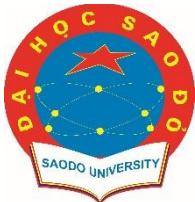


BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ



PHẠM ANH QUÂN

ĐỒ ÁN TỐT NGHIỆP

Ngành: Công nghệ thông tin

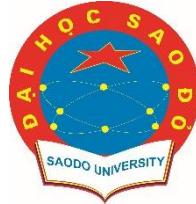
Họ và tên sinh viên: Phạm Anh Quân

Lớp, khoá: DK10-CNTT

NGÀNH: CÔNG NGHỆ THÔNG TIN

HẢI DƯƠNG – NĂM 2023

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ



ĐỒ ÁN TỐT NGHIỆP

Ngành: Công nghệ thông tin

TÊN ĐỀ TÀI:

NGHIÊN CỨU XÂY DỰNG ỨNG DỤNG THAY ĐỔI
TRANG PHỤC THEO MÔ TẢ CHO NGƯỜI TRONG ANH

Họ và tên sinh viên: Phạm Anh Quân

Lớp, khoá: DK10-CNTT

Giảng viên hướng dẫn: ThS. Hoàng Thị Ngát

HẢI ĐƯƠNG – NĂM 2023

LỜI CAM ĐOAN

Tôi xin cam đoan các kết quả đưa ra trong đồ án/khoa luận tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu, thực nghiệm của tôi dưới sự hướng dẫn của ThS. Hoàng Thị Ngát, không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục các tài liệu tham khảo.

Nếu sai tôi xin chịu mọi hình thức kỷ luật theo quy định.

Hải Dương, ngày.....tháng.....năm.....

Sinh viên thực hiện

(Ký, ghi rõ họ và tên)

MỤC LỤC

LỜI CAM ĐOAN	3
MỞ ĐẦU	6
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	9
1.1. Tổng quan về xử lý ảnh.....	9
1.1.1. Tổng quan.....	9
1.1.2. Phần tử ảnh và nồng độ	10
1.1.3. Quá trình xử lý ảnh.....	11
1.1.4. Các vấn đề trong xử lý ảnh.....	11
1.2. Học máy	15
1.2.1. Tổng quan về học máy	15
1.2.2. Lịch sử phát triển.....	16
1.2.3. Các loại học máy	16
1.2.4. Ứng dụng học máy trong thực tế.....	19
1.3. Ứng dụng của học máy và xử lý ảnh	20
CHƯƠNG 2 : PHÂN TÍCH MÔ HÌNH ỦNG DỤNG	22
2.1. Bài toán chuyển đổi trang phục	22
2.2. Mạng Content Generating and Preserving.....	23
2.2.1. Semantic Generation Module (SGM)	23
2.2.2. Clothes Warping Module (CWM)	24
2.2.3. Content Fusion Module (CFM).....	25
2.3. Tiền xử lí hình ảnh	25
2.3.1. Kiểm tra và chấp nhận ảnh.....	25
2.3.2. Resize và Crop Hình Ảnh	26
2.3.3. Lưu trữ hình ảnh.....	27
2.3.4. Tạo mặt nạ trang phục.....	28
2.3.5. Chuyển đổi và masking hình ảnh	29
2.3.6. Đồng bộ hóa dữ liệu	29
2.3.7. Resize và lưu trữ hình ảnh đã chuyển đổi	30
2.3.8. Tối ưu hóa và mở rộng	30
2.3.9. Xử lý lỗi	31
2.4. Áp dụng học máy vào bài toán Try-on	31
2.4.1. Xử lý dữ liệu và chuẩn bị	31
2.4.2. Tạo Mô Hình Học Máy	33
2.4.3. Huấn luyện mô hình trên tập dữ liệu try-on.....	34
2.4.4. Kiểm thử và Đánh giá Mô Hình.....	36
2.5. Môi trường xây dựng ứng dụng	38
2.5.1. Visual Studio Code	38

2.5.2. Môi trường ảo Python	40
CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG.....	42
3.1. Xây dựng ứng dụng.....	42
3.1.1. Xây dựng giao diện	42
3.1.2. Xây dựng chức năng	46
3.2. Giao diện và chức năng hoàn chỉnh	53
3.2.1. Giao diện Home	53
3.2.2. Chức năng thay đổi trang phục	55
Kết quả và đánh giá	56
Tài liệu tham khảo	58

Đại học Sao Đỏ

DANH MỤC HÌNH ẢNH

Hình 1: Quá trình xử lý ảnh.....	10
Hình 2: Phần tử trong ảnh.....	10
Hình 3: Các bước cơ bản trong một hệ thống xử lý ảnh	11
Hình 4: Giải hệ phương trình tuyến tính tìm được a1, b1, c1	12
Hình 5: Ảnh thu nhận và ảnh mong muốn	12
Hình 6: Các loại nhiễu trong ảnh.....	13
Hình 7: Ảnh xám	13
Hình 8: Nén ảnh.....	15
Hình 9: Học máy.....	15
Hình 10: Mô hình học có giám sát	17
Hình 11: Mô hình học không giám sát	18
Hình 12: Mô hình học tăng cường.....	18
Hình 13: Ví dụ nhận dạng khuôn mặt	20
Hình 14: Mô tả bài toán chuyển đổi trang phục	22
Hình 15: Cấu trúc Mạng Content Generating and Preserving.....	23
Hình 16: So sánh kết quả hình ảnh của 2 mô hình	24
Hình 17: Tập dữ liệu huấn luyện	32
Hình 18: Visual Studio Code.....	39
Hình 19: Môi trường ảo của Python	41
Hình 20: Giao diện Home.....	54
Hình 21:Chức năng thay đổi trang phục.....	55

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, sự phát triển nhanh chóng của công nghệ thông tin đã đưa đến một cuộc cách mạng trong cuộc sống hàng ngày và cách chúng ta thực hiện công việc. Trong bối cảnh này, việc ứng dụng Công nghệ thông tin vào việc thay đổi trang phục trở nên ngày càng quan trọng. Phản ảnh đúng thị hiếu của người dùng về việc sử dụng và nâng cấp các công cụ chỉnh sửa phục vụ cuộc sống.

Lý do chọn đề tài Trí tuệ nhân tạo (Artificial Intelligence) đã mở ra nhiều lĩnh vực mới và thú vị, trong đó có xử lý ảnh và học máy. Việc thay đổi trang phục cho người trong ảnh có ứng dụng rộng rãi trong lĩnh vực thời trang, quảng cáo, và thương mại điện tử. Bạn có thể phát triển công cụ giúp người dùng xem trước các trang phục trông sẽ như nào trước khi mua sắm.

Đề tài này nhằm mục đích nghiên cứu và phát triển một ứng dụng chuyên sâu có khả năng thực hiện việc thay đổi trang phục, đặc biệt là áo trên người trong bức ảnh dựa trên mô tả. Mục tiêu chính là tạo ra một công cụ thông minh và linh hoạt, giúp người dùng dễ dàng thực hiện việc thay đổi loại áo, mẫu mã, màu sắc, và các chi tiết thiết kế khác một cách tự động và chính xác.

Ứng dụng này sẽ tập trung vào việc cải thiện trải nghiệm người dùng trong quá trình chỉnh sửa hình ảnh bằng cách giúp họ mô phỏng và xem trước các lựa chọn áo một cách trực quan. Ngoài ra, mục đích còn bao gồm việc tối ưu hóa quy trình làm việc và giảm thiểu thời gian cần thiết cho việc thay đổi áo, đồng thời đảm bảo tính thẩm mỹ và phù hợp với người mẫu trong bức ảnh. Đề tài hứa hẹn mang lại ứng dụng có tính ứng dụng cao trong lĩnh vực chỉnh sửa ảnh và thời trang, đáp ứng nhu cầu ngày càng tăng của người sử dụng với sự thuận tiện và linh hoạt.

Chọn đề tài này mang lại sự tự do sáng tạo vì bạn có thể thử nghiệm với các mô hình học máy khác nhau, thuật toán xử lý ảnh và giao diện người dùng để tạo ra một sản phẩm hoặc ứng dụng độc đáo.

2. Mục tiêu nghiên cứu

- Nghiên cứu các thuật toán xử lý ảnh tiên tiến như GMM, TOM, Old Photo Restoration[1].
- Nghiên cứu các giải thuật, mô hình AI trong xử lý và thay đổi trang phục.
- Xây dựng ứng dụng Web để kết nối đến API và cung cấp giao diện người dùng.

3. Đối tượng nghiên cứu

- Các thuật toán xử lý ảnh nâng cao: mạng GAN, Transformer[2],...
- Các kỹ thuật đánh giá chất lượng mô hình thay đổi trang phục

4. Phạm vi nghiên cứu

- Nghiên cứu và lựa chọn các thuật toán AI hiện đại phù hợp với bài toán.
- Xây dựng mô hình End-to-End, huấn luyện và đánh giá trên tập dữ liệu.
- Triển khai mô hình dưới dạng API và xây dựng ứng dụng cho người dùng cuối.

5. Phương pháp nghiên cứu

Phương pháp lý thuyết:

- Thu thập, đọc và phân tích các công trình nghiên cứu, bài báo khoa học liên quan.
- Tổng hợp và so sánh các giải pháp, phương án thiết kế mô hình.

Phương pháp thực nghiệm: Thiết kế, xây dựng, chạy thử nghiệm ứng dụng và đánh giá kết quả.

6. Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học: Nghiên cứu về các thuật toán và phương pháp thay đổi trang phục như GMM, TOM và bài toán khôi phục ảnh cũ.

Ý nghĩa thực tế: Xây dựng ứng dụng thay đổi trang phục sử dụng các thuật toán và phương pháp thay đổi trang phục.

7. Kết cấu đề tài

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích mô hình ứng dụng

Chương 3: Xây dựng ứng dụng

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về xử lý ảnh

1.1.1. Tổng quan

Xử lý ảnh là một lĩnh vực khoa học và công nghệ mới mẻ, với tốc độ phát triển rất nhanh. Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Từ đó đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

Xử lý ảnh là quá trình chuyển đổi một hình ảnh sang dạng kỹ thuật số và thực hiện các thao tác nhất định để nhận được một số thông tin hữu ích từ hình ảnh đó. Hệ thống xử lý hình ảnh thường coi tất cả các hình ảnh là tín hiệu 2D khi áp dụng một số phương pháp xử lý tín hiệu đã xác định trước.

Các loại xử lý hình ảnh chính:

- Nhận diện – Phân biệt hoặc phát hiện các đối tượng trong hình ảnh
- Làm sắc nét và phục hồi – Tạo hình ảnh nâng cao từ hình ảnh gốc
- Nhận dạng mẫu – Đo các mẫu khác nhau xung quanh các đối tượng trong hình ảnh
- Truy xuất – Duyệt và tìm kiếm hình ảnh từ một cơ sở dữ liệu lớn gồm các hình ảnh kỹ thuật số tương tự như hình ảnh gốc

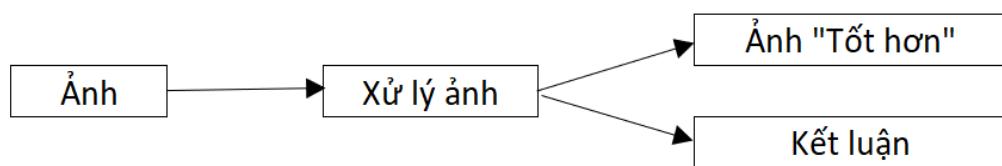
Xử lý hình ảnh là việc chọn lọc những thông tin mong muốn từ những bức ảnh chụp từ máy ảnh hay máy scan.

Để chọn lọc những thông tin mong muốn được chứa trong bức ảnh, phương pháp cơ bản được dùng trong xử lý hình ảnh là loại bỏ những phần thông tin ảnh không mong muốn.

Xử lý hình ảnh kỹ thuật số bắt đầu từ việc phân chia hình ảnh ra thành các phần tử ảnh đơn vị (gọi là Pixel = Picture Element).

Các máy ảnh kỹ thuật số ngày nay lưu các hình ảnh bằng các phần tử ảnh đơn vị (Pixel) nên những hình ảnh được lưu lại trở thành các hình ảnh kỹ thuật số.

Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



Hình 1: Quá trình xử lý ảnh

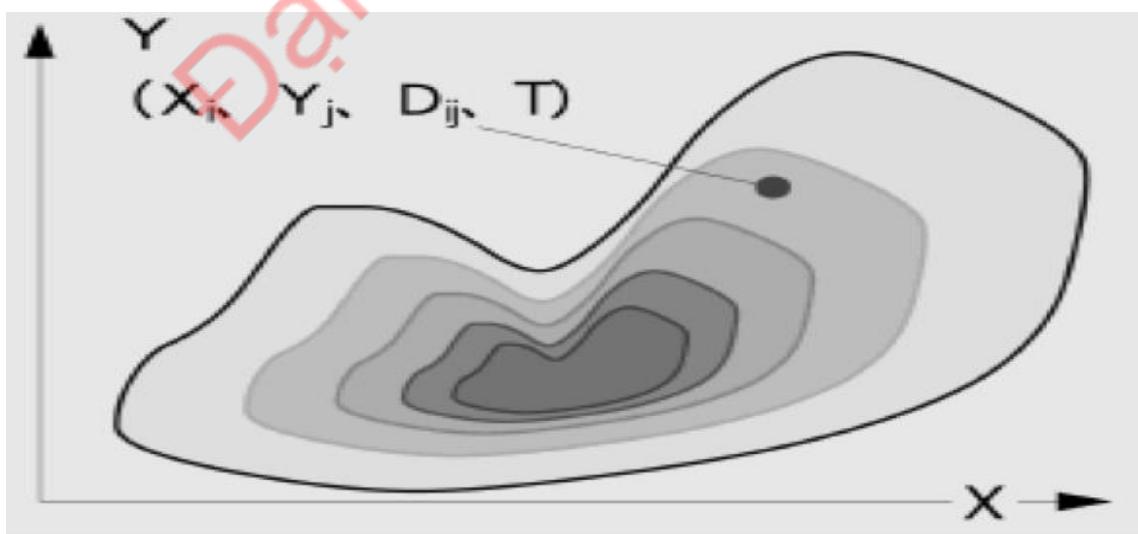
1.1.2. Phần tử ảnh và nồng độ

Bước thứ nhất trong xử lý hình ảnh kỹ thuật số là phân chia hình ảnh ra thành các phần tử ảnh đơn vị. Số lượng phần tử ảnh đơn vị cấu thành nên hình ảnh càng nhiều thì hình ảnh càng chi tiết.

Phần tử ảnh là đơn vị cơ bản của hình ảnh, mang trong mình thông tin vị trí và thông tin nồng độ của hình ảnh.

Tóm lại, hình ảnh kỹ thuật số là những thông tin kỹ thuật số được cấu thành từ thông tin toạ độ và thông tin nồng độ được biểu diễn bằng:

- Điểm $(X_i, Y_j, D_{ij}) \dots (IDP)$
- X_i : Vị trí phần tử ảnh trực X
- Y_j : Vị trí phần tử ảnh trực Y
- D_{ij} : Thông tin nồng độ giao điểm X,Y
- T: Thông tin thời gian (=số frame/tốc độ chụp)



Hình 2: Phần tử trong ảnh

Nồng độ ảnh là những thông tin về màu sắc (tỉ lệ 3 màu cơ bản đỏ, xanh lam, xanh lá). Thông thường, để hiển thị một phần tử ảnh cần thông tin về màu sắc của 3 màu cơ bản.

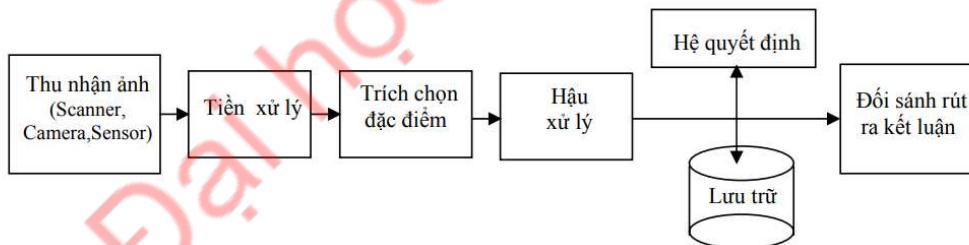
Dung lượng ảnh: Như đã nói ở trên, trong công nghệ ảnh kĩ thuật số, người ta chia bức ảnh ra làm các phần tử ảnh đơn vị. Như vậy, 1 bức ảnh được ghép từ 640×480 phần tử ảnh đơn vị, màu sắc 24 bit (mỗi phần tử ảnh cần chứa thông tin màu sắc của 3 màu cơ bản, mỗi màu cần 8bit lưu trữ nên gọi là màu sắc 24bit) thì cần dung lượng là:

$$\text{Số phần tử ảnh} \times \text{Dung lượng mỗi phần tử} = 640 \times 480 \times 24 \text{ (bit)} = 921.6 \text{ (KB)}.$$

1.1.3. Quá trình xử lý ảnh

Quá trình xử lý ảnh bao gồm nhiều bước khác nhau nhằm cải thiện, phân tích, và hiểu nội dung của hình ảnh. Các bước chính có thể bao gồm:

- Thu thập ảnh: thu thập dữ liệu hình ảnh từ nguồn đầu vào.
- Tiền xử lý: loại bỏ nhiễu, cân bằng màu sắc, điều chỉnh độ tương phản để chuẩn bị dữ liệu cho các bước tiếp theo.
- Trích xuất đặc trưng: nhận diện và trích xuất các đặc trưng quan trọng từ ảnh, như cạnh, góc, đối tượng.
- Phân loại và dự đoán: sử dụng các mô hình máy học để phân loại hoặc dự đoán nội dung của ảnh.
- Kết quả xử lý: xuất kết quả sau khi xử lý, có thể là ảnh cải thiện hoặc thông tin chi tiết từ quá trình phân loại.



Hình 3: Các bước cơ bản trong một hệ thống xử lý ảnh

1.1.4. Các vấn đề trong xử lý ảnh

a. Một số khái niệm cơ bản

Ảnh và điểm ảnh: Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại 1 toạ độ trong không gian của đối tượng và ảnh được xem như là 1 tập hợp các điểm ảnh.

Mức xám, màu: Là số các giá trị có thể có của các điểm ảnh của ảnh

b. Nắn chỉnh biến dạng

Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử.

Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển.

