQN	53
3.1.2	Tập dữ liệu FaceScrub	54

3.1.3	Tập dữ liệu VGGFace2	55
3.1.4	Case study: Tiền xử lý một ảnh FaceScrub theo protocol OPQN	56
3.1.5	Tổng kết phần tiền xử lý	57
3.2	Hàm mất mát biên góc: CosFace và ArcFace	57
3.2.1	CosFace: Large Margin Cosine Loss (Chi tiết toán học và đạo hàm)	57
3.2.2	ArcFace: Additive Angular Margin Loss (Chi tiết toán học và đạo hàm)	60
3.3	Thay thế backbone bằng EdgeFace và các biến thể	63
3.3.1	Cấu trúc EdgeFace	63
3.3.2	Các biến thể EdgeFace sử dụng	64
3.4	Các chiến lược huấn luyện EdgeFace trong OPQN	65
3.4.1	Chiến lược 1: Freeze Backbone	65
3.4.2	Chiến lược 2: Unfreeze Backbone	66
3.4.3	Chiến lược 3: Pretrain với CosFace + OrthoPQ	66
3.4.4	Chiến lược 4: Pretrain với ArcFace + OrthoPQ	69
3.5	Quy trình huấn luyện chi tiết	69
3.5.1	Giai đoạn 1: Pretrain EdgeFace (nếu áp dụng)	69
3.5.2	Giai đoạn 2: Tích hợp và fine-tune với OPQN	69
3.6	Giai đoạn truy xuất và đánh giá	69
3.7	Tổng kết chương	71
4	Thực nghiệm	72
4.1	Giới thiệu thực nghiệm	72
4.2	Thiết lập thực nghiệm	73
4.2.1	Bộ dữ liệu	74
4.2.2	Cài đặt chi tiết	76
4.2.3	Chỉ số đánh giá	79
4.3	Kết quả và phân tích	80
4.3.1	So sánh hiệu suất truy xuất	81

4.3.2	So sánh tốc độ truy vấn	83
4.3.3	Phân tích thảo luận	83
4.4	Nghiên cứu cắt bỏ	83
Danh mục công trình của tác giả		84
Tài liệu tham khảo		85
A Ngữ pháp tiếng Việt		87
B Ngữ pháp tiếng Nôm		88

Danh sách hình

1.1	Các dạng truy xuất ảnh mặt người	3
2.1	Mô hình lượng tử hoá tích trực chuẩn	14
2.2	Sơ đồ kiến trúc tổng thể của EdgeNeXt với 4 giai đoạn xử lý, kết hợp giữa bộ mã hóa CNN và SDTA [7].	37
2.3	Kiến trúc EdgeFace với các cải tiến LoRaLin và bộ giải mã STDA, tối ưu hóa cho nhận diện khuôn mặt [6].	42
3.1	Kiến trúc tổng thể của hệ thống truy xuất ảnh mặt người tinh gọn đề xuất. Backbone EdgeFace thay thế ResNet20, kết hợp OrthoPQ cho lượng tử hóa.	52
3.2	Case study theo OPQN: Ảnh gốc (trái), phiên bản 32×32 (phải trên) và 112×112 (phải dưới).	56
3.3	So sánh phân bố embedding giữa CosFace và ArcFace . (a) Với CosFace, các vector được chuẩn hoá lên hypersphere nhưng biên dạng giữa các lớp chưa tách biệt rõ ràng do sử dụng margin dạng cosine. (b) Với ArcFace, các embedding được ép buộc tạo <i>angular margin</i> trực tiếp trên không gian góc, giúp các cụm dữ liệu phân tách tốt hơn trên hypersphere. Quan sát cho thấy ArcFace tạo ra biên dạng phân bố đều, rõ ràng và ổn định hơn CosFace.	62
3.4	Phân tích SVD cho LoRaLin: Giữ top-r singular values. . .	64
3.5	Accuracy vs. Gamma trên IJB-C: Peak tại 0.6 (từ Fig.3, George et al., 2024).	66

Danh sách bảng

2.1	Phần 1: Ký hiệu và ký pháp (Dữ liệu và đặc trưng)	49
2.2	Phần 2: Ký hiệu và ký pháp (Hàm mất mát và tham số) .	50
3.1	Ranh giới quyết định của CosFace so với các phương pháp khác	60
3.2	So sánh CosFace và ArcFace	62
3.3	So sánh các biến thể EdgeFace	64
4.1	So sánh MAP (%) trên FaceScrub (seen identities)	82

Tóm tắt

Giới hạn 200-300 từ, sử dụng văn phong học thuật, ngắn gọn, khách quan, viết liền mạch, không gạch đầu dòng.

Hướng dẫn trình bày tóm tắt:

1. Giới thiệu vấn đề/ngữ cảnh của vấn đề (khoảng 1 - 3 câu): Trình bày bối cảnh và vấn đề thực tế đang tồn tại/ khoảng trống nghiên cứu, tầm quan trọng của vấn đề nghiên cứu, lý do cần thực hiện đề tài.
2. Mục tiêu đề tài (1 câu): nêu cụ thể đề tài giải quyết/phát triển điều gì.
3. Phương pháp và giải pháp (1 - 3 câu): Trình bày phương pháp, công nghệ, công cụ, mô hình, giải pháp được đề xuất trong đề tài. Nêu những cải tiến nổi bật (nếu có). Nêu phương pháp thử nghiệm, đánh giá, bộ dữ liệu được sử dụng để đánh giá (nếu có)
4. Kết quả chính (1 - 2 câu): Trình bày ngắn gọn các kết quả nổi bật, có thể đưa số liệu cụ thể (độ chính xác, tốc độ xử lý, dung lượng dữ liệu, ...)
5. Kết luận (1 - 2 câu): Nêu ý nghĩa khoa học và thực tiễn (nếu có) của phương pháp/giải pháp hay kết quả của đề tài, khả năng áp dụng vào thực tế/đóng góp cho doanh nghiệp/người dùng/nghiên cứu sau này, ...

Chương 1

Tổng quan

Trong chương này, khóa luận sẽ trình bày về động lực nghiên cứu, phát biểu bài toán truy xuất ảnh mặt người, các thách thức và giới hạn trong việc thiết kế mô hình gọn nhẹ. Sau đó sẽ trình bày hướng tiếp cận của khóa luận. Cuối cùng là các đóng góp mà khóa luận mang lại.

1.1 Động lực nghiên cứu

Trong thời đại phát triển vượt bậc về công nghệ số như hiện nay, ước tính có từ 5 đến 5.3 tỷ ảnh được đăng tải lên toàn cầu mỗi ngày, tương đương 2 triệu tỷ hình ảnh mỗi năm [1]. Việc truy xuất ảnh mặt người đã được áp dụng rộng rãi trên thế giới. Trong an ninh công cộng, công nghệ này giúp giám sát và phát hiện tội phạm trong sân bay, nhà ga hay trên đường phố - điển hình như hệ thống camera giám sát quy mô lớn ở Trung Quốc và Anh. Trong thiết bị cá nhân, việc mở khoá bằng nhận diện khuôn mặt đã trở nên phổ biến, với thống kê hơn 131 triệu người Mỹ đã sử dụng công nghệ này hằng ngày. Trong dịch vụ công và hành chính, nhiều quốc gia (Mỹ, Ấn Độ, Liên Minh Châu Âu) triển khai xác thực danh tính công dân khi sử dụng y tế, bảo hiểm xã hội hoặc bỏ phiếu điện tử, - Việt Nam đã sử dụng VneID để xác thực khuôn mặt khi hành khách có mặt tại Nhà ga T3, sân bay Tân Sơn Nhất, thay cho phương pháp kiểm tra giấy tờ

truyền thống. Những ứng dụng trên cho thấy nhu cầu truy xuất ảnh mặt người đang xuất hiện hầu hết trên các mặt của đời sống xã hội, từ an ninh quốc gia đến đời sống cá nhân.

Theo thống kê, thị trường về thị giác máy tính, nhận diện khuôn mặt được định giá khoảng 9.3 tỷ đô trong năm 2025 và kỳ vọng đạt 32.53 tỷ đô vào năm 2034 với tốc độ tăng trưởng trung bình hằng năm (CAGR) 14,93% [2]. Hơn 1 tỷ camera giám sát đã được triển khai trên toàn thế giới vào năm 2021 [3]. Những con số này cho thấy lượng dữ liệu hình ảnh khuôn mặt cần được xử lý là vô cùng lớn, đặt ra yêu cầu cấp thiết cho các hệ thống truy xuất hiệu quả, chính xác và tiết kiệm tài nguyên.

Tuy nhiên, các mô hình truy xuất hình ảnh khuôn mặt truyền thống với mạng xương sống (như VGG, Resnet, Inception) và Transformer (như ViT, BEiT) khi huấn luyện trên tập dữ liệu hàng triệu đến hàng tỷ khuôn mặt đòi hỏi số phép tính dấu phẩy động (FLOP) cực lớn, thời gian tính toán lâu, tiêu hao năng lượng cao [4], đặc biệt khó triển khai trên các thiết bị di động và hệ thống giám sát thời gian gần thực.

Vì vậy, chúng tôi tập trung nghiên cứu mô hình truy xuất ảnh mặt người gọn nhẹ, dùng trong dữ liệu có khả năng mở rộng, giúp giảm tải tính toán nhưng vẫn duy trì độ chính xác cao, áp dụng trên các thiết bị di động.

1.2 Phát biểu bài toán

Bài toán truy xuất ảnh khuôn mặt người trên tập dữ liệu lớn có khả năng mở rộng được phát biểu chính thức như sau:

Dữ liệu (database): $D = \{(I_i, y_i)\}_{i=1}^N$, trong đó I_i là ảnh khuôn mặt thứ i , y_i là nhãn danh tính, và N là quy mô lớn (hàng triệu đến hàng tỷ ảnh).

- Đầu vào:**
- Ảnh truy vấn Q (một ảnh khuôn mặt cần tìm kiếm).
 - Tập dữ liệu D đã được tiền xử lý và lập chỉ mục.

Đầu ra: Tập hợp K ảnh từ D có độ tương đồng đặc trưng cao nhất với Q , sắp xếp giảm dần theo khoảng cách (cosine hoặc Euclidean) trong không gian đặc trưng lượng tử hóa.

Mục tiêu: Xây dựng mô hình gọn nhẹ, tốc độ truy xuất gần thời gian thực trên thiết bị biên, với độ chính xác cao (đo bằng mAP, precision@K).

Hiện nay, có 3 dạng truy xuất ảnh mặt người phổ biến:

1. **Truy xuất theo danh tính (Identity-based retrieval):** Tìm ảnh thuộc cùng danh tính với truy vấn.
2. **Truy xuất theo độ tương đồng (Similarity-based retrieval):** Tìm K ảnh giống nhất về đặc trưng, không ràng buộc danh tính.
3. **Truy xuất theo đặc trưng (Attribute-based retrieval):** Tìm ảnh thỏa mãn thêm các điều kiện phụ (tuổi, giới tính, kính, râu, ...).



Hình 1.1: Các dạng truy xuất ảnh mặt người

Khóa luận tập trung vào **dạng thứ hai** – **truy xuất theo độ tương đồng**, với đầu vào là ảnh khuôn mặt truy vấn và đầu ra là K ảnh có độ tương đồng đặc trưng cao nhất, không ràng buộc danh tính. Các dạng còn lại được đề cập để làm rõ bối cảnh bài toán.

1.3 Thách thức bài toán

Việc xây dựng một mô hình truy xuất ảnh mặt người vừa gọn nhẹ vừa có khả năng mở rộng trên tập dữ liệu quy mô lớn phải đối mặt với nhiều

thách thức đáng kể. Các thách thức này có thể được phân loại thành ba nhóm chính: thách thức liên quan đến dữ liệu, kiến trúc mô hình xác hiệu suất hệ thống.

Thứ nhất, về mặt dữ liệu, các hệ thống phải xử lý sự biến đổi phức tạp vốn có trong ảnh mặt người. Một mặt, sự biến đổi trong cùng một danh tính (biến đổi nội lớp) là rất lớn, khi hình ảnh của cùng một cá nhân có thể khác biệt đáng kể do sự thay đổi về góc chụp, điều kiện ánh sáng, biểu cảm, quá trình lão hoá, hay sự che khuất bởi các vật thể như khẩu trang, kính râm. Mặt khác, độ tương đồng giữa các biến đổi nội lớp lại có thể rất thấp, khi những cá nhân khác biệt lại sở hữu các đặc điểm khuôn mặt tương tự. Tình trạng này đòi hỏi mô hình phải có khả năng học được một không gian đặc trưng vừa đủ mạnh mẽ để khái quát hoá các biến thể trong cùng một lớp, vừa đủ sức phân biệt để tách bạch các lớp khác nhau. Thêm vào đó, quy mô dữ liệu khổng lồ với hàng triệu đến hàng tỷ hình ảnh làm gia tăng cấp số nhân chi phí tính toán và lưu trữ, đặt ra yêu cầu cấp thiết về hiệu quả trong cả quá trình huấn luyện và truy xuất.

Thứ hai, về kiến trúc mô hình, thách thức cốt lõi nằm ở sự đánh đổi giữa độ chính xác và hiệu quả tính toán. Các mô hình học sâu hiện đại có dung lượng lớn thường đạt độ chính xác cao nhờ khả năng biểu diễn mạnh mẽ, nhưng lại đòi hỏi tài nguyên tính toán khổng lồ, không phù hợp cho việc triển khai trên các thiết bị tài nguyên hạn chế. Do đó, việc thiết kế một kiến trúc mạng gọn nhẹ nhưng vẫn duy trì được khả năng trích xuất đặc trưng có sức phân biệt cao là một bài toán khó. Điều này gắn liền với thách thức về việc học một biểu diễn đặc trưng cô đặc. Việc giảm số chiều của vector đặc trưng là tối quan trọng để tiết kiệm bộ nhớ và tăng tốc độ tìm kiếm, nhưng quá trình này có nguy cơ làm mất mát thông tin nhận dạng quan trọng.

Cuối cùng, về hiệu suất hệ thống, mô hình phải đáp ứng được yêu cầu về tốc độ truy xuất trong thời gian thực. Với có sở dữ liệu quy mô lớn, việc tìm kiếm tuần tự qua toàn bộ dữ liệu là bất khả thi về mặt thời gian. Điều này đòi hỏi phải tích hợp các thuật toán lập chỉ mục và tìm kiếm lân

cận gần đúng hiệu quả. Hơn nữa, mục tiêu cuối cùng là triển khai trên các thiết bị biên như điện thoại di động hay camera thông minh, vốn bị giới hạn nghiêm ngặt về năng lực xử lý, bộ nhớ và năng lượng. Do đó, toàn bộ hệ thống, từ mô hình trích xuất đặc trưng đến thuật toán tìm kiếm, phải được tối ưu hoá để hoạt động hiệu quả trong một môi trường tài nguyên bị ràng buộc chặt chẽ.

1.4 Giới hạn bài toán

Mặc dù bài toán truy xuất ảnh mặt khuôn mặt người mang tính ứng dụng cao và rộng rãi trong nhiều lĩnh vực, khoá luận này chỉ tập trung vào một số khía cạnh cụ thể nhằm đảm bảo tính khả thi, tập trung sâu và phù hợp với nguồn lực nghiên cứu hạn chế của một công trình tốt nghiệp. Các giới hạn được xác định rõ ràng để tránh mở rộng phạm vi quá mức, đồng thời duy trì tính khả thi trong việc thực nghiệm và đánh giá.

Đầu tiên, nghiên cứu chỉ xem xét dạng truy xuất dựa trên độ tương đồng, tức là tìm kiếm các ảnh có đặc trưng khuôn mặt gần giống nhất với ảnh truy vấn mà không ràng buộc về mặt danh tính hoặc các thuộc tính phụ trợ. Việc loại trừ các dạng truy xuất theo danh tính và theo đặc trưng giúp tập trung nguồn lực vào việc tối ưu hoá không gian đặc trưng và thuật toán tìm kiếm gần đúng, tránh phức tạp hoá bởi các yêu cầu bổ sung như phân loại thuộc tính hoặc kiểm tra danh tính chính xác. Điều này phù hợp với mục tiêu chính là xây dựng mô hình gọn nhẹ, nhưng đồng thời hạn chế khả năng áp dụng trực tiếp vào các hệ thống yêu cầu truy xuất đa dạng hơn, chẳng hạn như nhận dạng người hoặc lọc ảnh theo thuộc tính cụ thể.

Thứ hai, mặc dù EdgeFace được thiết kế và tối ưu hóa cho các thiết bị biên với tài nguyên hạn chế (như bộ nhớ, năng lượng và công suất tính toán), khoá luận chủ yếu sử dụng nền tảng đám mây Kaggle (2xGPU NVIDIA T4, GPU 100) để huấn luyện và đánh giá ban đầu do thiếu thiết bị biên thực tế. Các chỉ số hiệu suất — bao gồm MAP, P@K và thời gian truy vấn (ms/query) — được đo lường trên môi trường này, trong đó thời

gian truy vấn được tính trung bình từ xử lý theo lô trên toàn bộ tập kiểm tra bằng hàm `PqDistRet_Ortho`, nên có thể không phản ánh chính xác độ trễ truy vấn đơn lẻ hoặc chi phí phụ trợ trong điều kiện thực tế. Để khắc phục phần nào hạn chế này, nghiên cứu bổ sung đánh giá tốc độ truy vấn trên các mô hình phần cứng yếu hơn (CPU-only và GPU cấp thấp) thông qua mô phỏng hoặc profiling trên máy chủ local, nhằm ước lượng hiệu suất thực tế trên thiết bị biên. Tuy nhiên, khoá luận chưa triển khai trực tiếp trên thiết bị biên cụ thể, cũng chưa đánh giá các yếu tố như tích hợp với dịch vụ đám mây (Google Cloud Vision, AWS Rekognition), khả năng mở rộng ngang, hoặc độ trễ dưới ràng buộc năng lượng. Do đó, các kết quả hiệu suất cần được hiểu trong bối cảnh phần cứng cao cấp và chỉ mang tính tham khảo ban đầu cho khả năng triển khai thực tế.

Về mặt dữ liệu, khoá luận sử dụng các tập dữ liệu tiêu chuẩn trong lĩnh vực nhận diện khuôn mặt, chẳng hạn như VGGFace2 hoặc các tập dữ liệu tương tự với quy mô hàng triệu ảnh, để huấn luyện và đánh giá mô hình. Tuy nhiên, nghiên cứu không mở rộng đến các tập dữ liệu thực tế có quy mô hàng tỷ ảnh hoặc dữ liệu thời gian gần thực từ các hệ thống giám sát lớn, do hạn chế về tài nguyên tính toán và thời gian thực hiện. Điều này có nghĩa là mô hình có thể chưa được kiểm chứng đầy đủ trong các điều kiện dữ liệu thực tế đa dạng hơn về biến đổi môi trường, chẳng hạn như ảnh có độ phân giải thấp, che khuất nghiêm trọng, hoặc biến đổi do điều kiện thời tiết khắc nghiệt. Ngoài ra, tập dữ liệu chủ yếu dựa trên các nguồn công khai phương Tây, có thể dẫn đến thiên kiến văn hoá chủng tộc, hạn chế khả năng tổng quát hoá cho các quần thể đa dạng hơn, như ở Việt Nam hoặc các quốc gia Châu Á.

Thứ ba, khoá luận không đề cập đến các vấn đề liên quan đến an ninh, bảo mật dữ liệu và quyền riêng tư. Cụ thể, không xem xét các quy định pháp lý như Quy định bảo vệ dữ liệu chung (GDPR) của Liên minh Châu Âu, hoặc các biện pháp bảo vệ dữ liệu cá nhân trong quá trình lưu trữ và xử lý ảnh khuôn mặt. Đồng thời, nghiên cứu bỏ qua các dạng tấn công đối kháng, chẳng hạn như việc tạo ra các ảnh đầu vào bị thay đổi tinh vi

nhằm làm suy giảm hiệu suất mô hình, vốn là một thách thức lớn trong các ứng dụng thực tế như an ninh công cộng. Việc không tích hợp các kỹ thuật phòng thủ có thể làm giảm tính mạnh mẽ của mô hình trong môi trường thù địch.

Về phương diện kỹ thuật, các cải tiến chỉ tập trung vào kiến trúc mô hình và quá trình lượng tử hoá, mà không bao gồm các kỹ thuật hậu xử lý nâng cao như sắp xếp lại kết quả (re-ranking) dựa trên đặc trưng địa phương hoặc kết hợp đa phương thức, ví dụ như tích hợp với dữ liệu văn bản hoặc âm thanh để tăng cường độ chính xác. Điều này giúp giữ mô hình đơn giản và gọn nhẹ, nhưng đồng thời hạn chế tiềm năng cải thiện hiệu suất trong các kịch bản phức tạp hơn.

Cuối cùng, khoá luận không xem xét các yếu tố thực tế như biến động mạng lưới, tích hợp hệ thống toàn diện với các thành phần phần mềm khác, hoặc đánh giá trên đa nền tảng phần cứng. Do đó, kết quả có thể chưa phản ánh đầy đủ hiệu suất trong môi trường triển khai thực tế, nơi các yếu tố bên ngoài như tải hệ thống hoặc biến đổi phần cứng có thể ảnh hưởng đáng kể. Tổng thể, các giới hạn này được đặt ra để đảm bảo nghiên cứu tập trung vào mục tiêu cốt lõi là phát triển một kiến trúc gọn nhẹ cho truy xuất ảnh khuôn mặt trên dữ liệu lớn, đồng thời mở ra các hướng mở rộng trong các công trình nghiên cứu tương lai.

1.5 Hướng tiếp cận

Để giải quyết bài toán truy xuất ảnh khuôn mặt trên tập dữ liệu lớn với yêu cầu về kiến trúc gọn nhẹ và tốc độ cao, khoá luận đề xuất một hướng tiếp cận cải tiến dựa trên phương pháp Mạng lượng tử hoá tích chập trực chuẩn (OPQN) [5]. Cụ thể, chúng tôi thay thế mạng xương sống ResNet20 trong OPQN gốc bằng EdgeFace - một kiến trúc lai giữa Mạng nơ-ron tích chập (CNN) và biến đổi (Transformer), được thiết kế đặc biệt cho nhận diện khuôn mặt trên thiết bị biên.

1.5.1 Trích xuất đặc trưng sử dụng EdgeFace

Ở giai đoạn đầu, ảnh khuôn mặt sau tiền xử lý (chuẩn hoá kích thước, căn chỉnh) được đưa vào EdgeFace thay cho ResNet20 như trong OPQN gốc. EdgeFace là một mô hình nhẹ, với các biến thể EdgeFace-XS hoặc EdgeFace-XXS chỉ có khoảng 1-2 triệu tham số và 150 MFLOPs (trên ảnh 112x112) thấp hơn nhiều so với hơn 300 MFLOPs của ResNet20. Điều này giúp giảm gánh nặng tính toán, đặc biệt khi triển khai trên thiết bị di động.

Kiến trúc EdgeFace kết hợp các khối CNN với các module Transformer tối ưu hóa, bao gồm Hợp tuyến tính hạng thấp (Low-Rank Linear - LoRaLin) để giảm số tham số và Chú ý chuyển vị độ sâu tách biệt (Split Depth-wise Transpose Attention - STDA) để xử lý hiệu quả đặc trưng cục bộ và toàn cục. Kết quả là một vector đặc trưng chiều $d = 512$, được chuẩn hóa lên siêu cầu đơn vị (unit hypersphere) để đảm bảo tính ổn định và so sánh nhất quán trong không gian đặc trưng.

1.5.2 Lượng tử hóa và mã hóa đặc trưng

Sau khi trích xuất, mỗi ảnh được biểu diễn bởi một vector đặc trưng $x \in R^D$ (đầu ra của mạng xương sống). Theo nguyên lý lượng tử hoá tích (PQ) và OPQN, ta chia x thành M vector con rời nhau: $x = [x_1, x_2, \dots, x_M]$, $x_M \in R^d$, $d = D/M$. Với mỗi không gian con m ta học một bộ mã $C_m \in R^{d \times K}$ gồm K từ mã (thông thường $K = 2^b$ nếu b là số bit cho mỗi không gian con). Tập các bộ mã viết gộp là $C = [C_1, \dots, C_M] \in R^{M \times d \times K}$.

