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

Vũ Tuấn Hải[1], Nguyen Thanh Vu[2], Huỳnh Hồ Thị Mộng Trinh[1], Phạm Thế Bảo[2]

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

[2]Đại học Sài Gòn, TP. Hồ Chí Minh, Việt Nam

braces → teeth

Hình 1: Dựa trên [14], với hai tập dataset về braces và teeth, model của chúng tôi có thể chuyển đổi từ ảnh niềng răng có mắc cài sang ảnh không còn niềng răng (mắc cài đã được xóa và thay thế bằng răng). Kết quả của model tốt hơn so với [14] vì sử dụng thêm những phương pháp chỉ áp dụng với loại dữ liệu đặc trưng được mô tả ở hình 1: (trái mỗi cặp) điểm dữ liệu lớp braces và (phải mỗi cặp) điểm dữ liệu lớp teeth.

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, bệnh nhân niềng răng thường có một số vấn đề phát sinh do quá trình điều trị lâu dài, như việc bệnh nhân hay tự ti khi trò chuyện, chụp ảnh vì mắc cài khiến nụ cười không được đẹp. Do đó, trong bài báo này, chúng tôi đề xuất một số phương pháp giải quyết bài toán xóa mắc cài một cách triệt để bằng việc nghiên cứu và so sánh hai generative model: Pix2Pix và CycleGAN, model đầu tiên là ánh xạ $F: Y \rightarrow Y$, model thứ hai là ánh xạ $G: X \rightarrow Y$ với X là phân phối ảnh niềng răng chứa mắc cài và Y là phân phối ảnh răng không chứa mắc cài.

Từ khóa: braces2teeth, CycleGAN, Pix2Pix.

1. Giới thiệu

Img2Img (chuyển đổi ảnh sang ảnh) là một bài toán rộng trong xử lý ảnh, model Img2Img có thể chuyển đổi ảnh x sang ảnh y trong ngữ cảnh nào đó cho trước. Một số ứng dụng nổi bật như chuyển từ ảnh xám sang ảnh màu, ghép ảnh hoặc tăng độ phân giải, … Trong nội dung bài báo này, bài toán cụ thể là chuyển đổi ảnh niềng răng có mắc cài sang ảnh răng, hay nói cách khác là xóa mắc cài ra khỏi ảnh.

Từ những năm 2000 đã tồn tại hai hướng giải quyết bài toán kể trên bao gồm: PatchMatch [1] sử dụng cho ảnh có đặc trưng đơn giản hoặc độ phân giải thấp và generative model. Một số phương pháp tiếp cận trước đây [2, 3] có thể hồi phục các kết cấu tĩnh nhưng không ổn định với những vật thể phức tạp như khuôn mặt và

đồ vật. Cách tiếp cận thứ hai [4, 5, 6] có thể khai thác những đặc trưng từ dataset cho trước và xử lý những vấn đề cụ thể liên quan đến dataset. Do đó, chúng tôi sẽ sử dụng cách tiếp cận thứ hai kết hợp với nhiều phương pháp tiền xử lý khác. Chúng tôi đã thử nghiệm theo hai hướng là supervised và unsupervised.

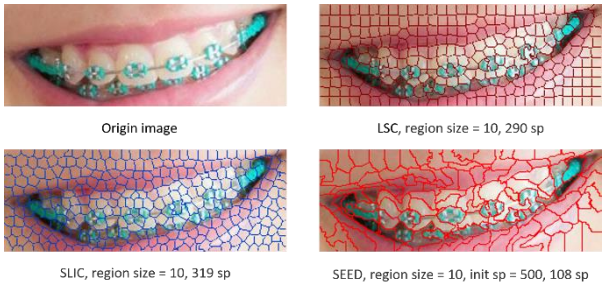
Theo hướng supervised, chúng tôi vẫn train model Pix2Pix với dataset là tập các cặp $\{x_i, y_i\}_{i=1}^{N}$. Tuy nhiên, việc chuẩn bị dataset bao gồm những cặp như hình 1 trong thực tế rất khó khăn và tốn nhiều chi phí. Ảnh chụp trong quá trình niềng răng và sau khi niềng răng thường không liên quan đến nhau do khác góc độ chụp hoặc thể trạng bệnh nhân đã thay đổi từ thời điểm niềng răng đến thời điểm tháo niềng. Do đó, mỗi điểm dữ liệu x_i sẽ tương ứng điểm dữ liệu y_i ($x_i = preProcessing(y_i)$), cả x_i lẫn y_i đều thuộc domain Y, để dễ phân biệt chúng tôi sẽ gọi x_i là y'_i. Về kiến trúc chúng tôi tham khảo phần lớn trong [4]. Cuối cùng, chúng tôi thử nghiệm thêm một model trong đó cho phép sự can thiệp từ người dùng nhằm điều chỉnh kết quả theo ý thích của họ.

. Theo hướng unsupervised, chúng tôi sẽ thử nghiệm với model CycleGAN vì lượng dữ liệu và tốc độ training không nhất thiết phải được cung cấp quá lớn.

