VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

**UNIVERSITY OF INFORMATION TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE**

----- ⭘ 🕮 ⭘ -----



# Báo cáo Đồ án Cuối kì

***Đề tài:* Image Matting sử dụng U2-Net Model**

Giảng viên hướng dẫn: **Đỗ Văn Tiến**

Student group:

|  |  |
| --- | --- |
| Hoàng Gia Huy  Trang Trường Giang | 19521607  19521450 |

Ho Chi Minh City, Thursday, January 25, 2024

Mục lục

[Báo cáo Đồ án Cuối kì 1](#_Toc157441611)

[PART I. Introduction 3](#_Toc157441612)

[I.1. Overview 3](#_Toc157441613)

[I.2. Input và output của bài toán 3](#_Toc157441614)

[I.3. Database 4](#_Toc157441615)

[Giới thiệu 4](#_Toc157441616)

[Thu thập dữ liệu: 5](#_Toc157441617)

[Cách làm mờ khuôn mặt: 5](#_Toc157441618)

[Xử lý dữ liệu 6](#_Toc157441619)

[Tăng cường dữ liệu 6](#_Toc157441620)

[PART II. Phương pháp 7](#_Toc157441621)

[II.1. Residual U-Blocks (RSU) 7](#_Toc157441622)

[II.2. Cấu trúc U2-Net 10](#_Toc157441623)

[PART III. Training và đánh giá mô hình 12](#_Toc157441624)

[III.1.Các bước tiến hành train model: 12](#_Toc157441625)

[III.1.1Môi trường train và đánh giá: 12](#_Toc157441626)

[III.1.1.2. Về mô hình U2-Net 13](#_Toc157441627)

[III.1.1.3. Quá trình training model: 13](#_Toc157441628)

[III.1.1.4. Đánh giá mô hình: 14](#_Toc157441629)

[PART IV.Ứng dụng và hướng phát triển 15](#_Toc157441630)

[IV.1.Cải tiến: 15](#_Toc157441631)

[IV.2.Hướng phát triển trong tương lai: 16](#_Toc157441632)

[IV.3.Demo: 16](#_Toc157441633)

# PART I. Introduction

## I.1. Overview

Trong bối cảnh hiện tại, việc loại bỏ background được ứng dụng trong rất nhiều lĩnh vực như: chỉnh sửa ảnh hay video, thực tế ảo, tăng cường thực tế,… Từ những phim trường nổi tiếng, trên những bộ máy tính có hiệu suất lớn chỉ nhằm phục vụ việc tách đối tượng ra khỏi nền, đến những chiếc điện thoại smartphone trên tay chúng ta cũng đã có thể tách đối tượng ra khỏi nền của nó trong cùng một tấm ảnh. Để giải quyết vấn đề này, các phương pháp truyền thống thường đòi hỏi sự tương tác của người dùng, như vẽ một hộp giới hạn hay cung cấp “trimaps”. Các thách thức của bài toán bao gồm sự mơ hồ về màu sắc giữa phần đối tượng và phần nền, ranh giới đối tượng phức tạp và đối tượng trong suốt. Trong bài báo cáo này, nhóm sẽ giới thiệu một mô hình học máy có thể giải đáp những thách thức của bài toán Image Matting.

## I.2. Input và output của bài toán

- Input: Một tấm ảnh chụp chân dung của 1 hoặc 2 người ở thế giới thực với đa dạng background, kích thước tấm ảnh đầu vào là 512x512.



- Output: Là một ảnh mask có cùng kích thước với input, nơi các pixel sẽ có các giá trị từ 0-1 đại diện cho xác suất các pixel đó thuộc về foreground hay background. 0 là background, 1 là foreground, các pixel có giá trị nằm trong khoảng (0, 1) sẽ mang các đặc trưng của cả foreground và background.