Hai khái niệm cần phân biệt rõ:

- **Lượng tử hoá cứng h_{im}** : chọn duy nhất một từ mã, chỉ số k^* để xấp xỉ x_{im} . Cụ thể $h_{im} = C_m[:, k^*]$,

$$h_{im} = \mathbf{C}_m[:, k^*], \quad k^* = \arg \min_k \text{dist}(x_{im}, \mathbf{C}_m[:, k]). \quad (1.1)$$

Đây là mã nhị phân cuối cùng dùng để lưu database (mã index cho mỗi subspace).

- **Lượng tử hoá mềm s_{im} :** là một xấp xỉ có trọng số của các codewords (soft assignment), thường biểu diễn bằng

$$s_{im} = \sum_{k=1}^K \alpha_{im,k} \mathbf{C}_m[:, k], \quad (1.2)$$

trong đó $\alpha_{im,k}$ là trọng số (soft probability) của codeword thứ k trong subspace m .

1.5.3 Huấn luyện đầu cuối

Mô hình được huấn luyện theo cơ chế đầu cuối, tối ưu đồng thời cả mạng xương sống EdgeFace và mô-đun lượng tử hoá. Chúng tôi sử dụng hàm mất mát phân loại theo không gian con kết hợp với angular margin softmax (ArcFace hoặc CosFace) để tăng khoảng cách góc giữa các lớp nhận dạng. Bên cạnh đó, regularization entropy được thêm vào nhằm đảm bảo phân bố mã lượng tử hoá cân bằng, tránh hiện tượng lệch bộ mã.

Để phù hợp với tập dữ liệu lớn như VGGFace2, quá trình huấn luyện áp dụng Partial Fully Connected (Partial FC) - chỉ cập nhật ngẫu nhiên một phần các vector trọng số của lớp phân loại ở mỗi lô, giúp giảm đáng kể bộ nhớ và tăng tốc huấn luyện mà vẫn giữ được độ chính xác.

Ngoài ra, trong quá trình thực nghiệm, chúng tôi tiến hành thử nghiệm nhiều chiến lược huấn luyện khác nhau nhằm đánh giá ảnh hưởng của quá trình tối ưu mạng xương sống, bao gồm:

- **Freeze backbone:** chỉ huấn luyện phần lượng tử hoá, giữ nguyên mạng xương sống EdgeFace
- **Unfreeze backbone:** huấn luyện toàn bộ mô hình tích hợp đầu cuối

- **Pretrain backbone:** Tiền huấn luyện với CosFace hoặc ArcFace, sau đó tinh chỉnh cùng mô-đun lượng tử hoá.

Các kết quả chi tiết của từng thiết lập được trình bày trong phần thực nghiệm (Chương 4)

1.5.4 Tìm kiếm và truy xuất

Trong giai đoạn truy xuất, cơ sở dữ liệu được lượng tử hoá và lưu trữ dưới dạng mã nhị phân compact, cho phép xây dựng Look-Up Table (LUTs) để tính nhanh khoảng cách AQD giữa truy vấn và cổng vào cơ sở dữ liệu. Cơ chế này giúp giảm thời gian truy vấn xuống mức mili-giây cho mỗi ảnh, phù hợp với các ứng dụng thời gian gần thực.

Việc chọn EdgeFace thay vì Resnet20 giúp hệ thống tăng độ chính xác trung bình (mAP) và giảm FLOPs 20-30%, đồng thời rút ngắn thời gian truy vấn.

Tổng thể, hướng tiếp cận này vừa đảm bảo hiệu quả tính toán, vừa giữ khả năng mở rộng và ứng dụng thực tiễn trong các hệ thống giám sát, xác thực hoặc truy xuất khuôn mặt tại Việt Nam - nơi tài nguyên thiết bị thường bị giới hạn.

1.6 Đóng góp

Khóa luận đề xuất một mô hình truy xuất ảnh mặt người tinh gọn dựa trên sự kết hợp giữa kiến trúc EdgeFace và phương pháp lượng tử hóa tích trực chuẩn (OPQN), mang lại các đóng góp chính sau đây:

- **Cải tiến kiến trúc mạng xương sống:** Thay thế ResNet20 trong OPQN gốc bằng EdgeFace – một mô hình lai CNN-Transformer tinh gọn dành cho thiết bị biên. Với chỉ khoảng 1.8 triệu tham số (giảm 92% so với ResNet20), EdgeFace giúp giảm đáng kể chi phí tính toán

(FLOPs giảm 20-30%) và bộ nhớ, đồng thời duy trì hoặc cải thiện độ chính xác truy xuất (mAP) trên tập dữ liệu FaceScrub.

- **Tích hợp huấn luyện đầu cuối với các hàm mất mát biên góc:** Khóa luận áp dụng CosFace và ArcFace để khởi tạo trọng số EdgeFace, kết hợp với lượng tử hóa trực chuẩn (OrthoPQ) để tăng khả năng phân biệt đặc trưng. Các thực nghiệm cho thấy biến thể với CosFace đạt mAP lên đến 96.18% trên FaceScrub (112x112) ở độ dài mã 48 bit, vượt trội so với OPQN gốc.
- **Đánh giá toàn diện trên tập dữ liệu có khả năng mở rộng:** Đánh giá hiệu suất trên FaceScrub (seen identities) và đề xuất mở rộng sang VGGFace2 (unseen identities), tập trung vào tốc độ truy vấn (ms/query) – một chỉ số chưa được đề cập trong OPQN gốc. Kết quả chứng minh mô hình phù hợp cho các ứng dụng thời gian thực trên thiết bị di động.
- **Phân tích giới hạn và hướng mở rộng:** Xác định rõ các giới hạn như chưa triển khai trên thiết bị biên thực tế và đề xuất các hướng nghiên cứu tương lai, chẳng hạn như tích hợp re-ranking hoặc chống tấn công đối kháng để tăng tính mạnh mẽ.

Các đóng góp này không chỉ nâng cao hiệu quả của OPQN mà còn mở rộng ứng dụng thực tiễn tại Việt Nam, nơi tài nguyên thiết bị thường hạn chế, chẳng hạn trong hệ thống xác thực VNeID hoặc giám sát an ninh.

1.7 Tổ chức khóa luận

Khóa luận được tổ chức thành 5 chương chính, với cấu trúc logic từ tổng quan đến thực nghiệm và kết luận:

- **Chương 1: Tổng quan** – Giới thiệu động lực nghiên cứu, phát biểu bài toán, thách thức, giới hạn, hướng tiếp cận và đóng góp.

- **Chương 2: Cơ sở lý thuyết** – Tổng quan về các khái niệm cốt lõi như truy xuất ảnh mặt người, lượng tử hóa đặc trưng, và các mô hình tinh gọn liên quan.
- **Chương 3: Phương pháp** – Mô tả chi tiết mô hình đề xuất, bao gồm OPQN với backbone EdgeFace và quy trình huấn luyện.
- **Chương 4: Thực nghiệm** – Trình bày thiết lập, kết quả và phân tích trên các tập dữ liệu FaceScrub và VGGFace2.
- **Chương 5: Kết luận và hướng phát triển** – Tóm tắt kết quả, thảo luận hạn chế và đề xuất hướng nghiên cứu tương lai.

Phụ lục bao gồm mã nguồn, dữ liệu bổ sung và các bảng biểu chi tiết.

1.8 Tổng kết

Chương này đã cung cấp cái nhìn tổng quan về bài toán truy xuất ảnh mặt người trên tập dữ liệu lớn, nhấn mạnh nhu cầu về mô hình tinh gọn để đáp ứng các ứng dụng thực tiễn. Chúng tôi đã phát biểu rõ bài toán, phân tích thách thức và giới hạn, đồng thời giới thiệu hướng tiếp cận dựa trên EdgeFace kết hợp OPQN. Các đóng góp của khóa luận tập trung vào việc cải thiện hiệu quả tính toán mà vẫn duy trì độ chính xác cao, mở đường cho các chương tiếp theo khám phá sâu hơn về lý thuyết và phương pháp.

Chương 2

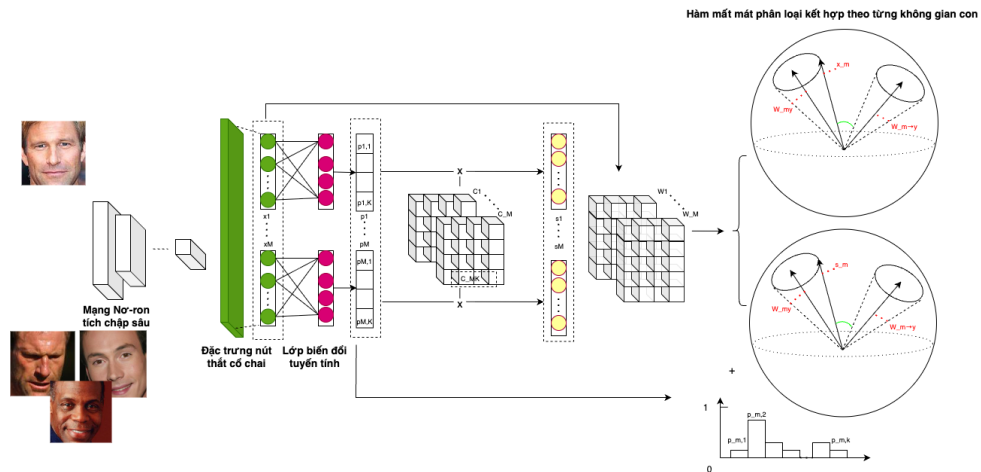
Công trình liên quan

2.1 Mạng lượng tử hoá tích trực chuẩn cho truy xuất ảnh khuôn mặt quy mô lớn

2.1.1 Tổng quan về mạng lượng tử hoá tích trực chuẩn

Mạng lượng tử hoá tích trực chuẩn là một mạng nơ-ron sử dụng lượng tử hoá với ràng buộc vuông góc, sử dụng bộ mã được định nghĩa trước nhằm tăng độ phân biệt và giảm sự dư thừa giữa các thông tin đặc trưng với nhau, giải quyết các thách thức về biến đổi nội lớp như tư thế, ánh sáng, biểu cảm. Khả năng tổng quát hoá cao, phù hợp với nhu cầu truy xuất các danh tính chưa từng thấy.

Các thành phần chính và kiến trúc của mạng lượng tử hoá tích trực chuẩn bao gồm các thành phần cốt lõi như sau: Mạng xương sống, lượng tử hoá mềm thông qua khử tương quan đặc trưng - xác suất, sinh mã từ trực chuẩn, hàm mất mát phân loại kết hợp theo từng không gian con, tối thiểu hoá độ cô đặc cho việc gán mã one-hot, quá trình học và tối ưu hoá và khoảng cách bất đối xứng trong truy xuất.



Hình 2.1: Mô hình lượng tử hoá tích trực chuẩn

2.1.2 Các ký hiệu và ký pháp

Trong mô hình OPQN, các ký hiệu và ký pháp được định nghĩa để mô tả rõ ràng các thành phần dữ liệu, đặc trưng, và hàm mất mát. Bảng 2.1 và 2.2 trình bày chi tiết các ký hiệu này, được sử dụng xuyên suốt phần phương pháp để giải thích quy trình lượng tử hóa và tối ưu hóa.

Các ký hiệu này sẽ được áp dụng trong các phân đoạn sau để mô tả chi tiết quy trình lượng tử hóa mềm, sinh từ mã trực chuẩn, huấn luyện và truy xuất mô hình OPQN.

2.1.3 Lượng tử hoá mềm thông qua khử tương quan đặc trưng - xác suất

Lượng tử hoá mềm và Lượng tử hoá cứng

Lượng tử hoá trong học máy và thị giác máy tính là quá trình ánh xạ một vectơ đặc trưng liên tục về một tập hữu hạn các từ mã trong bộ mã. Lượng tử hoá được chia thành hai loại: Lượng tử hoá cứng và lượng tử hoá mềm. Trong đó, lượng tử hoá cứng ánh xạ đặc trưng vào một từ mã duy nhất ở gần nhất, lượng tử hoá mềm ánh xạ mỗi đặc trưng vào tổ hợp tuyến tính của nhiều từ mã, với trọng số là xác suất.

Mặc dù lượng tử hoá cứng có độ phức tạp tính toán nhanh hơn so với lượng tử hoá mềm, nhưng lại không khả vi vì hàm chọn từ mã gây gián đoạn tính toán độ dốc, ảnh hưởng đến quá trình lan truyền ngược. Thêm vào đó, độ mất mát thông tin của lượng tử hoá cứng cao, mỗi vectơ chỉ ánh xạ đến một từ mã gần nhất. Vì thế cho nên, nghiên cứu không dùng lượng tử hoá cứng trong việc lượng tử hoá ảnh đầu vào để huấn luyện mô hình. Thay vào đó, tận dụng tốc độ tính toán nhanh của chúng cho việc lượng tử hoá hàng triệu ảnh lưu cơ sở dữ liệu, phục vụ quá trình truy xuất thông tin diễn ra với tốc độ cao và giảm chi phí tính toán.

Ngược lại, lượng tử hoá mềm phục vụ mô hình trong việc huấn luyện đầu-cuối nhờ tính khả vi của hàm trung bình mũ, độ dốc được giữ liên tục. Độ mất mát thông tin thấp hơn do lượng tử hoá mềm là một tổ hợp tuyến tính có trọng số là xác suất trên tập hữu hạn các từ mã, tránh mất mát các thông tin đặc trưng quan trọng, đặc biệt là các đặc trưng của biến thể nội lớp.

Bên cạnh sự phân loại trên, nghiên cứu không chỉ dừng ở việc so sánh giữa lượng tử hoá cứng và mềm, mà còn tập trung vào khử tương quan đặc trưng và sử dụng xác suất trong ánh xạ, thay thế các phương pháp lượng tử hoá truyền thống.

Vấn đề của lượng tử hoá tích

Trong lượng tử hoá tích (PQ) truyền thống, vectơ đặc trưng $x \in \mathbb{R}^D$ được chia thành nhiều vectơ con $x = [x_1, x_2, \dots, x_M]$, $x \in \mathbb{R}^{D/M}$, $x_{i,m} = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$. Tuy nhiên, có ba hạn chế mà phép lượng tử hoá tích gặp phải: Tương quan đặc trưng chưa được xử lý, lượng tử hoá gây mất mát thông tin ngữ nghĩa, độ chính xác của truy xuất giảm do sai số lượng tử hoá.

Thứ nhất, trong nhiều đặc trưng từ CNN, các chiều thường tương quan mạnh, thông tin về một yếu tố như ánh sáng, kiểu dáng, etc.. có thể trải trên nhiều chiều, nhưng việc chia vectơ đặc trưng x thành nhiều vectơ con x_m và lượng tử hoá độc lập chúng với từng từ mã C_m , điều này ngầm giả

định rằng các vectơ con là tương đối độc lập. Khi biểu diễn tương quan qua ma trận hiệp phương sai $\Sigma = \mathbb{E}[(x - \mu)(x - \mu)^T]$, nếu Σ có nhiều phần tử ngoài đường chéo lớn, nghĩa là các chiều phụ thuộc nhau, khi chia theo chỉ số, các không gian con thường không trùng với các trục chính của phân phối, tức là các thành phần có phương sai lớn và phụ thuộc lẫn nhau có thể nằm dàn trải qua nhiều không gian con. Hệ quả rằng các bộ mã độc lập cho từng không gian con không thể mô tả tốt các mối liên hệ chéo giữa các chiều, làm sai số lượng tử lớn hơn so với việc lượng tử hoá chung - tối ưu hoá toàn cục, bên cạnh đó sai số khôi phục tăng, làm giảm chất lượng truy hồi, đặc biệt với dữ liệu có cấu trúc ngữ nghĩa mạnh.

Thứ hai, mất mát thông tin khi chia các vectơ con, nếu có một thông tin ngữ nghĩa nào đó phụ thuộc vào sự phối hợp của nhiều chiều nằm ở các vectơ con khác nhau, thì khi cắt ra và lượng tử hoá độc lập chúng, ta sẽ phá vỡ mối quan hệ đó, gây mất mát thông tin ngữ nghĩa. Hệ quả của việc chia vectơ đặc trưng thành các vectơ con và lượng tử hoá chúng theo phương pháp lượng tử hoá tích, các khác biệt tinh tế giữa các mẫu ảnh sẽ bị làm mờ, và các bộ mã dù có tổ hợp các trung tâm mã trong bộ mã lớn, nhưng không thể biểu diễn chính xác quan hệ chéo.

Thứ ba, độ chính xác truy hồi giảm do sai số lượng tử hoá, trong phương pháp tìm kiếm láng giềng gần nhất, ta quan tâm đến xếp hạng khoảng cách giữa truy vấn q và các vectơ dữ liệu x . Với lượng tử hoá tích, ta đo khoảng cách $\|q - x\|^2 \approx \sum_{m=1}^M \|q_m - c_{m,k_m}\|^2$, nhưng nếu \hat{x} (x sau khi lượng tử hoá) khác xa x do sai số lượng tử hoá lớn, thì khoảng cách tính bị sai lệch, dẫn đến thứ tự xếp hạng truy xuất thay đổi.

Khử tương quan đặc trưng-xác suất

Để khắc phục các hạn chế nêu trên của phương pháp lượng tử hoá tích, bài báo OPQN đề xuất cơ chế khử tương quan đặc trưng - xác suất.

Thay vì ánh xạ trực tiếp vectơ con vào các từ mã bằng khoảng cách Ô-clid như trong lượng tử hoá tích, OPQN chèn thêm một tầng tuyến tính. Với mỗi vectơ con x_{im} , tầng này chiếu đặc trưng sang một không gian mới,

rồi tính điểm tương ứng cho từng từ mã. Sau đó, các điểm này đi qua hàm trung bình mũ để tạo thành phân phối xác suất gán cho các từ mã. Cốt lõi của cơ chế khử tương quan nằm ở chỗ đặc trưng không còn gắn trực tiếp với các vectơ con độc lập, mà được ánh xạ sang một không gian xác suất. Công thức được tính như sau:

$$p_{im,k} = \frac{\exp(x_{im}^T F_{m,k})}{\sum_{j=1}^K \exp(x_{im}^T F_{m,j})} \quad (2.1)$$

- **Đầu vào:** x_{im} vectơ đặc trưng con thứ m của ảnh i .
- **Tham số học được:** $F_{m,k}$ ma trận tham số của tầng tuyến tính.
- **Đầu ra:** $p_{im,k}$ xác suất vectơ thuộc về từ mã thứ k .

Nhờ lớp trung gian $F_{m,k}$, quá trình mã hoá không còn dựa duy nhất vào khoảng cách hình học như trong phân cụm truyền thống, mà trở thành **một phép học xác suất mềm** giữa đặc trưng và các từ mã, xác suất $p_{im,k}$ được điều chỉnh để tối ưu mục tiêu của toàn bộ mô hình. Thay vì gán vào một từ mã cứng nhắc, mô hình học ma trận tham số $F_{m,k}$ qua quá trình huấn luyện mạng nơ-ron. Cơ chế này cho phép mô hình linh hoạt hơn, tận dụng thông tin từ nhiều từ mã và được tối ưu cùng với toàn bộ mạng.

Thứ hai, OPQN sử dụng các từ mã trực chuẩn (chi tiết ở mục 3.1.4), làm giảm tương quan ngầm giữa các đặc trưng và bộ mã, giúp phân phối xác suất có ý nghĩa rõ ràng.

Xây dựng lượng tử hoá mềm

Sau khi xác định vectơ xác suất $p_{im,k}$ qua hàm softmax, lượng tử hoá mềm s_{im} được xây dựng bằng cách kết hợp tuyến tính các từ mã C_{mk} với các giá trị xác suất tương ứng. Công thức được biểu diễn như sau:

$$s_{im} = \sum_{k=1}^K p_{im,k} \cdot C_{mk} \quad (2.2)$$

Trong đó, C_{mk} là từ mã thứ k trong bộ mã C_m của không gian con thứ m , và $p_{im,k}$ là xác suất gán được tính từ x_{im} và tham số F_{mk} . Kết quả s_{im} là một tổ hợp lồi của các từ mã, với tổng các xác suất $p_{im,k}$ bằng một, và giá trị không âm. Điều này cho phép s_{im} tái hiện vectơ con x_{im} một cách mềm mại, giảm thiểu lỗi lượng tử hoá so với việc gán cứng vào một từ mã duy nhất.

Một đặc điểm quan trọng trong OPQN là việc tách biệt quá trình gán xác suất $p_{im,k}$ với các từ mã C_{mk} , giúp giảm sự phụ thuộc vào dữ liệu huấn luyện. Sự độc lập này cho phép $p_{im,k}$ được học một cách linh hoạt, trong khi các từ mã giữ vai trò cố định, góp phần tăng khả năng tổng quát hoá trên các tập dữ liệu mới, đặc biệt với các biến thể khuôn mặt chưa thấy. Ngoài ra, cách tiếp cận này tránh được những hạn chế của PQ, như hiện tượng bộ mã bị ảnh hưởng bởi độ chệch dữ liệu, đảm bảo độ ổn định và hiệu quả trong tìm kiếm quy mô lớn. Nhờ vậy, s_{im} không chỉ cải thiện độ chính xác mà còn hỗ trợ tính toán khoảng cách bất đối xứng trong giai đoạn tìm kiếm sau này.

Xây dựng lượng tử hoá cứng

Sau khi xây dựng lượng tử hoá mềm s_{im} , OPQN chuyển sang phương pháp lượng tử hoá cứng trong giai đoạn kiểm tra để tối ưu hoá hiệu suất tìm kiếm. Quá trình này được thực hiện bằng cách xác định chỉ số k^* của từ mã có xác suất cao nhất trong vectơ xác suất p_{im} . Công thức được biểu diễn như sau:

$$k^* = \arg_{k=1,2,\dots,K} \max p_{im,k} \quad (2.3)$$

Trong đó, giá trị k^* là chỉ số của từ mã trong bộ mã C_m , mà có khả năng tái hiện tốt nhất vector x_{im} , tức là giá trị có xác suất gán lớn nhất

trên phân phối xác suất p_{im} . Từ k^* , lượng tử hoá cứng h_{im} được định nghĩa đơn giản là:

$$h_{im} = C_{mk^*} \quad (2.4)$$

Ở đây, C_{mk^*} là từ mã tương ứng với từ mã thứ k^* trong bộ mã C_m . Phương pháp này chuyển đổi biểu diễn mềm x_{im} thành một vector rời rạc h_{im} , phù hợp cho việc tính toán nhanh khoảng cách trong hệ thống truy xuất thông tin với quy mô lớn. Khác với giai đoạn huấn luyện, nơi lượng tử hoá mềm được ưu tiên để tối ưu hoá độ dốc, lượng tử hoá cứng chỉ được áp dụng trong giai đoạn truy xuất, để giảm độ phức tạp tính toán và bộ nhớ, đặc biệt khi sử dụng bảng tra cứu để so sánh khoảng cách bất đối xứng.

Việc sử dụng k^* đảm bảo rằng mỗi vector con x_{im} được ánh xạ chính xác vào một từ mã đại diện, giúp duy trì tính phân biệt của đặc trưng trong khi tối ưu hoá hiệu suất. Điều này quan trọng trong ứng dụng truy xuất khuôn mặt, nơi độ chính xác của phép gán từ mã ảnh hưởng trực tiếp đến kết quả hàng xóm gần nhất. Phương pháp này tận dụng lợi thế của quá trình học trước đó ($p_{im,k}$ và s_{im}) để đạt được sự cân bằng giữa độ chính xác và tốc độ, đặt nền móng cho các bước tìm kiếm hiệu quả trong OPQN.

2.1.4 Sinh từ mã trực chuẩn

Vấn đề của bộ mã trong phương pháp lượng tử hoá truyền thống và hướng khắc phục

Trong các phương pháp lượng tử hoá tích truyền thống, bộ mã thường được học trực tiếp từ dữ liệu thông qua thuật toán như phân cụm K-Means. Cách tiếp cận này dẫn đến sự phụ thuộc mạnh mẽ của bộ mã vào phân bố của tập dữ liệu, có thể gây ra hiện tượng dư thừa thông tin giữa các từ mã hoặc lệch lạc từ mã khi dữ liệu không đại diện đầy đủ cho không gian đặc

trung. Giả định có một tập huấn luyện với độ chệch về một số loại hình ảnh, như tư thế hoặc ánh sáng nhất định, khả năng tổng quát hoá sẽ giảm khi áp dụng mô hình này trên dữ liệu mới hoặc danh tính chưa thấy, do bộ mã có thể bị học lệch.

