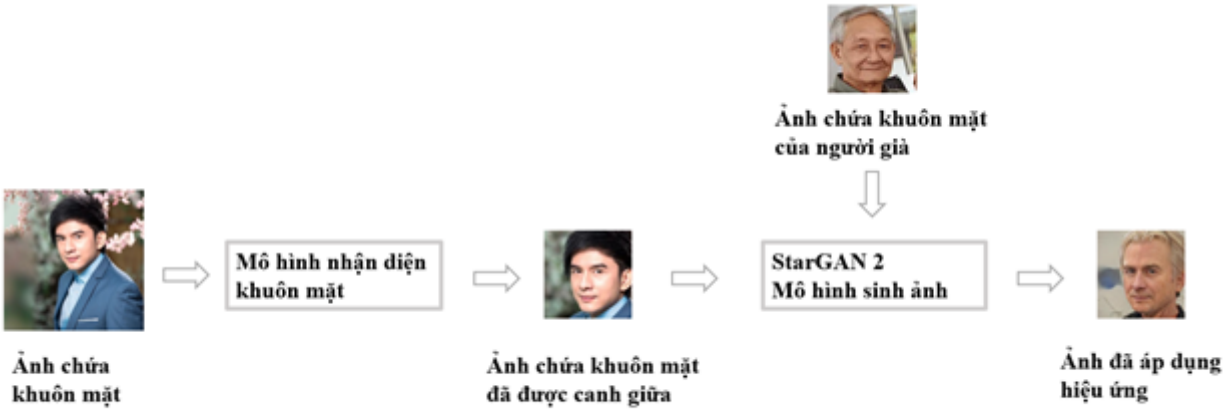
Chương 3

Giải pháp đề tài

3.1 Tổng quan giải pháp kiến trúc mô hình

Nhóm sinh viên đề xuất sử dụng mô hình StarGANv2 kết hợp cùng với mô hình nhận diện khuôn mặt dựa trên thuật toán HOG và SVM của thư viện Dlib để tạo hệ thống mô hình chỉnh sửa ảnh đầu cuối (end-to-end). Mô hình StarGANv2 được nhóm Clova AI công bố vào tháng 04/2020, có nhiều cải tiến so với phiên bản tiền nhiệm và có tiếp thu sự cải tiến từ mô hình tân tiến nhất Style GAN.

Kiến trúc tổng quan của hệ thống được minh họa cụ thể ở hình 3.1. Trong đó, nhóm sinh viên sẽ tập trung chính vào phần mô hình sinh ảnh StarGANv2.



Hình 3.1: Tổng quan kiến trúc mô hình

Tiếp theo đây, nhóm sẽ trình bày các giải pháp giải quyết vấn đề cho từng thành phần trong hệ thống chỉnh sửa ảnh, hướng xây dựng máy chủ và ứng dụng áp dụng trên nền tảng di động.

3.2 Giải pháp xây dựng mô hình nhận diện khuôn mặt

Hình 3.2: Tổng quan về dây chuyền trích xuất đặc trưng và dò tìm đối tượng. (Nguồn: [0])

