Tái tạo ảnh răng từ ảnh niềng răng có mắc cài sử dụng GAN

Vũ Tuấn Hải[1]

[1] Đại học Công nghệ Thông tin - ĐHQG TP. HCM, TP. Hồ Chí Minh, Việt Nam

braces → teeth

Hình 1: Dựa trên [14], với hai tập dataset về braces và teeth, model của tôi có thể chuyển đổi từ ảnh niềng răng có mắc cài sang ảnh không còn niềng răng.

Tổng quan

Xử lý ảnh đã được ứng dụng sâu rộng trong nhiều ngành nghề, đặc biệt trong y tế và chuyên ngành nha khoa. Trong lĩnh vực nha khoa, niềng răng từ lâu đã thành một dịch vụ phổ biến và được rất nhiều người sử dụng. Tuy nhiên, người niềng răng phải đeo mắc cài thường xuyên trong khoảng thời gian dài khiến quá trình giao tiếp gặp nhiều vấn đề. Do đó, trong bài báo này, tôi đề xuất một số phương pháp giải quyết bài toán braces2teeth, hay xóa mắc cài và khôi phục lại vùng đã bị xóa bằng ba generative model: CycleGAN, Pix2Pix và Inpainting, model đầu tiên là ánh xạ $F: Y \to Y$*, model thứ hai là ánh xạ* $G: X \to Y$ *với* X *là phân phối ảnh niềng răng chứa mắc cài và* Y *là phân phối ảnh răng không chứa mắc cài. Các model này có thể ứng dụng để xử lý video dưới dạng các bộ lọc trong các nền tảng nói chuyện, hội họp trực tuyến, giúp cho người đang niềng răng tự tin hơn.*

Từ khóa: braces2teeth, CycleGAN, Pix2Pix.

1. Giới thiệu

Trong những năm gần đây, đã có nhiều nghiên cứu liên quan đến tác vụ xóa vật thể không mong muốn ra khỏi ảnh và khôi phục lại vùng đã bị xóa như [1], [2] và [3]. Tuy nhiên, những phương pháp này còn đang gặp những hạn chế như phải trích xuất đặc trưng thủ công. Một số phương pháp sử dụng GAN như [4] và [14], … đã đạt được nhiều kết quả tốt nhưng chỉ được thử nghiệm trên các bộ dữ liệu phổ biến như phong cảnh, xe cộ và nhà cửa. Do đó, trong bài báo này, tôi vẫn sử dụng GAN nhưng đề xuất thêm một số phương pháp để giải quyết riêng về tác vụ loại bỏ niềng răng. Phương pháp của tôi có ưu điểm là có thể loại bỏ hoàn toàn mắc cài và tái tạo lại nơi vừa xóa với chất lượng tương đương những thợ chỉnh sửa ảnh chuyên nghiệp. Đặc biệt, đây là bài báo đầu tiên trình bày phương pháp giải quyết vấn đề braces2teeth. Bài báo có một số đóng góp và nghiên cứu về những nội dung sau: dataset về mắc cài và răng,

model CycleGAN và một số phương pháp tăng cường cải thiện chất lượng ảnh răng sinh ra, model Pix2Pix và phương pháp sinh dataset cho bài toán braces2teeth cuối cùng là model Inpainting.

Cấu trúc của bài báo được chia làm 6 phần: phần 1 là giới thiệu, phần 2 là nghiên cứu liên quan, phần 3 trình bày chi tiết về cách tiếp cận của tôi, phần 4 trình bày cách thực nghiệm của tôi, phần 5 là kết quả đánh giá so với các phương pháp trước đó và cuối cùng, phần 6 sẽ thảo luận về tính ứng dụng và hạn chế.

2. Nghiên cứu liên quan

GAN [7] là generative model có thể học ánh xạ từ vectơ nhiễu ngẫu nhiên z đến hình ảnh đầu ra y, $G: z \rightarrow y$. Ngược lại, các GAN có điều kiện học ánh xạ từ ảnh x và vectơ nhiễu ngẫu nhiên z, tới y, $G: \{x, z\} \rightarrow y$. Generator G được huấn luyện để tạo ra ảnh giả không thể phân biệt được với ảnh thật khi đưa vào discriminator D.