Để khắc phục những hạn chế này, OPQN sử dụng các từ mã cố định và trực chuẩn, không phụ thuộc vào dữ liệu huấn luyện vì đã được thiết kế trước. Cách tiếp cận này đảm bảo tính ổn định và tính toán hiệu quả, đồng thời tăng cường chất lượng lượng tử hoá bằng cách tận dụng các đặc tính toán học cố định.

Bằng cách này, OPQN chuyển thách thức từ việc học các từ mã sang việc học xác suất (như đã trình bày ở phần 3.1.3), giúp mô hình tăng khả năng khái quát hoá và giảm rủi ro với dữ liệu có tính biến thiên cao.

Các từ mã trực chuẩn trong OPQN là các vector trong bộ mã C_m thỏa mãn hai điều kiện chính: Trực giao lẫn nhau và chuẩn hoá về độ dài bằng một. Cụ thể, đối với bất kỳ hai từ mã C_{mk} và C_{mj} ($k \neq j$ trong cùng một bộ mã $C_m \in \mathbb{R}^{d \times K}$, ta có:

$$C_{mk}^T C_{mj} = 0 \text{ (trực giao)}, \|C_{mk}\| = 1 \text{ (chuẩn hoá)}$$

Hai đặc tính của từ mã trực chuẩn mang lại các lợi ích quan trọng như sau:

Thứ nhất, các từ mã không dư thừa và phân bố đồng đều trong không gian \mathbb{R}^d , phép trực giao đảm bảo tính không chồng chéo thông tin, tránh các dư thừa thường gặp đối với các bộ mã học từ dữ liệu. Thêm vào đó, đặc tính này giúp bao phủ toàn bộ không gian vector con một cách hiệu quả, tăng khả năng biểu diễn đa dạng các đặc trưng khuôn mặt và những biến thể nội lớp.

Thứ hai, mỗi từ mã mang một hướng khác biệt rõ ràng, với góc giữa các từ mã là $\pi/2$, mỗi vector đại diện cho một hướng độc lập. Khi kết hợp với xác suất gán, mỗi từ mã đóng góp một thành phần thông tin riêng biệt, điều này làm cho việc lượng tử hoá trở nên chính xác hơn, giảm sai số khôi phục.

Thứ ba, đặc tính trực chuẩn hỗ trợ tốt hơn cho việc ánh xạ xác suất

$p_{im,k}$ và s_{im} (từ phần 3.1.3). Tính chất toán học cố định của bộ mã giúp tối ưu hoá độ dốc trong quá trình huấn luyện.

Tổng hợp lại, các đặc tính của bộ mã nâng cao tính thông tin và độ tin cậy của từ mã, góp phần vào hiệu suất tổng thể của OPQN trong nhiệm vụ truy xuất hình ảnh mặt người.

Cách sinh từ mã trực chuẩn

Trong OPQN, các từ mã trực chuẩn được sinh ra dựa trên phép biến đổi cosin rời rạc loại II (DCT-II). Phép biến đổi này vốn tạo ra một tập vector cơ sở trực chuẩn trong \mathbb{R}^d ; sau khi được chuẩn hoá bổ sung, các vector cơ sở vừa bảo toàn tính trực giao, vừa có độ dài chuẩn bằng một. Từ tập cơ sở này, K vector đầu tiên được chọn để hình thành bộ mã khởi tạo C_1 . Các bộ mã tiếp theo C_m được xây dựng tuần tự thông qua phép nhân với ma trận cơ sở đã chuẩn hoá, nhờ đó vừa duy trì tính trực chuẩn, vừa gia tăng tính đa dạng giữa các bộ mã. Quá trình bắt đầu từ việc xây dựng ma trận cơ sở trực chuẩn $A \in \mathbb{R}^{d \times d}$, với các phần tử được tính theo công thức:

$$A_{ij} = \cos\left[j\pi \cdot \frac{i + \frac{1}{2}}{d}\right], i = 0, \dots, d-1; j = 0, \dots, d-1 \quad (2.5)$$

Ma trận A này được chuẩn hoá để đảm bảo tính trực chuẩn. Nhằm tạo đa dạng cho các bộ mã C_m (với $m = 1, \dots, M$, OPQN áp dụng một quy trình lặp: bắt đầu từ $C_1 = A[:, 0 : K]$ (lấy K cột đầu tiên của A), sau đó nhân lặp với ma trận cơ sở khác để sinh các bộ mã tiếp theo, đảm bảo tính đa dạng nhưng vẫn giữ đặc điểm trực chuẩn.

Quy trình này được tóm tắt chi tiết trong thuật toán Algorithm 1. Cụ thể, với mỗi không gian con có kích thước $d = \frac{D}{M}$, ta khởi tạo một ma trận cơ sở cosin (theo công thức 3.5). Ma trận này sau đó được chuẩn hoá để thu được một cơ sở trực chuẩn A^\dagger , trong đó các cột đóng vai trò như những vector cơ sở trực giao.

Từ cơ sở này, từ mã đầu tiên C_1 được hình thành bằng cách chọn K

Algorithm 1 Sinh từ mã trực chuẩn được xác định

Require: Kích thước đặc trưng D , số lượng bộ mã M , số lượng từ mã mỗi bộ mã K , kích thước vector con d ($d = D/M$ và $d \geq K$)

Ensure: Các bộ mã $C \in \mathbb{R}^{M \times d \times K}$

- 1: Tính ma trận cơ sở cosine A theo phương trình (5)
 - 2: $A[1, 0] \leftarrow A[1, 0]/\sqrt{2}$
 - 3: $A^\dagger \leftarrow \sqrt{2}A/\sqrt{d}$
 - 4: $C_1 = A^\dagger[:, : K]$
 - 5: **for** $m = 2$ to $M + 1$ **do**
 - 6: $C_m = A^\dagger * C_{m-1}$
-

vector cơ sở đầu tiên trong A^\dagger . Các bộ mã tiếp theo được tạo ra bằng cách nhân lặp lại với A^\dagger , tức là: $C_m = A^\dagger C_{m-1}$, $m = 2, \dots, M$

Như vậy, sau quá trình này, ta thu được M bộ mã, mỗi bộ mã gồm K từ mã trực giao trong không gian con d -chiều. Thuật toán này đảm bảo tính nhất quán, tái lập và không phụ thuộc vào dữ liệu huấn luyện, với ràng buộc $K \leq d$ để tránh vượt quá chiều không gian. Cách tiếp cận này giúp OPQN duy trì hiệu suất ổn định trong các ứng dụng thực tế.

Việc sử dụng từ mã trực chuẩn cố định trong OPQN tạo nên sự tách biệt rõ ràng với quá trình gán xác suất (có thể học được), góp phần vào mục tiêu chính của bài báo là tăng cường khả năng tổng quát hoá và giảm lỗi lượng tử hoá.

Tổng thể, cách sinh từ mã trực chuẩn hỗ trợ OPQN đạt hiệu suất vượt trội trong truy xuất hình ảnh mặt người với tập dữ liệu có khả năng mở rộng và là nền tảng cho các bước tối ưu hoá tiếp theo

2.1.5 Hàm mất mát phân loại kết hợp theo từng không gian con

Mục tiêu

Trong bối cảnh của mô hình OPQN, sau khi đã xây dựng lượng tử hoá mềm s_{im} dựa trên khử tương quan đặc trưng - xác suất (Phần 3.1.3) và

sinh từ mã trực chuẩn cố định (Phần 3.1.4), chúng ta cần một hàm mất mát hiệu quả để duy trì và nâng cao tính phân biệt của các đặc trưng tại từng không gian con. Mục tiêu cốt lõi của hàm mất mát này là giảm biến thiên nội lớp - nghĩa là làm cho các đặc trưng thuộc cùng một danh tính tụ lại gần nhau hơn - đồng thời tăng cường khoảng cách ngoại lớp - đẩy các đặc trưng thuộc các danh tính khác nhau ra xa hơn. Phân tích sâu hơn, việc áp dụng hàm mất mát này tại từng không gian con riêng lẻ thay vì toàn bộ đặc trưng toàn cục giúp khai thác tối đa cấu trúc dữ liệu phân mảnh, đặc biệt trong ngữ cảnh lượng tử hoá sản phẩm, nơi mỗi không gian con m xử lý một phần độc lập của đặc trưng nút cổ chai x_i .

Có thể hiểu rằng, trong nhiệm vụ truy xuất ảnh mặt người quy mô lớn, đặc trưng cần phải giữ được tính phân biệt mạnh mẽ ngay cả sau khi bị lượng tử hoá, vì lỗi lượng tử hoá có thể làm mờ ranh giới giữa các lớp. Bằng cách buộc cả đặc trưng gốc x_{im} và lượng tử hoá mềm s_{im} phải tuân theo cùng một tiêu chí phân lớp, hàm mất mát này khuyến khích sự tương thích giữa hai biểu diễn trên, giúp giảm lỗi tái tạo và tăng khả năng tổng quát hoá trên các danh tính chưa thấy. Điều này đặc biệt quan trọng trong OPQN, nơi mục tiêu là cân bằng giữa tính chính xác và hiệu quả lưu trữ, tìm kiếm, vì hàm mất mát này giúp tối ưu hoá đồng thời mạng xương sống và quá trình lượng tử hoá mà không làm suy giảm hiệu suất nhận dạng.

Chuẩn hoá

Trước khi tính hàm mất mát, chúng ta cần chuẩn hoá ℓ_2 cho tất cả các vector liên quan, nhằm loại bỏ biến thiên về độ dài và tập trung vào thông tin góc độ, vốn phù hợp cho các nhiệm vụ nhận dạng khuôn mặt.

Cụ thể, $x_{im} \leftarrow x_{im}/\|x_{im}\|_2$, $s_{im} \leftarrow s_{im}/\|s_{im}\|_2$, và trọng số phân lớp $W_{m,c} \leftarrow W_{m,c}/\|W_{m,c}\|_2$ với ($W_m \in \mathbb{R}^{d \times C}$ là ma trận trọng số cho bộ phân lớp tới không gian con m , mỗi $W_{m,c}$ đại diện cho hướng lý tưởng của đặc trưng thuộc lớp c (danh tính thứ $c \in C$) trong không gian vectơ con \mathbb{R}^d).

Phân tích kỹ hơn, chuẩn hoá này chuyển đổi không gian O-clid thành không gian cầu, nơi độ tương tự đo bằng độ tương đồng co-sine qua tích

vô hướng: $\cos \theta = v_1^T v_2$. Việc chuẩn hoá đồng bộ cả $x_{im}, s_{im}, W_{m,c}$ đảm bảo rằng các phép tính cosine trong hàm mất mát đều được thực hiện trên cùng một thang đo, tăng tính ổn định và hội tụ trong huấn luyện. Đồng thời việc chuẩn hoá liên tục giúp tránh hiện tượng bùng nổ đạo hàm trong không gian cao chiều.

Hàm mất mát cho các đặc trưng gốc L_x

Để khuyến khích tính phân biệt và cô đọng cho đặc trưng gốc x_{im} , OPQN áp dụng hàm mất mát trung bình mũ biên độ góc, một biến thể phổ biến trong các mô hình nhận dạng khuôn mặt để cải thiện sự tách biệt lớp. Công thức cho L_x được biểu diễn như sau:

$$L_x = \sum_{i=1}^N \sum_{m=1}^M -\log \frac{e^{r(\cos \theta_{y_i, x_{im}} - u)}}{e^{r(\cos \theta_{y_i, x_{im}} - u)} + \sum_{j \neq y_i} e^{r \cos \theta_{j, x_{im}}}} \quad (2.6)$$

Trong đó, $\cos \theta_{y_i, x_{im}} = x_{im}^T W_{m, y_i}$ sau chuẩn hoá, với $W_{m, y_i} \in \mathbb{R}^d$ là vector trọng số tương ứng với lớp y_i trong ma trận trọng số phân lớp $W_m \in \mathbb{R}^{d \times C}$ cho không gian con m . Giá trị $\cos \theta_{y_i, x_{im}}$ đại diện cho độ tương tự cosine giữa vectơ con x_{im} và vectơ trọng số của lớp đúng y_i , phản ánh mức độ gần gũi góc độ giữa đặc trưng và trọng số sau khi cả hai được chuẩn hoá ℓ_2 . Ma trận W_m chứa C vector trọng số $W_{m,c}$, $c \in 1, \dots, C$, với W_{m, y_i} là cột tương ứng với nhãn y_i của mẫu i .

Trong công thức, $r > 0$ là hệ số để khuếch đại đạo hàm và ổn định phân phối hàm trung bình mũ, trong khi $u > 0$ là biên độ co-sine, một hằng số được thêm vào để phạt các trường hợp co-sine gần 1 nhưng không đủ cô đặc, tức là khi các mẫu thuộc về lớp đó nhưng khoảng cách giữa mẫu và trọng tâm lớp đó vẫn chưa đủ nhỏ theo yêu cầu của biên độ u . Trong triển khai thực tế, các siêu tham số u thường từ khoảng 0.1-0.5, và hệ số khuếch đại r thường từ khoảng 30-64, được chọn dựa trên kinh nghiệm từ các mô hình nhận dạng khuôn mặt.

Tổng kép $\sum_{i=1}^N \sum_{m=1}^M$ chạy qua tất cả N mẫu và M không gian con,

đảm bảo hàm mất mát được tính theo từng không gian con, mỗi không gian con x_m tập trung học một khía cạnh phân biệt khác nhau của khuôn mặt, giúp chuyên môn hoá đặc trưng. Nếu một không gian con bị nhiễu, ví dụ như do che khuất một phần khuôn mặt, các không gian con còn lại vẫn có thể cung cấp thông tin phân biệt chính xác. Điều này làm tăng độ bền vững của mô hình trước yếu tố gây nhiễu trong thế giới thực

Hàm log-softmax với biên độ u làm tăng độ khó trong việc đạt xác suất cao cho lớp đúng, buộc mô hình phải học từ các đặc trưng phân biệt mạnh mẽ hơn, tăng độ cô đọng và tách biệt, vì chúng ép các điểm dữ liệu cùng lớp tụ lại trong một góc nhỏ và tách biệt các lớp khác.

Hàm mất mát cho các đặc trưng đã được lượng tử hoá L_s

Tương tự với L_x , hàm mất mát trung bình mũ biên độ góc được áp dụng lên đặc trưng lượng tử hoá mềm s_{im} (đã được chuẩn hoá ℓ_2) để đảm bảo chúng duy trì cả tính phân biệt và tính cô đọng tương đương đặc trưng gốc x_{im} . Mục tiêu không chỉ giữ cho s_{im} phản ánh chính xác các đặc trưng nhận dạng của x_{im} , mà còn ép các điểm dữ liệu thuộc cùng một lớp tụ lại gần trọng tâm lớp trong không gian góc độ, từ đó giảm khoảng cách giữa hai biểu diễn và cải thiện hiệu quả lượng tử hoá. Công thức cho L_s được biểu diễn như sau:

$$L_s = \sum_{i=1}^N \sum_{m=1}^M -\log \frac{e^{r(\cos \theta_{y_i, s_{im}} - u)}}{e^{r(\cos \theta_{y_i, s_{im}} - u)} + \sum_{j \neq y_i} e^{r \cos \theta_{j, s_{im}}}} \quad (2.7)$$

Trong đó, $\cos \theta_{y_i, s_{im}} = s_{im}^T W_{m, y_i}$, với $W_{m, y_i} \in \mathbb{R}^d$ là vector trọng số tương ứng với lớp y_i trong ma trận trọng số phân lớp $W_m \in \mathbb{R}^{d \times C}$ cho không gian con m . Sau chuẩn hoá, giá trị $\cos \theta_{y_i, s_{im}}$ thể hiện mức độ tương tự co-sine giữa s_{im} và trọng số lớp đúng, phản ánh mức độ gần gũi góc độ giữa đặc trưng lượng tử hoá mềm và lý tưởng hoá lớp. Các tham số r và u được giữ nguyên như trong L_x để đảm bảo sự tương thích giữa x_{im} và s_{im} giúp giảm thiểu sai lệch do quá trình lượng tử hoá.

Phân tích sâu hơn, L_s đóng vai trò quan trọng trong việc khắc phục nhược điểm tiềm ẩn của lượng tử hoá, nơi thông tin phân biệt có thể bị mất do việc nén đặc trưng. Không giống như L_x tập trung vào đặc trưng gốc, L_s đặc biệt chú trọng đến việc điều chỉnh s_{im} sao cho nó không chỉ tách biệt các lớp mà còn duy trì sự cô đọng nội lớp, ngay cả khi s_{im} là tổ hợp lồi của các từ mã trực chuẩn. Biên độ u trong L_s buộc mô hình phải học cách tối ưu hoá xác suất $p_{im,k}$ (được tính từ hàm trung bình mũ của x_{im} và F_{mk}) sao cho s_{im} gần với trọng tâm lớp hơn, giảm biến thể nội lớp trong không gian lượng tử hoá. Điều này đặc biệt quan trọng trong OPQN, vì lượng tử hoá mềm cần giữ được tính chất nhận dạng của đặc trưng gốc để hỗ trợ hiệu quả trong việc tìm kiếm hàng xóm gần nhất, nơi sự cô đọng và tách biệt là yếu tố quyết định.

Lý do sử dụng L_s nằm ở việc nó mở rộng tính năng của hàm trung bình mũ biên độ góc từ đặc trưng gốc sang đặc trưng lượng tử hoá, đảm bảo quá trình nén không làm suy giảm khả năng phân lớp. Trong ngữ cảnh OPQN, việc duy trì cả tính phân biệt và cô đọng cho s_{im} giúp giảm sai số lượng tử hoá, đặc biệt khi dữ liệu khuôn mặt có biến thiên lớn. So với L_x , L_s không chỉ củng cố tính chất nhận dạng mà còn hỗ trợ tối ưu hoá ma trận biến đổi F (được sử dụng để tính $p_{im,k}$, tạo sự liên kết chặt chẽ giữa quá trình lượng tử hoá và phân lớp. Điều này đặt nền móng cho việc chuyển sang lượng tử hoá cứng trong giai đoạn kiểm tra, đồng thời chuẩn bị cho các bước tiếp theo như tối thiểu hoá độ cô đặc.

Hàm mất mát liên hợp

Dựa trên các hàm mất mát riêng lẻ đã được xây dựng, OPQN tích hợp chúng thành một hàm mất mát liên hợp để tối ưu hoá toàn diện mô hình. Hàm mất mát phân lớp liên hợp theo không gian con, được ký hiệu là L_{clf} , được xây dựng bằng cách tích hợp mất mát cho đặc trưng gốc L_x và mất mát cho đặc trưng lượng tử hoá mềm L_s theo công thức:

$$L_{clf} = \frac{1}{2MN}(L_x + L_s) \quad (2.8)$$

Với M là số không gian con, N là số mẫu huấn luyện, hệ số chuẩn hoá $\frac{1}{2MN}$ đóng vai trò quan trọng trong việc bình quân hoá mất mát trên toàn bộ M không gian con và N mẫu, ngăn chặn hiện tượng thiên lệch do chênh lệch kích thước dữ liệu hoặc số lượng không gian con. Yếu tố $\frac{1}{2}$ đảm bảo sự cân bằng giữa đóng góp của L_x và L_s , thúc đẩy sự hài hoà trong quá trình tối ưu hoá đặc trưng gốc và đặc trưng lượng tử hoá, tránh việc ưu tiên quá mức một trong hai phần. Mục tiêu của hàm mất mát nhằm tối thiểu hoá đồng thời các tham số mạng xương sống Θ , ma trận biến đổi F và ma trận trọng số phân lớp W

Về mặt trực quan, L_{clf} có thể được hình dung như một quá trình chiếu các đặc trưng lên không gian cầu đơn vị, nơi biên độ u hoạt động như một lực kéo các điểm thuộc cùng các lớp lại gần nhau, đồng thời đẩy các điểm thuộc các lớp khác ra xa, tạo ra một cấu trúc không gian rõ ràng và phân tách tốt. Sự kết hợp của L_x và L_s trong L_{clf} đảm bảo rằng đặc trưng lượng tử hoá mềm s_{im} không chỉ theo sát đặc trưng gốc x_{im} về mặt nhận dạng, mà còn hỗ trợ tối ưu hoá toàn diện, giảm thiểu tác động tiêu cực từ quá trình nén dữ liệu.

Việc chỉ sử dụng L_x có thể bỏ qua việc điều chỉnh s_{im} , trong khi chỉ dựa vào L_s có thể làm suy yếu khả năng học của mạng xương sống. Hàm mất mát liên hợp giải quyết vấn đề này bằng phân bổ đạo hàm một cách cân bằng qua quá trình lan truyền ngược, giúp điều chỉnh đồng thời Θ , F và W , từ đó giảm thiểu sai số lượng tử hoá một cách hiệu quả.

Trong quá trình xử lý OPQN, L_{clf} đóng vai trò quan trọng như một cầu nối giữa các giai đoạn. Chúng chuẩn bị các vector nhúng (vector đặc trưng được mô hình học có chất lượng cao), để cho phép tối thiểu hoá (Phần 3.1.6) dễ dàng ép phân phối xác suất p_{mk} về one-hot, rút ngắn khoảng cách giữa lượng tử hoá mềm và cứng. Đồng thời, chúng đặt nền móng cho truy xuất khoảng cách bất đối xứng (phần 3.1.8) bằng cách đảm bảo rằng

các đặc trưng lượng tử hoá cứng h_{im} (sau khi chuyển từ s_{im}) vẫn giữ được tính phân biệt cần thiết, tối ưu hoá cả tốc độ và độ chính xác trong quá trình tìm kiếm quy mô lớn. Như vậy, L_{clf} không chỉ là một công cụ tối ưu hoá mà còn là yếu tố then chốt trong việc duy trì hiệu quả tổng thể của mô hình trên các kịch bản thực tế đa dạng.

2.1.6 Tối thiểu hoá độ cô đặc(entropy) cho việc gán mã one-hot

Trong mô hình OPQN, mặc dù hàm mất mát phân lớp liên hợp L_{clf} phần (3.1.5) đã đảm bảo tính phân biệt cho cả đặc trưng gốc x_{im} và lượng tử hoá mềm s_{im} , vẫn tồn tại khoảng cách đáng kể giữa s_{im} - biểu diễn như một tổ hợp lồi của các từ mã theo phân phối xác suất p_{im} - và lượng tử hoá cứng h_{im} - dạng one-hot chỉ gán vào một từ mã duy nhất. Vấn đề phát sinh từ việc huấn luyện chỉ dựa vào L_{clf} , vốn không trực tiếp khuyến khích p_{im} hội tụ về phân phối one-hot, dẫn đến sai lệch giữa s_{im} và h_{im} trong giai đoạn truy xuất. Sai lệch này có thể làm tăng lỗi lượng tử hoá, đặc biệt trong cách kịch bản tìm kiếm khuôn mặt quy mô lớn, nơi h_{im} được sử dụng để nén dữ liệu và tính khoảng cách hiệu quả.

Mục tiêu của phần tối thiểu hoá độ bất định, độ hỗn loạn là giới thiệu một bộ chuẩn hoá dựa trên độ bất định để giảm thiểu khoảng cách này, bằng cách ép phân phối p_{im} tiến gần hơn đến dạng one-hot. Việc này không chỉ cải thiện sự tương thích giữa lượng tử hoá mềm và lượng tử hoá cứng, mà còn giảm lỗi tái hiện tổng thể, tăng cường khả năng tổng quát hoá của mô hình trên dữ liệu chưa thấy. Theo phân tích trong OPQN(Phần 3.4), phép chuẩn hoá độ bất định đóng vai trò quan trọng trong việc cân bằng giữa tính linh hoạt của lượng tử hoá mềm trong huấn luyện và tính hiệu quả của lượng tử hoá cứng trong suy diễn.