Vấn đề khác biệt trong bài toán so với những bài toán Img2Img khác là mắc cài có hình dáng, màu sắc rất đa dạng và thường xuyên thể hiện không đầy đủ. Do đó chúng tôi tiền xử lý dữ liệu bằng cách xóa và tách mắc cài trước, sau đó mới đưa ảnh vào model. Chúng ta sẽ thảo luận thêm về vấn đề này trong phần 3.

2. Nghiên cứu liên quan

Superpixel generation Là phương pháp ánh xạ từ ảnh có kích thước bất kì sang ma trận những pixel lớn (pixel chứa những pixel gần giống nhau). Hiện tại có 3 phương pháp phổ biến: SEEDS [18], SLIC [19] và LCS [17]. Trong đó SEEDS yêu cầu số superpixel khởi tạo, SLIC và LCS yêu cầu region size. Vì tham số số superpixel khởi tạo không cần thay đổi khi kích thước ảnh thay đổi nên chúng tôi sẽ ưu tiên sử dụng SEED. Kết quả của ba phương pháp trên ảnh niềng răng có mắc cài là tương đối giống nhau.

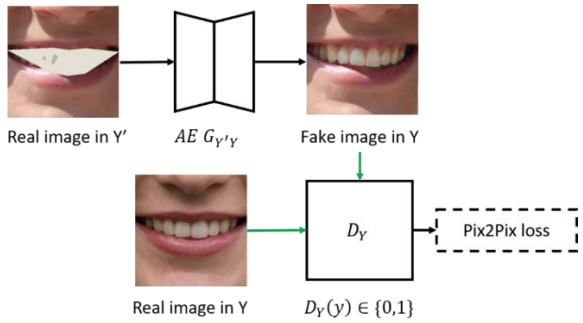


Hình 2: Superpixel generation bằng những phương pháp khác nhau, LSC và SLIC yêu cầu tính toán lại region size để đạt kết quả tốt nhất. Trong khi đối với SEED, kích thước mắc cài sẽ thay đổi theo kích thước ảnh nên không cần thay đổi tham số là số lượng superpixel khởi tạo.

GraphCut [20] Là phương pháp phân đoạn ảnh áp dụng lý thuyết đồ thị. Cho đồ thị $G(E,V)$, cut là đường cắt chia G thành 2 đồ thị con với min-cut là đường cut với tổng số lượng edge cắt qua là nhỏ nhất và ngược lại với max-cut. Trong flow network, cho $G(E,V)$ hữu hạn và mỗi edge (u,v) có trọng số c là giá trị thực không âm. Giả sử có hai node, sink node và source node đã được xác định. $s-t$ cut là đường cut chia G thành hai tập S và T sao cho $\forall s \in S, s$ là node chứa pixel là foreground và $\forall t \in T, t$ là node chứa pixel là background. $s-t$ cut là được gọi là max-flow khi tổng trọng số của các edge đường cut đi qua là cực đại và ngược lại với min-flow. Sau khi có đồ thị $G(E,V)$, [20] đã áp dụng thuật toán Boykov-Kolmogorov: $G \rightarrow [S,T]$ để thu được đồ thị S và T. Với S là tập các superpixel được đánh dấu là braces và T là tập các superpixel được đánh dấu là teeth.

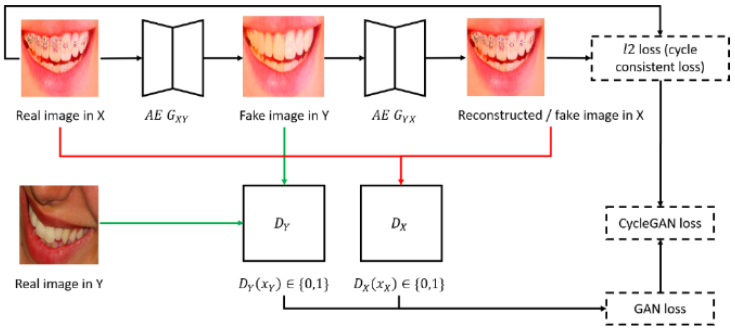
Generative Adversarial Networks (GAN) [7, 8] đã đạt được nhiều kết quả ấn tượng trong ứng dụng sinh ảnh và sửa ảnh. Những nghiên cứu gần đây về Conditional GAN đã cho ra nhiều ứng dụng như text2image [9], image inpainting [10], … Chìa khóa khiến GAN thành

công là ý tưởng hàm adversarial loss giúp generator, theo lý thuyết sẽ cho ra kết quả khiến chuyên gia cũng không thể phân biệt thật giả.



Hình 3: Mô hình tổng quát của Pix2Pix, bao gồm một Generator $G_{Y'Y}$ và một Discriminator D_Y, $G_{Y'Y}$ là ánh xạ $Y' \rightarrow Y$ trong khi đó D_Y là mạng classification giữa Y và Y_{fake}.

Pix2Pix [15] được mô tả trong hình 3 là biến thể của GAN nên cũng có Generator để sinh ảnh fake và Discriminator là mạng classification. Tuy nhiên khác với GAN bình thường khi input cho Generator là noise thì trong Pix2Pix generator đã được cung cấp sẵn từ bên ngoài.