CoGAN [21] là model bao gồm cho hai generator, một cho domain X và một cho domain Y với trọng số ràng buộc trên vài lớp đầu tiên cho latent space. Việc chuyển đổi từ ảnh niềng răng sang ảnh răng có thể đạt được bằng cách tìm một latent representation tạo ra ảnh niềng răng và hiển thị latent representation này thành kiểu ảnh răng.

SimGAN [22], trong model này, Shrivastava và cộng sự sử dụng adversarial loss [7] để tạo ra một ánh xạ từ X sang Y với chuẩn $\|x - G(x)\|_1$ được sử dụng để phạt nếu như có thay đổi lớn ở cấp độ pixel.

BiGAN/ALI [23] Unconditional GANs là model có generator G là ánh xạ từ $Z \rightarrow Y$, với $z \in Z$ là vector noise ngẫu nhiên. BiGAN và ALI đề xuất việc tạo thêm một generator F, hay ánh xạ nghịch đảo $Y \rightarrow Z$. Ý tưởng này được phát triển trong CycleGAN bằng việc thay đổi Z bằng X.

3. Hướng tiếp cận

Phương hướng giải quyết bài toán của tôi là tìm một ánh xạ f từ ảnh răng có mắc cài (X) sang ảnh răng (Y) với dataset được cho trước bao gồm $x_{i=1}^{N}$ với $x \in X$ và $y_{j=1}^{m}$ với $y \in Y$. f có thể xóa mắc cài và khôi phục lại vùng bị xóa hoàn toàn tự động.

3.1. CycleGAN

CycleGAN là model kết hợp ý tưởng của SimGAN [22] và BiGAN/ALI [23] và được thiết kế và tối ưu về mặt kiến trúc để tạo ra ảnh chỉ có thay đổi nhỏ so với ảnh gốc và thực hiện thay đổi về diện mạo (răng có mắc cài → răng không có mắc cài) mà không làm thay đổi về kết cấu (răng đẹp → răng xấu). Vì cùng mục tiêu, tôi sử dụng CycleGAN đầu tiên để tiến hành kiểm tra tính khả thi của bài toán.

Như đã giới thiệu trong phần nghiên cứu liên quan, CycleGAN có thể học được hai ánh xạ $G: X \rightarrow Y$ và $F: Y \rightarrow X$. Nó cũng bao gồm hai discriminator D_X và D_Y, với D_X có thể phân biệt được ảnh thật x và ảnh giả $F(y)$, tương tự D_Y có thể phân biệt giữa y và $G(x)$. Hàm loss sẽ bao gồm hai phần: adversarial loss [7] để xấp xỉ phân phối giữa ảnh giả và ảnh thật và cycle consistency loss để ngăn hai ánh xạ G và F không mâu thuẫn với nhau.

Với ánh xạ $G: X \rightarrow Y$ và discriminator D_Y, chúng ta có adversarial loss:

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[log D_Y(y)] + E_{x \sim p_{data}(x)}[log(1 - D_y(G(x))] \quad (1)$$

và tương tự đối với ánh xạ $F: Y \rightarrow X$ và discriminator D_X. Ngoài ra, cycle consistency loss có công thức như sau:

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1] \quad (2)$$

Hình 2: Các cặp ảnh (trái: ảnh thật, phải: ảnh giả) là kết quả từ model CycleGAN. Một số trường hợp chất lượng ảnh sinh ra kém và nguyên nhân. Hàng đầu tiên là niềng răng có cùng màu với môi và nướu (trái) hoặc niềng răng nhiều màu (phải). Hàng thứ hai là niềng răng có dây cung mảnh. Hàng cuối cùng là niềng răng có độ phân giải thấp.

Hàm loss tổng quát:

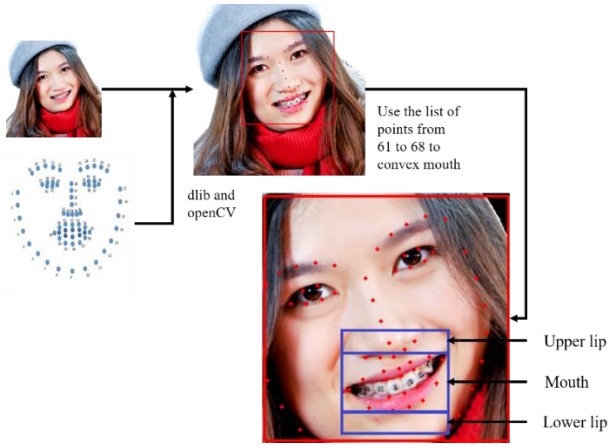
$$L(G, F, D_X, D_Y, X, Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, X, Y) + \lambda L_{cyc}(G, F) \quad (3)$$

trong đó λ là trọng số cho L_{cyc} và $\lambda = 10$ là tối ưu đã được thực nghiệm trong [14].

