

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**Tách phong đối tượng người trong ảnh sử dụng
kỹ thuật phân vùng ngữ nghĩa FCN-CRFs**

**Sinh viên thực hiện: Nguyễn Nhật Tín
Mã số: B1507321
Khoá: 41**

Cần Thơ, 5/2019

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**Tách phong đối tượng người trong ảnh sử dụng
kỹ thuật phân vùng ngữ nghĩa FCN-CRFs**

**Giảng viên hướng dẫn:
TS. Thái Minh Tuấn**

**Sinh viên thực hiện: Nguyễn Nhật Tín
Mã số: B1507321
Khoá: 41**

Cần Thơ, 5/2019

NHẬN XÉT CỦA GIẢNG VIÊN

[illegible]

LỜI CAM ĐOAN

Tôi xin cam đoan tất cả nội dung cùng số liệu được trình bày trong luận văn đều minh bạch, trung thực được trích dẫn nguồn rõ ràng và được thực hiện bởi tôi dưới sự hỗ trợ, giúp đỡ của giảng viên hướng dẫn Tiến sĩ Thái Minh Tuấn.

Cần Thơ, ngày ... tháng ... năm ...

Người viết

Nguyễn Nhựt Tín

LỜI CẢM ƠN

Luận văn được thực hiện dưới sự hỗ trợ tận tình của Tiến sĩ Thái Minh Tuấn, Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ. Với sự giúp sức cả về định hướng đề tài và sự cần mẫn chỉ bảo, chỉnh sửa trong toàn bộ luận văn, tôi đã được Thầy truyền lại kinh nghiệm và kỹ thuật để thực hiện một bài báo cáo máy học đúng phương pháp. Do đó, tôi xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn của mình.

Bên cạnh đó, tôi cũng xin cảm ơn các thầy cô đã tạo điều kiện để tôi được tiếp xúc với nền tảng kiến thức khác nhau thông qua các học phần trong chương trình đào tạo, những kiến thức cơ sở sẽ là hành trang tốt để tôi có thể thực hiện những công việc của mình sau này.

Và quan trọng hơn, tôi cũng xin cảm ơn gia đình, bạn bè, đồng nghiệp đã động viên, góp ý và tin tưởng tôi rất nhiều trong suốt thời gian thực hiện luận văn.

Tuy rằng có sự nỗ lực không nhỏ khi thực hiện báo cáo nhưng những sai sót vẫn không tránh khỏi, hy vọng sẽ nhận được sự góp ý của thầy cô và bạn bè để luận văn ngày một tốt hơn.

Xin gửi lời cảm ơn chân thành và ý nghĩa!

MỤC LỤC

1	GIỚI THIỆU	1
1.1	Tổng quan về tách phoneme	1
1.2	Mục tiêu đề tài	2
1.3	Bố cục của bài báo cáo luận văn	3
2	CƠ SỞ LÝ THUYẾT	4
2.1	Mạng tích chập đầy đủ (Fully Convolutional Network):	4
2.1.1	Bộ giảm mẫu - mã hóa (Downsampling - Encoder):	6
2.1.2	Bộ tăng mẫu - giải mã (Upsampling - Decoder):	9
2.1.3	Gradient và cập nhật trọng số:	15
2.2	Hậu xử lý (Post-processing):	19
2.2.1	Trường điều kiện ngẫu nhiên (CRFs):	19
2.2.2	Hàm năng lượng (Energy function):	20
2.2.3	Suy luận (Inference):	22
3	KẾT QUẢ THỰC HIỆN	23
3.1	Phân vùng ngữ nghĩa và ứng dụng tách phoneme:	23
3.1.1	Sơ đồ chức năng ứng dụng:	23
3.1.2	Phân vùng ngữ nghĩa:	24
3.1.3	Hàm mất mát:	26
3.1.4	CRFs và các mô hình thống kê:	28
3.1.5	Truyền thông điệp CRFs kết hợp tích chập Gaussian:	29
3.2	Tập dữ liệu:	31
3.2.1	Thu thập dữ liệu:	31
3.2.2	Phân tích tập dữ liệu sử dụng:	32
3.3	Kiểm tra kết quả:	32
3.3.1	Các thông số sử dụng:	33
3.3.2	Tính toán các độ đo:	33
3.3.3	Kết quả trên tập dữ liệu huấn luyện:	35
3.3.4	Kết quả trên tập dữ liệu kiểm tra:	37
3.4	Kết quả vận hành mô hình:	39
3.4.1	Kịch bản vận hành:	39
3.4.2	Một số kết quả phân vùng ngữ nghĩa:	41
4	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	42
4.1	Nhận xét kết quả đạt được:	42
4.2	Hạn chế:	42
4.3	Hướng phát triển:	43

DANH SÁCH HÌNH VẼ

1.1	Phát hiện đối tượng thông thường (Nguồn: www.cgtrader.com)	1
1.2	Phân vùng ngữ nghĩa đối tượng (Nguồn: www.cgtrader.com)	2
2.1	Mạng nơ-ron tích chập truyền thống [Nguồn: towardsdatascience.com]	4
2.2	Các lớp tích chập thuần nối tiếp	5
2.3	Phép toán tích chập	6
2.4	Những bộ lọc phổ biến [Nguồn: medium.com]	7
2.5	Bộ lọc và mối quan hệ với ảnh	8
2.6	Kết quả ví dụ của lớp thăm dò	9
2.7	Kết quả ví dụ phép tích chập thuần	10
2.8	Bộ lọc sau khi thay đổi	10
2.9	Tích chập kiểm tra bộ lọc sau thay đổi	11
2.10	Tích chập với bộ lọc chuyển vị	12
2.11	Kết quả mô phỏng hình ảnh thu được [Nguồn: [1]]	13
2.12	Kết quả hình ảnh sau khi chuyển tiếp nối kết [Nguồn: [1]]	13
2.13	Mô phỏng quá trình chuyển tiếp nối kết [Nguồn: i.stack.imgur.com]	14
2.14	Tích chập với bộ lọc 1x1	14
2.15	Mô phỏng quá trình kết hợp toàn bộ	15
2.16	Sơ đồ mối quan hệ giữa các biểu thức	16
2.17	Mô phỏng quá trình lan truyền ngược	17
2.18	Đồ thị trường ngẫu nhiên	19
2.19	Trường điều kiện ngẫu nhiên	20
3.1	Sơ đồ vận hành ứng dụng	23
3.2	Phân vùng ngữ nghĩa cắt ghép ảnh	24
3.3	Kiến trúc sử dụng trong bài báo cáo	25
3.4	Nhãn của ảnh [Nguồn: www.jeremyjordan.me]	26
3.5	Dữ liệu mô hình đoán được [Nguồn: www.jeremyjordan.me]	27
3.6	So sánh kết quả đầu ra	28
3.7	So sánh các loại đồ thị xác suất.	29
3.8	So sánh ảnh chưa xử lý và đã xử lý	30
3.9	Đồ thị so sánh các biến thể của FCN trên tập huấn luyện.	36
3.10	Đồ thị so sánh các biến thể của FCN trên tập kiểm tra.	38
3.11	Sơ đồ chi tiết hệ thống và hình ảnh	40
3.12	Kết quả cuối cùng.	41

DANH SÁCH BẢNG

3.1	Các tập dữ liệu phổ biến cho phân vùng ngữ nghĩa ảnh	31
3.2	Các tập dữ liệu được sử dụng	32
3.3	Thông tin các loại nhãn trong tập dữ liệu sử dụng	32
3.4	Phân chia tập dữ liệu huấn luyện	32
3.5	Các thông số dùng để huấn luyện FCN	33
3.6	Các thông số dùng để tối ưu CRFs	33
3.7	Ma trận contingency	34
3.8	So sánh các biến thể của FCN trên tập huấn luyện.	35
3.9	Kết quả tập huấn luyện FCN-32s	35
3.10	Kết quả tập huấn luyện FCN-16s	35
3.11	Kết quả tập huấn luyện FCN-8s	35
3.12	Kết quả tập kiểm tra FCN-32s	37
3.13	Kết quả tập kiểm tra FCN-16s	37
3.14	Kết quả tập kiểm tra FCN-8s	37
3.15	Kết quả tập kiểm tra sau khi sử dụng CRFs	38

TỪ ĐIỂN CHÚ GIẢI

2D Two-dimensional space. 7

CNN Convolutional Neural Network. 4, 6, 9, 24, 26

CPU Central Processing Unit. 39

CRFs Conditional Random Fields. 2, 3, 19, 21, 28–30, 33, 37–40, 42

FCN Fully Convolutional Networks. vii, viii, 2, 3, 5, 9, 14, 24, 26, 30–33, 35–38, 40, 42

GPU Graphics Processing Unit. 35

HMM Hidden Markov Model. 28, 29

IoU Intersection over Union. 34

KL-divergence Kullback–Leibler divergence. vii, viii, 22, 42

MEMM Maximum-Entropy Markov Model. 28, 29

MP Max-Pooling. 8

POS Pixel objectness. 39

RGB Red-Green-Blue. 7

RNN Recurrent Neural Network. 24

SGD Stochastic Gradient Descent. 18

TÓM TẮT

Kỹ thuật phân vùng ngữ nghĩa có nhiều ứng dụng trong lĩnh vực đồ họa kỹ thuật số, xác định vị trí chính xác đối tượng được miêu tả trong ảnh và cắt ghép, chỉnh sửa chúng mang lại những trải nghiệm thú vị, thu hút những ý tưởng sáng tạo để hình thành những ứng dụng vô cùng tiện ích. Hướng tiếp cận của tác giả là tạo ra ứng dụng có khả năng tách phong ảnh mang tới sự độc đáo khi đối tượng người có thể xuất hiện với nhiều phong nền khác nhau mặc dù chỉ chụp ở một chỗ nhất định.

Trong bài báo cáo này, tác giả tận dụng phương pháp tính toán của mạng tích chập đầy đủ (FCN) để giảm mẫu và tăng mẫu lại bằng tích chập chuyển vị (Transposed Convolution) mang lại chỉ số meanIU trên tập đánh giá (130 tấm ảnh) bằng 74.5% và kỹ thuật trường điều kiện ngẫu nhiên với phương pháp suy luận tối ưu KL-divergence ($Inference = 20$, $Scale = 0.7$) mang lại chỉ số meanIU bằng 80.0%. Ứng dụng cho phép tinh chỉnh các tham số và kết hợp nhiều loại ảnh nền khác nhau.

Từ khóa: *phân vùng ngữ nghĩa, mạng tích chập đầy đủ, trường điều kiện ngẫu nhiên.*

ABSTRACT

The semantic image segmentation technique has many applications in digital graphics, determines the exact location of the object described in the image and cuts, edits them, brings interesting experiences, attracts innovative ideas to form the useful applications. The author's approach direction is to create an application that is capable of separating photo fonts, bringing uniqueness when the object can appear with many different backgrounds even if taken from one place.

In this report, the author uses the calculation method of Fully Convolutional Network (FCN) to reduce the sample and increase the sample by transposed convolution to bring the meanIU index on the assessment set (130 images) equals 74.5% and conditional random fields technique with optimal KL-divergence method (*Inference* = 20, *Scale* = 0.7) gives meanIU index equal to 80.0%. The application allows to adjust parameters and combine different types of background images.

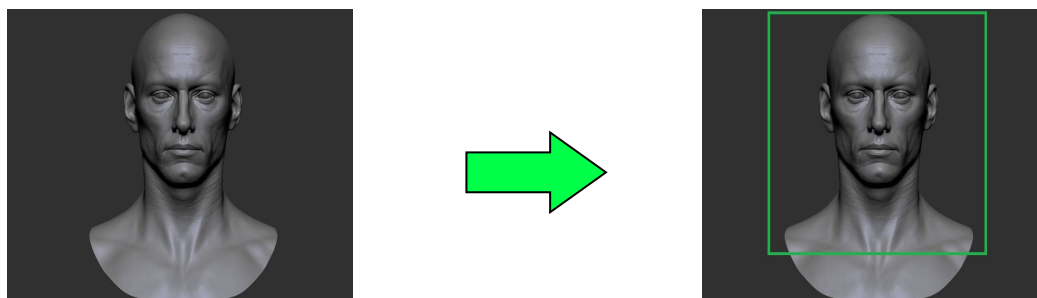
Keywords: *semantic image segmentation, fully convolutional network, conditional random fields.*

CHƯƠNG 1: GIỚI THIỆU

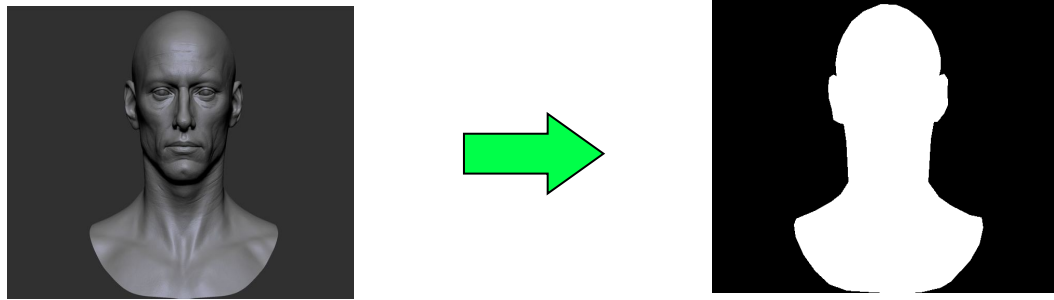
1.1 Tổng quan về tách phong ảnh

Hiện nay, các tập đoàn công nghệ không ngừng đầu tư và ứng dụng các loại công nghệ trong đó có tách phong ảnh vào các sản phẩm công ty của họ đem lại doanh thu cũng như thu hút lượng khách hàng rất lớn, mang lại trải nghiệm mới mẻ cho người dùng. Cụ thể, tập đoàn Bkav thay vì tốn chi phí cho phần cứng phải đầu tư máy ảnh kép, họ tiết kiệm chi phí bằng cách tích hợp khả năng tách phong vào máy chụp ảnh trên điện thoại cho phép người dùng làm mờ cảnh xung quanh người, làm nổi bật đối tượng được chụp. Ứng dụng SNOW [2] ra đời cho phép ghép ảnh, thêm các chi tiết như râu, nón, mắt kiếng,... lên từng bộ phận đã được tách phong trên mặt người trực tiếp khi quay rất độc đáo. Ngoài ra, phải kể đến thành công của tách phong ảnh khi làm nổi bật các bộ phận trong cơ thể con người như tim, gan, phổi,... từ ảnh X-quang hoặc tách phong thiết bị máy móc y học khi hoạt động phẫu thuật trong cơ thể người.

Tách phong ảnh là quá trình cho phép tạo ra những hình ảnh mới từ ảnh gốc với những vùng chi tiết quan trọng được giữ lại, xóa bỏ những vùng nền không có tác dụng phục vụ rất nhiều cho y tế, phần mềm công nghệ máy ảnh, đồ họa kỹ thuật số và tất nhiên chúng đòi hỏi thách thức cao khi thực hiện. Với những lợi ích đem lại cũng cho thấy tính cần thiết của vai trò tách phong ảnh cần được nghiên cứu, đầu tư nhiều hơn. Tách phong ảnh thực sự là ứng dụng của bài toán phân vùng ngữ nghĩa đối tượng người trong ảnh, chúng giải quyết vấn đề xác định vị trí chính xác của đối tượng chiếm trong ảnh, khắc phục nhược điểm của bài toán phát hiện đối tượng khi chỉ có khả năng xác định khung chứa đại khái đối tượng.



Hình 1.1: Phát hiện đối tượng thông thường (Nguồn: www.cgtrader.com)



Hình 1.2: Phân vùng ngữ nghĩa đối tượng (Nguồn: www.cgtrader.com)

Có thể nói, sự xuất hiện họ FCN (FCN-8s, FCN-16s, FCN-32s)[1] với những kiến trúc đặc thù đã ươm mầm cho sự ra đời của những phương pháp nghiên cứu về phân vùng ngữ nghĩa và sự xuất hiện họ DeepLab (DeepLab-v1, DeepLab-v2, DeepLab-v3)[3] ứng dụng lại và kết hợp họ FCN với một bộ các kỹ thuật kết nối đi kèm đã đẩy sự thành công của phân vùng ngữ nghĩa lên cao. Bên cạnh đó sự ra đời của các phiên bản kiến trúc thay đổi SegNet[4] với kỹ thuật unpooling để tăng mẫu ở bộ mã hóa, UNet[5] với kỹ thuật nối kết tất cả bộ mã hóa và bộ giải mã, Mask-RCNN[6] với kỹ thuật mạng đặc trưng kết hợp xếp tầng [7],... đã cho thấy sự thu hút các nhà nghiên cứu đối với lĩnh vực rất tiềm năng này.

Vì sự thiếu hiệu quả của FCN khi thực hiện độc lập và sự phức tạp vốn có của DeepLab cũng như các giải thuật khác với kiến trúc lên đến hàng trăm lớp, tác giả đề xuất kết hợp kiến trúc đơn giản của FCN-8s và tận dụng lại khả năng tối ưu trường điều kiện ngẫu nhiên (CRFs)[8] bên trong DeepLab vẫn đáp ứng được mục tiêu đề ra cũng như thời gian và cấu hình máy chủ huấn luyện để thực hiện luận văn.

1.2 Mục tiêu đề tài

Dựa vào ý tưởng đã nêu, luận văn kết hợp FCN-CRFs tận dụng khả năng phân lớp các điểm ảnh của FCN kèm theo sự thay đổi của quá trình tăng mẫu bằng tích chập chuyển vị (Transposed Convolution) và chuyển tiếp nối kết (Skip Connection) thay cho tích chập giãn nở (Dilated Convolution) và nội suy song tuyến tính (Bi-linear Interpolation)[3]. Cố gắng tối ưu kết quả đầu ra sử dụng kỹ thuật CRFs bằng cách lan truyền thông điệp trên một không gian đã tích chập bằng bộ lọc Gaussian [9] để làm giảm kích thước bài toán.

Quá trình thu thập dữ liệu được tác giả kết hợp sử dụng những tập dữ liệu có sẵn trên mạng với nguồn được công khai và thu thập trực tiếp từ các sinh viên khoa Công nghệ thông tin - truyền thông. Sau đó, tiến hành xử lý các điểm ảnh về 2 loại nhãn: người và nền bằng kỹ thuật xử lý giá trị điểm ảnh cơ bản.

Để thực hiện kiểm tra, tác giả tiến hành trên cùng một tập dữ liệu với mỗi tập ảnh như nhau. Giải thuật huấn luyện gồm có FCN-32s, FCN-16s, FCN-8s được cài đặt bằng bộ thư viện Keras của Python trên hệ điều hành Ubuntu trả về các chỉ số MeanIU đánh giá mức độ chính xác của các điểm ảnh. Giải thuật có chỉ số MeanIU cao nhất trên tập dữ liệu đang xét sẽ được kết hợp với CRFs để đẩy sự chính xác lên cao hơn.

Quá trình phân tích kết quả được thực hiện trên 4 loại giải thuật gồm 3 giải thuật họ FCN và 1 giải thuật tốt nhất kết hợp với CRFs nhằm đánh giá mức độ hơn kém của họ FCN kèm khả năng tối ưu cả về mặt chính xác và thời gian của giải thuật CRFs đã cải tiến.