A white silhouette of a person

Description automatically generated

## I.3. Database

### Giới thiệu

Nhóm sẽ sử dụng bộ dữ liệu bộ dữ liệu P3M-10k(Li, et al., 2021), đây là bộ dữ liệu chuẩn lớn đầu tiên được ẩn danh cho bài toán Privacy-Preserving Portrait Matting. P3M-10k bao gồm 10.000 hình ảnh chân dung mờ chất lượng cao kèm theo alpha mattes chất lượng cao. Những hình ảnh trong bộ dữ liệu sẽ có từ một hoặc tối đa là hai người trong cùng một tấm ảnh với phần bối cảnh đa dạng. Ngoài ra, những tấm ảnh có thể được chụp từ chính diện hoặc từ đằng sau lưng.

A person standing next to a pillar

Description automatically generatedA person and person with their faces over their faces

Description automatically generatedA person with long hair and beard smiling

Description automatically generated

*Figure 1: few images extracted from our dataset*

### Thu thập dữ liệu:

"Specifically, we carefully collect, filter, and annotate about 10,000 high-resolution images from the Internet with free use license." (Li et al., 2021). Bộ dữ liệu được thu thập, chọn lọc và ghi chú với tầm 10000 tấm ảnh từ nguồn Internet với bản quyền tự do. Bộ dữ liệu sẽ bao gồm những tấm ảnh được làm mờ khuôn mặt do cần phải đáp ứng được yêu cầu của bài toán Privacy-Preserving Portrait Matting. Do đó, với những tấm ảnh chụp chân dung của người không phải là người nổi tiếng sẽ được làm mờ khuôn mặt.

Nhóm cũng sẽ dùng cách tương tự để thu thập dữ liệu từ những trang: và sau đó nhóm cũng sẽ thực hiện phương pháp làm mờ khuôn mặt như nhóm nghiên cứu.

### Cách làm mờ khuôn mặt:

Để làm mờ khuôn mặt, tác giả sử dụng các thuật toán phát hiện điểm mốc khuôn mặt (facial landmark detection) để tạo ra những tấm ảnh với khuôn mặt được làm mờ, cụ thể:

1. Sử dụng thuật toán điểm mốc để lấy các điểm mốc trên khuôn mặt ( mắt, mũi, miệng,…)
2. Tự động tạo ra mask dựa trên các điểm mốc, bao gồm toàn bộ khuôn mặt.
3. Loại bỏ phần chuyển tiếp giữa khuôn mặt và nền để chỉ giữ phần khuôn mặt rõ ràng.
4. Sử dụng Gaussian blur để làm mờ phần trong mask.

A collage of a person with a purple face

Description automatically generated

### Xử lý dữ liệu

Trước khi huấn luyện, nhóm sẽ chia bộ dữ liệu thành 2 phần từ tổng hơn 11,500 tấm ảnh từ 10000 tấm có sẵn từ bộ dữ liệu và 1500 tấm do nhóm thu thập thành:

1.Training dataset: Bao gồm tầm 10000 tấm ảnh, đối với những ảnh chụp chân dung của người không phải người nổi tiếng thì sẽ được làm mờ khuôn mặt để đảm bảo tính bảo mật.

2.Validation dataset: Bao gồm 1,500 tấm ảnh, nhóm sẽ chọn những tấm ảnh của người nổi tiếng và phần foreground sẽ chiếm khoảng 50% diện tích tấm ảnh.

### Tăng cường dữ liệu

Tiến hành tăng cường dữ liệu trên tập train. Quá trình tăng cường được thực hiện bằng thử viện Albumentation, với các kĩ thuật:

|  |  |
| --- | --- |
| Tên kĩ thuật | Công dụng |
| HorizontalFlip | Các biến đổi gồm việc lật ngang hình ảnh |
| RandomBrightnessContrast | Thay đổi sáng và độ tương phản ngẫu nhiên |
| ElasticTransform | Biến đổi đàn hồi |
| CLAHE | Cân bằng Histogram |