Giả sử ảnh bị biến đổi chỉ bao gồm: Tịnh tiến, quay, tỷ lệ, biến dạng bậc nhất tuyến tính. Khi đó hàm f có dạng:

$$f(x, y) = (a_1x + b_1yc_1, a_2xb_2y + c_2)$$

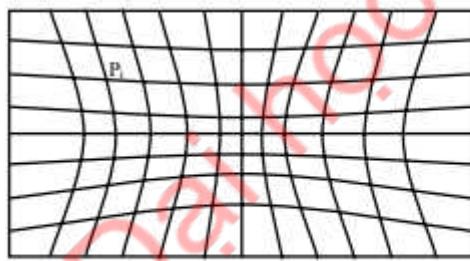
Ta có:

$$\phi = \sum_{i=1}^n (f(P_i) - P_i)^2 = \sum_{i=1}^n [(a_1x_i + b_1y_i + c_1 - x_i)^2 + (a_2x_i + b_2y_i + c_2 - y_i)^2]$$

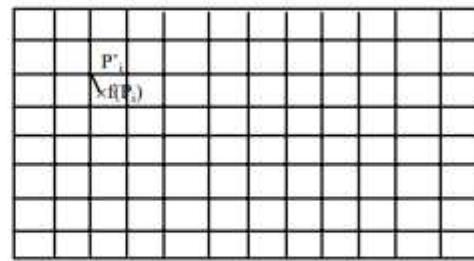
Để cho $\phi \rightarrow \min$

$$\begin{cases} \frac{\partial \phi}{\partial a_1} = 0 \\ \frac{\partial \phi}{\partial b_1} = 0 \\ \frac{\partial \phi}{\partial c_1} = 0 \end{cases} \Leftrightarrow \begin{cases} \sum_{i=1}^n a_1x_i^2 + \sum_{i=1}^n b_1x_iy_i + \sum_{i=1}^n c_1x_i = \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n a_1x_iy_i + \sum_{i=1}^n b_1y_i^2 + \sum_{i=1}^n c_1y_i = \sum_{i=1}^n y_i^2 \\ \sum_{i=1}^n a_1x_i + \sum_{i=1}^n b_1y_i + nc_1 = \sum_{i=1}^n x_i \end{cases}$$

Hình 4: Giải hệ phương trình tuyến tính tìm được a_1, b_1, c_1 , tương tự tìm được a_2, b_2, c_2



Ảnh thu nhận



Ảnh mong muốn

Hình 5: Ảnh thu nhận và ảnh mong muốn

c. Các loại nhiễu trong ảnh

Nhiễu có quy luật (nhiễu hệ thống): là loại nhiễu mà chúng ta có thể mô hình và khử bằng cách sử dụng các phép biến đổi. Điều này đặc biệt hữu ích khi ta biết được nguồn gốc và cách nhiễu được tạo ra. Việc này cho phép áp dụng các phương pháp xử lý chuyên sâu để giảm thiểu ảnh hưởng của nhiễu.

Nhiễu ngẫu nhiên: là loại nhiễu không tuân theo bất kỳ quy luật nào và thường xuất hiện như vết bẩn không rõ nguyên nhân. Khắc phục nhiễu ngẫu nhiên thường đòi hỏi các phương pháp lọc và xử lý thông minh, vì không có mô hình cụ thể nào để áp

dụng. Các phép lọc và kỹ thuật xử lý hình ảnh có thể được sử dụng để giảm thiểu ảnh hưởng của nhiễu và tái tạo hình ảnh một cách chất lượng.

Việc hiểu rõ về các loại nhiễu này là quan trọng để lựa chọn và triển khai các phương pháp xử lý ảnh hiệu quả trong quá trình khôi phục và cải thiện chất lượng hình ảnh.



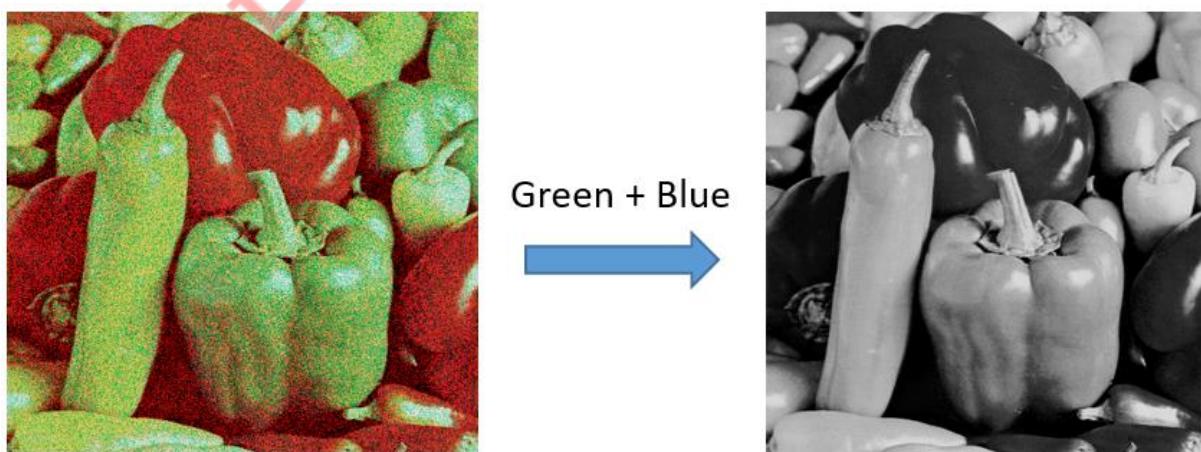
Hình 6: Các loại nhiễu trong ảnh

d. Chính mức xám

Nhằm khắc phục tính không đồng đều của hệ thống gây ra. Thông thường có 2 hướng tiếp cận:

Giảm số mức xám: Thực hiện bằng cách nhóm các mức xám gần nhau thành một bó. Trường hợp chỉ có 2 mức xám thì chính là chuyển về ảnh đen trắng. Ứng dụng: In ảnh màu ra máy in đen trắng.

Tăng số mức xám: Thực hiện nội suy ra các mức xám trung gian bằng kỹ thuật nội suy. Kỹ thuật này nhằm tăng cường độ mịn cho ảnh.



Hình 7: Ảnh xám

e. Trích chọn đặc điểm

Các đặc điểm của đối tượng được trích chọn tuỳ theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây: Đặc điểm không gian: Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn v.v.. Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v.v..)

Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng và do vậy rất hữu ích trong việc trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử la bàn, toán tử Laplace, toán tử “chéo không” (zero crossing) v.v..

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

f. Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity), được xác định một cách ang áng (vaguely defined) và có thể gán cho nó một tên gọi nào đó”. Ví dụ mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analysis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.

Hoặc phân loại không có mẫu (unsupervised classification hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.



Hình 8: Nén ảnh

1.2. Học máy

1.2.1. Tổng quan về học máy

Học máy là một lĩnh vực của trí tuệ nhân tạo (AI) mà mô hình máy tính được lập trình để tự động học và cải thiện từ dữ liệu mà nó xử lý. Mục tiêu chính của học máy là phát triển các thuật toán có khả năng tự học và áp dụng kiến thức mới mà không cần sự can thiệp trực tiếp của con người.

Machine learning cho phép máy tính giải quyết các tác vụ mà trước đây chỉ con người mới thực hiện được từ phiên dịch cho đến lái xe, học máy đã và đang tạo tiền đề cho sự bùng nổ của trí tuệ nhân tạo, giúp phần mềm nhận thức được thế giới vốn rất phức tạp và không thể dự đoán được.

Khái niệm học máy không hề mới mẻ, nhưng nó chỉ thực sự được ứng dụng trong doanh nghiệp khi internet trở nên phổ biến cùng những thành tựu gần đây trong lĩnh vực phân tích dữ liệu lớn và điện toán đám mây. Nguyên nhân là bởi việc huấn luyện thuật toán học máy tìm kiếm mẫu hình trong dữ liệu đòi hỏi rất nhiều tài nguyên điện toán và phải tiếp cận được dữ liệu lớn.



Hình 9: Học máy

1.2.2. Lịch sử phát triển

Học máy đã trải qua nhiều giai đoạn và động lực khác nhau, từ các ý tưởng ban đầu trong thập kỷ 1940 và 1950 đến sự phổ biến và tiến bộ đáng kể trong thế kỷ 21. Dưới đây là một số điểm quan trọng trong lịch sử phát triển của học máy:

- Thập kỷ 1940 và 1950

Những ý tưởng đầu tiên về máy tính và trí tuệ nhân tạo xuất hiện, với những nhà nghiên cứu như Alan Turing đưa ra khái niệm về máy tính có khả năng học.

- Thập kỷ 1960 và 1970

Các nghiên cứu về học máy trở nên chính thức hơn với ý tưởng về học máy. Hệ thống Dendral, một trong những hệ thống học máy đầu tiên, được phát triển để phân tích dữ liệu hoá học

- Thập kỷ 1980 và 1990

Các thuật toán truyền thống như Support Vector Machines và Decision Trees bắt đầu được phát triển và áp dụng rộng rãi. Tuy nhiên, tiến triển chủ yếu dựa vào các phương pháp thống kê.

- Thập kỷ 1990 đến thế kỷ 21

Sự tăng mạnh mẽ về khả năng tính toán và lượng dữ liệu không lò đã định hình lại học máy. Mạng nơ-ron được tái khám phá, đặt nền tảng cho sự phát triển của học sâu.

- Thập kỷ 2000 chứng kiến sự phổ biến của các phương pháp học máy trong các ứng dụng thực tế, nhận dạng khuôn mặt, dịch ngôn ngữ tự nhiên và tư vấn sản phẩm.

Sự tiến bộ trong việc thu thập dữ liệu và mô hình, cùng sự xuất hiện các framework học sâu như TensorFlow và PyTorch, đã giúp học sâu trở nên ngày càng mạnh mẽ và linh hoạt.

- Hiện nay

Học máy và học sâu đã trở thành một lực lượng mạnh mẽ trong nhiều ngành, từ công nghiệp, y tế tài chính đến giáo dục và rất nhiều lĩnh vực khác.

Cộng đồng nghiên cứu học máy và trí tuệ nhân tạo đang nghiên cứu và phát triển những ứng dụng mới, đồng thời đối mặt với các thách thức như đạo đức và an ninh dữ liệu.

Lịch sử phát triển của học máy thể hiện sự tiếp tục và đổi mới, từ những ý tưởng ban đầu cho đến những tiến bộ đáng kể nhờ vào sự tiến bộ về tính toán, thu thập dữ liệu và phương pháp học máy mới.

1.2.3. Các loại học máy

Machine learning thường được chia thành 4 nhóm chính: học giám sát, học không giám sát, học nửa giám sát và học tăng cường.

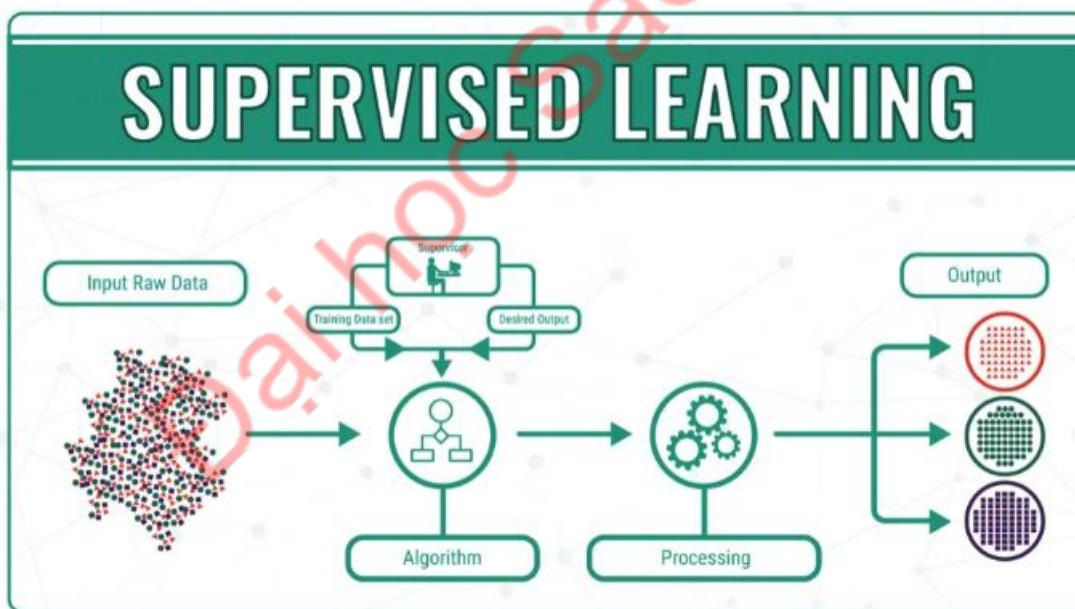
Học giám sát : đây là phương pháp dạy máy thông qua ví dụ.

Trong quá trình học giám sát, các hệ thống sẽ được truyền đạt những lượng lớn dữ liệu có gắn nhãn, ví dụ như hình ảnh những con số viết tay được ghi chú để chỉ rõ chúng tương ứng với con số nào. Khi được truyền đạt đủ ví dụ, một hệ thống học giám sát sẽ học được cách nhận biết những cụm điểm ảnh và hình dáng gắn liền với mỗi con số, và cuối cùng sẽ có thể nhận biết được những con số viết tay, phân biệt được chính xác những con số như 9 và 4, hay 6 và 8.

Tuy nhiên, huấn luyện những hệ thống như vậy đòi hỏi lượng dữ liệu gắn nhãn rất lớn – một số hệ thống cần đến hàng triệu ví dụ mới thuần thục một tác vụ nhất định!

Kết quả là, bộ dữ liệu được sử dụng để huấn luyện các hệ thống đó có thể cực “khủng” (bộ dữ liệu Open Images của Google có đến khoảng 9 triệu ảnh, kho video gắn nhãn YouTube-8M có 7 triệu video gắn nhãn, và ImageNet, một trong những cơ sở dữ liệu đầu tiên thuộc loại này, có hơn 14 triệu ảnh được sắp xếp theo từng danh mục).

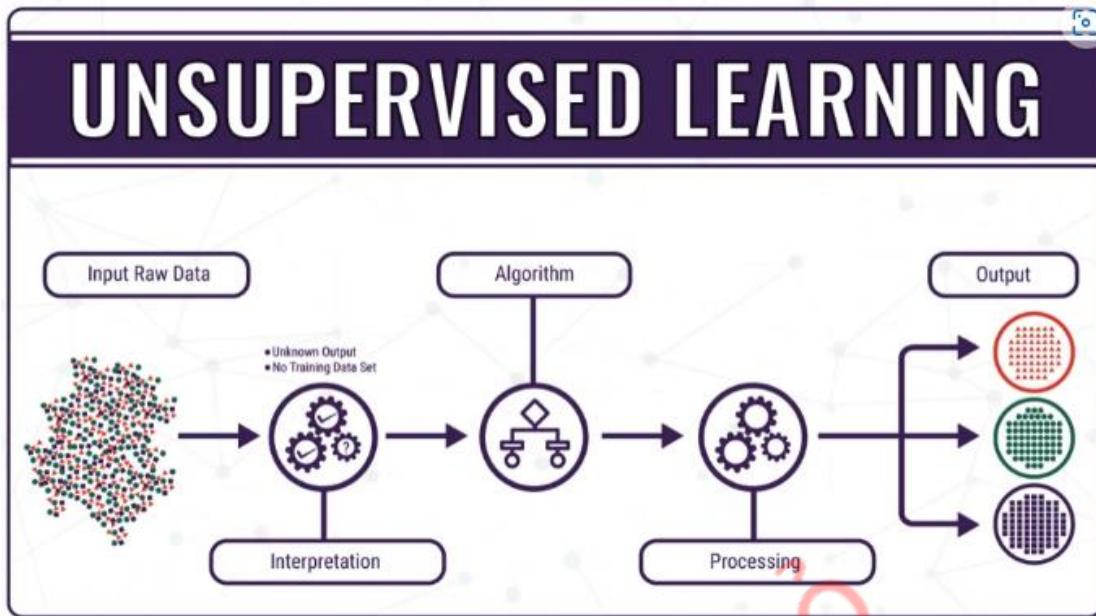
Quy trình gắn nhãn cho các bộ dữ liệu dùng trong học giám sát thường được thực hiện thông qua các dịch vụ tuyển dụng tự do, như Amazon Mechanical Turk. Ví dụ, ImageNet được xây dựng trong hơn 2 năm bởi gần 50.000 người, chủ yếu được tuyển dụng thông qua Amazon Mechanical Turk.



Hình 10: Mô hình học có giám sát

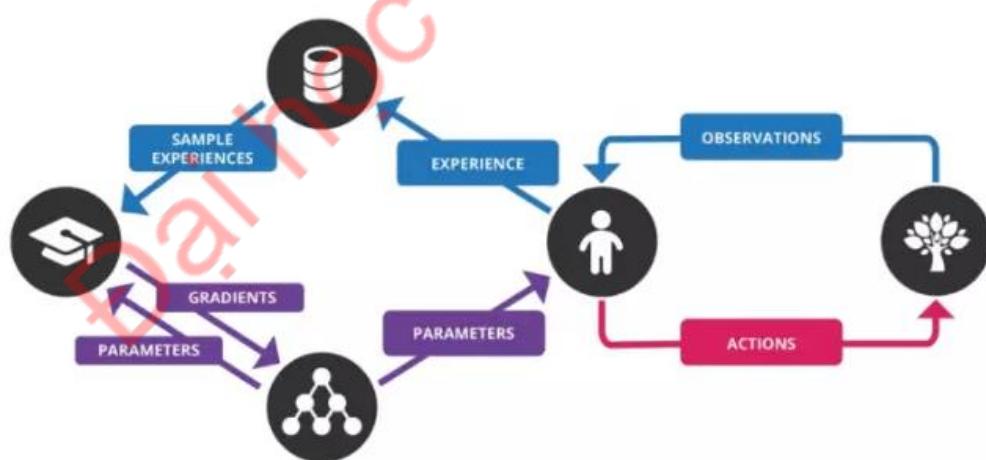
Học không giám sát: Hệ thống tự tìm ra cấu trúc dữ liệu chứ không cần các nhãn. Nó phân tích dữ liệu để phân cụm, phân loại mà không cần bất kỳ hướng dẫn nào về việc phân loại.

