

HỌC VIỆN CÔNG NGHỆ BUỔI CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG WEB DỰA TRÊN MÔ HÌNH CNN STARDIST ĐỂ
NHẬN DẠNG ĐỒI TUỢNG

Giảng viên hướng dẫn: TS. NGUYỄN TẤT THẮNG

Sinh viên thực hiện: VŨ TRUNG KIÊN

Lớp: D19CNPM5

Khóa: 2019 – 2024

Hệ: ĐẠI HỌC CHÍNH QUY

HÀ NỘI – 1/2024

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM (Của người hướng dẫn)

Điểm: (Bằng chữ:)

Hà Nội, ngày tháng năm 20...

Giảng viên hướng dẫn

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM (Của giảng viên phản biện)

Điểm: (Bằng chữ:

Hà Nội, ngày tháng năm 20...

Giảng viên phản biện

LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành đến Ban giám hiệu nhà trường cùng toàn thể thầy cô trong Học viện Công nghệ Bưu chính Viễn thông đã luôn dành tâm huyết, sự quan tâm, dạy dỗ, giúp đỡ và đồng hành cùng em trong suốt 4 năm học tập tại trường. Nhờ được sự dìu dắt, dạy dỗ của thầy cô mà em ngày một trưởng thành, tự tin hơn trong học tập cũng như trong cuộc sống. Em không chỉ tiếp thu được kiến thức, kỹ năng của ngành học mình chọn mà còn có thêm kỹ năng sống, sự tự tin và mong muốn được cống hiến những kiến thức mà thầy cô truyền đạt vào cuộc sống, vì sự phát triển của đất nước.

Đặc biệt, em xin gửi lời cảm ơn đến thầy Nguyễn Tất Thắng, thầy đã luôn tin tưởng, giúp đỡ em trong suốt quá trình làm Đồ án tốt nghiệp. Nếu không có những lời động viên, chỉ dẫn tận tình của thầy thì thực sự khó khăn để có thể tự mình hoàn thiện bản đồ án này. Một lần nữa em xin chân thành cảm ơn thầy rất nhiều.

Em cũng xin chân thành cảm ơn các Thầy, Cô trong Ban hội đồng đã dành thời gian quan tâm, lắng nghe, đóng góp ý kiến và nhận xét cho bản đồ án tốt nghiệp của em. Các thầy cô đã trang bị cho em không chỉ những kiến thức chuyên môn mà còn cả những kỹ năng mềm để em có thể vận dụng vào thực tiễn cuộc sống và tự hoàn thiện bản thân mình hơn. Qua đây em cũng mong học hỏi được thêm nhiều điều, khắc phục những thiếu sót, non nớt của mình để ngày càng hoàn thiện hơn kiến thức và kỹ năng của bản thân.

Trong thời gian làm đồ án, em đã dồn hết tâm huyết để xây dựng bản đồ án này. Tuy nhiên do kiến thức còn hạn hẹp, kinh nghiệm thực tế của em chưa nhiều nên trong quá trình làm đồ án tốt nghiệp không thể tránh khỏi những thiếu sót, em rất mong nhận được sự thông cảm và góp ý của Thầy, Cô để bản đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn

MỤC LỤC

CHƯƠNG I. ĐẶT VẤN ĐỀ	5
1.1. Thị giác máy tính	5
1.2. Phân đoạn hình ảnh	9
1.3. Ứng dụng trong nghiên cứu và công nghiệp	13
1.4. Bài toán của đồ án	17
CHƯƠNG II. CÁC PHƯƠNG PHÁP TIẾP CẬN THÔNG THƯỜNG	17
2.1. Giới thiệu về mạng CNN (mạng nơ-ron tích chập)	18
2.1.1. Mở đầu	18
2.1.2. Khái niệm CNN?	19
2.1.3. Cách hoạt động của CNN	19
2.1.4. Tầng tổng hợp - Pooling Layer	23
2.1.5. Tầng kết nối đầy đủ - Fully Connected Layer	25
2.2. Phương pháp tiếp cận Bottom-Up (từ dưới lên)	25
2.3. Phương pháp tiếp cận Top-Down	31
2.4. So sánh 2 cách tiếp cận	35
2.4.1. Top-down	35
2.4.2. Bottom-up	35
CHƯƠNG III. PHƯƠNG PHÁP TIẾP CẬN CỦA ĐỒ ÁN.	36
3.1. Mô hình Stardist	37
3.1.1. Tổng quan về Stardist	37
3.1.2. Nguyên lý của Stardist	40
3.1.3. Quá trình xử lý	42
3.2. Quá trình huấn luyện (Training process)	46
3.3. Đánh giá các chỉ số	47
3.4. So sánh các phương pháp	48
CHƯƠNG IV. ÁP DỤNG VÀ TRIỂN KHAI MÔ HÌNH STARDIST VÀO BÀI TOÁN CỦA ĐỒ ÁN	49
4.1. Chuẩn bị dữ liệu và huấn luận mô hình	50
4.2. Triển khai mô hình lên ứng dụng web	62
CHƯƠNG V. KẾT LUẬN	69

DANH MỤC HÌNH ẢNH

Hình 1.1. Các vấn đề của thị giác máy tính.....	8
Hình 1.2. Ba loại tác vụ phân đoạn hình ảnh.....	10
Hình 1.3. Ảnh trước và sau khi nhị phân hóa	11
Hình 1.4. Ảnh trước và sau khi phân đoạn cạnh.....	12
Hình 1.5. Phân đoạn hình ảnh trong y tế, sinh học	13
Hình 1.6. Bọt khí	14
Hình 1.7. Kim cương.....	15
Hình 1.8. Nhận dạng khiếm khuyết của thực phẩm.....	16
Hình 1.9. Nhận dạng bao bì khi đóng gói thực phẩm	18
Hình 2.1. Mạng nơ-ron tích chập	19
Hình 2.2. Quá trình hoạt động của mạng nơ-ron tích chập	20
Hình 2.3. Ảnh RGB	20
Hình 2.4. Cách thức hoạt động của CNN	21
Hình 2.5. Cách hoạt động của CNN với ảnh RGB	22
Hình 2.6. Các lớp trong mạng nơ-ron tích chập	23
Hình 2.7. Quá trình chuyển đổi qua từng lớp	24
Hình 2.8. Ví dụ về một hàm pooling là Max Pooling của tầng tổng hợp - pooling layer	24
Hình 2.9. Hàm max pooling và average pooling	25
Hình 2.10. Tầng kết nối đầy đủ trong mạng nơ-ron tích chập	26
Hình 2.11. Các tiếp cận từ Bottom-up(từ dưới lên)	28
Hình 2.12. Mạng nơ-ron U-net	30
Hình 2.13. Dữ liệu gốc và mask.....	30
Hình 2.14. Dữ liệu gốc và dữ liệu dự đoán	32
Hình 2.15. Mạng nơ-ron Mask R-CNN.....	34
Hình 2.16. Dữ liệu thực và dự đoán(Mask R-CNN).....	35
Hình 2.17. So sánh 2 cách tiếp cận Top-down và Bottom-up	37
Hình 3.1. Mô hình StarDist.....	40
Hình 3.2. Nguyên lý của mô hình StarDist	42
Hình 3.3. Quá trình xử lý của mô hình StarDist	44
Hình 3.4. Non-Maximum Suppression, Hình 3.5. Đa giác sao-lòi.....	46
Hình 3.6. Quá trình huấn luyện của mô hình StarDist	47
Hình 3.7. Hai bước trong quá trình huấn luyện của StarDist	47
Hình 3.8. Minh họa các chỉ số đánh giá	48
Hình 3.9. Độ chính xác trung bình của các mô hình khác nhau với cùng ảnh đầu vào	49
Hình 3.10. Ngưỡng IoU của từng mô hình với cùng một tập dữ liệu	49
Hình 3.11. So sánh các mô hình với dữ liệu bình thường và dữ liệu dày đặc.....	50
Hình 4.1. Ảnh kim cương 28mm	51