Thực hiện huấn luyện trên 395 tấm ảnh từ 2 tập dữ liệu: Face/Headseg và Part Labels. Thực hiện kiểm tra mô hình trên tập dữ liệu 130 tấm ảnh chân dung độc lập được thu thập từ 13 sinh viên khoa Công nghệ thông tin - truyền thông. Kết quả kiểm tra cho thấy chỉ số MeanIU trung bình của IoU lần lượt bằng 71.9%, 72.4%, 74.5%, 80.0% tương ứng với 4 loại giải thuật phân vùng ngữ nghĩa FCN-32s, FCN-16s, FCN-8s, FCN-8s-CRFs.

Ứng dụng tách phong là sự vận dụng của phân vùng ngữ nghĩa và xử lý ảnh nhằm tạo vị trí trống trên ảnh nền và xóa phong trên ảnh người sau đó sẽ cộng lại, trả về kết quả ghép phong. Đây cũng là tiền đề để tạo ra những ứng dụng phục vụ rất nhiều cho lĩnh vực thị giác máy tính.

1.3 Bố cục của bài báo cáo luận văn

Bố cục bài báo cáo gồm 4 chương:

Chương 1 - Giới thiệu: Chương mở đầu của luận văn mô tả tổng quan về tách phong ảnh, ứng dụng của chúng trong thực tế, khả năng giải quyết vấn đề của chúng, mục tiêu chi tiết của đề tài và cấu trúc luận văn.

Chương 2 - Cơ sở lý thuyết:

Trình bày lý thuyết tiền xử lý mạng tích chập đầy đủ với các kỹ thuật tích chập thuần cùng các bộ lọc phổ biến, quá trình thăm dò đặc trưng từ ảnh, quá trình tích chập của đầu ra với bộ lọc chuyển vị nhằm thu lại được đầu vào với kết quả tốt nhất có thể, kỹ thuật hỗ trợ chuyển tiếp nối kết bổ sung các giá trị khuyết thiếu.

Tiếp sau là phương pháp tính lại kết quả đầu ra bằng 3 kỹ thuật chính: lan truyền thông điệp, biến đổi tương thích và cập nhật cục bộ của trường điều kiện ngẫu nhiên.

Chương 3 - Kết quả thực hiện:

Trình bày sơ đồ vận hành của hệ thống cùng hoạt động giao tiếp giữa các thành phần bên trong. Nội dung kiến trúc các lớp của thư viện Keras dùng để huấn luyện mô hình FCN, cấu trúc kết quả đầu ra đo lường bằng hàm mất mát và quá trình tích chập bằng bộ lọc Gaussian nhằm xấp xỉ không gian tính toán.

Mô tả chi tiết và so sánh giữa các tập dữ liệu, các độ đo cùng phương pháp để tính toán, các thông số huấn luyện, các giá trị kết quả thu được trên cả tập huấn luyện và kiểm tra.

Kịch bản chi tiết giải quyết vấn đề sai sót của giải thuật cùng với những hình ảnh kết quả sau khi đã tách phong.

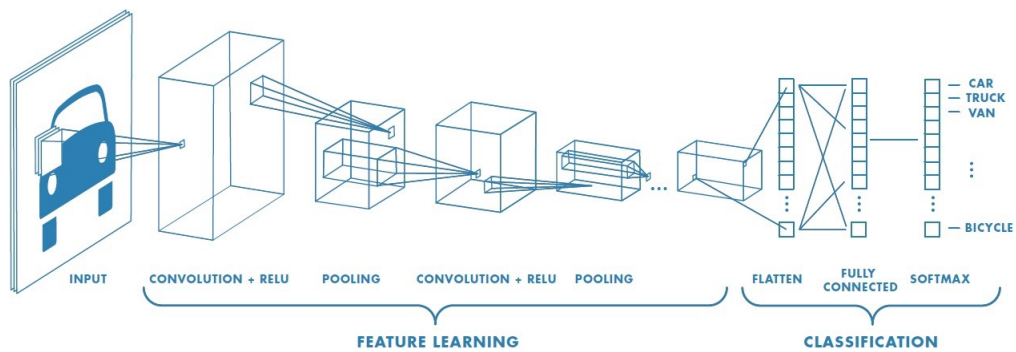
Chương 4 - Kết luận và hướng phát triển: phần cuối luận văn kết luận những ưu và nhược điểm của toàn bộ hệ thống nói chung, tập dữ liệu, giải thuật nói riêng. Hướng phát triển nhằm giải quyết những nhược điểm và ứng dụng những ưu điểm vốn có của ý tưởng giải thuật vào những hệ thống chuyên dùng phục vụ người dùng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Mạng tích chập đầy đủ (Fully Convolutional Network):

Ứng dụng phân vùng ngữ nghĩa được xây dựng thao tác trực tiếp với ảnh và mạng nơ-ron tích chập nguyên thủy CNN (Convolutional Neural Network) là giải pháp rất thành công ở thời điểm hiện tại khi thao tác với các loại ảnh. Chính vì vậy, CNN thường được sử dụng làm điểm tựa để nghiên cứu về vấn đề trên.

Theo lý thuyết, mạng nơ-ron tích chập nguyên thủy [10] là sự tổ hợp hợp lý các lớp tích chập (Convolution layer), lớp phi tuyến (Nonlinear layer) và lớp thăm dò (Pooling layer) với mỗi lớp sẽ đảm nhiệm những vai trò tính toán chuyên biệt nhằm rút lấy đặc trưng (Feature Learning), sau đó sẽ đưa qua bộ phân lớp (Classification) để xuất ra một véc-tơ chứa xác suất hay khả năng đối tượng đầu vào thuộc từng lớp đó.



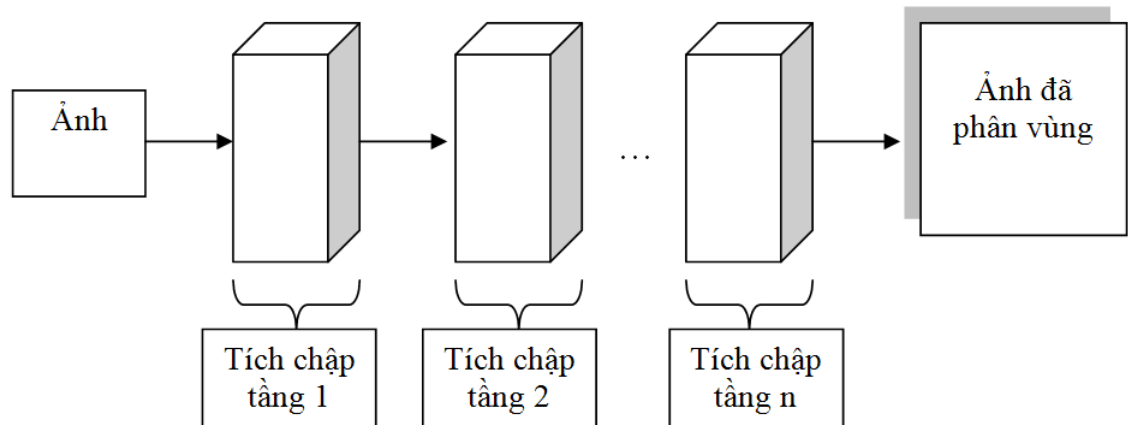
Hình 2.1: Mạng nơ-ron tích chập truyền thống [Nguồn: towardsdatascience.com]

Đối với bài toán phân vùng ngữ nghĩa được đặt ra ở trên thì theo phương pháp thông thường sẽ phát sinh 2 vấn đề.

Vấn đề đầu tiên là mô hình CNN nguyên thủy chứa đựng những lớp làm giảm kích thước chiều cao và rộng đồng thời gia tăng chiều sâu của tấm ảnh, cụ thể là lớp thăm dò nên việc dữ liệu ảnh bị đứt đoạn, rời rạc là điều chắc chắn xảy ra mà nếu hình ảnh bị biến đổi như vậy sẽ không còn giữ được sự nguyên vẹn cũng như ý nghĩa của chúng so với tấm ảnh ban đầu.

Vấn đề thứ hai là sẽ lấy nội dung gì để làm nhãn vì CNN là một mô hình học có giám sát mà nhãn để so khớp không có thì không thể nào thực hiện những phép toán như lan truyền ngược [11] để cập nhật lại trọng số dẫn đến quá trình huấn luyện và thay đổi trọng số của chúng sẽ thất bại trầm trọng.

Vấn đề lại có hướng giải quyết là tiến hành bỏ đi những lớp thăm dò và bộ phân lớp để giảm thiểu sự thiệt hại cũng như không cần lo lắng vấn đề về nhãn, chỉ giữ lại những lớp tích chập nhưng lại phát sinh vấn đề nữa rằng, sẽ phải tốn khả năng xử lý phức tạp của máy cũng như bài toán sẽ càng khó giải quyết hơn.



Hình 2.2: Các lớp tích chập thuần nối tiếp

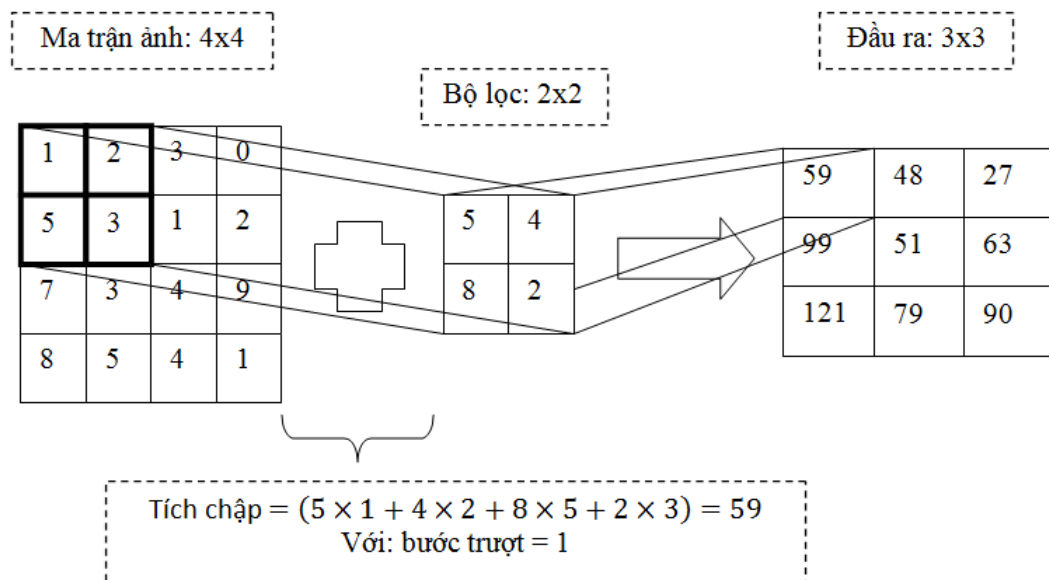
Bài toán xuất hiện thu hút những bộ não lớn trong lĩnh vực thị giác máy tính và cho ra rất nhiều những bài báo nghiên cứu khoa học có giá trị, tiêu biểu trong số đó có "Fully Convolutional Networks for Semantic Segmentation"[1] của Jonathan Long cùng các cộng sự của ông. Thay vì tiếp cận theo cách cũ, Jonathan Long đã đề xuất một mạng nơ-ron khác có tên là mạng tích chập đầy đủ Fully Convolutional Network - FCN, bằng cách tiếp cận rằng dữ liệu đầu vào vẫn chấp nhận bị biến đổi hay giảm mẫu nhưng FCN sẽ làm thêm một công việc đó là thay thế bộ phân lớp nguyên thủy thành bộ tăng mẫu để khôi phục dữ liệu lại ban đầu hay còn được hiểu là đem những mảng rời rạc trong cấu trúc xếp tầng của dữ liệu khi bị tích chập ở phía dưới để đưa lên phía trên và được thực hiện một cách có ý nghĩa để tạo những hình ảnh có ý nghĩa nhất. Vì bài báo cáo luận văn bị giới hạn về năng lực tính toán và thời gian xử lý của máy chủ huấn luyện nên để có thể vẫn đáp ứng được mục tiêu đề ra và sử dụng kỹ thuật cần thiết, tác giả ưu tiên sử dụng mạng FCN.

2.1.1 Bộ giảm mẫu - mã hóa (Downsampling - Encoder):

Quá trình tính toán nhằm mục tiêu lấy ra thông tin ngữ nghĩa, ngữ cảnh của các đối tượng trong ảnh, mục tiêu của phương pháp này là xác định đối tượng gì trong tấm ảnh. Kỹ thuật giảm mẫu (downsampling) chính là bước đầu của những mạng nơ-ron tích chập nguyên thủy [10] và tất nhiên chỉ kế thừa lại những thành công mà mạng nơ-ron tích chập CNN đem lại.

2.1.1.1 Lớp tích chập:

Tích chập được hiểu là kỹ thuật chính để rút lấy đặc trưng của tấm ảnh, đưa những chi tiết nổi bật lên trên và ít nổi bật xuống dưới tạo thành cấu trúc xếp tầng của tấm ảnh đầu vào, gia tăng chiều sâu. Tên của nó biểu thị cho cách mà nó tính toán đối với dữ liệu ảnh đầu vào như sau:



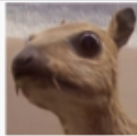

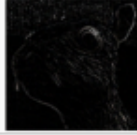


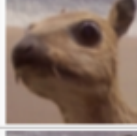
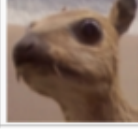
Hình 2.3: Phép toán tích chập

2.1.1.2 Bộ lọc và những thông số có liên quan:

Bộ lọc (filter) là sự đột phá trong việc rút trích đặc trưng của những tấm ảnh, biểu thị tính chất kết hợp cục bộ cho các cấp độ biểu diễn thông tin từ thấp đến cao và trừu tượng hơn.

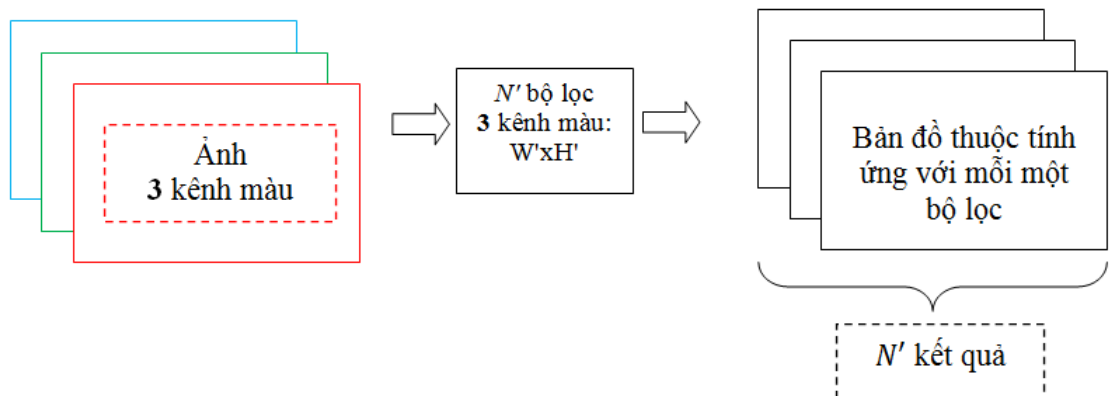
Thông thường, bộ lọc sẽ có kích thước 3x3 hoặc 5x5 nhưng có thể trong những mô hình đòi hỏi yêu cầu khác nhau kích thước cũng sẽ khác nhau cũng như số lượng bộ lọc sẽ gia tăng hoặc giảm xuống sao cho phù hợp.

Bằng sự tổ hợp các giá trị ngẫu nhiên trên bộ lọc sẽ cho ta các con số để tính toán nhằm mục đích rút lấy những đặc trưng của ảnh, sự tổ hợp của các con số bất kỳ nêu trên theo một thứ tự nào đó có thể rút ra được những nội dung khác nhau.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Hình 2.4: Những bộ lọc phổ biến [Nguồn: medium.com]

Mỗi giá trị vị trí trên ma trận của ảnh màu là sự tổ hợp của 3 màu cơ bản (đỏ-lục-lam). Số lượng giá trị đầu ra hay bản đồ thuộc tính (feature map) được xếp tầng dưới dạng 2D ở mỗi tầng. Mỗi vị trí trên bộ lọc là một trọng số, chúng được chia sẻ qua các lớp và k bộ lọc sẽ bằng k bản đồ thuộc tính đầu ra. Mỗi bộ lọc với mỗi kênh màu(RGB) sẽ tính tích chập từng kênh với ảnh và cộng lại kết quả các kênh màu của mỗi bộ lọc để cho ra giá trị đúng với 1 dữ liệu đầu ra đại diện cho ảnh 3 kênh màu đầu vào.



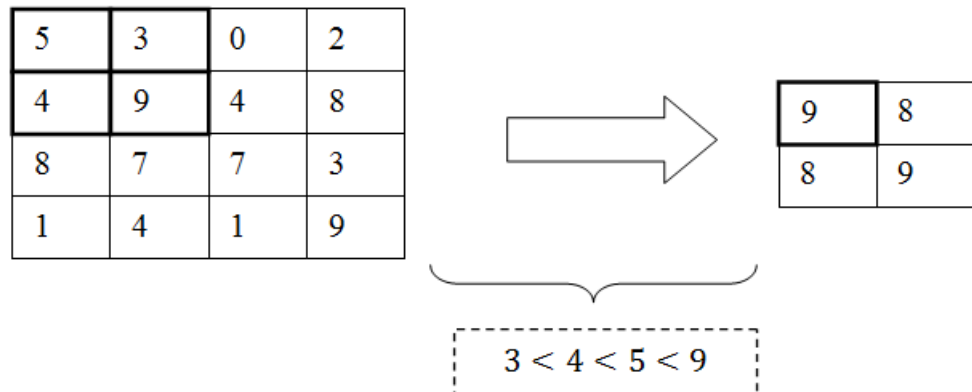
Hình 2.5: Bộ lọc và mối quan hệ với ảnh

2.1.1.3 Lớp thăm dò (Pooling layer):

Lớp thăm dò có chức năng đơn giản hóa thông tin đầu ra, giảm nhiễu, giữ lại những thông tin quan trọng, làm cho kích thước dữ liệu co lại nhưng vẫn đủ tốt để trích xuất dữ liệu và được sử dụng sau mỗi tầng tích chập.

Các phương thức phổ biến được sử dụng gồm có: thăm dò tối đa (MaxPooling), thăm dò trung bình (AveragePooling) và thăm dò tối thiểu (MinPooling) nhưng trong thực tế, chỉ có thăm dò tối đa được tin dùng nhiều nhất với câu hỏi "đặc trưng nào là đặc trưng nhất".

Kỹ thuật chính của thăm dò tối đa (MP) là làm giảm số lượng nơ-ron đi một nửa. Cụ thể, lớp thăm dò tối đa sử dụng một cửa sổ trượt có kích thước 2×2 (hoặc 4×4 với ảnh đầu vào có kích thước lớn) để quét qua ảnh giống như lớp tích chập, so sánh và giữ lại giá trị cao nhất trong cửa sổ trượt đó. Chính vì điều này đã làm cho việc phân vùng ngữ nghĩa ảnh bị mất đi những thông tin đáng lẽ phải được giữ lại nhưng nó cũng giúp đỡ không nhỏ trong việc giảm kích thước bài toán.



Hình 2.6: Kết quả ví dụ của lớp thăm dò

Lớp thăm dò biểu diễn tính chất vô cùng quan trọng của mạng tích chập đầy đủ (FCN), tính bất biến, dù cho có chuyển dịch, xoay chuyển, co giãn như thế nào, luôn tìm được vị trí của đối tượng ở bất kỳ đâu trong tấm ảnh.

