CHÚ THÍCH TƯƠNG TÁC HIỆU QUẢ CỦA PHÂN KHÚC DATASETS VỚI ĐA GIÁC-RNN ++

**Tóm tắt**

Ghi nhãn thủ công các bộ dữ liệu với mặt nạ đối tượng là vô cùng tốn thời gian. Trong tác phẩm này, chúng tôi theo ý tưởng của Polygon RNN [4] để tạo ra các chú thích đa giác của các đối tượng tương tác bằng cách sử dụng mô hình đòi hỏi sự tương tác của con người. Chúng tôi giới thiệu một số cải tiến quan trọng cho mô hình: 1) chúng tôi thiết kế kiến ​​trúc bộ mã hóa CNN mới, 2) cho thấy cách đào tạo mô hình hiệu quả với Học tăng cường,và 3) tăng đáng kể độ phân giải đầu ra bằng cách sử dụng Mạng nơ-ron đồ thị, cho phép mô hình chú thích chính xác các đối tượng có độ phân giải cao trong ảnh.

Đánh giá mở rộng trên bộ dữ liệu Cityscapes [8] cho thấy mô hình của chúng tôi, được gọi là Polygon-RNN ++, vượt trội đáng kể so với mô hình ban đầu ở cả hai chế độ tự động (cải thiện 10% tuyệt đối và 16% tương đối trong IoU trung bình) và chế độ tương tác (cần ít hơn 50 % nhấp chuột bởi các chú thích). Chúng tôi phân tích sâu hơn về kịch bản tên miền chéo trong đó mô hình của chúng tôi được đào tạo trên một tập dữ liệu và được sử dụng ngoài hộp trên các bộ dữ liệu từ các miền thay đổi. Kết quả cho thấy Polygon-RNN ++ thể hiện khả năng khái quát hóa mạnh mẽ, đạt được những cải tiến đáng kể so với các phương pháp pixel-wise hiện có. Sử dụng tinh chỉnh trực tuyến đơn giản, chúng tôi tiếp tục giảm được nhiều thời gian chú thích cho các bộ dữ liệu mới, tiến một bước gần hơn tới một công cụ chú thích tương tác sẽ được sử dụng trong thực tế.

1. **Giới thiệu**

Suy đoán chi tiết về các cấu trúc trong hình ảnh là cần thiết cho nhiều ứng dụng thị giác máy tính. Ví dụ, nó rất quan trọng trong lĩnh vực lái xe tự động để nội địa hóa và phác thảo tất cả xe ô tô, người đi bộ và đối tượng tĩnh và động linh tinh [1, 19, 12]. Để ánh xạ, có là cần thiết để có được dấu chân chi tiết của các tòa nhà và đường từ hình ảnh trên không / vệ tinh [35], trong khi lĩnh vực y tế / chăm sóc sức khỏe yêu cầu phương thức tự động để phác thảo chính xác các tế bào, mô và các cấu trúc liên quan khác [16, 11]. Mạng lưới thần kinh đã được chứng minh là một cách hiệu quả để suy ra ngữ nghĩa [6, 20] và thông tin phân đoạn đối tượng [12, 19] trong hình ảnh đầy thách thức. Người ta biết số lượng và sự đa dạng của dữ liệu mà các mạng nhìn thấy trong quá trình đào tạo ảnh hưởng mạnh mẽ đến hiệu suất của chúng trong thời gian chạy.. Tuy nhiên, thu thập mặt nạ ví dụ thực địa là một công việc cực kỳ tốn thời gian, thường đòi hỏi các chú thích của con người phải dành 20-30 giây cho mỗi đối tượng trong một hình ảnh.

Cuối cùng, trong [4], các tác giả đã giới thiệu Polygon-RNN, một mô hình khái niệm cho ghi nhãn bán tự động và tương tác để giúp tăng tốc chú thích đối tượng. Thay vì tạo phân đoạn pixel-wise của một đối tượng của một đối tượng như trong các công cụ tương tác hiện có như Grabcut [31], [4] dự đoán các đỉnh của một đa giác phác thảo đối tượng. Lợi ích của việc sử dụng biểu diễn đa giác gấp ba lần, 1) rất thưa thớt chỉ một vài đỉnh biểu thị các vùng có số lượng pixel lớn), 2) dễ dàng cho một bộ chú thích tương tác và 3) nó cho phép tương tác hiệu quả, thường chỉ cần một vài chỉnh sửa từ bộ chú thích [4]. Sử dụng mô hình của họ, các tác giả đã cho thấy tốc độ tăng tốc chú thích cao trên hai bộ dữ liệu lái xe tự trị [8, 10].

Trong công việc này, chúng tôi giới thiệu một số cải tiến cho mô hình đa giác-RNN. Đặc biệt, chúng tôi 1) làm cho một vài thay đổi kiến ​​trúc mạng thần kinh, 2) đề xuất một thuật toán học tập tốt hơn để đào tạo mô hình bằng cách sử dụng học tăng cường, và 3) chỉ ra cách tăng đáng kể độ phân giải đầu ra của đa giác (một trong những hạn chế chính của mô hình ban đầu) bằng cách sử dụng Mạng lưới thần kinh đồ thị [32, 18]. Chúng tôi phân tích sự mạnh mẽ của phương pháp tiếp cận nhiễu và khả năng khái quát hóa của nó đối với hình ảnh ngoài miền.

Trong chế độ hoàn toàn tự động (không có chú thích trong vòng lặp), mô hình của chúng tôi đạt được những cải tiến đáng kể so với phương pháp tiếp cận gốc,vượt trội hơn 10% so với IoU trên bộ dữ liệu Cityscapes [8]. In interactive mode, our approach requires 50% fewer clicks as compared to [4]. Để thể hiện khái quát hóa, chúng tôi sử dụng một mô hình được đào tạo trên bộ dữ liệu Cityscapes để chú thích một tập hợp con của bộ dữ liệu phân tích cảnh [43], hình ảnh trên không [34] và hai bộ dữ liệu y tế [16, 11].Mô hình vượt trội đáng kể so với đường cơ sở ghi nhãn pixel-wise mạnh mẽ, cho thấy rằng nó vốn đã học cách tuân theo các ranh giới đối tượng, do đó khái quát hóa tốt hơn. Chúng tôi tiếp tục chỉ ra rằng một cách tiếp cận hoàn thiện trực tuyến đơn giản đạt được tốc độ tăng tốc chú thích cao trên chú thích dữ liệu ngoài miền.

1. **Công việc liên quan**

**Chú thích tương tác.** Kể từ khi phân đoạn đối tượng tốn thời gian để chú thích bằng tay, một số tác phẩm đã nhằm mục đích đẩy nhanh quá trình này bằng cách sử dụng các kỹ thuật tương tác. Trong công việc tinh dịch, [2] đã sử dụng các nét vẽ nguệch ngoạc để mô hình hóa sự xuất hiện của tiền cảnh / hậu cảnh và thực hiện đề cập phân tách thông qua các biểu đồ cắt [3]. Ý tưởng này đã được mở rộng bởi [21] để sử dụng nhiều nét vẽ nguệch ngoạc trên cả đối tượng và nền, và đã được thể hiện trong việc chú thích các đối tượng trong video. GrabCut [31] đã khai thác các hộp giới hạn 2D được cung cấp bởi trình thông báo, và thực hiện ghi nhãn nền trước / nền sau pixel-wise bằng EM.[26] kết hợp GrabCut với CNN để chú thích các cấu trúc trong hình ảnh y tế. Hầu hết các tác phẩm này hoạt động ở cấp độ pixel và thường gặp khó khăn trong trường hợp nền trước và nền sau có màu tương tự nhau.