Hình 4: Mô hình tổng quát của CycleGAN, bao gồm hai Generator G_{XY}, G_{YX} và hai Discriminator D_X, D_Y. Vì số lượng tham số gấp đôi nên model học được cả hai ảnh xạ $X \rightarrow Y$ lẫn $Y \rightarrow X$.

CycleGAN [14] được mô tả trong hình 4 cũng là một biến thể của GAN [7, 8] với hai hàm loss được cộng thêm đại lượng cycle consistency loss $CCL = F(G(X)) - X$ với ánh xạ $G: X \rightarrow Y$ và ánh xạ ngược $G^{-1} = F = Y \rightarrow X$.

3. Hướng tiếp cận

3.1. CycleGAN

Ground truth x thuộc domain X qua AE (auto-encoder) network đầu tiên có được y_{fake} trong domain Y, y' tiếp tục đi qua AE network thứ hai (đối xứng với AE network đầu tiên) để phục hồi x (x'). $l2$ loss (x, x') chính là CCL. Model có D_X phân biệt x và x', D_Y phân biệt y và y'. Chúng tôi hy vọng model có thể học được ánh xạ $G: X \rightarrow Y$. Chúng tôi thử nghiệm với tập test 200 ảnh niềng răng có mắc cài. Mô hình CycleGAN có thể giải quyết các trường hợp phổ thông nhưng không thể giải quyết được các trường hợp đặc biệt được trình bày trong phần 3.2.

3.2. Failure cases

Hình 5: Từ trái sang phải, ảnh gốc và ảnh do CycleGAN sinh ra. Từ trên xuống dưới: trường hợp mắc cài trùng màu môi và lợi (bên trái) và mắc cài đa sắc (bên phải), mắc cài có dây cung nhưng quá mỏng, khu vực chứa mắc cài có độ phân giải thấp.

Vì mắc cài có thể có nhiều màu sắc khác nhau nên model vẫn không thể xóa được những trường hợp có màu sắc đặc biệt như màu trùng với những thành phần của răng như hồng (lợi), răng (trắng). Chúng tôi giải quyết vấn đề này bằng một số phương pháp tăng cường ảnh. Đối với trường hợp dây cung, liên quan trực tiếp đến nguyên nhân độ phân giải thấp nên chúng tôi tiền xử lý bằng cách cắt phần miệng trước.

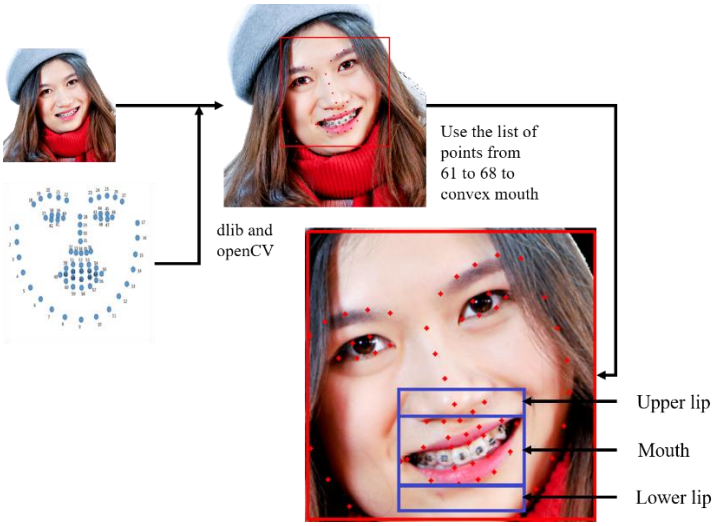
3.3. Data augmentation

Với mỗi điểm dữ liệu trong dataset **braces2teeeth**, chúng tôi thay đổi ngẫu nhiên kênh màu S và V trong không gian HSV để màu răng và màu mắc cài được thay đổi 10 lần khác nhau. Sau khi thử nghiệm, những mắc cài có màu sắc sặc sỡ đã được xóa, tuy nhiên model đã thay đổi sắc thái của ảnh khá nhiều.

Hình 6: Điểm dữ liệu gốc (góc trên bên trái) và những phiên bản khác sau khi áp dụng data augmentation.

Hình 7: Từ trái sang phải, ảnh gốc và ảnh do model CycleGAN + data augmentation sinh ra.

3.4. Crop mouth



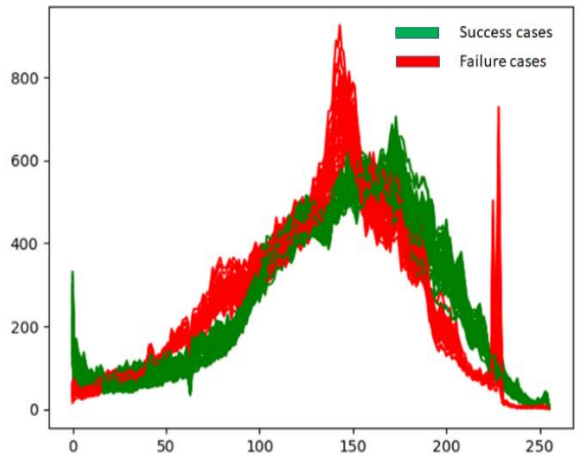
Hình 8: Phương pháp cắt ra vùng miệng, bằng dlib và OpenCV để tạo facial landmark detection. Phần mouth được đánh dấu bởi các point trên landmark có index từ 61 đến 68.