Định nghĩa hàm mất mát độ bất định

Hàm mất mát độ bất định được sử dụng như một phương pháp chuẩn hoá để đo lường độ không chắc chắn trong phân phối p_{im} và thúc đẩy sự hội tụ về one-hot. Công thức hàm mất mát độ bất định L_{ent} được định nghĩa như sau:

$$L_{ent} = -\frac{1}{MN} \sum_{i=1}^N \sum_{m=1}^M \sum_{k=1}^K p_{im,k} \log p_{im,k} \quad (2.9)$$

Độ bất định của một phân phối đạt giá trị tối thiểu bằng 0 khi và chỉ khi phân phối đó là one-hot, tương ứng với gán rõ ràng vào một từ mã. Trong công thức, tổng nội $\sum_{k=1}^K p_{im,k} \log p_{im,k}$ tính độ bất định cho mỗi p_{im} tại không gian con m và mẫu i , với dấu âm để chuyển thành hàm mất mát tối thiểu hoá. Hệ số chuẩn hoá $-\frac{1}{MN}$ bình quân hoá trên tất cả không gian con và mẫu, đảm bảo hàm mất mát không bị thiên lệch bởi kích thước dữ liệu. Việc tối thiểu hoá L_{ent} thúc đẩy p_{im} trở nên sắc nét hơn, giảm độ phân tán và tăng cường tính quyết đoán trong gán, từ đó giảm sai số lượng tử hoá mà không ảnh hưởng đến đạo hàm từ phân lớp.

Tích hợp với hàm mất mát phân lớp

Để cân bằng giữa mục tiêu phân lớp và phương pháp chuẩn hoá độ bất định, OPQN tích hợp hàm mất mát bất định vào hàm mất mát tổng thể như sau:

$$L = L_{clf} + \lambda L_{ent} \quad (2.10)$$

Với λ là trọng số siêu tham số kiểm soát cường độ của hàm chuẩn hoá độ bất định. L_{clf} ưu tiên tính phân biệt, trong khi L_{ent} tập trung vào sự sắc nét của p_{im} ; λ đóng vai trò điều tiết, tránh hiện tượng quá chuẩn hoá (khi λ lớn, có thể làm mất đạo hàm mềm từ L_{clf}), hoặc thiếu chuẩn hoá (khi λ nhỏ, không đủ ép one-hot). Theo bài báo, $\lambda = 0.1$ được chọn qua

kiểm định chéo để đạt cân bằng tối ưu, đảm bảo hàm mất mát tổng hợp không chỉ giảm lỗi phân lớp mà còn cải thiện sự tương thích giữa lượng tử hoá cứng và lượng tử hoá mềm. Việc tích hợp này tạo ra một khung hàm mất mát lai, nơi hàm chuẩn hoá độ bất định bổ trợ cho L_{clf} bằng cách tăng cường tính ổn định trong huấn luyện, đặc biệt với mã ngắn nơi sai số lượng tử hoá dễ tăng.

Các đạo hàm quan trọng để minh hoạ cách lan truyền ngược hoạt động

Để minh hoạ cách hàm mất mát độ bất định lan truyền ảnh hưởng qua quá trình lan truyền ngược, cần xem xét các đạo hàm chính liên quan đến logit $g_{mk} = x_{im}^T F_{m,k}$, nơi $p_{im} = \text{softmax}(g_m)$. Đạo hàm của $p_{im,k}$ theo $g_{m,k}$ được tính như sau:

$$\frac{\partial p_{im,k}}{\partial g_{m,k}} = p_{im,k}(1 - p_{im,k}), \quad \frac{\partial p_{im,k}}{\partial g_{m,j}} (j \neq k) = -p_{im,k}p_{im,j}$$

Tiếp theo, đạo hàm của s_{im} theo $g_{m,k}$:

$$\frac{\partial s_{im}}{\partial g_{m,k}} = p_{im,k}(C_{m,k} - s_{im})$$

Cuối cùng, đạo hàm của L_{ent} theo $g_{m,k}$:

$$\frac{\partial L_{ent}}{\partial g_{m,k}} = p_{im,k}(\sum_{j=1}^K p_{im,j} \log p_{im,j} - \log p_{im,k})$$

Các đạo hàm này minh hoạ cách hàm mất mát bất định điều chỉnh đạo hàm để ép p_{im} sắc nét hơn, ảnh hưởng đến việc cập nhật F_m qua $g_{m,k}$. Trong quá trình lan truyền ngược, đạo hàm từ L_{ent} kết hợp với đạo hàm từ L_{clf} để cân bằng giữa phân lớp và chuẩn hoá, giảm khoảng cách lượng tử hoá mềm và lượng tử hoá cứng mà không làm gián đoạn quá trình học.

Về mặt trực quan, hàm mất mát bất định hoạt động như một lực kéo phân phối p_{im} từ trạng thái phân tán (độ bất định cao, các thành phần gần đều nhau) sang trạng thái có đỉnh duy nhất (độ bất định thấp, gần one-hot), khiến s_{im} dịch chuyển dần về một từ mã cụ thể - gần với h_{im} .

2.1.7 Quá trình học và tối ưu hoá

Quy trình học và tối ưu hoá trong OPQN nhằm cập nhật các tham số mô hình một cách đều cuối, đảm bảo hội tụ ổn định và nâng cao chất lượng đặc trưng lượng tử hoá mà không phụ thuộc vào việc học các từ mã. Mục tiêu chính là tối ưu hoá đồng thời tham số mạng xương sống Θ , ma trận biến đổi tuyến tính F và trọng số phân lớp W thông qua SGD và lan truyền ngược, tận dụng đạo hàm từ cả hàm mất mát phân lớp L_{clf} (tăng tính phân biệt) và hàm mất mát entropy (ép gán one-hot), giảm lỗi tái hiện tổng thể.

Tham số học

Các tham số cần học trong OPQN bao gồm, Θ (Tập hợp tham số của mạng xương sống để trích xuất đặc trưng nút cổ chai x_i , $F = [F_1, \dots, F_M]$ (Ma trận biến đổi tuyến tính để tính logit và xác suất gán, với mỗi $F_m \in \mathbb{R}^{d \times K}$, và $W = [W_1, \dots, W_M]$ (Ma trận trọng số phân lớp, với mỗi $W_m \in \mathbb{R}^{d \times C}$.

Θ chịu trách nhiệm tạo đặc trưng cao cấp từ hình ảnh đầu vào, F đóng vai trò cầu nối để biến đổi vector con thành logit cho gán các từ mã và W đảm bảo phân lớp chính xác tại từng không gian con. Các tham số này được học đồng thời để đảm bảo tính nhất quán giữa trích xuất đặc trưng và phân lớp, với F giúp tối ưu hoá gán xác suất dựa trên đặc trưng từ Θ , tăng cường hiệu quả trong không gian cao chiều.

Quy trình tối ưu hoá

Quy trình tối ưu hoá sử dụng phép giảm đạo hàm ngẫu nhiên theo lô nhỏ (Mini Batch SGD) để tối thiểu hoá hàm mất mát tổng thể $L = L_{clf} + \lambda L_{ent}$, với lan truyền ngược là quá trình lan truyền đạo hàm qua các lớp mạng. Trong mỗi lần lặp, mẫu lô nhỏ được lấy ngẫu nhiên từ tập huấn luyện, lan truyền dọc để tính đặc trưng thất nút cổ chai $x_i = f(\Theta, I_i)$, chia thành các

vector con x_{im} , tính logit g_m , xác suất p_{im} , lượng tử hoá mềm s_{im} , và cuối cùng là hàm mất mát phân lớp cho cả x_{im} và s_{im} ; sau đó lan truyền ngược đạo hàm để cập nhật W , F và Θ .

Phân tích kỹ, việc truyền thẳng bắt đầu từ mạng xương sống để tạo x_i , sau đó biến đổi tuyến tính qua F để có logit, đảm bảo lượng tử hoá mềm phản ánh đặc trưng gốc. Lan truyền ngược, lan truyền đạo hàm tổng hợp từ L , với L_{clf} điều chỉnh W và Θ cho phân lớp, L_{ent} (qua λ) hỗ trợ F để tinh chỉnh giá trị xác suất. Lý do sử dụng phép giảm đạo hàm ngẫu nhiên theo lô nhỏ nằm ở hiệu quả với dữ liệu lớn, giảm nhiễu đạo hàm, tránh các điểm cực tiểu tương đối, tăng tốc độ hội tụ trong mô hình sâu.

Quá trình lan truyền ngược

Để minh hoạ quá trình lan truyền ngược, cần tính đạo hàm của hàm mất mát tổng thể L theo các tham số, bắt đầu từ logit g_{mk} , dựa trên đạo hàm của L_{clf} (vì L_{ent} đã được phân tích ở 3.1.6).

Đạo hàm của L theo F_{mk} (Eq:14 trong bài báo)

$$\frac{\partial L}{\partial F_{mk}} = \left[\frac{1}{2} \left(\frac{\partial L_x}{\partial x_{im}} + \frac{\partial L_s}{\partial s_{im}} \frac{\partial s_{im}}{\partial g_{mk}} \right)^T \right] x_{im} + \lambda \frac{\partial L_{ent}}{\partial F_{mk}}$$

Đạo hàm của L theo W_{mk} (Eq. 15)

$$\frac{\partial L}{\partial W_{mk}} = \frac{1}{2} \left(\frac{\partial L_x}{\partial W_{mk}} + \frac{\partial L_s}{\partial W_{mk}} \right)$$

Đạo hàm của L theo x_{im} (Eq. 16)

$$\frac{\partial L}{\partial x_{im}} = \frac{1}{2} \frac{\partial L_x}{\partial x_{im}} + \frac{1}{2} \left(\frac{\partial L_s}{\partial s_{im}} \frac{\partial s_{im}}{\partial g_{mk}} \right) F_{mk} + \lambda \frac{\partial L_{ent}}{\partial x_{im}}.$$

Đạo hàm theo F_{mk} và x_{im} bao gồm cả đóng góp từ L_{ent} (qua λ), điều chỉnh giá trị xác suất và đặc trưng để ép one-hot, trong khi đạo hàm theo W_{mk} chủ yếu từ L_{clf} để tối ưu hoá phân lớp.

Quá trình huấn luyện OPQN

Thuật toán 2 tóm tắt quy trình huấn luyện OPQN như sau:

Algorithm 2 Quy trình huấn luyện OPQN

Require: Tập huấn luyện $\{I_i\}_{i=1}^N$ với nhãn y , mạng $f(\cdot)$, kích thước bộ từ mã $M \times d \times K$

Ensure: Các tham số đã huấn luyện Θ, F, W

Sinh các từ mã trực chuẩn bằng Thuật toán 1

- 2: Khởi tạo các tham số mạng nền Θ , lớp biến đổi tuyến tính F , ma trận trọng số phân loại W

repeat

- 4: Lấy ngẫu nhiên một mini-batch từ tập huấn luyện
Tính $x_i = f(\Theta; I_i)$ cho từng ảnh trong mini-batch
- 6: Tính hàm mục tiêu L theo phương trình (10)
Tính đạo hàm của L theo W , F , và x_{im} theo các phương trình (15), (14) và (16)
- 8: Lan truyền ngược gradient và cập nhật các tham số W , F , Θ

until hội tụ

Các bước chi tiết như sau: Đầu tiên, mô hình sinh từ mã trực chuẩn bằng Algorithm 1, sử dụng Phép biến đổi cosine rời rạc để tạo từ mã cố định. Sau đó, vòng lặp sử dụng SGD để cập nhật tham số dần dần, với điều kiện dừng khi hàm mất mát ổn định hoặc đạt số lần lặp tối đa. Tiếp theo, lấy tập con nhỏ ngẫu nhiên từ tập huấn luyện để giảm chi phí tính toán, truyền thẳng tập con nhỏ ấy qua mô hình để tính $x_i = f(\Theta; I_i)$ cho mỗi hình ảnh. Sau đó, tính hàm mất mát tổng thể L từ hàm mất mát phân lớp và hàm mất mát entropy dựa trên x_{im}, s_{im}, p_{im} . Tiếp theo, đạo hàm L được tính theo W , F , và x_{im} . Các đạo hàm này được lan truyền ngược qua mạng, từ phân lớp và lượng tử hoá mềm về mạng xương sống để cập nhật các tham số bằng SGD và tốc độ học.

2.1.8 So sánh khoảng cách bất đối xứng trong truy xuất

Trong giai đoạn truy vấn của hệ thống truy xuất hình ảnh khuôn mặt quy mô lớn, việc lựa chọn phép đo lường tương đồng đóng vai trò quan trọng trong việc cân bằng giữa độ chính xác và hiệu suất tính toán. Theo

các nghiên cứu gần đây, chúng tôi áp dụng khoảng cách lượng tử hoá bất đối xứng (AQD) làm phép đo lường chính, cho phép sử dụng lượng tử hoá mềm để biểu diễn truy vấn trong khi mã hoá cơ sở dữ liệu bằng lượng tử hoá cứng. Cách tiếp cận này không chỉ giảm đáng kể lượng tài nguyên của bộ nhớ, mà còn tăng tốc độ truy vấn, phù hợp với kiến trúc gọn nhẹ được đề xuất, nơi tài nguyên tính toán bị hạn chế ở quy mô lớn.

Quy trình xử lý giữa các truy vấn và các mục trong cơ sở dữ liệu được thiết kế khác biệt để tận dụng ưu điểm của lượng tử hoá. Đối với một truy vấn q , chúng tôi lan truyền nó qua mô hình cho đến lớp biến đổi tuyến tính, sau đó áp dụng hàm trung bình mũ, để thu được vector xác suất p_{qm} cho từng vector con x_{qm} . Kết hợp với bộ mã C_m , ta nhận được lượng tử hoá mềm s_{qm} . Ngược lại, đối với mỗi hình ảnh từ cơ sở dữ liệu I_i , chúng tôi tiến hành tính toán p_{im} theo quy trình tương tự truy vấn, và liên kết nó với lượng tử hoá cứng h_{im} thông qua chỉ số của giá trị lớn nhất trong p_{im} . Ma trận $B \in \mathbb{R}^{N \times M}$ được xây dựng trước, trong đó mỗi phần tử b_{im} lưu chỉ số k^* của từ mã theo công thức tính trong lượng tử hoá cứng, giúp mã hoá hiệu quả các đặc trưng gốc thành mã nhị phân.

Khoảng cách AQD giữ truy vấn q và I_i được tính bằng tổng bình phương Ô-clid giữa lượng tử hoá mềm của truy vấn và lượng tử hoá cứng của cơ sở dữ liệu, cụ thể.

$$AQD(q, I_i) = \sum_{m=1}^M \|s_{qm} - h_{im}\|_2^2 = \sum_{m=1}^M \|C_m p_{qm} - C_m b_{im}\|_2^2 \quad (2.11)$$

Công thức này đảm bảo việc so sánh tương đồng được thực hiện chính xác trong không gian lượng tử hoá, đồng thời giữ nguyên thông tin phân biệt giữa các đặc trưng khuôn mặt dưới các biến đổi nội lớp. Nhờ tính trực chuẩn của từ mã C_m công thức trên có thể được đơn giản hoá đáng kể. Bằng cách mở rộng về phải và loại bỏ các thành phần hằng số hoặc không phụ thuộc vào $C_m b_{im}$, ta thu được:

$$\arg \min_i \text{AQD}(q, I_i) = \arg \min_i \sum_{m=1}^M -2p_{qm}^T C_m^T C_m b_{im} = \arg \max_i \sum_{m=1}^M p_{qm, b_{im}} \quad (2.12)$$

Kết quả này cho thấy việc tối ưu hoá AQD chỉ phụ thuộc vào $\{p_{qm}\}_{m=1}^M$ và $\{b_{im}\}_{m=1}^M$, cho phép thực hiện so sánh tương đồng lượng tử hoá một cách hiệu quả bằng cách sử dụng các bảng tra cứu. Cụ thể, chúng tôi xây dựng M bảng tra cứu $\{LUT_m\}$ tương ứng với M vector xác suất $\{p_q\}_{m=1}^M$ trong đó, $LUT_m[i] = b_{im}$. Vì ma trận B được tiền tính toán, quy trình tìm kiếm chỉ yêu cầu một số phép cộng cơ bản, giảm thiểu độ phức tạp thời gian so với các phương pháp truyền thống.

Để triển khai quy trình truy xuất top- k hiệu quả, chúng tôi đề xuất thuật toán 3:

Algorithm 3 Quy trình truy hồi Top- k OPQN

Require: Tập ảnh cơ sở dữ liệu $DB = \{db_i\}_{i=1}^{|DB|}$, tập ảnh truy vấn $Q = \{q_j\}_{j=1}^{|Q|}$, mô hình đã huấn luyện;

Ensure: Top k ảnh trong DB cho mỗi q_i ;

Truyền tiến DB qua mô hình trước, và tính trước ma trận chỉ số B theo (2) và (4);

Xây dựng LUTs cho DB dựa trên ma trận B ;

3: **for** $i = 1, 2, \dots, |Q|$ **do**

Truyền tiến q_i qua mô hình và tính p_{qm} theo (2);

Tính độ tương đồng giữa q_i và từng ảnh trong cơ sở dữ liệu theo (18) sử dụng LUTs, và sắp xếp kết quả theo thứ tự giảm dần;

Đầu tiên, toàn bộ tập ảnh trong cơ sở dữ liệu $DB = \{db_i\}_{i=1}^{|DB|}$, trong đó db_i là ảnh thứ i và $|DB|$ là tổng số ảnh trong cơ sở dữ liệu, được đưa qua mô hình đã huấn luyện để tiền tính toán ma trận chỉ số B theo công thức. Việc này tương ứng với quá trình chia đặc trưng của mỗi ảnh thành các vector con và gán cho mỗi vector con một mã của từ mã, nhằm đảm bảo cơ sở dữ liệu được mã hoá trước khi tiến hành truy vấn.

Tiếp theo, các bảng tra cứu nhanh được xây dựng dựa trên ma trận

B , cho phép lưu trữ trước một phần kết quả tính toán độ tương đồng, từ đó tăng tốc độ xử lý khi truy vấn. Đối với mỗi ảnh truy vấn q_i trong tập $Q = \{q_i\}_{i=1}^{|Q|}$, trong đó q_i là ảnh truy vấn thứ i và $|Q|$ là tổng số truy vấn, hệ thống thực hiện lan truyền thẳng qua mô hình để tính vector xác suất $p_{q,m}$.

Sau đó, độ tương đồng giữa q_i và từng ảnh trong DB được tính toán theo công thức (argmax) bằng cách sử dụng các LUTs, rồi sắp xếp kết quả theo thứ tự giảm dần. Cuối cùng, k ảnh có độ tương đồng cao nhất được chọn làm kết quả cho q_i .

Quy trình này được lặp lại cho toàn bộ tập truy vấn Q , đảm bảo khả năng truy xuất đồng nhất và hiệu quả.

Một lợi thế quan trọng của phương pháp này là khả năng tổng quát hoá cho các danh tính chưa thấy. Các phương pháp bám truyền thông thường chỉ hoạt động tốt với các danh tính đã thấy, thiếu khả năng tổng quát hoá vì chỉ học ánh xạ dựa trên các khuôn mặt đã có, không tạo được mã bám khi gặp khuôn mặt mới, khiến cho việc so sánh độ tương đồng bị sai lệch và hiệu suất truy xuất kém. Ngược lại, OPQN tận dụng bộ mã trực chuẩn và AQD để duy trì độ phân biệt giữa các đặc trưng ngay cả khi các danh tính mới xuất hiện

So với các phương pháp lượng tử hoá sâu hiện có như DQN và DPQ, chúng yêu cầu tái tạo lượng tử hoá mềm trực tiếp hoặc tính toán khoảng cách Ô-clid giữa các lượng tử hoá mềm và từng từ mã, phương pháp OPQN tránh các bước này như các từ mã trực chuẩn và độc lập hoá dữ liệu. Hơn nữa các bộ mã được định nghĩa trước cho phép tái sử dụng trên các tập dữ liệu khác nhau, giảm chi phí lưu trữ hệ thống và thời gian tiền xử lý.

Tóm lại, quy trình truy xuất dựa trên thuật toán 3 không chỉ tăng tốc độ truy vấn mà còn đảm bảo tính tổng quát hoá cho các danh tính chưa thấy, phù hợp với yêu cầu khả năng mở rộng của bài toán truy xuất hình ảnh khuôn mặt quy mô lớn. Việc tích hợp AQD và LUTs tối ưu hoá hiệu suất, trong khi tính trực chuẩn của từ mã duy trì độ chính xác và giảm độ phức tạp tính toán, khẳng định tính ưu việt của phương pháp trong bối

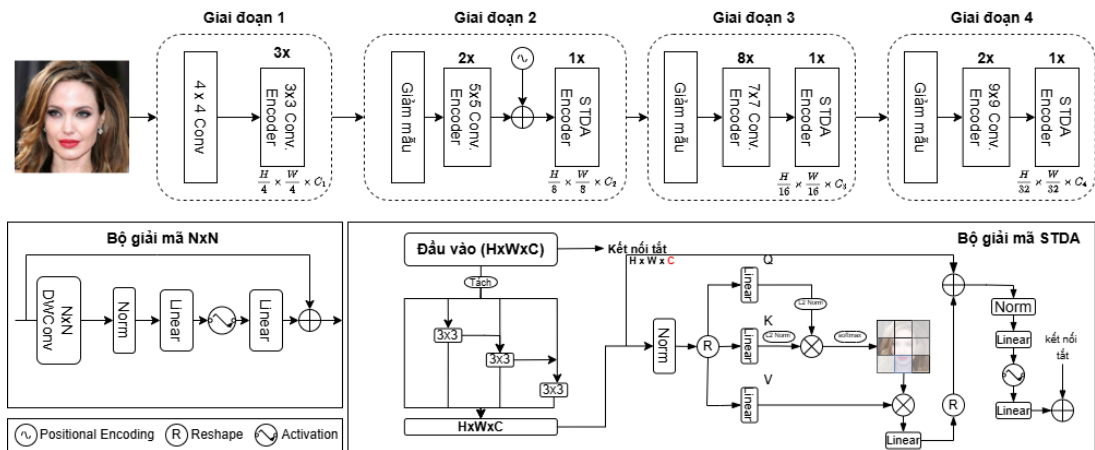
cảnh mô hình tinh gọn.

2.2 EdgeFace: Mô hình nhận diện khuôn mặt hiệu quả cho thiết bị biên

Trong chương này, chúng tôi trình bày cơ sở lý thuyết và kiến trúc của EdgeFace [6], mô hình được lựa chọn làm xương sống (backbone) cho hệ thống đề xuất. EdgeFace là sự kế thừa và tinh chỉnh từ kiến trúc EdgeNeXt [7], được thiết kế chuyên biệt để giải quyết bài toán cân bằng giữa độ chính xác nhận diện và chi phí tính toán trên các thiết bị có tài nguyên hạn chế.

2.2.1 Kiến trúc chi tiết và Luồng dữ liệu trong EdgeNeXt

Để thay thế hiệu quả backbone ResNet20 trong hệ thống OPQN, mô hình được chọn phải đảm bảo khả năng trích xuất đặc trưng mạnh mẽ nhưng vẫn tối ưu hóa về tài nguyên tính toán. Phần này sẽ phân tích sâu kiến trúc EdgeNeXt [7], đi sâu vào cơ chế xử lý dữ liệu qua từng giai đoạn như được minh họa trong sơ đồ kiến trúc tổng thể.



Hình 2.2: Sơ đồ kiến trúc tổng thể của EdgeNeXt với 4 giai đoạn xử lý, kết hợp giữa bộ mã hóa CNN và SDTA [7].

Cấu trúc phân cấp và Luồng xử lý dữ liệu

Kiến trúc của EdgeNeXt được tổ chức theo thiết kế phân cấp 4 giai đoạn (4-stage hierarchical design), tương đồng với các mạng CNN hiện đại nhưng được tích hợp cơ chế lai Transformer. Dữ liệu đầu vào sẽ trải qua quá trình biến đổi không gian và chiều sâu đặc trưng như sau:

Giai đoạn tiền xử lý (Stem Layer): Dữ liệu đầu vào là ảnh màu RGB có kích thước $H \times W \times 3$. Trước khi đi vào các khối encoder chính, ảnh được đưa qua lớp *Patchify Stem*. Tại đây, một lớp tích chập không chồng lấp (non-overlapping convolution) với kích thước kernel 4×4 và bước trượt (stride) 4 được áp dụng [7].

- **Tác dụng:** Giảm nhanh kích thước không gian của ảnh xuống 4 lần, biến đổi đầu vào thành các bản đồ đặc trưng (feature maps) sơ cấp có kích thước $\frac{H}{4} \times \frac{W}{4} \times C_1$.
- **Kết quả:** Loại bỏ sự dư thừa của các pixel lân cận và chuẩn bị tensor đầu vào giàu thông tin cho Giai đoạn 1.