2.1.2 Bộ tăng mẫu - giải mã (Upsampling - Decoder):

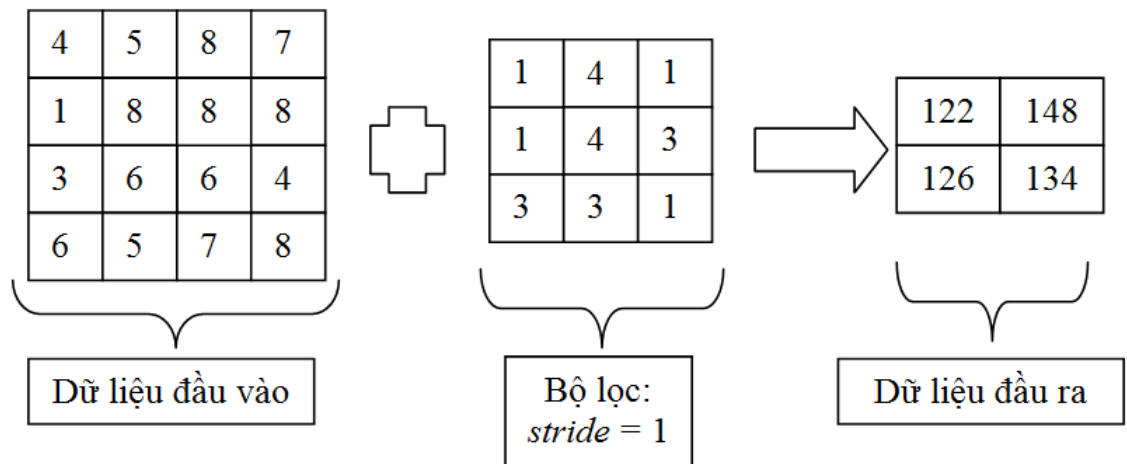
Khác với những mạng nơ-ron tích chập nguyên thủy (CNN), mạng tích chập đầy đủ (FCN) không sử dụng bộ phân loại hay còn được hiểu là tầng kết nối đầy đủ (Fully connected layer) vì bộ phân loại này sẽ chỉ phân loại được khả năng lớn nhất của một lớp nào đó cho toàn bộ ảnh đầu vào nhưng thứ chúng ta cần là ý nghĩa của điểm ảnh hay nói một cách dễ hiểu, chúng ta cần phân lớp và gán nhãn cho từng điểm ảnh. Tuy nhiên, để thực hiện dự đoán những điểm ảnh của đầu vào thì cần phải khôi phục nội dung đã bị các tầng tích chập phía trước làm thay đổi và sắp xếp chúng lại song song với quá trình khôi phục đã nêu để tạo ra những phân vùng ảnh ngữ nghĩa của đối tượng.

Để thực hiện quá trình đó thông thường người ta sẽ sử dụng cách ước lượng giá trị của một điểm ảnh ở giữa hai điểm ảnh đã biết, hay còn gọi với cái tên khoa học là nội suy ảnh [12]. Các phép nội suy thông thường được sử dụng là nội suy các điểm ảnh gần nhất, nội suy song tuyến tính, nội suy song khối nhưng các phép nội suy này lại được xem là những kỹ thuật thủ công và kiến trúc mạng sẽ không tận dụng được lợi thế của các phép tính toán này vì vốn dĩ chúng không có trọng số để học.

2.1.2.1 Tích chập chuyển vị (Transposed Convolution):

Tích chập chuyển vị đôi lúc còn có tên gọi là tích chập ngược (Deconvolution) [1] là hoạt động tìm lại giá trị của dữ liệu đầu vào bằng các kỹ thuật đặc biệt từ kết quả đầu ra và bộ lọc của chúng.

Tích chập chuyển vị là một thủ thuật đi ngược lại với kỹ thuật tích chập thông thường, hay nói cách khác là cố gắng khôi phục lại hình trạng ban đầu của dữ liệu đầu vào bằng bộ lọc đã được thay đổi bởi những ô số đặc biệt (fancy padding) và dữ liệu đầu ra. Với ví dụ tích chập thuận như sau:



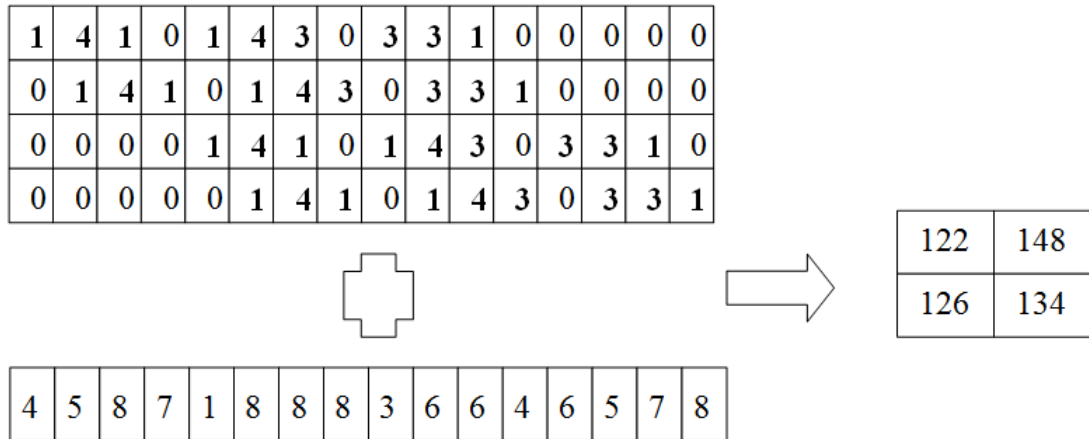
Hình 2.7: Kết quả ví dụ phép tích chập thuận

Tiếp theo, ta thay đổi một cách ngẫu nhiên ma trận bộ lọc từ kích thước 3x3 thành 4x16 với 4 và 16 lần lượt là số ô dữ liệu của ma trận dữ liệu đầu ra và đầu vào như sau:

1	4	1	0	1	4	3	0	3	3	1	0	0	0	0	0
0	1	4	1	0	1	4	3	0	3	3	1	0	0	0	0
0	0	0	0	1	4	1	0	1	4	3	0	3	3	1	0
0	0	0	0	0	1	4	1	0	1	4	3	0	3	3	1

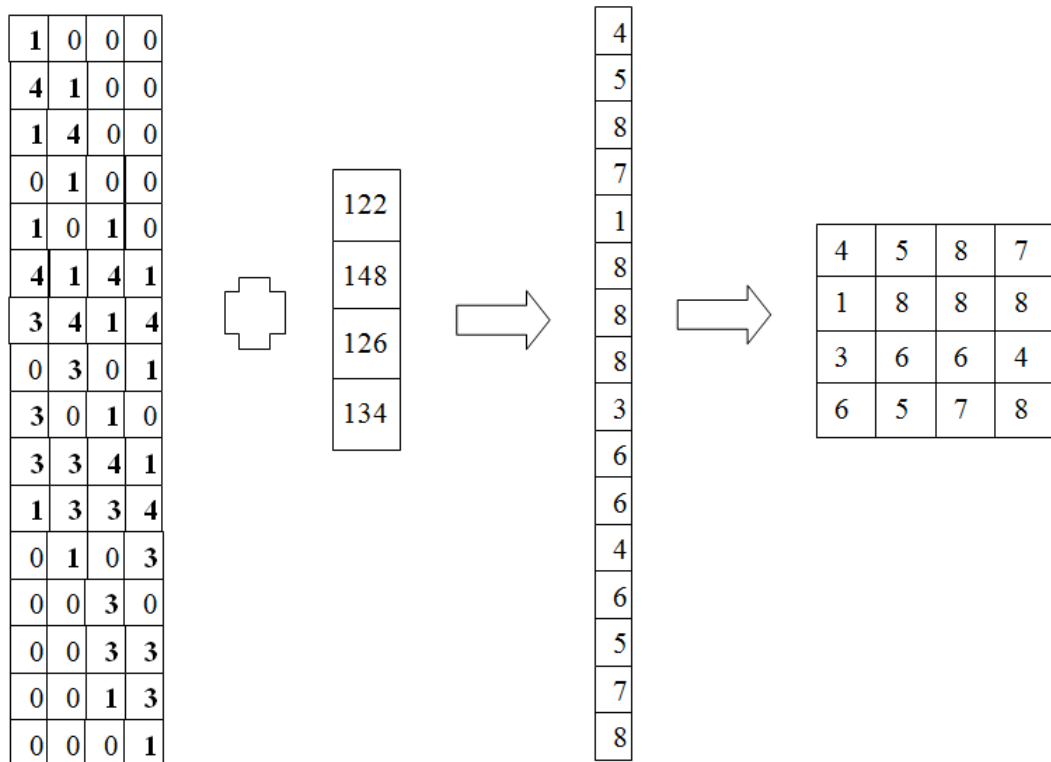
Hình 2.8: Bộ lọc sau khi thay đổi

Để kiểm tra ma trận này có thực sự đại diện cho bộ lọc cũ khi thực hiện tính toán vẫn cho ra kết quả giống với bộ lọc cũ hay không, ta cũng thực hiện tích chập lần nữa với dữ liệu đầu vào đã được sắp xếp lại dưới dạng 1×16 :



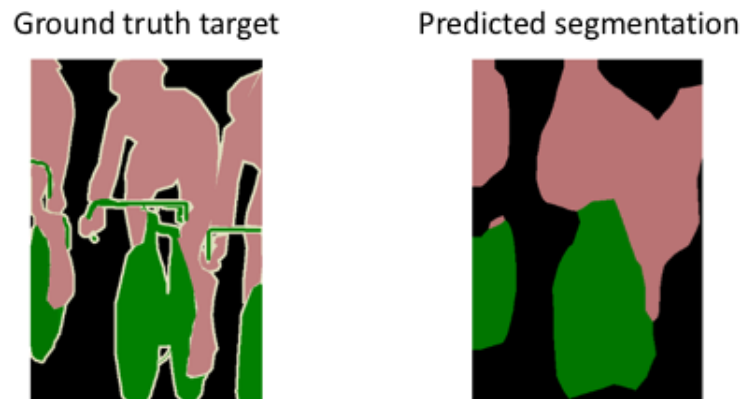
Hình 2.9: Tích chập kiểm tra bộ lọc sau thay đổi

Đến đây ta hoàn toàn kết luận được ma trận bộ lọc này có thể đại diện được cho ma trận bộ lọc cũ và sử dụng ma trận chuyển vị của bộ lọc này tiến hành tích chập thử với kết quả đầu ra đã được đưa về dạng véc-tơ có kích thước 1×4 để xem có thu được ma trận đầu vào hay không.



Hình 2.10: Tích chập với bộ lọc chuyển vị

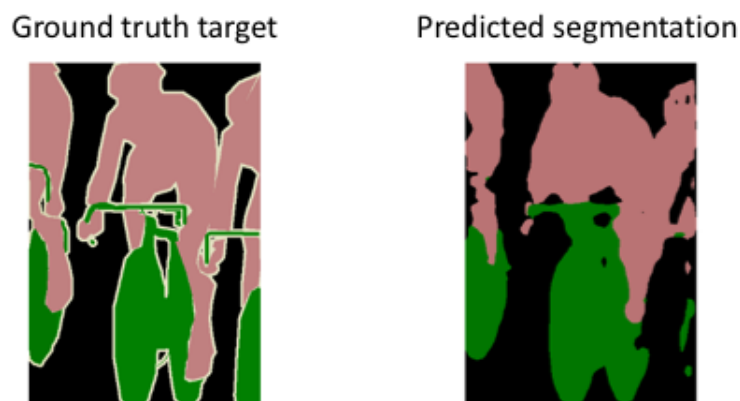
Xét thấy kết quả tính toán quả thật trả về cho ta kết quả giống y đầu vào. Kỹ thuật vừa được sử dụng có tên là tích chập chuyển vị, tức là lấy chuyển vị của ma trận bộ lọc đã được thêm các ô số đặc biệt (fancy padding) rồi tích chập với kết quả sẽ trả về được những gì đã đưa vào. Có thể nói rằng, việc khó khăn ở đây là làm sao tìm được cách sắp xếp các ô số đặc biệt để tạo nên một ma trận bộ lọc đủ khả năng đại diện cho bộ lọc ban đầu mà giảm thiểu sự sai sót xuống thấp nhất và sẽ cho ra kết quả thông thường như sau:



Hình 2.11: Kết quả mô phỏng hình ảnh thu được [Nguồn: [1]]

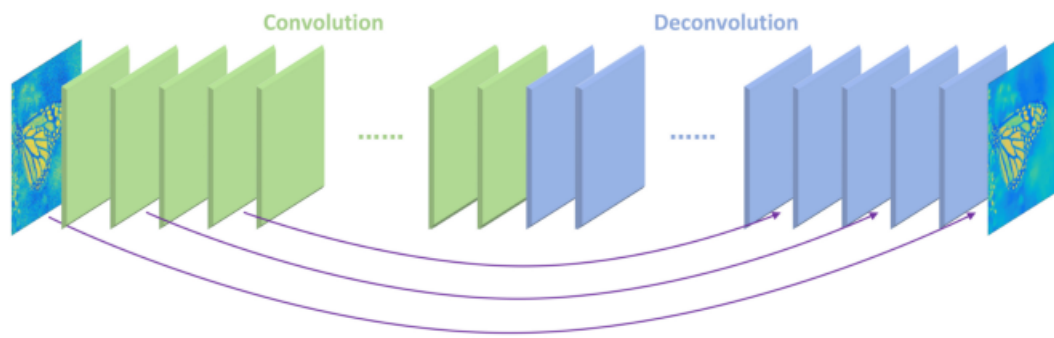
2.1.2.2 Chuyển tiếp kết nối (Skip Connection):

Bằng phương pháp tích chập chuyển vị, ta cũng có thể quan sát được, kết quả chỉ dừng lại ở mức mô tả hình trạng khái quát và chưa đủ để hình dung đối tượng vì dữ liệu đã bị mất đi do quá trình tích chập ban đầu nên việc kết hợp thêm phương pháp chuyển tiếp kết nối là điều cần thiết để giảm thiểu sự mất mát vốn có. Kết quả như sau:



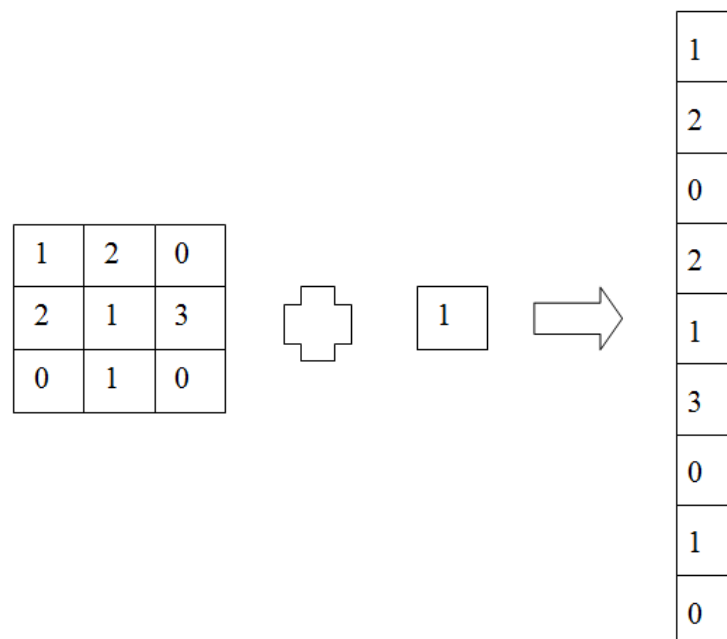
Hình 2.12: Kết quả hình ảnh sau khi chuyển tiếp nối kết [Nguồn: [1]]

Kỹ thuật chuyển tiếp kết nối (Skip Connection) [1] là việc lưu trữ lại kết quả sau khi đã đi ra khỏi lớp thăm dò của các tầng ở phía trước và thực hiện phép cộng độ phân giải được phân bố trong ảnh với các kết quả sau khi đã thực hiện tăng mẫu ở các tầng phía sau nhằm bổ sung những thông tin còn thiếu sót trong quá trình tính toán.



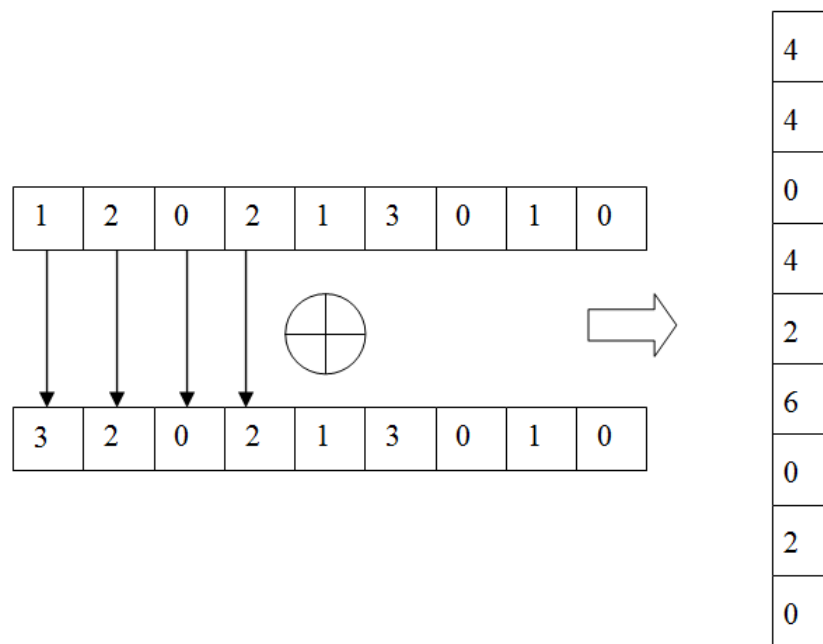
Hình 2.13: Mô phỏng quá trình chuyển tiếp nối kết [Nguồn: i.stack.imgur.com]

Nếu xét đúng bản chất thì trong mạng nơ-ron tích chập thực sự không tồn tại khái niệm kết nối đầy đủ (fully connected) mà chỉ tồn tại lớp tích chập với kích thước bộ lọc bằng 1×1 di chuyển từng ô trong ma trận ảnh và thực hiện tính toán, khi đó phần đầu tiên của lớp kết nối đầy đủ thực sự tính toán giống y như cách lớp tích chập (bộ lọc 1×1) làm việc. Chính vì vậy trong mô hình FCN đã bỏ đi tầng kết nối đầy đủ và thay thế chúng bằng bộ lọc 1×1 .