Trong [4], các tác giả đã sử dụng đa giác thay thế. Sức mạnh chính của việc sử dụng một đại diện như vậy làm nó thưa thớt; chỉ một vài đỉnh của đa giác thể hiện các vùng ảnh lớn. Điều này giúp người dùng dễ dàng đưa ra các chỉnh sửa, bằng cách đơn giản là di chuyển các đỉnh sai. Một RNN cũng có hiệu quả nắm bắt các hình dạng điển hình của các đối tượng vì nó tạo thành một biểu diễn tuần tự phi tuyến tính của hình dạng. Điều này đặc biệt quan trọng trong các khu vực mơ hồ, tức là bóng tối và bão hòa, nơi không thể quan sát được các ranh giới.Chúng tôi theo dòng công việc này, và giới thiệu một số sửa đổi quan trọng đối với kiến ​​trúc và đào tạo. Hơn nữa, mô hình ban đầu chỉ có thể đưa ra dự đoán ở độ phân giải thấp (28 × 28), do đó tạo ra các đa giác khối cho các đối tượng lớn. Mô hình của chúng tôi tăng đáng kể độ phân giải đầu ra (112 × 112).

**Phân đoạn đối tượng.** Hầu hết các cách tiếp cận để phân đoạn đối tượng [17, 30, 41, 39, 22, 23, 12, 1, 19] hoạt động ở cấp độ pixel. Nhiều người dựa vào phát hiện đối tượng,và sử dụng một mạng lưới trên một đề xuất hộp để thực hiện ghi nhãn [22, 23, 12]. Trong [40, 34], các tác giả tạo ra một đa giác xung quanh một đối tượng.Những cách tiếp cận này trước tiên phát hiện các đoạn ranh giới, tiếp theo là tìm một chu kỳ tối ưu liên kết các ranh giới thành các vùng đối tượng.[9] tạo ra các siêu phẩm dưới dạng đa giác nhỏ được kết hợp thành một vật thể.Ở đây, như trong [4] chúng tôi sử dụng các mạng thần kinh để tạo ra các đa giác, và đặc biệt giải quyết kịch bản ghi nhãn tương tác chưa được khám phá trong các tác phẩm này.

1. **Polygon-RNN++**

Trong phần này, chúng tôi giới thiệu Polygon-RNN ++. Theo [4], mô hình của chúng tôi hy vọng một bộ chú thích sẽ cung cấp một hộp xung quanh đối tượng quan tâm. Chúng tôi trích xuất một hình ảnh được bao quanh bởi hộp phóng to 15%. Chúng tôi sử dụng kiến trúc CNN + RNN như trong [4], với CNN đóng vai trò là trình trích xuất tính năng hình ảnh và RNN giải mã một đỉnh đa giác tại một thời điểm. Các đỉnh đầu ra được biểu diễn dưới dạng các vị trí trong một lưới.

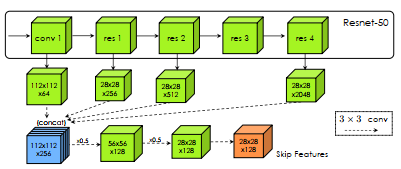
Mô hình đầy đủ được mô tả trong hình 2. Bộ mã hóa được thiết kế lại của chúng tôi tạo ra các tính năng hình ảnh được sử dụng để dự đoán đỉnh đầu tiên. Đỉnh đầu tiên và các tính năng hình ảnh sau đó được đưa đến bộ giải mã định kỳ. RNN của chúng tôi khai thác sự chú ý trực quan tại mỗi bước thời gian để tạo ra các đỉnh đa giác. Một mạng đánh giá đã học chọn đa giác tốt nhất từ một tập hợp các ứng cử viên được đề xuất bởi bộ giải mã. Cuối cùng, một mạng nơ ron đồ thị điều chỉnh lại các đa giác, được tăng thêm các đỉnh bổ sung, ở độ phân giải cao hơn.

Mô hình này tự nhiên kết hợp một mô hình đòi hỏi sự tương tác của con người cho phép bộ chú thích sửa một đỉnh được dự đoán sai. Đỉnh này sau đó được đưa trở lại mô hình,giúp mô hình sửa lỗi dự đoán trong bước tiếp theo.

* 1. **Bộ mã hóa dư với kết nối bỏ qua**

Hầu hết các mạng thực hiện các hoạt động lấy mẫu xuống lặp đi lặp lại ở các lớp liên tiếp của CNN, điều này ảnh hưởng đến độ phân giải đầu ra hiệu quả trong các tác vụ như phân tách hình ảnh [6, 24].Để giảm bớt vấn đề này, chúng tôi theo dõi [7] và sửa đổi kiến ​​trúc ResNet-50 [13] bằng cách giảm bước tiến của mạng và giới thiệu các yếu tố giãn nở.Điều này cho phép chúng tôi tăng độ phân giải của bản đồ tính năng đầu ra mà không làm giảm trường tiếp nhận của từng nơ-ron riêng lẻ. Chúng tôi cũng loại bỏ các lớp pooling trung bình và các lớp FC.

Chúng tôi tiếp tục thêm một kiến ​​trúc bỏ qua [20, 42] nhằm mục đích nắm bắt cả hai, các chi tiết cấp thấp như các cạnh và lõi, cũng như thông tin ngữ nghĩa cấp cao.Trong [4], các tác giả thực hiện lấy mẫu xuống trong kiến ​​trúc lớp bỏ qua, được xây dựng trên đỉnh VAG, trước khi ghép các tính năng từ các lớp khác nhau.Thay vào đó, chúng tôi ghép tất cả các đầu ra của các lớp bỏ qua ở độ phân giải cao nhất có thể,và sử dụng kết hợp các lớp đối lưu và các hoạt động gộp tối đa để có được bản đồ tính năng cuối cùng.Chúng tôi sử dụng các bộ lọc đối lưu với kích thước hạt nhân là 3 × 3, chuẩn hóa hàng loạt [15] và phi tuyến tính ReLU. Trong trường hợp các kết nối bỏ qua có kích thước không gian khác nhau, chúng tôi sử dụng song tuyến nâng cấp trước khi gep. Kiến trúc được hiển thị trong Hình 4. Chúng tôi gọi bản đồ tính năng cuối cùng là các tính năng bỏ qua.



Hình 4: Kiến trúc mã hóa dư. Tenor màu xanh được đưa đến GNN, trong khi tenxơ màu cam được đưa vào bộ giải mã RNN.

* 1. **Bộ giải mã định kỳ**

Như trong [4], chúng tôi sử dụng Mạng thần kinh tái phát để mô hình hóa chuỗi các đỉnh 2D của đa giác phác thảo một đối tượng. Phù hợp với công việc trước đây, chúng tôi cũng nhận thấy rằng việc sử dụng LSTM Convolutional [37] là điều cần thiết: 1) để lưu giữ thông tin không gian và 2) để giảm số lượng tham số cần học.Trong RNN của chúng tôi, chúng tôi thêm một cơ chế chú ý, cũng như dự đoán đỉnh đầu tiên trong cùng một mạng (không giống như [4] có hai mạng riêng biệt).