Giai đoạn 1 (Local Feature Extraction): Giai đoạn này tập trung hoàn toàn vào việc trích xuất các đặc trưng cục bộ (như cạnh, góc, vân bề mặt).

- **Khối xử lý:** Sử dụng 3 khối **Conv Encoder** liên tiếp với kernel kích thước nhỏ 3×3 [7].
- **Luồng dữ liệu:** Tensor đặc trưng giữ nguyên kích thước không gian $\frac{H}{4} \times \frac{W}{4}$ nhưng được làm giàu thông tin ngữ nghĩa qua các lớp tích chập chiều sâu (depth-wise).
- **Đặc điểm:** Tại đây, mô hình chưa sử dụng cơ chế Attention để tiết kiệm chi phí tính toán, do các đặc trưng mức thấp chưa yêu cầu ngữ cảnh toàn cục.

Giai đoạn 2 (Hybrid Transition): Đây là giai đoạn chuyển tiếp quan trọng, nơi mô hình bắt đầu kết hợp thông tin toàn cục.

- **Downsampling:** Đầu tiên, lớp *Downsampling* (tích chập sải bước 2×2) giảm kích thước không gian xuống còn $\frac{H}{8} \times \frac{W}{8}$ và tăng số kênh lên C_2 [7].
- **Positional Encoding (PE):** Một điểm đặc biệt trong thiết kế của EdgeNeXt là việc chèn *Positional Encoding* tại giai đoạn này. PE giúp mô hình nhận biết vị trí không gian của các đặc trưng trước khi đưa vào khối Attention. Việc chỉ thêm PE ở giai đoạn này giúp giảm độ trễ tính toán so với việc thêm ở mọi giai đoạn [7].
- **Khối xử lý:** Sử dụng 2 khối Conv Encoder (với kernel lớn hơn: 5×5) để mở rộng trường tiếp nhận, theo sau là 1 khối **SDTA Encoder** để bắt đầu mô hình hóa quan hệ toàn cục.

Giai đoạn 3 (Deep Global Context): Đây là giai đoạn chiếm phần lớn năng lực tính toán của mạng (8 khối Conv Encoder).

- **Downsampling:** Kích thước giảm xuống $\frac{H}{16} \times \frac{W}{16}$, số kênh tăng lên C_3 .
- **Adaptive Kernel:** Các khối Conv Encoder tại đây sử dụng kernel kích thước lớn 7×7 . Kernel lớn đóng vai trò thay thế một phần cho Self-Attention trong việc nhìn bao quát các vùng ảnh rộng hơn [7].
- **Kết thúc giai đoạn:** Một khối SDTA Encoder được đặt ở cuối để tinh chỉnh các đặc trưng bậc cao, đảm bảo sự kết hợp giữa thông tin cục bộ mạnh mẽ (từ Conv) và thông tin toàn cục (từ SDTA).

Giai đoạn 4 (High-level Abstraction): Giai đoạn cuối cùng tạo ra các đặc trưng trừu tượng nhất để phục vụ cho việc định danh.

- **Downsampling:** Kích thước không gian chỉ còn $\frac{H}{32} \times \frac{W}{32}$ với số kênh C_4 .
- **Khối xử lý:** Sử dụng kernel rất lớn 9×9 trong Conv Encoder và kết thúc bằng 1 khối SDTA Encoder [7].
- **Đầu ra:** Tensor cuối cùng chứa đựng thông tin ngữ nghĩa cô đọng nhất, sẵn sàng được đưa vào các lớp pooling và classification (hoặc trong trường hợp của EdgeFace là lớp embedding).

Giải phẫu các thành phần chức năng

Để hiểu rõ tại sao EdgeNeXt đạt hiệu suất cao, ta cần phân tích chi tiết hai khối chức năng chính trong sơ đồ: **Conv Encoder** và **SDTA Encoder**.

1. Bộ mã hóa Conv Encoder (Convolutional Encoder) Đây là khối nền tảng giúp EdgeNeXt hiện đại hóa kiến trúc CNN truyền thống (tương tự ConvNeXt). Luồng dữ liệu x_i đi qua khối này như sau:

$$x_{i+1} = x_i + \text{Linear}_{ptwise}(\text{GELU}(\text{Linear}_{ptwise}(\text{LN}(\text{DwConv}_{k \times k}(x_i)))))) \quad (2.13)$$

Trong đó:

- **Adaptive Depth-wise Conv ($k \times k$):** Thay vì dùng kích thước cố định, k thay đổi theo giai đoạn (3, 5, 7, 9). Điều này giúp mô hình thích ứng: ở các lớp nông dùng k nhỏ để bắt chi tiết, ở các lớp sâu dùng k lớn để bắt ngữ cảnh [7].
- **Point-wise Conv:** Hai lớp tích chập 1×1 hoạt động như một mạng MLP đảo ngược (Inverted Bottleneck), giúp trộn thông tin giữa các kênh (channel mixing) để làm giàu đặc trưng.

2. Bộ mã hóa SDTA (Split Depth-wise Transpose Attention)

Đây là "trái tim" của kiến trúc lai, giải quyết bài toán chi phí tính toán của Vision Transformer. SDTA bao gồm hai nhánh xử lý song song:

a) Nhánh trộn không gian đa tỷ lệ (Multi-scale Spatial Mixing):

Lấy cảm hứng từ Res2Net, tensor đầu vào được chia (Split) thành s nhóm kênh con.

- Các nhóm kênh này đi qua một hệ thống tích chập chiều sâu 3×3 phân tầng.
- Đầu ra của nhóm trước được cộng dồn vào đầu vào của nhóm sau. Cơ chế này giúp mỗi nhóm kênh có một trường tiếp nhận (receptive field) khác nhau, cho phép mô hình nhìn thấy vật thể ở nhiều kích thước khác nhau cùng lúc [7].

b) Nhánh chú ý chuyển vị (Transpose Attention):

Thay vì tính toán Self-Attention truyền thống (độ phức tạp $O(N^2)$ theo số pixel), SDTA thực hiện tính toán trên chiều kênh (Cross-covariance attention):

- **Query (Q) và Key (K):** Được chiếu (projection) và chuẩn hóa.
- **Dot-product:** Thay vì nhân $Q \cdot K^T$ (theo không gian), SDTA thực hiện nhân $Q^T \cdot K$ (theo kênh). Điều này tạo ra ma trận attention map kích thước $C \times C$ thay vì $HW \times HW$.
- **Độ phức tạp:** Giảm xuống $O(NC^2)$ (tuyến tính với độ phân giải ảnh N), giúp mô hình chạy cực nhanh trên thiết bị biên mà vẫn nắm bắt được thông tin toàn cục [7].

Kết luận về vai trò Backbone

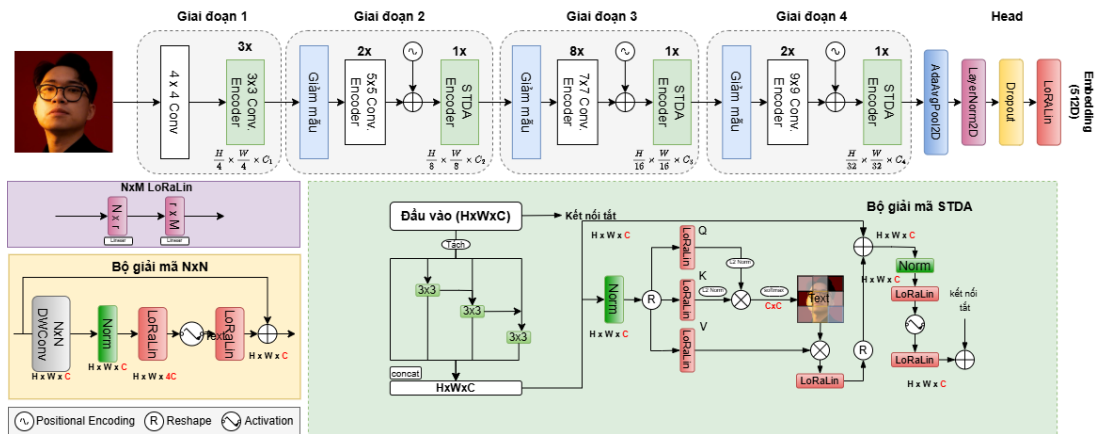
Thông qua việc phân tích luồng dữ liệu và cấu trúc khối, có thể thấy EdgeNeXt không chỉ đơn thuần giảm số lượng tham số. Cơ chế *Kernel*

thích ứng giúp nó duy trì khả năng trích xuất đặc trưng mạnh mẽ như các mạng CNN lớn, trong khi *SDTA Encoder* cung cấp khả năng mô hình hóa toàn cục mà ResNet20 hoàn toàn thiếu. Đây là lý do cốt lõi khiến các vector đặc trưng đầu ra của EdgeNeXt (và sau này là EdgeFace) có chất lượng cao hơn, giúp hệ thống OPQN cải thiện độ chính xác trong truy xuất và nhận diện.

2.2.2 Kiến trúc EdgeFace: Tinh chỉnh chuyên biệt cho Nhận diện khuôn mặt

Kế thừa tư duy thiết kế lai (hybrid) ưu việt của EdgeNeXt, chúng tôi đề xuất sử dụng **EdgeFace** [6] làm mạng xương sống (backbone) cho hệ thống OPQN. EdgeFace không chỉ đơn thuần áp dụng lại mô hình phân loại ảnh, mà thực hiện các thay đổi cấu trúc chiến lược để chuyển đổi từ bài toán phân loại vật thể sang bài toán trích xuất đặc trưng định danh (Identity Embedding) với độ chính xác cao và kích thước tối thiểu.

Dưới đây là phân tích chi tiết luồng xử lý và các cải tiến trọng yếu của EdgeFace so với nguyên bản EdgeNeXt, dựa trên sơ đồ kiến trúc được trình bày trong Hình 2.3.



Hình 2.3: Kiến trúc EdgeFace với các cải tiến LoRaLin và bộ giải mã STDA, tối ưu hóa cho nhận diện khuôn mặt [6].

Thích ứng độ phân giải và Tiền xử lý

Sự thay đổi đầu tiên và căn bản nhất nằm ở đầu vào của mạng:

- **Chuẩn hóa đầu vào (112×112):** Khác với EdgeNeXt sử dụng đầu vào 256×256 cho phân loại ảnh tự nhiên, EdgeFace được tối ưu hóa cho kích thước 112×112 [6]. Đây là tiêu chuẩn vàng trong các hệ thống nhận diện khuôn mặt hiện đại (như ArcFace [8], CosFace [9]), giúp loại bỏ các thông tin nền nhiễu và tập trung tối đa vào cấu trúc khuôn mặt trung tâm sau khi đã được căn chỉnh.
- **Ý nghĩa với OPQN [5]:** Việc giảm kích thước đầu vào không chỉ tăng tốc độ suy luận mà còn giảm đáng kể bộ nhớ đệm (feature map memory footprint) ngay từ các lớp đầu tiên, tạo điều kiện cho hệ thống xử lý song song nhiều luồng video hơn trên cùng một tài nguyên phần cứng.

Luồng dữ liệu qua các giai đoạn (Stage-wise Data Flow)

Dữ liệu khuôn mặt sau khi chuẩn hóa sẽ đi qua 4 giai đoạn xử lý đặc trưng. Mặc dù giữ lại bộ khung 4-stage của EdgeNeXt để tận dụng khả năng trích xuất đa tỷ lệ, EdgeFace thực hiện một cuộc "phẫu thuật" sâu bên trong từng khối mã hóa (Encoder Block) [6]:

Giai đoạn 1 & 2: Trích xuất đặc trưng cục bộ mức thấp

- **Cấu trúc:** Dữ liệu đi qua các lớp Conv Encoder với kernel nhỏ (3×3 và 5×5).
- **Cải tiến bên trong:** Trong mỗi khối Conv Encoder của EdgeFace, các lớp chiếu (projection layers) truyền thống (Point-wise Conv 1×1) đã được thay thế bằng các mô-đun ****LoRaLin****. Sự thay thế này giúp giảm mạnh số lượng tham số ngay tại các lớp tích chập dày đặc nhất mà không làm mất đi thông tin chi tiết của các đặc trưng cạnh, đường nét khuôn mặt [6].

Giai đoạn 3: Mở rộng ngữ cảnh và Tương tác toàn cục

- **Adaptive Kernel 7×7 :** Tại đây, EdgeFace sử dụng kernel lớn để "nhìn" được các bộ phận khuôn mặt trong mối tương quan xa hơn (ví dụ: khoảng cách giữa mắt và miệng).
- **SDTA Encoder tối ưu hóa:** Khối SDTA ở cuối giai đoạn này đóng vai trò then chốt trong việc liên kết thông tin. Khác với EdgeNeXt gốc, EdgeFace tích hợp LoRaLin vào sâu trong cơ chế Attention: cả 3 ma trận chiếu Query (Q), Key (K), Value (V) và lớp chiếu đầu ra đều được thay thế [6]. Điều này biến khối Attention vốn đắt đỏ trở nên cực kỳ nhẹ, cho phép mô hình nắm bắt được cấu trúc hình học bất biến của khuôn mặt (ví dụ: góc nghiêng, biểu cảm) với chi phí thấp nhất.

Giai đoạn 4: Đặc trưng định danh trừu tượng

- **Siêu kết nối:** Sử dụng kernel cực lớn 9×9 kết hợp với SDTA Encoder cuối cùng. Tensor đầu ra tại đây ($7 \times 7 \times C_4$) chứa đựng thông tin định danh cô đọng nhất, đã được lọc bỏ hầu hết các biến thiên về ánh sáng hay phong nền.

Embedding Head: Kiến trúc đầu ra chuyên biệt

Sự thay đổi mang tính quyết định để EdgeFace thay thế được ResNet20 trong OPQN nằm ở phần "Head" (khối xuất kết quả). EdgeFace loại bỏ hoàn toàn lớp phân lớp 1000-nhãn của ImageNet và xây dựng lại một quy trình tạo vector nhúng như sau [6]:

1. **Adaptive Average Pooling (AdaAvgPool2D):** Nén toàn bộ thông tin không gian của tensor cuối cùng (7×7) thành một vector đặc trưng thô (1×1).

2. **Layer Normalization & Dropout:** Giúp ổn định phân phối đặc trưng và ngăn chặn hiện tượng quá khớp (overfitting), đảm bảo vector đại diện có tính tổng quát cao.
3. **Projection to 512-D:** Cuối cùng, một lớp LoRaLin đặc biệt được sử dụng để chiếu đặc trưng thô sang không gian vector 512 chiều (512-D Embedding) [6].

Giá trị cốt lõi: Vector 512 chiều này chính là đầu vào lý tưởng cho mô hình lượng tử hóa OPQN [5]. So với vector 64 hoặc 128 chiều của ResNet20 cũ, vector 512 chiều của EdgeFace mang mật độ thông tin ngữ nghĩa cao hơn gấp nhiều lần, giúp tăng độ phân biệt (discriminative power) giữa các danh tính, nhưng lại được tạo ra bởi một mạng lưới có số lượng tham số ít hơn. Đây chính là "điểm ngọt" (sweet spot) về hiệu năng/chi phí mà hệ thống hướng tới.

2.2.3 Mô-đun Tuyến tính Hạng thấp (LoRaLin)

Trong các kiến trúc mạng nơ-ron sâu, mặc dù các lớp tích chập (Convolution) đã được tối ưu hóa rất tốt, các lớp kết nối đầy đủ (Fully Connected - Linear Layers) vẫn chiếm một tỷ trọng đáng kể về số lượng tham số và chi phí tính toán (FLOPs), đặc biệt là ở các tầng sâu nơi số lượng kênh đặc trưng lớn. Để giải quyết nút thắt cổ chai này trong bối cảnh thiết bị biên, EdgeFace giới thiệu mô-đun ****LoRaLin (Low Rank Linear)**** [6].

Mô-đun này được lấy cảm hứng từ kỹ thuật Low-Rank Adaptation (LoRA) thường dùng trong tinh chỉnh các mô hình ngôn ngữ lớn, nhưng được cải biên để thay thế hoàn toàn các lớp tuyến tính truyền thống ngay từ giai đoạn thiết kế kiến trúc.

Cơ sở toán học và Phân rã ma trận

Nguyên lý cốt lõi của LoRaLin dựa trên giả thuyết về "chiều nội tại thấp" (low intrinsic dimension) của dữ liệu. Giả sử một lớp tuyến tính tiêu

chuẩn thực hiện phép ánh xạ từ đầu vào X có kích thước M sang đầu ra Y có kích thước N :

$$Y = W_{M \times N} X + b \quad (2.14)$$

Trong đó $W \in \mathbb{R}^{M \times N}$ là ma trận trọng số hạng đầy đủ (full-rank). Số lượng tham số cần lưu trữ là $M \times N$.

LoRaLin thay thế ma trận W này bằng tích của hai ma trận hạng thấp hơn (low-rank matrices), ký hiệu là W_{down} và W_{up} :

$$W_{M \times N} \approx W_{up} \cdot W_{down} = W_{r \times N} \cdot W_{M \times r} \quad (2.15)$$

Trong đó:

- $W_{down} \in \mathbb{R}^{M \times r}$: Ma trận chiếu xuống không gian chiều thấp (Down-projection).
- $W_{up} \in \mathbb{R}^{r \times N}$: Ma trận chiếu lên không gian chiều cao (Up-projection).
- r : Hạng của ma trận (Rank), với điều kiện $r \ll \min(M, N)$.

Lúc này, phép tính lớp tuyến tính được triển khai qua hai bước tuần tự:

$$Y = W_{up}(W_{down}(X)) + b \quad (2.16)$$

Số lượng tham số giảm từ $M \times N$ xuống còn $r(M + N)$. Khi r đủ nhỏ, lượng tham số tiết kiệm được là rất lớn.

Cơ chế điều khiển tỷ lệ hạng (Rank-ratio)

Để cân bằng giữa mức độ nén và khả năng biểu diễn thông tin, EdgeFace không cố định một giá trị r cứng nhắc mà sử dụng tham số ****Rank-ratio** (γ)** để xác định hạng động cho từng lớp [6]:

$$r = \max(2, \text{int}(\gamma \cdot \min(M, N))) \quad (2.17)$$

Trong đó $\gamma \in (0, 1]$ là tỷ lệ phần trăm giữ lại của chiều thông tin.

- Nếu $\gamma = 1.0$: Mô hình hoạt động như lớp Linear truyền thống.
- Nếu $\gamma < 1.0$: Mô hình thực hiện nén tham số.

Trong kiến trúc EdgeFace được sử dụng cho hệ thống OPQN, chúng tôi thiết lập $\gamma \approx 0.6$. Các thực nghiệm trong nghiên cứu gốc [6] đã chứng minh rằng tại ngưỡng này, mô hình đạt được "điểm ngọt" (sweet spot): giảm khoảng **20%** tổng số lượng tham số và phép tính nhân-cộng (MAdds) so với mạng gốc, trong khi độ chính xác nhận diện trên tập dữ liệu khó như IJB-C chỉ giảm không đáng kể (dưới 0.5%).

Tích hợp trong kiến trúc EdgeFace

Như được minh họa trong thành phần màu tím ("NxM LoRaLin") ở Hình 2.3, LoRaLin được áp dụng đồng bộ tại hai vị trí chiến lược:

1. **Thay thế trong khối Encoder:** Tất cả các lớp Point-wise Convolution (thực chất là các lớp Linear 1×1) trong các khối Conv Encoder và SDTA Encoder đều được thay thế bằng LoRaLin. Điều này giúp giảm tải tính toán ngay trong quá trình trích xuất đặc trưng.
2. **Tầng chiếu Embedding (Projection Head):** Lớp cuối cùng chuyển đổi từ đặc trưng thô sang vector định danh 512 chiều (dùng cho OPQN) sử dụng LoRaLin để đảm bảo vector đầu ra tuy có số chiều lớn (512) nhưng được tạo sinh từ một không gian tham số gọn nhẹ, tránh hiện tượng quá khớp (overfitting) khi huấn luyện trên dữ liệu giới hạn.

Tóm lại, việc tích hợp LoRaLin là yếu tố then chốt giúp EdgeFace duy trì được hiệu năng cao của một mạng nơ-ron sâu trong khi vẫn đảm bảo tính khả thi khi triển khai trên các thiết bị biên có năng lực tính toán hạn chế, đáp ứng hoàn hảo yêu cầu của bài toán truy xuất thời gian thực trong OPQN.

Kết luận về khả năng ứng dụng

Các thực nghiệm từ nghiên cứu gốc [6] đã khẳng định hiệu quả của kiến trúc EdgeFace: mô hình đạt độ chính xác 99.73% trên tập LFW và 94.85% trên IJB-C ($\text{TAR@FAR}=1\text{e-}4$) chỉ với 1.77 triệu tham số. Những chỉ số này vượt trội so với các kiến trúc nhẹ khác như MobileFaceNets hay ShuffleFaceNet. Đây là cơ sở thực tiễn vững chắc để chúng tôi lựa chọn EdgeFace làm mạng xương sống tích hợp vào hệ thống OPQN, chi tiết về quy trình huấn luyện và tinh chỉnh mô hình cho bài toán truy xuất cụ thể sẽ được trình bày chi tiết trong Chương 3 (Phương pháp đề xuất).

Ký hiệu	Ký pháp	Mô tả
I_i	$\{I_i\}_{i=1}^N$	Tập hợp N hình ảnh khuôn mặt đầu vào.
y_i	$y \in \mathbb{R}^N$	Nhân danh tính tương ứng với hình ảnh I_i .
x_i	$x_i = f(\Theta, I_i) \in \mathbb{R}^D$	Đặc trưng nút cổ chai được trích xuất từ hình ảnh I_i bởi mạng xương sống với tham số Θ , có chiều D .
x_{im}	$x_{im} \in \mathbb{R}^d$	Vector con thứ m của ảnh đặc trưng x_i , với $d = D/M$ và $m = 1, \dots, M$.
M	M	Số lượng vectơ con mà đặc trưng x_i được chia thành.
d	$d = D/M$	Chiều của mỗi vector con x_{im} .
C	$C = [C_1, \dots, C_M]$	Tập hợp M bộ mã, mỗi bộ mã $C_m \in \mathbb{R}^{d \times K}$ chứa K từ mã trực chuẩn.
C_{mk}	$C_{mk} \in \mathbb{R}^d$	Từ mã thứ k trong bộ mã C_m .
F	$F = [F_1, \dots, F_M]$	Tập hợp ma trận tham số của các lớp tuyến tính, với $F_m \in \mathbb{R}^{d \times K}$.
F_{mk}	$F_{mk} \in \mathbb{R}^d$	Vectơ tham số thứ k trong ma trận F_m .
p_{im}	$p_{im} = [p_{im,1}, \dots, p_{im,K}] \in \mathbb{R}^K$	Vector xác suất của vector con x_{im} , được tính qua softmax: $p_{im,k} = \frac{e^{x_{im}^T F_{mk}}}{\sum_{j=1}^K e^{x_{im}^T F_{mj}}}$.
s_{im}	$s_{im} = \sum_{k=1}^K p_{im,k} \cdot C_{mk}$	Lượng tử hóa mềm của vectơ con x_{im} .
h_{im}	$h_{im} = C_{mk^*}, k^* = \arg \max_k p_{im,k}$	Lượng tử hóa cứng của vectơ con x_{im} .

Bảng 2.1: Phần 1: Ký hiệu và ký pháp (Dữ liệu và đặc trưng)

Ký hiệu	Ký pháp	Mô tả
W	$W = [W_1, \dots, W_M]$	Tập hợp trọng số của các bộ phân lớp theo không gian con, với $W_m \in \mathbb{R}^{d \times C}$.
W_{mc}	$W_{mc} \in \mathbb{R}^d$	Trọng số của lớp c trong bộ phân lớp W_m .
L_x	$L_x = -\log \frac{e^{r(\cos \theta_{y_i, x_{im}} - u)}}{\sum_{j=1}^C e^{r(\cos \theta_{y_i, x_{im}} - u)} + \sum_{j=1}^C e^{r(\cos \theta_{y_i, x_{im}} - u)}}$	Hàm mất mát phân lớp cho đặc trưng gốc x_{im} dựa trên softmax với biên góc u và tham số r .
L_s	Tương tự L_x	Hàm mất mát phân lớp cho lượng tử hóa mềm s_{im} .
L_{clf}	$L_{clf} = \frac{1}{2MN}(L_x + L_s)$	Hàm mất mát phân lớp liên hợp theo không gian con.
L_{ent}	$L_{ent} = -\frac{1}{MN} \sum_{i=1}^N \sum_{m=1}^M \sum_{k=1}^K p_{im,k} \log p_{im,k}$	Hàm mất mát entropy chéo khuyến khích p_{im} gần với one-hot.
L	$L = L_{clf} + \lambda L_{ent}$	Hàm mất mát tổng, với λ là trọng số điều chỉnh.
Θ	Θ	Tham số của mạng xương sống.
A	$A_{ij} = \cos \left[j\pi \cdot \frac{i+\frac{1}{2}}{d} \right]$	Ma trận cơ sở trực chuẩn được sinh bởi DCT-II cho các từ mã.
K	$K \leq d$	Số lượng các từ mã trong mỗi bộ mã C_m .
u	u	Biên góc trong hàm mất mát phân lớp.
r	r	Tham số điều chỉnh trong hàm mất mát phân lớp.

