Thị Giác Máy Tính
Báo cáo đồ án đợt 2:
Object Segmentation

Huỳnh Thiết Gia
20120070@student.hcmus.edu.vn

Đường Yến Ngọc
20120145@student.hcmus.edu.vn

Ngày 30 tháng 6 năm 2023

Tóm tắt nội dung

Bài toán Phân Đoạn Vật Thể Trong Ảnh (Object Segmentation) là một trong những bài toán kinh điển trong lĩnh vực thị giác máy tính, từ những năm 2010 trở đi với sự ra đời của các thuật toán Deep Learning tận dụng sức mạnh của card đồ họa đã tạo ra một bước nhảy vọt về độ chính xác của các giải thuật hàng đầu so với các phương pháp cổ điển. Báo cáo này trình bày quá trình nghiên cứu và thử nghiệm các giải thuật trên máy tính phổ thông của nhóm, với sự chú trọng nhiều hơn đến các giải thuật dùng Deep Learning. Báo cáo lần 2 của nhóm sẽ dừng lại ở bước đề ra hướng giải quyết bài toán.

1 Giới thiệu

1.1 Yêu cầu bài toán

Input: Ảnh chưa được phân đoạn. Output: Ma trận có kích thước tương đương, trong đó mỗi điểm trong ma trận tương ứng với pixel tại tọa đó trong ảnh gốc và giá trị của điểm đó chỉ ra phân loại (class) mà pixel đó thuộc về. Note: Để đơn giản hóa yêu cầu bài toán và nhằm giới hạn lại lĩnh vực nghiên cứu, đối với video ta xem nó như tập hợp các ảnh.

1.2 Lịch sử bài toán

Có rất nhiều phương pháp khác nhau ngoài những thứ được đề cập trong phần này, nhưng vì thời gian nghiên cứu có hạn nên nhóm chỉ tập trung vào những phương pháp nổi tiếng nhất[WALH22].

1.2.1 Object segmentation và object classification

Sự khác biệt giữa hai bài toán trên là Object Classification chỉ yêu cầu ta chỉ ra trong ảnh có một hay nhiều vật nào, còn Object Segmentation cần ta chỉ ra "Vật thể đó xuất hiện ở các pixel nào trong ảnh?", có thể thấy Segmentation là một sự mở rộng của Classification và vài phương pháp giải cho bài toán Segmentation được xây dựng từ bài toán Classification (Xem phần Fully Convolutional Neural Network).

1.2.2 Các phương pháp cổ điển

Từ những năm 1980, việc phân đoạn ảnh đã được sự chú ý của giới nghiên cứu, như Otsu's thresholding [Ots79], với mục tiêu là phân đoạn ảnh sao cho phương sai của các phần con là nhỏ nhất, hay phương pháp phát hiện biên cạnh (Canny Edge Detection)[Can86].
Các phương pháp nhận dạng truyền thống có đặc điểm chung là đều tập trung vào việc trích xuất các đặc trưng bậc thấp như biên cạnh, hình dáng và độ dốc (gradient) trong khu vực. Chúng tiêu tốn ít tài nguyên tính toán và thường không đạt hiệu quả cao. Các phương pháp ấy thường gặp khó khăn trong các khung cảnh phức tạp hoặc các vật thể bị che khuất, và đôi lúc yêu cầu người dùng tối ưu tham số thủ công. Tuy nhiên đây là nền tảng cho các mô hình học sâu (Deep Learning) sau này, đồng thời các phương pháp cổ điển cũng được sử dụng trong các nghiên cứu hiện đại như một thành phần giúp tăng hiệu quả của mô hình.