Chúng tôi sử dụng ConvLTSM hai lớp với nhân 3 × 3 với 64 và 16 kênh tương ứng.Chúng tôi áp dụng định mức lô [15] tại mỗi bước thời gian, mà không chia sẻ ước tính trung bình / phương sai theo các bước thời gian. Chúng tôi đặt đầu ra của chúng tôi tại thời điểm bước t là một one-hot mã hóa của (*D × D*) + 1 yếu tố. Trong đó D là độ phân giải mà chúng tôi dự đoán. Trong các thí nghiệm của chúng tôi, D được đặt thành 28. Đầu tiên Kích thước D × D đại diện cho các vị trí đỉnh có thể có và kích thước cuối cùng tương ứng với mã thông báo cuối seq báo hiệu rằng đa giác được đóng.

**Các tính năng có trọng số chú ý:** Trong RNN, chúng tôi khai thác một cơ chế gần giống với sự chú ý.Cụ thể, tại thời điểm bước t, chúng tôi tính toán bản đồ tính năng có trọng số là,



Trong đó ◦ là sản phẩm Hadamard, x là tính năng bỏ qua, và *h*1*,t*, *h*2*,t* là các tensor trạng thái ẩn từ hai lớp ConvLSTM. Ở đây, f1 và f2 ánh xạh 1 , t và h 2 , t đến RD×D×128 bằng một lớp được kết nối đầy đủ. *fatt* lấy tổng số đầu vào của nó và ánh xạ nó tới D × D thông qua một lớp được kết nối đầy đủ, tạo ra một trọng lượng của một chú ý trên mỗi vị trí.

Theo trực giác, chúng tôi sử dụng trạng thái ẩn RNN trước đó để chuyển các vị trí nhất định trong bản đồ tính năng hình ảnh,cho phép RNN chỉ tập trung vào các thông tin liên quan trong bước tiếp theo.Bản đồ tính năng gated Ft sau đó được nối với các bảng mã một nóng của hai đỉnh trước đó yt - 1; yt − 2 và đỉnh đầu tiên y0, và được chuyển đến RNN tại bước thời gian t.

**Đỉnh đầu tiên:** Cho một đỉnh trước và một phép di chuyển ngầm, đỉnh tiếp theo của đa giác luôn được xác định duy nhất, ngoại trừ đỉnh đầu tiên. Để giải quyết vấn đề này, các tác giả trong [4] đã coi đỉnh đầu tiên là trường hợp đặc biệt và sử dụng một kiến ​​trúc bổ sung (được đào tạo riêng) để dự đoán nó. Trong mô hình của chúng tôi, chúng tôi thêm một nhánh khác từ kiến ​​trúc lớp bỏ qua, tạo thành hai lớp mỗi chiều D × D. Theo [4], lớp đầu tiên dự đoán các cạnh, trong khi lớp thứ hai dự đoán các đỉnh của đa giác. Tại thời điểm thử nghiệm, đỉnh đầu tiên được lấy mẫu từ lớp cuối cùng của nhánh này.

* 1. **Đào tạo sử dụng học tăng cường**

Trong [4], các tác giả đã đào tạo mô hình bằng cách sử dụng tổn thất entropy chéo ở mỗi bước thời gian. Tuy nhiên,việc đào tạo như vậy có hai hạn chế chính: 1) MLE xử phạt quá mức mô hình (ví dụ: khi đỉnh được dự đoán nằm trên cạnh của đa giác GT nhưng không phải là một trong các đỉnh GT),và 2) nó tối ưu hóa một số liệu rất khác so với số liệu đánh giá cuối cùng (tức là IoU). Hơn nữa, mô hình trong [4] đã được đào tạo theo chế độ đào tạo điển hình trong đó đỉnh GT được đưa đến bước tiếp theo thay vì dự đoán mô hình. Chế độ đào tạo này, được gọi là cưỡng bức giáo viên tạo ra sự không phù hợp giữa đào tạo và kiểm tra được gọi là vấn đề sai lệch phơi nhiễm [27].

Để giảm thiểu những vấn đề này, chúng tôi chỉ sử dụng đào tạo MLE làm giai đoạn khởi tạo. Sau đó, chúng tôi định dạng lại nhiệm vụ dự đoán đa giác như một vấn đề học tập củng cố và tinh chỉnh mạng bằng RL. Trong giai đoạn này, chúng tôi cho phép mạng khám phá các chính sách tối ưu hóa mong muốn, nhưng số liệu đánh giá không khác biệt (IoU) trong khi cũng đưa ra dự đoán đó trong quá trình đào tạo.

* + 1. *Xây dựng vấn đề*

Chúng tôi xem bộ giải mã định kỳ của chúng tôi như là một tác nhân quyết định tuần tự. Các tham số *θ* của kiến ​​trúc bộ mã hóa-giải mã của chúng tôi xác định chính sách p*θ*của nó để chọn đỉnh v*t* tiếp theo. Vào cuối chuỗi, chúng tôi nhận được phần thưởng r. Chúng tôi tính toán phần thưởng của chúng tôi là IoU giữa mặt nạ được bao bọc bởi đa giác được dựng lên và mặt nạ chân thực m.Để tối đa hóa phần thưởng dự kiến,hàm mất mát của chúng tôi trở thành:



trong đó *vs* = (*vs1;…; vTs*) và *vts* là đỉnh được lấy mẫu từ mô hình tại thời điểm *t.* Ở đây, *r* = IoU (*mask(vs); m*).

* + 1. *Đào tạo tự phê bình với chính sách độ dốc*

Sử dụng thủ thuật REINFORCE [36] để tính toán độ dốc của kỳ vọng,chúng ta có:



Trong thực tế, độ dốc dự kiến ​​được tính bằng cách sử dụng lấy mẫu Monte-Carlo đơn giản với một mẫu duy nhất. Thủ tục này được biết là thể hiện phương sai cao và rất không ổn định nếu không chuẩn hóa phụ thuộc vào ngữ cảnh thích hợp. Một cách tự nhiên để đối phó với điều này là sử dụng đường cơ sở đã học được trừ vào phần thưởng. Trong công việc này, chúng tôi thực hiện theo phương pháp tự phê bình [29] và sử dụng phần thưởng suy luận thời gian thử nghiệm của mô hình của chúng tôi làm cơ sở.Theo đó, chúng tôi điều chỉnh độ dốc của hàm mất mát:



Trong đó  là phần thưởng mà mô hình đạt được bằng cách sử dụng giải mã tham lam. Để kiểm soát mức độ ngẫu nhiên trong các đỉnh được mô hình khám phá, chúng tôi giới thiệu một tham số nhiệt độ τ trong phần mềm của chính sách. Điều này đảm bảo rằng các đỉnh được lấy mẫu dẫn đến đa giác hoạt động tốt. Trong các thử nghiệm của chúng tôi, chúng tôi đặt τ = 0,6.