Chúng tôi sử dụng hàm OpenCV convex Rectangle để và lấy thêm nửa trên và nửa dưới để ảnh không bị biến dạng khi đưa vào model. Phương pháp này sẽ được áp dụng chủ yếu trong model Pix2Pix.

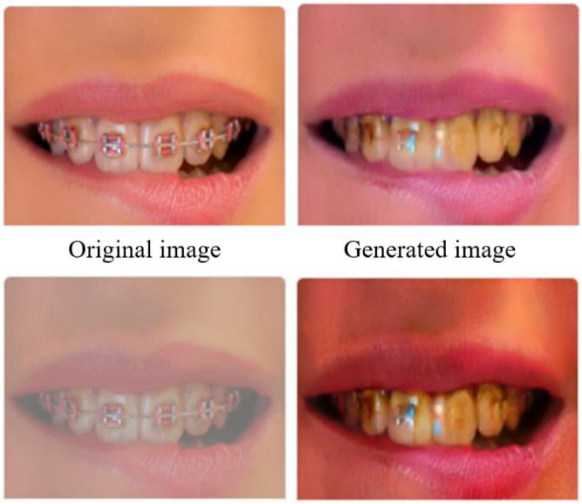
3.5. Cân bằng Histogram

Chúng tôi tiến hành phân tích histogram của trung bình RGB trong tập 200 ảnh test. Dễ thấy những trường hợp failure có 2 đỉnh ở khoảng 130 – 140 và 220 – 230 vì elastic tie thường có màu sắc sặc sỡ (sáng).

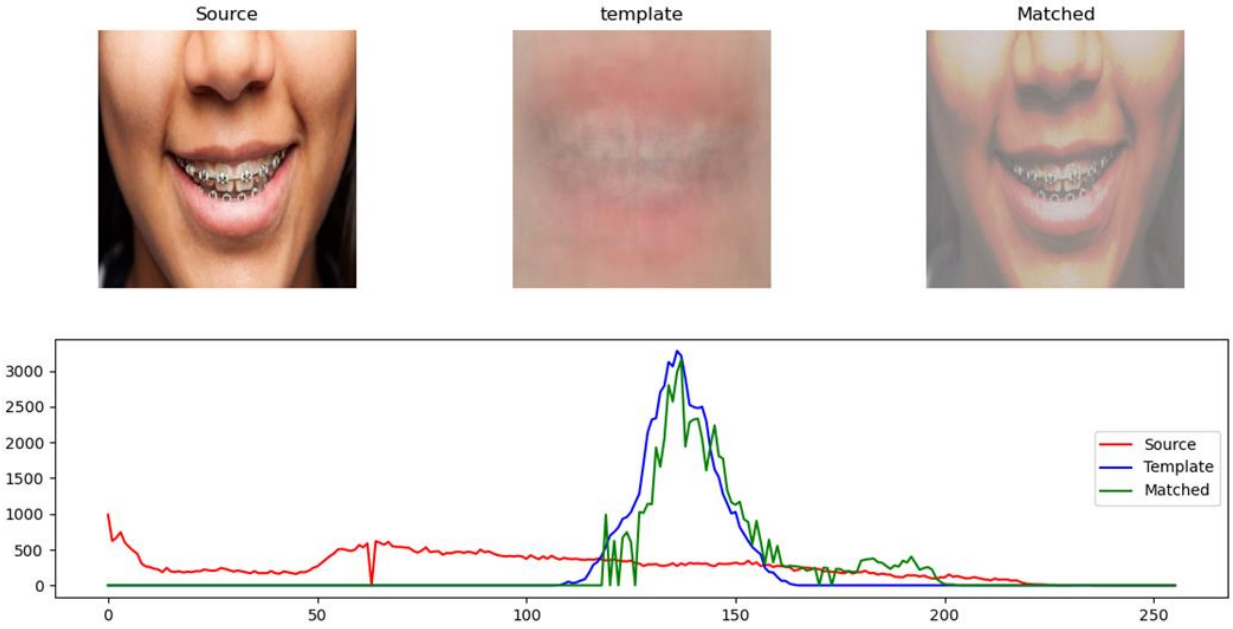
Chúng tôi thử nghiệm việc cân bằng histogram của những trường hợp failure với histogram trung bình của những trường hợp success (hình 9) bằng phương pháp nội suy tuyến tính được mô tả trong hình 11.