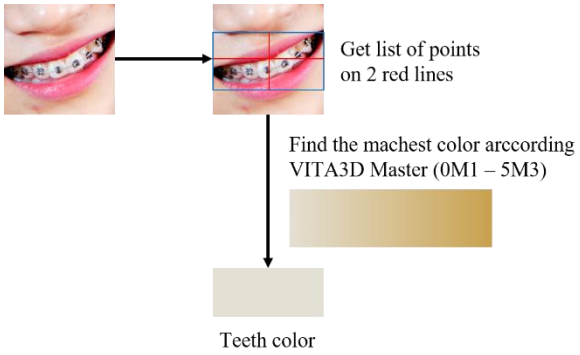
CycleGAN có thể đạt được kết quả khả quan trong nhiều trường hợp thông thường với mắc cài kim loại với đầy đủ dây cung, khung và dây buộc đàn hồi. Tuy nhiên, vẫn có một số điều kiện đặc biệt khiến ảnh sinh ra không đạt được chất lượng như mong đợi. Để giải quyết một số bất lợi được mô tả tại hình 2, đối với trường hợp mắc cài đa sắc, tôi áp dụng phương pháp tăng cường dữ liệu cho dataset braces2teeeth, tôi thay đổi ngẫu nhiên kênh màu S và V trong không gian HSV để màu răng và màu mắc cài được đa dạng hơn. Tuy số lượng mắc cài có màu sặc sỡ có thể xóa đã tăng nhưng model đã thay đổi sắc thái của ảnh khá nhiều.

Hình 3: Kết quả sau khi áp dụng phương pháp tăng cường ảnh

Đối với trường hợp dây cung mảnh và độ phân giải thấp, tôi trích xuất vùng miệng trước khi đưa vào model. Bằng dlib và OpenCV để tạo facial landmark detection. Phần mouth được đánh dấu bởi các point trên landmark có index từ 61 đến 68. Để giữ tỷ lệ chiều rộng và chiều dài của vùng miệng được trích xuất, tôi xác định lại vùng cắt bằng cách mở rộng về phía môi trên và môi dưới trước khi đưa vào mô hình. Phương pháp này sẽ được áp dụng chủ yếu trong model Pix2Pix.