* 1. **Mạng lưới đánh giá**

Sự lựa chọn thông minh của đỉnh đầu tiên là rất quan trọng vì nó ảnh hưởng đến các dự đoán ban đầu của RNN, khi mô hình không có lịch sử mạnh mẽ để lý giải về đối tượng cần chú thích. Điều này đặc biệt quan trọng trong các trường hợp che khuất các đối tượng. Điều mong muốn là đỉnh đầu tiên nằm cách xa ranh giới tắc để mô hình đi theo đối tượng quan tâm. Trong RNN, tìm kiếm chùm tia thường được sử dụng để cắt xén các chuỗi không thể thực hiện được (chẳng hạn như khi mô hình bắt đầu theo dõi một đối tượng ẩn). Tuy nhiên, vì tìm kiếm chùm tia cổ điển sử dụng xác suất nhật ký để đánh giá chùm tia, nên nó không trực tiếp áp dụng cho mô hình của chúng tôi nhằm mục đích tối ưu hóa IoU. Một điểm trên ranh giới tắc thường thể hiện một cạnh mạnh và do đó sẽ có xác suất log cao trong quá trình dự đoán, làm giảm khả năng nó bị cắt xén bởi tìm kiếm chùm tia.

Để giải quyết vấn đề này, chúng tôi đề xuất sử dụng mạng đánh giá tại thời điểm suy luận, nhằm mục đích lựa chọn hiệu quả giữa nhiều đa giác ứng cử viên. Mạng đánh giá của chúng tôi lấy đầu vào là các tính năng bỏ qua, tensor trạng thái cuối cùng của ConvLSTM và đa giác dự đoán và cố gắng ước tính chất lượng của nó bằng cách dự đoán IoU của nó với GT. Mạng có hai lớp chập 3 × 3 theo sau là lớp FC, tạo thành một nhánh khác trong mô hình. Hình 3 mô tả kiến trúc của nó. Mặc dù mô hình đầy đủ có thể được đào tạo từ đầu đến cuối trong bước RL, chúng tôi chọn đào tạo mạng đánh giá riêng sau khi tinh chỉnh RL đã hội tụ.

Trong quá trình huấn luyện,chúng tôi giảm thiểu lỗi bình phương trung bình:



Trong đó *p* là mạng lưới dự đoán IoU, mvs là mặt nạ cho các đỉnh được lấy mẫu và m là mặt nạ chân thực. Để đảm bảo tính đa dạng trong các đỉnh nhìn thấy, chúng tôi lấy mẫu đa giác với *τ* = 0,3. Chúng tôi nhấn mạnh rằng chúng tôi không sử dụng mạng này làm công cụ ước tính cơ bản trong bước đào tạo RL vì chúng tôi thấy rằng phương pháp tự phê bình tạo ra kết quả tốt hơn.

Suy luận: Tại thời điểm kiểm tra, chúng tôi đưa ra K dự đoán đỉnh đầu tiên. Đối với mỗi loại này, chúng tôi tạo ra các đa giác thông qua tìm kiếm chùm tia cổ điển (sử dụng đầu dò log với độ rộng B). Điều này mang lại K đa giác khác nhau, một cho mỗi ứng cử viên đỉnh đầu tiên. Chúng tôi sử dụng mạng đánh giá để chọn đa giác tốt nhất. Trong các thí nghiệm của chúng tôi, chúng tôi sử dụng K = 5. Trong khi người ta có thể sử dụng mạng đánh giá thay vì tìm kiếm chùm tia tại mỗi bước thời gian, điều này sẽ dẫn đến thời gian suy luận dài vô tận. Mô hình đầy đủ nhanh hơn của chúng tôi (sử dụng B = K = 1) chạy ở tốc độ 295ms cho mỗi phiên bản đối tượng trên Titan XP.

**Chú thích trong Vòng lặp:** Chúng tôi theo cùng một giao thức như trong [4], trong đó chú thích sửa các đỉnh theo thứ tự liên tiếp. Mỗi hiệu chỉnh sau đó được đưa trở lại mô hình, dự đoán lại phần còn lại của đa giác.

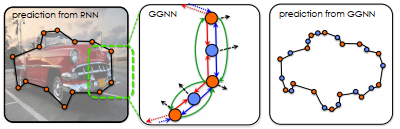
* 1. **Nâng cấp với mạng nơ ron đồ thị**

Mô hình được mô tả ở trên tạo ra các đa giác với độ phân giải D × D, trong đó chúng ta đặt D là 28 để đáp ứng giới hạn bộ nhớ và để giữ cho độ chính xác của không gian đầu ra có thể chấp nhận được. Trong phần này, chúng tôi khai thác Mạng thần kinh đồ thị Gated (GGNN) [18], để tạo đa giác ở độ phân giải cao hơn nhiều. GNN đã được chứng minh hiệu quả cho phân đoạn ngữ nghĩa [25], trong đó nó được sử dụng ở cấp độ pixel.

Lưu ý rằng khi đào tạo bộ giải mã RNN, các gon GT được đơn giản hóa ở độ phân giải mục tiêu của chúng (các đỉnh đồng tuyến được loại bỏ) để giảm bớt sự mơ hồ của nhiệm vụ dự đoán. Do đó, ở độ phân giải cao hơn, đối tượng có thể có các đỉnh bổ sung, do đó thay đổi cấu trúc liên kết của đa giác.

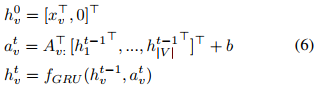
Mô hình nâng cấp của chúng tôi lấy đầu vào là chuỗi các đỉnh được tạo bởi bộ giải mã RNN. Chúng tôi coi các đỉnh này là các nút trong biểu đồ. Để mô hình chi tiết tốt hơn ở độ phân giải cao hơn, chúng tôi thêm một nút ở giữa hai nút liên tiếp, với vị trí của nó nằm ở giữa cạnh tương ứng của chúng. Chúng tôi cũng kết nối đỉnh cuối cùng và đỉnh đầu tiên, có hiệu quả chuyển hướng chuỗi thành một chu kỳ. Chúng tôi kết nối các nút lân cận bằng 3 loại cạnh khác nhau, như trong Hình.5.

GGNN định nghĩa một mô hình lan truyền mở rộng RNN thành các biểu đồ tùy ý, truyền thông tin có hiệu quả là mười hai nút, trước khi tạo ra một đầu ra tại mỗi nút. Ở đây, GGNN định nghĩa một mô hình lan truyền mở rộng RNN thành các biểu đồ tùy ý, truyền thông tin có hiệu quả là mười hai nút, trước khi tạo ra một đầu ra tại mỗi nút.



Hình 5: Mô hình GGNN: Chúng tôi lấy đa giác dự đoán từ RNN (các đỉnh màu cam) và thêm trung điểm (màu xanh lam) giữa mỗi cặp đỉnh liên tiếp (màu cam).GGNN của chúng tôi có ba loại cạnh (đỏ, xanh dương, xanh lục), mỗi loại có trọng lượng riêng để truyền thông điệp.Mũi tên nét đứt màu đen chỉ ra khỏi các nút (sơ đồ giữa) cho thấy GGNN nhằm mục đích dự đoán vị trí tương đối cho mỗi nút (đỉnh),sau khi hoàn thành nhân giống. Bên phải là đầu ra đa giác có độ phân giải cao của GGNN.