Ví dụ như Airbnb gom nhóm các nhà còn chõ cho thuê theo từng khu phố, hoặc Google News gộp các tin tức về chủ đề tương tự mỗi ngày. Các thuật toán học không giám sát không được thiết kế để lọc ra những loại dữ liệu cụ thể – chúng đơn giản là tìm kiếm dữ liệu có thể được gom nhóm thông qua những điểm tương đồng, hoặc tìm những điểm bất thường trong dữ liệu.



Hình 11: Mô hình học không giám sát

Học tăng cường: Hệ thống sẽ tương tác mạnh mẽ với môi trường, thông qua "thực nghiệm và sai lầm". Nó liên tục điều chỉnh quyết định của mình để tối đa hóa phần thưởng mà nó nhận được từ môi trường. Đây là cơ sở cho AI tự chơi game như cờ vây, cờ vua.



Hình 12: Mô hình học tăng cường

Học nửa giám sát: Kỹ thuật này dựa vào một lượng nhỏ dữ liệu gắn nhãn và một lượng lớn dữ liệu chưa gắn nhãn để huấn luyện hệ thống. Dữ liệu gắn nhãn được sử dụng để huấn luyện cơ bản một mô hình học máy, sau đó mô hình đã qua huấn luyện cơ bản này sẽ được sử dụng để gắn nhãn dữ liệu chưa gắn nhãn – một quy trình gọi là “gắn nhãn giả”. Mô hình sau đó sẽ được huấn luyện bằng tổng hợp dữ liệu gắn nhãn và gắn nhãn giả.

Hiệu suất học nêu giám sát thời gian qua đã được cải thiện khá nhiều nhờ các Mạng đối nghịch tạo sinh (GAN), tức các hệ thống học máy có thể sử dụng dữ liệu gắn nhãn để tạo ra dữ liệu hoàn toàn mới, từ đó lại sử dụng dữ liệu mới để giúp huấn luyện một mô hình học máy.

Các ứng dụng học máy

Một số ứng dụng chính của học máy:

- Nhận dạng hình ảnh và video
- Ngôn ngữ tự nhiên và xử lý ngôn ngữ tự nhiên
- Dự báo thị trường tài chính và dự đoán thời tiết
- Y tế : Chuẩn đoán bệnh và dự đoán cảnh báo y tế
- Hệ thống tư vấn cá nhân, quảng cáo định hình
- Xe tự lái và Robot hợp tác
- Quản lý chuỗi cung ứng

Những ứng dụng trên chỉ là một số ví dụ và học máy tiếp tục mở ra nhiều cơ hội mới trong nhiều lĩnh vực khác nhau.

1.2.4. Ứng dụng học máy trong thực tế

Học máy đang được triển khai rộng rãi trong nhiều lĩnh vực và mang lại nhiều ứng dụng thực tế có ảnh hưởng sâu rộng đến cuộc sống hàng ngày và các ngành công nghiệp. Dưới đây là một số ứng dụng học máy trong thực tế:

Google Translate: Học máy được sử dụng để dịch văn bản và lời nói giữa các ngôn ngữ tự nhiên với độ chính xác rất cao. Đây là công cụ tuyệt vời giúp thấu hiểu thế giới và giao tiếp bằng nhiều ngôn ngữ, kết nối với con người, nơi chốn, văn hóa.

Nhận diện khuôn mặt: Các hệ thống camera an ninh và mở khóa điện thoại bằng khuôn mặt đều dựa trên công nghệ nhận dạng khuôn mặt AI. Độ chính xác của các hệ thống này ngày càng cao, thậm chí có thể vượt qua con người. Đảm bảo tính tiện dụng, nhanh và khó sao chép cho người sử dụng

Netflix, Amazon,...: Học máy phân tích lịch sử tương tác và thói quen người dùng để đưa ra gợi ý sản phẩm, nội dung, hoặc âm nhạc phù hợp. Mỗi người dùng có sở thích, thói quen khác nhau. Bằng cách phân tích lịch sử tương tác các trang có thể đề xuất nội dung phù hợp với mỗi khách hàng.

Chẩn đoán y tế: AI trợ giúp bác sĩ chẩn đoán chính xác hơn dựa trên các xét nghiệm hình ảnh như X-quang, MRI. Hệ thống có thể phát hiện sớm bệnh ung thư chỉ với vài té bào khi mà mắt thường khó nhận biết.

Tesla Autopilot, Waymo: Ngày nay các hình thức tự lái xe được ứng dụng ngày càng phổ biến để hỗ trợ con người trong nhiều trường hợp đặc biệt : say rượu,... Học máy giúp xe tự lái phân tích dữ liệu từ cảm biến để hiểu môi trường xung quanh và đưa ra quyết định lái xe an toàn.

Facebook Ads, Google Ads: Học máy được sử dụng để dự đoán hiệu suất quảng cáo và tối ưu hóa chiến dịch quảng cáo trực tuyến để mỗi bài quảng cáo có độ nhận diện cao đối với người dùng

Siêu máy tính : Học máy hỗ trợ con người nghiên cứu, tính toán, dự báo thời tiết, sản xuất máy bay, tên lửa, vệ tinh... để chúng ta đạt được bước tiến mới trong ngành khoa học vũ trụ, khai phá những điều mới mẻ chưa từng biết tới, những vùng đất mới chưa thể đặt chân đến.

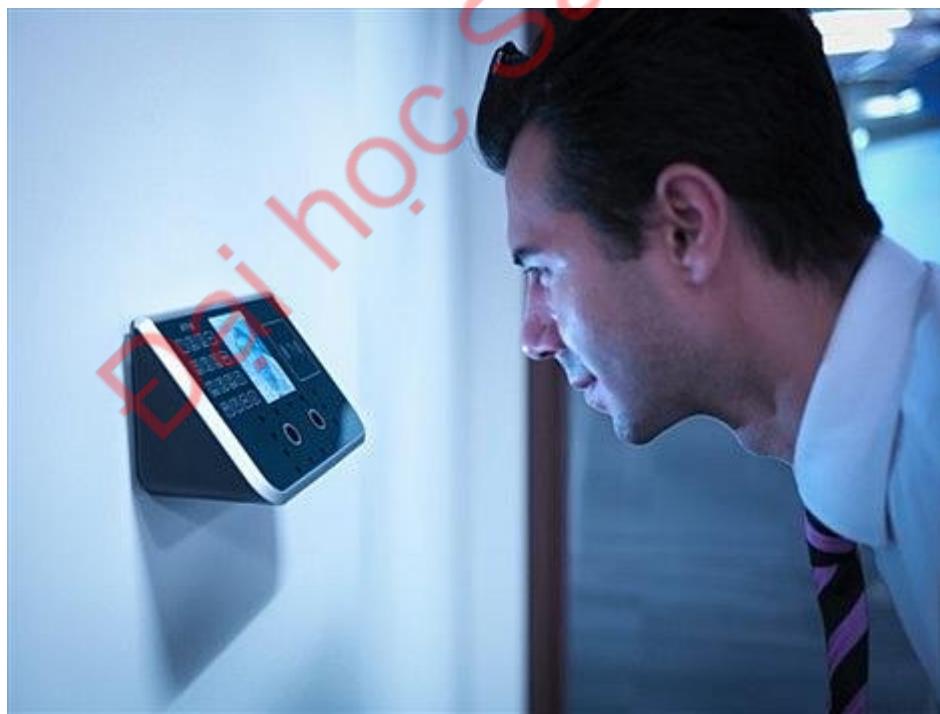
Ngoài ra, học máy còn được ứng dụng trong rất nhiều lĩnh vực khác : nông nghiệp thông minh, điện toán đám mây, tự động hóa quy trình sản xuất, giáo dục, tài chính,... Học máy đang dần thâm nhập mọi mặt của cuộc sống.

1.3. Ứng dụng của học máy và xử lý ảnh

Học máy và xử lý ảnh là một trong những kết hợp mạnh mẽ, mở ra nhiều ứng dụng quan trọng trong nhiều lĩnh vực khác nhau. Dưới đây là một số cách mà học máy được tích hợp vào xử lý ảnh:

1.3.1. Nhận dạng vật thể và khuôn mặt

Học máy có thể được sử dụng để nhận dạng và phân loại vật thể trong ảnh, từ đồ vật hàng ngày đến đối tượng phức tạp như ô tô hay máy bay. Các mô hình nhận dạng khuôn mặt cũng trở nên phổ biến trong việc xác thực và an ninh



Hình 13: Ví dụ nhận dạng khuôn mặt

1.3.2. Phân loại ảnh

Học máy có thể phân loại ảnh thành các nhóm hoặc danh mục khác nhau. Điều này có thể được sử dụng trong các ứng dụng như triển khai hệ thống thư viện ảnh tự động hoặc tạo chủ thích

1.3.3. Xử lý ảnh y khoa

Trong lĩnh vực y tế, học máy có thể giúp phân tích hình ảnh y khoa như cắt lớp MRI, phát hiện tế bào ung thư, và hỗ trợ trong việc chẩn đoán các vấn đề y tế khác.

1.3.4. Nhận dạng biểu đồ và chú thích hình ảnh

Trong lĩnh vực nghiên cứu và giáo dục, học máy có thể giúp nhận dạng và chú thích các phần trong biểu đồ khoa học, hình ảnh y học, các loại hình ảnh phức tạp khác.

1.3.5. Xử lý hình ảnh điều khiển xe tự lái

Trong xe tự lái, học máy được sử dụng để phân tích và hiểu hình ảnh từ cảm biến, giúp xe nhận biết các vật thể, biển báo giao thông và xác định con đường.

1.3.6. Chỉnh sửa ảnh và hiệu ứng nghệ thuật

Các ứng dụng chỉnh sửa ảnh như Adobe Photoshop sử dụng học máy để cải thiện chức năng như làm mờ, làm sáng, lọc màu, và loại bỏ đối tượng không mong muốn.

1.3.7. Xử lý dữ liệu hình ảnh trong dự án GIS

Trong dự án thông tin địa lý (GIS), học máy có thể được sử dụng để phân tích và phân loại dữ liệu từ hình ảnh vệ tinh, ảnh UAV (Unmanned Aerial Vehicle), giúp tạo ra bản đồ địa lý chính xác.

1.3.8. Phân loại dữ liệu Y sinh và phục hồi ảnh Y sinh

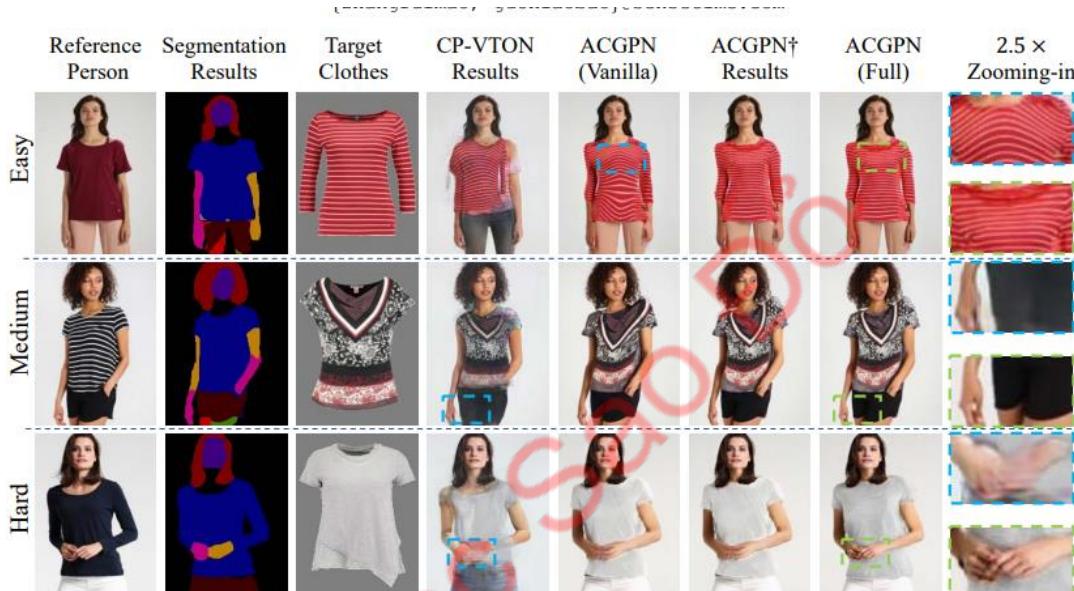
Học máy có thể giúp phân loại và phục hồi hình ảnh y sinh, từ hình ảnh chụp từ máy quét MRI đến hình ảnh từ siêu âm.

Những ứng dụng này chỉ là một số ví dụ và sự tích hợp giữa học máy và xử lý ảnh tiếp tục mở ra nhiều cơ hội.

CHƯƠNG 2 : PHÂN TÍCH MÔ HÌNH ỨNG DỤNG

2.1. Bài toán chuyển đổi trang phục

Thực tế vấn đề trực quan bằng hình ảnh nhằm mục đích chuyển hình ảnh quần áo mục tiêu sang một người hình người mẫu tham khảo đã trở thành chủ đề nóng trong những năm gần đây. Các phương pháp trước đây thường tập trung vào việc bảo tồn đặc tính của hình ảnh quần áo (ví dụ: họa tiết, biểu tượng, đường thêu) khi biến nó thành tư thế tùy ý của con người. Tuy nhiên, việc tạo ra các hình ảnh thay đổi trang phục ảo giống như ảnh chụp vẫn là một thách thức lớn khi các khớp cắn lớn và tư thế con người được thể hiện ở người tham chiếu.



Hình 14: Mô tả bài toán chuyển đổi trang phục

Để giải quyết vấn đề này, cần sử dụng Content Generating and Preserving Network (ACGPN). Cụ thể, ACGPN trước tiên dự đoán bộ cục ngữ nghĩa của hình ảnh tham chiếu sẽ được thay đổi sau khi thử (ví dụ: áo sơ mi dài tay → cánh tay, cánh tay → áo khoác), sau đó xác định xem nội dung hình ảnh của nó có cần được tạo hoặc bảo quản theo dự đoán hay không. bộ cục ngữ nghĩa, dẫn đến hình ảnh thử thực tế và chi tiết quần áo phong phú. ACGPN thường bao gồm ba mô-đun chính.

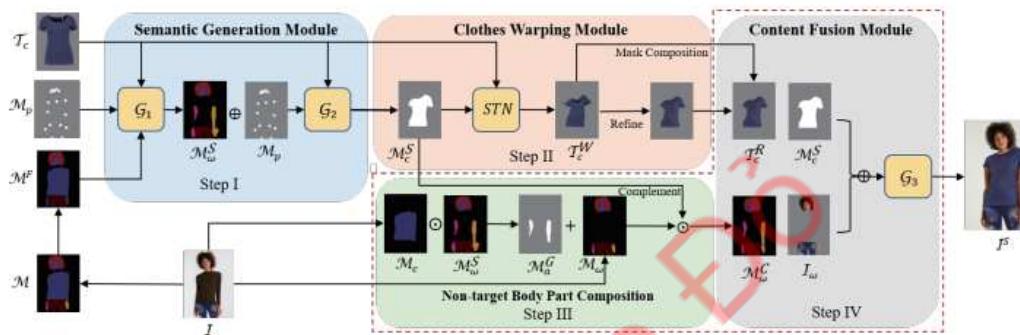
Đầu tiên, Semantic layout generation module (mô hình tạo phân đoạn bộ cục) của hình ảnh tham chiếu để dự đoán dần dần bộ cục của hình ảnh.

Thứ hai, Clothes warping module (mô hình tạo độ gấp của trang phục) sẽ làm cong hình ảnh quần áo theo bộ cục ngữ nghĩa được tạo ra, trong đó giới hạn chênh lệch bậc hai được đưa ra để ổn định quá trình cong vênh trong quá trình đào tạo.

Thứ ba, Module inpainting để tổng hợp nội dung tích hợp tất cả thông tin (ví dụ: hình ảnh tham chiếu, bộ cục ngữ nghĩa, quần áo bị biến dạng) để tạo ra từng bộ phận ngữ nghĩa của cơ thể con người một cách thích ứng. So với các phương pháp tiên tiến nhất, ACGPN có thể tạo ra hình ảnh chân thực với chất lượng cảm nhận tốt hơn nhiều và chi tiết phong phú hơn.

2.2. Mạng Content Generating and Preserving

ACGPN được đề xuất bao gồm ba mô-đun, như sau: Đầu tiên, Semantic layout generation (SGM), Clothes warping module (CWM), Content Fusion Module (CFM) tích hợp thông tin từ các mô-đun trước đó để xác định một cách thích ứng việc tạo hoặc bảo tồn các bộ phận riêng biệt của con người trong hình ảnh tổng hợp đầu ra. Thành phần bộ phận cơ thể có thể xử lý các tình huống khác nhau một cách linh hoạt trong tác vụ thử trong khi tính năng vẽ mặt nạ khai thác triệt để khả năng thích ứng bô cục của ACGPN khi xử lý các hình ảnh ở mức độ khó dễ, trung bình và khó.



Hình 15: Cấu trúc Mạng Content Generating and Preserving

2.2.1. Semantic Generation Module (SGM)

Mô-đun tạo ngữ nghĩa (SGM) được sử dụng để tách vùng quần áo mục tiêu cũng như bảo toàn các bộ phận cơ thể (tức là cánh tay) của người mà không làm thay đổi tư thế và các chi tiết còn lại trên cơ thể con người bằng cách chỉ đưa trực tiếp hình dạng cơ thể thô vào mạng, dẫn đến mất chi tiết bộ phận cơ thể. Để giải quyết vấn đề này, cơ chế tạo mặt nạ được áp dụng trong mô-đun này để tạo ra sự phân đoạn ngữ nghĩa của các bộ phận cơ thể và vùng quần áo mục tiêu một cách chính xác.

Cho một hình ảnh tham chiếu I và mặt nạ M tương ứng của nó, hình ảnh cánh tay M_a và thân M_t trước tiên được hợp nhất thành một khu vực không thể phân biệt được, dẫn đến bản đồ MF là một trong những đầu vào của SGM. Chiến lược, mô-đun tạo mặt nạ thử trước tiên tổng hợp mặt nạ của các bộ phận cơ thể $MS \omega$ ($\omega = \{h, a, b\}$ (h : đầu, a : tay, b : quần áo dưới)). Sau đó huấn luyện một GAN $G1$ phân tích cơ thể để tạo ra $MS \omega$ bằng cách tận dụng thông tin từ bản đồ hợp nhất MF, bản đồ tư thế M_p và quần áo mục tiêu hình ảnh T_c . Sử dụng thông tin được tạo ra của các bộ phận cơ thể, bản đồ tư thế tương ứng và hình ảnh quần áo mục tiêu, có thể điều khiển được vùng quần áo ước tính. Trong giai đoạn thứ hai, $MS \omega$, M_p và T_c được kết hợp để tạo ra mặt nạ tổng hợp của quần áo $MS c$ của $G2$.