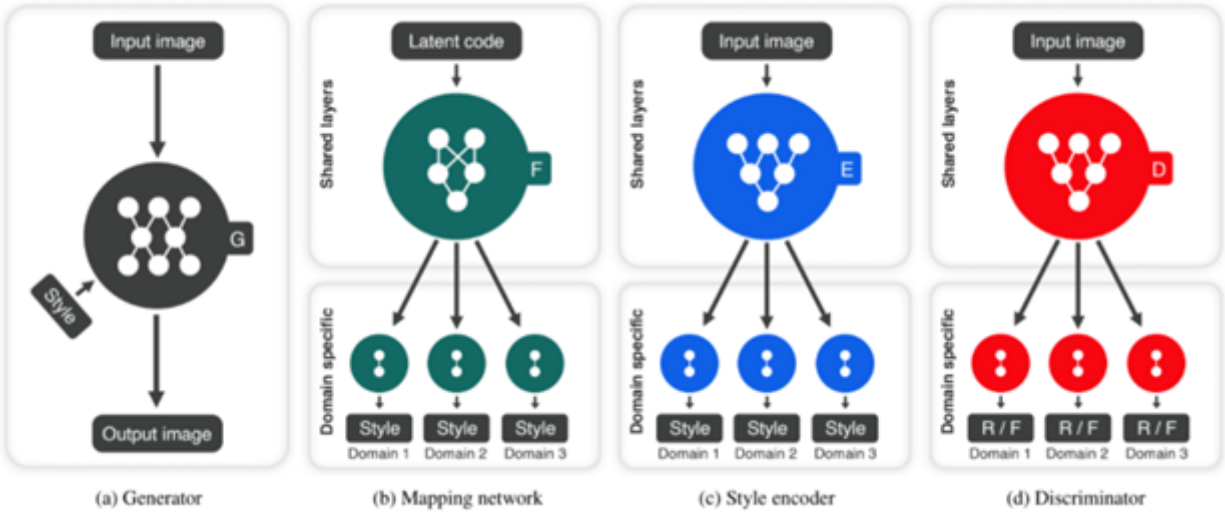
Phương pháp này dựa trên việc đánh giá các biểu đồ cục bộ của các hướng gradient hình ảnh (image gradient orientations) trong một lưới dày đặc được chuẩn hóa. Các tính năng tương tự đã được sử dụng ngày càng tăng trong thập kỷ qua. Ý tưởng cơ bản là sự xuất hiện cục bộ và hình

dạng của đối tượng thường có thể đặc trưng hóa khá tốt bởi sự phân bố các cường độ gradient cục bộ (local intensity gradients) hoặc các hướng cạnh, ngay cả khi không có kiến thức chính xác về các vị trí gradient hoặc cạnh tương ứng. Trong thực tế, điều này được thực hiện bằng cách chia hình ảnh thành các vùng không gian nhỏ (gọi là "ô"), mỗi ô này sẽ được tính để tạo thành biểu đồ 1 chiều của các hướng gradient cục bộ hoặc hướng cạnh trên các pixel của ô. Các mục nhập biểu đồ kết hợp tạo thành sự hiện diện. Để bớt phụ thuộc vào ánh sáng, bóng đổ,... chúng ta nên "tương phản-chuẩn hóa" (contrast-normalize) các phản hồi cục bộ trước khi sử dụng chúng. Điều này có thể được thực hiện bằng cách tích lũy một thước đo biểu đồ địa phương "năng lượng" trên phần nào các vùng không gian lớn hơn (gọi là "khối") và sử dụng kết quả để chuẩn hóa tất cả các ô trong khối. Các khối descriptor chuẩn hóa dưới dạng Biểu đồ descriptor Gradient Định hướng (HOG). Lát ô cửa sổ dò tìm bằng một lưới các descriptor HOG dày đặc (trên thực tế là chồng chéo) và sử dụng vectơ đặc trưng kết hợp với mô hình SVM dựa trên phân loại ô cửa sổ đem lại dây chuyền phát hiện con người.

3.3 Giải pháp xây dựng mô hình sinh ảnh

3.3.1 Khung chương trình đề xuất

Gọi $\mathcal{X}$ và $\mathcal{Y}$ lần lượt là tập ảnh và các miền. Cho một ảnh $\mathbf{x} \in \mathcal{X}$ và một miền tùy ý $y \in \mathcal{Y}$, mục tiêu của chúng ta là huấn luyện một bộ sinh G duy nhất có thể tạo ra các ảnh đa dạng của mỗi miền y tương ứng với ảnh $\mathbf{x}$. Chúng ta tạo vectơ kiểu cụ thể cho miền trong không gian kiểu đã học của mỗi miền và huấn luyện G để phản ánh các vectơ kiểu. Hình 3.3 minh họa tổng quan về khung mô hình, bao gồm bốn mô-đun được mô tả bên dưới.