Hình 2.14: Tích chập với bộ lọc 1×1

Trước khi thực hiện tích chập chuyển vị phía sau, dữ liệu đã biến thành cấu trúc xếp tầng với mỗi tầng là một véc-tơ có độ dài bằng nhau và bằng số lượng vùng đối tượng có trong ảnh do quá trình tính toán của bộ lọc 1×1 . Kỹ thuật cộng độ phân giải được phân bố trong ảnh có thể được miêu tả như sau:

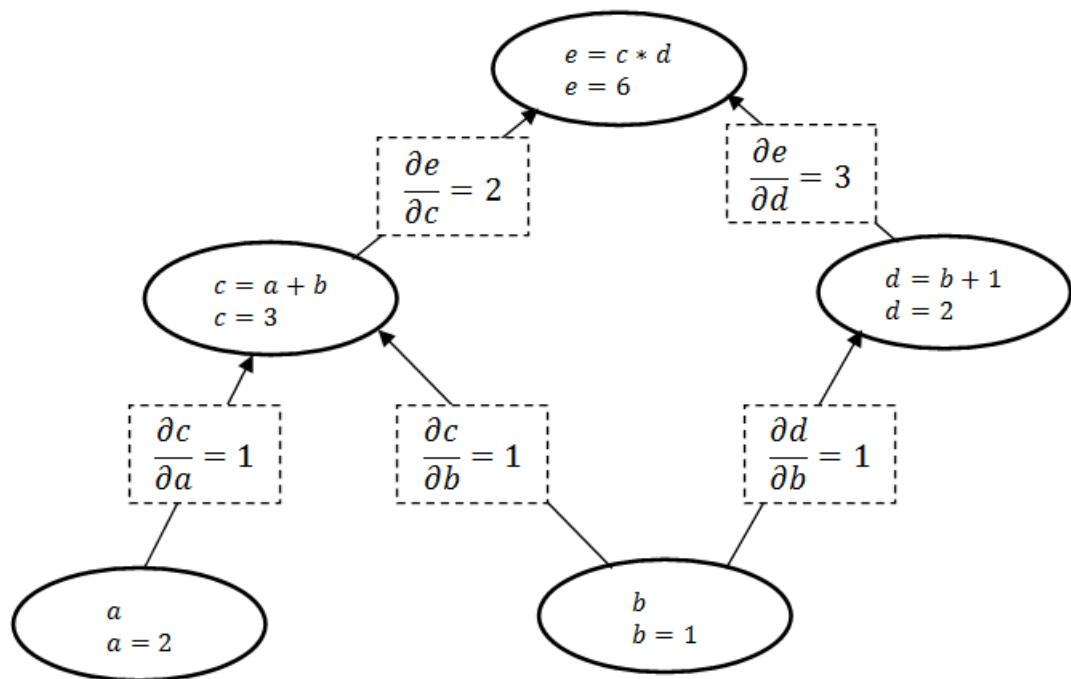


Hình 2.15: Mô phỏng quá trình kết hợp toàn bộ

2.1.3 Gradient và cập nhật trọng số:

2.1.3.1 Đạo hàm và lan truyền ngược:

Trước tiên, để không mơ hồ về lý thuyết áp dụng, ta cần hiểu các nội dung cơ bản như đạo hàm là gì và làm gì sau khi tính được giá trị giúp hàm lỗi đạt nhỏ nhất. Đạo hàm được hiểu là sự ảnh hưởng của điểm này tới điểm khác, lấy ví dụ như ta thay đổi một giá trị của biến này thì sẽ ảnh hưởng như thế nào tới biến kia hoặc những biến khác. Cụ thể, nếu tượng hình thì ta hoàn toàn có thể hình dung chúng là những giá trị nằm trên đoạn nối các điểm biểu thị sự ảnh hưởng từ điểm này tới điểm khác:



Hình 2.16: Sơ đồ mối quan hệ giữa các biểu thức

Để tính được sự ảnh hưởng của e nếu thay đổi giá trị tại b , ta áp dụng cách tính lấy tích của mỗi đoạn nối từ b đến e rồi tổng chúng lại với nhau:

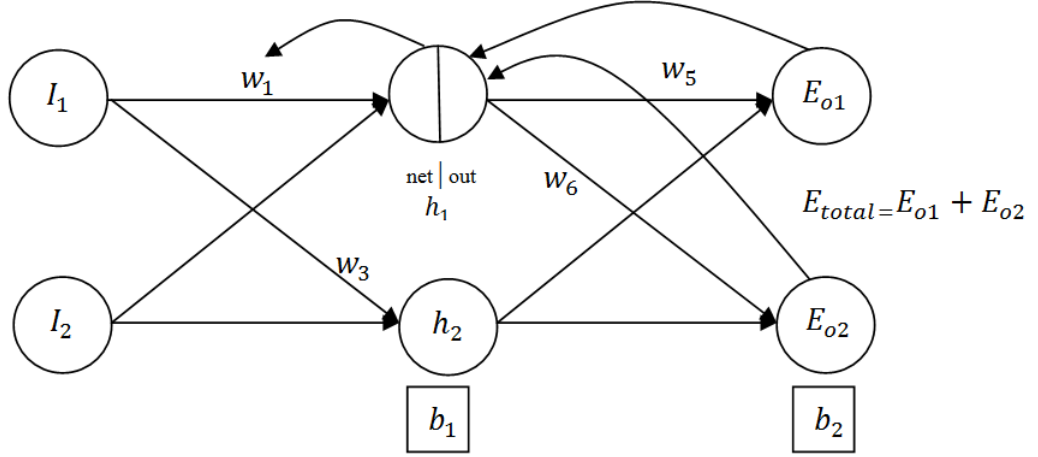
$$\frac{\partial e}{\partial b} = \left(\frac{\partial c}{\partial b} \times \frac{\partial e}{\partial c} \right) + \left(\frac{\partial d}{\partial b} \times \frac{\partial e}{\partial d} \right) = (1 \times 2) + (2 \times 3) = 6 \quad (2.1)$$

Nếu thay đổi giá trị tại b bằng 1 đơn vị thì sẽ thay đổi giá trị tại e đến 6 đơn vị. Trong những trường hợp khác, nếu các cạnh vào hoặc ra cùng 1 điểm thì chỉ việc tính tổng của các cạnh vào rồi nhân với tổng các cạnh ra vẫn cho ta kết quả đạo hàm của các điểm cần tính thông qua điểm chung đó.

Quá trình lan truyền ngược hay đạo hàm ngược là sự tập hợp tính toán của đạo hàm với ý nghĩa như trên nhưng nếu giống hoàn toàn như trên thì sẽ gọi là đạo hàm tiến chứ không phải lan truyền ngược.

Quá trình lan truyền ngược được hiểu trong bài toán trên là nếu muốn tính đạo hàm tại e cần phải tính tất cả các đạo hàm "chiều mũi tên" tới e . Như vậy, việc tính đạo hàm từ điểm trên ngọn e theo b xuống, cũng vô tình tính tất cả các đạo hàm của e theo những biến khác. Chính vì thế, đạo hàm ngược hay lan truyền ngược tăng tốc tối đa tính toán, tiết kiệm thời gian và vô cùng thích hợp với môi trường của các loại mạng nơ-ron.

Kết quả được đoán khác với đầu ra mong muốn được tính toán ở lớp cuối cùng và chỉ có lớp cuối cùng biết, các lớp phía trên không hề biết nên việc lan truyền độ sai khác giữa giá trị đoán được và nhãn mong muốn cũng như cập nhật trọng số của các lớp phía trên là điều hết sức cần thiết.



Hình 2.17: Mô phỏng quá trình lan truyền ngược

Dựa vào hình 2.17, nếu muốn cập nhật w_1 ảnh hưởng như thế nào tới hàm E_{total} thì ta có cách tính sau:

$$w_{1(\text{mới})} = w_1 - \eta \times \frac{\partial E_{total}}{\partial w_1} \quad (2.2)$$

Với η là tốc độ học,

$net|out$ lần lượt là hàm mạng | giá trị đầu ra,

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1} \quad (2.3)$$

Trong đó:

$$\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}}, \quad (2.4)$$

$$\frac{\partial out_{h_1}}{\partial net_{h_1}} = out_{h_1} \times (1 - out_{h_1}), \quad (2.5)$$

$$\frac{\partial net_{h_1}}{\partial w_1} = i_1, \text{ do } net_{h_1} = w_1 \times i_1 + w_3 \times i_2 + b_1 \times 1 \quad (2.6)$$

Chi tiết công thức 2.4:

$$\frac{\partial E_{o_1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} + \frac{\partial net_{o_1}}{\partial out_{h_1}} \quad (2.7)$$

Chi tiết công thức 2.7:

$$\frac{\partial E_{o_1}}{\partial net_{o_1}} = \frac{\partial E_{o_1}}{\partial out_{o_1}} + \frac{\partial out_{o_1}}{\partial net_{o_1}} \quad (2.8)$$

$$E_{o_1} = \frac{1}{2}(target_{o_1} - out_{o_1})^2 \text{ và } out_{o_1} = \frac{1}{1 + e^{-net_{o_1}}} \quad (2.9)$$

$$\frac{\partial net_{o_1}}{\partial out_{h_1}} = w_5, \text{ do } net_{o_1} = w_5 \times out_{h_1} + w_6 \times out_{h_2} + b_2 \times 1 \quad (2.10)$$

Hàm tổng lỗi còn được hiểu:

$$E_{total} = \frac{1}{2}(target_{h_1} - out_{h_1})^2 + \frac{1}{2}(target_{h_2} - out_{h_2})^2 \quad (2.11)$$

Quá trình cập nhật trọng số được thực hiện dây chuyền (chain-rule). Độ sai khác sẽ được lan truyền đi để thông báo cho các trọng số phía trước có sự điều chỉnh cho phù hợp sau mỗi lần huấn luyện để lần sau dự đoán sẽ cho ra kết quả tốt hơn.

2.1.3.2 Giảm Gradient ngẫu nhiên (Stochastic Gradient Descent):

Tiếp tục quay trở lại câu hỏi đã được đặt ra rằng làm thế nào để tính được những giá trị biến giúp cho hàm lỗi đạt được giá trị nhỏ nhất. Có một sự thật rằng, hàm lỗi hoàn toàn có khả năng để đạo hàm nhưng không dễ dàng để tính được giá trị của chúng khi cho phương trình đạo hàm đó bằng 0.

Tồn tại một phương pháp có thể khắc phục vấn đề trên đó là giảm gradient (Gradient Descent) [11] và ta cần biết gradient là gì trước khi muốn giảm nó. Gradient được hiểu là tập hợp các giá trị đạo hàm của một phương trình nào đó theo các biến của nó, vậy giảm gradient tức là tìm các cách để thay đổi các giá trị thành phần bên trong gradient sao cho phương trình đạo hàm gốc tiến tới bằng 0 thì dừng. Kỹ thuật giảm gradient ngẫu nhiên SGD (Stochastic Gradient Descent) được sử dụng để tối ưu hàm lỗi và có công thức như sau [13]:

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta) \quad (2.12)$$

Với:

v_t là vận tốc tại thời điểm trước khi chuyển động,

η là tốc độ học,

γ là momentum,

$\nabla_{\theta} J(\theta)$ là độ dốc của thời điểm trước khi chuyển động.

2.2 Hậu xử lý (Post-processing):

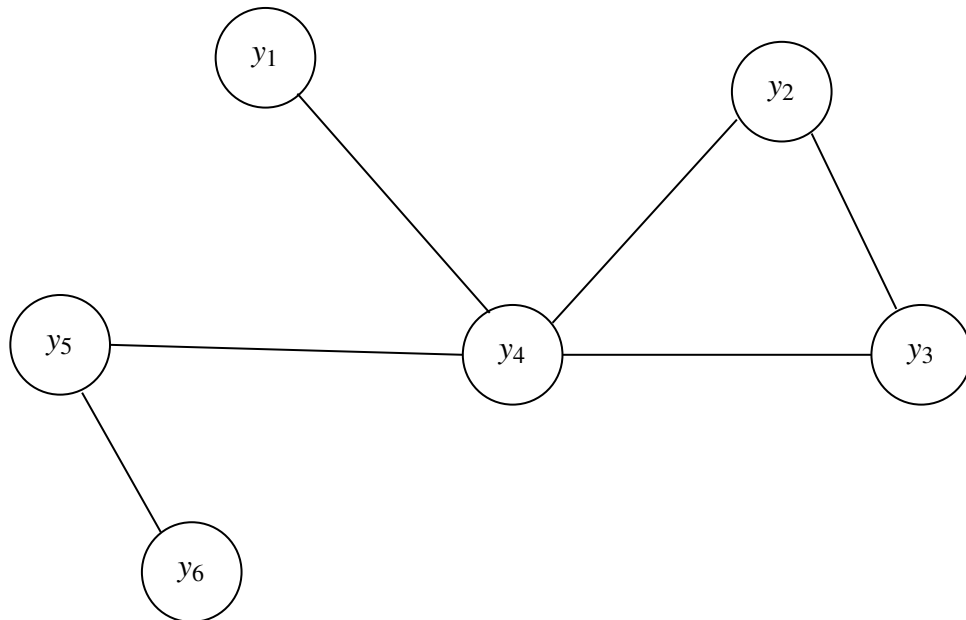
2.2.1 Trường điều kiện ngẫu nhiên (CRFs):

CRFs là một trường ngẫu nhiên và có điều kiện. Lấy ví dụ sau, ta có một đồ thị vô hướng G , có tập đỉnh Y tức các điểm ảnh kết quả của quá trình dự đoán gồm y_1, y_2, \dots, y_v và tập cung E nối các đỉnh. Nếu như $P(y_n|Y_m) = P(y_n|Y_l)$ thì Y là trường ngẫu nhiên, với:

y_n là một đỉnh ngẫu nhiên bất kỳ

Y_m là tập đỉnh còn lại (all others) của Y sau khi đã bỏ đi y_n

Y_l là tập đỉnh kề (neighbors) của y_n



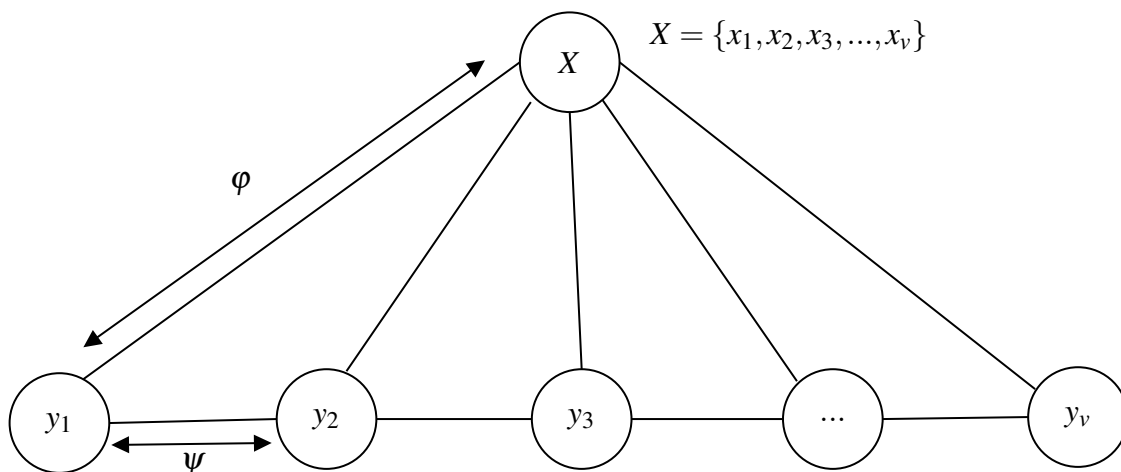
Hình 2.18: Đồ thị trường ngẫu nhiên

Trong ví dụ trên, ta xét:

$$\begin{aligned} y_n &= y_5 \\ Y_m &= y_1, y_2, y_3, y_4, y_6 \\ Y_l &= y_4, y_6 \end{aligned} \quad (2.13)$$

Nếu như: $P(y_n|Y_m) = P(y_n|Y_l)$ thì đồ thị trên là một trường ngẫu nhiên.

Giả sử $P(y_n)$ không chỉ đơn thuần là xác suất tức khả năng xảy ra độc lập của một trường hợp y_n mà nó còn phụ thuộc vào một biến nào đó trong tập các điểm ảnh gốc $X = x_1, x_2, \dots, x_v$, nói một cách dễ hiểu $P(y_n)$ trở thành $P(y_n|X)$ và $P(y_n|X, Y_m) = P(y_n|X, Y_l)$ thì lúc đó đồ thị vô hướng có các đỉnh X và Y sẽ trở thành trường ngẫu nhiên có điều kiện.



Hình 2.19: Trường điều kiện ngẫu nhiên

y_1, \dots, y_v là tập hợp các điểm ảnh đã được dự đoán nhãn bằng mô hình.

x_1, \dots, x_v là tập hợp các điểm ảnh từ ảnh gốc.

φ là hàm tiềm năng đơn phương (Unary potential), dùng để mã hóa thông tin điểm ảnh cục bộ đã được đưa vào và cho ra khả năng điểm ảnh thuộc về một lớp nào đó.

ψ là hàm tiềm năng cặp đôi (Pairwise potential), dùng để mã hóa thông tin của những điểm ảnh kề cận điểm ảnh đang xét bằng cách so sánh cường độ, màu sắc, kết cấu, đường biên,...

2.2.2 Hàm năng lượng (Energy function):

Hàm năng lượng là một hàm mà các thành phần cấu trúc (configuration) của kết quả và đầu vào đều được cho biết trước, bằng cách cố gắng tối ưu hàm này từ các thông số cho trước để tìm ra thành phần cấu trúc thỏa phân phối xác suất nào đó là có thể kết luận các loại điểm ảnh đoán được thuộc lớp nào cần phải giữ lại hoặc loại bỏ nhằm thu được hình ảnh tốt nhất có thể. Để dễ hình dung hơn ta xét công thức phía dưới.

Ta có $X = x_1, x_2, \dots, x_n$ là tập các điểm ảnh đầu vào, $Y = y_1, y_2, \dots, y_n$ là tập các nhãn đoán được, S là tập các đỉnh của Y đôi một kề nhau (clique), kết quả tối ưu được tính:

$$y^* = \operatorname{argmax}_{y \in Y} (P(Y|X)) \quad (2.14)$$

Với:

$$P(Y|X) = \frac{1}{Z(X)} \exp(-E(Y|X)) \text{ là phân phối xác suất cần tìm}$$

$$Z = \sum_{x \in X} \exp(-E(Y = y|X)) \text{ là hàm phân vùng (partition function)}$$

Trong CRFs, hàm năng lượng được mô tả theo công thức:

$$E(y|X) = \sum_i \varphi(y_i|X) + \sum_{ij} \psi(y_i, y_j) \quad (2.15)$$

Với:

$\varphi(y_i|X) = -\log(P(y_i))$, $P(y_i)$ là khả năng điểm ảnh thuộc về một lớp nào đó

$$\psi(y_i, y_j) = \mu(y_i, y_j) \sum_{m=1}^K w_m * k^m(f_i, f_j)$$

Nếu $y_i \neq y_j$ thì $\mu(y_i, y_j) = 1$, ngược lại $\mu(y_i, y_j) = 0$

k^m là hàm nhân Gaussian phụ thuộc vào véc-tơ đặc trưng f

w_m là trọng số liên kết

Công thức tính hàm nhân được mô phỏng lại giống như bộ lọc hai chiều (bilateral filter) chuyên dùng để thay thế các điểm ảnh bằng giá trị trọng số trung bình của các điểm kề cận. Hàm nhân được tính theo công thức:

$$w_1 \cdot kernel_1 + w_2 \cdot kernel_2 =$$

$$= w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \quad (2.16)$$

Với:

p là vị trí của điểm ảnh

I là cường độ của điểm ảnh

$\sigma_\alpha, \sigma_\beta, \sigma_\gamma$ là các siêu thông số co giãn (scale) của hàm nhân

$kernel_1$ là hàm nhân hình dáng (appearance kernel) nhằm tính toán những điểm ảnh gần nhau và độ tương đồng về màu sắc

$kernel_2$ là hàm nhân mịn (smoothness kernel) nhằm xóa bỏ những vùng bị cô lập

2.2.3 Suy luận (Inference):

Xét thấy công việc tối ưu hàm P là điều vô cùng mất thời gian và công sức nên cần có một phương pháp nhằm suy luận xấp xỉ một hàm Q nào đó và tính toán lại độ phân kỳ của chúng bằng KL-divergence (Kullback–Leibler divergence)[8] sẽ tiết kiệm thời gian, công sức rất nhiều.