Hình 4.2. Ảnh phóng to kim cương	52
Hình 4.3. Ảnh bọt khí	52
Hình 4.4. Ảnh phóng to bọt khí	53
Hình 4.5. Giao diện trang chủ QuPath	54
Hình 4.6. Giao diện project trong QuPath	54
Hình 4.7. Giao diện sau khi đánh nhãn cho dữ liệu	59
Hình 4.8. Ảnh kim cương gốc và nhãn trước khi huấn luyện	59
Hình 4.9. Ảnh bọt khí gốc và nhãn trước khi huấn luyện	60
Hình 4.11. Ảnh bọt khí gốc và dự đoán sau khi huấn luyện	60
Hình 4.12. Biểu đồ các chỉ số của mô hình với dữ liệu kim cương.....	61
Hình 4.13. Biểu đồ các chỉ số của mô hình với dữ liệu bọt khí	62
Hình 4.14. Mô hình của ứng dụng	63
Hình 4.15. Giao diện trang chủ của ứng dụng	65
Hình 4.16. Giao diện sau khi tải dữ liệu ảnh kim cương	66
Hình 4.17. Giao diện sau khi tải dữ liệu ảnh bọt khí	66
Hình 4.18. Giao diện sau khi ứng dụng hoàn thành phân đoạn.....	67
Hình 4.19. Giao diện sau khi ứng dụng hoàn thành phân đoạn.....	67
Hình 4.20. Ảnh sau khi dự đoán từ ứng dụng.....	68

Tóm Tắt

Lĩnh vực Thị giác Máy tính đóng một vai trò then chốt trong sự giao thoa giữa khoa học dữ liệu, trí tuệ nhân tạo, nghiên cứu khoa học cũng như công nghiệp. Dự án này giải quyết một thách thức cụ thể trong Thị giác Máy tính: phân đoạn hình ảnh, tập trung vào lĩnh vực công nghiệp, phân đoạn các hình dạng độc đáo như kim cương và bọt khí. Các phương pháp tiếp cận truyền thống để phân chia các hạt nhân gấp khó khăn trong các tình huống có cấu trúc chồng chéo và dày đặc. Nghiên cứu này khám phá và triển khai mô hình Stardist, một kỹ thuật mới, để vượt qua những thách thức này.

Dự án bắt đầu bằng việc giới thiệu tầm quan trọng của việc phân chia hạt nhân trong công nghiệp và trong bối cảnh rộng hơn về thị giác máy tính và những thách thức chung gặp phải trong phân tích hình ảnh. Các cách tiếp cận từ dưới lên và từ trên xuống thông thường, cùng với những hạn chế của chúng trong việc xử lý các đối tượng dày đặc và chồng chéo, sẽ được thảo luận. Việc khám phá chi tiết về mạng nơ ron tích chập (CNN), đặc biệt là kiến trúc U-net, cung cấp nền tảng cần thiết để hiểu phương pháp được đề xuất.

Cốt lõi của dự án được dành riêng cho mô hình Stardist. Các nguyên tắc, quy trình huấn luyện và ví dụ của nó được làm sáng tỏ. Một phân tích so sánh với các phương pháp phân đoạn khác được tiến hành, nhấn mạnh đến độ chính xác tính toán đạt được. Sau đó, trình bày việc áp dụng mô hình Stardist vào các tình huống trong thế giới thực, đặc biệt là lĩnh vực công nghiệp. Điều này liên quan đến việc lựa chọn và chuẩn bị dữ liệu huấn luyện, đánh giá độ chính xác và triển khai mô hình trên máy chủ web.

Tóm lại, nghiên cứu cung cấp những phát hiện và đánh giá sự thành công của mô hình Stardist trong việc phân chia hạt nhân với hình dạng độc đáo. Dự án không chỉ góp phần nâng cao hiểu biết về phân khúc chuyên biệt này mà còn cung cấp những hiểu biết thực tế cho ứng dụng của nó. Hành trình thông qua các phương pháp tiếp cận truyền thống đối với mô hình Stardist đã làm sáng tỏ những thách thức và cơ hội đang diễn ra trong lĩnh vực thị giác máy tính.