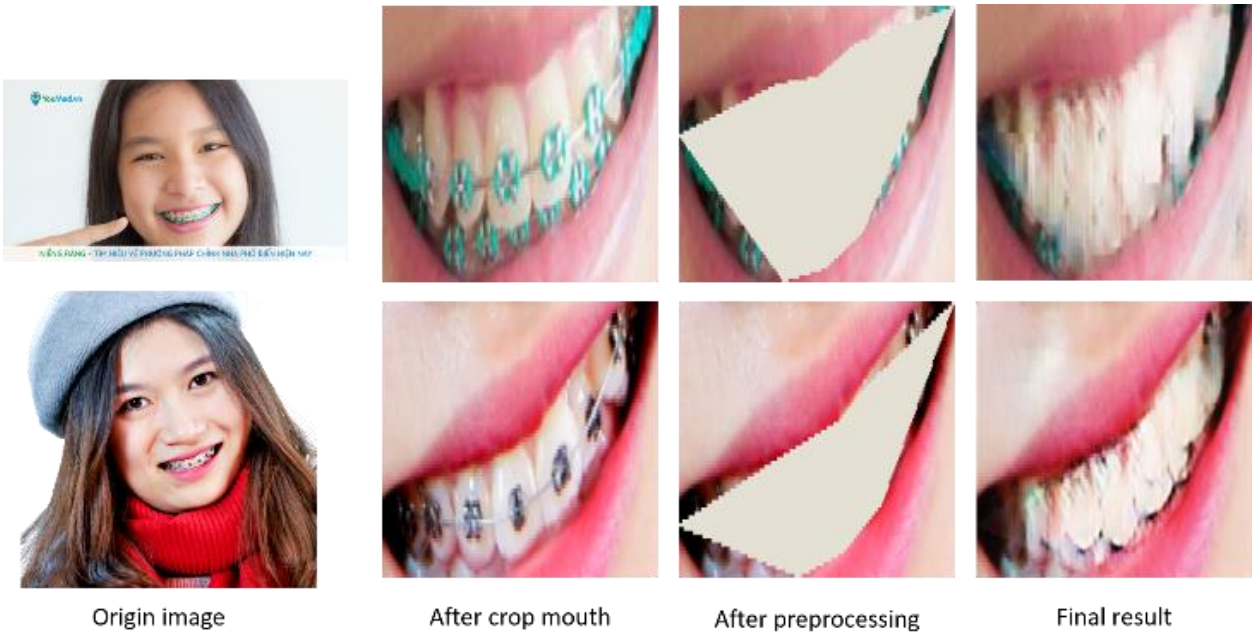
Hình 9: Histogram trung bình của những điểm dữ liệu thuộc tập test mà model CycleGAN đã xóa thành công (màu xanh) và những điểm dữ liệu thuộc tập test mà model CycleGAN đã không xử lý được (màu đỏ).



Hình 10: Kết quả thử nghiệm vẫn không tốt sau khi áp dụng phương pháp cân bằng histogram.

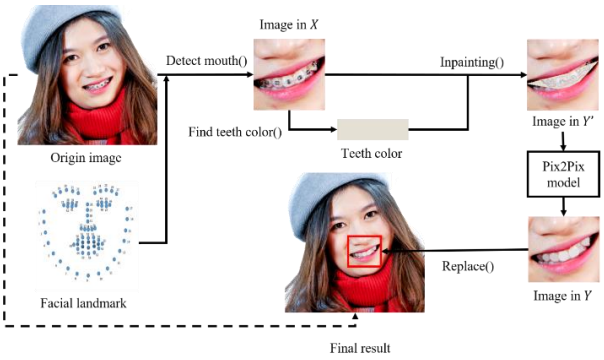


Hình 11: Quá trình cân bằng histogram bằng phương pháp nội suy tuyến tính, trong đó source là điểm dữ liệu cần điều chỉnh (histogram đỏ), template là histogram mẫu hướng đến (histogram màu xanh nước biển) và matched (histogram màu xanh lá) là histogram đã được biến đổi từ source.



Hình 12: Kết quả của model Pix2Pix với phương pháp xóa mắc cài *A*, lần lượt từ trái sang phải, ảnh gốc, phần mouth đã được cắt, selection zone được inpainting bằng màu răng và ảnh được model Pix2Pix trả về.

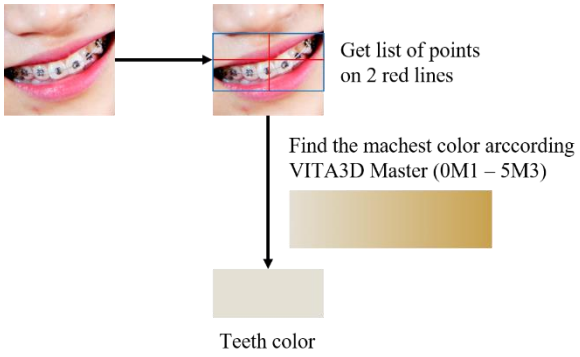
3.6. Pix2Pix with preprocessing



Hình 13: Tổng quan về toàn bộ quá trình xử lý, sau khi có được kết quả từ giai đoạn crop mouth như được mô tả trong phần 3.4. Chúng tôi tìm màu răng và inpainting trên selection zone (là vùng nằm trong môi), cuối cùng là đưa vào model Pix2Pix.

3.6.1. Tìm màu răng

Chúng tôi sử dụng bảng màu dựa trên VITA3D Master để tham chiếu trực tiếp đến những điểm ảnh lấy được có khả năng là răng. Sau đó 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 nào đó trên bảng mẫu.