Hình 3.3: Tổng quan mô hình StarGANv2. (Nguồn [0])

Bộ sinh (Hình 3.3a). Bộ sinh G có nhiệm vụ chuyển hình ảnh đầu vào $\mathbf{x}$ thành hình ảnh đầu ra $G(\mathbf{x}, \mathbf{s})$ phản ánh mã kiểu thuộc miền cụ thể $\mathbf{s}$. Mã kiểu $\mathbf{s}$ sẽ được cung cấp bởi mạng ánh xạ F hoặc bởi bộ mã hóa kiểu E. Sau đó, chuẩn hóa phiên bản thích ứng (AdaIN) được sử dụng để truyền $\mathbf{s}$ vào G. Chúng ta có thể thấy rằng $\mathbf{s}$ được thiết kế để đại diện cho kiểu của một miền cụ thể y. Điều này loại bỏ sự cần thiết của việc cung cấp y cho bộ sinh G và cho phép G tổng hợp ảnh của tất cả các miền.

Mạng ánh xạ (Hình 3.3b). Cho một mã tiềm ẩn $\mathbf{z}$ và một miền y, mạng ánh xạ F cho ra một mã kiểu $\mathbf{s} = F_y(\mathbf{z})$, trong đó $F_y(.)$ biểu thị một đầu ra của F tương ứng với miền y. F bao gồm một MLP với nhiều nhánh đầu ra để cung cấp mã kiểu cho tất cả các miền có sẵn. F có thể tạo ra các mã kiểu đa dạng bằng cách lấy mẫu vectơ tiềm ẩn $\mathbf{z} \in \mathcal{Z}$ và miền $y \in \mathcal{Y}$ một cách ngẫu nhiên. Kiến trúc đa tác vụ cho phép F học cách biểu diễn kiểu của tất cả các miền một cách hiệu quả.

Bộ mã hóa kiểu (Hình 3.3c). Cho một hình ảnh $\mathbf{x}$ và miền tương ứng của nó là y, bộ mã hóa E có nhiệm vụ trích xuất mã kiểu $\mathbf{s} = E_y(\mathbf{x})$ của $\mathbf{x}$. Ở đây, $E_y(.)$ biểu thị đầu ra của E tương ứng với miền y. Tương tự như F, bộ mã hóa kiểu E cũng được hưởng lợi từ việc thiết lập học đa tác vụ. E có thể tạo ra các mã kiểu đa dạng sử dụng các hình ảnh tham khảo

khác nhau. Điều này cho phép G tổng hợp một hình ảnh đầu ra phản ánh kiểu dáng của một hình ảnh tham chiếu $\mathbf{x}$.

Bộ phân biệt (Hình 3.3d). Bộ phân biệt D chính là bộ phân biệt đa tác vụ. Nó bao gồm nhiều nhánh đầu ra. Mỗi nhánh Dy học cách phân loại xem ảnh $\mathbf{x}$ là ảnh thực của miền y hay ảnh giả $G(\mathbf{x}, \mathbf{s})$ do bộ sinh G tạo ra.

3.3.2 Các mục tiêu huấn luyện mô hình

Cho một hình ảnh $\mathbf{x} \in \mathcal{X}$ với miền gốc của nó là $y \in \mathcal{Y}$, khung mô hình được đào tạo bằng cách sử dụng các mục tiêu sau.

Mục tiêu đối kháng. Trong quá trình đào tạo, ta lấy mẫu một mã tiềm ẩn $\mathbf{z} \in \mathcal{Z}$ và miền đích $\tilde{y} \in \mathcal{Y}$ một cách ngẫu nhiên, và tạo mã kiểu đích $\tilde{\mathbf{s}} = F_{\tilde{y}}(\mathbf{z})$. Bộ sinh G lấy hình ảnh $\mathbf{x}$ và $\tilde{\mathbf{s}}$ làm đầu vào và học cách tạo đầu ra hình ảnh $G(\mathbf{x}, \tilde{\mathbf{s}})$ dựa trên hàm mất mát đối nghịch.