Nội Dung

CHƯƠNG I. ĐẶT VẤN ĐỀ

1.1. Thị giác máy tính

Thị giác máy tính là một trong những lĩnh vực “hot” nhất của khoa học máy tính và nghiên cứu trí tuệ nhân tạo. Dù chúng vẫn chưa thể cạnh tranh với sức mạnh thị giác của mắt người nhưng đã có rất nhiều ứng dụng hữu ích được tạo ra để khai thác tiềm năng của chúng.

Sơ lược về bối cảnh lịch sử:

Năm 1966, Seymour Papert và Marvin Minsky, hai nhà tiên phong về trí tuệ nhân tạo, đã khởi động một dự án mang tên “Summer Vision Project”, một nỗ lực kéo dài hai tháng và kéo theo 10 người để tạo ra một hệ thống máy tính có thể nhận dạng các vật thể trong ảnh.

Để hoàn thành nhiệm vụ, một chương trình máy tính phải có khả năng xác định pixel nào thuộc về đối tượng nào. Đây là một vấn đề mà hệ thống thị giác của con người, được cung cấp bởi kiến thức rộng lớn của chúng ta về thế giới thực và hàng tỷ năm tiến hóa, có thể giải quyết một cách dễ dàng. Nhưng đối với máy tính, thế giới chỉ bao gồm các con số, đó là một nhiệm vụ đầy thách thức.

Vào thời điểm của dự án này, phân nhánh thống trị chủ lực của trí tuệ nhân tạo là symbolic AI, còn được gọi là AI dựa trên quy tắc (rule-based AI): Các lập trình viên tự chỉ định các quy tắc để phát hiện các đối tượng trong hình ảnh. Nhưng vấn đề là các vật thể trong ảnh có thể xuất hiện từ các góc khác nhau và trong nhiều điều kiện ánh sáng khác nhau. Đối tượng có thể xuất hiện trên một loạt các nền khác nhau hoặc bị các đối tượng khác che khuất một phần. Mỗi kịch bản này tạo ra các giá trị pixel khác nhau và thực tế không thể tạo quy tắc thủ công cho từng cái một trong số chúng.

Hắn nhiên, Summer Vision Project đã không đi xa và mang lại kết quả khá hạn chế. Vài năm sau đó, vào năm 1979, nhà khoa học Nhật Bản Kunihiko Fukushima đã đề xuất neocognitron , một hệ thống thị giác máy tính dựa trên nghiên cứu khoa học thần kinh được thực hiện trên vỏ não về thị giác của con người. Mặc dù neocognitron của Fukushima không thể thực hiện bất kỳ nhiệm vụ trực quan phức tạp nào, nhưng nó đã

đặt nền tảng cho một trong những phát triển quan trọng nhất trong lịch sử thị giác máy tính.

Cuộc cách mạng học sâu - Deep Learning

Vào những năm 1980s, nhà khoa học máy tính người Pháp Yan LeCun đã giới thiệu mạng nơ-ron tích chập (convolutional neural network, CNN), một hệ thống AI lấy cảm hứng từ neocognitron của Fukushima. Một CNN bao gồm nhiều lớp tế bào nơ-ron nhân tạo, các thành phần toán học mô phỏng gần giống hoạt động của các phiên bản sinh học của chúng.

Khi một CNN xử lý một hình ảnh, mỗi lớp của nó sẽ trích xuất các đặc trưng cụ thể từ các pixel. Lớp đầu tiên phát hiện những thứ rất cơ bản, chẳng hạn như các cạnh dọc và ngang. Khi bạn di chuyển sâu hơn vào mạng nơ-ron, các lớp sẽ phát hiện các đặc trưng phức tạp hơn, bao gồm các góc và hình dạng. Các lớp cuối cùng của CNN phát hiện những thứ cụ thể như khuôn mặt, cánh cửa và xe hơi. Lớp đầu ra của CNN cung cấp một bảng các giá trị số biểu thị xác suất mà một đối tượng cụ thể được phát hiện trong ảnh.