1.2.3 Fully Convolutional Neural Network

Ý tưởng chính của các phương pháp này là sử dụng lại các mô hình CNN (Convolutional Neural Network), lấy phần trích xuất đặc trưng của các mô hình đó bằng việc loại bỏ các lớp Fully Connected cuối cùng đi, kết quả nhận được là các bản đồ đặc trưng (Feature Map). Với bản đồ đặc trưng đó ta thực hiện các phương pháp nội suy song tuyến (Bilinear Interpolation) hoặc Deconvolution (Là các lớp upsample chứa các tham số có thể học được) để upsample các bản đồ đặc trưng thành bản đồ nhiệt (Heatmap) chứa sự phân lớp các pixel trong ảnh mà vẫn giữ các thông tin không gian (Spatial Information) của ảnh gốc nhờ việc loại bỏ lớp Fully Connected.[WALH22]
Quá trình biến đổi từ ảnh gốc sang các bản đồ đặc trưng bằng lớp Convolution được gọi là Encode (mã hóa), và quá trình biến đổi bản đồ đặc trưng đó thành bản đồ nhiệt phân lớp gọi là Decode (giải mã) Tuy nhiên, phương pháp này chỉ pháp hiện được các đặc trưng cục bộ trong ảnh, và kết quả nó tạo ra có biên cạnh không chính xác vì đã mất đi các thông tin trong quá trình encode-decode.[WALH22]
Model phổ biến nhất của phương pháp này là U-Net, nhờ vào các kết nối giữa các cặp lớp sớm nhất trong encoder với lớp tương đương sau cùng trong decoder để các thông tin bị mất từ các layer encode đầu tiên được truyền tải đến các layer decode cuối cùng, giúp giải quyết vấn đề về sự mất mát thông tin trong các model FCN truyền thống.

1.2.4 Cơ chế Attention

Phương pháp Attention ("Sự chú ý") cho phép model tập trung sự chú ý vào một phần nào đó trong dữ liệu hơn những phần còn lại thông qua việc tính toán trọng số cho các thành phần.

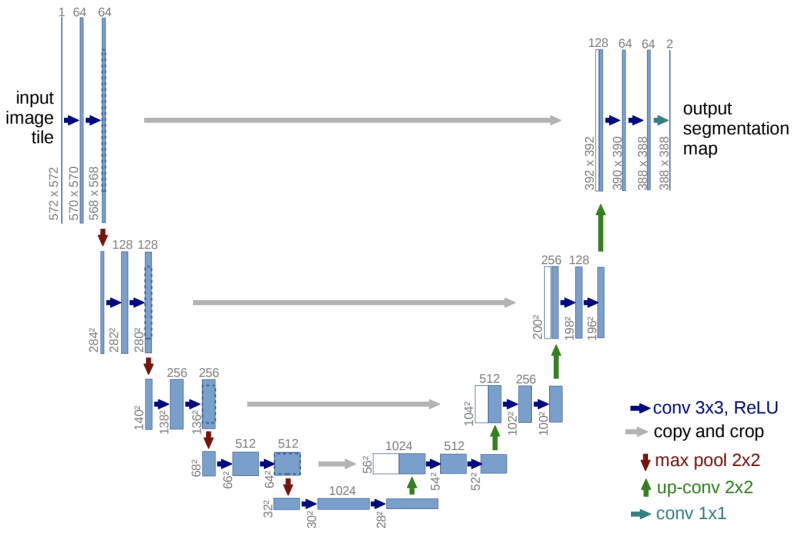


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Hình 1: Cấu trúc mô hình U-Net