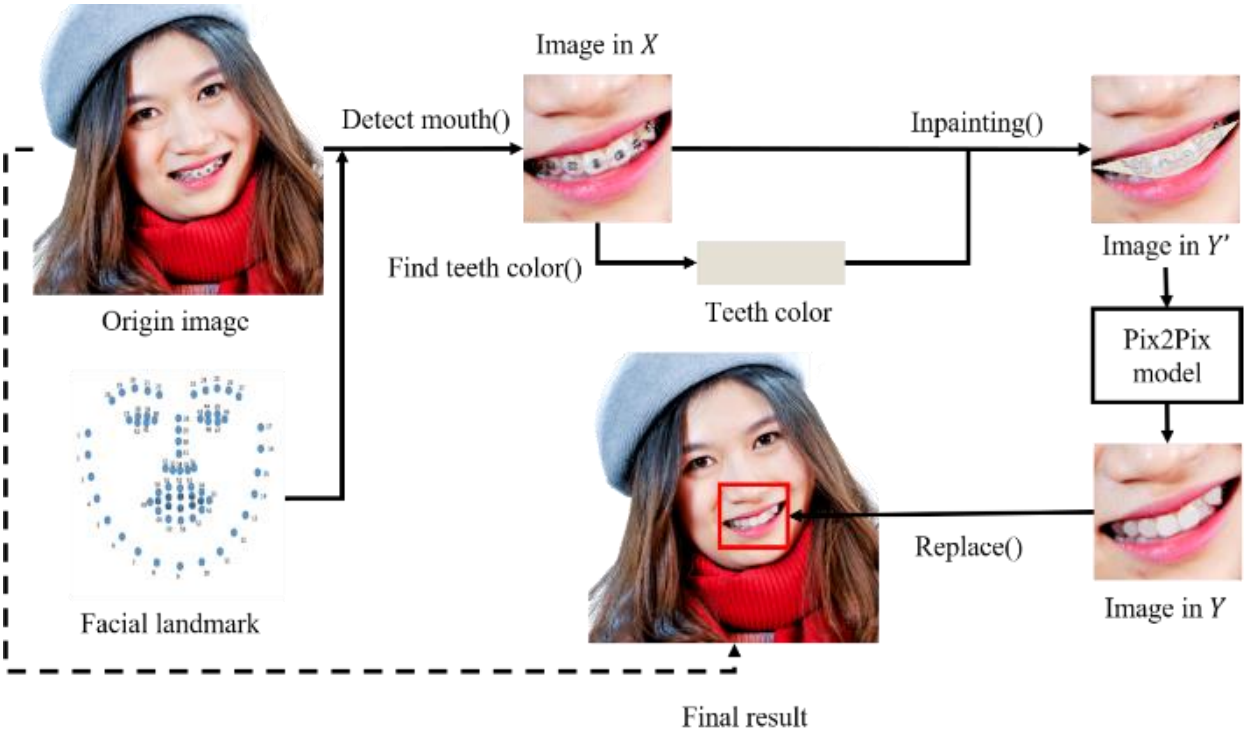


Hình 4: Phương pháp cắt ra vùng miệng.



Hình 5: Phương pháp tìm màu răng. Với 01M – 5M3 là thang đo màu răng từ trắng sáng đến ố vàng.

3.2. Pix2Pix



Hình 6: Tổng quan về quá trình preprocess

Pix2Pix [15] là model sử dụng conditional GAN để học ánh xạ từ X theo Y với từng điểm dữ liệu trong dataset là các cặp $\{x_i, y_i\}$. Pix2Pix đã được thực nghiệm và chứng minh là tốt hơn CycleGAN [14] về chất lượng ảnh sinh ra. Tuy nhiên, việc chuẩn bị tập dữ liệu bao gồm các cặp như trong hình 1 thực sự rất khó và tốn kém. Ảnh chụp trong quá trình niềng răng và sau khi niềng răng thường không liên quan đến nhau do các góc chụp khác nhau hoặc do tình trạng bệnh nhân thay đổi từ lúc niềng răng đến lúc tháo niềng. Để khắc phục khó khăn này, tôi sẽ giới thiệu một phương pháp (có tên preprocess) để tạo ra các cặp dữ liệu giả để có thể train model Pix2Pix.

Đầu tiên, tôi chỉ sử dụng ảnh răng $y_{j=1}^{m} \in Y$, với mỗi ảnh y_i tôi trích xuất phần miệng như đã mô tả ở hình 8, sau đó tìm màu răng (được mô tả chỉ tiết trong phần 3.6.1) và lấp đầy toàn bộ khu vực khoang miệng (răng và lợi) bằng màu răng (được mô tả chi tiết trong phần 3.6.2). Tôi lựa chọn phương pháp lấp đầy để giải quyết triệt để tình trạng mắc cài đa sắc trước đây và xa hơn nữa là mọi loại mắc cài. Sau khi lấp đầy, tôi có được ảnh mới y_j', là ảnh trung gian để chuyển đổi giữa x_i và y_j. y_j' vốn có nguồn gốc từ y_i, nhưng chúng ta sẽ tạm xem nó như là ảnh đã được tiền xử lý từ x_i. Nếu tồn tại x_j là phần tử trong cặp $\{x_j, y_j\}$ thì $preprocess(x_j) = y_j'$ vì phần thông tin về mắc cài đã bị xóa. Khi có các cặp $\{y'_i, y_i\}_{i=1}^{N}$, tôi có thể đưa vào model Pix2Pix để thực hiện quá trình train.