# PART II. Phương pháp

Chúng tôi chọn mô hình U2-Net là mô hình chính cho bài toán Image Matting. Mô hình U2-Net có cấu trúc U lồng 2 cấp độ, được công bố vào năm 2020 bởi nhóm nghiên cứu dẫn dắt bởi Xuebin Qin. Mô hình đã đạt được giải thưởng “2020 Pattern Regconition BEST PAPER AWARD” với đề tài: “U2-Net: Going deeper with nested U-structure for salient object dectection.” Mô hình U2-Net sẽ không sử dụng bất kì pretrained backbones từ những bài toán phân loại hình ảnh từ trước. Cấu trúc chữ U lồng 2 cấp độ: ở cấp độ thấp nhất, là một khối ReSidual U-Block(RSU), nó có thể trích xuất đặc trưng đa tỷ lệ mà không làm giảm độ phân giải của feature map. Ở cấp độ cao nhất, giống như cấu trúc của U-Net, mỗi giai đoạn được lấp đầy bởi khối RSU.

## II.1. Residual U-Blocks (RSU)

Được lấy cảm hứng từ cấu trúc của mô hình U-Net, ReSidual U-block (RSU) để trích xuất những đặc trưng đa tỉ lệ trong cùng một giai đoạn mà không giảm chất lượng bản đồ đặc trưng. Cấu trúc của RSU-L(*Cin,M,Cout )* với L là số lượng lớp của phần giải mã; *Cin*, *Cout* là kênh đầu vào và đầu ra; M là số lượng kênh đầu ra của lớp bên trong là một lớp tích chập (convolutional layers) trong khối RSU. Tổng hợp sẽ có 3 thành phần như sau:

1. Là một input convolution layer để biến cái input feature map x có (HxWxCin) thành intermediate map F1(x) với kênh của Cout. Đây là lớp tích chập đơn giản để trích xuất các features cục bộ của ảnh.
2. Là một dạng như U-net sử dụng đối xứng cấu trúc encoder và decoder với nhau có chiều dài L là 7 mà nó sẽ lấy Intermediate feature map F1(x) là input và học để trích xuất và mã hóa nó thành muti-scale feature ( nó có thể học được cái đặc trưng từ nhiều tỷ lệ khác nhau). Các muti-scale feature thì được trích xuất xuống dần các feature maps và sẽ được mã hóa lại thành các feature maps có độ phân giải cao bằng cách lấy mẫu tăng dần, ghép nối và tích chập với nhau. Lớp convolutional cuối cùng trong chuỗi có tốc độ giãn nở là 2, được biểu thị bằng ( d=2 ), cho phép mạng có trường tiếp nhận rộng hơn và thu được nhiều thông tin theo ngữ cảnh hơn mà không cần tăng số lượng tham số.
3. Cuối cùng là sự kết hợp của đặc trưng cục bộ được lấy từ ban đầu với đặc trưng đa tỉ lệ qua bước U-block: F1(**x**)+U(F1(**x**)).

A screenshot of a computer

Description automatically generated A diagram of a block diagram

Description automatically generated

Vậy tại sao chúng ta sử dụng khối RSU thay vì các thiết kế CNN hiện đại như VGG, ResNet, DenseNet, v.v., các bộ lọc tích chập nhỏ có kích thước 1×1 hoặc 3×3 là thành phần được sử dụng thường xuyên nhất để trích xuất đặc trưng. Chúng được ưa chuộng vì chúng yêu cầu ít không gian lưu trữ hơn và hiệu quả về mặt tính toán. Hình dưới đây minh họa các khối tích chập hiện có điển hình với các trường tiếp nhận nhỏ. Bản đồ đặc trưng đầu ra của các lớp nông chỉ chứa các đặc điểm cục bộ vì trường tiếp nhận của các bộ lọc 1×1 hoặc 3×3 quá nhỏ để thu thập thông tin toàn cầu. Để có được nhiều thông tin toàn cầu hơn ở các bản đồ đặc trưng có độ phân giải cao từ các lớp nông, ý tưởng trực tiếp nhất là mở rộng trường tiếp nhận. Khối (d) hiển thị một khối giống như khối inception, khối này cố gắng trích xuất cả các tính năng cục bộ và không cục bộ bằng cách mở rộng các trường tiếp nhận bằng cách sử dụng các tích chập giãn nở. Tuy nhiên, việc tiến hành nhiều tích chập giãn nở trên bản đồ tính năng đầu vào (đặc biệt là trong giai đoạn đầu) với độ phân giải gốc đòi hỏi quá nhiều tài nguyên bộ nhớ và tính toán.