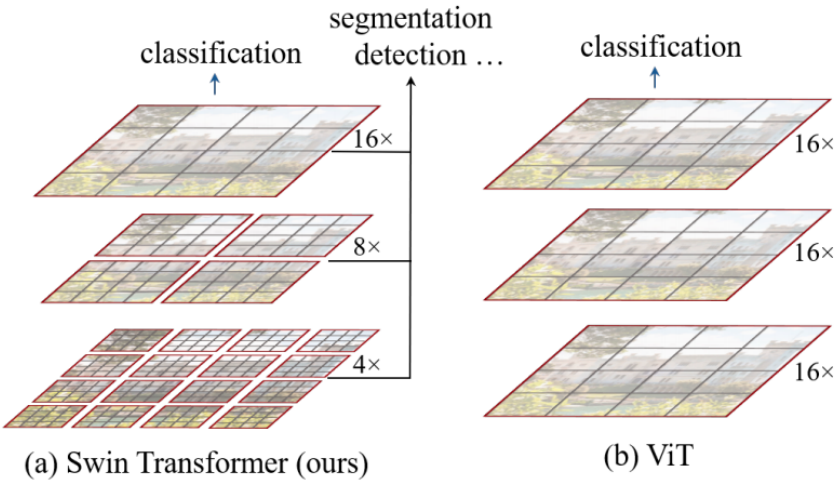
Trước đây các bài báo thường sử dụng cơ chế Attention với các Feature Maps đã được tính toán bởi lớp Convolution, vì mục đích ban đầu của Attention là được sử dụng với đầu vào là chuỗi thông tin, và feature maps được flattened thành chuỗi các feature maps rất phù hợp với mục đích nói trên.
Tuy nhiên ứng dụng gây đột phá nhất là Shifted Windows Transformer (Swin Transformer) khi Liu et al. đã tạo ra một backbone cho phép tính toán các feature maps có thứ bậc. Các feature maps này có thể được dùng để kết hợp với các model khác như FPN hay U-Net. Độ phức tạp thuật toán cũng thấp do nó chỉ tính attention dựa trên các phần (windows) cố định được chia ra từ ảnh gốc. Nhưng vì các layer khác nhau có cách phân vùng khác nhau, nó vẫn cho phép các phần gần nhau tạo ra sự kết nối. Rất nhiều bài báo đạt SOTA trên bộ dataset COCO sử dụng Swin Transformer.

1.3 Dataset

Có rất nhiều Dataset liên quan đến chủ đề này như ImageNet, COCO, CityScape, ADE20K,... Nhưng nhóm chọn bộ dataset COCO vì nó lớn, phổ biến và có thể được dùng để so sánh với kết quả của các model khác. và cũng vì ImageNet chỉ đóng bounding box vào các vật thể chứ không có segment các vật như COCO.
COCO (Microsoft Common Objects in Context): Gồm 123 nghìn ảnh, 91 class (Thực tế có xấp xỉ 80 class), được chia làm 2 tập: Train gồm 118 nghìn ảnh và Test gồm 5 nghìn ảnh. Năm 2022 đã có 1861 bài báo dẫn chứng về bộ dataset này.

1.3.1 Sự thiếu cân bằng của tập COCO

Bộ dữ liệu COCO tuy có số lượng ảnh lớn và số lượng class đa dạng nhưng phân bố của số lượng ảnh trong các class là không cân bằng, với số lượng nhiều nhất lần lượt



Hình 2: Swin Transformer

thuộc về class con người, xe hơi, ghế ngồi,...

1.3.2 Mục tiêu của nhóm với tập Coco

Vì bộ dữ liệu có số lượng class lớn và số lượng ảnh khổng lồ, nhóm chọn tập trung vào N class có số lượng nhiều nhất trong bộ dữ liệu để thu nhỏ tầm vực của nghiên cứu. Nhóm đang thử nghiệm với việc chọn 10 class có số lượng lớn nhất nhưng không phải class "Con người" (Class thứ 2 đến thứ 11), chứa hơn 80 nghìn ảnh, và chọn riêng 1 class con người, chứa hơn 64 nghìn ảnh.