Tóm tắt phương pháp trên:

$$\{y_j\}_{j=1}^{M} \xrightarrow{preprocess} \{preprocess(y_j), y_j\}_{j=1}^{M} \rightarrow \{y'_j, y_j\}_{j=1}^{M} \quad (4)$$

3.2.1. Tìm màu răng

Giai đoạn thứ hai trong quá trình trên là xác định màu răng để răng sinh ra có màu tương đồng với màu răng gốc. Generator có khả năng cao sẽ tạo ra vùng răng giống nhau trong mọi trường hợp nếu chúng ta lấp đầy khoang miệng bằng cùng màu trong quá trình chuẩn bị dataset. Tôi sử dụng bảng màu răng dựa trên VITA3D Master để tham chiếu trực tiếp đến những điểm ảnh lấy được có khả năng là răng. Sau đó quá trình tìm màu răng bằng lấy tập pixel thông qua 2 đường trung trực của ảnh và so sánh pixel có màu gần nhất với màu trên bảng mẫu như mô tả trong hình 5.

3.2.2. Lấp đầy

Tôi sử dụng hàm convexPolygon OpenCV các point có index từ 61 – 68 trên facial landmark để xác định một đa giác lồi gọi là selection zone. Tôi thử nghiệm 3 phương pháp khác nhau để so sánh chất lượng lấp đầy:

Phương pháp A: những pixel nằm trong selection zone được inpainting lại bằng màu răng.

Phương pháp B: những pixel nằm trong selection zone và có khoảng cách với màu răng lớn hơn ngưỡng cho trước sẽ được inpainting lại bằng màu răng. Tại đây tôi chuyển ảnh sang không gian màu CIELab và sử dụng khoảng cách CIE2000 [20].

Phương pháp C: bao gồm 3 công đoạn:

a. Chuyển ảnh sang ma trận superpixel

b. Đánh nhãn những superpixel có density bé hơn threshold là "teeth", giữ nguyên, ngược lại thì đánh dấu là "not teeth".

c. Đối với những superpixel có label "not teeth", tôi tìm những pixel xung quanh thuộc superpixel được gán nhãn "teeth" và lấy trung bình màu của chúng, nếu không có neighbour teeth nào thì superpixel "not teeth" được inpaint bằng màu răng

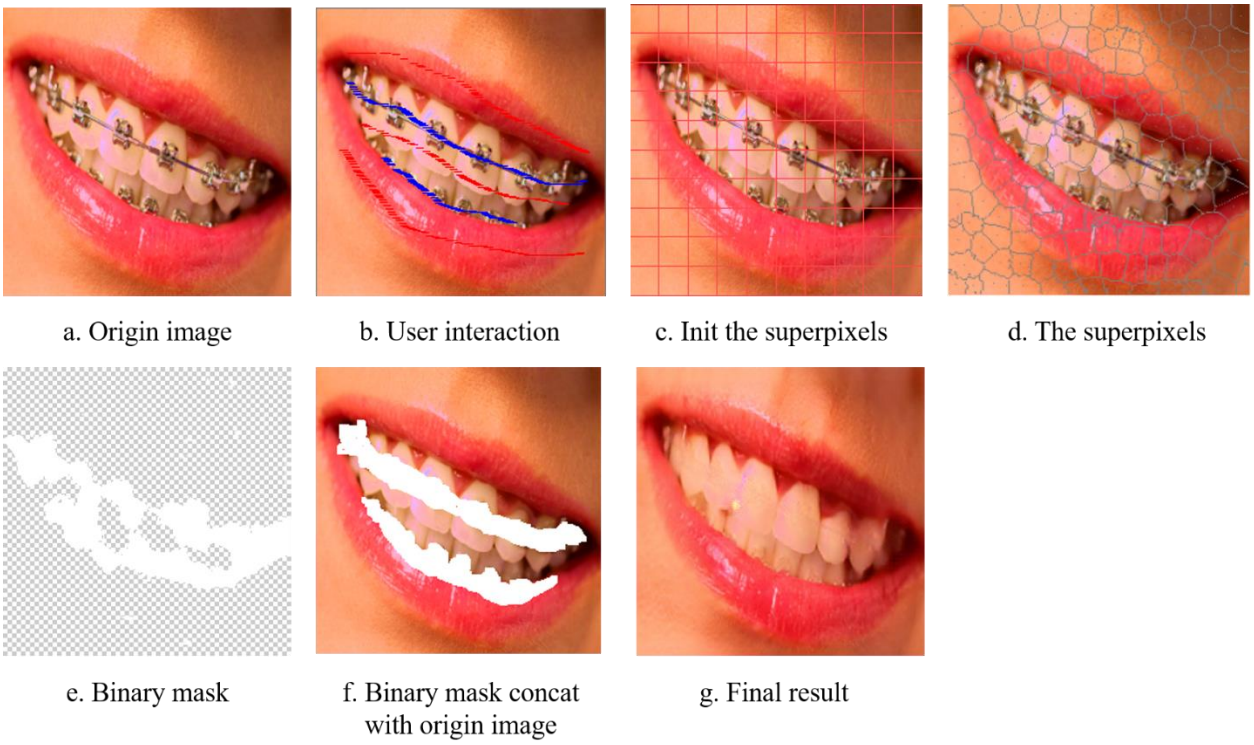
Cả ba phương pháp trên đều có thể lấp đầy khoang miệng triệt để tuy nhiên mỗi phương pháp đều có nhược điểm riêng: phương pháp A làm mất đi thông tin về kết cấu hàm răng, phương pháp B xóa mắc cài không triệt để và phương pháp C đạt kết quả tốt nhất tuy nhiên lại làm chậm quá trình xử lý khá nhiều.