A screenshot of a computer

Description automatically generated

## II.2. Cấu trúc U2-Net

A diagram of a diagram of a graph

Description automatically generated with medium confidence

Cấu trúc của U2-Net có 3 phần chính : (1) 6 giai đoạn mã hóa, (2) 5 giai đoạn giải mã và (3) một module tổng hợp bản đồ saliency được gắn với các giai đoạn giải mã và giai đoạn mã hóa cuối cùng :

1. Trong các giai đoạn mã hóa En 1, En 2, En 3 và En 4, chúng tôi sử dụng các khối U còn lại lần lượt là RSU-7, RSU-6, RSU-5 và RSU-4. Như đã đề cập trước đó, “7”, “6”, “5” và “4” biểu thị chiều cao (L) của khối RSU. Đối với các feature maps có chiều cao và chiều rộng lớn, chúng tôi sử dụng L lớn hơn để thu được nhiều thông tin tỷ lệ lớn hơn. Do đó, trong cả hai giai đoạn En 5 và En 6, RSU-4F là RSU là một phiên bản giãn nở, trong đó thay thế các hoạt động gộp và lấy mẫu lại bằng các convolutions giãn nở.
2. Các giai đoạn giải mã có cấu trúc tương tự như các giai đoạn mã hóa đối xứng của chúng. Trong De 5, chúng tôi cũng sử dụng phần dư U-block RSU-4F phiên bản giãn nở tương tự như phiên bản được sử dụng trong các giai đoạn mã hóa En 5 và En 6. Mỗi giai đoạn giải mã lấy sự kết hợp của các bản đồ tính năng được lấy mẫu từ giai đoạn trước đó và các bản đồ từ giai đoạn bộ mã hóa đối xứng của nó làm đầu vào, xem Hình 5.
3. Phần cuối cùng là mô-đun tổng hợp saliency map fusion module được sử dụng để tạo ra các bản đồ xác suất nổi. trước tiên tạo ra sáu xác suất saliency maps đầu ra bên từ các giai đoạn En 6, De 5, De 4, De 3, De 2 và De 1 bằng lớp chập 3 × 3 và hàm sigmoid. Sau đó, nó lấy mẫu các logit (đầu ra tích chập trước các hàm sigmoid) của ánh xạ độ mặn đầu ra bên với kích thước hình ảnh đầu vào và kết hợp chúng với thao tác nối, sau đó là lớp chập 1×1 và hàm sigmoid để tạo ra bản đồ xác suất độ mặn cuối cùng S*fuse.*

# PART III. Training và đánh giá mô hình

Như đã giới thiệu trước đó, U2-Net mô hình sẽ không sử dụng những kiến trúc cũng như tiếp tục pre-trained từ những mô hình trước đó, nên việc huấn luyện mô hình từ đầu là điều có thể thực hiện được. Nếu như đã huấn luyện mô hình từ đầu vậy thì tại sao không thử huấn luyện mô hình khác so với những phần U2-Net đã được công bố trước đó? Nhận thấy việc đó khả thi nên chúng tôi đã chọn fine-tuned mô hình trên bộ dữ liệu P3M-10k.