**Mạng lưới thần kinh Gated:** Để hoàn thiện, chúng tôi tóm tắt ngắn gọn mô hình GGNN [18]. GGNN sử dụng biểu đồ *{V, E}*; Ví dụ, trong đó *V* và *E* lần lượt là tập hợp các nút và cạnh. Nó bao gồm một mô hình lan truyền thực hiện thông điệp truyền trong biểu đồ và một mô hình đầu ra cho các nhiệm vụ dự đoán. Chúng tôi biểu thị trạng thái ban đầu của một nút *v* là *xv* và trạng thái ẩn của nút *v* tại thời điểm bước *t* là ht*v*. Sự tái phát cơ bản của mô hình nhân giống là:



Ma trận  xác định cách các nút trong đồ thị giao tiếp với nhau, trong đó *N* đại diện cho số loại cạnh khác nhau. Tin nhắn được tuyên truyền cho các bước *T*. Đầu ra cho nút *v* sau đó được định nghĩa là:



Ở đây, *f1* và *f2* là MLP và *outv* là đầu ra mong muốn của *v*.

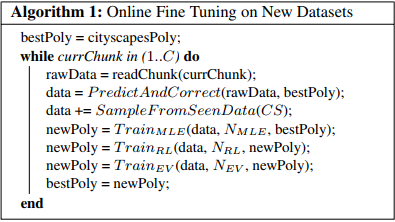
**PolygonRNN ++ với GGNN:** Để có được các quan sát cho mô hình GGNN của chúng tôi, chúng tôi thêm một nhánh khác lên trên kiến ​​trúc lớp bỏ qua của chúng tôi, cụ thể, từ bản đồ đặc trưng 112 × 112 × 256 (được đánh dấu màu xanh lam trong Hình 4). Chúng tôi khai thác một lớp đối lưu với 256 bộ lọc có kích thước 15 × 15, cung cấp cho chúng tôi bản đồ tính năng có kích thước 112 × 112 × 256. Đối với mỗi nút *v* trong biểu đồ, chúng tôi trích xuất một bản vá *S × S* xung quanh vị trí tỷ lệ *(vx;vy)*, cho chúng ta vectơ quan sát *xv*. Sau khi lan truyền, chúng tôi dự đoán đầu ra của một nút *v* là một vị trí trong lưới không gian *D’ × D’*. Chúng tôi tạo lưới này liên quan đến vị trí *(vx;vy)*, khiến nhiệm vụ dự đoán trở thành một sự dịch chuyển tương đối so với vị trí ban đầu của nó. Dự đoán này được coi là một nhiệm vụ phân loại và mô hình được đào tạo với tổn thất entropy chéo. Đặc biệt, để đào tạo mô hình của chúng tôi, trước tiên chúng tôi lấy dự đoán từ bộ giải mã RNN, và sửa một dự đoán sai nếu nó lệch khỏi đỉnh thực địa hơn một ngưỡng. Các mục tiêu để đào tạo GGNN của chúng ta sau đó là các chuyển vị tương đối của từng đỉnh này đối với các đỉnh thực địa tương ứng của chúng.

**Chi tiết triển khai:** Chúng tôi đặt *S* thành 1 và *D’* thành 112. Mặc dù mô hình của chúng tôi hỗ trợ độ phân giải đầu ra cao hơn nhiều, chúng tôi thấy rằng *D’* lớn hơn không cải thiện kết quả. Trạng thái ẩn của GRU trong GGNN có 256 kích thước. Chúng tôi sử dụng *T* = 5 bước lan truyền. Trong mô hình đầu ra, *f1* là lớp FC 256 × 256 và *f2* là 256 × 15 × 15 MLP. Trong đào tạo, chúng tôi lấy các dự đoán từ RNN và thay thế các đỉnh bằng các đỉnh GT nếu chúng lệch hơn 3 ô.

* 1. **Chú thích. Tên miền mới thông qua tinh chỉnh trực tuyến**

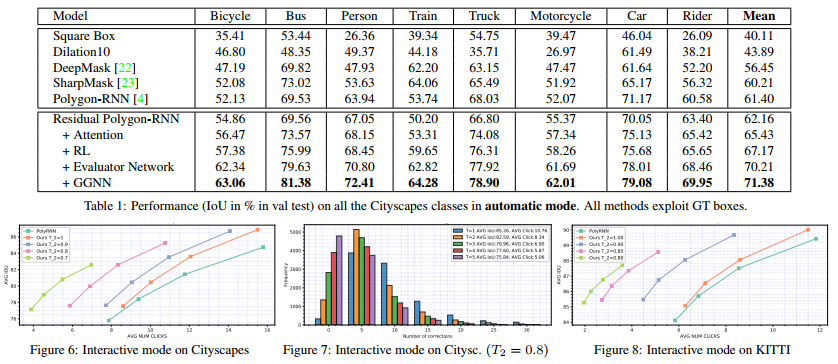
Bây giờ chúng tôi cũng giải quyết tình huống trong đó mô hình của chúng tôi được đào tạo trên một tập dữ liệu và được sử dụng để chú thích một tập dữ liệu mới. Khi dữ liệu mới đến, bộ chú thích sử dụng mô hình của chúng tôi để chú thích các đối tượng và sửa các dự đoán sai khi cần thiết. Chúng tôi đề xuất một cách tiếp cận đơn giản để tinh chỉnh mô hình của chúng tôi trong một kịch bản như vậy, theo kiểu trực tuyến.

Chúng ta hãy biểu thị C là số lượng dữ liệu mới được chia thành, CS là kích thước khối, NEV là số bước đào tạo cho mạng đánh giá và NMLE; NRL là số bước đào tạo cho mỗi đoạn với MLE và RL, tương ứng. Tinh chỉnh trực tuyến của chúng tôi được mô tả trong Thuật toán 1 trong đó Dự đoán chính xác đề cập đến bộ chú thích (mô phỏng) trong vòng lặp. Bởi vì chúng tôi đào tạo về dữ liệu đã được sửa, chúng tôi làm trơn tru các mục tiêu của mình để đào tạo MLE với một biến đổi khoảng cách manhattan bị cắt ở khoảng cách 2.



1. **Kết quả thực nghiệm**

Trong phần này, chúng tôi cung cấp một đánh giá toàn diện về mô hình của chúng tôi.Chúng tôi báo cáo cả kết quả tự động và tương tác một kết quả ký hiệu trên bộ dữ liệu Cityscapes đầy thách thức [8] và so sánh với các phương pháp pixel-khôn ngoan.Sau đó, chúng tôi nhân vật hóa khả năng khái quát hóa của mô hình của chúng tôi bằng cách đánh giá bộ dữ liệu KITTI [10] và bốn bộ dữ liệu ngoài miền bao trùm các cảnh chung [43],trên không [34], và y tế im agery [16, 11]. Cuối cùng, chúng tôi đánh giá sơ đồ tinh chỉnh trực tuyến của chúng tôi, chứng minh giảm đáng kể thời gian chú thích cho các bộ dữ liệu mới. Lưu ý rằng như trong [4], chúng tôi giả định rằng các hộp sự thật mặt đất do người dùng cung cấp xung quanh các đối tượng được đưa ra. Chúng tôi tiếp tục phân tích sự mạnh mẽ của mô hình của chúng tôi với nhiễu đối với các hộp này, bắt chước các chú thích nhiễu.