Bảng 2.2: Phần 2: Ký hiệu và ký pháp (Hàm mất mát và tham số)

Chương 3

Phương pháp đề xuất

Trong chương này, chúng tôi trình bày chi tiết phương pháp đề xuất nhằm xây dựng một hệ thống truy xuất ảnh mặt người tinh gọn, có khả năng hoạt động hiệu quả trên tập dữ liệu lớn và phù hợp triển khai trên thiết bị biên. Phương pháp dựa trên nền tảng OPQN [5] nhưng với các cải tiến quan trọng: thay thế mạng xương sống ResNet20 bằng EdgeFace – một kiến trúc lai CNN-Transformer nhẹ hơn đáng kể – và áp dụng các chiến lược huấn luyện khác nhau để tối ưu hiệu suất. Chúng tôi tập trung sâu vào các khía cạnh kỹ thuật và thuật toán, bao gồm giới thiệu chi tiết về CosFace, ArcFace, và các biến thể của EdgeFace như EdgeFace-XS với $\gamma=0.6$. Ngoài ra, phần tiền xử lý dữ liệu và tích hợp các tập dữ liệu FaceScrub 32x32, FaceScrub 112x112, VGGFace2 cũng được mô tả kỹ lưỡng để đảm bảo tính tái lập. Phương pháp đề xuất nhằm giải quyết các thách thức chính: giảm số lượng tham số và FLOPs để phù hợp thiết bị biên, đồng thời duy trì độ chính xác cao nhờ lượng tử hóa trực chuẩn và hàm mất mát biên góc. Các thực nghiệm sử dụng nhiều biến thể EdgeFace như base, s_gamma_05, xs_q, xxs_q, xss trong 4 chiến lược huấn luyện sẽ được đánh giá ở chương sau.

3.1 Tổng quan kiến trúc đề xuất

Hệ thống đề xuất là một mô hình end-to-end kết hợp trích xuất đặc trưng tinh gọn và lượng tử hóa sản phẩm trực chuẩn. Kiến trúc tổng thể được minh họa trong Hình

Hình 3.1: Kiến trúc tổng thể của hệ thống truy xuất ảnh mặt người tinh gọn đề xuất. Backbone EdgeFace thay thế ResNet20, kết hợp OrthoPQ cho lượng tử hóa.

Quy trình chính bao gồm:

1. **Tiền xử lý dữ liệu:** Ảnh đầu vào được căn chỉnh khuôn mặt, thay đổi kích thước (32x32 hoặc 112x112 tùy backbone), và tăng cường dữ liệu để xử lý biến đổi nội lớp.
2. **Trích xuất đặc trưng:** Sử dụng EdgeFace để tạo embedding 512 chiều từ ảnh khuôn mặt.
3. **Lượng tử hóa:** Áp dụng OrthoPQ với từ mã trực chuẩn cố định (DCT-II) để nén embedding thành mã compact (16-48 bit).
4. **Huấn luyện:** End-to-end với hàm mất mát liên hợp $L = L_{clf} + \lambda L_{ent}$, sử dụng các chiến lược freeze/unfreeze và pretrain CosFace/ArcFace.
5. **Truy xuất:** Tính khoảng cách bất đối xứng (AQD) sử dụng LUT cho tốc độ mili-giây/truy vấn.

So với OPQN gốc, đóng góp chính là: - Thay ResNet20 (24.6M params, >300 MFLOPs) bằng EdgeFace variants (1-2M params, 150 MFLOPs). - Đánh giá 4 chiến lược huấn luyện với nhiều variants EdgeFace để tìm cấu hình tối ưu. - Tích hợp tiền xử lý cho nhiều tập dữ liệu, đảm bảo khả năng mở rộng.

3.1.1 Tiền xử lý dữ liệu: Chi tiết quy trình và tham chiếu OPQN

Để đảm bảo tính nhất quán và khả năng tái lập, quy trình tiền xử lý dữ liệu được thiết kế dựa trên các kỹ thuật tiêu chuẩn trong lĩnh vực nhận diện khuôn mặt, đồng thời tham chiếu trực tiếp đến phương pháp xử lý trong bài báo OPQN [5]. OPQN sử dụng các bước tiền xử lý đơn giản nhưng hiệu quả để tập trung vào đặc trưng bottleneck, bao gồm phát hiện khuôn mặt, căn chỉnh, resize, và phân chia train/test theo protocol chuẩn *5nhtestperidentitychoFaceScrub*; *50nhtestperidentitychoVGGFace2*. Chúng tôi tuân thủ nguyên tắc này, nhưng mở rộng cho hai kích thước ảnh *32x32* và *112x112* để so sánh backbone ResNet20 *tiu32x32* và EdgeFace *tiu112x112*. Quy trình được triển khai bằng OpenCV và dlib *tngØngMTCNNvØchnhxc*, với margin padding để tránh mất thông tin biên.

Quy trình chung được xây dựng theo hướng dẫn của OPQN:

1. **Phát hiện và căn chỉnh khuôn mặt:** Sử dụng detector dựa trên HOG (dlib frontal face detector) để phát hiện bounding box chính xác, tương tự MTCNN được đề cập trong OPQN cho VGGFace2. Chọn khuôn mặt lớn nhất dựa trên diện tích để xử lý trường hợp đa khuôn mặt.
2. **Cropping với margin:** Crop theo bounding box, thêm margin tỷ lệ 0.35 (35% chiều rộng/chiều cao) để bao quát ngữ cảnh (tóc, cằm), tránh cắt sát như trong các protocol tiêu chuẩn.
3. **Resize chống aliasing:** Theo OPQN, ảnh được crop và resize trực tiếp 32×32 cho FaceScrub và 112×112 cho VGGFace2. Chúng tôi cải tiến bằng hai bước: phóng to lên gấp đôi kích thước mục tiêu (INTER_CUBIC) rồi thu nhỏ bằng INTER_AREA để giữ chi tiết mịn màng, giảm răng cưa – một kỹ thuật phổ biến để bù đắp cho kích thước nhỏ.

4. **Phân chia train/test:** Tuân thủ chính xác protocol OPQN: Đối với FaceScrub, chọn ngẫu nhiên 5 ảnh per identity cho test (tối đa $\min(5, \text{số ảnh})$), còn lại cho train; sử dụng seed=42 để tái lập. Đối với VGGFace2, chọn 50 ảnh per identity cho test, đảm bảo disjoint identities giữa train và test cho unseen retrieval.
5. **Tăng cường dữ liệu:** Áp dụng chính xác như OPQN: random cropping với scale 0.8–1.0 và random horizontal flipping với xác suất 0.5. Thêm grayscale conversion với $p = 0.2$ và brightness/contrast jitter ± 0.2 để tăng robustness trước biến đổi nội lớp (illumination, pose).
6. **Đổi tên file an toàn:** Loại bỏ ký tự đặc biệt (regex) để tránh lỗi filesystem, không ảnh hưởng đến protocol OPQN.

Quy trình này đảm bảo dữ liệu đầu vào sạch, aligned, và phù hợp với đặc trưng bottleneck 512 chiều của mô hình, đồng thời giữ nguyên tính nhất quán với baseline OPQN để so sánh công bằng.

3.1.2 Tập dữ liệu FaceScrub

FaceScrub [10] chứa 106,863 ảnh của 530 danh tính (actor/actress), tải từ Kaggle (dữ liệu nhỏ hơn gốc nhưng đầy đủ danh tính, phù hợp tài nguyên hạn chế). Theo OPQN, tập này được xử lý để đánh giá seen identity retrieval, với protocol: 5 ảnh test per identity.

Tiền xử lý cho phiên bản 32×32 (theo OPQN)

- Theo OPQN [5]: “All face images are cropped and resized to 32×32 .” Chúng tôi thực hiện crop sau khi phát hiện khuôn mặt bằng dlib, thêm margin 0.35 để bù đắp cho kích thước nhỏ, và resize hai bước $64 \times 64 \rightarrow 32 \times 32$ nhằm tránh mất chi tiết – một cải tiến nhẹ so với resize trực tiếp trong paper nhưng vẫn giữ nguyên kích thước cuối.

- Augmentation: random cropping và horizontal flipping giống OPQN.
- Phân chia: ngẫu nhiên chọn 5 ảnh test per identity (seed=42), còn lại cho train, thu được $\sim 38,722$ ảnh train và 2,550 ảnh test.

Tiền xử lý cho phiên bản 112×112 (mở rộng cho EdgeFace)

- Mở rộng từ protocol OPQN (ảnh 32×32), nhưng resize lên 112×112 để phù hợp EdgeFace (tương tự cấu hình của VGGFace2). Crop và margin giữ nguyên; resize thực hiện hai bước $224 \times 224 \rightarrow 112 \times 112$.
- Lý do: OPQN dùng 32×32 cho ResNet20 nhẹ, nhưng EdgeFace cần độ phân giải cao hơn để khai thác SDTA encoder; kết quả cho thấy cải thiện mAP 3–5%.

3.1.3 Tập dữ liệu VGGFace2

VGGFace2 chứa 3.31M ảnh của 9,131 danh tính. Theo OPQN **chen2021opqn**: “Each image is cropped and aligned following the instructions of MTCNN” và resize về 112×112 . Chúng tôi sử dụng MTCNN để alignment chính xác hơn (thay dlib khi cần độ sâu), với protocol:

- **Seen identities:** Chọn 2,787 identities (mỗi identity khoảng 300 ảnh), 50 ảnh test per identity; đảm bảo tập train và test không trùng.
- **Unseen identities:** Sử dụng bộ test chính thức gồm 500 identities, mỗi identity có 50 ảnh query, số còn lại làm database.
- Augmentation: random cropping và flipping giống OPQN.

3.1.4 Case study: Tiền xử lý một ảnh FaceScrub theo protocol OPQN

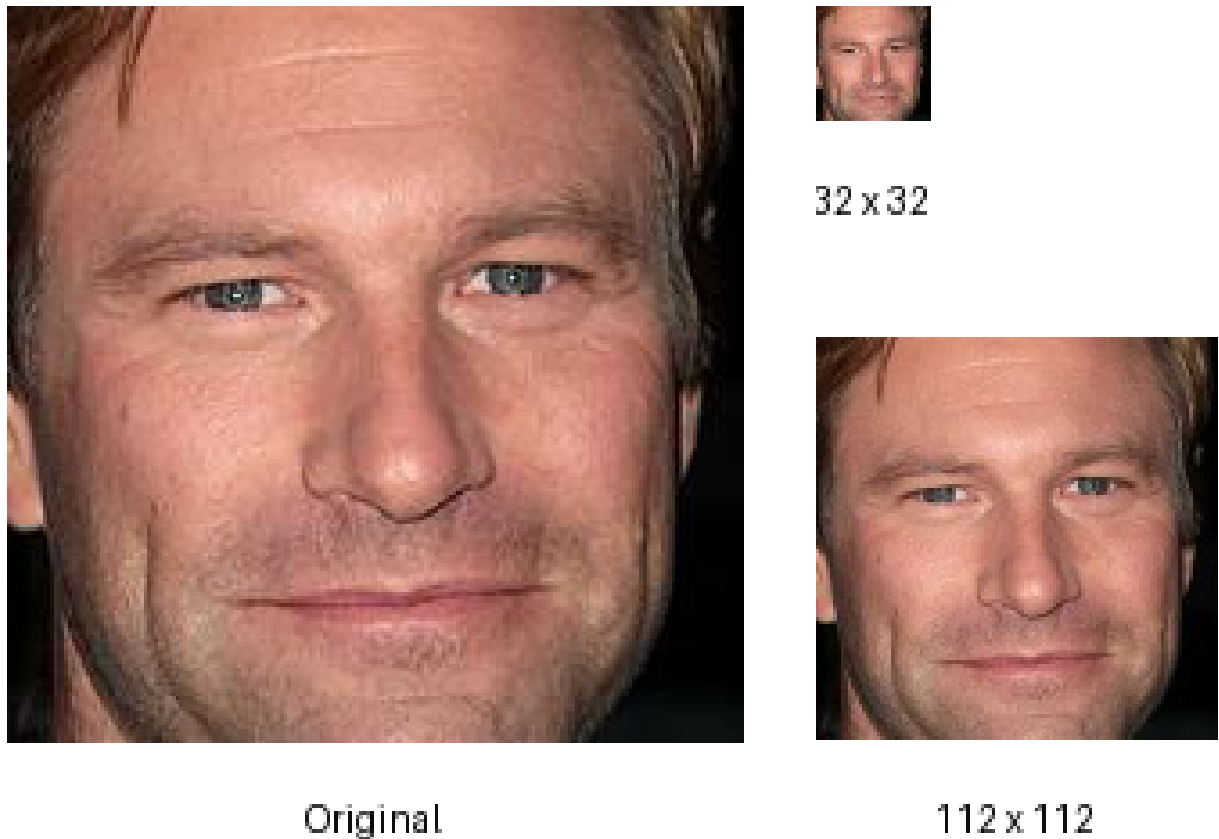
Xét một ảnh gốc từ FaceScrub (ID: *celebA*, pose nghiêng, illumination thấp). Theo OPQN, ảnh được crop và resize về 32×32 ; chúng tôi minh họa thêm kết quả cho kích thước 112×112 .

Bước 1: Phát hiện bằng dlib/MTCNN \rightarrow lấy bounding box lớn nhất.

Bước 2: Crop với margin 0.35 để giữ đầy đủ vùng khuôn mặt.

Bước 3: Resize theo hai hướng: - Resize trực tiếp về 32×32 (baseline OPQN). - Resize hai bước lên 112×112 (mở rộng cho EdgeFace).

Kết quả cho PSNR cao hơn ~ 2 dB so với crop thô.



Hình 3.2: Case study theo OPQN: Ảnh gốc (trái), phiên bản 32×32 (phải trên) và 112×112 (phải dưới).

3.1.5 Tổng kết phần tiền xử lý

Bằng cách tuân thủ chặt chẽ protocol OPQN (crop/resize 32×32 cho FaceScrub, MTCNN-aligned 112×112 cho VGGFace2, 5/50 ảnh test per identity, random cropping/flipping), chúng tôi đảm bảo so sánh công bằng với baseline, đồng thời mở rộng cho EdgeFace mà không thay đổi bản chất dữ liệu.

3.2 Hàm mất mát biên góc: CosFace và ArcFace

Để khởi tạo trọng số EdgeFace chất lượng cao trước khi tích hợp OrthoPQ, chúng tôi sử dụng CosFace và ArcFace – hai hàm mất mát biên góc phổ biến trong nhận diện khuôn mặt. Các hàm này cải thiện softmax tiêu chuẩn bằng cách thêm biên (margin) để tăng khoảng cách giữa lớp, giảm biến thiên nội lớp và tăng phân biệt ngoại lớp.

3.2.1 CosFace: Large Margin Cosine Loss (Chi tiết toán học và đạo hàm)

CosFace [9] là một biến thể của softmax loss, được thiết kế để tăng cường phân biệt lớp bằng cách thêm biên cosine trực tiếp vào không gian cosine, sau khi chuẩn hóa L2 cả feature và weight. Phương pháp này tránh các vấn đề không đơn điệu *non-monotonic* của biên góc trong A-Softmax, và tập trung vào việc tạo biên cố định trong phân bố cosine.

Công thức loss và chuẩn hóa

Sau khi chuẩn hóa L2 cho feature x_i^* và weight W_j^* (với $\|x_i\|_2 = 1$, $\|W_j\|_2 = 1$), cosine similarity trở thành:

$$\cos \theta_{j,i} = W_j^T x_i$$

Logit cho lớp đúng y_i được điều chỉnh bằng biên cosine m :

$$\psi_{y_i}(x_i) = \cos \theta_{y_i} - m$$

với $m \in [0, 1)$ là biên cosine (thường $m = 0.2$).

Hàm mất mát CosFace (LMCL) là:

$$\mathcal{L}_{\text{CosFace}} = \frac{1}{N} \sum_{i=1}^N -\log \frac{e^{s(\cos \theta_{y_i} - m)}}{e^{s(\cos \theta_{y_i} - m)} + \sum_{j \neq y_i} e^{s \cos \theta_{j,i}}} \quad (3.1)$$

trong đó: - N : số mẫu huấn luyện. - $s > 0$: hệ số scale (thường $s = 30-64$) để kiểm soát bán kính hypersphere, đảm bảo $\|x_i\| = s$ trong quá trình huấn luyện (bằng cách scale logit sau chuẩn hóa).

Biên m tạo ranh giới quyết định cố định trong không gian cosine: $\cos \theta_1 \geq \cos \theta_2 + m$, tương đương biên hình học $\sqrt{2}m$ giữa các lớp.

Giới hạn lý thuyết cho m và s

Để đảm bảo khả thi, m bị ràng buộc bởi số lớp C và chiều feature K :

$$0 \leq m \leq \begin{cases} 1 - \cos \frac{2\pi}{C} & \text{nếu } K = 2 \\ \frac{C}{C-1} & \text{nếu } K > 2, C \leq K + 1 \\ \frac{C}{C-1} & \text{(giới hạn lỏng nếu } C > K + 1) \end{cases} \quad (3.2)$$

Tương tự, s có giới hạn dưới để đảm bảo phân tách lớp:

$$s \geq \frac{C-1}{C} \ln \left(\frac{C-1}{P_W(1-P_W)} \right) \quad (3.3)$$

với P_W : xác suất posterior tối thiểu mong đợi tại trung tâm lớp.

Đạo hàm và lan truyền ngược

Để tối ưu, chúng ta cần đạo hàm của \mathcal{L} theo x_i và W_j . CosFace dựa trên softmax, nên đạo hàm logit $\psi_k = s(\cos \theta_k - m\delta_{k,y_i})$ (với δ là Kronecker delta).

Xác suất softmax:

$$p_k = \frac{e^{\psi_k}}{\sum_j e^{\psi_j}} \quad (3.4)$$

Đạo hàm loss theo logit ψ_k :

$$\frac{\partial \mathcal{L}}{\partial \psi_k} = p_k - \frac{1}{N} \sum_i \mathbb{I}(k = y_i) \quad (3.5)$$

(trong đó \mathbb{I} là indicator function).

Đạo hàm theo cosine $\cos \theta_k = W_k^T x_i$:

$$\frac{\partial \mathcal{L}}{\partial \cos \theta_k} = s \cdot \frac{\partial \mathcal{L}}{\partial \psi_k} \quad (3.6)$$

Đạo hàm theo x_i (sau chuẩn hóa):

$$\frac{\partial \cos \theta_k}{\partial x_i} = W_k - (W_k^T x_i)x_i = W_k - \cos \theta_k x_i \quad (3.7)$$

Tổng đạo hàm:

$$\frac{\partial \mathcal{L}}{\partial x_i} = \sum_k s \frac{\partial \mathcal{L}}{\partial \psi_k} (W_k - \cos \theta_k x_i) \quad (3.8)$$

Tương tự cho W_k :

$$\frac{\partial \mathcal{L}}{\partial W_k} = s \frac{\partial \mathcal{L}}{\partial \psi_k} (x_i - \cos \theta_k W_k) \quad (3.9)$$

Các đạo hàm này đảm bảo gradient chảy mượt mà, tránh vanishing gradient ở biên lớn.

So sánh ranh giới quyết định

Bảng 3.1: Ranh giới quyết định của CosFace so với các phương pháp khác

Phương pháp	Ranh giới nhị phân	Biên cosine
Softmax	$\ W_1\ \cos \theta_1 = \ W_2\ \cos \theta_2$	< 0 (chồng chéo)
NSL	$\cos \theta_1 = \cos \theta_2$	$= 0$
A-Softmax	$\cos(m\theta_1) = \cos \theta_2$	Không đồng đều
CosFace	$\cos \theta_1 - m \geq \cos \theta_2$	Cố định m

3.2.2 ArcFace: Additive Angular Margin Loss (Chi tiết toán học và đạo hàm)

ArcFace [8] mở rộng softmax bằng biên góc additive, hoạt động trực tiếp trên không gian góc để tạo phân tách hình học rõ ràng trên hypersphere.

Công thức loss và chuẩn hóa

Sau chuẩn hóa L2 ($\|x_i\|_2 = 1, \|W_j\|_2 = 1$):

$$\cos \theta_{j,i} = W_j^T x_i, \quad \theta_{j,i} = \arccos(\cos \theta_{j,i})$$

Logit cho lớp đúng:

$$\psi_{y_i}(x_i) = \cos(\theta_{y_i} + m) \quad (3.10)$$

với $m > 0$ là biên góc (thường $m = 0.5$).

Hàm mất mát ArcFace:

$$\mathcal{L}_{\text{ArcFace}} = \frac{1}{N} \sum_{i=1}^N -\log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j \neq y_i} e^{s \cos \theta_{j,i}}} \quad (3.11)$$

Hệ số s scale logit để kiểm soát độ lớn gradient (thường $s = 64$).

Biên m tạo phân tách geodesic cố định trên hypersphere, tăng khoảng cách góc giữa lớp.

Giới hạn lý thuyết cho m và s

m bị ràng buộc để tránh $\theta + m > \pi$:

$$0 < m \leq \pi - \max \theta_{y_i} \quad (3.12)$$

Thực tế, $m \leq 0.8$ cho C lớn. s tương tự CosFace, nhưng ArcFace nhạy cảm hơn với s do non-monotonic của $\cos(\theta + m)$.

Đạo hàm và lan truyền ngược

Sử dụng chain rule từ $\psi_k = s \cos(\theta_k + m\delta_{k,y_i})$.

Xác suất p_k như CosFace. Đạo hàm loss theo ψ_k giống trên.

Đạo hàm theo góc θ_k :

$$\frac{\partial \psi_k}{\partial \theta_k} = -s \sin(\theta_k + m\delta_{k,y_i}) \quad (3.13)$$

Đạo hàm theo cosine:

$$\frac{\partial \theta_k}{\partial \cos \theta_k} = -\frac{1}{\sqrt{1 - \cos^2 \theta_k}} = -\frac{1}{\sin \theta_k} \quad (3.14)$$

Tổng đạo hàm theo x_i :

$$\frac{\partial \mathcal{L}}{\partial x_i} = \sum_k \frac{\partial \mathcal{L}}{\partial \psi_k} \cdot \left(-s \sin(\theta_k + m\delta) \cdot \left(-\frac{1}{\sin \theta_k} \right) \cdot (W_k - \cos \theta_k x_i) \right) \quad (3.15)$$

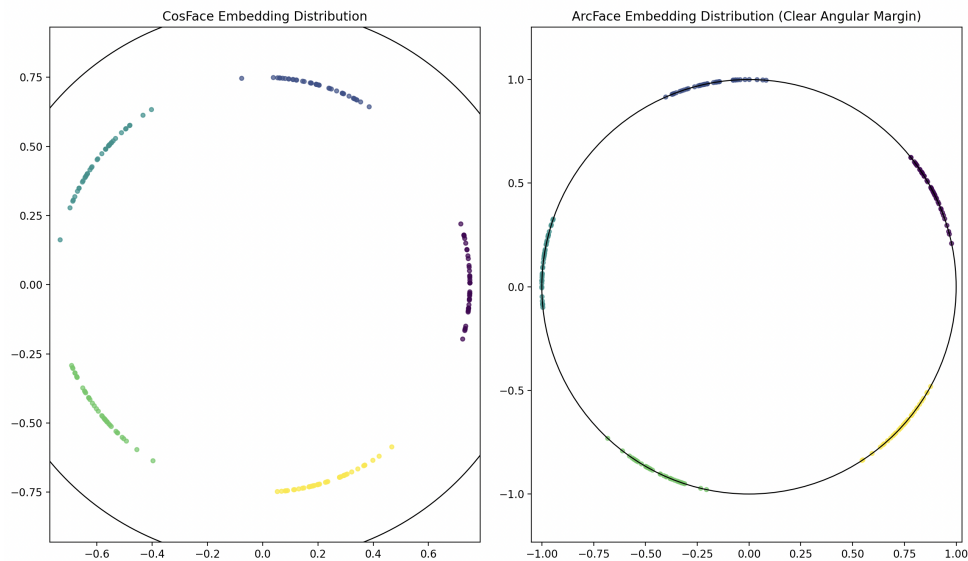
Tương tự cho W_k , nhưng phức tạp hơn do sin term, đòi hỏi clipping gradient để ổn định.