Để đào tạo SGM, cả hai giai đoạn đều áp dụng mạng đối thủ tạo ra có điều kiện (cGAN), trong đó cấu trúc U-Net được sử dụng làm bộ tạo trong khi bộ phân biệt đối xử được đưa ra trong pix2pixHD [43] được triển khai để phân biệt mặt nạ được tạo với mặt nạ sự thật mặt đất của chúng. Đối với mỗi giai đoạn, tổn thất CGAN có thể được tính bằng

$$L1 = \text{Ex.y} [\log(D(x, y))] + \text{Ex,z} [\log(1 - D(x, G(x, z)))]$$

Trong đó:

x biểu thị đầu vào và y là mặt nạ sự thật mặt đất.

z là nhiễu là kênh bổ sung của đầu vào được lấy mẫu từ phân phối chuẩn chuẩn.

Hàm mục tiêu tổng thể cho từng giai đoạn của mặt nạ thử được đề xuất mô-đun tạo được công thức là

$$L_m, L_m = \lambda_1 L_1 + \lambda_2 L_2$$

Trong đó:

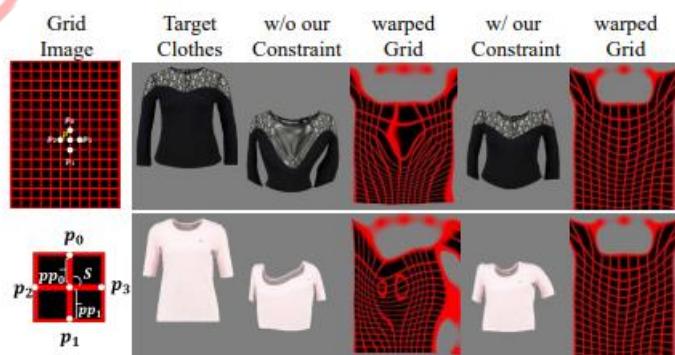
L_2 là mất entropy chéo theo pixel, giúp cải thiện chất lượng của mặt nạ tổng hợp từ trình tạo với kết quả phân đoạn ngữ nghĩa chính xác hơn.

λ_1 và λ_2 là các tham số đánh đổi cho từng số hạng tồn thất trong phương trình.

SGM hai giai đoạn có thể đóng vai trò là thành phần cốt lõi để hiểu chính xác cách bố trí các bộ phận cơ thể và quần áo trong thử nghiệm trực quan và hướng dẫn bảo quản thích ứng nội dung hình ảnh theo bố cục. SGM có hiệu quả đối với các nhiệm vụ khác cần phân vùng bố cục ngữ nghĩa.

2.2.2. Clothes Warping Module (CWM)

Tạo sự cong vênh của quần áo nhằm mục đích làm cho quần áo vừa với hình dạng của vùng quần áo mục tiêu với sự biến dạng tự nhiên về mặt trực quan theo tư thế của con người cũng như giữ lại đặc tính của quần áo. Tuy nhiên, chỉ cần huấn luyện Mạng chuyển đổi không gian (STN) và áp dụng Thin -Plate Spline (TPS) không thể đảm bảo chuyển đổi chính xác, đặc biệt khi xử lý các trường hợp cứng (tức là quần áo có kết cấu phức tạp và màu sắc phong phú), dẫn đến kết quả sai lệch và mờ.



Hình 16: So sánh kết quả hình ảnh của 2 mô hình

Việc chuyển đổi quần áo mục tiêu mà không có ràng buộc cho thấy sự biến dạng rõ ràng về hình dạng và sự lộn xộn không hợp lý trên kết cấu. với Tc và MSc làm đầu vào, sử dụng huấn luyện STN để học ánh xạ giữa chúng. Hình ảnh quần áo bị cong vênh được biến đổi bởi các tham số đã học từ STN, Đã kiểm tra, hình ảnh quần áo mục độ

khó thể giữ lại đầy đủ đặc tính của quần áo. Chúng tôi tin vào công thức của mình của CWM có hiệu quả trong việc thực thi tính cộng tuyến của các phép biến đổi affine cục bộ trong khi vẫn duy trì tính linh hoạt của cong vênh TPS, điều này có lợi cho các kết quả bị cong vênh thực tế và phù hợp về mặt hình học.

2.2.3. Content Fusion Module (CFM)

Ngoài việc phân vùng ảnh và căn chỉnh hình ảnh đối tượng người, việc thực hiện điều chỉnh bối cảnh trong nhiệm vụ thử trực quan vẫn là một thách thức lớn. Để đạt được mục đích này, cả vùng quần áo mục tiêu đều phải được hiển thị rõ ràng và các chi tiết ở tỷ lệ nhỏ về việc giữ lại các bộ phận cơ thể (tức là, khoảng trống ngón tay) là cần thiết để thích ứng Quần áo Mục tiêu không có Ràng buộc của chúng tôi với Ràng buộc của chúng tôi bị cong vênh Lưới bị vênh.

So sánh kết quả cong vênh STN có và không có chênh lệch bậc hai hạn chế được bảo tồn. Các phương pháp hiện tại thường sử dụng hình dạng cơ thể thô làm gợi ý để tạo ra hình ảnh thử cuối cùng và không tái tạo lại các chi tiết đẹp. Ngược lại, mô-đun hợp nhất nội dung (CFM) được đề xuất bao gồm hai bước chính, tức là Các bước 3 và 4. Đặc biệt, Bước 3 được thiết kế để duy trì hoàn toàn các bộ phận cơ thể không có mục tiêu cũng như bảo tồn thích ứng bộ phận cơ thể có thể thay đổi (tức là cánh tay). Bước 4 điền vào bộ phận cơ thể có thể thay đổi bằng cách sử dụng mặt nạ và hình ảnh được tạo từ các bước trước tương ứng bằng GAN tổng hợp dựa trên nội dung.

2.3. Tiền xử lý hình ảnh

2.3.1. Kiểm tra và chấp nhận ảnh

Trong quá trình phát triển hệ thống thay đổi trang phục dựa trên ảnh, quá trình kiểm tra và chấp nhận ảnh đóng một vai trò không thể phủ nhận trong việc đảm bảo chất lượng và tính đúng đắn của dữ liệu đầu vào. Điều này không chỉ đảm bảo tính an toàn mà còn đối mặt với thách thức của việc xử lý một lượng lớn dữ liệu ảnh đa dạng.

Đầu tiên và trước hết, hệ thống tiến hành kiểm tra định dạng của tệp ảnh để đảm bảo rằng chúng tuân theo các định dạng được hỗ trợ, như PNG, JPG hoặc JPEG. Việc này không chỉ giúp duy trì tính đồng nhất trong việc xử lý ảnh số mà còn đảm bảo tính tương thích với các phần mềm và thư viện xử lý ảnh. Loại bỏ các tệp tin ảnh bị hỏng hoặc không đúng định dạng. Đồng thời, kích thước của ảnh cũng được kiểm tra để đảm bảo rằng chúng có kích thước phù hợp với yêu cầu cụ thể của hệ thống, từ đó giúp duy trì hiệu suất ổn định trong quá trình xử lý. Kiểm tra độ sáng và màu sắc ảnh để đảm bảo rằng chúng đủ rõ ràng và có chất lượng tốt. Cần nhắc việc điều chỉnh độ sáng và màu sắc đồng nhất nếu cần thiết. Loại bỏ các ảnh không liên quan đến bài toán hoặc có thể gây nhiễu loạn cho mô hình. Nếu ảnh chứa nền, hãy đảm bảo rằng nền được chuẩn hóa hoặc loại bỏ để tập trung vào tăng phục. Kiểm tra và xử lý các ảnh có thể bị lỗi, mờ, hoặc không hoàn chỉnh để tránh làm bẩn dữ liệu đào tạo, chỉ chấp nhận các ảnh liên

quan đến định dạng trang phục bài toán khai thác: đối với dữ liệu lớn, kiểm tra và loại bỏ các ảnh trùng lặp để tránh làm chệch mô hình kiểm tra và xử lý nhiễu trong ảnh để đảm bảo rằng dữ liệu đầu vào làm tăng khả năng học của mô hình trước khi thực hiện bất kỳ biến đổi nào, tạo bản sao lưu của dữ liệu để tránh mất mát không mong muốn.

Ngoài ra, hệ thống thực hiện các biện pháp an ninh mạnh mẽ để ngăn chặn các loại tấn công phổ biến từ các tệp ảnh tải lên, nhằm bảo vệ hệ thống khỏi những nguy cơ an ninh tiềm ẩn. Sử dụng thư viện werkzeug.utils.secure_filename giúp đảm bảo an toàn tên tệp, từ đó ngăn chặn các hành động tấn công như injection và cross-site scripting. Điều này không chỉ đảm bảo tính toàn vẹn của dữ liệu mà còn giúp duy trì môi trường làm việc an toàn và tin cậy.

Đồng thời, hệ thống quản lý quá trình kiểm tra hiệu suất của ảnh để đảm bảo tính khả dụng của hệ thống. Việc này giúp tránh quá tải và giữ cho hiệu suất hệ thống ổn định, đáp ứng được yêu cầu của người dùng. Cuối cùng, quá trình kiểm tra còn xác minh tính chất tự nhiên của ảnh để tránh xử lý những hình ảnh được tạo ra bằng cách chỉnh sửa quá mức hoặc sử dụng công nghệ tạo ảnh giả mạo.

Những bước kiểm tra này không chỉ làm tăng tính an toàn và tính đúng đắn của dữ liệu đầu vào mà còn giúp hệ thống hoạt động một cách hiệu quả và ổn định. Chúng đảm bảo rằng chỉ những dữ liệu hợp lệ và đáng tin cậy nhất sẽ được sử dụng để thực hiện các bước xử lý và tạo ra trải nghiệm thử đồ ảo vô cùng ổn định và đáng tin cậy cho người dùng cuối. Điều này không chỉ làm tăng chất lượng của dịch vụ mà còn tạo nên sự tin tưởng và hài lòng từ phía người sử dụng.

2.3.2. *Resize và Crop Hình Ảnh*

Sau khi hệ thống đã kiểm tra và chấp nhận các tệp ảnh, quá trình tiền xử lý là một bước quan trọng tiếp theo để chuẩn bị dữ liệu đầu vào cho các bước xử lý hình ảnh phức tạp hơn trong quá trình thử đồ ảo. Bước này bao gồm việc thực hiện các phương pháp resize và crop hình ảnh, nhằm đảm bảo tính đồng nhất, hiệu quả, chất lượng của dữ liệu.

a. **Resize Hình Ảnh**

Quá trình resize hình ảnh thường bao gồm việc thay đổi kích thước của ảnh để đáp ứng nhu cầu cụ thể chẳng hạn giảm kích thước để tăng tốc độ xử lý, thích ứng với yêu cầu của mô hình học máy, hoặc hiển thị trên các thiết bị có kích thước màn khác nhau.

Trước tiên ta cần xác định kích thước mong muốn, điều này có thể bao gồm chiều rộng, chiều cao và hoặc cả hai chiều. Tiếp đó là chọn phương pháp giữa giữ nguyên tỷ lệ và thay đổi tỷ lệ. Giữ nguyên tỷ lệ đảm bảo cho tỷ lệ giữa chiều rộng và chiều cao được nguyên vẹn không làm méo hình ảnh. Thay đổi có thể theo yêu cầu.

Hệ thống thực hiện quá trình resize, một phương pháp quan trọng để điều chỉnh kích thước của ảnh ban đầu để nó đạt được kích thước mong muốn. Trong bài toán thử đồ ảo, cả hình ảnh của người mẫu và trang phục đều được chuyển đổi về kích thước 192 x 256 pixel (chiều rộng x chiều cao). Việc này không chỉ giúp duy trì tỷ lệ khung hình chính xác mà còn tối ưu hóa quá trình xử lý hình ảnh tiếp theo.

Quá trình resize không chỉ giúp chuyển đổi kích thước mà còn bảo toàn thông tin chính xác của hình ảnh, giúp duy trì tính chất tỷ lệ và hình dạng của người mẫu và trang phục. Điều này là quan trọng để đảm bảo rằng mô hình thử đồ sẽ có khả năng áp dụng trang phục một cách tự nhiên và chính xác trên người mẫu.

b. Crop Hình Ảnh

Bước tiếp theo trong quá trình tiền xử lý là crop hình ảnh, một phương pháp tác động chủ yếu vào việc cắt bớt các phần không cần thiết để tập trung vào phần quan trọng của hình ảnh. Đối với việc này ta cần chọn khu vực ảnh muốn giữ lại điều này có thể là đối tượng chính hoặc bất kỳ vùng nào quan tâm. Tiếp đó là xác định kích thước mong muốn. Ta có thể sử dụng thư viện ảnh hoặc các thư viện, công cụ chuyên dụng khác đưa hình ảnh và xác định toạ độ của hộp bao quanh khu vực mà bản thân muốn giữ lại. Sử dụng toạ độ của hộp bao để cắt ảnh theo kích thước và vị trí mong muốn. Đảm bảo giá trị pixel của ảnh sau khi crop nằm trong khoảng hợp lý. Đảm bảo rằng quá trình crop không làm mất quá nhiều thông tin quan trọng liên quan đến mục tiêu của mô hình. Nếu có đối tượng chính trong ảnh, giữ nguyên tỷ lệ khía cạnh để không làm biến đổi hình dạng đối tượng. Trong bối cảnh thử đồ ảo, việc này được thực hiện để chú trọng vào vùng hình ảnh chứa người mẫu và trang phục, giảm bớt các chi tiết không quan trọng xung quanh.

Quá trình crop giúp tối ưu hóa tập trung vào vùng quan trọng của hình ảnh, từ đó cải thiện sự chính xác và hiệu quả của mô hình thử đồ ảo. Bằng cách loại bỏ các phần không quan trọng, hệ thống tập trung vào những đặc điểm quan trọng nhất của người mẫu và trang phục, giúp tạo ra trải nghiệm thử đồ ảo một cách chân thực và tự nhiên.

c. Tối ưu hóa và cải thiện chất lượng

Những bước tiền xử lý này không chỉ giúp tối ưu hóa việc sử dụng tài nguyên hệ thống mà còn có ảnh hưởng lớn đến chất lượng và hiệu suất của hệ thống thử đồ ảo. Dữ liệu đầu vào được chuẩn hóa và tiêu chuẩn hóa kích thước, tạo nền tảng vững chắc cho các bước xử lý hình ảnh phức tạp tiếp theo.

2.3.3. Lưu trữ hình ảnh

Trong bài toán thay đổi trang phục, việc lưu trữ hình ảnh tiền xử lý là một phần quan trọng để đảm bảo dữ liệu sẵn sàng đào tạo mô hình hoặc triển khai vào ứng dụng thực tế. Đặt cấu trúc thư mục chính chuẩn, lưu trữ metadata, sử dụng cơ sở dữ liệu hoặc dịch vụ lưu trữ đám mây. Tiếp đó sao lưu định kỳ và quản lý phiên bản, chuẩn hóa và chuẩn bị cho mô hình, quản lý các biến thể bảo mật và quyền truy cập.

Đối với bài toán đầu tiên ta có, hệ thống lưu trữ hình ảnh của người mẫu và trang phục vào các thư mục riêng biệt, giữ nguyên tên file gốc của ảnh. Tệp ảnh được lưu trữ tại đường dẫn "data/test/image/" cho hình ảnh người mẫu và "data/test/cloth/" cho hình ảnh trang phục. Điều này giúp duy trì sự rõ ràng và dễ dàng theo dõi trong quá trình phát triển và quản lý dữ liệu.

Sau khi đã lưu trữ tệp ảnh, hệ thống sử dụng thư viện OpenCV để đọc lại chúng và thực hiện thao tác resize, chuyển đổi kích thước của hình ảnh về 192 x 256 pixel. Những bức ảnh đã được xử lý sẽ được lưu trữ lại tại cùng địa chỉ "data/test/image/" và "data/test/cloth/", ghi đè lên tệp ảnh gốc.

Bước tiếp theo trong quy trình lưu trữ là di chuyển ảnh nguyên thủy của người mẫu và trang phục vào thư mục "static/original/" để chúng có thể được hiển thị trực tiếp trên giao diện người dùng cuối (UI). Hình ảnh sau khi qua bước này sẽ được sử dụng để hiển thị so sánh giữa trước và sau quá trình thử đồ ảo.

Quy trình lưu trữ này không chỉ giúp duy trì sự tổ chức và quản lý dữ liệu hiệu quả mà còn đảm bảo rằng hình ảnh được chuẩn bị sẵn sàng để tham gia vào các bước xử lý phức tạp tiếp theo trong quá trình thử đồ ảo.

2.3.4. Tạo mặt nạ trang phục

Trong bài toán này việc tạo mặt nạ là một phần quan trọng tiền xử lý để xác định khu vực chứa trang phục trong ảnh.

Trước tiên ta cần chọn phương pháp. Phương pháp hình chữ nhật là xác định hình chữ nhật bao quanh trang phục, mặt nạ hình đối tượng là sử dụng mô hình segmentation để xác định. Sử dụng mô hình phân đoạn ảnh để dự đoán vùng trang phục trên ảnh. Có thể sử dụng mô hình được huấn luyện hoặc fine-tuning trên tập dữ liệu chứa thông tin về trang phục. Nếu trang phục có màu sắc đặc trưng độc đáo ta có thể sử dụng mô hình học máy đơn giản để dự đoán mặt nạ. Áp dụng các kỹ thuật xử lý ảnh để chọn ra các vùng có màu sắc hoặc đặc trưng tương ứng với trang phục. Huấn luyện một mô hình học máy để phân lớp ảnh thành các nhóm trang phục khác nhau. Sau đó sử dụng dự đoán của mô hình để tạo mặt nạ. Tiếp đó là sử dụng các phương pháp đồng nguyên. Chuyển đổi ảnh sang không gian màu phù hợp và áp dụng các ngưỡng để tạo ra mặt nạ, các phương pháp như ngưỡng màu hoặc ngưỡng độ sáng có thể được sử dụng. Lưu ý đánh giá hiệu suất của mô hình hoặc phương pháp tạo mặt nạ và tinh chỉnh chúng dựa trên dữ liệu đào tạo và thử nghiệm. Kiểm soát và đối phó với sai số của mô hình, đặc biệt khi có các trường hợp phức tạp như trang phục gần màu với môi trường xung quanh. Hiểu biết chi tiết về đặc điểm dữ liệu, bao gồm cả các loại trang phục ánh sáng và nâu, giúp cải thiện quá trình tạo mặt nạ.