* 1. **Chú thích trong tên miền**

Trước tiên chúng tôi đánh giá cách tiếp cận của chúng tôi trong đào tạo và đánh giá trên cùng một tên miền. Điều này bắt chước kịch bản trong đó người ta lấy một tập dữ liệu hiện có và sử dụng nó để chú thích các hình ảnh mới từ cùng một tên miền.Cụ thể, chúng tôi sử dụng bộ dữ liệu Cityscapes [8], hiện là một trong những điểm chuẩn toàn diện nhất cho phân khúc cá thể.Nó chứa 2975 đào tạo, 500 xác nhận và 1525 hình ảnh thử nghiệm với 8 lớp ngữ nghĩa. Để đảm bảo so sánh công bằng, chúng tôi thực hiện theo cách phân chia thay thế tương tự được đề xuất bởi [4].Như trong [4],chúng tôi xử lý các đa giác chân thực theo thứ tự độ sâu để thu được các đa giác chỉ cho các vùng hiển thị của mỗi trường hợp.

**Đánh giá số liệu:** Chúng tôi sử dụng hai biện pháp định lượng để đánh giá mô hình của chúng tôi.1) Chúng tôi sử dụng chỉ số giao nhau trên liên minh (IoU) để đánh giá chất lượng của đa giác được tạo và 2) chúng tôi tính toán số lần nhấp của chú thích cần thiết để sửa lỗi trướcmô hình được thực hiện bởi các mô hình. Chúng tôi mô tả các giao thức điều chỉnh chi tiết trong phần tiếp theo.

**Baselines:** Theo [4], chúng tôi so sánh với Deep Mask [22], SharpMask [23], cũng như Polygon-RNN [4] là các đường cơ sở hiện đại. Lưu ý rằng hai cách tiếp cận đầu tiên là các phương pháp pixel-pixel và lỗi trong đầu ra của chúng không thể dễ dàng được sửa chữa bởi một chú thích. Để công bằng, chúng tôi chỉ so sánh chế độ tự động của chúng tôi với phương pháp của họ. Theo cách tiếp cận ban đầu của họ, [22, 23] mẫu vá đầy đủ ở các tỷ lệ khác nhau trên toàn bộ hình ảnh. Ở đây, chúng tôi đánh giá [22, 23] bằng cách cung cấp các hộp chính xác cho các mô hình của họ.

Chúng tôi cũng sử dụng hai đường cơ sở bổ sung là SquareBox từ [4] và Dilation10 từ [38]. SquareBox coi hộp giới hạn được cung cấp như dự đoán của nó. Dilation10 được lấy từ các kết quả phân đoạn của [38] từ mô hình đã được đào tạo trên bộ dữ liệu Cityscapes.

**Chế độ tự động:** Chúng tôi so sánh Polygon-RNN ++ với các đường cơ sở trong Bảng 1, cũng như bỏ qua việc sử dụng từng thành phần trong mô hình của chúng tôi. Ở đây, Residual Polygon RNN đề cập đến mô hình ban đầu với kiến ​​trúc hình ảnh mới lạ của chúng tôi thay vì VGG. Cách tiếp cận đầy đủ của chúng tôi vượt trội so với người thực hiện hàng đầu [4] gần 10% IoU và đạt được hiệu suất tốt nhất cho mỗi lớp. Hơn nữa, trung bình Polygon-RNN ++ vượt qua thỏa thuận được báo cáo của con người [4] là 78,6% IoU trên ô tô. Sử dụng thỏa thuận của con người về ô tô như một ủy quyền, chúng tôi nhận thấy rằng mô hình cũng đạt được hiệu suất ở cấp độ con người cho các loại xe tải và xe buýt.

**Chế độ tương tác:** Chế độ tương tác nhằm giảm thiểu thời gian chú thích trong khi có được chú thích chất lượng cao. Theo mô phỏng được đề xuất trong [4], chúng tôi tính toán số lần nhấp của chú thích cần thiết để điều chỉnh dự đoán từ mô hình. Bộ chú thích sửa một dự đoán nếu nó lệch khỏi đỉnh GT tương ứng với khoảng cách tối thiểu là T, trong đó siêu tham số T chi phối chất lượng của các chú thích được tạo ra. Để so sánh công bằng, khoảng cách được tính bằng khoảng cách manhattan ở độ phân giải đầu ra mô hình sử dụng ngưỡng khoảng cách T [1; 2; 3; 4], như trong [4].

Ngoài ra, chúng tôi giới thiệu ngưỡng thứ hai T2, được định nghĩa là IoU giữa đa giác dự đoán và mặt nạ GT, trong đó chúng tôi coi các đa giác đạt được thỏa thuận trên T2 là không cần thiết để bộ chú thích can thiệp. Chúng tôi khai thác ngưỡng này do mô phỏng hiệu chỉnh có phần không đạt yêu cầu ở trên: ví dụ: nếu đỉnh được dự đoán nằm dọc theo cạnh đa giác GT, đỉnh này trong thực tế là chính xác và không nên sửa. Lưu ý rằng, trong trường hợp cực đoan của T2 = 1, trình giả lập của chúng tôi giả định rằng việc hiệu chỉnh là cần thiết cho mọi đa giác dự đoán. Trong trường hợp này, mô phỏng tương đương với mô phỏng được trình bày trong [4].

Trong hình 6, chúng tôi so sánh số lần nhấp trung bình cho mỗi phiên bản được yêu cầu để chú thích tất cả các lớp trên tập giá trị Cityscapes (500 hình ảnh) với các giá trị khác nhau của T2. Sử dụng T2 = 1, chúng tôi thấy rằng mô hình của chúng tôi vượt trội hơn [4], yêu cầu ít lần nhấp hơn để có được cùng một IoU. Điều đặc biệt là tôi rất ấn tượng rằng tại T2 = 0.8, mô hình của chúng tôi vẫn chính xác hơn Polygon-RNN tại T2 = 1.0.Với T2 = 0.7, chúng tôi đạt được hơn 80% mIoU chỉ với 5 lần nhấp cho mỗi đối tượng trung bình, tức là giảm hơn 50% so với [4]. Hình 7 hiển thị tần suất hiệu chỉnh cần thiết cho T khác nhau tại T2 = 0.8. Trong 4.4, chúng tôi hiển thị kết quả với các chú thích thực sự của con người.

Tính mạnh mẽ đối với nhiễu của hộp giới hạn: Để mô phỏng hiệu ứng của bộ chú thích lười biếng, chúng tôi phân tích ảnh hưởng của nhiễu trong hộp được cung cấp cho mô hình. Chúng tôi ngẫu nhiên mở rộng hộp theo tỷ lệ phần trăm chiều rộng và chiều cao của nó. Kết quả trong Bảng 4 minh họa rằng mô hình của chúng tôi rất mạnh đối với một số lượng nhiễu (0-5%). Ngay cả khi có nhiễu vừa và lớn (5-10%, 10-15%),nó vượt trội so với hiệu suất được báo cáo của các đường cơ sở trước đó sử dụng các hộp hoàn hảo.