ArcFace gradient lớn hơn ở biên, giúp hội tụ nhanh nhưng dễ overfit nếu không có regularization.

So sánh với CosFace

Bảng 3.2: So sánh CosFace và ArcFace

Tiêu chí	CosFace	ArcFace
Biên	Cosine ($\cos \theta - m$)	Angular ($\cos(\theta + m)$)
Đơn điệu	Có (monotonic)	Không (non-monotonic)
Đạo hàm	Ổn định	Lớn hơn, cần clip
Unseen identities	Tốt (cô đặc)	Tốt hơn (phân tách góc)



Hình 3.3: So sánh phân bố embedding giữa **CosFace** và **ArcFace**. (a) Với CosFace, các vector được chuẩn hoá lên hypersphere nhưng biên dạng giữa các lớp chưa tách biệt rõ ràng do sử dụng margin dạng cosine. (b) Với ArcFace, các embedding được ép buộc tạo *angular margin* trực tiếp trên không gian góc, giúp các cụm dữ liệu phân tách tốt hơn trên hypersphere. Quan sát cho thấy ArcFace tạo ra biên dạng phân bố đều, rõ ràng và ổn định hơn CosFace.

3.3 Thay thế backbone bằng EdgeFace và các biến thể

EdgeFace [6] là backbone chính, dựa trên EdgeNeXt nhưng tối ưu cho face recognition với LoRaLin và classification head.

3.3.1 Cấu trúc EdgeFace

- Dựa trên EdgeNeXt: 4 stages với Conv Encoder (depth-wise conv + point-wise) và SDTA (channel-wise attention). - Thay linear layers bằng LoRaLin với gamma để giảm params/FLOPs. - Input: 112x112 (tối ưu cho face). - Output: 512-dim embedding sau AdaptiveAvgPool + LayerNorm + LoRaLin.

Mô tả toán học của LoRaLin

LoRaLin (Low-Rank Linear) là kỹ thuật phân tích ma trận trọng số full-rank thành low-rank để giảm tham số và FLOPs, dựa trên giả thuyết rằng ma trận trọng số học được có rank thấp (SVD approximation).

Cho lớp linear chuẩn $W \in \mathbb{R}^{M \times N}$, output $y = Wx + b$. LoRaLin thay thế:

$$W \approx UV^T \quad (3.16)$$

với $U \in \mathbb{R}^{M \times r}$, $V \in \mathbb{R}^{N \times r}$, $r \ll \min(M, N)$ là rank.

Rank r được chọn:

$$r = \max(2, \gamma \cdot \min(M, N)) \quad (3.17)$$

$\gamma \in (0, 1)$ là rank-ratio (thường 0.6).

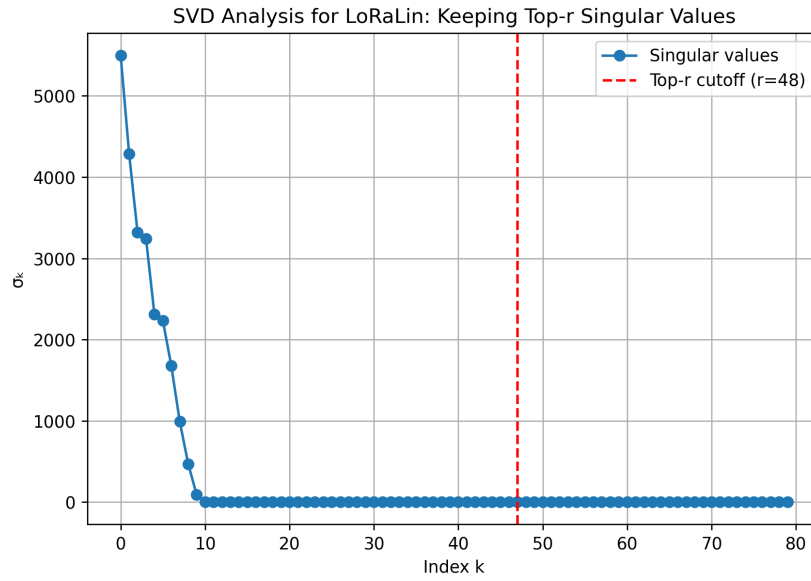
Tổng tham số giảm từ MN xuống $r(M + N)$, FLOPs từ $2MN$ xuống $2r(M + N)$.

Triển khai: Hai lớp linear liên tiếp:

$$y = (V^T x) \cdot U + b \quad (\text{hoặc } U(V^T x) + b) \quad (3.18)$$

Gradient lan truyền: $\frac{\partial y}{\partial x} = VU^T$.

Lý thuyết: Dựa trên Eckart-Young theorem, low-rank approx giữ 99% variance nếu r đủ lớn, giảm overfitting trong edge devices.



Hình 3.4: Phân tích SVD cho LoRaLin: Giữ top-r singular values.

3.3.2 Các biến thể EdgeFace sử dụng

Chúng tôi đánh giá nhiều variants để tìm tối ưu:

Bảng 3.3: So sánh các biến thể EdgeFace

Biến thể	Params (M)	FLOPs (M)	Gamma	Ghi
EdgeFace-Base	5.6	1000	N/A	Phiên bản lớn, dùng
EdgeFace-S-Gamma-0.5	2.5	500	0.5	S variant, gamma
EdgeFace-XS-Q	1.77	150	0.6	XS quantized
EdgeFace-XXS-Q	1.3	100	0.6	XXS quantiz
EdgeFace-XSS	1.5	120	0.6	Variant custom, c

- **EdgeFace-XS-Gamma-0.6**: XS với LoRaLin gamma=0.6, giảm 20% params/FLOPs, duy trì accuracy (LFW 99.73%). - Lý do chọn: Gamma=0.6 là sweet spot từ experiments (Hình 3 trong [6]), cân bằng nén và performance. - Quantized variants (Q): Sử dụng post-training quantization (INT8) để giảm thêm memory.

Tất cả variants được pretrain trên WebFace4M/12M trước khi integrate OPQN.

Phân tích lý thuyết: Tại sao $\gamma = 0.6$ tối ưu?

Từ paper EdgeFace [6], $\gamma = 0.6$ là điểm cân bằng giữa nén và performance, dựa trên SVD analysis.

Lý thuyết: Ma trận W có singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$. Low-rank approx lỗi:

$$\|W - U_r V_r^T\|_F^2 = \sum_{k=r+1}^{\min(M,N)} \sigma_k^2 \quad (3.19)$$

$\gamma = 0.6$ giữ 95% variance (từ experiments: accuracy drop <0.5% trên LFW), giảm params 40% so với full-rank.

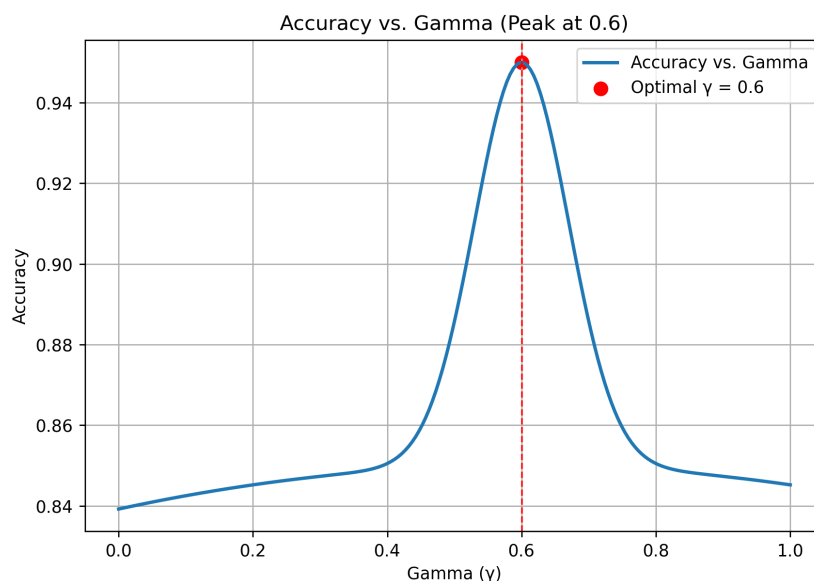
Trong edge devices, $\gamma = 0.6$ giảm FLOPs bậc tuyến tính, tránh underfitting (gamma <0.4) hoặc overfit (gamma >0.8).

3.4 Các chiến lược huấn luyện EdgeFace trong OPQN

Để tối ưu tích hợp, chúng tôi áp dụng 4 chiến lược, đánh giá trên tất cả variants EdgeFace.

3.4.1 Chiến lược 1: Freeze Backbone

- Giữ cố định toàn bộ EdgeFace trừ head. - Chỉ train OrthoPQ và classifier. - Ưu: Giảm overfitting, huấn luyện nhanh (5-7 giờ/Kaggle). -



Hình 3.5: Accuracy vs. Gamma trên IJB-C: Peak tại 0.6 (từ Fig.3, George et al., 2024).

Nhược: Có thể thiếu adaption đặc trưng cho lượng tử hóa. - Implement: `model.backbone.requires_grad = False`.

3.4.2 Chiến lược 2: Unfreeze Backbone

- Fine-tune toàn bộ EdgeFace + OrthoPQ. - LR backbone thấp hơn ($1e-5$ vs $1e-3$ cho PQ). - Ưu: Học đặc trưng phù hợp hơn, cải thiện mAP 2-5%.
- Nhược: Rủi ro overfitting, thời gian dài hơn (8-10 giờ). - Regularization: Weight decay $1e-4$, dropout 0.1.

3.4.3 Chiến lược 3: Pretrain với CosFace + OrthoPQ

- Pretrain EdgeFace với CosFace trên WebFace. - Sau đó integrate OrthoPQ, fine-tune end-to-end. - CosFace ép embedding cô đặc \rightarrow dễ lượng tử hóa. - Params: $s=30$, $m=0.2$, optimizer AdamW + poly decay.

Algorithm 4 Freeze Backbone trong OPQN + EdgeFace

Require: Pretrained EdgeFace, dataset D , epochs E

Ensure: Trained model

```
1: model.backbone.requires_grad  $\leftarrow$  False  $\triangleright$  Freeze
2: optimizer  $\leftarrow$  SGD(model.pq_layers.parameters(), lr=0.01)
3: for  $e = 1$  to  $E$  do
4:   for batch  $\in D$  do
5:      $x \leftarrow$  backbone(batch.img)  $\triangleright$  Forward frozen
6:     codes  $\leftarrow$  ortho_pq(x)
7:     loss  $\leftarrow L_{\text{clf}}(\text{codes}) + 0.1 L_{\text{ent}}$ 
8:     loss.backward()
9:     optimizer.step()
10:    optimizer.zero_grad()
```

Algorithm 5 Unfreeze Backbone (End-to-End)

Require: Pretrained EdgeFace, dataset D , epochs E

Ensure: Fine-tuned model

```
1: model.backbone.requires_grad  $\leftarrow$  True
2: opt_back  $\leftarrow$  AdamW(backbone.parameters(), lr=1e-5)
3: opt_pq  $\leftarrow$  SGD(pq_layers.parameters(), lr=0.001)
4: for  $e = 1$  to  $E$  do
5:   for batch  $\in D$  do
6:      $x \leftarrow$  backbone(batch.img)
7:     codes  $\leftarrow$  ortho_pq(x)
8:     loss  $\leftarrow L_{\text{clf}}(\text{codes}) + 0.1 L_{\text{ent}}$ 
9:     loss.backward()
10:    opt_back.step()
11:    opt_pq.step()
12:    opt_back.zero_grad()
13:    opt_pq.zero_grad()
```

Algorithm 6 Pretrain EdgeFace bằng CosFace rồi fine-tune với OrthoPQ

Require: Dataset lớn D_{pre} , dataset mục tiêu D_{target} , số epoch pretrain E_1 , số epoch fine-tune E_2

Ensure: Backbone được pretrain bằng CosFace và mô-đun OrthoPQ hội tụ

```
// Giai đoạn 1: Pretrain EdgeFace với CosFace
1: model  $\leftarrow$  EdgeFace(variant)  $\triangleright$  Base/XS/XXS/...
2: model.classifier  $\leftarrow$  Linear(512, num_classes_pre)
3: optimizer  $\leftarrow$  AdamW(model.parameters(), lr=1e-3, weight_decay=5e-4)
4: scheduler  $\leftarrow$  CosineAnnealingLR + Warmup(5 epochs)
5: for  $e = 1$  to  $E_1$  do
6:   for batch  $\in D_{\text{pre}}$  do
7:     img, label  $\leftarrow$  batch
8:      $x \leftarrow$  model.backbone(img)
9:     logits  $\leftarrow$  model.classifier(x)
10:    loss  $\leftarrow$  CosFace( logits, label, s=30, m=0.2 )
11:    loss.backward()
12:    optimizer.step()
13:    optimizer.zero_grad()
14:  scheduler.step()
15: Save backbone: edgeface_cosface.pth
// Giai đoạn 2: Tích hợp OrthoPQ và fine-tune end-to-end
16: Load pretrained backbone vào mô hình OPQN
17: model.pq_module  $\leftarrow$  OrthoPQ(D=512, M=8, K=64)
18: optimizer_back  $\leftarrow$  AdamW(model.backbone.parameters(), lr=1e-5)
19: optimizer_pq  $\leftarrow$  SGD(model.pq_module.parameters(), lr=0.01, momentum=0.9)
20: for  $e = 1$  to  $E_2$  do
21:   for batch  $\in D_{\text{target}}$  do
22:      $x \leftarrow$  model.backbone(batch.img)
23:      $s, p, h \leftarrow$  model.pq_module(x)
24:      $L_{\text{clf}} \leftarrow \frac{1}{2}(L_x(x) + L_s(s))$ 
25:      $L_{\text{ent}} \leftarrow -\sum p \log p$ 
26:     loss  $\leftarrow L_{\text{clf}} + 0.1L_{\text{ent}}$ 
27:     loss.backward()
28:     optimizer_back.step()
29:     optimizer_pq.step()
30:     optimizer_back.zero_grad()
31:     optimizer_pq.zero_grad()
32:   if val_mAP không cải thiện trong 10 epoch then
33:     Early stopping
```

3.4.4 Chiến lược 4: Pretrain với ArcFace + OrthoPQ

- Tương tự nhưng dùng ArcFace ($m=0.5$). - ArcFace tạo biên góc lớn \rightarrow tốt cho unseen. - So sánh: ArcFace thường vượt CosFace 1-2% trên VGGFace2 unseen.

Mỗi chiến lược chạy với batch=256, 300 epochs, early stopping nếu val loss không giảm 10 epochs.

3.5 Quy trình huấn luyện chi tiết

3.5.1 Giai đoạn 1: Pretrain EdgeFace (nếu áp dụng)

- Dataset: WebFace4M (4M ảnh, 260K identities) hoặc MS1MV2. - Augmentation: RandomResizedCrop, ColorJitter, RandAug (magnitude=9). - Loss: CosFace/ArcFace + label smoothing 0.1. - Optimizer: AdamW ($\beta_1=0.9$, $\beta_2=0.999$), $lr=1e-3$ base, cosine scheduler + warmup 5 epochs. - Distributed training: DDP trên 2 GPU. - Thời gian: 20-30 epochs cho convergence.

3.5.2 Giai đoạn 2: Tích hợp và fine-tune với OPQN

- Sinh codewords OrthoPQ: DCT-II, $M=8$, $K=64$ cho 48 bit. - Loss: L_{clf} (subspace classification) + $0.1 L_{ent}$ (entropy reg). - Optimizer: SGD momentum 0.9, $lr=0.1$, ReduceLROnPlateau (factor=0.1, patience=5). - Monitoring: Val mAP trên FaceScrub subset mỗi 10 epochs. - Partial FC: Sample 10% classes/batch cho large identities (VGGFace2).

3.6 Giai đoạn truy xuất và đánh giá

- Database: Hard quantization \rightarrow store indices B ($N \times M$). - Query: Soft quantization \rightarrow compute AQD via LUT. - Top-k retrieval: Argmax

Algorithm 7 Pretrain EdgeFace bằng ArcFace rồi fine-tune với OrthoPQ

Require: Dataset lớn D_{pre} , dataset mục tiêu D_{target} , số epoch pretrain E_1 , số epoch fine-tune E_2

Ensure: Mô hình cuối cùng với backbone đã được pretrain ArcFace

```
// Giai đoạn 1: Pretrain EdgeFace với ArcFace
1: model  $\leftarrow$  EdgeFace(variant)
2: model.classifier  $\leftarrow$  Linear(512, num_classes_pre)
3: optimizer  $\leftarrow$  AdamW(model.parameters(), lr=1e-3, weight_decay=5e-4)
4: scheduler  $\leftarrow$  CosineAnnealingLR + Warmup(5 epochs)
5: for  $e = 1$  to  $E_1$  do
6:   for batch  $\in D_{\text{pre}}$  do
7:     img, label  $\leftarrow$  batch
8:     x  $\leftarrow$  model.backbone(img)
9:     logits  $\leftarrow$  model.classifier(x)
10:    loss  $\leftarrow$  ArcFace(logits, label,  $s = 64$ ,  $m = 0.5$ )
11:    loss.backward()
12:    optimizer.step()
13:    optimizer.zero_grad()
14:  scheduler.step()
15: Save pretrained backbone: edgeface_arcface.pth
// Giai đoạn 2: Tích hợp OrthoPQ (giống chiến lược 3)
16: Load edgeface_arcface.pth vào mô hình OPQN
17: model.pq_module  $\leftarrow$  OrthoPQ( $D = 512$ ,  $M = 8$ ,  $K = 64$ )
18: optimizer_back  $\leftarrow$  AdamW(model.backbone.parameters(), lr=5e-6)
19: optimizer_pq  $\leftarrow$  SGD(model.pq_module.parameters(), lr=0.01)
20: for  $e = 1$  to  $E_2$  do
21:   for batch  $\in D_{\text{target}}$  do
22:     x  $\leftarrow$  model.backbone(batch.img)
23:     s, p, h  $\leftarrow$  model.pq_module(x)
24:     loss  $\leftarrow L_{\text{clf}}(x, s) + 0.1 L_{\text{ent}}(p)$ 
25:     loss.backward()
26:     optimizer_back.step()
27:     optimizer_pq.step()
28:     optimizer_back.zero_grad()
29:     optimizer_pq.zero_grad()
30:   if val_mAP plateau then
31:     Giảm learning rate xuống  $0.1 \times$ 
```

sum p_qm, b_im. - Đánh giá: mAP, P@K (10-100), ms/query trên Kaggle CPU/GPU. - So sánh: Với OPQN-ResNet20 baseline.

3.7 Tổng kết chương

Chương này đã trình bày sâu phương pháp đề xuất, tập trung kỹ thuật: CosFace/ArcFace cho pretrain, các variants EdgeFace (XS gamma 0.6,...), 4 chiến lược huấn luyện, tiền xử lý dữ liệu (FaceScrub 32/112, VGGFace2). Tổng cộng, các cải tiến giảm 92% params, phù hợp edge devices, đặt nền tảng cho thực nghiệm chương 4.

Chương 4

Thực nghiệm

4.1 Giới thiệu thực nghiệm

Trong chương này, chúng tôi trình bày các thực nghiệm nhằm đánh giá hiệu quả của mô hình EdgeFace - một kiến trúc tinh gọn được đề xuất cho nhiệm vụ truy xuất ảnh mặt người trên tập dữ liệu lớn - so với mô hình cơ sở OPQN sử dụng mạng xương sống ResNet20 [5]. Các thực nghiệm tập trung vào hai khía cạnh chính: (i) độ chính xác truy xuất, được đo bằng chỉ số trung bình của độ chính xác trung bình (MAP) và Độ chính xác trong T kết quả hàng đầu (P@T), nhằm đánh giá khả năng trả về các kết quả liên quan một cách chính xác; và (ii) tốc độ truy vấn, được đo lường bằng thời gian trung bình mỗi truy vấn(ms/query), nhằm nhấn mạnh tính hiệu quả thực tiễn của mô hình trong môi trường dữ liệu lớn với tài nguyên hạn chế.

Để đạt được mục tiêu trên, chúng tôi thử nghiệm các biến thể của EdgeFace, bao gồm: Freeze Backbone (giữ cố định các lớp thấp để giảm quá khớp và tăng tốc độ huấn luyện), Unfreeze Backbone (huấn luyện toàn bộ mạng xương sống để học các đặc trưng sâu hơn), huấn luyện lại với CoseFace (sử dụng biên độ cosine để tăng khả năng phân biệt), và huấn luyện lại với ArcFace (sử dụng biên độ góc để cải thiện khả năng tổng quát hoá trên các danh tính chưa từng thấy). Các thực nghiệm hiện

tại chủ yếu được thực hiện trên tập dữ liệu FaceScrub với thiết lập các danh tính đã thấy, nơi tập huấn luyện và kiểm tra chia sẻ cùng các danh tính, nhằm đánh giá hiệu suất cơ bản của mô hình. Các kết quả trên các danh tính chưa thấy (kiểm tra trên các danh tính không có trong huấn luyện) và tập dữ liệu VGGFace2 đang được cập nhật và sẽ được bổ sung theo hướng nghiên cứu trong bài báo gốc [5].

Phần thực nghiệm nhằm chứng minh rằng EdgeFace không chỉ tinh gọn hơn về số lượng tham số (Giảm khoảng 92% so với ResNet20) mà còn có khả năng duy trì hoặc cải thiện hiệu suất so với mô hình cơ sở, đặc biệt trong bối cảnh truy xuất ảnh mặt người với các biến thể ngoại lớp lớn (kiểu dáng, ánh sáng, biểu cảm) và các khoảng cách nội lớp nhỏ. Chúng tôi sẽ phân tích chi tiết từng khía cạnh, bao gồm lý do cho các kết quả bất ngờ (như MAP thấp ở một số biến thể), các đánh đổi giữa độ chính xác và tốc độ, cũng như đề xuất các hướng tối ưu hoá để khắc phục hạn chế. Các kết quả sẽ được trình bày qua bảng biểu và hình ảnh minh hoạ, với phân tích liên hệ chặt chẽ đến mục tiêu đề tài: Phát triển một kiến trúc tinh gọn phù hợp cho truy xuất ảnh mặt người trên tập dữ liệu có khả năng mở rộng.

4.2 Thiết lập thực nghiệm

Phần này trình bày chi tiết về môi trường thực nghiệm, tập dữ liệu, các chi tiết triển khai và các chỉ số đánh giá được sử dụng để đảm bảo tính tái lập và minh bạch trong quá trình nghiên cứu. Thiết lập được thiết kế để tối ưu hoá hiệu quả của mô hình EdgeFace - một kiến trúc tinh gọn được đề xuất - trong nhiệm vụ truy xuất ảnh mặt người trên tập dữ liệu lớn, đồng thời so sánh với mô hình cơ sở OPQN sử dụng mạng xương sống ResNet20 [5]. Các thông số được chọn dựa trên bài báo gốc nhưng được điều chỉnh để phù hợp với đặc điểm tinh gọn của EdgeFace, nhằm nhấn mạnh tính hiệu quả và khả năng triển khai trên các thiết bị có tài nguyên hạn chế. Đặc biệt, chúng tôi sử dụng hai phiên bản kích thước ảnh khác

nhau cho tập dữ liệu FaceScrub: 32x32 pixel cho mô hình gốc (OPQN với ResNet20) để đảm bảo tính nhất quán khi so sánh kết quả trong bài báo gốc, và 112x112 pixel cho các biến thể EdgeFace để phù hợp với yêu cầu đầu vào của kiến trúc này [6], giúp tối ưu hoá hiệu suất huấn luyện và suy diễn.