## III.1.Các bước tiến hành train model:

### III.1.1Môi trường huấn luyện và đánh giá:

#### III.1.1.1Môi trường huấn luyện:

Tiến hành train trên Kaggle, Kaggle là một nền tảng trực tuyến cho cộng đồng Machine Learning (ML) và Khoa học dữ liệu. Kaggle cho phép người dùng chia sẻ, tìm kiếm các bộ dữ liệu; tìm hiểu và xây dựng models, tương tác với những nhà khoa học và kỹ sư ML trên toàn thế giới; tham gia các cuộc thi để có cơ hội chiến thắng những giải thưởng giá trị. Người dùng Kaggle sẽ được hỗ trợ Graphic Processing Unit (GPU) và gần đây có thêm Tensor Processing Unit (TPU) để tăng tốc độ tính toán trong quá trình training cũng như inference.

A white text on a white background

Description automatically generated

Kaggle có một điểm tiện lợi hơn so với Google Colab là ở chỗ ta có thể save lại các version, nên ta có thể lưu trữ các file weight, cũng như những file log. Tuy nhiên, do có giới hạn về thời gian, cụ thể là 30 giờ/tuần, sẽ cập nhật lại vào thứ 7 hằng tuần nên quá trình train diễn ra thường bị ngắt quãng.

### III.1.1.2. Về mô hình U2-Net

Mô hình U2-Net có 2 version: mô hình U2-Net, và mô hình U2-Net-lite. Để tiết kiệm thời gian cũng như tài nguyên, nhóm em sẽ sử dụng phiên bản lite của mô hình U2-Net.

### III.1.1.3. Quá trình training model:

Upload dataset lên kaggle, define mô hình U2-Net bản lite

Chuẩn bị lại các file cần thiết và tài nguyên để chuẩn bị cho việc training:

1. Cài đặt các thư viện cần thiết
2. Load dataset, file train sẽ chứa các ảnh trong tập train, file validation sẽ chứa các ảnh trong tập validate
3. Fine-tune hàm loss, sử dụng tổ hợp 3 hàm loss: alpha\_loss(hàm tính toán mean squared error giữa y\_true và y\_predict của alpha matte, Hàm loss này thường được sử dụng trong các tác vụ làm mờ hình ảnh trong đó mục tiêu là ước tính độ trong suốt của từng pixel trong hình ảnh), hàm ssim\_loss(Hàm mất này thường được sử dụng trong các tác vụ làm mờ hình ảnh trong đó mục tiêu là ước tính độ trong suốt của từng pixel trong hình ảnh), và cả binary cross entropy loss. Lí do sử dụng tổ hợp này là vì nó giúp mô hình có thể bắt được những chi tiết nhỏ tốt hơn so với phiên bản gốc được sử dụng.

Tỉnh chỉnh các thông số:

1. Width, height =512x512
2. Batch\_size = 4
3. Learning\_rate = 1e-7
4. Numper of epochs = 500
5. Lưu lại kết quả tốt nhất sau 5 epochs
6. File weights được lưu tại /kaggle/output
7. Tiến hành train model.

Notebook: <https://www.kaggle.com/code/huyhonggia/u2net>

### III.1.1.4. Đánh giá mô hình:

Sau khi thực hiện quá trình training model, để xác định model của chúng ta có đủ tốt hay không cũng như đảm bảo khả năng nhận diện trong tương lai, ta cần có một phương pháp đánh giá với tiêu chí cụ thể. Đối với bài toán Image Matting, các phương pháp đánh giá thường được sử dụng là MAD, MSE, SAD, BCE,… Trong bài toán này, nhóm quyết định sử dụng 3 thang đo gồm: MAD,MSE, SAD để đánh giá độ chính xác của model. Cũng như có thể so sánh với các model khác khi huấn luyện trên dataset P3M-10k.

#### III.1.1.4.1. Khái niệm:

MSE( Mean Squared Error) : MSE là thang đo cho phép xác định độ lệch không khí của giá trị dự đoán và giá trị thực tế. Để tính MSE, ta tính tổng các giá trị lệch không khí của các giá trị dự đoán và giá trị thực tế, sau đó chia cho số lượng các giá trị dự đoán và giá trị thực tế.