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x},y}[\log D_y(\mathbf{x})] + \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}}[\log(1 - D_{\tilde{y}}(G(\mathbf{x}, \tilde{\mathbf{s}})))]$$

trong đó $D_y(.)$ biểu thị đầu ra của D tương ứng với miền y. Mạng ánh xạ F học cách tạo các mã kiểu $\tilde{\mathbf{s}}$ có thể có trong miền đích $\tilde{y}$ và G học cách sử dụng các $\tilde{\mathbf{s}}$ và tạo ra hình ảnh $G(\mathbf{x}, \tilde{\mathbf{s}})$ không thể phân biệt được so với hình ảnh thực của miền $\tilde{y}$.

Tái tạo kiểu. Để bắt buộc hàm sinh G sử dụng mã kiểu $\tilde{\mathbf{s}}$ khi tạo hình ảnh $G(\mathbf{x}, \tilde{\mathbf{s}})$, một hàm mất mát tái tạo kiểu được sử dụng.

$$\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}}[\|\tilde{\mathbf{s}} - E_{\tilde{y}}(G(\mathbf{x}, \tilde{\mathbf{s}}))\|_1]$$

Mục tiêu này tương tự như các cách tiếp cận trước đây, sử dụng nhiều bộ mã hóa để học cách ánh xạ từ một hình ảnh sang mã tiềm ẩn của nó. Sự khác biệt đáng chú ý là chúng ta đào tạo một bộ mã hóa E duy nhất để khuyến khích các đầu ra đa dạng hóa cho nhiều miền. Tại thời điểm thử nghiệm, bộ mã hóa E đã được huấn luyện cho phép G biến đổi hình ảnh

đầu vào và phản ánh kiểu của hình ảnh tham chiếu.

Đa dạng hóa kiểu. Để cho phép bộ sinh G tạo ra các hình ảnh đa dạng hơn nữa, chúng ta chính quy hóa tường minh G bằng hàm mất mát đa dạng nhạy cảm.

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x},\tilde{y},\mathbf{z}_1,\mathbf{z}_2}[\|G(\mathbf{x},\tilde{\mathbf{s}}_1) - G(\mathbf{x},\tilde{\mathbf{s}}_2)\|_1]$$

trong đó mã kiểu đích $\tilde{\mathbf{s}}_1$ và $\tilde{\mathbf{s}}_2$ được tạo ra bởi F có điều kiện dựa trên hai mã tiềm ẩn ngẫu nhiên $\mathbf{z}_1$ và $\mathbf{z}_2$ ($\tilde{\mathbf{s}}_i = F_{\tilde{y}}(\mathbf{z}_i), i \in \{1,2\}$). Việc tối đa hóa sự chính quy buộc G phải khám phá không gian hình ảnh và khám phá các đặc trưng kiểu có ý nghĩa để tạo ra các hình ảnh một cách đa dạng. Lưu ý rằng ở giai đoạn ban đầu, sự chênh lệch nhỏ của $\|\mathbf{z}_1 - \mathbf{z}_2\|_1$ ở mẫu số làm tăng tổn thất đáng kể. Điều này dẫn đến việc huấn luyện không ổn định do có gradient lớn. Do đó, chúng ta loại bỏ phần mẫu số và đưa ra một phương trình mới để luyện tập ổn định.