Sau khi hình ảnh của người mẫu và trang phục đã được chuẩn bị, bước tiếp theo của quy trình là tạo ra mặt nạ trang phục. Mặt nạ này sẽ được sử dụng trong các bước xử lý tiếp theo để tính toán và áp dụng hiệu ứng thử đồ ảo một cách chính xác.

Hệ thống sử dụng một mô-đun đặc biệt để tạo mặt nạ cho trang phục, được triển khai trong hàm cloth_masking. Bằng cách này, mọi pixel trong hình ảnh trang phục sẽ được gán giá trị nhất định tùy thuộc vào việc chúng thuộc về phần nào của trang phục. Mặt nạ này sẽ chứa thông tin về vị trí và hình dạng của trang phục, làm cơ sở cho việc kết hợp hình ảnh của trang phục với hình ảnh của người mẫu một cách tự nhiên.

Kết quả của bước này là một mặt nạ được lưu trữ tại đường dẫn "data/test/cloth-mask/", giữ nguyên tên file của hình ảnh trang phục. Mặt nạ này sẽ được sử dụng trong các bước xử lý sau đó, cung cấp thông tin quan trọng để tạo ra hiệu ứng thử đồ ảo trên hình ảnh của người mẫu. Điều này đóng vai trò quan trọng trong việc tạo ra ảnh ghép chất lượng và chân thực.

2.3.5. Chuyển đổi và masking hình ảnh

Sau bước kiểm tra, hình ảnh của người mẫu và trang phục được chuyển đổi để đảm bảo chúng có kích thước thống nhất, là 192x256 pixels. Việc này giúp tạo ra sự đồng đều trong kích thước giữa hai hình ảnh, làm cho việc ghép ảnh trở nên dễ dàng và chính xác hơn.

Sau khi kích thước đã được chuẩn hóa, hệ thống tiến hành tạo mặt nạ cho trang phục. Mặt nạ này sẽ xác định vị trí và hình dạng của trang phục trên hình ảnh. Quá trình này giúp tạo ra một lớp mặt nạ trang phục, có thể được sử dụng để áp dụng các hiệu ứng và thay đổi trực tiếp lên trang phục.

Tiếp theo, hình ảnh của người mẫu được chuyển đổi và mask bằng cách sử dụng thông tin từ mặt nạ trang phục và mặt nạ của người mẫu. Quá trình này tạo ra một mặt nạ kết hợp, biểu diễn một cách chính xác vị trí của người mẫu và trang phục trên hình ảnh. Mặt nạ kết hợp này sẽ định hình hình dạng cuối cùng của trang phục trên người mẫu và sẽ được sử dụng trong các bước thử đồ tiếp theo để tạo ra hình ảnh kết quả chân thực và thẩm mỹ.

Quá trình này không chỉ đảm bảo tính đồng đều trong kích thước ảnh mà còn tạo ra một biểu diễn chính xác về vị trí và hình dạng của trang phục trên người mẫu, tạo ra cơ sở cho việc thực hiện thử đồ ảo hiệu quả và thú vị.

2.3.6. Đồng bộ hóa dữ liệu

Sau khi các bước trước đã tạo ra các mặt nạ chính xác cho trang phục và người mẫu, hệ thống tiếp tục quá trình đồng bộ hóa dữ liệu. Điều này đặc biệt quan trọng để đảm bảo rằng thông tin về hình dạng cơ thể và trang phục được tích hợp một cách chính xác và tự nhiên.

Kích thước ảnh được định rõ để đảm bảo chúng cùng kích thước, điều này quan trọng khi đào tạo mô hình học máy.

Đảm bảo giá trị pixel thuộc khoảng nhất định: chuẩn hóa giá trị pixel của ảnh để nằm trong một khoảng nhất định chẳng hạn từ 0 đến 1 hoặc từ -1 đến 1. Điều này giúp mô hình học máy hội tụ nhanh hơn và ổn định hơn.

Nếu sử dụng mặt nạ để xác định vùng chứa trang phục, đảm bảo rằng mặt nạ cũng được đồng bộ hoá với ảnh, nếu có thông tin được metadata về ảnh đảm bảo rằng các thông tin này được đồng bộ hoá và quản lý một cách chặt chẽ.

Đồng bộ hoá các phép biến đổi đảm bảo rằng chúng được áp dụng một cách đồng đều cho tất cả các ảnh, đảm bảo định dạng đầu vào sao cho phù hợp. Sử dụng các hàm

trong thư viện xử lý ảnh để thực hiện đồng bộ hoá một cách thuận tiện và hiệu quả, kiểm tra kết quả để đảm bảo rằng dữ liệu được đồng bộ hoá đúng cách và không làm mất thông tin quan trọng.

Dữ liệu từ các bước trước, bao gồm cả mặt nạ trang phục, mặt nạ người mẫu, và hình ảnh đã được chuyển đổi và mask, được đồng bộ hóa để tạo ra một biểu diễn thống nhất của cả trang phục và người mẫu. Quá trình này giúp kết hợp thông tin từ hai nguồn để tạo ra một hình ảnh thử đồ ảo chân thực, trong đó trang phục được hiển thị một cách tự nhiên trên cơ thể người mẫu.

Đồng bộ hóa dữ liệu không chỉ tạo ra sự nhất quán về thông tin giữa trang phục và người mẫu mà còn đảm bảo tính chính xác trong việc tái tạo chi tiết nhỏ nhất của trang phục trên hình ảnh kết quả. Quá trình này đóng vai trò quan trọng trong việc tạo ra trải nghiệm thử đồ ảo độc đáo và chân thực cho người dùng, giúp họ có cái nhìn rõ ràng và chân thực về cách trang phục sẽ trông như khi mặc trên cơ thể của họ.

2.3.7. Resize và lưu trữ hình ảnh đã chuyển đổi

Sau quá trình đồng bộ hóa dữ liệu, hình ảnh trang phục và người mẫu được đưa vào bước resize để đảm bảo kích thước chính xác cho quá trình thử đồ ảo. Bước này không chỉ giúp tối ưu hóa kích thước của hình ảnh mà còn đảm bảo tính nhất quán và chất lượng hình ảnh.

Hệ thống thực hiện quá trình resize bằng cách chuyển đổi kích thước hình ảnh trang phục và người mẫu về kích thước mong muốn, thường là 192 x 256 (width x height). Quá trình này không chỉ giúp chuẩn hóa kích thước ảnh mà còn tạo ra một định dạng chung cho dữ liệu, giúp quá trình xử lý tiếp theo diễn ra một cách hiệu quả.

Sau khi resize hoàn tất, hình ảnh được lưu trữ lại để sử dụng trong các bước thử đồ và tạo ảnh kết quả. Việc lưu trữ hình ảnh này có ý nghĩa quan trọng trong việc bảo quản dữ liệu và tạo ra một bản sao chính xác của hình ảnh chuyển đổi, đảm bảo rằng quá trình thử đồ và hiển thị kết quả sẽ diễn ra một cách mượt mà và chính xác.

2.3.8. Tối ưu hóa và mở rộng

Sau bước resize và lưu trữ hình ảnh, hệ thống thực hiện các quy trình tối ưu hóa và mở rộng để chuẩn bị dữ liệu cho quá trình xử lý tiếp theo. Bước tối ưu hóa này đảm bảo rằng dữ liệu lành mạnh và sẵn sàng cho các mô hình máy học, đồng thời giảm bớt tình trạng trùng lặp và không mong muốn.

Quá trình này có thể bao gồm việc tạo mặt nạ cho trang phục, nơi mà một mặt nạ độc lập được tạo ra để chỉ bao gồm phần trang phục trong hình ảnh. Mặt nạ này sẽ được sử dụng trong các bước thử đồ và tạo ảnh kết quả, giúp tạo ra một hiệu ứng mượt mà và tự nhiên.

Sau đó, hệ thống có thể thực hiện chuyển đổi hình ảnh, trong đó dữ liệu được chuẩn bị để áp dụng các mô hình máy học như mô hình tìm kiếm người mẫu và mô hình thử đồ ảo. Quy trình này bao gồm việc tạo ra các tệp tin đầu vào và chuẩn bị dữ liệu để có thể tích hợp chúng vào các mô hình được triển khai.

Từ bước này, dữ liệu đã được tối ưu hóa và mở rộng sẵn sàng để áp dụng vào các mô hình thử đồ và tạo ra trải nghiệm thử đồ ảo tốt nhất cho người dùng cuối. Quá trình này không chỉ giúp cải thiện chất lượng và hiệu suất mô hình mà còn đảm bảo tính ổn định và mở rộng của hệ thống trải nghiệm thử đồ ảo.

2.3.9. Xử lý lỗi

Quá trình xử lý ảnh trong hệ thống cũng cần tập trung vào việc quản lý và xử lý lỗi một cách linh hoạt và hiệu quả. Điều này đặc biệt quan trọng để đảm bảo tính ổn định của ứng dụng và tránh các vấn đề không mong muốn khi người dùng tương tác với hệ thống.

Hệ thống có thể triển khai các cơ chế kiểm tra lỗi để đảm bảo rằng dữ liệu đầu vào từ người dùng được kiểm tra kỹ lưỡng và tuân thủ các quy tắc xử lý hợp lý. Nếu phát hiện lỗi, hệ thống sẽ cung cấp thông báo lỗi chính xác và dễ hiểu để thông báo cho người dùng về vấn đề cụ thể.

Ngoài ra, quá trình xử lý lỗi cũng có thể bao gồm việc đảm bảo an toàn và bảo mật dữ liệu. Các biện pháp bảo mật có thể được tích hợp để ngăn chặn các cuộc tấn công từ bên ngoài và bảo vệ thông tin nhạy cảm của người dùng.

Thông qua các biện pháp xử lý lỗi kỹ thuật số và an toàn, hệ thống không chỉ mang lại trải nghiệm ổn định và an toàn cho người dùng mà còn giúp duy trì uy tín và độ tin cậy của ứng dụng thử đồ ảo trong thời gian dài.

2.4. Áp dụng học máy vào bài toán Try-on

2.4.1. Xử lý dữ liệu và chuẩn bị

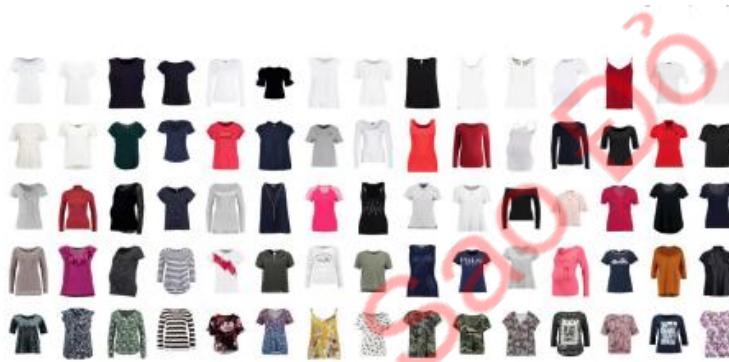
Trước hết, quá trình thu thập dữ liệu hình ảnh người và quần áo là một bước quan trọng trong việc xây dựng mô hình thử đồ dựa trên học máy. Để đảm bảo tính đa dạng và đại diện của dữ liệu, chúng tôi đã tiến hành một chiến dịch thu thập rộng lớn từ nhiều nguồn khác nhau.

Dữ liệu hình ảnh người được thu thập từ các nguồn công cộng và bao gồm nhiều loại người với đặc điểm về hình dáng, giới tính, và phong cách thời trang khác nhau. Để đảm bảo tính đa dạng, chúng tôi đã tập trung vào việc bao quát cả nền văn hóa và thị trường thời trang toàn cầu, từ quần áo hàng ngày đến trang phục truyền thống.

Đối với dữ liệu hình ảnh quần áo, chúng tôi đã tích hợp một loạt các nguồn, bao gồm cả hình ảnh từ các cửa hàng thời trang trực tuyến, bộ sưu tập thời trang nổi tiếng, và cả các ảnh độc đáo từ các sự kiện thời trang và triển lãm. Mục tiêu là có được một bộ dữ liệu phong phú về loại quần áo và phong cách.

Chúng tôi đã đặc biệt chú trọng đến việc thu thập dữ liệu chính xác và đầy đủ về mô hình người và chi tiết của quần áo. Điều này bao gồm cả việc xác định các điểm chính trên cơ thể người như đầu, vai, cổ, và việc gắn nhãn chi tiết của quần áo như dạng cổ áo, dài ngắn, màu sắc, và hoa văn.

Tập dữ liệu thu thập chứa khoảng 19.000 cặp hình ảnh phụ nữ dáng đứng phía trước. Có 16253 cặp đã được làm sạch, được chia thành tập huấn luyện và tập xác thực với cặp tương ứng là 14221 và 2032, và đã được sắp xếp lại các hình ảnh trong bộ xác thực thành các cặp không ghép đôi làm bộ thử nghiệm.



Hình 17: Tập dữ liệu huấn luyện

a. Tiền xử lý dữ liệu hình ảnh: cắt, chuyển đổi kích thước, và chuẩn hóa

Sau khi thu thập dữ liệu một cách kỹ lưỡng, bước tiền xử lý dữ liệu hình ảnh là quan trọng để chuẩn bị dữ liệu cho quá trình học máy. Chúng tôi đã thực hiện một loạt các bước tiền xử lý để đảm bảo dữ liệu đầu vào là đồng nhất và phù hợp với yêu cầu của mô hình thử nghiệm.

Đầu tiên, chúng tôi thực hiện quá trình cắt ảnh để tập trung vào vùng chính chứa người và quần áo, loại bỏ nhiễu và giảm kích thước dữ liệu. Việc này giúp tăng cường khả năng của mô hình trong việc tập trung vào các đặc trưng quan trọng của người và trang phục.

Tiếp theo, quá trình chuyển đổi kích thước đã được thực hiện để đồng nhất kích thước của tất cả hình ảnh trong tập dữ liệu. Sự đồng nhất này giúp đảm bảo rằng mô hình có thể hiểu được mối liên quan giữa các phần khác nhau của quần áo và cơ thể người, không phụ thuộc vào sự biến đổi kích thước.

Một trong những bước quan trọng khác trong tiền xử lý là chuẩn hóa dữ liệu. Việc này bao gồm việc điều chỉnh cường độ pixel và màu sắc của hình ảnh để giảm độ chênh và đảm bảo rằng mô hình được huấn luyện trên dữ liệu có tính chất thống nhất.

Điều này giúp đồng bộ hóa giữa hai loại dữ liệu, tạo điều kiện thuận lợi cho mô hình hiểu được mối quan hệ giữa cơ thể và trang phục.

Cuối cùng, chúng tôi đã đảm bảo rằng dữ liệu tiền xử lý được lưu trữ trong định dạng phù hợp để có thể được sử dụng dễ dàng trong quá trình đào tạo và kiểm thử mô hình. Các ảnh đã được lưu trữ sau tiền xử lý được chứa trong các thư mục và cấu trúc thích hợp để dễ dàng truy cập và quản lý.

b. Tạo dữ liệu đầu vào cho mô hình học máy

Sau khi hoàn thành quá trình tiền xử lý dữ liệu hình ảnh, bước tiếp theo quan trọng là tạo dữ liệu đầu vào chính xác và hiệu quả cho mô hình học máy. Chúng tôi đã thực hiện các bước này với mục tiêu tối ưu hóa việc đào tạo mô hình thử đồ và đảm bảo rằng nó có khả năng tổng quát hóa trên nhiều loại hình ảnh.

Đầu tiên, chúng tôi đã chia tách tập dữ liệu thành hai phần chính: tập dữ liệu huấn luyện và tập dữ liệu kiểm thử. Tập dữ liệu huấn luyện được sử dụng để mô hình học các mối quan hệ giữa hình ảnh người và quần áo, trong khi tập dữ liệu kiểm thử được sử dụng để đánh giá hiệu suất của mô hình trên dữ liệu mới.

Chúng tôi đã chú ý đặc biệt đến việc đảm bảo cân bằng giữa các loại hình ảnh người và quần áo trong cả tập dữ liệu huấn luyện và kiểm thử. Điều này giúp mô hình không bị chệch và có khả năng hiểu biết đầy đủ về mối quan hệ giữa cơ thể và trang phục ở mức độ tổng quát.

Mỗi bức ảnh trong tập dữ liệu đầu vào được gắn kết với các nhãn tương ứng để chỉ định vị trí và chi tiết của cơ thể người và quần áo. Các nhãn bao gồm các điểm chính trên cơ thể như đầu, vai, cổ, cũng như các chi tiết về quần áo như kiểu dáng, màu sắc, và kích thước. Việc này giúp mô hình có khả năng học được các đặc trưng quan trọng và đưa ra dự đoán chính xác.

Chúng tôi đã chọn định dạng dữ liệu phù hợp để đảm bảo tính linh hoạt và tiện lợi trong quá trình đào tạo. Cụ thể, dữ liệu đầu vào được tổ chức thành các batch để tối ưu hóa quá trình đào tạo và giảm thiểu tải lên bộ nhớ. Mỗi batch chứa một lượng ảnh đủ lớn để mô hình có thể cập nhật trọng số của nó một cách hiệu quả.

2.4.2. Tạo Mô Hình Học Máy

Lựa chọn kiến trúc mô hình: sử dụng mô hình GMM (Geometric Matching Module) và TOM (Try-on Module)

Trong quá trình xây dựng hệ thống thử đồ sử dụng học máy, việc lựa chọn kiến trúc mô hình đóng một vai trò quan trọng, quyết định tính linh hoạt và khả năng học của hệ thống. Chúng tôi đã chọn sử dụng mô hình GMM (Geometric Matching Module) và TOM (Try-on Module), hai thành phần quan trọng đóng vai trò chính trong quá trình thử đồ áo.

Geometric Matching Module (GMM):

GMM là một phần quan trọng của hệ thống, được thiết kế để hiểu biết về mối liên quan không gian giữa hình ảnh người và hình ảnh quần áo. Mục tiêu của GMM là tìm ra các biến đổi hình học để "đặt" trang phục lên cơ thể người một cách tự nhiên và chân thực. Điều này đòi hỏi khả năng nhận diện được các điểm chính trên cơ thể người và cấu trúc của quần áo.