3.3. Inpainting

Cuối cùng, tôi giới thiệu một model kết hợp hai model GraphCut [20] và Generative Image Inpainting with

Contextual Attention [4] có khả năng mang lại kết quả tốt hơn, đồng thời còn cho phép người dùng can thiệp để đầu ra đạt được kết quả như ý muốn. (tôi sẽ gọi model này là **Inpainting** để ngắn gọn)

Inpainting sẽ được chia làm hai giai đoạn xử lý, đầu tiên sẽ là GraphCut (xóa mắc cài) và sau đó là Generative Image Inpainting with Contextual Attention (khôi phục). GraphCut sẽ nhận vào ảnh gốc (hình 7. a) và user interaction (hình 7. b) đánh dấu vị trí mắc cài, sau đó ánh xạ ảnh sang tập superpixel (hình 7. c và 7. d) và sử dụng thêm thông tin từ user interaction (những đường màu xanh đánh dấu vùng cần loại bỏ và những đường màu đỏ để đánh dấu những vùng cần giữ lại) để ánh xạ sang đồ thị ba chiều. Cuối cùng đưa đồ thị ba chiều vào thuật toán Boykov-Kolmogorov min-cut/max-follow để thu được 2 đồ thị con *S* là tập các superpixel được đánh dấu là braces và *T* là tập các superpixel được đánh dấu là teeth. Tôi sử dụng thông tin của S và T lên ảnh gốc và có được binary mask với giá trị 1 đánh dấu mắc cài và giá trị 0 đánh dấu không phải mắc cài (hình 7. e). Generative Image Inpainting with Contextual Attention nhận ảnh gốc và binary mask, tiến hành khôi phục tại những vị trí đã được đánh dấu trên binary mask theo thứ tự từ ngoài vào trong để đảm bảo tính nhất quán và cuối cùng trả về ảnh răng không chứa mắc cài (hình 7. f).



a. Origin image | b. User interaction | c. Init the superpixels | d. The superpixels

e. Binary mask | f. Binary mask concat with origin image | g. Final result

Hình 7: Các giai đoạn xử lý trong model Inpainting.

4. Thử nghiệm

4.1. Dataset

Để đánh giá các phương pháp được đề xuất, tôi thử nghiệm trên nhiều dataset, bao gồm hai lớp là ảnh niềng răng có mắc cài và ảnh răng. Vì là lĩnh vực đặc thù, dataset về ảnh niềng răng có mắc cài và ảnh răng cho đến nay đều không có sẵn hoặc không công khai, vì vậy tôi đã tự thu thập từ google image và mô hình StyleGAN2. Ngoài ra còn có một số lượng nhỏ được cung cấp từ các phòng khám nha khoa.