MAD (Mean Absolute Difference): MAD là thang đo cho phép xác định độ lệch trung bình của giá trị dự đoán và giá trị thực tế. Để tính MAD, ta tính tổng các giá trị lệch trung bình của các giá trị dự đoán và giá trị thực tế, sau đó chia cho số lượng các giá trị dự đoán và giá trị thực tế.

SAD (Sum of Absolute Differences): SAD là thang đo cho phép xác định độ lệch trung bình của giá trị dự đoán và giá trị thực tế. Để tính SAD, ta tính tổng các giá trị lệch trung bình của các giá trị dự đoán và giá trị thực tế.

#### III.1.1.4.2. Đánh giá:

Sau khi huấn luyện mô hình trên 450 epochs, mỗi epoch tốn khoảng 1 tiếng hơn, giá trị các hàm loss trên tập validation không thay đổi đáng kể, còn hàm loss thì có xu hướng giảm nhẹ xung quanh 0.985. Cho rằng mô hình đã hội tụ nên nhóm dừng train, chọn ra best.weights để tiến hành đánh giá.

Thêm bảng so sánh

|  |  |  |  |
| --- | --- | --- | --- |
| Mô hình | SAD | MSE | MAD |
| P3M-Net | 6.24 | 0.0015 | 0.0036 |
| StyleMatte | 6.97 | 0.0019 | 0.004 |
| GFM | 13.20 | 0.0050 | 0.0080 |
| U2-Net lite (Ours) | 10217.56 | 0.0062 | 0.0097 |

Từ kết quả, ta đúc kết được rằng, mô hình chưa thật sự hiệu quả do thiếu tài nguyên huấn luyện cũng như phương pháp tiếp cận chưa tốt. Cần phải huấn luyện thêm để có thể đạt được kết quả tiệm cận với với những mô hình khác, thậm chí là tốt hơn vì đây là một mô hình đủ tốt để có thể phát triển trong tương lai

# PART IV.Ứng dụng và hướng phát triển

## IV.1.Cải tiến:

Để có thể đưa mô hình này vào ứng dụng rộng rãi, cần phải cải tiến về một số khía cạnh:

* Về data:
  + - Thu thập thêm những hình ảnh để có thể tăng cường dữ liệu. Có thể tăng cường bằng cách thu thập hình ảnh được chụp của các diễn viên, nghệ sĩ người nổi tiếng.
    - Sử dụng các thư viện tăng cường data mạnh mẽ như Albumentation để có thể thay đổi.
    - Huấn luyện bộ dữ liệu bằng model U^2-Net phiên bản đầy đủ và với số lượng epochs lớn hơn.
    - Thay đổi phương pháp tiếp cận cũng như thay đổi cách fine-tune mô hình để có thể cải thiện mô hình và kết quả dự đoán.

## IV.2.Hướng phát triển trong tương lai:

* Làm cái ứng dụng remove background chạy Real-time.
* Loại bỏ và thay thế bằng nền khác.
* Ứng dụng nhiều hơn và các ứng dụng photoshop như là một công cụ hỗ trợ.

## IV.3.Demo:

Nhóm đã tổng hợp các bước cài đặt cũng như tiến hành chạy demo trong file Manual.docx

Source code: <https://github.com/ItsHuyne/CS406.O12.KHCL>

## IV.4.Phân công công việc:

A screenshot of a phone

Description automatically generated

**REFERENCES**

|  |  |
| --- | --- |
|  | [1] [U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection](https://arxiv.org/abs/2005.09007)  [2] [Dataset P3M-10k](https://paperswithcode.com/dataset/p3m-10k)  [3] [Privacy -Preserving Portrait Matting](https://arxiv.org/abs/2104.14222) |
|  |  |