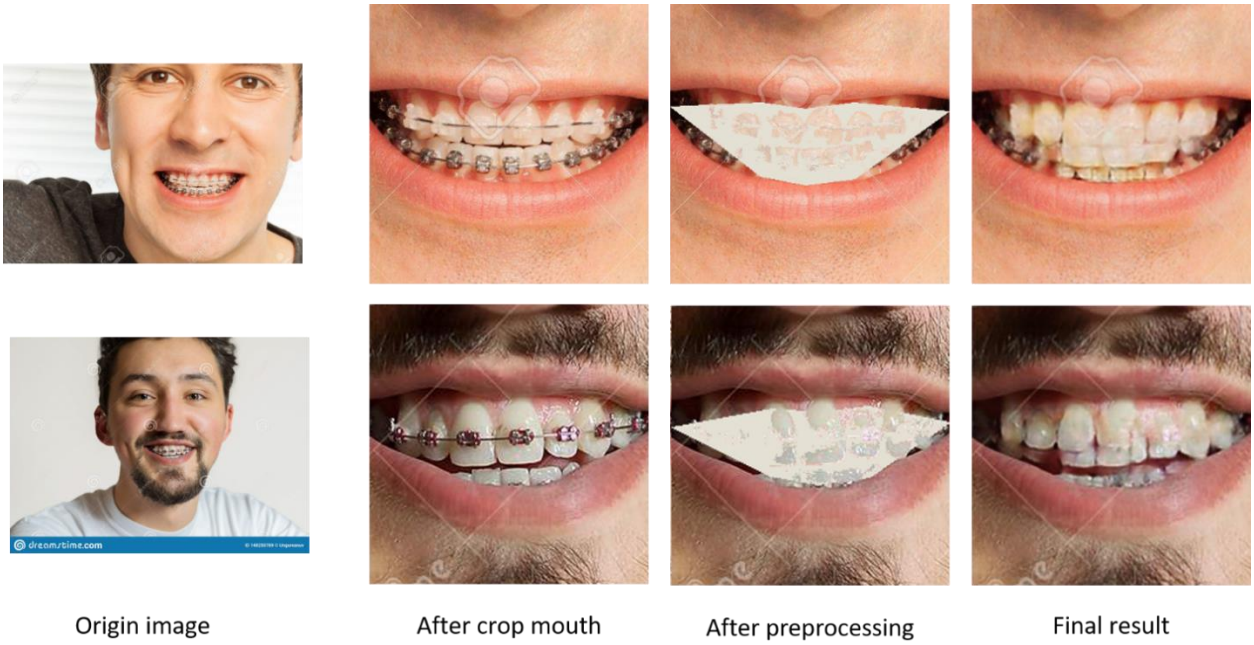
Hình 14: Phương pháp tìm màu răng.

3.6.2. Xóa mắc cài

Chúng tôi sử dụng hàm convexPolygon trong OpenCV các point có index từ 61 – 68 trên facial landmark để xác định selection zone. Chúng tôi thử nghiệm 3 phương pháp khác nhau để so sánh:

Phương pháp A (hình 12): Những pixel nằm trong selection zone được inpainting lại bằng màu răng.

Phương pháp B (hình 15): 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 chúng tôi chuyển ảnh sang không gian màu CIELab và sử dụng khoảng cách CIE2000 [20].



Hình 15: Kết quả khi của model Pix2Pix với phương pháp xóa mắc cài *B*, lần lượt từ trái sang phải, ảnh gốc, phần mouth đã được cắt, selection zone được inpainting bằng màu răng và ảnh được model Pix2Pix trả về.

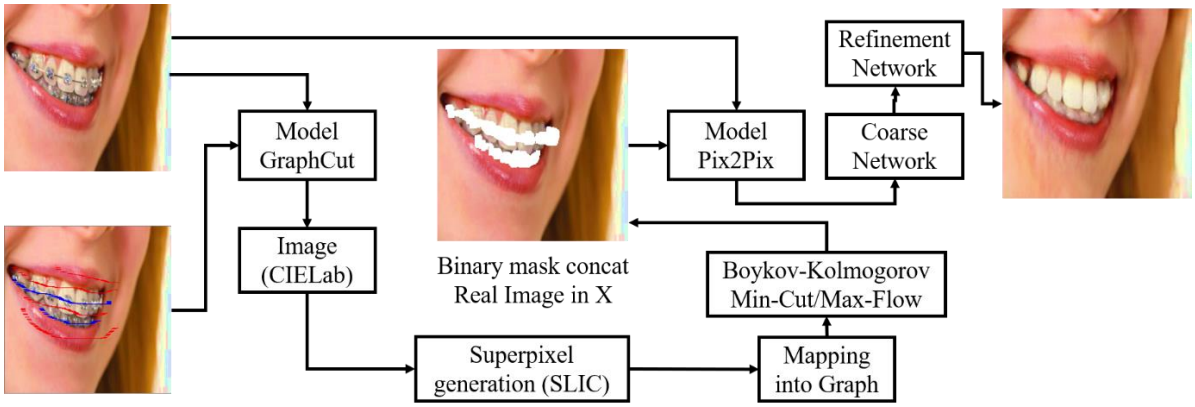
Hình 16: Vùng mouth đã được inpainting sau khi áp dụng phương pháp C.