Mạng nơ-ron tích chập của LeCun rất tuyệt vời và cho thấy rất nhiều hứa hẹn, nhưng chúng bị cản trở bởi một vấn đề nghiêm trọng: Điều chỉnh và sử dụng chúng đòi hỏi một lượng lớn dữ liệu và tài nguyên tính toán không có sẵn tại thời điểm đó. CNN cuối cùng đã tìm thấy việc sử dụng thương mại trong một số lĩnh vực hạn chế như ngân hàng và dịch vụ buro chính, nơi chúng được sử dụng để xử lý các chữ số và chữ viết tay trên phong bì và các tờ séc. Nhưng trong lĩnh vực nhận diện đối tượng, họ đã thất bại và nhường chỗ cho các kỹ thuật học máy khác, như ‘support vector machines’ và ‘random forests’.

Vào năm 2012, các nhà nghiên cứu AI từ Toronto đã phát triển AlexNet, một mạng nơ-ron tích chập chiếm ưu thế trong cuộc thi nhận dạng hình ảnh ImageNet nổi tiếng. Chiến thắng của AlexNet cho thấy với sự gia tăng sẵn có của dữ liệu và tài nguyên điện toán, có lẽ đã đến lúc phải trở lại với CNN. Sự kiện này đã làm hồi sinh sự quan tâm đến các CNN và tạo ra một cuộc cách mạng trong Deep Learning, phân nhánh của Machine Learning liên quan đến việc sử dụng các mạng nơ-ron nhân tạo nhiều lớp.

Nhờ những tiến bộ trong mạng nơ-ron tích chập và học sâu, từ đó, lĩnh vực thị giác máy tính đã phát triển nhờ những bước nhảy vọt.

Ứng dụng của Thị giác máy tính

Nhiều ứng dụng bạn sử dụng hàng ngày sử dụng công nghệ thị giác máy tính. Google sử dụng nó để giúp bạn tìm kiếm các đối tượng và cảnh vật như là, “con chó” hoặc “hoàng hôn” trong một thư viện hình ảnh của bạn. Các công ty khác sử dụng thị giác máy tính để giúp nâng cao hình ảnh. Một ví dụ là Adobe Lightroom CC, sử dụng thuật toán Machine Learning để tăng cường chi tiết của hình ảnh được phóng to. Cơ chế phóng to (zoom in) truyền thống sử dụng các kỹ thuật nội suy để tô màu các khu vực được phóng to, nhưng Lightroom sử dụng thị giác máy tính để phát hiện các đối tượng trong hình ảnh và làm sắc nét các đặc trưng của chúng sau khi được phóng to.

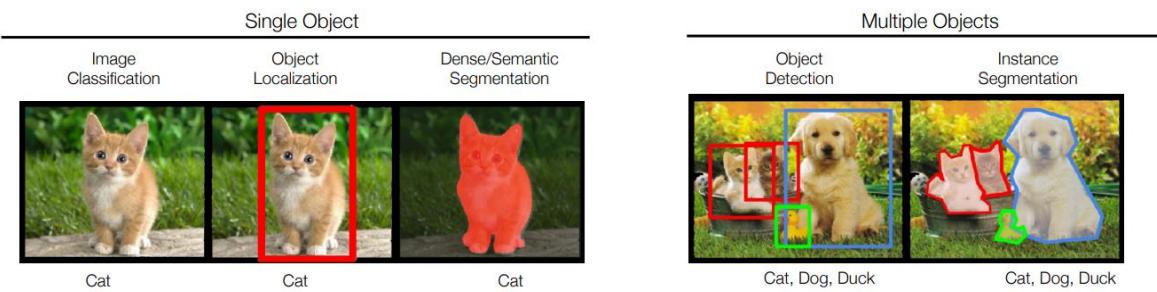
Một lĩnh vực đã đạt được tiến bộ rõ rệt nhờ những tiến bộ trong thị giác máy tính là nhận diện khuôn mặt. Apple sử dụng thuật toán nhận dạng khuôn mặt để mở khóa iPhone. Facebook sử dụng nhận dạng khuôn mặt để phát hiện người dùng trong ảnh bạn đăng lên mạng (mặc dù không phải ai cũng thích điều này). Tại Trung Quốc, nhiều nhà bán lẻ hiện cung cấp công nghệ thanh toán qua nhận diện khuôn mặt, giúp khách hàng không cần phải tiếp cận với túi tiền của họ.