Công thức tính độ phân kỳ của 2 phân phối xác suất P và Q là:

$$D_{KL}(P||Q) = \sum_i P_i \log \left(\frac{P_i}{Q_i} \right) \quad (2.17)$$

Suy luận theo cách truyền thống:

$$Q_i(Y) = \prod_i Q_i(Y_i) \quad (2.18)$$

Bước lặp suy luận:

Khởi tạo suy luận khi $j = i$:

$$Q_i(y_i) = \frac{1}{Z_i} \exp \left(-\phi_u(y_i) \right) \quad (2.19)$$

while số bước lặp suy luận **do**:

$$\hat{Q}_i^{(m)}(Y) = \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(Y') \quad (2.20)$$

$$\hat{Q}_i(y_i) = \sum_{y' \in Y} \mu(y, y') \sum_{m=1}^K w^{(m)} \hat{Q}_i^{(m)}(Y) \quad (2.21)$$

$$Q_i(y_i = Y) = \frac{1}{Z_i} \exp \left(-\phi(y_i|X) - \hat{Q}_i(y_i) \right) \quad (2.22)$$

Cập nhật lại $Q_i(y_i)$

end while

Theo lý thuyết xử lý ảnh, tồn tại 3 loại bộ lọc: thượng thông, hạ thông, trung bình. Bộ lọc thượng thông có chức năng làm nổi rõ chi tiết và đường biên, bộ lọc hạ thông có chức năng làm trơn ảnh và khử nhiễu, bộ lọc trung bình vừa có chức năng của thượng thông vừa có chức năng của hạ thông nên thường được sử dụng. Bên cạnh đó, bộ lọc Gaussian với khả năng của chúng vẫn có thể làm được chức năng của bộ lọc trung bình nhưng lại tốt hơn.

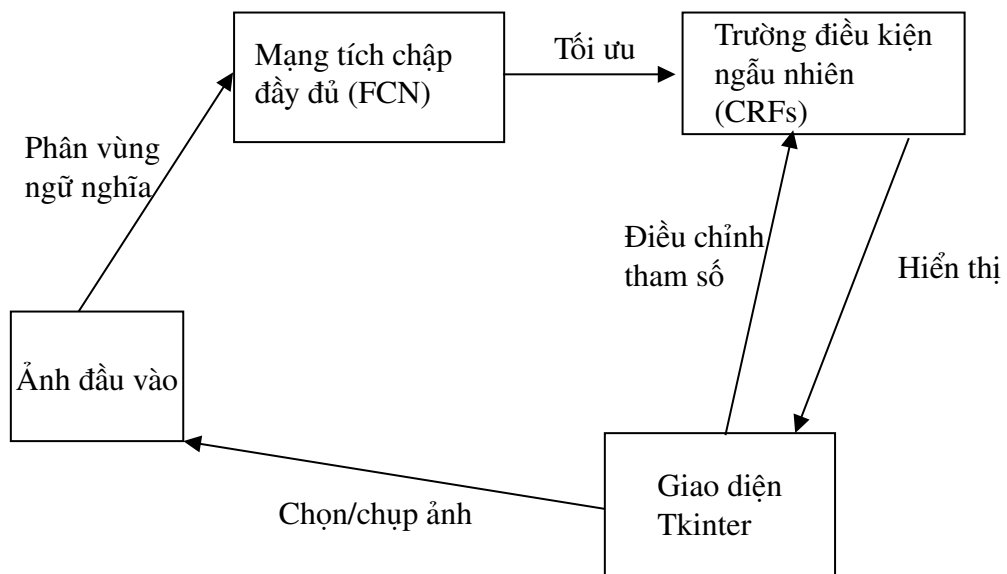
Bộ lọc với hàm nhân Gaussian được sử dụng để lan truyền thông điệp từ một đỉnh i đang xét tới đỉnh kề j nhằm cập nhật các giá trị ảnh hưởng giữa các điểm kề cận.

CHƯƠNG 3: KẾT QUẢ THỰC HIỆN

3.1 Phân vùng ngữ nghĩa và ứng dụng tách phong:

3.1.1 Sơ đồ chức năng ứng dụng:

Ứng dụng xây dựng trên bộ công cụ lập trình giao diện Tkinter của Python được sử dụng trong bài báo cáo để làm bộ phận hiển thị cho người dùng và kết hợp với các tính năng phân vùng ảnh ngữ nghĩa của hệ thống, đồng thời ghép ảnh nền vào ảnh đã tách phong, tạo ra những hình ảnh thú vị.



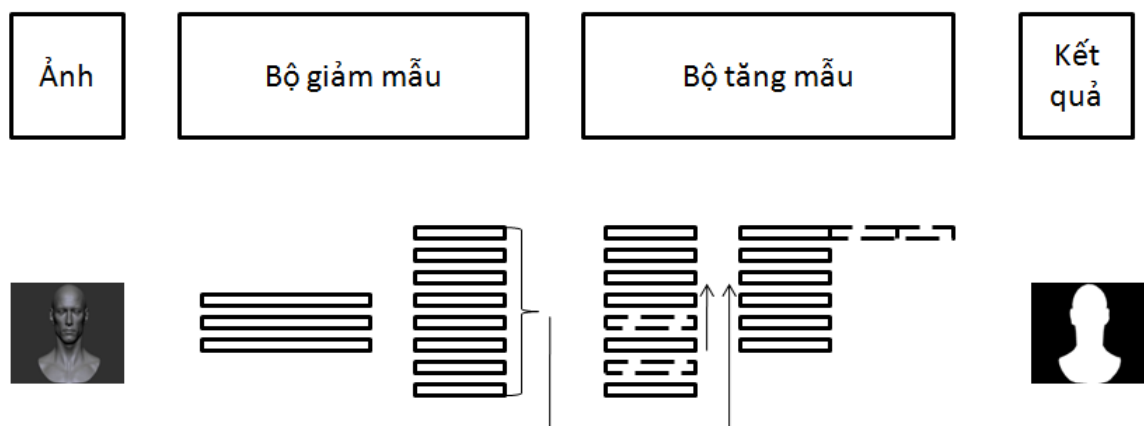
Hình 3.1: Sơ đồ vận hành ứng dụng

Sơ đồ vận hành ứng dụng với 2 phần chính: tính toán và hiển thị. Phần tính toán tích hợp mạng tích chập đầy đủ được nạp lại từ mô hình đã huấn luyện có định dạng HDF5, sau đó tiến hành dự đoán (predict) ảnh đã được tiền xử lý về đúng định dạng với các tấm ảnh huấn luyện có kích thước 224x224x3 bằng bộ thư viện OpenCV, quá trình tính toán lại xác suất sẽ tiếp nhận tại đầu ra và áp dụng luật của trường điều kiện ngẫu nhiên dựa trên 2 tham số, bước lặp tính lại xác suất Q xấp xỉ xác suất P cần xét và độ co giãn (scale) trên tấm ảnh. Phần hiển thị giao tiếp gửi ảnh gốc và nhận kết quả đã tính toán với phần tính toán, thực hiện quá trình hiển thị kết quả với các tham số thay đổi, thực hiện phép cộng độ phân giải giữa ảnh gốc người đã xóa phong và ảnh nền tùy chọn đã tạo vùng trống.

3.1.2 Phân vùng ngữ nghĩa:

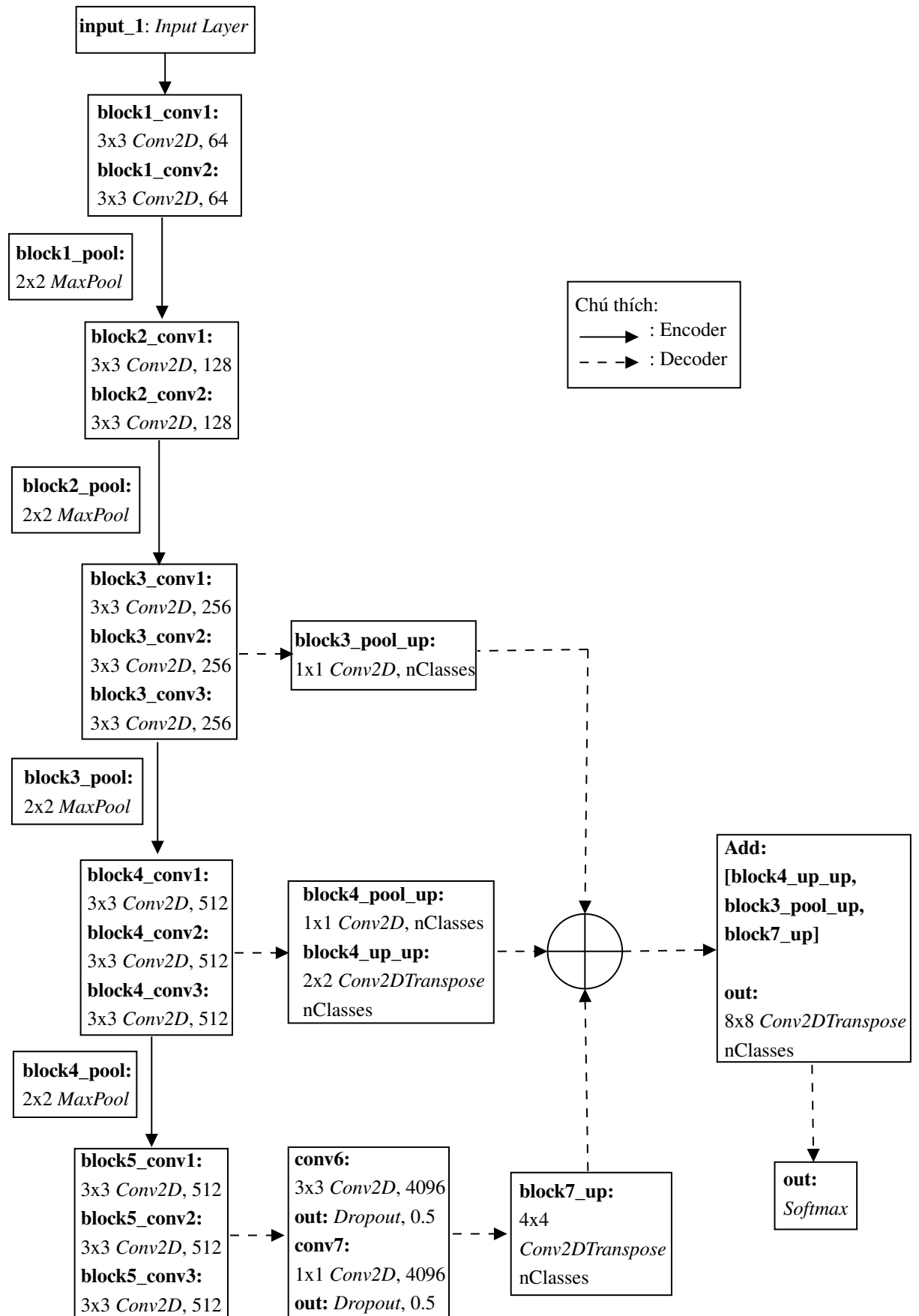
Quá trình phân vùng ngữ nghĩa có kết quả đầu ra tương đối giống với kỹ thuật phân vùng xử lý ảnh cơ bản nhưng cao cấp hơn vì chúng không chỉ xác định được một vùng tiền cảnh phân biệt với hậu cảnh mà chúng còn phân biệt được nhiều vùng với nhau. Kỹ thuật phân vùng ngữ nghĩa ứng dụng các kỹ thuật học sâu (deep learning) nhằm mục đích cắt nhỏ ảnh từ ảnh gốc, với kết quả thu được, lấy từng mảng ghép lại với nhau sẽ dễ dàng định hình được vị trí cụ thể của đối tượng nằm chính xác ở đâu trong ảnh.

Hoạt động xây dựng sử dụng kỹ thuật phân vùng ngữ nghĩa FCN với bộ giảm mẫu gồm các lớp tích chập và thăm dò, bộ tăng mẫu gồm tích chập chuyển vị và chuyển tiếp nối kết, nối tiếp nhau tạo ra kiến trúc huấn luyện mô hình khá chặt chẽ. Bộ thư viện Keras của Python được sử dụng để xây dựng kiến trúc đã nêu vì chúng khá đơn giản, thông thường cũng được sử dụng để tạo ra các loại mạng nơ-ron như CNN, RNN,...



Hình 3.2: Phân vùng ngữ nghĩa cắt ghép ảnh

Kiến trúc lõi được sử dụng là VGGNet (Visual Geometry Group Network), được miêu tả cụ thể theo sơ đồ sau:



Hình 3.3: Kiến trúc sử dụng trong bài báo cáo

Kiến trúc phân vùng ngữ nghĩa với tất cả 9 khối tích chập và thăm dò xen kẽ nhau và 5 khối tích chập chuyển vị với các lớp liên quan. Các hoạt động chuyển tiếp nối kết không được liên kết hết từ lớp đầu tiên tới lớp cuối cùng mà được sắp đặt một cách vừa phải từ block3, block4, block5 nhằm không gây ra tình trạng học vẹt hay lãng phí bộ nhớ. Các kỹ thuật này tác động trực tiếp lên các trọng số nối kết. Chính vì thế, chúng cũng duy trì lại những thông tin đã và đang thay đổi hơn là những phương pháp nội suy thủ công, không có trọng số để học. Bộ tích chập chuyển vị Conv2DTranspose được sử dụng để khôi phục lại hình ảnh ban đầu nhưng lại không phải là tích chập ngược Deconvolution vì chúng không đảm bảo chắc chắn kết quả khôi phục giống y đầu vào nhưng vẫn thực hiện chúng một cách xấp xỉ tốt nhất có thể.

Sự phân biệt giữa 3 giải thuật FCN-32s, FCN-16s, FCN-8s căn cứ vào số lượng nối kết chuyển tiếp và bước trượt tăng mẫu. Đối với FCN-8s, được xem là kiến trúc đầy đủ nhất và cũng là kiến trúc 3.3 ứng với tất cả nối kết của block3, block4, block5 cùng bước trượt bằng 8. Đối với FCN-16s, là kiến trúc đơn giản hơn khi chỉ có nối kết của block4, block5 cùng bước trượt bằng 16. Đối với FCN-32s, kiến trúc đơn giản nhất với 1 nối kết của block5 và bước trượt khá thưa lên đến 32. Kết quả đầu ra của mỗi kiến trúc được truyền qua hàm mất mát nhằm đánh giá mức độ ảnh hưởng đúng sai của các trọng số phía trước tác động vào dữ liệu thu được ở đầu ra.

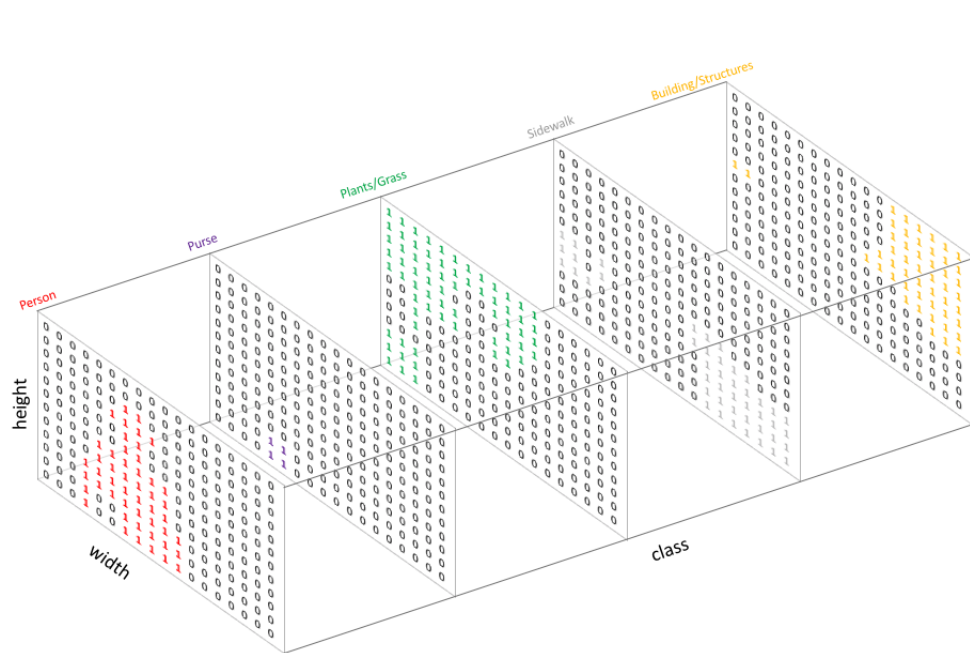
3.1.3 Hàm mất mát:

Có thể nói rằng, sự sai khác của quá trình đoán kết quả với kết quả thực sự luôn luôn xảy ra và dĩ nhiên ở mạng tích chập đầy đủ cũng vậy. Mạng tích chập đầy đủ (FCN) cũng sử dụng hàm mất mát Cross-entropy của các loại mạng nơ-ron tích chập CNN nguyên thủy khác để tính toán lỗi sau khi đã đưa dữ liệu qua hàm Softmax giúp cân bằng chúng sao cho tổng của toàn bộ phải bằng 1. Lý do sử dụng hàm Cross-entropy vì chỉ cần thực hiện quá trình so khớp đúng sai giữa vị trí điểm ảnh đã đoán được trên ma trận số đầu ra với điểm ảnh thực sự của nhãn, sau đó trả về giá trị trung bình của chúng. Ta có thể hình dung như sau:

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4

Hình 3.4: Nhãn của ảnh [Nguồn: www.jeremyjordan.me]

Dữ liệu thô của quá trình đầu ra thực sự có dạng:



Hình 3.5: Dữ liệu mô hình đoán được [Nguồn: www.jeremyjordan.me]

Ta có công thức tính hàm mất mát trên toàn bộ dữ liệu như sau:

$$J(W; X, Y) = - \sum_{i=1}^N \sum_{j=1}^C y_{ji} \log(a_{ji}) \quad (3.1)$$

với hàm Softmax:

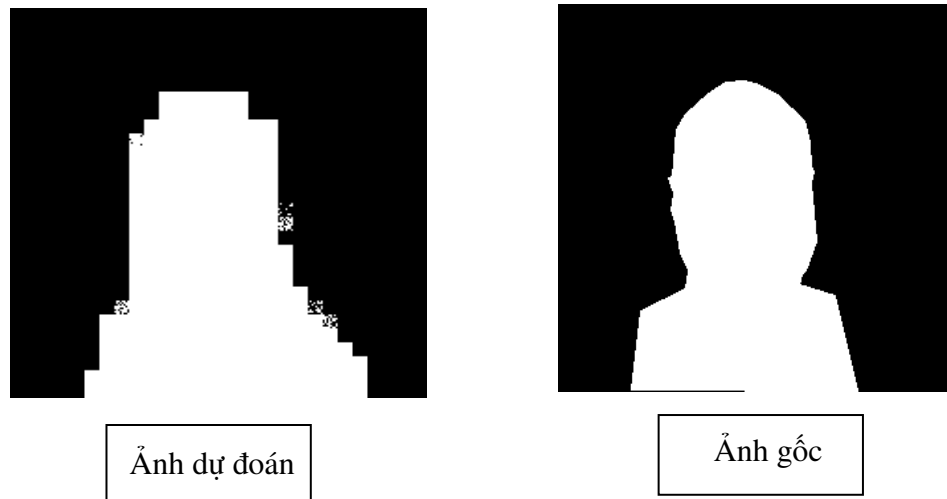
$$a_{ji} = \frac{e^{W_j^T x_i}}{\sum_{k=1}^C e^{W_k^T x_i}} \quad (3.2)$$

- y_{ji} và a_{ji} lần lượt là phần tử thứ j của vectơ y_i và a_i
 x_i là dữ liệu đầu vào có phân phối xác suất $[y_i; 1 - y_i]$
 y_i là xác suất tính toán đầu vào sẽ nằm ở lớp i
 a_{ji} là giá trị đầu ra nằm trong đoạn $[0,1]$ đã tính Softmax.
 i là số nguyên nằm trong đoạn từ 1 đến N
 N là số lượng tất cả các cặp dữ liệu x và y
 C là số lớp cần phân lớp.
 W là ma trận trọng số từ 1 đến C

Nhiệm vụ của chúng ta là tìm được cực trị thỏa yêu cầu từ công thức trên và tất nhiên khi nhắc đến cực trị, người ta sẽ nhắc đến sự biến thiên và cụ thể hơn người ta nhắc đến đạo hàm. Sau đó, có thể tính được mức độ ảnh hưởng của các trọng số để đưa ra kết quả đúng sử dụng kỹ thuật giảm gradient ngẫu nhiên đã nêu và lan truyền tới các lớp phía trước chính là những bước cần phải làm của mỗi một lần huấn luyện trên từng tấm ảnh. Kỹ thuật tính toán cổ điển này luôn được áp dụng cho các loại mạng nơ-ron để cập nhật trọng số nối kết.