Phương pháp C (hình 16): 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", chúng 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



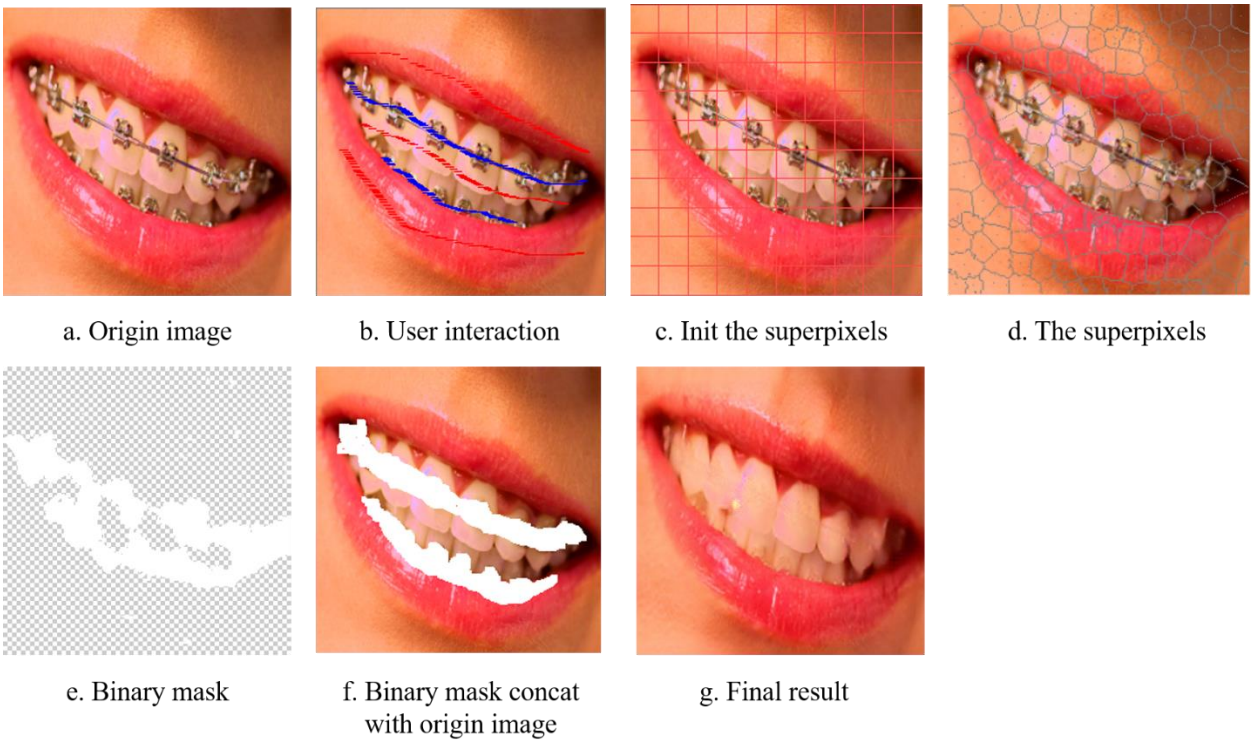
Hình 17: Sơ đồ tổng quan của mô hình GraphCut + Generative Image Inpainting with Contextual Attention

3.7. GraphCut + Generative Image Inpainting with Contextual Attention

Kết quả do hai model đã tương đối, tuy nhiên có một số trường hợp người dùng muốn can thiệp để đầu ra đạt được kết cấu theo như ý muốn. Do đó chúng tôi kết hợp hai model GraphCut [20] và Generative Image Inpainting with Contextual Attention [4] để hiện thực hóa nhu cầu trên (tôi sẽ gọi model này là **Inpainting** để ngắn gọn)

Model GraphCut sẽ nhận vào ảnh gốc (hình 17. a) và user interaction (hình 17. b) đánh dấu vị trí mắc cài (màu đỏ đánh dấu không phải mắc cài và màu xanh đánh dấu mắc cài). Model GraphCut ánh xạ ảnh sang tập superpixel (hình 17. c và 17. d) và sử dụng thêm thông tin từ user interaction để ánh xạ tiếp sang đồ thị ba chiều. Cuối cùng đưa đồ thị ba chiều vào thuật toán của 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. Chúng tôi ánh xạ ngược lại S và T về ảnh gốc và có được một binary mask với giá trị 1 đánh dấu mắc cài và giá trị 0 đánh dấu không phải mắc cài (hình 17. e).

Model Inpainting nhận ảnh gốc và binary mask, tiến hành inpainting tại những vị trí đã được đánh dấu trên binary mask và cuối cùng trả về ảnh răng không chứa mắc cài (hình 17. 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 17: Các giai đoạn xử lý ảnh trong model Inpainting.

4. Thử nghiệm

4.1. Dataset

Dataset được crawl tự động từ google image bằng công cụ của tác giả Yabin Zheng được công khai trên Github. Chúng tôi sử dụng các từ khóa liên quan đến braces và teeth trên nhiều ngôn ngữ khác nhau để tăng cường độ đa dạng của dữ liệu. Tất cả điểm dữ liệu trong 3 bộ dataset đều có kích thước 256 x 256 x 3.