GMM thực hiện quá trình tìm kiếm các điểm tương ứng giữa cơ thể người và quần áo, xác định được các biến đổi không gian cần thiết để đưa quần áo vào vị trí chính xác. Các phép biến đổi này có thể bao gồm di chuyển, xoay, và co giãn để phù hợp với hình dạng cơ thể người một cách chính xác. GMM sử dụng thông tin hình học này để tạo ra trang phục đã được "đặt" một cách hợp lý trên người.

Try-on Module (TOM):

Sau khi GMM đã thực hiện xong quá trình đặt trang phục lên cơ thể, TOM đảm nhận nhiệm vụ quan trọng khác - tạo ra hình ảnh cuối cùng của việc thử đồ ảo. TOM sử dụng thông tin về trang phục đã được đặt và hình ảnh cơ thể người để tạo ra hình ảnh tổng hợp, hiển thị một cách chân thực và tự nhiên.

Một trong những thách thức lớn của TOM là giữ cho chi tiết của trang phục và cơ thể người được kết hợp một cách hài hòa và tự nhiên. Điều này đòi hỏi TOM có khả năng học được các đặc trưng quan trọng trong hình ảnh và kết hợp chúng một cách mượt mà. Mô hình cần phải nhận biết được các biến giữa trang phục và cơ thể, và thực hiện các điều chỉnh hợp lý để tránh các hiện tượng mờ hoặc nặng nề.

Mô hình TOM cũng cần phải xử lý các thách thức liên quan đến ánh sáng, màu sắc, và góc nhìn. Nó phải tạo ra hình ảnh cuối cùng sao cho trang phục không chỉ "đặt" chính xác trên cơ thể mà còn phản ánh đúng cảm giác về màu sắc và chi tiết.

Đồng bộ hóa GMM và TOM:

Một phần quan trọng của quá trình thử đồ ảo là đồng bộ hóa GMM và TOM để đảm bảo rằng quá trình đặt trang phục và tạo ra hình ảnh tổng hợp diễn ra một cách mượt mà và hiệu quả. GMM và TOM cần phải làm việc cùng nhau để tạo ra trải nghiệm thử đồ tốt nhất cho người dùng, đồng thời đảm bảo tính chân thực và tự nhiên của kết quả cuối cùng.

2.4.3. Huấn luyện mô hình trên tập dữ liệu try-on

Huấn luyện mô hình trên tập dữ liệu thử đồ là một quá trình quan trọng để đảm bảo rằng mô hình có khả năng tổng quát hóa và hiệu quả trong việc thực hiện thử đồ ảo. Quá trình này đòi hỏi sự chú ý đặc biệt đến chi tiết, sự đa dạng của dữ liệu, và việc cân nhắc đến các yếu tố ảnh hưởng đến chất lượng kết quả cuối cùng.

Chuẩn bị dữ liệu huấn luyện:

Trước khi bắt đầu quá trình huấn luyện, chúng tôi đã thực hiện các bước chuẩn bị dữ liệu cần thiết. Đầu tiên, tập dữ liệu try-on đã được chia thành hai phần chính: tập dữ liệu huấn luyện và tập dữ liệu kiểm thử. Tập dữ liệu huấn luyện chứa một

lượng lớn các ảnh người và quần áo được đặt cùng nhau, trong khi tập dữ liệu kiểm thử sẽ được sử dụng để đánh giá hiệu suất của mô hình trên dữ liệu mới.

Chúng tôi đã đảm bảo rằng cả hai tập dữ liệu đều đa dạng và đại diện cho nhiều loại hình ảnh người và trang phục khác nhau. Điều này giúp mô hình hiểu biết được sự đa dạng trong hình dạng cơ thể, chi tiết trang phục, và mối liên quan giữa chúng.

Bảo đảm tính chân thực và tổng quát:

Trong quá trình huấn luyện, chúng tôi đã chú ý đến việc bảo đảm tính chân thực và tổng quát của mô hình. Điều này đòi hỏi sự chăm chỉ trong việc hiểu và mô phỏng đúng các đặc trưng của quần áo khi được đặt lên cơ thể. Mô hình cần phải học được cách xử lý độ chênh về kích thước, hình dạng, và màu sắc giữa các hình ảnh khác nhau.

Chúng tôi đã thực hiện việc đặt ra các nhiệm vụ và mục tiêu cụ thể trong quá trình huấn luyện, bao gồm việc đạt được sự linh hoạt trong việc đặt trang phục theo hình dạng cơ thể, giữ lại chi tiết quan trọng của trang phục, và hiển thị sự chính xác trong màu sắc và chi tiết.

Thiết lập tham số và kiến trúc mô hình:

Trước khi bắt đầu huấn luyện, chúng tôi đã chọn kiến trúc mô hình và thiết lập các tham số huấn luyện sao cho chúng phản ánh đúng yêu cầu của bài toán thử đồ. Mô hình GMM và TOM được tinh chỉnh để đảm bảo rằng chúng có khả năng học mối liên quan giữa hình ảnh người và hình ảnh quần áo một cách chính xác.

Tham số như hệ số học (learning rate), số lượng epoch, và kích thước batch đã được điều chỉnh để đạt được sự hội tụ nhanh chóng và đảm bảo tính ổn định của quá trình huấn luyện. Việc này đặt ra thách thức trong việc cân nhắc giữa sự học mạnh mẽ và tránh hiện tượng quá mức (overfitting).

Giám sát và đánh giá hiệu suất:

Trong suốt quá trình huấn luyện, chúng tôi đã tiến hành giám sát và đánh giá hiệu suất của mô hình. Việc này bao gồm việc sử dụng các metric như loss function, accuracy, và visual inspection để đảm bảo rằng mô hình đang học và cải thiện theo thời gian.

Chúng tôi đã sử dụng tập dữ liệu kiểm thử để đánh giá khả năng tổng quát hóa của mô hình, đảm bảo rằng nó không chỉ học "quá đà" trên tập dữ liệu huấn luyện mà còn đạt được kết quả tốt trên dữ liệu mới.

Hiệu chỉnh và tối ưu hóa:

Dựa vào kết quả đánh giá, chúng tôi đã thực hiện các bước hiệu chỉnh và tối ưu hóa để đảm bảo rằng mô hình đáp ứng được các yêu cầu của ứng dụng thử đồ ảo. Việc này bao gồm việc điều chỉnh các hyperparameter, cải thiện kiến trúc mô hình, và đối mặt với các vấn đề xuất hiện trong quá trình huấn luyện.

Đào tạo bảo mật và hiệu suất:

Cuối cùng, chúng tôi đã thực hiện quá trình đào tạo bảo mật và kiểm tra hiệu suất để đảm bảo rằng hệ thống thử đồ ảo có thể hoạt động một cách an toàn và hiệu quả trong môi trường thực tế. Việc này đòi hỏi sự đảm bảo về tính bảo mật của dữ liệu người dùng và đảm bảo rằng mô hình đáp ứng được yêu cầu về thời gian thực hiện thử đồ.

Chúng tôi đào tạo các mô-đun được đề xuất một cách riêng biệt và kết hợp chúng để cuối cùng tạo ra hình ảnh thử. Quần áo mục tiêu được sử dụng trong quá trình đào tạo giống như trong hình ảnh tham chiếu vì khó có được hình ảnh thực tế của kết quả thử. Mỗi mô-đun trong phương pháp đề xuất được huấn luyện cho 20 vòng lặp bằng cách đặt trọng số tồn thắt $\lambda_r = \lambda_s = 0,1$, $\lambda_1 = \lambda_2 = 1$ và kích thước lô 8. LearningRate được khởi tạo là 0,0002 và mạng được tối ưu hóa bởi trình tối ưu hóa Adam với siêu tham số $\beta_1 = 0,5$ và $\beta_2 = 0,999$. Tất cả các mã được triển khai bằng bộ công cụ học sâu PyTorch và GPU NVIDIA 1080Ti được sử dụng để huấn luyện mô hình.

2.4.4. Kiểm thử và Đánh giá Mô Hình

Chuẩn bị tập dữ liệu kiểm thử với các cặp hình ảnh người và quần áo

Quá trình kiểm thử và đánh giá mô hình là bước quan trọng để đảm bảo rằng hệ thống thử đồ ảo đáp ứng được yêu cầu và mang lại trải nghiệm chất lượng cho người dùng. Trong mục này, chúng tôi mô tả quá trình chuẩn bị tập dữ liệu kiểm thử, một yếu tố quan trọng để đánh giá hiệu suất và tính chất tổng quát của mô hình.

Thu thập Dữ liệu Kiểm thử:

Trước hết, chúng tôi đã tiến hành thu thập một tập dữ liệu kiểm thử đa dạng, chứa các cặp hình ảnh người và quần áo để đảm bảo rằng mô hình có khả năng thử nghiệm trên nhiều điều kiện và tình huống. Dữ liệu này có thể bao gồm hình ảnh của người mẫu đứng, ngồi, hoặc trong các tư thế khác nhau, cũng như đối với nhiều loại trang phục.

Chúng tôi đã chú trọng đến sự đa dạng về kiểu dáng và màu sắc của trang phục, đảm bảo rằng mô hình có khả năng thích ứng và đặt trang phục đúng cách trên nhiều loại trang phục. Đồng thời, chúng tôi cũng đảm bảo rằng dữ liệu kiểm thử có sự biến động đủ về môi trường ánh sáng, góc chụp, và độ đa dạng của cơ thể người mẫu.

Chia thành Tập dữ liệu Kiểm thử:

Sau khi thu thập dữ liệu, chúng tôi đã chia thành hai phần chính: tập dữ liệu kiểm thử chính và tập dữ liệu kiểm thử thử nghiệm. Tập dữ liệu kiểm thử chính được sử dụng để đánh giá hiệu suất chung của mô hình trên các trường hợp sử dụng thông thường, trong khi tập dữ liệu kiểm thử thử nghiệm được giữ lại để đối mặt với các tình huống đặc biệt và kiểm tra tính linh hoạt của mô hình.

Chúng tôi đã đảm bảo rằng cả hai tập dữ liệu đều bao gồm các cặp hình ảnh có đối ứng và thích hợp với mô hình GMM và TOM, có nghĩa là cả người và trang phục được biểu diễn một cách chi tiết và đầy đủ.

Phân loại Dữ liệu Kiểm thử:

Dữ liệu kiểm thử được phân loại vào các nhóm khác nhau tùy thuộc vào mục tiêu kiểm thử cụ thể. Các nhóm này có thể bao gồm:

Thử nghiệm Đa dạng: Bao gồm các tình huống đặc biệt như người mẫu với tư thế và vị trí không thông thường, và các kiểu trang phục độc đáo.

Thử nghiệm Hiệu suất: Đánh giá hiệu suất của mô hình dưới ánh sáng yếu, nền nôi có nhiễu, và các điều kiện khác nhau để đảm bảo rằng mô hình có thể hoạt động hiệu quả trong mọi tình huống.

Thử nghiệm Linh hoạt: Đối mặt với các biến động về màu sắc, kích thước, và kiểu dáng của trang phục để đảm bảo tính linh hoạt và đa dạng của mô hình.

Tiền xử lý Dữ liệu Kiểm thử:

Trước khi bắt đầu kiểm thử, chúng tôi đã thực hiện các bước tiền xử lý dữ liệu kiểm thử. Điều này có thể bao gồm việc cắt ảnh, chuyển đổi kích thước, và chuẩn hóa màu sắc để đảm bảo rằng dữ liệu đầu vào cho mô hình là đồng nhất và tương thích.

Đồng thời, chúng tôi đã xác định và loại bỏ những hình ảnh có chất lượng kém, mờ nhòe, hoặc không phản ánh đúng nội dung của tình huống thử đồ. Điều này giúp chúng tôi đảm bảo rằng mô hình được đánh giá trên dữ liệu chất lượng cao và phản ánh khả năng của nó.

Tạo Đánh giá Hiệu suất:

Cuối cùng, chúng tôi đã thiết lập các tiêu chí đánh giá hiệu suất dựa trên các yếu tố như chính xác của việc đặt trang phục, tính tự nhiên của kết quả tổng hợp, và khả năng tổng quát hóa trên nhiều dạng người và trang phục. Điều này bao gồm việc sử dụng các metric như IoU (Intersection over Union), loss function, và các phản hồi từ người dùng để đánh giá chất lượng cuối cùng của hệ thống.

Đánh giá hiệu suất của mô hình trên các tiêu chí như độ chính xác, độ chêch và thời gian thực thi

Sau khi chuẩn bị tập dữ liệu kiểm thử, quá trình quan trọng tiếp theo là đánh giá hiệu suất của mô hình trên nhiều tiêu chí khác nhau. Việc này giúp đảm bảo rằng mô hình không chỉ hoạt động chính xác trên tập dữ liệu huấn luyện mà còn đáp ứng được các yêu cầu của ứng dụng thử đồ trong môi trường thực tế.

Độ chính xác (Accuracy):

Một trong những tiêu chí chính để đánh giá hiệu suất của mô hình là độ chính xác trong việc đặt trang phục lên cơ thể người. Độ chính xác được đo bằng các metric

như IoU (Intersection over Union), một phương pháp đo lường sự giống nhau giữa hai khu vực trên hình ảnh.

Chúng tôi tiến hành đánh giá IoU trên tập dữ liệu kiểm thử để xác định mức độ trùng khớp giữa vị trí thực tế của trang phục và vị trí mà mô hình đặt. Giá trị IoU càng gần 1 cho thấy mô hình thực hiện đặt trang phục một cách chính xác và tự nhiên. Kết quả này cung cấp thông tin quan trọng về khả năng của mô hình trong việc điều chỉnh vị trí và hình dạng của trang phục dựa trên cơ thể người mẫu.

Độ chêch (Bias):

Để đảm bảo tính công bằng và đa dạng trong hiệu suất, chúng tôi cũng đánh giá độ chêch của mô hình trên các nhóm dữ liệu khác nhau. Độ chêch có thể xuất hiện trong nhiều khía cạnh, bao gồm độ chêch về kích thước, màu sắc, và kiểu dáng của trang phục.

Chúng tôi đo lường độ chêch bằng cách so sánh kết quả thử đồ trên các nhóm dữ liệu đặc biệt, chẳng hạn như trang phục với các màu sắc đậm và trang phục với các kiểu dáng đặc biệt. Kết quả này giúp chúng tôi xác định xem mô hình có thể đối mặt với những thách thức đặc biệt này một cách hiệu quả hay không.

Thời gian thực thi:

Khả năng thực hiện thử đồ ảo một cách nhanh chóng là một yếu tố quan trọng, đặc biệt trong các ứng dụng yêu cầu phản hồi người dùng thời gian thực. Chúng tôi đánh giá thời gian thực thi bằng cách đo lường thời gian mô hình mất để xử lý một cặp hình ảnh và tạo ra hình ảnh thử đồ kết quả.

Việc này đòi hỏi sự tối ưu hóa của thuật toán và kiến trúc mô hình để đảm bảo rằng mô hình có thể hoạt động một cách hiệu quả trong môi trường thực tế. Đồng thời, chúng tôi theo dõi thời gian thực thi trên nhiều loại hình ảnh và trang phục để đảm bảo tính linh hoạt của mô hình.

Đánh giá toàn diện:

Trên cơ sở các kết quả thu được từ các tiêu chí trên, chúng tôi thực hiện một đánh giá toàn diện về hiệu suất của mô hình. Điều này bao gồm việc xem xét kết quả IoU trên tất cả các nhóm dữ liệu, đảm bảo tính chính xác và đa dạng. Ngoài ra, chúng tôi đánh giá độ chêch và thời gian thực thi để đảm bảo rằng mô hình không chỉ hoạt động tốt trên dữ liệu kiểm thử chính mà còn đáp ứng được các yêu cầu về hiệu suất, thời gian.

Phản hồi từ người dùng:

Cuối cùng, chúng tôi thu thập phản hồi từ người dùng để có cái nhìn tổng quan về trải nghiệm sử dụng của họ. Phản hồi này không chỉ giúp chúng tôi hiểu rõ hơn về những khía cạnh cụ thể của mô hình mà còn cung cấp thông tin quan trọng về việc cải thiện và tối ưu hóa trong các phiên bản sau.

2.5. Môi trường xây dựng ứng dụng

2.5.1. Visual Studio Code

Giới thiệu VS Code

Visual Studio Code (hay còn gọi là VS Code) là một trình soạn thảo mã nguồn mở, được phát triển bởi Microsoft và hoạt động trên các nền tảng Windows, Linux và macOS. Nó hỗ trợ cho nhiều ngôn ngữ lập trình như JavaScript, Node.js, TypeScript, C++, Python, Java,... và có thể mở rộng để hỗ trợ thêm nhiều ngôn ngữ khác nhau.

Một số tính năng nổi bật của VS Code bao gồm:

Giao diện đơn giản và dễ sử dụng: VS Code có giao diện đơn giản, trực quan và dễ dàng để người dùng làm quen. Nó cung cấp các thanh công cụ và menu tương tự như các trình soạn thảo mã nguồn khác, giúp người dùng dễ dàng tìm hiểu và sử dụng.

Hỗ trợ đa nền tảng: VS Code có thể hoạt động trên các nền tảng Windows, Linux và macOS, giúp cho người dùng có thể làm việc trên nhiều máy tính khác nhau mà không phải lo lắng về sự tương thích.

Công cụ Debug và Git tích hợp: VS Code tích hợp sẵn các công cụ Debug và Git, giúp người dùng dễ dàng kiểm tra và gỡ lỗi code cũng như quản lý phiên bản của dự án.

Hỗ trợ đa ngôn ngữ: VS Code hỗ trợ cho nhiều ngôn ngữ lập trình và có thể mở rộng để hỗ trợ thêm nhiều ngôn ngữ khác nhau. Điều này giúp cho người dùng có thể làm việc trên nhiều dự án khác nhau mà không cần phải chuyển sang các trình soạn thảo mã nguồn khác.

Hệ sinh thái mở rộng đa dạng: VS Code có một hệ sinh thái mở rộng đa dạng với hàng ngàn các tiện ích và theme được cung cấp bởi cộng đồng GitHub. Người dùng có thể tùy chỉnh giao diện và tính năng của VS Code theo ý muốn.



Hình 18: Visual Studio Code

Hoạt động đa nền tảng

Một trong những lợi thế lớn nhất của VS Code là khả năng hoạt động trên nhiều nền tảng khác nhau. Điều này giúp cho người dùng có thể làm việc trên bất kỳ máy tính nào mà không cần phải lo lắng về sự tương thích. Nếu bạn làm việc trên nhiều máy tính

khác nhau, hoặc đang sử dụng cả Windows và macOS, VS Code sẽ là một công cụ hữu ích để giúp bạn tiết kiệm thời gian và công sức.