1.4 Phương pháp đánh giá model

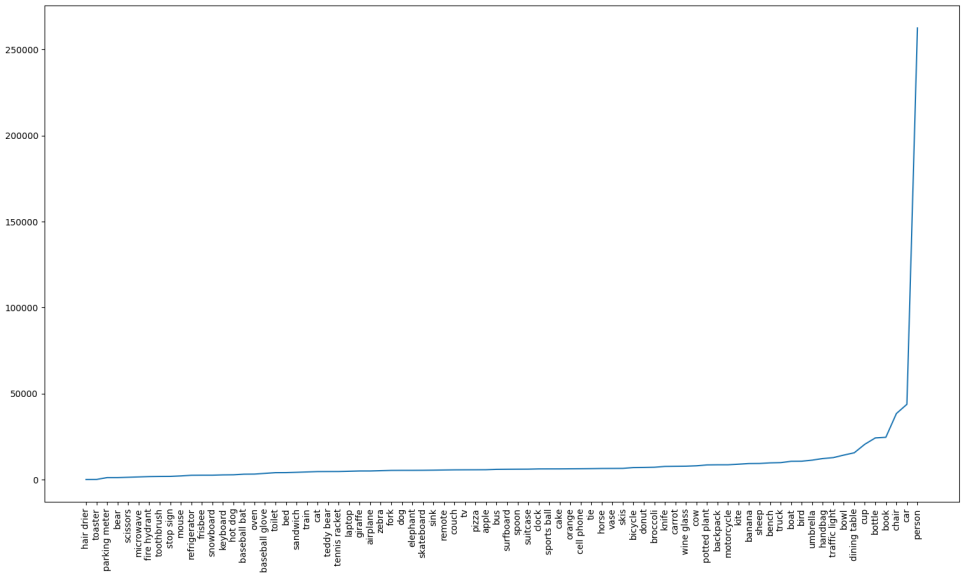
Phương pháp đánh giá chính thường được sử dụng cho các mô hình chạy với bộ dataset COCO là AP[0.5:0.95]. Tuy nhiên vì mục tiêu của nhóm đơn giản hơn nên nhóm chọn phương thức là Binary Cross Entropy Loss kết hợp với IOU (Intersection Over Union).

2 Hướng tiếp cận giải quyết bài toán

Nhóm sẽ ưu tiên các Model Convolution như U-Net vì nó là sự cân bằng giữa độ chính xác và độ đơn giản trong quá trình code và huấn luyện, đồng thời vì độ phổ biến nên các tài liệu về nó rất nhiều.

2.1 Môi trường thực hiện

Toàn bộ quá trình được thực hiện trên máy tính có cấu hình như sau: Cpu Ryzen 5 1600 6 nhân 12 luồng, 24 Gb Ram và 1xGpu Tesla M40 24Gb, sử dụng batch size 44. Không thực hiện thử nghiệm trên Google Colab được vì kích thước bộ dữ liệu quá lớn. Hệ điều hành Windows 10 với ngôn ngữ Python 3.11.
Phiên bản các thư viện mà nhóm sử dụng: Pytorch 2.0.1, Numpy 1.23.5, Matplotlib 3.7.1, OpenCv-python 4.7.0.72



Hình 3: Plot số lượng ảnh trong từng class của tập COCO

2.2 Xử lý dữ liệu đầu vào

Thông qua quá trình thực nghiệm, nhóm nhận thấy quá trình train một Epoch tốn trung bình 3 tiếng, với số lượng ảnh khổng lồ và đa dạng, thêm việc không đảm bảo quá trình augmentation biến đổi ảnh gốc sẽ làm tương tự với bản đồ segment đáp án nên đã không thực hiện biến đổi làm tăng dữ liệu (Data Augmentation).
Chi tiết về vấn đề trên: Các hàm RandomResizedCrop, RandomHorrizontalFlip, RandomColorJitter, RandomRotation của transforms được gọi riêng lẻ trên Input (Ảnh chụp) và Target (Bản đồ Segment), các hàm cũng sử dụng thuật toán tạo số ngẫu nhiên nên mỗi lần gọi sẽ cho ra kết quả khác nhau. Giả sử ta đảm bảo được nó sẽ thực hiện biến đổi giống nhau bằng việc gọi chung biến Transform trên cả hai, ta cũng không để gọi hàm RandomColorJitter trên bản đồ Segment được vì nó có hơn 3 chiều.
Nhóm chọn thu nhỏ chiều nhỏ hơn của ảnh về kích thước 224 cho giống các báo cáo khác, và cắt phần dư của chiều còn lại tại trung tâm.