* + 1. Phân đoạn cấp độ

Chúng tôi đánh giá mô hình của chúng tôi về nhiệm vụ phân đoạn đối tượng hình ảnh đầy đủ (tự động). Do PolygonRNN ++ yêu cầu các hộp giới hạn, chúng tôi sử dụng FasterRCNN [28] để phát hiện các đối tượng. Các hộp dự đoán sau đó được đưa đến mô hình của chúng tôi để tạo các phân đoạn thể hiện đa giác. Đánh giá Polygon-RNN ++ với FasterRCNN trên bộ kiểm tra Cityscapes đạt được 22.8% AP và 42.6% AP50. Theo [19], chúng tôi cũng thêm phân đoạn ngữ nghĩa [42] để xử lý hậu quả. Chúng tôi chỉ đơn giản thực hiện một logic và hoạt động giữa bản đồ ngữ nghĩa lớp dự đoán và dự đoán của chúng tôi. Theo sơ đồ này, chúng tôi đạt được 25.49% AP và 45.47% AP50 trên bộ thử nghiệm. Thông tin chi tiết có trong Phụ lục.

* 1. **Đánh giá tên miền chéo**

Trong phần này, chúng tôi phân tích hiệu suất của mô hình của chúng tôi trên các bộ dữ liệu khác nhau, ghi lại cả hai sự thay đổi trong môi trường (KITTI [10]) và miền (cảnh chung, trên không, y tế). Trước tiên, chúng tôi sử dụng mô hình của chúng tôi được đào tạo về Cityscapes mà không có bất kỳ tinh chỉnh nào trên các bộ dữ liệu này.

KITTI [10]: Chúng tôi sử dụng Polygon-RNN ++ để chú thích 741 trường hợp của KITTI [10] được cung cấp bởi [5]. Các kết quả ở chế độ tự động được báo cáo trong Bảng 3 và hiệu suất với một người trong vòng lặp được minh họa trong Hình 8. Phương pháp của chúng tôi vượt trội hơn tất cả các đường cơ sở cho thấy sự mạnh mẽ của nó để thay đổi trong môi trường trong khi ở trong một miền tương tự. Với một chú thích trong vòng lặp, mô hình của chúng tôi yêu cầu trung bình ít hơn 5 lần nhấp so với [4] để đạt được cùng một IoU.Nó đạt được thỏa thuận cấp độ con người là 85% như được báo cáo bởi [5] bằng cách chỉ cần trung bình 2 lần nhấp bởi bộ chú thích (mô phỏng).

* + 1. Hình ảnh ngoài miền

Chúng tôi xem xét các bộ dữ liệu thể hiện các mức độ dịch chuyển tên miền khác nhau từ bộ dữ liệu Cityscapes để đánh giá khả năng khái quát hóa của mô hình của chúng tôi.

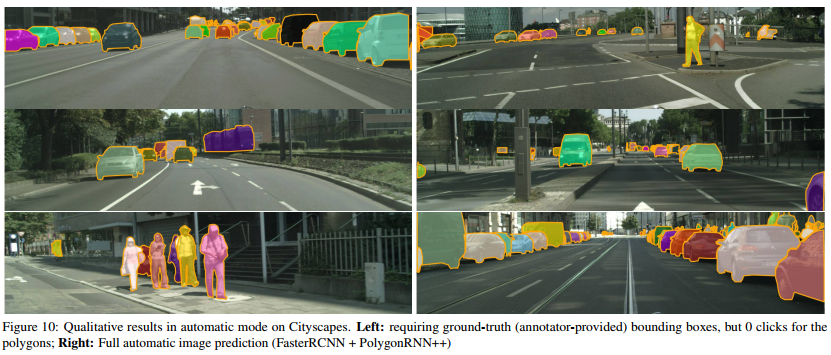
**ADE20K [43]:** Bộ dữ liệu ADE20K là bộ dữ liệu phân tích cú pháp cảnh eral đầy thách thức có chứa 20.210 hình ảnh trong tập huấn luyện, 2.000 hình ảnh trong bộ xác thực và 3,000 hình ảnh trong bộ thử nghiệm. Chúng tôi chọn tập hợp con các danh mục sau từ bộ xác thực: máy thu truyền hình, xe buýt, xe hơi, lò nướng, người và xe đạp trong đánh giá của chúng tôi.Chúng tôi hiển thị kết quả cho nhiều lớp hơn trong Phụ lục.

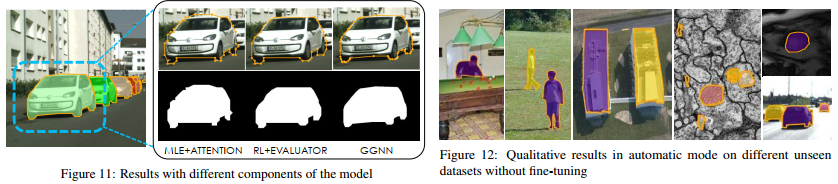
**Hình ảnh trên không [34]:** Bộ dữ liệu trên không [34] bao gồm 65 hình ảnh trên không về cảnh nông thôn chứa bảy mái nhà của tòa nhà eral, phần lớn trong số đó thể hiện hình học đa giác khá phức tạp. Hiệu suất cho bộ dữ liệu này được báo cáo cho bộ thử nghiệm.

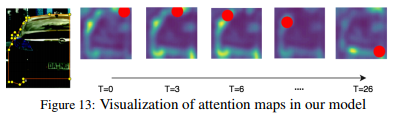
**Hình ảnh y tế [16, 33, 11]:** Chúng tôi sử dụng hai bộ dữ liệu đề cập phân biệt y tế [16, 33] và [11] cho các thí nghiệm của chúng tôi. được sử dụng trong Thử thách phân đoạn tâm thất trái [33], chia dữ liệu của 200 bệnh nhân như nhau trong các tập huấn luyện và xác nhận. Chúng tôi báo cáo hiệu suất trên một tập hợp con của bộ xác nhận chỉ bao gồm các đường viền bên ngoài phân đoạn biểu mô. Phần sau cung cấp hai ngăn xếp hình ảnh (đào tạo và kiểm tra) mỗi phần chứa 20 phần từ phần nối tiếp Hình ảnh kính hiển vi điện tử (ssTEM) của dây thần kinh bụng. Chúng tôi sử dụng các phân đoạn ty thể và khớp thần kinh từ dữ liệu này để kiểm tra mô hình của chúng tôi.Vì các trường hợp thực tế cho bộ kiểm tra không có sẵn công khai, chúng tôi đánh giá về tập huấn luyện.

**Kết quả định lượng:** Đối với các bộ dữ liệu ngoài miền,chúng tôi trong việc cắt xén một đường cơ sở (có tên là Ellipse) vừa với hình elip vào hộp giới hạn GT, được thúc đẩy bởi sự quan sát rằng nhiều trường hợp trong [33] là hình elip. Chúng tôi hiển thị kết quả với các hộp giới hạn hoàn hảo và mở rộng (mở rộng tương tự như mô hình của chúng tôi) cho Square Box và Ellipse. DeepMask và SharpMask đã được đánh giá với các hộp giới hạn hoàn hảo với ngưỡng được đề xuất bởi các tác giả. Bảng 2, chứng minh khả năng khái quát hóa cao của mô hình của chúng tôi.