Tên	Số lượng ảnh	Phân lớp
braces2teeth	2196	testX: 201 trainX: 805 testY: 240 trainY: 950
braces2teeth Augmentation	21960	testX: 2010 trainX: 8050 testY: 2400 trainY: 9500
teeth2	7000	train: 5000 val: 1000 test: 1000

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

4.2. Model CycleGAN và model Pix2Pix

Về kiến trúc model, chúng tôi sử dụng kiến trúc Unet vì kết quả đã được chứng minh trong [14]. Model CycleGAN sử dụng dataset braces2teeth và braces2teethAugmented, model có tổng cộng 28286000 weight (G_A: 11378000, G_B: 11378000, D_A: 2765000, D_B: 2765000). Model Pix2Pix sử dụng dataset teeth2 và có số lượng tham số bằng một nửa model CycleGAN. Cả hai đều được thực thi trên môi trường Google Colab và training trong 200 epoch trong vòng 168 tiếng.

4.3. GraphCut + Generative Image Inpainting with Contextual Attention

Model được thực thi bằng Tensorflow 1.15, trên phần cứng CPU Intel Xeon E3 12xx v2 (Ivy Bridge) 3GHz (8 processor), không có GPU. Model thử nghiệm xử lý khoảng 4s trên CPU với ảnh 256 x 256. Thời gian training đến là 900 tiếng, tổng số epoch hoàn thành là 40.

5. Kết quả

Chúng 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 chúng tôi có thể được tìm thấy tại đây.

5.1. Chỉ số đánh giá

AMT perceptual studies (AMT score) Để xác định chất lượng ảnh sinh ra braces → teeth, chúng tôi sử dụng phương pháp Amazon Mechanical Turk (AMT) theo như quy trình nghiên cứu từ Isola và cộng sự [16] nhưng với quy mô nhỏ hơn. Những người tham gia được cho xem một chuỗi các cặp ảnh, một là ảnh thật và một là ảnh giả (được tạo bởi model) và yêu cầu chọn hình ảnh mà họ cho là thật. Mỗi người tham gia chỉ được phép thử nghiệm một lần duy nhất. Tất cả các model đều được đánh giá trên cùng một test dataset.

5.2. Baseline

CoGAN [21] model này bao gồm 2 Generator domain X và domain Y cộng thêm một số trọng số rằng buộc tại lớp latent. Việc chuyển đổi ảnh từ domain X sang domain Y có thể thực hiện bằng cách tìm latent phù hợp sinh ra x và chuyển đổi nó sang domain Y.

BiGAN/ALI [23] bao gồm một generator $G: Z \rightarrow X$, với z là random noise. Model này có thêm một ánh xạ nghịch đảo $F: X \rightarrow Z$.

SimGAN [22] tương tự CycleGAN, model này sử dụng hàm adversarial loss để chuyển đổi từ X sang Y.

Model	**Training dataset**	**% Turker labeled real**
CoGAN	braces2teeth	$0.7\% \pm 0.5\%$
BiGAN/ALI	braces2teeth	$2.1\% \pm 0.5\%$
SimGAN	braces2teeth	$1.1\% \pm 1\%$
Inpainting	braces2teeth/teeth	$1.7\% \pm 0.3\%$
CycleGAN [14]	braces2teeth	**$28.3\% \pm 4.3\%$**
CycleGAN + Augmented data	braces2teeth Augmented	**$32.8\% \pm 4.7\%$**
Pix2Pix + preprocessing	teeth2	**$38\% \pm 5.2\%$**

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

6. Thảo luận

Mặc dù CycleGAN và những model áp dụng thêm những phương pháp preprocessing đã xử lý tốt nhiều trường hợp những vẫn còn những ngoại lệ chưa thể mắc cài hoàn hảo như đã trình bày trong phần 3.2 và trong hình 18.

Nguyên nhân cơ bản vẫn là do phân phối của dataset chưa đủ để thể hiện tất cả các trường hợp của bệnh nhân đeo niềng răng, ví dụ model không thể xử lý niềng răng vô hình, hoặc mắc cài màu sắc độc lạ như màu tím vì trong dataset không hề có những trường hợp đó. Chúng tôi cũng cảm nhận thấy sự khác biệt về mặt kết quả nếu có thể áp dụng triệt để hướng supervised so với unsupervised. Tuy nhiên, việc chuẩn bị các cặp điểm dữ liệu thật bao gồm ảnh đang trong quá trình niềng răng và ảnh sau khi tháo niềng răng mà không cần qua quá trình tiền xử lý gặp rất nhiều khó khăn như đã mô tả trong phần 1.

Lý do model Inpainting đạt kết quả khá thấp vì còn phụ thuộc vào số lượng epoch hoàn thành và chất lượng tương tác đến từ người dùng. Tuy nhiên chúng tôi nghĩ rằng model vẫn có khả năng xử lý tốt hơn nếu được đầu tư về sức mạnh tính toán.

Hình 18: Một số kết quả thử nghiệm. Bên trái mỗi cặp (ảnh gốc), bên phải mỗi cặp (ảnh do model sinh ra).

References

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).