3.1.4 CRFs và các mô hình thống kê:

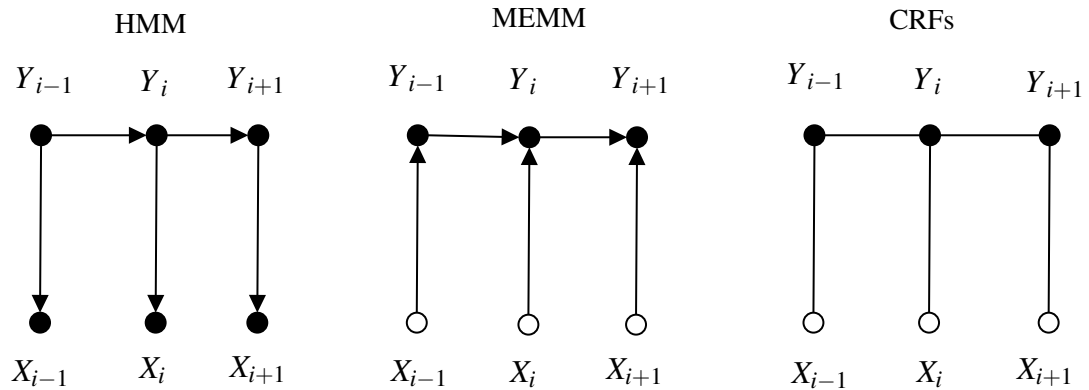
Mặc dù đã trải qua rất nhiều sự tính toán phức tạp của mạng tích chập đầy đủ gồm giảm mẫu (downsampling), tăng mẫu (upsampling), cập nhật trọng số từ kết quả của hàm Softmax nhưng kết quả khi thực nghiệm vẫn rất tệ.



Hình 3.6: So sánh kết quả đầu ra

Việc phân vùng ngữ nghĩa xuất hiện tình trạng trên là vô cùng thường xuyên, để khắc phục thay vì thực hiện những kiến trúc cao cấp hơn và sâu hơn thì tác giả quyết định sử dụng một phương pháp nhằm hậu xử lý (post-processing) các chi tiết sau khi đã thực hiện huấn luyện và giữ lại những điểm ảnh có độ chắc chắn cũng như ổn định cao, đồng thời loại bỏ hay trừng phạt những điểm ảnh mang dấu hiệu không chắc chắn bằng kỹ thuật từ các mô hình đồ thị xác suất thống kê.

HMM, MEMM, CRFs được xem là 3 loại đồ thị thống kê rất nổi tiếng trong lĩnh vực tối ưu xác suất nhưng mỗi loại lại có đặc trưng khác nhau và cách tổ chức khác nhau.



Hình 3.7: So sánh các loại đồ thị xác suất.

Mô hình Markov ẩn (HMM) tính toán giá trị xác suất dựa vào xác suất chung (Joint probability distribution) $P(Y, X)$ và quá trình gán nhãn (labeling) xảy ra trên từng ngữ cảnh (context) tách biệt nên không thể xem xét hết toàn bộ bối cảnh xảy ra của sự việc.

Mô hình Markov cực đại hóa entropy (MEMM) chỉ thực hiện phép tính xác suất có điều kiện $P(Y|X)$ giảm thiểu sự tính toán phức tạp, tìm hiểu được sự nhất quán giữa kết quả mục tiêu và kết quả đoán được nhưng lại gặp phải vấn đề. Cụ thể, MEMM chỉ thực hiện phép tính cục bộ vì một trạng thái chỉ biết kết quả đầu ra trạng thái trước và đầu vào trạng thái sau đó. Cũng chính vì vậy, MEMM sẽ xảy ra hiện tượng gán nhãn thiên vị (labeling bias), tức là một trạng thái khi xét ở qui mô cục bộ nếu liên kết với nhiều trạng thái hơn thì sẽ phân bổ xác suất cho mỗi phần ít hơn. Như vậy, trạng thái ít có sự chuyển đổi lại được MEMM chọn lựa để tính toán và gây thất bại trong quá trình tối ưu.

Xét thấy HMM và MEMM đều là những đồ thị xác suất có hướng, CRFs là đồ thị xác suất vô hướng. Khác với HMM, CRFs không tự đặt ra giới hạn ngữ cảnh riêng biệt cho mỗi thành phần. Còn đối với MEMM, CRFs có thể quan sát được tất cả các trạng thái bên trong và khắc phục được tình trạng gán nhãn thiên vị (labeling bias).

Điểm đáng chú ý của trường điều kiện ngẫu nhiên là sự phức tạp của thuật toán nhưng sự đột phá xuất hiện trong nghiên cứu [8] đã nêu ra phương pháp để tối ưu kỹ thuật tối ưu này.

3.1.5 Truyền thông điệp CRFs kết hợp tích chập Gaussian:

Kết quả đầu ra được tối ưu xác suất bằng trường điều kiện ngẫu nhiên thông qua 2 dữ liệu đầu vào: ảnh gốc và ảnh dự đoán nhằm mục tiêu thiết đặt xác suất có điều kiện ảnh gốc cho mỗi giá trị xác suất ứng với từng điểm ảnh trên ảnh dự đoán. Phương pháp suy luận của trường điều kiện ngẫu nhiên được chú trọng hơn nhằm tiết kiệm thời gian nhưng vẫn đảm bảo kết quả được tối ưu hết mức có thể.

Quay trở lại mục 2.2.3, ta có công thức 2.22 có độ phức tạp là $O(n)$, thực hiện quá trình cập nhật xác suất cho n điểm với mỗi điểm căn cứ vào xác suất trước đó để tính toán.

Công thức 2.21 có độ phức tạp là $O(n)$, thực hiện quá trình biến đổi tương thích cho n điểm với mỗi điểm được thay đổi và cân bằng lại sau khi đã thực hiện truyền đi giá trị biến đổi giữa thành phần phụ thuộc.

Công thức 2.20 lại có độ phức tạp là $O(n^2)$, thực hiện quá trình truyền đi thông điệp lần lượt của n điểm ảnh với $Q_j(Y')$ là công thức được lặp lại trước đó của 2.22 gồm các thành phần sở hữu độ phức tạp $O(n)$.

Chính vì thế, để làm cho độ phức tạp từ dạng bậc 2 ($O(n^2)$) chuyển thành dạng tuyến tính ($O(n)$) là sự đột phá độc đáo của quá trình này [8]. Xét thấy trong công thức 2.20, mục tiêu là truyền đi thông điệp từ một đỉnh i đang xét tới đỉnh kề j của nó bằng hàm tiềm năng đơn phương $\varphi(y_i|X)$ nhưng nó cũng là nguyên nhân chính vì bản thân chúng phải thực hiện phép tổng mà bên trong là một phép tổng khác trên một không gian lớn cố định.

Sử dụng phép toán tích chập để làm giảm không gian lớn đã nêu với đầu ra có khoảng cách so với đầu vào tỉ lệ thuận với độ lệch chuẩn của chúng, thực hiện phép tổng kết hợp ban đầu trên không gian đã biến đổi, có thể giải quyết được vấn đề độ phức tạp. Hàm nhân Gaussian thay vì được sử dụng để truyền thông điệp thì được sử dụng để tích chập sẽ thu được những giá trị đã được chuẩn hóa (bằng 0 nếu nằm bên ngoài khoảng giá trị độ lệch chuẩn) và không đổi.

Kết quả sau khi đã thực hiện kỹ thuật CRFs như sau:



Hình 3.8: So sánh ảnh chưa xử lý và đã xử lý

Xét thấy kết quả đã xử lý bằng mô hình xác suất tối ưu CRFs mang lại tốt hơn và có thể mô tả được hình dáng của vật thể chiếm trong ảnh. Điều đó cũng cho thấy sự đúng đắn khi kết hợp FCN và CRFs.

3.2 Tập dữ liệu:

3.2.1 Thu thập dữ liệu:

Quá trình huấn luyện của mạng tích chập đầy đủ (FCN) cần tập dữ liệu đủ tốt, có nghĩa là tập dữ liệu có được sự phong phú về góc sáng, bối cảnh, đối tượng, hình dạng của đối tượng, độ phân bố của điểm ảnh, số lượng đối tượng trong ảnh, hành động của đối tượng trong ảnh,... Tuy nhiên, tập dữ liệu huấn luyện quá trình phân vùng ngữ nghĩa không cần quá nhiều ảnh vì nhu cầu của bài báo cáo chỉ đơn giản là xây dựng phân vùng ảnh người đơn giản nhưng cần có chất lượng được đảm bảo, dưới đây là một số tập dữ liệu đủ tốt để phân vùng ngữ nghĩa ảnh:

Tập dữ liệu	Chủ đề	Số lượng	Năm ra đời	Mô tả
<i>Pascal Voc 2012</i>	Chung	17125	2012	Tập dữ liệu chứa nhiều đối tượng như máy bay, người, xe, tàu lửa, động vật,...
<i>Unite the People (UP-S31)</i>	Dáng người	8515	2017	Hình ảnh toàn thân con người khi di chuyển.
<i>Part Labels (Labeled Faces in the Wild - LFW)</i>	Chân dung người	2927	2013	Gán nhãn mặt, đầu và phong nền.
<i>Face/Headseg (FH)</i>	Chân dung người	75	2018	Tập dữ liệu được trích từ 19002 ảnh gán nhãn mặt, mũi, tóc, tai, mắt, lông mày và phong nền.
<i>SVCNTT-2019</i>	Chân dung người	130	2019	Tập dữ liệu được tác giả thu thập trực tiếp từ các sinh viên khoa Công nghệ thông tin và truyền thông (10 tấm/1 sinh viên).

Bảng 3.1: Các tập dữ liệu phổ biến cho phân vùng ngữ nghĩa ảnh

Xét thấy tập dữ liệu Pascal Voc 2012 có được sự phong phú nhưng tồn tại nhiều đối tượng không phải con người và cần có quá trình tiền xử lý phức tạp, còn tập dữ liệu UP-S31 có rất nhiều hình dáng con người không phù hợp với mục tiêu của đề tài. Tập dữ liệu Part Labels (LFW) tồn tại hình chân dung người nhưng có độ nhiễu khá lớn sẽ tăng độ khó của tập dữ liệu. Chính vì lý do trên, tác giả quyết định sử dụng kết hợp hai tập dữ liệu FH và LFW với mục tiêu huấn luyện diện mạo con người. Ngoài ra, tác giả còn kết hợp tập SVCNTT-2019 để kiểm tra mô hình.

Như vậy, có thể đáp ứng được nhu cầu huấn luyện mô hình chuyên dùng nhằm tách lấy chân dung người trong ảnh.

3.2.2 Phân tích tập dữ liệu sử dụng:

Tập dữ liệu với 525 tấm ảnh màu được chia làm tập huấn luyện và kiểm tra. Mỗi tập được chia làm 2 loại: ảnh nhân và ảnh gốc (đơn vị: tấm ảnh).

STT	Tập dữ liệu	Số lượng tổng	Kích thước	Huấn luyện	Kiểm tra
1	<i>FH</i>	75	300 x 280	75	0
2	<i>LFW</i>	450	250 x 250	320	0
3	<i>SVCNTT-2019</i>	130	640 x 480	0	130
			Tổng:	395	130

Bảng 3.2: Các tập dữ liệu được sử dụng

Tập dữ liệu đã được tác giả tiền xử lý thành 2 loại nhân như sau:

STT	Loại nhân	Giá trị điểm ảnh đại diện	Màu vùng ảnh
1	Đối tượng người	$[R, G, B] = [255, 255, 255]$	Trắng
2	Ảnh nền	$[R, G, B] = [0, 0, 0]$	Đen

Bảng 3.3: Thông tin các loại nhân trong tập dữ liệu sử dụng

Quá trình huấn luyện với 395 tấm ảnh được chia ra thành 2 phần: 1 phần dùng để huấn luyện FCN, phần còn lại dùng để đánh giá trong từng lần lặp (epochs) với số lượng chi tiết như sau (đơn vị: tấm ảnh):

Huấn luyện FCN	Đánh giá FCN
335	60

Bảng 3.4: Phân chia tập dữ liệu huấn luyện

3.3 Kiểm tra kết quả:

Mỗi tập dữ liệu đều có độ khó khác nhau và vai trò khác nhau nhằm đại diện cho một vấn đề nào đó, cung cấp một đại diện cho vấn đề giúp người xử lý thực hiện các giải thuật giải quyết chúng trong một điều kiện lý tưởng nhất định. Tập dữ liệu kết hợp được sử dụng trong bài báo cáo đại diện cho bài toán tách lấy người trong ảnh và sử dụng giải thuật để phân vùng ngữ nghĩa chân dung người độc lập (1 người) trong ảnh.

3.3.1 Các thông số sử dụng:

Giải thuật phân vùng ngữ nghĩa được sử dụng là mô hình mạng tích chập đầy đủ (FCN) đã được huấn luyện và hậu xử lý để tối ưu kết quả hiển thị bằng trường điều kiện ngẫu nhiên (CRFs). Các thông số của quá trình phân vùng ngữ nghĩa đối tượng người độc lập trong ảnh bằng FCN với tập dữ liệu trên như sau:

Thông số huấn luyện	Số liệu	Chú thích
Kích thước ảnh đã được tinh chỉnh (W x H)	224 x 224	Kích thước của ảnh phải là kết quả của phép toán $2^n, n \in \mathbb{N}$ do kiến trúc FCN trượt (stride) bộ lọc theo đơn vị của phép toán trên.
Bó dữ liệu (batchsize)	32	Tham khảo từ [1]
Số lần lặp (epochs)	200	Thông số được đề xuất từ quá trình thực nghiệm
Momentum	0.9	Tham khảo từ [1]
Tốc độ học (learning rate)	0.01	Tham khảo từ [1]
Trọng số tiêu biến (decay)	0.01 / 200	(Tốc độ học / số lần lặp)

Bảng 3.5: Các thông số dùng để huấn luyện FCN

Các thông số được sử dụng của trường điều kiện ngẫu nhiên để tối ưu gồm:

Thông số tối ưu	Số liệu	Chú thích
Độ co giãn (Scale)	0.7	$image * 0.7 + numpy.ones(image.shape) * (1 - 0.7) / classes$
Siêu thông số σ_α	80	Tham khảo tại [8]
Siêu thông số σ_β	13	Tham khảo tại [8]
Siêu thông số σ_γ	3	Tham khảo tại [8]
Số bước lặp suy luận (Inference)	5	Tham khảo tại [8]

Bảng 3.6: Các thông số dùng để tối ưu CRFs

3.3.2 Tính toán các độ đo:

Độ đo được xem là đại lượng cực kỳ quan trọng giúp đánh giá chính xác kết quả vận hành sau khi quá trình huấn luyện của mạng tích chập đầy đủ (FCN) ứng với tập dữ liệu cụ thể kết thúc. Có 2 đại lượng cần quan tâm, một là chỉ số meanIU, hai là chỉ số F1 nhưng để tính được các chỉ số trên ta cần xây dựng ma trận contingency.

3.3.2.1 Ma trận contingency:

Ma trận được xây dựng dựa trên 2 loại nhãn của điểm ảnh (tiền cảnh (foreground), hậu cảnh (background)) và 4 thuộc tính quan trọng:

- TP (True Positive): số lượng điểm của lớp foreground được phân loại đúng là foreground.
- TN (True Negative): số lượng điểm của lớp background được phân loại đúng là background.
- FP (False Positive): số lượng điểm của lớp background bị phân loại nhầm thành foreground.
- FN (False Negative): số lượng điểm của lớp foreground bị phân loại nhầm thành background.

Dự đoán =>	Dương	Âm
Dương	TP	FN
Âm	FP	TN

Bảng 3.7: Ma trận contingency

3.3.2.2 Chỉ số meanIU:

Đây được xem là loại chỉ số được sử dụng phổ biến nhất khi phân vùng ngữ nghĩa đối tượng trong ảnh. Chỉ số meanIU là kết quả sau khi đã thực hiện phép chia trung bình với tập hợp các chỉ số IoU (Intersection over Union) của tất cả các lớp. Chỉ số IoU chính là tỉ số giữa những vùng chứa các độ phân giải bị trùng lặp của ma trận kết quả thực sự và ma trận kết quả dự đoán được với tổng số độ phân giải của cả hai ma trận kết quả.

$$meanIU = average(IoU) = average\left(\frac{TP}{TP + FP + FN}\right) \quad (3.4)$$

3.3.2.3 Chỉ số F1:

Chỉ số F1 là một phép đo quan trọng được tính toán bằng kết quả của:

$$prec = \frac{TP}{TP + FP} \quad (3.5)$$

$$rec = \frac{TP}{TP + FN} \quad (3.6)$$

Công thức để tính toán F1 là:

$$F1 = \frac{2 * prec * rec}{prec + rec} \quad (3.7)$$

3.3.3 Kết quả trên tập dữ liệu huấn luyện:

Bảng máy chủ chuyên dụng Tesla K80 GPU miễn phí của Google Colab với giới hạn vùng nhớ sử dụng 11.281.553.818 Byte (11.281 GigaByte), tác giả tiến hành đánh giá với 3 biến thể [1] của mạng tích chập đầy đủ (FCN) gồm: FCN-32s, FCN-16s và FCN-8s. Các biến thể trên với FCN-32s là đơn giản nhất, kể đến là FCN-16s và cuối cùng đầy đủ nhất là FCN-8s.

	Độ chính xác cuối cùng (%)	MeanIU	Tổng thời gian
FCN-32s	80.29	0.644	1 giờ 10 phút 26 giây
FCN-16s	84.46	0.715	1 giờ 6 phút 43 giây
FCN-8s	84.91	0.721	1 giờ 6 phút 43 giây

Bảng 3.8: So sánh các biến thể của FCN trên tập huấn luyện.