Đơn giản trong việc chỉnh sửa, xây dựng và gỡ lỗi

VS Code có giao diện đơn giản và dễ sử dụng, giúp cho người dùng có thể tập trung vào việc viết code mà không bị phân tâm bởi các tính năng phức tạp. Nó cũng tích hợp sẵn các công cụ Debug và Git, giúp cho quá trình chỉnh sửa, xây dựng và gỡ lỗi trở nên dễ dàng hơn. Bạn có thể dễ dàng theo dõi và sửa lỗi trong code của mình mà không cần phải chuyển sang các công cụ khác.

Sự hỗ trợ của cộng đồng GitHub

Visual Studio Code được phát triển bởi Microsoft và được cộng đồng GitHub hỗ trợ rất nhiều. Các tính năng mới và các tiện ích mở rộng được cập nhật liên tục từ cộng đồng, giúp cho VS Code trở nên đa dạng và phù hợp với nhiều ngôn ngữ lập trình. Bạn có thể tìm thấy hàng ngàn các tiện ích và theme khác nhau trên GitHub để tùy chỉnh giao diện và tính năng của VS Code theo ý muốn.

Kiến trúc cực mạnh mẽ và có thể mở rộng

Visual Studio Code được xây dựng trên nền tảng Electron, một framework mã nguồn mở cho phép viết ứng dụng desktop bằng HTML, CSS và JavaScript. Điều này giúp cho VS Code có một kiến trúc cực kỳ mạnh mẽ và linh hoạt, cho phép người dùng có thể mở rộng và tùy chỉnh theo ý muốn.

2.5.2. Môi trường ảo Python

Tổng quan về môi trường ảo Python

Môi trường ảo Python được sử dụng để cài đặt môi trường Python giữa các ứng dụng.

Khi cài đặt các thư viện và phụ thuộc vào hệ thống chính, sẽ dễ gây ra xung đột nếu phiên bản thư viện không tương thích giữa các ứng dụng.

Do đó, môi trường ảo được sinh ra để các ứng dụng có thể cài đặt các thư viện một cách độc lập.

Lợi ích của môi trường ảo Python

Môi trường ảo Python mang lại các lợi ích sau:

- Cài đặt môi trường giữa các ứng dụng, tránh xung đột thư viện
- Cài đặt các phiên bản thư viện khác nhau cho từng ứng dụng
- Dễ dàng sao chép môi trường cho các máy khác
- Không ảnh hưởng đến hệ thống chính khi thử nghiệm

Các công cụ tạo môi trường ảo

Có một số công cụ để tạo ra môi trường ảo Python, phổ biến nhất là:

- venv: Được tích hợp sẵn trong Python 3
- virtualenv: Một thư viện bên thứ 3, hỗ trợ cả Python 2 và 3

Trong chương trình này, chúng ta sẽ sử dụng module venv để tạo môi trường ảo.



Hình 19: Môi trường ảo của Python

Để tạo môi trường ảo Python bằng module venv, chúng ta thực hiện các bước sau:

Bước 1: Tạo thư mục môi trường ảo

Đầu tiên, chúng ta cần tạo một thư mục để chứa môi trường ảo.

Ví dụ, tạo thư mục [venv]

Bước 2: Tạo môi trường ảo

Sau đó, chạy lệnh sau để tạo môi trường ảo trong thư mục vừa tạo ở trên:

`py -m venv [venv]`

Lệnh này sẽ tạo ra môi trường ảo Python trong thư mục my_venv.

Lưu ý: Phython 2.7 thì cần cài thêm *virtualenv* hoặc nâng cấp lên Python 3 để dùng được tính năng này.

Bước 3: Kích hoạt môi trường ảo

Để sử dụng được môi trường ảo vừa tạo, chúng ta cần kích hoạt nó bằng cách chạy lệnh sau:

Trên Windows: \[venv]\Scripts\activate

Trên Linux/macOS: source my_venv/bin/activate

CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG

3.1. Xây dựng ứng dụng

3.1.1. Xây dựng giao diện

– Giao diện chọn hình ảnh

Giao diện cho phép người dùng kéo thả hình ảnh, tải hình ảnh từ máy lên trên server.

```
<div class="flex flex-wrap -m-2">
    <div class="p-1 xl:w-1/2 md:w-1/2 w-full">
        <center><label class="block text-lr font-medium text-white-700">
Cloth Image </label></center>
        <div
            class="mt-1 flex justify-center px-6 pt-5 pb-6 border-2 border-gray-300 border-dashed rounded-md">
            <div class="space-y-1 text-center">
                <svg class="mx-auto h-12 w-12 text-gray-400"
stroke="currentColor" fill="none"
viewBox="0 0 48 48" aria-hidden="true">
                <path
                    d="M28 8H12a4 4 0 00-4 4v20m32-12v8m0 0v8a4 4 0 01-4
4H12a4 4 0 01-4-4v-4m32-4l-3.172-3.172a4 4 0 00-5.656 0L28 28M8 32l9.172-
9.172a4 4 0 015.656 0L28 28m0 0l4 4m-24h8m-4-4v8m-12 4h.02"
stroke-width="2" stroke-linecap="round" stroke-
linejoin="round" />
                </svg>
                <div class="flex text-sm text-gray-600">
                    <!-- <label for="file-upload"
class="relative cursor-pointer rounded-md font-medium text-indigo-600 hover:text-indigo-500 focus-within:outline-none focus-within:ring-2
focus-within:ring-offset-2 focus-within:ring-indigo-500">
                    <span>Upload a file</span> -->
                    <input class="block w-full text-sm text-slate-500
file:mr-4 file:py-2 file:px-4
file:rounded-full file:border-0
file:text-sm file:font-semibold
file:bg-violet-50 file:text-violet-700
hover:file:bg-violet-100" id="file-upload" type="file"
name="cloth" class="sr-only">
                    </label>
```

```
<p class="pl-1">or drag and drop</p>
</div>
<p class="text-xs text-gray-500">PNG, JPG up to 10MB</p>
</div>
</div>

</div>
<div class="p-1 xl:w-1/2 md:w-1/2 w-full">
    <center> <label class="block text-lr font-medium text-white-700">
Model Image </label></center>
    <div
        class="mt-1 flex justify-center px-6 pt-5 pb-6 border-2 border-gray-300 border-dashed rounded-md">
        <div class="space-y-1 text-center">
            <svg class="mx-auto h-12 w-12 text-gray-400"
stroke="currentColor" fill="none"
viewBox="0 0 48 48" aria-hidden="true">
            <path
                d="M28 8H12a4 4 0 00-4 4v20m32-12v8m0 0v8a4 4 0 01-4
4H12a4 4 0 01-4-4v-4m32-4l-3.172-3.172a4 4 0 00-5.656 0L28 28M8 32l9.172-
9.172a4 4 0 015.656 0L28 28m0 0l4 4m-24h8m-4-4v8m-12 4h.02"
                stroke-width="2" stroke-linecap="round" stroke-
linejoin="round" />
            </svg>
            <div class="flex text-sm text-gray-600">
                <!-- <label for="file-upload"
                    class="relative cursor-pointer rounded-md font-medium text-
indigo-600 hover:text-indigo-500 focus-within:outline-none focus-within:ring-2 focus-
within:ring-offset-2 focus-within:ring-indigo-500">
                    <span>Upload a file</span> -->
                    <input class="block w-full text-sm text-slate-500
file:mr-4 file:py-2 file:px-4
file:rounded-full file:border-0
file:text-sm file:font-semibold
file:bg-violet-50 file:text-violet-700
                    hover:file:bg-violet-100" id="file-upload" type="file"
name="model" class="sr-only">
                </label>
```

```

<p class="pl-1">or drag and drop</p>
</div>
<p class="text-xs text-gray-500">PNG, JPG up to 10MB</p>
</div>
</div>
</div>
</div>

```

– Giao diện hiển thị kết quả hình ảnh

Sau khi người dùng đã tải hình ảnh lên kết quả sau quá trình thực thi sẽ là file hình ảnh dạng base64 được them vào src trong thẻ .

```

<div>
  { % if op %}
    <center style="color: white; font-size: x-large;">HERE'S IS YOUR RESULT
    😊 </center>

    <center>
      <div class="sm: w-3/4 mb-10 lg:mb-0 rounded-lg overflow-hidden">
        
      </div>
    </center>
  { % endif %}
</div>

```

– Giao diện chân website

Chân website là phần hiển thị hình ảnh logo website và thông tin tên tác giả thực hiện nội dung được đặt trong thẻ <footer>.

```

<footer class="text-gray-400 bg-gray-900 body-font">
  <div
    class="container px-5 py-24 mx-auto flex md:items-center lg:items-start
    md:flex-row md:flex-nowrap flex-wrap flex-col">
    <div class="w-64 flex-shrink-0 md:mx-0 mx-auto text-center md:text-left
    md:mt-0 mt-10">
      <a class="flex title-font font-medium items-center md:justify-start
      justify-center text-white">

```

```

height="50" width="50" style="border-radius: 50%;"/>
<span class="ml-3 text-xl">V-Cloth Assistant</span>
</a>
</div>

<div class="flex-grow flex flex-wrap md:pr-20 -mb-10 md:text-left text-center order-first">
<div class="lg:w-1/4 md:w-1/2 w-full px-4">
    <h2 class="title-font font-medium text-white tracking-widest text-sm mb-3">SWAYAM</h2>
    </div>
<div class="lg:w-1/4 md:w-1/2 w-full px-4">
    <h2 class="title-font font-medium text-white tracking-widest text-sm mb-3">PARTH</h2>
    </div>
<div class="lg:w-1/4 md:w-1/2 w-full px-4">
    <h2 class="title-font font-medium text-white tracking-widest text-sm mb-3">KEERTHI</h2>
    </div>
<div class="lg:w-1/4 md:w-1/2 w-full px-4">
    <h2 class="title-font font-medium text-white tracking-widest text-sm mb-3">NAVANEETH</h2>
    </div>
</div>
</div>

<div class="bg-gray-800 bg-opacity-75">
<div class="container mx-auto py-4 px-5 flex flex-wrap flex-col sm:flex-row">
    <p class="text-gray-400 text-sm text-center sm:text-left">© 2022 Cework Batch 3 —
```

```

<a href="https://twitter.com/knytneve" class="text-gray-500 ml-1" rel="noopener noreferrer"
    target="_blank">@Cework</a>
</p>
</div>
</div>
</footer>

```

3.1.2. Xây dựng chức năng

- Tạo mô hình mạng

Khởi tạo hàm với cấu trúc là mạng Unet sử dụng để phân đoạn hình ảnh. Bộ mã hóa (đường dẫn lấy mẫu xuống) ghi lại các đặc điểm phân cấp, trong khi bộ giải mã (đường dẫn lấy mẫu lên) khôi phục thông tin không gian. Bỏ qua các kết nối nối các bản đồ đặc điểm từ bộ mã hóa với các bản đồ từ bộ mã hóa bộ giải mã để lưu giữ thông tin chi tiết trong quá trình lấy mẫu.

Các lớp tích chập (self.conv1 đến self.conv9) được định nghĩa, mỗi lớp sau bởi chuẩn hóa, kích hoạt (ReLU), và trong một số trường hợp, dropout.

Các lớp upsampling (self.up6 đến self.up9) được định nghĩa để thực hiện

Lớp self.pool thực hiện max pooling.

Lớp self.drop là một lớp dropout với tỷ lệ dropout là 0.5.

Lớp self.sigmoid áp dụng hàm kích hoạt sigmoid.

Các phương thức print_network và init_weights được gọi để in thông tin mạng và khởi tạo trọng số.

```
def __init__(self, opt, input_nc, output_nc=13, norm_layer=nn.InstanceNorm2d):
    super(SegGenerator, self).__init__()
```

```
    self.conv1 = nn.Sequential(nn.Conv2d(input_nc, 64, kernel_size=3, padding=1),
        norm_layer(64), nn.ReLU(),
```

```
        nn.Conv2d(64, 64, kernel_size=3, padding=1), norm_layer(64),
        nn.ReLU())
```

```
    self.conv2 = nn.Sequential(nn.Conv2d(64, 128, kernel_size=3, padding=1),
        norm_layer(128), nn.ReLU(),
```

```
        nn.Conv2d(128, 128, kernel_size=3, padding=1),
        norm_layer(128), nn.ReLU())
```

```
    self.conv3 = nn.Sequential(nn.Conv2d(128, 256, kernel_size=3, padding=1),
        norm_layer(256), nn.ReLU(),
```

```
        nn.Conv2d(256, 256, kernel_size=3, padding=1),
        norm_layer(256), nn.ReLU())
```

```

    self.conv4 = nn.Sequential(nn.Conv2d(256, 512, kernel_size=3, padding=1),
norm_layer(512), nn.ReLU(),
                nn.Conv2d(512, 512, kernel_size=3, padding=1),
norm_layer(512), nn.ReLU())

    self.conv5 = nn.Sequential(nn.Conv2d(512, 1024, kernel_size=3, padding=1),
norm_layer(1024), nn.ReLU(),
                nn.Conv2d(1024, 1024, kernel_size=3, padding=1),
norm_layer(1024), nn.ReLU())

    self.up6 = nn.Sequential(nn.Upsample(scale_factor=2, mode='nearest'),
nn.Conv2d(1024, 512, kernel_size=3, padding=1),
norm_layer(512), nn.ReLU())
    self.conv6 = nn.Sequential(nn.Conv2d(1024, 512, kernel_size=3, padding=1),
norm_layer(512), nn.ReLU(),
                nn.Conv2d(512, 512, kernel_size=3, padding=1),
norm_layer(512), nn.ReLU())

    self.up7 = nn.Sequential(nn.Upsample(scale_factor=2, mode='nearest'),
nn.Conv2d(512, 256, kernel_size=3, padding=1),
norm_layer(256), nn.ReLU())
    self.conv7 = nn.Sequential(nn.Conv2d(512, 256, kernel_size=3, padding=1),
norm_layer(256), nn.ReLU(),
                nn.Conv2d(256, 256, kernel_size=3, padding=1),
norm_layer(256), nn.ReLU())

    self.up8 = nn.Sequential(nn.Upsample(scale_factor=2, mode='nearest'),
nn.Conv2d(256, 128, kernel_size=3, padding=1),
norm_layer(128), nn.ReLU())
    self.conv8 = nn.Sequential(nn.Conv2d(256, 128, kernel_size=3, padding=1),
norm_layer(128), nn.ReLU(),
                nn.Conv2d(128, 128, kernel_size=3, padding=1),
norm_layer(128), nn.ReLU())

    self.up9 = nn.Sequential(nn.Upsample(scale_factor=2, mode='nearest'),
nn.Conv2d(128, 64, kernel_size=3, padding=1), norm_layer(64),
nn.ReLU())
    self.conv9 = nn.Sequential(nn.Conv2d(128, 64, kernel_size=3, padding=1),
norm_layer(64), nn.ReLU(),
                nn.Conv2d(64, 64, kernel_size=3, padding=1), norm_layer(64),
nn.ReLU(),
                nn.Conv2d(64, output_nc, kernel_size=3, padding=1))

    self.pool = nn.MaxPool2d(2)
    self.drop = nn.Dropout(0.5)
    self.sigmoid = nn.Sigmoid()

```

```

    self.print_network()
    self.init_weights(opt.init_type, opt.init_variance)

def forward(self, x):
    conv1 = self.conv1(x)
    conv2 = self.conv2(self.pool(conv1))
    conv3 = self.conv3(self.pool(conv2))
    conv4 = self.drop(self.conv4(self.pool(conv3)))
    conv5 = self.drop(self.conv5(self.pool(conv4)))

    conv6 = self.conv6(torch.cat((conv4, self.up6(conv5)), 1))
    conv7 = self.conv7(torch.cat((conv3, self.up7(conv6)), 1))
    conv8 = self.conv8(torch.cat((conv2, self.up8(conv7)), 1))
    conv9 = self.conv9(torch.cat((conv1, self.up9(conv8)), 1))
    return self.sigmoid(conv9)

```

– Loại bỏ hình nền ảnh

Phương thức để loại bỏ nền của một hình ảnh bằng cách sử dụng API gói đến trang remove.bg và để biến đổi hình ảnh kết quả. Trong phần header của API cần thêm thông tin X-Api-Key là thông tin token sau khi tạo tài khoản trên trang remove.bg.

Phương thức transform thay đổi kích thước của hình ảnh đã xử lý thành chiều rộng và chiều cao đã cho. Hình ảnh đã thay đổi kích thước được dán lên một nền RGBA mới. Sau đó hình ảnh kết quả được lưu dưới dạng một tệp JPEG với nền trắng. Hình ảnh đã lưu được mở và chuyển đổi thành một mảng NumPy, sau đó được trả về.

```

def remove_bg(self, file_path:str,
api_key='qYVjUusWpt9ew8AUgAEEdsxv'):

    self.save_path = file_path[:-3] + '.png'
    response = requests.post(
        'https://api.remove.bg/v1.0/removebg',
        files={'image_file': open(file_path, 'rb')},
        data={'size': 'auto'},
        headers={'X-Api-Key': api_key},
    )
    pic = Image.open(file_path)
    self.o_width = np.asarray(pic).shape[1]
    self.o_height = np.asarray(pic).shape[0]
    try:
        self.o_channels = np.asarray(pic).shape[2]
    except Exception as e:
        print("Single channel image and error", e)

```

```

os.remove(file_path)
if response.status_code == requests.codes.ok:
    with open(self.save_path, 'wb') as out:
        out.write(response.content)
else:
    # raise("Error:", response.status_code, response.text)
    print(response.status_code)

self.o_image = Image.open(self.save_path)
os.remove(self.save_path)
# .convert('RGB')
return np.asarray(self.o_image)

def transform(self, width=768, height=1024):
    newsize = (width, height)
    self.t_height = height
    self.t_width = width
    pic = self.o_image
    img = pic.resize(newsize)
    self.t_image = img
    background = Image.new("RGBA", newsize, (255, 255, 255, 255))
    background.paste(img, mask=img.split()[3]) # 3 is the alpha channel
    self.save_path = self.save_path[:-3] + '.jpg'
    background.convert('RGB').save(self.save_path, 'JPEG')
    return np.asarray(background.convert('RGB'))

```

– Tạo mặt nạ phân đoạn hình ảnh

Tạo lớp chuẩn hóa, lớp này định nghĩa một biến đổi chuẩn hóa cho các hình ảnh đầu vào. Nó nhận giá trị trung bình và độ lệch chuẩn mong muốn và thực hiện chuẩn hóa dựa trên số lượng kênh trong tensor hình ảnh đầu vào. Hàm palette tạo một bảng màu cho việc hiển thị mặt nạ phân đoạn. Nó tạo một danh sách các giá trị RGB cho mỗi lớp. Bảng màu được tạo ra hiện đang được thiết lập cho 4 lớp.