Những tiến bộ trong nhận dạng khuôn mặt cũng gây ra lo lắng cho những người ủng hộ quyền riêng tư, đặc biệt là khi các cơ quan chính phủ ở các quốc gia khác nhau đang sử dụng nó để giám sát công dân của họ.

Chuyển sang các lĩnh vực chuyên biệt hơn, thị giác máy tính nhanh chóng trở thành một công cụ không thể thiếu trong y học. Các thuật toán học sâu đang cho thấy độ chính xác ấn tượng trong việc phân tích hình ảnh y tế. Các bệnh viện và trường đại học đang sử dụng thị giác máy tính để dự đoán các loại ung thư khác nhau bằng cách kiểm tra tia X và quét MRI.

Xe tự lái cũng phụ thuộc rất nhiều vào thị giác máy tính để hiểu được môi trường xung quanh. Các thuật toán học sâu phân tích các nguồn cấp dữ liệu video từ các camera được cài đặt trên xe và phát hiện người, xe hơi, mặt đường và các vật thể khác để giúp chiếc xe di chuyển trong môi trường của nó.

Các vấn đề thông thường của Thị giác máy tính.



Hình 1.1. Các vấn đề của thị giác máy tính

Trong lĩnh vực thị giác máy tính (Computer Vision), có nhiều bài toán phổ biến mà các nhà nghiên cứu và chuyên gia phải đổi mới. Dưới đây là một số bài toán chính:

1. Phân đoạn hình ảnh (Image Segmentation):

Bài toán: Tách biên và phân loại các đối tượng trong hình ảnh.

Ứng dụng: Trong y học, công nghiệp, và xe tự hành.

2. Nhận dạng đối tượng (Object Recognition):

Bài toán: Nhận diện và phân loại đối tượng trong hình ảnh.

Ứng dụng: Nhận dạng khuôn mặt, xe hơi, và vật thể trong môi trường công nghiệp.

3. Phân loại hình ảnh (Image Classification):

Bài toán: Gán một nhãn cho toàn bộ hình ảnh dựa trên nội dung của nó.

Ứng dụng: Nhận dạng chủ đề hình ảnh, phân loại sản phẩm trong bán lẻ.

4. Nhận dạng khuôn mặt (Facial Recognition):

Bài toán: Xác định và nhận diện khuôn mặt trong hình ảnh hoặc video.

Ứng dụng: Điện thoại di động, an ninh, và quản lý tập trung.

5. Khôi phục đa dạng hình ảnh (Image Super-Resolution):

Bài toán: Tăng độ phân giải của hình ảnh để có chất lượng cao hơn.

Ứng dụng: Hiển thị hình ảnh trên màn hình cao độ phân giải.

6. Phát hiện đối tượng (Object Detection):

Bài toán: Xác định và định vị vị trí của đối tượng trong hình ảnh.

Ứng dụng: Hệ thống an ninh, xe tự hành, và theo dõi vật thể.

1.2. Phân đoạn hình ảnh

Phân đoạn hình ảnh (image segmentation) là một trong những lĩnh vực chính của thị giác máy tính, được hỗ trợ bởi một lượng lớn nghiên cứu liên quan đến cả thuật toán xử lý hình ảnh và kỹ thuật học tập. Phân đoạn hình ảnh đứng đầu sau các ứng dụng nổi bật như Robotics, Hình ảnh y tế, Xe tự lái và Phân tích video thông minh. Lĩnh vực này cũng được biết đến bởi lịch sử nghiên cứu, phát triển lâu dài, với các công trình đầu tiên ra mắt vào đầu năm 1970-1972.

Mục đích của phân đoạn hình ảnh là nhóm các vùng hoặc phân đoạn một hình ảnh theo các nhãn lớp tương ứng. Tác vụ này tương đương với việc nhóm các pixel. Ngoài phân loại, phân đoạn hình ảnh cũng yêu cầu khoanh vùng (xác định vị trí chính xác của một đối tượng bằng cách xác định ranh giới của chúng). Do đó, có thể coi đây là bài toán mở rộng của phân loại hình ảnh (image classification).