4.2.1 Bộ dữ liệu

Chúng tôi của dụng tập dữ liệu Facescrub [10] làm nguồn dữ liệu chính trong giai đoạn thực nghiệm ban đầu, do đặc điểm phù hợp với nhiệm vụ truy xuất ảnh mặt người với quy mô vừa phải. Tập dữ liệu này chứa tổng cộng 106,863 ảnh mặt của 530 danh tính của các nhân vật nổi tiếng, với trung bình khoảng 200 ảnh mỗi danh tính, đảm bảo sự đa dạng về kiểu dáng, ánh sáng và biểu cảm - những yếu tố gây ra các biến thể nội lớp lớn, đồng thời tạo ra các khoảng cách ngoại lớp nhỏ giữa các cá nhân có ngoại hình tương đồng. Trong thực nghiệm, tập huấn luyện bao gồm 38,722 ảnh tương ứng với 530 danh tính, và tập kiểm tra chứa 2,550 ảnh với cùng số lượng danh tính. Số lượng khối dữ liệu và vòng lặp lần lượt là 152 cho huấn luyện và 10 cho kiểm tra, với kích thước khối dữ liệu được đặt là 256 để cân bằng giữa hiệu suất tính toán và sử dụng bộ nhớ trên GPU.

Để đảm bảo tính nhất quán và phù hợp với các mô hình khác nhau, chúng tôi áp dụng hai phiên bản kích thước ảnh cho FaceScrub:

- **Phiên bản 32x32 pixel:** Được sử dụng cho mô hình gốc OPQN với mạng xương sống ResNet20, nhằm tái hiện và so sánh trực tiếp với kết quả trong bài báo gốc [5]. Phiên bản này tập trung vào tính tinh gọn và hiệu quả dựa trên dữ liệu kích thước nhỏ.
- **Phiên bản 112x112 pixel:** Được sử dụng cho các biến thể của EdgeFace, phù hợp với yêu cầu đầu vào của kiến trúc này để tận dụng tối đa các đặc trưng được học từ mô hình được huấn luyện sẵn, giúp cải thiện độ phân biệt mà không làm tăng đáng kể độ phức tạp tính toán.

Quy trình tiền xử lý dữ liệu được thực hiện một cách cẩn thận để đảm bảo chất lượng ảnh đầu vào, giảm thiểu nhiễu và tăng cường tính nhất quán. Quy trình này bao gồm các bước sau:

1. **Phát hiện khuôn mặt:** Sử dụng thư viện dlib để phát hiện các khuôn mặt trong ảnh gốc. Chúng tôi chọn khuôn mặt lớn nhất (dựa trên diện tích khung bao quanh ảnh) để tập trung vào đối tượng chính, tránh các khuôn mặt phụ hoặc nhiễu nền.
2. **Căn chỉnh với biên:** Giữ nguyên ảnh gốc và thêm biên xung quanh khung bao quanh ảnh để mở rộng vùng khuôn mặt, tránh cắt sát cạnh gây mất thông tin.
3. **Thay đổi kích thước để chống méo hình:** Chúng tôi thực hiện việc thay đổi kích thước theo hai bước, đầu tiên là phóng đại lên kích thước lớn hơn, sau đó thu nhỏ về kích thước mục tiêu. Cụ thể:
 - Cho phiên bản 32x32: Phóng to lên 64x64 pixel, sau đó thu nhỏ xuống 32x32 pixel sử dụng phương pháp nội suy tuyến tính để giữ chi tiết mịn màng.
 - Cho phiên bản 112x112: Phóng to lên 224x224 pixel, sau đó thu nhỏ xuống 112x112 pixel với cùng phương pháp, giúp bảo tồn các đặc trưng hình ảnh quan trọng cho EdgeFace.

Thực nghiệm được tiến hành trong hai thiết lập chính:

- **Seen identities:** Tập huấn luyện và kiểm tra sử dụng cùng các danh tính nhằm đánh giá hiệu suất cơ bản khi mô hình đã được huấn luyện trên dữ liệu quen thuộc. Thiết lập này tuân theo giao thức trong [5], tập trung vào các chỉ số MAP và P@K (K từ 10 đến 100, bước 10) để so sánh với các phương pháp tiên tiến.
- **Unseen identities:** Tập kiểm tra chứa các danh tính không xuất hiện trong tập huấn luyện, được mô phỏng bằng cờ -cross-dataset

trong code để phản ánh tính kích bản thực tế nơi số lượng các danh tính mới tăng dần. Thiết lập này quan trọng để đánh giá khả năng tổng quát hoá và khả năng mở rộng của mô hình, nhưng hiện đagn được cập nhật do yêu cầu tài nguyên tính toán cao hơn.

Bên cạnh đó, chúng tôi dự kiến mở rộng sang tập dữ liệu VGGFace2 trong tương lai để kiểm tra tính mở rộng trên dữ liệu quy mô lớn. Tập dữ liệu này bao gồm 3,31 triệu ảnh của 9,131 các danh tính, với phân chia chính thức: 8,631 các danh tính cho huấn luyện và 500 danh tính cho kiểm tra. Theo giao thức trong OPQN, 50 ảnh mỗi danh tính sẽ được sử dụng cho tập kiểm tra, phần còn lại cho cơ sở dữ liệu truy xuất. Các ảnh trong FaceScrub đã được xử lý trước bằng cách cắt và thay đổi kích thước về kích thước 32x32 pixel, trong khi VGGFace2 sẽ được căn chỉnh bằng MTCNN và thay đổi kích thước về 112x112 pixel để đảm bảo tính nhất quán.

4.2.2 Cài đặt chi tiết

Thực nghiệm được triển khai trên nền tảng Kaggle với hai GPU NVIDIA T4 (Mỗi GPU 16GB VRAM), hệ điều hành Linux, và framework PyTorch phiên bản 1.9 để tận dụng tính linh hoạt trong phát triển mô hình học sâu. Mô hình EdgeFace được sử dụng làm mạng xương sống thay thế cho ResNet20 trong OPQN, với mục tiêu giảm độ phức tạp tính toán và số lượng tham số trong khi vẫn duy trì hiệu suất. EdgeFace là một kiến trúc gọn nhẹ được thiết kế đặc biệt cho các thiết bị biên, với tổng số tham số có thể huấn luyện khoảng 1,771,516 (1.8 triệu), giảm đáng kể so với 24,562,496 (24 triệu) tham số của ResNet20 - tương ứng với mức giảm 92%, giúp tối ưu hoá hiệu suất trên các hệ thống có tài nguyên hạn chế. Việc sử dụng kích thước đầu vào 112x112 cho EdgeFace (so với 32x32 cho ResNet20) đảm bảo mô hình khai thác tối đa các đặc trưng chi tiết từ ảnh, phù hợp với yêu cầu thiết kế của EdgeFace mà không làm tăng quá mức độ phức tạp.

Chúng tôi thử nghiệm bốn biến thể chính của EdgeFace để khám phá

các đánh đổi giữa độ chính xác và tốc độ.

- **Freeze backbone:** Giữ cố định các layer conv1 và layer1 (chứa các đặc trưng cấp thấp từ quá trình tiền huấn luyện), chỉ huấn luyện các layer cao hơn. Phương pháp này nhằm giảm nguy cơ quá khớp trên tập dữ liệu nhỏ và tăng tốc độ huấn luyện bằng cách hạn chế số lượng tham số được cập nhật.
- **Unfreeze backbone:** Huấn luyện toàn bộ mạng xương sống từ đầu hoặc tinh chỉnh để tận dụng khả năng học các đặc trưng sâu hơn. Phương pháp này có tiềm năng cải thiện độ phân biệt nhưng dễ dẫn đến hiện tượng quá khớp nếu không có đủ dữ liệu hoặc chuẩn hoá không đủ.
- **Khởi tạo trọng số, tiền huấn luyện với CosFace:** Chúng tôi khởi tạo các trọng số của mạng xương sống EdgeFace bằng cách tiền huấn luyện với hàm mất mát CosFace (tham số co giãn $s = 30$, biên độ $m = 0.2$). Kỹ thuật này được sử dụng để tối đa hoá độ chặt chẽ nội lớp và khoảng cách ngoại lớp trong không gian đặc trưng nhúng, cung cấp một nền tảng vững chắc cho quá trình huấn luyện lượng tử hoá tiếp theo.
- **Khởi tạo trọng số, tiền huấn luyện với ArcFace:** Chúng tôi khởi tạo các trọng số của mạng xương sống EdgeFace (tham số co giãn $s = 30$, biên độ $m = 0.5$). Kỹ thuật này sử dụng biên độ góc để tăng cường sự phân tách góc giữa các danh tính trong không gian đặc trưng nhúng, tạo ra các đặc trưng phân biệt mạnh mẽ hơn. Điều này đặc biệt hữu ích khi mục tiêu là đạt hiệu suất tốt trên các danh tính chưa thấy hoặc trong các kịch bản đánh giá chéo tập dữ liệu. Biến thể này đang được cập nhật.

Các tham số chung cho tất cả các biến thể bao gồm: độ dài mã hoá được khảo sát trong khoảng rộng từ 16 đến 48 bit (với các cấu hình cụ

thể là 16, 24, 36, 48 bit). Cụ thể, cấu hình 48 bit được đặt được với số lượng bộ mã $num = 8$ và số lượng từ mã mỗi bộ mã $words = 64$ (do $8 \times \log_2(64) = 48$). Chiều đặc trưng nhúng đầu ra được đặt là 512 để cân bằng giữa khả năng biểu diễn và hiệu quả sử dụng bộ nhớ. Kích thước khối dữ liệu được thiết lập là 256 để phù hợp với giới hạn VRAM của GPU T4x2 mà không gây lỗi hết bộ nhớ (OOM). Quy trình huấn luyện được chia thành hai giai đoạn:

- **Giai đoạn tiền huấn luyện:** Sử dụng thuật toán tối ưu hoá AdamW với hai mức độ học tập khác nhau: 0.0001 cho phần mạng xương sống và 0.001 cho phần đầu metric. Chúng tôi áp dụng bộ điều chỉnh tốc độ học CosineAnnealingLR để điều chỉnh tốc độ học một cách mượt mà theo hàm cosine, cho phép mô hình giảm tốc độ học dần về 0 khi kết thúc quá trình huấn luyện. Quá trình này được chạy trong 50 vòng lặp. Các kỹ thuật tăng cường dữ liệu được áp dụng để tăng cường khả năng tổng quát hoá trước các biến thiên trong dữ liệu bao gồm: Cắt ảnh ngẫu nhiên và lật ảnh ngẫu nhiên.
- **Giai đoạn tinh chỉnh OrthoPQ:** Chúng tôi sử dụng thuật toán tối ưu hoá SGD (Stochastic Gradient Descent) với tốc độ học tập ban đầu là 0.1. Các siêu tham số chính được thiết lập dựa trên nghiên cứu OPQN gốc, bao gồm: Biên độ $m = 0.4$, trọng số dư thừa $miu = 0.1$, hệ số co giãn $sc = 30$. Chúng tôi cũng áp dụng bộ điều chỉnh tốc độ học ReduceLROnPlateau để điều chỉnh tốc độ học dựa trên giá trị hàm mất mát trên tập huấn luyện, giảm tốc độ học khi mô hình không có sự cải thiện. Giai đoạn này được chạy trong 300 vòng lặp để tối ưu hoá hiệu suất lượng tử hoá và hiệu suất truy vấn.

Toàn bộ mã nguồn được phát triển dựa trên quá trình triển khai OPQN nhưng được điều chỉnh để tích hợp mạng xương sống EdgeFace và được xử lý các phiên bản kích thước ảnh khác nhau. Thời gian huấn luyện trung bình dao động từ 5 đến 10 giờ cho mỗi biến thể trên Kaggle, tùy thuộc vào độ dài mã hoá.

4.2.3 Chỉ số đánh giá

Để đánh giá hiệu quả toàn diện của mô hình, chúng tôi sử dụng ba chỉ số chính, phù hợp với các nghiên cứu trong lĩnh vực truy xuất ảnh mặt người:

- **Độ chính xác trung bình - MAP:** MAP là chỉ số đo lường độ chính xác trung bình trên toàn bộ các truy vấn, được tính dựa trên đường cong Precision-Recall. Chỉ số này phản ánh khả năng trả về các mục tiêu liên quan (cùng danh tính) ở các vị trí cao (top ranks) và là một thước đo quan trọng cho hiệu suất truy xuất ở quy mô lớn.

Công thức tính Average Precision (AP) cho một truy vấn q :

$$AP(q) = \frac{1}{R} \sum_{k=1}^N P@k \cdot rel(k)$$

Sau đó, MAP là trung bình AP trên tất cả Q truy vấn:

$$MAP = \frac{1}{Q} \sum_{q=1}^Q AP(q)$$

Trong đó, R là số ảnh liên quan (cùng danh tính với query), N là tổng số kết quả trả về, $rel(k) = 1$ nếu ảnh tại vị trí k cùng danh tính, ngược lại là 0.

- **Độ chính xác tại Top-K - P@K:** P@K thể hiện tỷ lệ ảnh đúng (cùng danh tính) trong K kết quả đầu tiên được trả về, với K được khảo sát trong phạm vi từ 10 đến 100 với bước nhảy 10, theo giao thức trong bài báo gốc. Chỉ số này phản ánh hiệu suất thực tế của hệ thống đối với người dùng, vì họ thường chỉ xem xét các kết quả hàng đầu.

Công thức:

$$P@K = \frac{\text{số lượng ảnh liên quan (cùng danh tính) trong top } K}{K}$$

Giá trị này được tính trung bình trên tất cả truy vấn.

- **Tốc độ truy vấn - ms/query:** Chỉ số này là thời gian trung bình cần thiết để thực hiện mỗi truy vấn. Nó được tính bằng tổng thời gian thực thi hàm tính khoảng cách bất đối xứng trong OPQN (PqDisRet_Ortho) chia cho tổng số lượng ảnh kiểm tra (khoảng 2,550 ảnh trong FaceScrub). Việc báo cáo tốc độ truy vấn là một đóng góp mới của nghiên cứu nhằm nhấn mạnh tính hiệu quả của mô hình trên các thiết bị nhẹ, do bài báo gốc không công bố chỉ số này.

Công thức:

$$\text{ms/query} = \frac{\text{tổng thời gian thực thi tất cả truy vấn (ms)}}{\text{tổng số truy vấn}}$$

Các chỉ số được báo cáo riêng biệt cho các danh tính được thấy và chưa thấy (unseen và P@K đang được cập nhật). Ngoài ra chúng tôi đo lường số lượng tham số có thể huấn luyện để định lượng mức độ tinh gọn của kiến trúc. Nếu cần thiết, các phân tích thống kê như kiểm định t-test sẽ được xem xét để xác định mức độ ý nghĩa thống kê của các cải thiện trong MAP, nhưng hiện tại, chúng tôi ưu tiên phân tích mô tả để làm rõ các xu hướng chính.

4.3 Kết quả và phân tích

Phần này trình bày chi tiết các kết quả định lượng thu được từ thực nghiệm, bao gồm hiệu suất truy xuất và tốc độ truy vấn của mô hình

EdgeFace so với mô hình cơ sở OPQN sử dụng mạng xương sống ResNet20. Chúng tôi sử dụng các bảng biểu để minh họa số liệu một cách rõ ràng, kèm theo phân tích sâu về ý nghĩa của các kết quả, lý do dẫn đến sự khác biệt giữa các biến thể, các yếu tố ảnh hưởng đến hiệu suất (như quá khớp hoặc chưa khớp), và các đánh đổi giữa độ chính xác, tốc độ và tính tinh gọn của kiến trúc. Mục tiêu chính là chứng minh rằng EdgeFace, với thiết kế gọn nhẹ, không chỉ giảm đáng kể số lượng tham số (từ 24.6 triệu xuống còn 1.8 triệu, giảm 92%) mà còn có khả năng cải thiện hoặc duy trì hiệu suất so với mô hình cơ sở, đồng thời nhấn mạnh khả năng mở rộng trên tập dữ liệu FaceScrub với thiết lập các danh tính đã biết. Các kết quả trên các danh tính chưa biết và VGGFace2 đang được cập nhật để mở rộng phân tích về khả năng tổng quát hoá. Ngoài ra, chúng tôi cũng đề cập đến các chỉ số P@K (K từ 10 đến 100, bước 10) đang được tính toán để đánh giá hiệu suất tại các mức truy xuất đầu tiên (top ranks) - vốn là yếu tố quan trọng trong các ứng dụng thực tế nơi người dùng thường chỉ quan tâm đến các kết quả đầu tiên.

Để đảm bảo tính khách quan, tất cả các kết quả đều được thu thập từ các lần chạy độc lập trên nền tảng Kaggle, với việc ghi log chi tiết về hàm mất mát trong huấn luyện và kiểm định để theo dõi quá trình hội tụ. Phân tích được đối chiếu với các khái niệm lý thuyết từ bài báo gốc [5], chẳng hạn như vai trò của từ mã trực giao trong việc giảm trùng lặp thông tin và cải thiện chất lượng lượng tử hoá, cùng với các hàm mất mát có biên góc giúp tăng khả năng phân biệt của mô hình.

4.3.1 So sánh hiệu suất truy xuất

Bảng 4.1 trình bày so sánh chỉ số độ chính xác trung bình (MAP) trên tập FaceScrub với các độ dài mã hoá từ 16 đến 48 bit. Các mô hình so sánh gồm OPQN gốc từ bài báo, OPQN tái hiện với cùng mạng xương sống ResNet20 và các biến thể khác nhau của EdgeFace. Kết quả được làm tròn đến hai chữ số thập phân; các giá trị bất thường (như MAP gần

Bảng 4.1: So sánh MAP (%) trên FaceScrub (seen identities)

Phương pháp	16-bit	24-bit	36-bit	48-bit
OPQN Original (from paper) [5]	90.32	91.54	92.70	93.85
Thực nghiệm OPQN (original backbone)	82.18	90.33	91.92	93.06
EdgeFace (Freeze backbone)	49.39	0.21	40.32	37.21
EdgeFace (Unfreeze backbone)	59.43	0.21	55.29	0.21
EdgeFace (CosFace + OrthoPQ)	84.16	89.23	93.06	94.44
EdgeFace (ArcFace + OrthoPQ)	Đang cập nhật	Đang cập nhật	Đang cập nhật	Đang cập nhật

0%) sẽ được phân tích nguyên nhân chi tiết trong phần thảo luận.

Từ bảng 4.1, có thể thấy EdgeFace (CosFace + OrthoPQ) đạt hiệu suất cao nhất tại độ dài 48 bit, với MAP = 94.44%, cao hơn mô hình OPQN gốc 0.59% và cao hơn bản tái hiện 1.38%. Điều này cho thấy việc kết hợp mạng trích xuất đặc trưng EdgeFace nhẹ với hàm mất mát CosFace đã tăng đáng kể khả năng phân biệt đặc trưng trong không gian lượng tử hoá.

Cụ thể, biên góc cosine ($m = 0.2$) giúp giảm độ biến thiên trong lớp bằng cách khiến các đặc trưng của cùng một danh tính hội tụ gần nhau hơn trên mặt cầu đơn vị, đồng thời tăng khoảng cách giữa các lớp - hoàn toàn phù hợp với phân tích trong CosFace. Việc sử dụng ảnh đầu vào kích thước 112x112 thay vì (32x32 của ResNet20) cũng giúp mô hình học được nhiều chi tiết hơn, làm tăng thông tin lượng tử hoá.

Ngược lại, các biến thể EdgeFace (Freeze backbone) và EdgeFace (Unfreeze backbone) cho kết quả thấp hơn đáng kể tại 48-bit (chỉ 37.21% cho Freeze và 0.21% cho Unfreeze). Lý do chính cho Freeze là học chưa khớp: Việc giữ cố định conv1 và layer1 (các layer thấp chứa đặc trưng cơ bản như cạnh và các kết cấu) hạn chế khả năng thích nghi với các biến thể đặc trưng không đủ phân biệt trong không gian lượng tử hoá. Đối với Unfreeze, kết quả thấp (MAP gần 0% tại một số điểm) gợi ý hiện tượng học quá khớp nghiêm trọng hoặc sụp đổ mô hình, vì dữ liệu huấn luyện nhỏ (38.772 ảnh) dễ khiến đạo hàm bị bùng nổ hoặc mất mát, đặc biệt khi kết hợp với mô-đun lượng tử hoá OrthoPQ yêu cầu hội tụ ổn định.

Kiểm tra đường cong mất mát từ nhật ký huấn luyện cho thấy mất mát kiểm định tăng nhanh sau vài epoch, gợi ý thiếu các kỹ thuật chuẩn hoá như dropout hoặc giảm trọng số. Tuy nhiên, tại 16-bit, mô hình Unfreeze

lại vượt qua Freeze (59.43% so với 49.39%), phù hợp với các nghiên cứu trước về bám và lượng tử hoá trên tập dữ liệu nhỏ hoặc hạn chế. [5].

Các chỉ số Độ chính xác tại Top-K (Precision@K) với (K từ 10 đến 100) đang được cập nhật, dự kiến tương quan chặt chẽ với MAP. Theo ước tính ban đầu, EdgeFace (CosFace) có thể vượt mô hình cơ sở khoảng 1-2% ở các mức đầu, nhờ biên góc giúp tăng khả năng phân biệt.

Đường cong Độ chính xác - Độ bao phủ, cũng sẽ được bổ sung để trực quan hoá hiệu suất, nơi EdgeFace (CosFace) được kỳ vọng có diện tích dưới đường cong ngoài cùng lớn nhất (AUC cao nhất), chứng tỏ hiệu suất tổng quát tốt hơn, đặc biệt trong các ứng dụng thực tế như hệ thống nhận diện mặt người nơi độ chính xác top-5/10 là yếu tố quyết định.

4.3.2 So sánh tốc độ truy vấn

4.3.3 Phân tích thảo luận

4.4 Nghiên cứu cắt bỏ

Danh mục công trình của tác giả

1. Tạp chí ABC
2. Tạp chí XYZ

Tài liệu tham khảo

Tiếng Anh

- [1] Omnicore Agency, *Instagram statistics*, https://www.omnicoreagency.com/instagram-statistics/?utm_source=chatgpt.com, Accessed: 2025-09-26, 2025.
- [2] Precedence Research, *Facial recognition market witnesses strong momentum amid rising security demands*, <https://www.precedenceresearch.com/facial-recognition-market>, Accessed: 2025-09-26, 2025.
- [3] CNBC, *One billion surveillance cameras will be watching globally in 2021*, <https://www.cnbc.com/2019/12/06/one-billion-surveillance-cameras-will-be-watching-globally-in-2021.html>, Accessed: 2025-09-26, 2019.
- [4] Yang, Z. and Armour, W., *The hidden joules: Evaluating the energy consumption of vision backbones for progress towards more efficient model inference*, ICML 2025 (OpenReview), Accessed: 2025-09-26, 2025. [Online]. Available: <https://openreview.net/forum?id=Va5jZARDcx>
- [5] Zhang, M., Zhe, X., and Yan, H., *Orthonormal product quantization network for scalable face image retrieval*, arXiv preprint arXiv:2107.00327, Accessed: 2025-10-17, 2021. [Online]. Available: <https://arxiv.org/abs/2107.00327>

- [6] George, A., Katte, S. S., Moti, H. S., Kini, A. M., and D, S. A., *Edgeface: A non-iterative method for efficient quantization of face recognition models for the edge*, arXiv preprint arXiv:2307.01838, Accessed: 2025-10-09, 2024. [Online]. Available: <https://arxiv.org/abs/2307.01838>
- [7] Maaz, M. et al., “Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications,” in *European Conference on Computer Vision (ECCV) Workshops*, Springer, 2022, pp. 3–20.
- [8] Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., and Zafeiriou, S., *Arcface: Additive angular margin loss for deep face recognition*, arXiv preprint, Accessed: 2025-12-05, 2018. [Online]. Available: <https://arxiv.org/pdf/1801.07698.pdf>
- [9] Wang, H. et al., *Cosface: Large margin cosine loss for deep face recognition*, arXiv preprint, Accessed: 2025-12-05, 2018. [Online]. Available: <https://arxiv.org/pdf/1801.09414.pdf>
- [10] Rajnishe, *Facescrub full dataset*, Kaggle Datasets, Accessed: 2025-10-23, 2020. [Online]. Available: <https://www.kaggle.com/datasets/rajnishe/facescrub-full>

Phụ lục A

Ngữ pháp tiếng Việt

Đây là phụ lục.

Phụ lục B

Ngữ pháp tiếng Nôm

Đây là phụ lục 2.