Bảo toàn đặc tính nguồn. Để đảm bảo rằng hình ảnh được tạo $G(\mathbf{x},\tilde{\mathbf{s}})$ duy trì đúng các đặc điểm bất biến thuộc miền (ví dụ: tư thế) của hình ảnh đầu vào $\mathbf{x}$, chúng ta sử dụng hàm mất mát nhất quán của chu trình

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x},y,\tilde{y},\mathbf{z}}[\|\mathbf{x} - G(G(\mathbf{x},\tilde{\mathbf{s}}),\hat{\mathbf{s}})\|_1]$$

trong đó $\hat{\mathbf{s}} = E_y(\mathbf{x})$ là mã kiểu ước tính của ảnh đầu vào $\mathbf{x}$, và y là miền gốc của $\mathbf{x}$. Bằng cách khuyến khích bộ sinh G tái tạo lại hình ảnh đầu vào $\mathbf{x}$ với mã kiểu ước tính $\hat{\mathbf{s}}$, G học cách bảo toàn các đặc điểm ban đầu của $\mathbf{x}$ trong khi thay đổi kiểu của nó một cách trung thực.

Mục tiêu đầy đủ. Hàm mục tiêu đầy đủ được tóm tắt như sau:

$$\min_{G,F,E} \max_{D} \quad \mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc}$$

trong đó λ_{sty}, λ_{ds} và λ_{cyc} là các siêu tham số cho mỗi số hạng. Một mô hình khác với cùng mục tiêu kể trên cũng được huấn luyện với sự thay đổi nhỏ là sử dụng hình ảnh tham chiếu thay thế cho vector tiềm ẩn khi tạo

mã kiểu.

3.3.3 Kiến trúc mạng nơ-ron

Ở mục này, nhóm sinh viên mô tả chi tiết kiến trúc của mô hình StarGAN v2, bao gồm bốn mô-đun sau

Bộ sinh (Generator) (Bảng 3.1): Với tập dữ liệu AFHQ, mạng này gồm bốn khối giảm mẫu (downsampling), bốn khối trung gian ở giữa, bốn khối tăng mẫu (upsample), tất cả đều sử dụng chung đơn vị tiền kích hoạt phần dư (pre-activation residual units). Nhóm sử dụng Instance normalization (IN) và Adaptive Instance Normalization (AdaIN) cho lần lượt các khối giảm mẫu và tăng mẫu tương ứng. Một mã kiểu được truyền vào toàn bộ các lớp AdaIN, cung cấp các vector tỉ lệ và vector tịnh tiến thông qua các biến đổi afin (Affine transformation) học được. Với tập dữ liệu CelebA-HQ, số lượng của các tầng giảm mẫu và tầng tăng mẫu đều được tăng từng đôi một. Toàn bộ các lối tắt (shortcut) trong khối phần dư tăng mẫu được xóa bỏ và thêm vào skip connection với adaptive wing dựa trên heatmap.

Tầng	Tái lấy mẫu	Chuẩn hóa	Chiều đầu ra
Hình ảnh x	-	-	256 x 256 x 3
Conv1x1	-	-	256 x 256 x 64
ResBlk	AvgPool	IN	128 x 128 x 128
ResBlk	AvgPool	IN	64 x 64 x 256
ResBlk	AvgPool	IN	32 x 32 x 512
ResBlk	AvgPool	IN	16 x 16 x 512
ResBlk	-	IN	16 x 16 x 512
ResBlk	-	IN	16 x 16 x 512
ResBlk	-	AdaIN	16 x 16 x 512
ResBlk	-	AdaIN	16 x 16 x 512
ResBlk	Upsample	AdaIN	32 x 32 x 512
ResBlk	Upsample	AdaIN	64 x 64 x 256
ResBlk	Upsample	AdaIN	128 x 128 x 128
ResBlk	Upsample	AdaIN	256 x 256 x 64
Conv1x1	-	-	256 x 256 x 3

Bảng 3.1: Kiến trúc bộ sinh

Mạng ánh xạ (Mapping network) (Bảng 3.2): Mạng ánh xạ bao gồm một mạng MLP với K nhánh đầu ra, với K đại diện cho số lượng miền. Bốn tầng kết nối đầy đủ (fully-connected layer) được chia sẻ với toàn bộ các miền, theo sau là bốn tầng kết nối đầy đủ riêng biệt cho mỗi miền. Chiều cho các mã tiềm ẩn, tầng ẩn (hidden layer), mã kiểu lần lượt là 16, 512, 64. Mã tiềm ẩn được lấy mẫu từ phân phối Gaussian. Pixel Normalization không được áp dụng với mã tiềm ẩn, bởi vì nó đã được quan sát rằng không cải thiện hiệu suất mô hình cho công việc trên. Chuẩn hóa đặc trưng (feature normalization) cũng đã được thử nghiệm nhưng nó cũng làm giảm hiệu suất.