Tôi lấy hình ảnh từ google với công cụ Downloader do Yabin Zheng cung cấp bằng cách sử dụng nhiều từ khóa

trong nhiều ngôn ngữ khác nhau liên quan đến mắc cài và răng, kết quả chỉ tạm chấp nhận với 0 - 100 hình ảnh cho mỗi từ khóa. StyleGAN2 là model khổng lồ đến từ các nhà nghiên cứu của NVIDIA, nó tạo ra khuôn mặt người với chất lượng cao và tôi lấy tài nguyên này bằng cách sử dụng công nghệ webdriver Selenium, sử dụng một phần của preprocessing module để có được phần miệng với kích thước 256 x 256 x 3. Hình ảnh từ StyleGAN2 không bị giới hạn nhưng phần răng của các khuôn mặt khá giống nhau (tất cả đều trắng, sáng và đẹp) vì vậy tôi chỉ sử dụng nó với tỷ lệ vừa phải trong dataset. Cuối cùng, nguồn ảnh từ các phòng khám nha khoa rất có giá trị, tuy nhiên hầu hết chúng được giữ kín và tôi chỉ được cung cấp một ít trong số đó.

Tất cả các hình ảnh trong hai bộ dữ liệu được chuẩn hóa trước khi sử dụng để train.Tác vụ chuẩn hóa bao gồm thay đổi kích thước thành 256 x 256 x 3 bằng cách sử dụng phép nội suy lưỡng tính, loại bỏ các hình ảnh nhiễu như ảnh quảng cáo, tranh vẽ,… và chỉnh sửa ảnh, các mắc cài không nằm đúng vị trí cũng được điều chỉnh gần phần trung tâm bằng tay. Tôi có hai bộ dataset để kiểm tra trong đó model CycleGAN sử dụng cả hai lớp và model Pix2Pix chỉ sử dụng lớp ảnh răng.

Tên	**Số lượng ảnh**	**Phân lớp**
braces2teeth	2196	testX: 201 trainX: 805 testY: 240 trainY: 950
teeth2	7000	train: 5000 val: 1000 test: 1000

Bảng 1: Hai bộ dataset khác nhau về braces và teeth.

4.2. Training

Kế thừa kết quả đến từ [14], tôi không thay đổi nhiều về các hyperparameter và sử dụng lại mô hình ổn định đã ổn định. Tôi cũng sử dụng optimizer Adam với batch size 1. Tất cả các model đều được train từ đầu với learning rate là 0,0002, tôi giữ learning rate trong 100 epoch và giảm dần tuyến tính xuống 0 trong 100 epoch tiếp theo. Chi tiết về quá trình training có thể được tìm thấy tại phần phụ lục.

5. Kết quả

Tôi sẽ so sánh chất lượng ảnh sinh ra với các phương pháp trước đó lẫn phương pháp gốc (CycleGAN, Pix2Pix). Toàn bộ mã nguồn bằng Python và minh chứng cho đánh giá của tôi có thể được tìm thấy tại đây. Tôi sử dụng phương pháp Amazon Mechanical Turk (AMT) để xác định chất lượng ảnh sinh ra braces → teeth theo như quy trình nghiên cứu từ Isola và cộng sự [16] nhưng với quy mô nhỏ hơn. Turker (người đánh giá) được làm bài kiểm tra với đề bài là danh sách các cặp bao gồm một ảnh răng thật và một ảnh răng giả do model của tôi tạo ra, ảnh giả nằm ngẫu nhiên ở bên phải hoặc bên trái. Trong mỗi lần thử, từng cặp xuất hiện trong 1 - 2 giây, sau đó các Turker trả lời bên nào là giả trong 5 - 10 giây. Không có phản hồi về tính đúng sai của câu trả lời. Mỗi phiên chỉ thử nghiệm một model và người tham gia chỉ được phép hoàn thành một phiên duy nhất.

Model	**Training dataset**	**% Turker labeled real**
CoGAN	braces2teeth	0.7% ± 0.5%
BiGAN/ALI	braces2teeth	2.1% ± 0.5%
SimGAN	braces2teeth	1.1% ± 1%
Inpainting	braces2teeth/teeth	1.7% ± 0.3%

CycleGAN	braces2teeth	**28.3% ± 4.3%**
CycleGAN kết hợp phương pháp của tôi	braces2teeth	**32.8% ± 4.7%**
Pix2Pix + preprocessing	teeth2	**38% ± 5.2%**

Bảng 2: AMT score thử nghiệm trên cùng một tập dữ liệu với độ phân giải 256 x 256 x 3.