Các thông số huấn luyện chi tiết như sau:

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 716774	$prec = 0.846$	#TP = 1700542	$prec = 0.786$
#TN= 1700542	$rec = 0.608$	#TN= 716774	$rec = 0.929$
#FP= 130380	$F1 = 0.707$	#FP= 462864	$F1 = 0.851$
#FN= 462864	$IoU = 0.547$	#FN= 130380	$IoU = 0.741$
Mean IU= 0.644			

Bảng 3.9: Kết quả tập huấn luyện FCN-32s

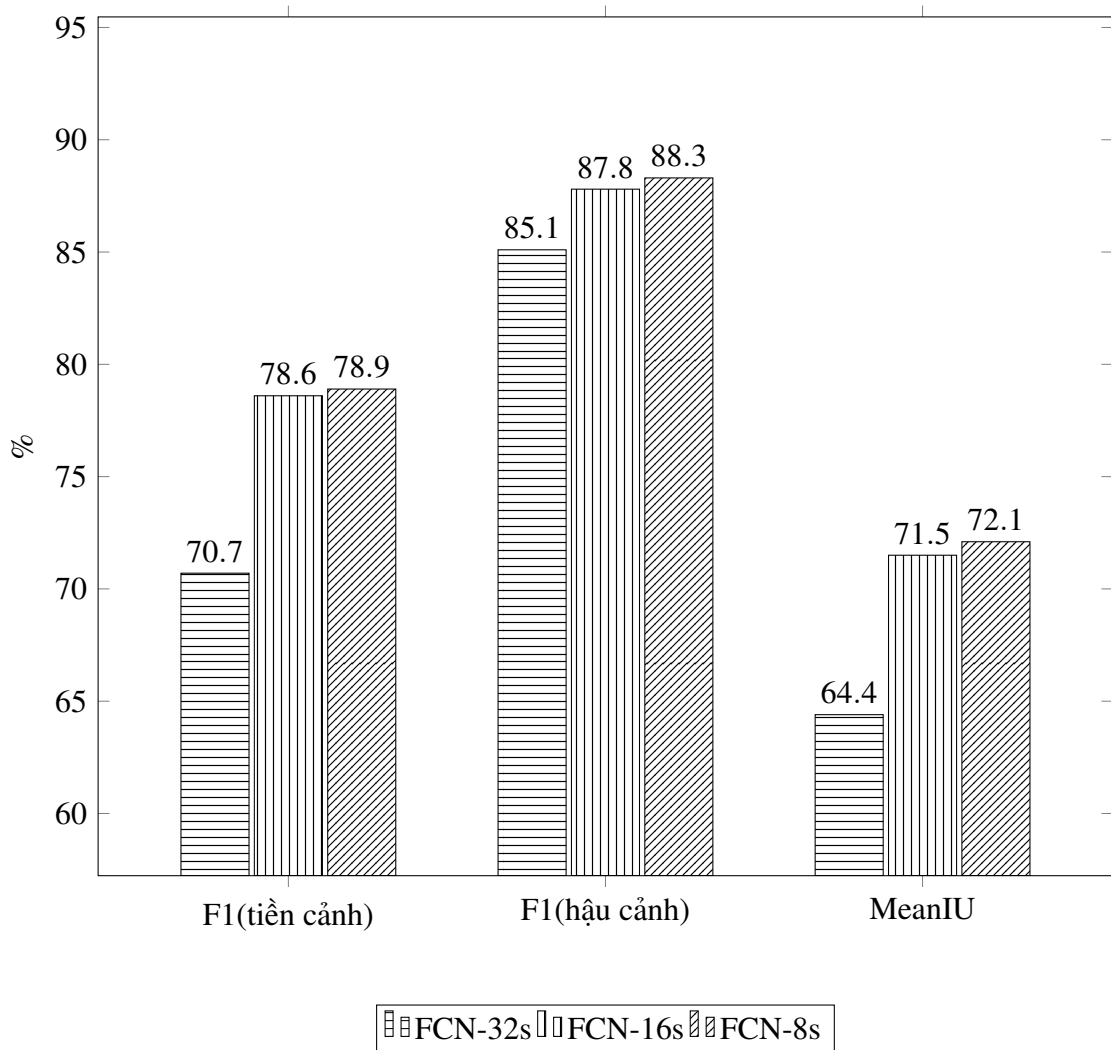
Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 858074	$prec = 0.854$	#TP = 1684506	$prec = 0.840$
#TN= 1684506	$rec = 0.727$	#TN= 858074	$rec = 0.920$
#FP= 146416	$F1 = 0.786$	#FP= 321564	$F1 = 0.878$
#FN= 321564	$IoU = 0.647$	#FN= 146416	$IoU = 0.783$
Mean IU= 0.715			

Bảng 3.10: Kết quả tập huấn luyện FCN-16s

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 849453	$prec = 0.873$	#TP = 1706839	$prec = 0.838$
#TN= 1706839	$rec = 0.720$	#TN= 849453	$rec = 0.932$
#FP= 124083	$F1 = 0.789$	#FP= 330185	$F1 = 0.883$
#FN= 330185	$IoU = 0.652$	#FN= 124083	$IoU = 0.790$
Mean IU= 0.721			

Bảng 3.11: Kết quả tập huấn luyện FCN-8s

Đồ thị mô tả như sau:



Hình 3.9: Đồ thị so sánh các biến thể của FCN trên tập huấn luyện.

Xét thấy kết quả trên cả 3 giải thuật: FCN-32s, FCN-16s và FCN-8s có sự phân biệt rõ ràng ứng với tập dữ liệu huấn luyện, điều đó cho thấy quá trình huấn luyện diễn ra khá tốt và kết quả có thể được sử dụng như một đại lượng tham khảo ứng với khả năng tính toán của từng loại giải thuật.

3.3.4 Kết quả trên tập dữ liệu kiểm tra:

Bảng kỹ thuật giống như phương pháp huấn luyện nhằm kiểm tra sự chính xác cho hệ thống. Kết quả như sau:

3.3.4.1 Kiểm tra FCN thuần:

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 1395914	$prec = 0.700$	#TP = 4236871	$prec = 0.936$
#TN= 4236871	$rec = 0.828$	#TN= 1395914	$rec = 0.876$
#FP= 599288	$F1 = 0.758$	#FP= 290807	$F1 = 0.905$
#FN= 290807	$IoU = 0.611$	#FN= 599288	$IoU = 0.826$
Mean IU= 0.719			

Bảng 3.12: Kết quả tập kiểm tra FCN-32s

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 1571452	$prec = 0.663$	#TP = 4038257	$prec = 0.972$
#TN= 4038257	$rec = 0.932$	#TN= 1571452	$rec = 0.835$
#FP= 797902	$F1 = 0.775$	#FP= 115269	$F1 = 0.898$
#FN= 115269	$IoU = 0.632$	#FN= 797902	$IoU = 0.816$
Mean IU= 0.724			

Bảng 3.13: Kết quả tập kiểm tra FCN-16s

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 1464869	$prec = 0.718$	#TP = 4262000	$prec = 0.951$
#TN= 4262000	$rec = 0.868$	#TN= 1464869	$rec = 0.881$
#FP= 574159	$F1 = 0.786$	#FP= 221852	$F1 = 0.915$
#FN= 221852	$IoU = 0.648$	#FN= 574159	$IoU = 0.843$
Mean IU= 0.745			

Bảng 3.14: Kết quả tập kiểm tra FCN-8s

Sự khác biệt chính giữa 3 biến thể FCN-32s, FCN-16s và FCN-8s là bước trượt (*stride*) được sử dụng để tăng mẫu (*upsampling*) của FCN lần lượt là 32, 16 và 8. Có thể nói rằng, không thể so sánh các giải thuật máy học bằng khả năng tính toán của chúng một cách riêng lẻ mà còn phụ thuộc vào tập dữ liệu nhưng tập dữ liệu dù lớn cách mấy cũng không thể bao phủ toàn bộ giả thuyết của đời sống do đó, kết quả thu được chỉ đúng trong bối cảnh của từng tập dữ liệu và không thể đánh giá tổng quát sự hơn kém của các loại giải thuật máy học mà chỉ dựa vào số liệu ở trên.

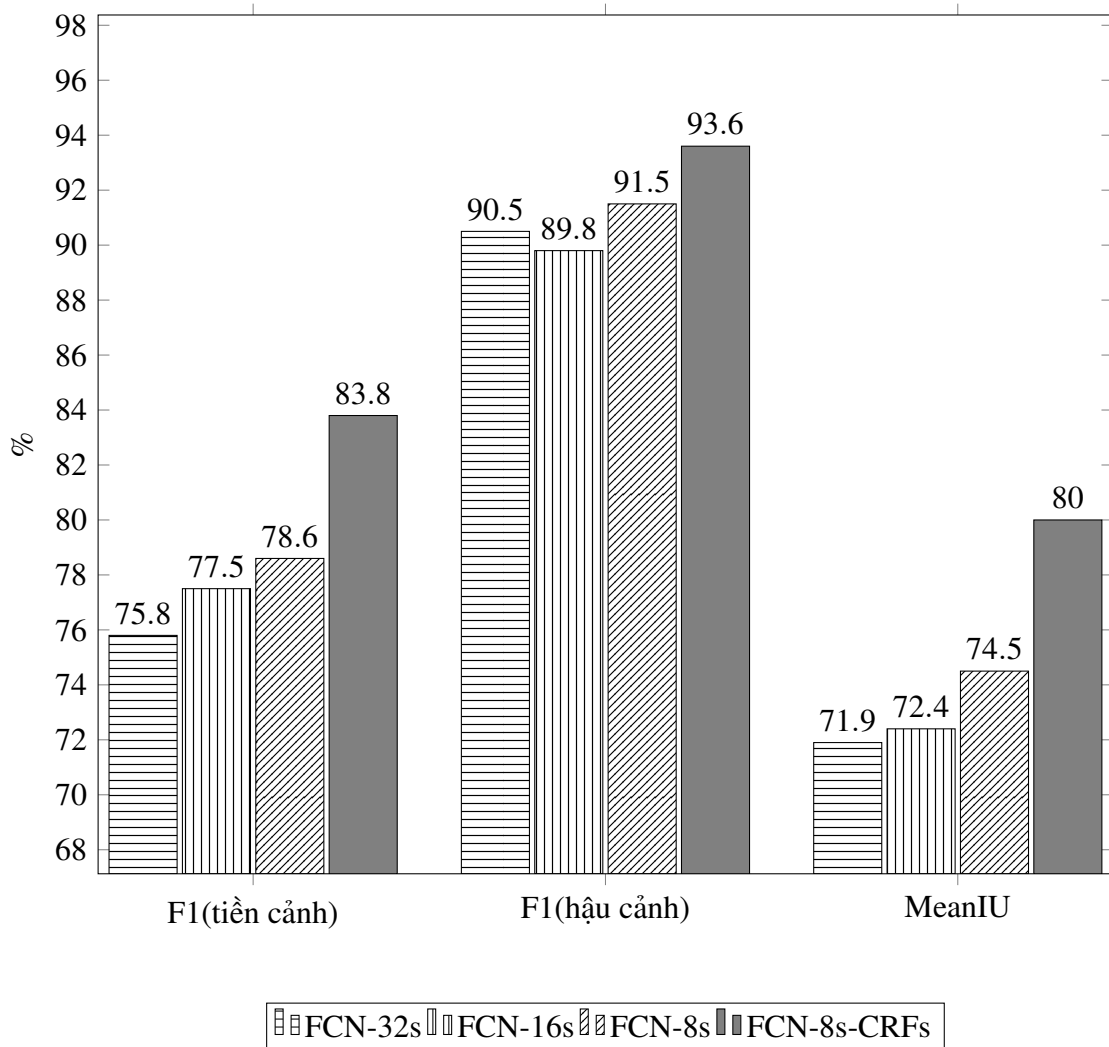
Xét thấy FCN-8s có thông số meanIU lớn nhất trong ngữ cảnh đang xét. Theo một cách chủ quan, tác giả ưu tiên thực hiện tối ưu CRFs trên mô hình FCN-8s.

3.3.4.2 Tối ưu kết quả kiểm tra:

Kết quả tiền cảnh (foreground)		Kết quả hậu cảnh (background)	
#TP = 1550264	$prec = 0.769$	#TP = 4371210	$prec = 0.970$
#TN= 4371210	$rec = 0.919$	#TN= 1550264	$rec = 0.904$
#FP= 464949	$F1 = 0.838$	#FP= 136457	$F1 = 0.936$
#FN= 136457	$IoU = 0.720$	#FN= 464949	$IoU = 0.879$
Mean IU= 0.800			

Bảng 3.15: Kết quả tập kiểm tra sau khi sử dụng CRFs

Đồ thị đánh giá chi tiết như sau:



Hình 3.10: Đồ thị so sánh các biến thể của FCN trên tập kiểm tra.

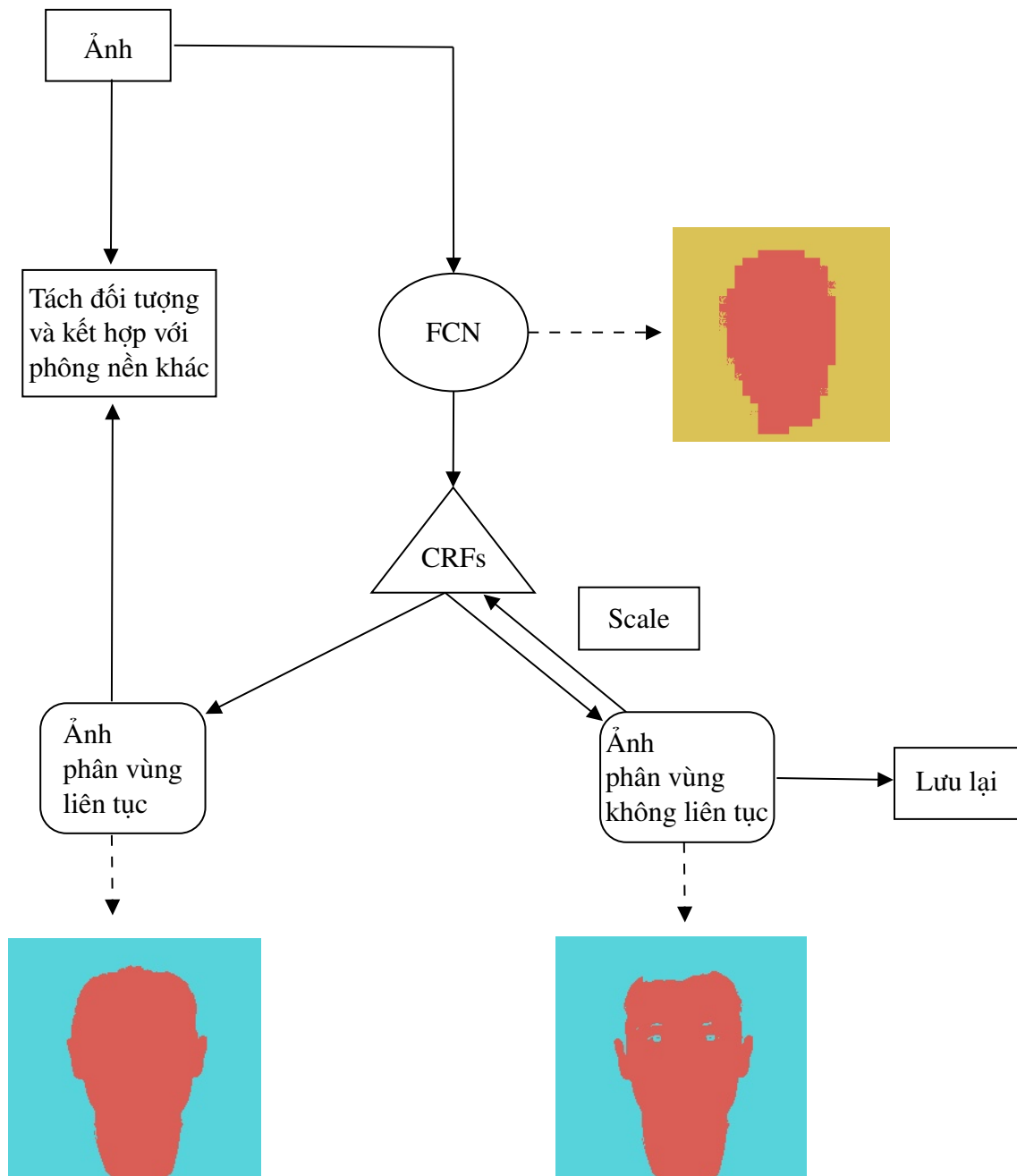
Xét thấy kết quả sau khi tối ưu mang lại hình ảnh khá tốt nhưng không phải trong mọi trường hợp CRFs đều đáp ứng được. CRFs phụ thuộc vào dữ liệu ảnh gốc khi tiến hành tối ưu và tất nhiên nếu ảnh gốc cần tách phòng có độ khó lớn tức là nhiều lớn, hoặc sự phân bố giá trị điểm ảnh của đối tượng người trên ảnh gốc tương đồng với phòng nền phía sau sẽ gây ra tình trạng mất cân bằng các lớp (Class Imbalance) có nghĩa là hậu cảnh chiếm tỉ lệ rất cao so với tiền cảnh và tiền cảnh bị CRFs kết luận nhầm thành hậu cảnh.

Để cải thiện tình trạng trên Yu Liu và các cộng sự đã công bố bài báo [14] với hai đóng góp quan trọng. Một là trước khi thực hiện lan truyền ngược [11], đại lượng lan truyền ngoài đầu ra thông thường (Softmax loss function) còn được cộng thêm đại lượng bổ sung (Positive-sharing loss function) nhằm cân bằng giá trị xác suất của tiền cảnh và hậu cảnh. Hai là phương pháp bổ sung đại lượng POS vào hàm tiềm năng đơn phương (Unary potential) nhằm mục tiêu đo lường xác suất một điểm được định vị vào đối tượng nào đó kèm theo giả thiết rằng nếu như nhiều đối tượng cùng chứa một điểm ảnh nào đó thì điểm ảnh đó phải được đặt trọng số lớn hơn. Nội dung bài báo cáo không cố gắng tích hợp thêm lý thuyết trên vì với lý thuyết trước đó đã đáp ứng được nhu cầu đề ra.