2.3 Mô hình U-Net

U-Net là mô hình chia làm 2 phân khúc, khúc đầu "thu nhỏ" và khúc sau "phóng to". Nửa đầu rất giống các mô hình Convolutional khác, với mỗi khối gồm các lớp Convolutional kết hợp Relu rồi áp dụng Max Pooling kích thước 2x2 và stride 2, qua mỗi khối như vậy feature maps output có kích thước giảm phân nửa, còn số lượng sẽ được gấp đôi lên. Sau đó nửa sau sử dụng Up-Convolution (ConvTranspose2d) output sẽ là feature maps có kích thước gấp đôi, còn số lượng thì giảm đi phân nửa, các output của khối có độ sâu tương ứng trong nửa đầu sẽ được nối tiếp vào feature maps của các khối trong bước này (Sau khi bị cắt xén đi vì biên bị mất đi một chút trong các khối của nửa sau). [RFB15]. Dùng code gốc của U-Net [Github], nhóm có thể tạo ra được mô hình U-Net có đầu ra tương ứng nhu cầu, tương thích hoàn toàn với PyTorch.

Hình 4: Mô hình dự đoán class "Car"

3 Thực nghiệm

3.1 Kết quả

3.1.1 10 class số lượng lớn nhất

Nhóm ban đầu muốn giới hạn lại, chỉ huấn luyện và kiểm chứng trên 10 class có số lượng lớn nhất, bỏ qua class con người do sự thiếu cân bằng của bộ dữ liệu. (So với mục dưới) Có số lượng ảnh ít hơn (40 nghìn ảnh) quá trình huấn luyện nhanh hơn (1 Epoch tốn 3 tiếng) và validation loss thấp hơn (2.4) nhưng model không đạt được hiệu quả như kỳ vọng, hầu hết trường hợp các ảnh được segment về class "car" nhiều nhất, có lẽ do đây là class có số lượng ảnh nhiều thứ nhì sau "con người".

3.1.2 Con người vs vũ trụ

Nhóm sau đó đã thử chỉ lấy mỗi class "Con người", thay vì giới hạn số lượng ảnh thì chỉ giới hạn số class. Ngược lại hoàn toàn so với trên, số lượng ảnh nhiều hơn (64 nghìn ảnh), quá trình huấn luyện lâu hơn (1 epoch tốn 5 tiếng) và có validation loss cao hơn nhiều (10.2) nhưng model trả về một phần nào đó có thể nhận diện con người, chỉ với 1 epoch.

3.2 Kết luận, phát triển

3.2.1 Chất lượng của mô hình

Nhóm đã không cân nhắc giới hạn về phần cứng của nhóm (1 Gpu M40) nên đã sai lầm khi chọn bộ dữ liệu có kích thước quá lớn, và vì đã tạo mô hình U-Net theo ý nhóm, vì thời gian huấn luyện quá lớn và không có nhiều file U-Net được huấn luyện sẵn trên mạng, và nếu có cũng khó load được vì cấu trúc mô hình khác nhau.

Hình 5: Mô hình dự đoán class "Human"

Ở trạng thái hiện tại, mô hình chỉ cho thấy tiềm năng của mình, cần huấn luyện thêm để có được kết quả thõa mãn.

3.2.2 Cải thiện, phát triển

Huấn luyện nhiều hơn, tuy nhiên một số người dùng khác tốn 2 tuần để huấn luyện với 4 gpu M40.
Đáng lẽ phải thực hiện Image Augmentation trong class DatasetCOCO luôn, nhưng vì nhóm đã lệ thuộc vào Transform của Dataloader và không đủ thời gian nên đã không kịp làm.

Tài liệu

[Can86] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698, 12 1986.

[Ots79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[WALH22] Yuanbo Wang, Unaiza Ahsan, Hanyan Li, and Matthew Hagen. A comprehensive review of modern object segmentation approaches. *Foundations and Trends® in Computer Graphics and Vision*, 13(2-3):111–283, 2022.