Các loại phân đoạn hình ảnh.

Phân đoạn ảnh có thể được chia thành ba nhóm tác vụ dựa trên số lượng và loại thông tin mà chúng truyền tải. Trong khi semantic segmentation phân đoạn một ranh giới rộng của các đối tượng thuộc về một lớp cụ thể, thì instance segmentation cung cấp bản đồ phân đoạn cho từng đối tượng chi tiết trong mỗi nhãn. Panoptic segmentation cho đến nay là nhiều thông tin nhất, bởi kết hợp các tác vụ semantic segmentation và instance segmentation. Panoptic segmentation cho ta bản đồ phân đoạn của tất cả các đối tượng thuộc bất kỳ lớp cụ thể nào có trong ảnh.

* Phân đoạn Semantic

Semantic segmentation đề cập đến việc phân loại các pixel trong một hình ảnh thành các lớp ngữ nghĩa. Các pixel thuộc về một lớp cụ thể được phân loại một cách đơn giản vào lớp đó mà không cần xem xét thông tin hoặc bối cảnh nào khác.

Tác vụ này không phù hợp khi có nhiều đối tượng được nhóm chặt chẽ trên cùng một lớp trong hình ảnh. Ví dụ: Hình ảnh đám đông trên đường phố sẽ có mô hình semantic segmentation dự đoán toàn bộ khu vực đám đông thuộc về lớp “người đi bộ”, do đó cung cấp rất ít chi tiết hoặc thông tin chuyên sâu về hình ảnh.

* Phân đoạn Instance

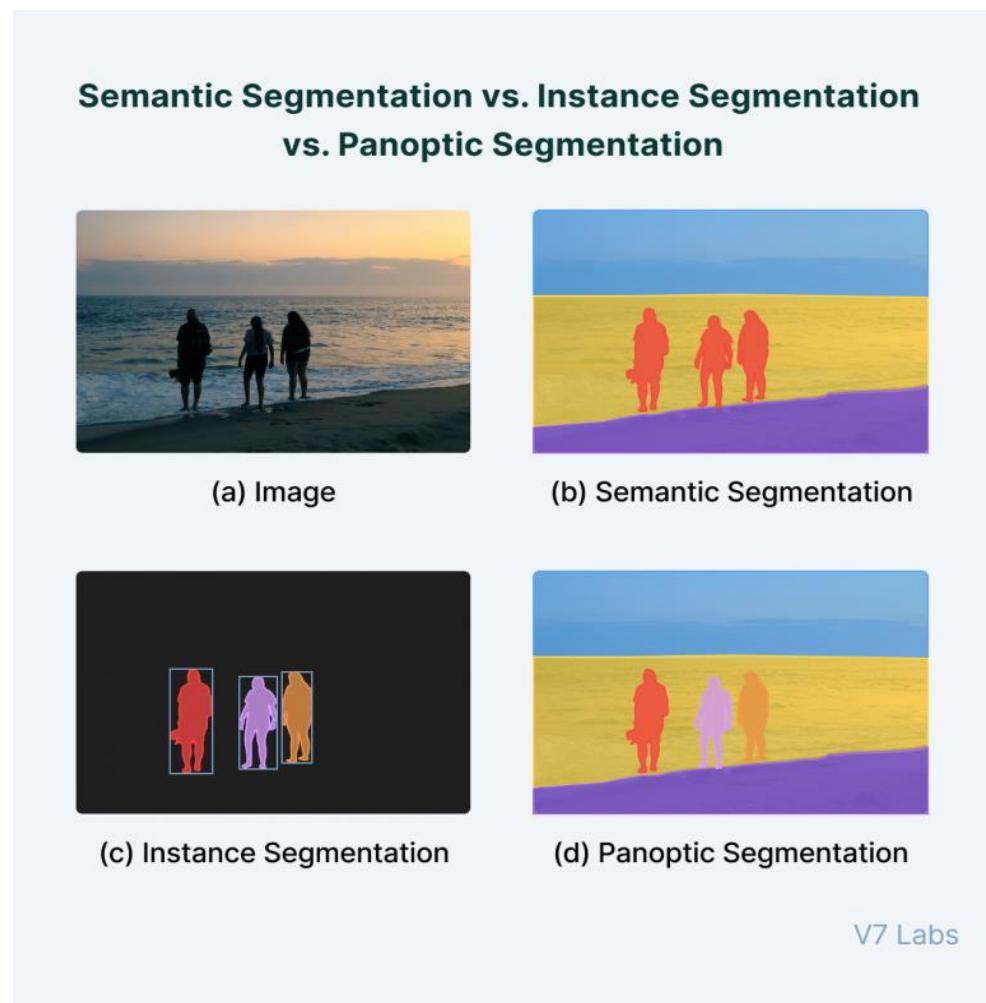
Các mô hình instance segmentation phân loại các pixel thành các danh mục trên cơ sở “các đối tượng” (instances) với nhãn cụ thể.