3.4 Kết quả vận hành mô hình:

3.4.1 Kịch bản vận hành:

Phần cứng để thực hiện kiểm thử các chức năng của ứng dụng là máy tính cá nhân Dell, hệ điều hành Windows 7 64-bit, bộ xử lý Intel Core i5-2540M CPU @ 2.60GHz và RAM 8GigaByte. Quá trình vận hành và biến đổi dữ liệu chi tiết được thể hiện như sau:

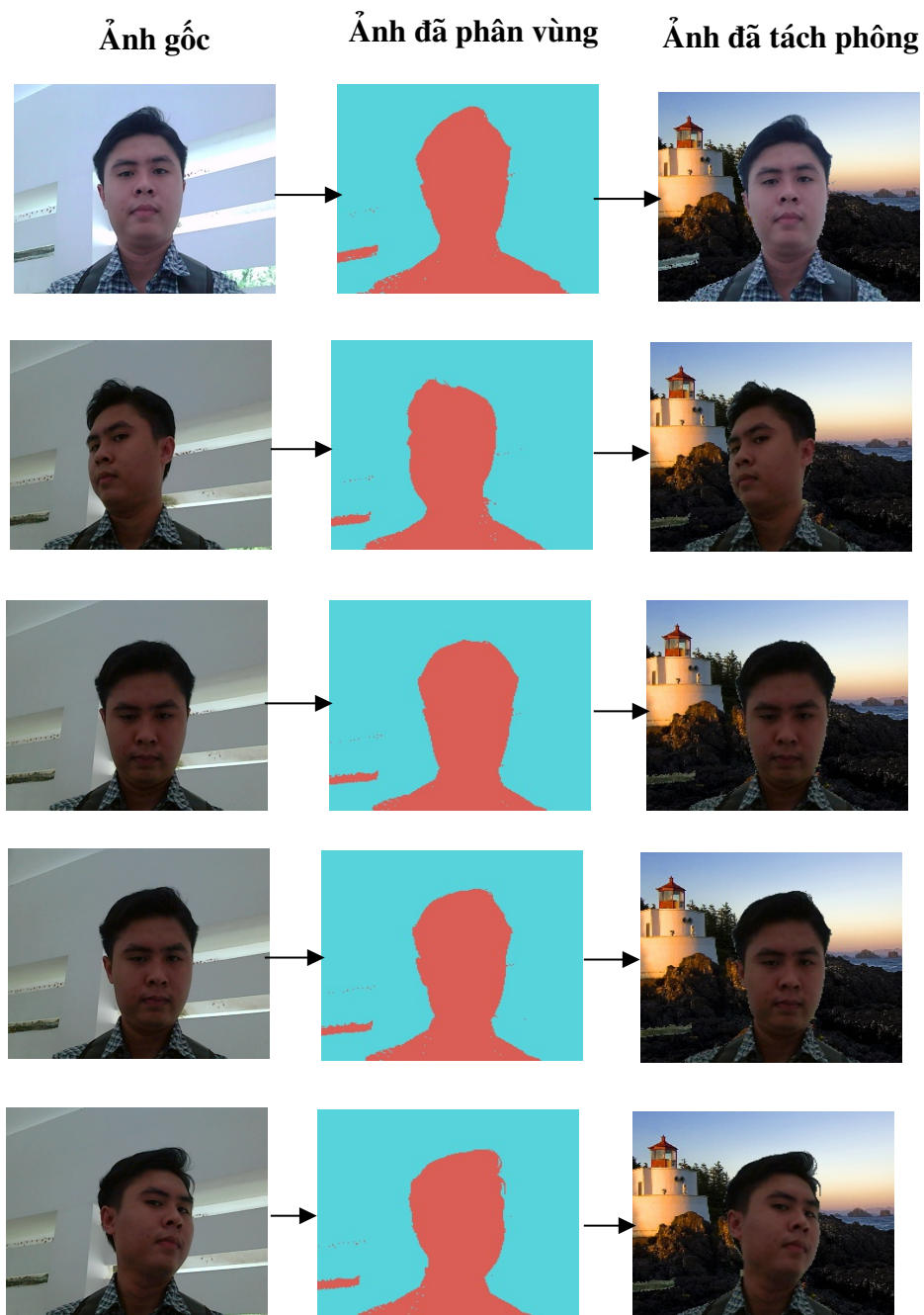


Hình 3.11: Sơ đồ chi tiết hệ thống và hình ảnh

Vì FCN phân vùng các điểm ảnh thành 1 vùng liên tục nhưng CRFs lại căn cứ vào điều kiện trên điểm ảnh gốc để kết luận nhân của vùng liên tục đó nên việc vùng ảnh bị đứt đoạn là vô cùng thường xuyên. Tác giả đề xuất 2 giải pháp, nếu như ảnh đầu ra sau khi thực hiện CRFs liên tục và đồng bộ với nhân thì tiến hành kết hợp với ảnh gốc để tách lấy người trong ảnh gốc, nếu ngược lại thì tinh chỉnh lại độ co giãn (scale) và bước suy luận (inference) hoặc lưu lại kết quả đó thành ảnh, đồng thời xem quá trình trên là kỹ thuật vẽ người bằng điểm ảnh.

3.4.2 Một số kết quả phân vùng ngữ nghĩa:

Kết quả thu được có độ mượt khá tốt và có chất lượng được đảm bảo:



Hình 3.12: Kết quả cuối cùng.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Nhận xét kết quả đạt được:

Sau thời gian nghiên cứu, tìm hiểu về đề tài dưới sự hỗ trợ của thầy cô bạn bè, báo cáo đã hoàn thành các mục tiêu được đặt ra và đạt kết quả tương đối tốt.

Quá trình giảm mẫu của FCN-8s gồm tích chập thuận và thăm dò được tận dụng khá hiệu quả để rút trích đặc trưng của ảnh và giảm mẫu ảnh về dạng xếp tầng. Quá trình tăng mẫu của FCN-8s gồm tích chập chuyển vị và chuyển tiếp nối kết, với kỹ thuật đầu tiên - tích chập chuyển vị được sử dụng để khôi phục ảnh thay cho kỹ thuật tích chập giãn nở (Dilated Convolution) kết hợp với kỹ thuật thứ hai - chuyển tiếp nối kết vẫn đem lại kết quả như mong muốn.

Kết hợp thành công kỹ thuật FCN và sự cải tiến thành công kỹ thuật tối ưu CRFs[8] bằng cách truyền thông điệp trên không gian nhỏ hơn đã tích chập giảm mẫu bằng bộ lọc Gaussian [9] mang lại tốc độ và tiết kiệm thời gian đến không ngờ.

Xây dựng thành công ứng dụng tách phong nền và đối tượng người độc lập bằng Python trên những tấm ảnh độc lập, cho phép thiết đặt số lần tính lại chỉ số KL-divergence, chọn lựa giá trị phân kỳ thấp nhất của 2 xác suất P và Q cũng góp phần đẩy nhanh tốc độ suy luận của bài toán. Ngoài ra, ứng dụng cho phép nạp và lưu ảnh khá thuận tiện cho phép ghép đối tượng người với những ảnh nền khác nhau, phục vụ cho mục đích đồ họa xử lý ảnh kỹ thuật số của người dùng.

Kết quả thu được (F1(tiền cảnh) - MeanIU) trên tập kiểm tra 130 tấm ảnh khi qua từng giai đoạn FCN-8s, CRFs lần lượt là (0.786 - 0.745), (0.838 - 0.800).

4.2 Hạn chế:

Với sự nỗ lực hết mức nhưng luận văn vẫn không thể tránh khỏi những thiếu sót do cả lý do khách quan lẫn chủ quan làm giảm tính chất lượng và trải nghiệm của người dùng.

Tập dữ liệu còn đơn giản khi chỉ tách phong được từng người độc lập bên cạnh đó các yếu tố ánh sáng, chất lượng độ phân giải, phong nền quá phức tạp, đối tượng không phải con người,... vẫn tạo ra sự cản trở cho ứng dụng phân vùng ngữ nghĩa.

Tồn tại tình trạng mất cân bằng giữa các lớp do sự ảnh hưởng của các lớp chiếm tỉ số cao làm cho ảnh thu được không liên tục, không tạo thành một vùng đồng nhất gây khó khăn trong việc tách ảnh, tạo ra những vùng ảnh loang lổ.

Ứng dụng trình bày khá đơn sơ còn chưa gây thu hút và ấn tượng với người dùng vì sử dụng các loại công nghệ lập trình giao diện cơ bản của Python. Ngoài ra, chỉ có thể phân vùng ngữ nghĩa trên từng tấm ảnh, không thể làm trên video hay thời gian thực do cấu hình phần cứng.

4.3 Hướng phát triển:

Hướng phát triển của bài luận văn với rất nhiều ý tưởng và phương pháp cần xét tới nhưng vẫn chú trọng vào 2 điểm chính.

Một là, nghiên cứu cải thiện những khuyết điểm, hạn chế đã nêu bằng cách gia tăng qui mô của tập dữ liệu với nhiều đối tượng, đa dạng số lượng ảnh, bên cạnh đó áp dụng kỹ thuật từ nghiên cứu của Yu Liu [14] chuyên sử dụng cho kỹ thuật DeepLab để cải thiện kỹ thuật đang xét của luận văn, đồng thời cũng sử dụng các loại công nghệ lập trình giao diện ấn tượng hơn để hoàn thiện về mặt thẩm mỹ cho ứng dụng tách phân đồ.

Hai là, ứng dụng ý tưởng kỹ thuật của chúng đi kèm tập dữ liệu chuyên biệt vào những hệ thống đặc biệt phục vụ người dùng như hệ thống thương mại điện tử có tích hợp thử đồ trực tuyến, ứng dụng ghép ảnh trên điện thoại thông minh, hoặc là hệ thống tách phân đồ các thiết bị máy móc khi hoạt động phẫu thuật cơ thể.

TÀI LIỆU THAM KHẢO

- [1] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *PAMI*, 2016. [Online]. Available: <http://arxiv.org/abs/1605.06211>
- [2] *SNOW mobile application*, Camp Mobile, 9 2015, online; accessed 02-April-2019. [Online]. Available: [https://en.wikipedia.org/wiki/Snow_\(app\)](https://en.wikipedia.org/wiki/Snow_(app))
- [3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018. [Online]. Available: <https://doi.org/10.1109/TPAMI.2017.2699184>
- [4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *CoRR*, vol. abs/1511.00561, 2015. [Online]. Available: <http://arxiv.org/abs/1511.00561>
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [6] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: <http://arxiv.org/abs/1703.06870>
- [7] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: <http://arxiv.org/abs/1612.03144>
- [8] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 109–117. [Online]. Available: <http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials.pdf>
- [9] E. Gedraite and M. Hadad, “Investigation on the effect of a gaussian blur in image filtering and segmentation,” 01 2011, pp. 393–396.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, 1998, pp. 2278–2324.
- [12] F. P. Ferrarese, *Image Interpolation Slides*. Verona University, 2010.
- [13] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016. [Online]. Available: <http://arxiv.org/abs/1609.04747>

- [14] Y. Liu and M. S. Lew, “Improving the discrimination between foreground and background for semantic segmentation,” pp. 1272–1276, 09 2017.

PHỤ LỤC: CÀI ĐẶT VÀ SỬ DỤNG CHƯƠNG TRÌNH

1. Cài đặt thư viện:

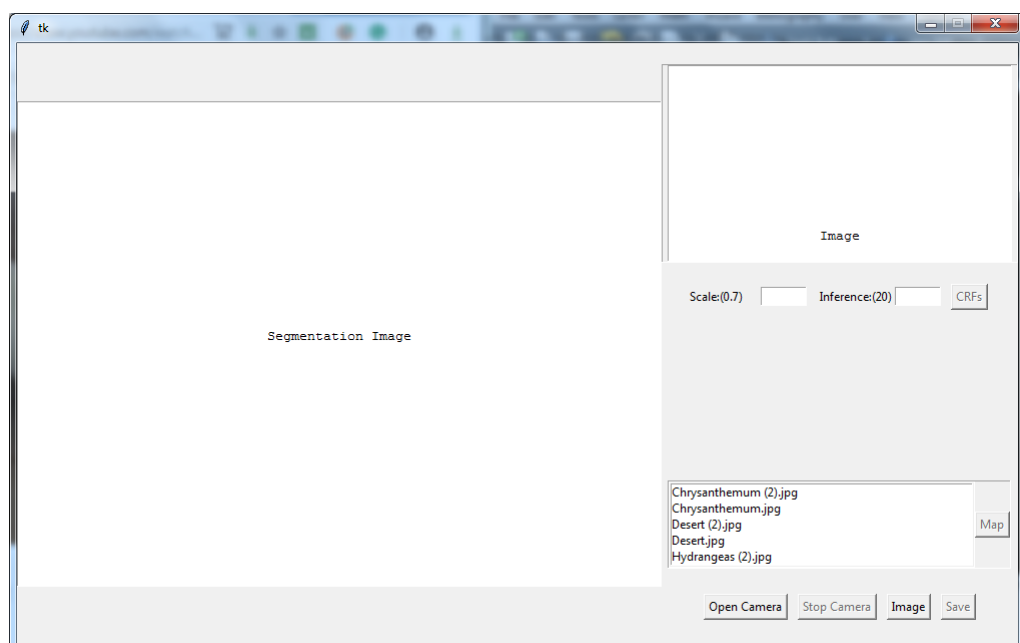
Các thư viện sử dụng đều được tạo trên hệ điều hành Windows 7 và python 3.6 hoặc lớn hơn, tải về qua liên kết <https://www.python.org/> .

- Sử dụng câu lệnh python để biên dịch và chạy một tập tin theo ngôn ngữ python.
- Cài đặt câu lệnh pip và sử dụng chúng để tải các nội dung khác.
- Các câu lệnh trực tiếp được chạy trên cửa sổ dòng lệnh:
 - python -m pip install -U pip setuptools
 - python -m pip install --upgrade matplotlib numpy pandas scipy scikit-learn opencv-python tensorflow keras imutils

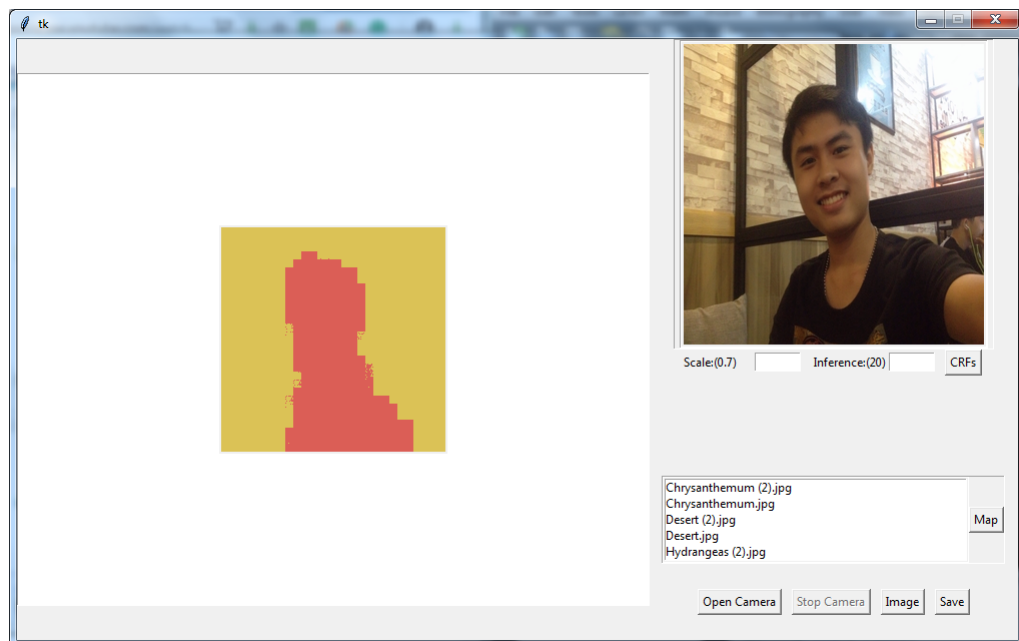
Một trong những lỗi tốn thời gian nhất khi cài đặt thư viện opencv-python trên Windows là "import cv2 ImportError dll load failed the specified module could not be found". Quá trình giải quyết thông thường là bổ sung các phần mềm hỗ trợ Microsoft Visual Studio để biên dịch opencv nhưng nếu vẫn không thể giải quyết thì cần cập nhật bổ sung Windows bằng chính chức năng Windows Update từ Control Panel của hệ điều hành hoặc giảm phiên bản opencv-python sẽ giải quyết vấn đề trên.

2. Sử dụng chương trình:

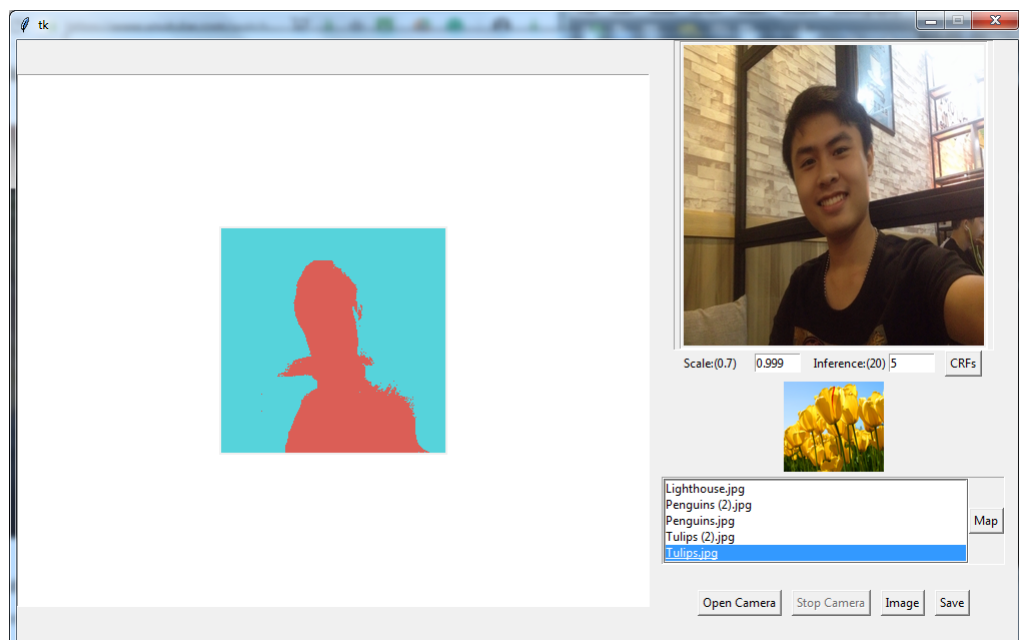
Giao diện mở đầu được hiển thị như sau:



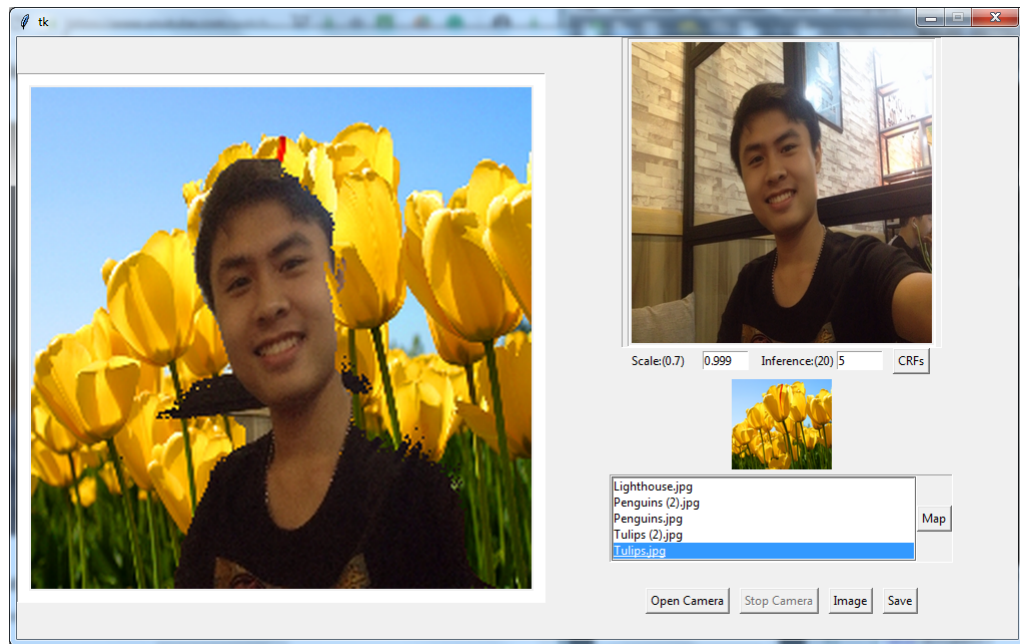
Giao diện khi đã chọn hoặc chụp ảnh:



Giao diện khi đã đặt tham số và tối ưu bằng trường điều kiện ngẫu nhiên:



Giao diện khi đã xóa phông và kết hợp với nền mới:



HẾT.