**Tinh chỉnh trực tuyến:** Trong các thử nghiệm này, bộ chú thích mô phỏng của chúng tôi có các tham số T = 1 và T2 = 0.8. Hình 9 báo cáo tỷ lệ phần trăm các lần nhấp được lưu đối với đa giác GT cho mô hình Cityscapes của chúng tôi và các mô hình tinh chỉnh trực tuyến. Chúng tôi thấy rằng cách tiếp cận thích ứng của chúng tôi khắc phục sự thay đổi miền nghiêm trọng với ít nhất một đoạn dữ liệu (40 hình ảnh cho Sunnybrook, 3 cho ssTEM, 200 cho ADE và 20 cho Aerial) thể hiện sự khái quát hóa mạnh mẽ. Nhìn chung, chúng tôi cho thấy giảm ít nhất 65% tổng số lần nhấp trên tất cả các bộ dữ liệu, với các con số gần như 100% cho bộ dữ liệu MR của Sunnybrook. Chúng tôi tin rằng những kết quả này mở đường cho một công cụ chú thích thực sự có thể học cùng với người chú thích và giảm đáng kể nỗ lực của con người.







* 1. **Kết quả định tính**

Hình. 10 cho thấy các dự đoán ví dụ thu được trong chế độ tự động trên Cityscapes. Chúng tôi minh họa các cải tiến từ các phần cụ thể của mô hình trong Hình 11. Chúng tôi thấy cách sử dụng RL và mạng đánh giá dẫn đến các dự đoán sắc nét hơn, trong khi nâng cấp GGNN, thêm điểm và xây dựng một đa giác giống như chú thích của con người. Hình 12 giới thiệu các dự đoán tự động từ PolygonRNN ++ trên các bộ dữ liệu ngoài miền. Chúng tôi nhắc nhở người đọc rằng việc ghi nhãn này có được bằng cách khai thác các hộp giới hạn GT và không tinh chỉnh.

* 1. **Tương tác với chú thích con người**

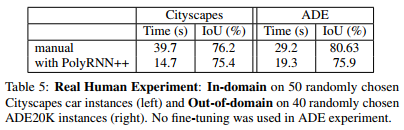
Để chứng thực những phát hiện của chúng tôi với một chú thích mô phỏng, chúng tôi cũng đã tiến hành một thí nghiệm quy mô nhỏ với các chú thích thực sự của con người trong vòng lặp. Để kết thúc, chúng tôi đã triển khai một công cụ chú thích rất đơn giản chạy mô hình của chúng tôi ở phần phụ trợ. Chúng tôi đã thu được 54 trường hợp ô tô từ Cityscapes được sử dụng bởi [4] cho thí nghiệm trên người của họ. Chúng tôi đã yêu cầu hai đối tượng con người chú thích những tương tác này bằng mô hình của chúng tôi và hai đối tượng chú thích bằng tay. Mặc dù chúng tôi giải thích cách thức hoạt động của công cụ, chúng tôi không đào tạo các chú thích sử dụng công cụ của chúng tôi. Tất cả các chú thích của chúng tôi đã ở trong nhà.

Thời gian bắt đầu khi một chú thích lần đầu tiên nhấp vào một đối tượng và dừng lại khi nhấp vào nút "gửi". Trong khi sử dụng mô hình của chúng tôi, bộ chú thích cần vẽ một hộp giới hạn xung quanh đối tượng, mà chúng tôi đưa vào thời gian báo cáo của chúng tôi. Lưu ý rằng chúng tôi hiển thị đối tượng cho chú thích bằng cách cắt hình ảnh bên trong hộp phóng to. Do đó, các chú thích của chúng tôi rất nhanh trong việc vẽ các hộp, mất trung bình khoảng 2 giây. Trong kịch bản thực, bộ chú thích sẽ chú thích toàn bộ hình ảnh, do đó sẽ dành nhiều thời gian hơn cho việc đặt các hộp xung quanh các đối tượng - tuy nhiên, bước này thường là phổ biến đối với các công cụ chú thích hoàn toàn thủ công.

Các kết quả được trình bày trong Bảng 5. Chúng tôi quan sát thấy rằng khi sử dụng mô hình của chúng tôi, các trình chú thích nhanh hơn gấp 3 lần, chỉ với thỏa thuận IoU thấp hơn một chút với sự thật cơ bản. Lưu ý rằng chúng tôi đã sử dụng một phiên bản cơ bản của công cụ, với phạm vi để cải thiện các khía cạnh kỹ thuật khác nhau. Các tác giả của [4] đã báo cáo rằng trên các ví dụ này, các đối tượng con người cần trung bình 42,2 giây mỗi lần sử dụng GrabCut [31], đồng thời đạt được IoU thấp hơn (70.7).

Chúng tôi cũng sử dụng mô hình của chúng tôi trên chú thích tên miền chéo. Cụ thể, chúng tôi sử dụng bộ dữ liệu ADE20k và mô hình của chúng tôi được đào tạo về Cityscapes (không tinh chỉnh). Chúng tôi chọn ngẫu nhiên tổng cộng 40 trường hợp xe hơi, người, ghế sofa và chó. Ở đây xe hơi và người là hai lớp được nhìn thấy trong Cityscapes (tức là, người đi bộ trong Cityscapes), và sofa và chó là những thể loại không nhìn thấy được. Từ kết quả trong Bảng 5, chúng tôi quan sát thấy rằng con người vẫn nhanh hơn khi sử dụng công cụ của chúng tôi, nhưng ít hơn, như mong đợi. Thỏa thuận về IoU cũng thấp hơn ở đây, cho thấy sự thiên vị tiềm năng của các nhà chú thích để chấp nhận các dự đoán ít hơn hoàn hảo khi cần sửa chữa nhiều hơn. Chúng tôi đang phát triển một công cụ chú thích đầy đủ hơn và có kế hoạch điều tra các hiện tượng như vậy ở quy mô lớn hơn.

**Hạn chế:** Mô hình của chúng tôi dự đoán chỉ có một đa giác trên mỗi hộp và thường chú thích đối tượng trung tâm hơn.Nếu một đối tượng phá vỡ đối tượng kia, cách tiếp cận của chúng ta có xu hướng dự đoán đối tượng bị chặn là một đa giác đơn lẻ. Do đó, các trường hợp thất bại hiện tại chủ yếu xoay quanh các đối tượng đa thành phần lớn. Cũng lưu ý rằng chúng tôi không xử lý các lỗ hổng không xuất hiện trong bất kỳ bộ dữ liệu được thử nghiệm nào của chúng tôi. Trong chế độ tương tác, chúng tôi sẽ có lợi rất nhiều bằng cách cho phép con người thêm / xóa điểm.



1. **Kết luận**

Trong bài báo này, chúng tôi đã đề xuất Polygon-RNN ++, một mô hình cho phân đoạn đối tượng có thể được sử dụng để chú thích tương tác các bộ dữ liệu phân đoạn. Mô hình được xây dựng dựa trên Polygon-RNN [4], nhưng giới thiệu một số cải tiến quan trọng vượt trội đáng kể so với phương pháp trước đó ở cả hai chế độ tự động và tương tác. Chúng tôi tiếp tục thể hiện sự mạnh mẽ của mô hình của mình đối với các chú thích ồn ào và cho thấy cách nó khái quát hóa cho các miền mới. Chúng tôi cũng chỉ ra rằng với sơ đồ tinh chỉnh trực tuyến đơn giản, mô hình của chúng tôi có thể được sử dụng để thích ứng hiệu quả với các bộ dữ liệu mới, ngoài miền.