6. Thảo luận và hạn chế

Mặc dù CycleGAN và Pix2Pix đã xử lý tốt đa số những vẫn những trường hợp liên quan đến kết cấu của răng chưa thể xử lý hoàn hảo do góc độ ảnh hoặc do vật cản. Nguyên nhân cơ bản vẫn là do dataset chưa đủ để thể hiện tất cả các trường hợp niềng răng, ví dụ model không thể xử lý niềng răng vô hình vì trong dataset rất hiếm những trường hợp đó. Lý do thứ hai là khi lấp đầy khoang miệng sẽ dẫn đến việc mất đi một phần kết cấu răng, dẫn đến một số điểm bất hợp lý trong ảnh. Tuy nhiên, nhờ vào thời gian xử lý thấp, CycleGAN và Pix2Pix hoàn toàn có thể áp dụng để xử lý video dưới dạng các bộ lọc trong các nền tảng nói chuyện, hội họp trực tuyến.

Cuối cùng, tôi nghĩ rằng model Inpainting sẽ đạt kết quả tốt hơn và là hướng đi tương lai cho bài toán braces2teeth, tuy nhiên Generative Image Inpainting with Contextual Attention và các model chuyên dùng để tái tạo ảnh tương tự vẫn quá cồng kềnh, chúng yêu cầu số kích thước dataset lẫn khối lượng tính toán lớn để cho ra chất lượng ảnh tốt nhất.

Hình 8: Một số kết quả thử nghiệm. Bên trái mỗi cặp (ảnh thật), bên phải mỗi cặp (ảnh giả).

Model	Dataset	Kiến trúc	Hyperparameter	Môi trường
CycleGAN	braces2teeth Augmentation	Resnet for generator. PatchGANs for discriminator.	– Total weight: 28.286 M (G_X: 11.378 M, G_Y: 11.378 M, D_X: 2.765 M, D_Y: 2.765 M). – lr: 0.0002 – batch size: 1 – optimizer: Adam – epoch: 200	Google Colab with GPU Pytorch
Pix2Pix	braces2teeth Augmentation/teeth	Unet for generator. PatchGANs for discriminator	– Total weight: 57.183 M ($G_{Y'Y}$: 54.414 M, D_Y: 2.769 M). – lr: 0.0002 – batch size: 1 – optimizer: Adam – epoch: 200	

Bảng 3: Thông số về quá trình training của hai model CycleGAN và Pix2Pix.

Tham khảo

1. C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2009), 2009.
2. Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 341–346. ACM, 2001.
3. Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1033–1038. IEEE, 1999.
4. Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5505– 5514, 2018.
5. Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. ACM Transactions on Graphics (TOG), 36(4):107, 2017.
6. Chao Yang, Yuhang Song, Xiaofeng Liu, Qingming Tang, and C-C Jay Kuo. Image inpainting using block-wise procedural training with annealed adversarial counterpart. arXiv preprint arXiv:1803.08943, 2018.
7. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014.
8. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In ICLR, 2017.
9. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In ICML, 2016.
10. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. CVPR, 2016
11. S. Vicente, V. Kolmogorov and C/ Rother.. Graph cut based image segmentation with connectivity priors. In IEEE conference on computer vision and pattern recognition (pp. 1-8). IEEE, 2008.
12. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028, 2017.

13. M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.
14. Zhu, Jun-Yan, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision. 2017.
15. ISOLA, Phillip, et al. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1125-1134.
16. Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
17. Li, Zhengqin, and Jiansheng Chen. "Superpixel segmentation using linear spectral clustering." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.
18. Van den Bergh, Michael, et al. "Seeds: Superpixels extracted via energy-driven sampling." European conference on computer vision. Springer, Berlin, Heidelberg, 2012.
19. Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." IEEE transactions on pattern analysis and machine intelligence 34.11 (2012): 2274-2282.
20. NAIK, D., et al. Fast interactive superpixel based image region generation. 2019.
21. Liu, Ming-Yu, and Oncel Tuzel. "Coupled generative adversarial networks." Advances in neural information processing systems. 2016.
22. Shrivastava, Ashish, et al. "Learning from simulated and unsupervised images through adversarial training." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
23. Dumoulin, Vincent, et al. "Adversarially learned inference." arXiv preprint arXiv:1606.00704 (2016).