Sau đó thực hiện gói hàm U2NET được tạo ra với số lượng kênh đầu vào và đầu ra cụ thể. Mô hình được tải với trọng số từ một file checkpoint đã được huấn luyện sử dụng hàm load_checkpoint_mgpu.

Tiếp theo, đoạn mã duyệt qua một danh sách các hình ảnh đầu vào trong một thư mục (image_dir). Mỗi hình ảnh được chạy qua, thay đổi kích thước thành (768, 768).

Hình ảnh đã biến đổi được chuyển qua mô hình U2NET, và tensor đầu ra được xử lý

sau để có được mặt nạ phân đoạn. Mặt nạ phân đoạn sau đó được lưu dưới dạng hình ảnh xám với kích thước gốc, sử dụng bảng màu được tạo trước đó. Các hình ảnh đã xử lý được lưu trong một thư mục kết quả thẻ (result_dir).

```
class Normalize_image(object):
    """Normalize given tensor into given mean and standard dev
```

Args:

mean (float): Desired mean to subtract from tensors

std (float): Desired std to divide from tensors

"""

```
def __init__(self, mean, std):
    assert isinstance(mean, (float))
    if isinstance(mean, float):
        self.mean = mean

    if isinstance(std, float):
        self.std = std

    self.normalize_1 = transforms.Normalize(self.mean, self.std)
    self.normalize_3 = transforms.Normalize([self.mean] * 3, [self.std] * 3)
    self.normalize_18 = transforms.Normalize([self.mean] * 18, [self.std] *
```

18)

```
def __call__(self, image_tensor):
    if image_tensor.shape[0] == 1:
        return self.normalize_1(image_tensor)

    elif image_tensor.shape[0] == 3:
        return self.normalize_3(image_tensor)

    elif image_tensor.shape[0] == 18:
        return self.normalize_18(image_tensor)

    else:
```

```
assert "Please set proper channels! Normalization implemented only for  
1, 3 and 18"
```

```
def get_palette(num_cls):
```

```
    """ Returns the color map for visualizing the segmentation mask.
```

```
Args:
```

```
    num_cls: Number of classes
```

```
Returns:
```

```
    The color map
```

```
"""
```

```
n = num_cls
```

```
palette = [0] * (n * 3)
```

```
for j in range(0, n):
```

```
    lab = j
```

```
    palette[j * 3 + 0] = 0
```

```
    palette[j * 3 + 1] = 0
```

```
    palette[j * 3 + 2] = 0
```

```
    i = 0
```

```
    while lab:
```

```
        palette[j * 3 + 0] = 255
```

```
        palette[j * 3 + 1] = 255
```

```
        palette[j * 3 + 2] = 255
```

```
        # palette[j * 3 + 0] |= (((lab >> 0) & 1) << (7 - i))
```

```
        # palette[j * 3 + 1] |= (((lab >> 1) & 1) << (7 - i))
```

```
        # palette[j * 3 + 2] |= (((lab >> 2) & 1) << (7 - i))
```

```
        i += 1
```

```
    lab >>= 3
```

```
return palette
```

```
transforms_list = []
```

```
transforms_list += [transforms.ToTensor()]
```

```
transforms_list += [Normalize_image(0.5, 0.5)]
```

```
transform_rgb = transforms.Compose(transforms_list)
```

```
net = U2NET(in_ch=3, out_ch=4)
```

```
net = load_checkpoint_mgpu(net, checkpoint_path)
```

```
net = net.to(device)
```

```

net = net.eval()
palette = get_palette(4)
images_list = sorted(os.listdir(image_dir))
for image_name in images_list:
    img = Image.open(os.path.join(image_dir, image_name)).convert('RGB')
    img_size = img.size
    img = img.resize((768, 768), Image.BICUBIC)
    image_tensor = transform_rgb(img)
    image_tensor = torch.unsqueeze(image_tensor, 0)

    output_tensor = net(image_tensor.to(device))
    output_tensor = F.log_softmax(output_tensor[0], dim=1)
    output_tensor = torch.max(output_tensor, dim=1, keepdim=True)[1]
    output_tensor = torch.squeeze(output_tensor, dim=0)
    output_tensor = torch.squeeze(output_tensor, dim=0)
    output_arr = output_tensor.cpu().numpy()

    output_img = Image.fromarray(output_arr.astype('uint8'), mode='L')
    output_img = output_img.resize(img_size, Image.BICUBIC)

    output_img.putpalette(palette)
    output_img = output_img.convert('L')
    output_img.save(os.path.join(result_dir, image_name[:-4] + '.jpg'))

```

Tích hợp Flask

Hiển thị một mẫu HTML có tên "index.html" khi người dùng truy cập địa chỉ URL gốc. Đây là trang chính của ứng dụng web. Trang web sẽ lắng nghe sự kiện mà người dùng yêu cầu POST với các tệp hình ảnh (giả sử là một hình ảnh của quần áo và một tệp mô hình). Sau đó sẽ gửi thông qua địa chỉ cụ thể qua đường dẫn tới máy chủ.

Website sẽ đợi một phản hồi từ máy chủ chứa một hình ảnh kết quả. Mở hình ảnh từ phản hồi sử dụng PIL (Image.open) và chuyển đổi nó thành một chuỗi mã hóa base64. Hiển thị mẫu HTML "index.html" với hình ảnh đã mã hóa base64 trên trang web.

```

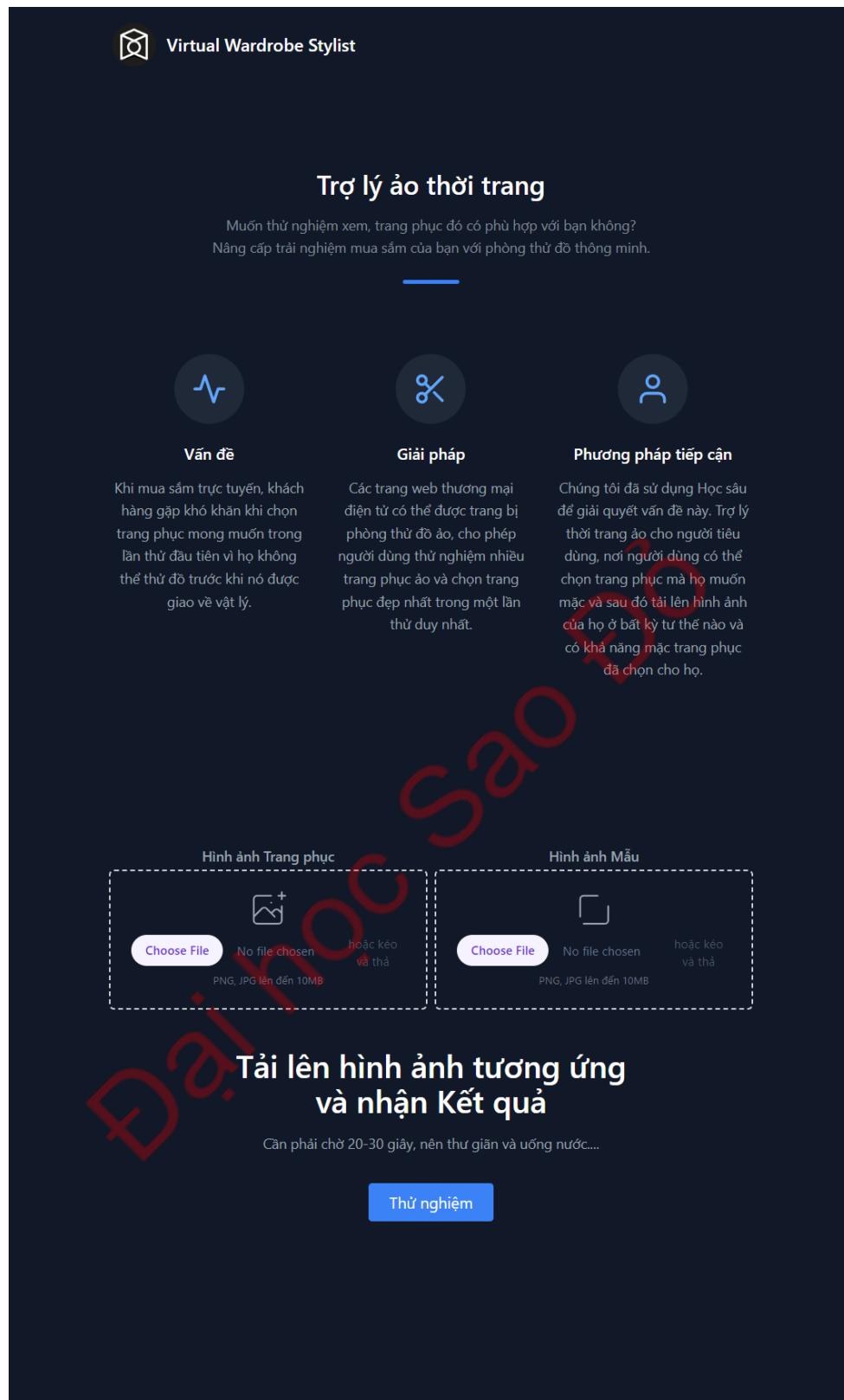
from flask import Flask, request, jsonify, render_template
from PIL import Image

```

```
import requests
from io import BytesIO
import base64
app = Flask(__name__)
@app.route('/')
def home():
    return render_template("index.html")
@app.route("/preds", methods=['POST'])
def submit():
    cloth = request.files['cloth']
    model = request.files['model']
    ## replace the url from the ngrok url provided on the notebook on server.
    url = "http://b89a-34-79-184-67.ngrok.io/api/transform"
    print("sending")
    response = requests.post(url=url, files={"cloth":cloth.stream,
"model":model.stream})
    op = Image.open(BytesIO(response.content))
    buffer = BytesIO()
    op.save(buffer, 'png')
    buffer.seek(0)
    data = buffer.read()
    data = base64.b64encode(data).decode()
    return render_template('index.html', op=data)
    # return render_template('index.html', test=True)
if __name__ == '__main__':
    app.run(debug=True)
```

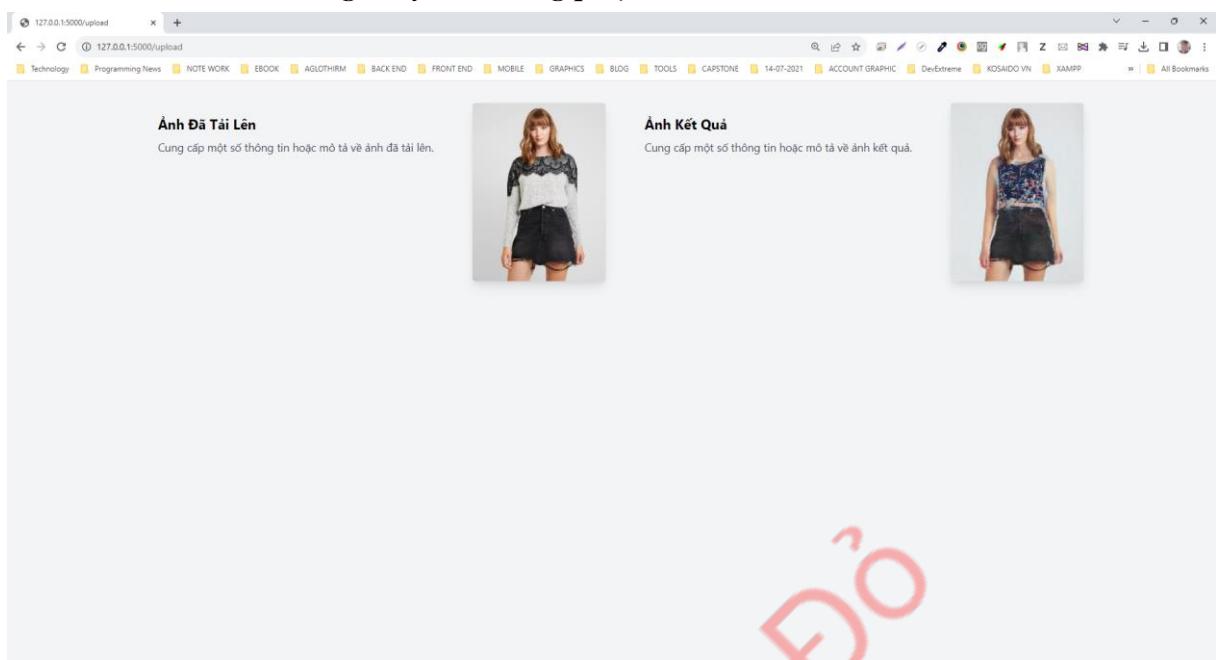
3.2. Giao diện và chức năng hoàn chỉnh

3.2.1. Giao diện Home



Hình 20: Giao diện Home

3.2.2. Chức năng thay đổi trang phục



Hình 21:Chức năng thay đổi trang phục

Kết quả và đánh giá

1. Kết quả đạt được

Kết quả đạt được từ quá trình "Chỉnh sửa Trang Phục" trong ứng dụng Chính sửa Trang Phục là sự thuận tiện và linh hoạt cho người dùng trong việc thực hiện các chỉnh sửa trên trang phục. Sự tích hợp hiệu quả giữa các thành phần như Flask, Quản lý Dữ liệu, và Xử lý Ảnh và Áo đồ giúp người dùng tương tác một cách dễ dàng.

Khi người dùng nhập mô tả trang phục, hệ thống sử dụng Flask để xử lý và thực hiện các phép biến đổi trên hình ảnh theo yêu cầu. Dữ liệu tương ứng được lưu trữ và quản lý trong thành phần Quản lý Dữ liệu. Trong trường hợp mô tả không rõ ràng, người dùng có thể nhập thêm thông tin để đảm bảo chính xác và đáp ứng đúng nhu cầu của họ.

Quá trình tiếp theo là áp dụng các biến đổi trang phục lên ảnh thông qua thành phần Xử lý Ảnh và Áo đồ. Sau đó, giao diện Người dùng hiển thị ảnh đã chỉnh sửa, cung cấp cho người dùng khả năng xem trước và kiểm tra kết quả cuối cùng của quá trình chỉnh sửa.

Kết quả là người dùng có khả năng sáng tạo linh hoạt hơn trên trang phục, tạo ra những hình ảnh độc đáo và cá nhân hóa. Điều này nâng cao trải nghiệm người dùng, giúp họ thực hiện ý tưởng thời trang của mình một cách dễ dàng và hiệu quả, không chỉ bằng cách tương tác mà còn thông qua việc thử nghiệm và kiểm tra kết quả trực tiếp trên ảnh.

2. Hướng phát triển đề tài

Định hướng phát triển cho ứng dụng Chính sửa Trang Phục được tập trung vào việc nâng cao trải nghiệm người dùng và mở rộng khả năng sáng tạo thông qua tích hợp các công nghệ mới. Một trong những chiến lược quan trọng là mở rộng khả năng của thành phần Xử lý NLP thông qua sự tích hợp sâu rộng của công nghệ AI và machine learning. Điều này giúp hệ thống hiểu rõ hơn về ý định và sở thích của người dùng, mang lại trải nghiệm cá nhân hóa và hiệu quả.

Hướng phát triển tiếp theo là tích hợp hệ thống đề xuất trang phục, giúp người dùng khám phá các lựa chọn mới dựa trên lịch sử sử dụng và ưa thích của họ. Điều này giúp tăng tính tương tác và sự linh hoạt trong việc tạo ra những trải nghiệm thời trang độc đáo.

Chúng tôi cũng đề xuất một bước tiến mới bằng cách tích hợp công nghệ thực tế ảo và thực tế ảo mở rộng, tạo ra trải nghiệm tương tác chân thực hơn khi người dùng có thể thử nghiệm trực tiếp trang phục trên ảnh của mình. Điều này không chỉ tạo ra sự thuận tiện mà còn tăng cường sự hứng thú của người dùng.

Để đáp ứng nhu cầu ngày càng đa dạng của người dùng, chúng tôi đề xuất mở rộng cơ sở dữ liệu với đa dạng hơn về trang phục và phụ kiện. Cập nhật thường xuyên từ xu hướng thời trang và sự đóng góp từ cộng đồng người dùng sẽ giúp duy trì tính mới mẻ và sáng tạo trong ứng dụng.

Ngoài ra, việc kết hợp tính năng chia sẻ trực tuyến và tương tác xã hội sẽ tạo ra một cộng đồng sáng tạo, cho phép người dùng chia sẻ và nhận phản hồi về ý tưởng thời trang của họ. Cuối cùng, việc đảm bảo ứng dụng có khả năng tương thích trên nhiều nền tảng và thiết bị sẽ tăng cường tiện ích và linh hoạt cho người dùng, từ điện thoại di động đến máy tính bảng và các thiết bị thông minh khác. Những hướng phát triển này sẽ không chỉ cung cấp cho người dùng trải nghiệm độc đáo mà còn giữ cho ứng dụng Cảnh sửa Trang Phục luôn đứng vững trước xu hướng thời trang và công nghệ mới.

Đại học Sao Đỏ

Tài liệu tham khảo

- [1] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, Fang WenCity University of Hong Kong Microsoft Research Asia Microsoft Cloud + AI University of Science and Technology of China “Bringing Old Photos Back to Life”
- [2] Xintao Wang Yu Li Honglun Zhang Ying Shan Applied Research Center (ARC), Tencent PCG “Towards Real-World Blind Face Restoration with Generative Facial Prior”
- [4] Wikipedia, Mean squared error
- [5] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, “Image Restoration Using Convolutional Auto-encoders with Symmetric Skip”, pp. 1-17.
- [5] Neptune.ai. (2023). Image Processing in Python: Algorithms, Tools, and Methods You Should Know. [Online]. Available: Neptune.ai Article.
- [6] Geekflare. (2023). 6 Python Image Processing Libraries for Efficient Visual. [Online]. Available: Geekflare Article.
- [7] Dean2020, "Deep learning là gì? Tổng quan kiến thức về Deep learning," 2020. [Online].