Loại	Tầng	Kích hoạt	Chiều đầu ra
Được chia sẻ	z tiềm ẩn	-	16
Được chia sẻ	Tuyến tính	ReLU	512
Được chia sẻ	Tuyến tính	ReLU	512
Được chia sẻ	Tuyến tính	ReLU	512
Được chia sẻ	Tuyến tính	ReLU	512
Không chia sẻ	Tuyến tính	ReLU	512
Không chia sẻ	Tuyến tính	ReLU	512
Không chia sẻ	Tuyến tính	ReLU	512
Không chia sẻ	Tuyến tính	ReLU	64

Bảng 3.2: Kiến trúc mạng ánh xạ

Bộ mã hóa kiểu (Style encoder) (Bảng 3.3): bộ mã hóa bao gồm một mạng nơ-ron tính chập (CNN) với K nhánh đầu ra, trong đó K là số lượng các miền. Sáu khối tiền kích hoạt phần dư (pre-activation residual block) được chia sẻ với toàn bộ các miền, theo sau bởi một tầng kết nối đầy đủ (fully connected layer) cho mỗi miền. Gộp trung bình toàn cục (Global average pooling) không được sử dụng để trích xuất kiểu toàn diện (fine stye) đặc trưng cho tấm ảnh tham chiếu. Chiều đầu ra của D được gán bằng 64, chính là chiều của mã kiểu..

Tầng	Tái lấy mẫu	Chuẩn hóa	Chiều đầu ra
Hình ảnh x	-	-	256 x 256 x 3
Conv1x1	-	-	256 x 256 x 64
ResBlk	AvgPool	-	128 x 128 x 128
ResBlk	AvgPool	-	64 x 64 x 256
ResBlk	AvgPool	-	32 x 32 x 512
ResBlk	AvgPool	-	16 x 16 x 512
ResBlk	AvgPool	-	8 x 8 x 512
ResBlk	AvgPool	-	4 x 4 x 512
LReLU	-	-	4 x 4 x 512
Conv4x4	-	-	1 x 1 x 512
LReLU	-	-	1 x 1 x 512
Reshape	-	-	512
Linear * K	-	-	D * K

Bảng 3.3: Kiến trúc bộ mã hóa kiểu và bộ phân biệt. D và K đại diện cho chiều đầu ra và số lượng miền.

Bộ phân biệt (Discriminator) (Bảng 3.3). Bộ phân biệt đề xuất là một bộ phân biệt đa tác vụ, chứa nhiều nhánh đầu ra tuyến tính. Bộ phân biệt gồm sáu khối tiền kích hoạt phần dư với đơn vị tuyến tính chỉnh lưu rò rỉ (Leaky ReLU). K tầng kết nối đầy đủ được sử dụng cho sự phân loại thật/giả của mỗi miền, trong đó K chính là số lượng các miền. Chiều của đầu ra "D" được gán bằng 1 cho sự phân loại thật/giả. Các kỹ thuật chuẩn hóa đặc trưng hoặc PatchGAN đều không được sử dụng vì tất cả đã được quan sát rằng không cải thiện chất lượng đầu ra. Theo như quan sát được trong phần cài đặt, bộ phân biệt đa tác vụ cho kết quả tốt hơn những dạng khác của bộ phân biệt có điều kiện.

3.4 Giải pháp xây dựng mô hình nhận dạng giới tính

3.5 Giải pháp xây dựng máy chủ

3.6 Giải pháp xây dựng ứng dụng

3.6.1 Thiết kế giao diện

3.6.2 Thiết kế kiến trúc