Ví dụ: Nếu một hình ảnh về đám đông được đưa vào mô hình instance segmentation, thì mô hình đó sẽ có thể tách biệt từng người khỏi đám đông cũng như các đối tượng xung quanh (lý tưởng nhất).

* Phân đoạn Panoptic

Panoptic segmentation, tác vụ phân đoạn được phát triển gần đây nhất, là sự kết hợp giữa semantic segmentation và instance segmentation, trong đó ranh giới của mỗi đối tượng trong ảnh được tách biệt và danh tính của đối tượng được dự đoán.

Các thuật toán panoptic segmentation có khả năng ứng dụng quy mô lớn trong các tác vụ phổ biến như ô tô tự lái, khi một lượng lớn thông tin về môi trường xung quanh phải được ghi lại ngay lập tức với sự trợ giúp của một luồng hình ảnh.



Hình 1.2. Ba loại tác vụ phân đoạn hình ảnh

Các kỹ thuật phân đoạn hình ảnh truyền thống

Khởi nguyên của phân đoạn hình ảnh bắt nguồn từ Xử lý hình ảnh kỹ thuật số cùng với các thuật toán tối ưu hóa. Kỹ thuật truyền thống này sử dụng thuật toán region growing and snakes, trong đó thiết lập các vùng ban đầu và thuật toán so sánh giá trị pixel để xác định bản đồ phân đoạn. Các phương pháp như vậy quan sát cục bộ các đặc trưng trong ảnh và tập trung vào sự khác biệt của từng phần cũng như gradients tính bằng pixel.

Những thuật toán tiên tiến hơn, xem xét toàn bộ hình ảnh đầu vào, thì xuất hiện muộn hơn nhiều, cùng với các phương pháp như adaptive thresholding, thuật toán Otsu's và phân cụm được đề xuất nằm trong số các phương pháp xử lý ảnh cổ điển.

* Thresholding

Thresholding là một trong những phương pháp phân đoạn hình ảnh đơn giản nhất, trong đó ngưỡng được đặt để chia pixel thành hai lớp. Các pixel có giá trị lớn hơn giá trị ngưỡng được đặt thành 1 trong khi các pixel có giá trị nhỏ hơn giá trị ngưỡng được đặt thành 0.

Do đó, hình ảnh được chuyển đổi thành bản đồ nhị phân, dẫn đến quá trình thường được gọi là nhị phân hóa. Nguồn hình ảnh rất hữu ích trong trường hợp sự khác biệt về giá trị pixel giữa hai lớp mục tiêu là rất cao và dễ dàng chọn giá trị trung bình làm ngưỡng.



Hình 1.3. Ảnh trước và sau khi nhị phân hóa

Thresholding thường được sử dụng để nhị phân hóa hình ảnh, nhằm cho phép sử dụng các thuật toán khác như phát hiện và nhận dạng đường viền (contour detection and identification) vốn chỉ hoạt động trên loại hình ảnh này.

* Phân đoạn Region-Based

Các thuật toán phân đoạn dựa trên khu vực hoạt động bằng cách tìm kiếm sự giống nhau giữa các pixel liền kề và nhóm chúng dưới một lớp chung. Thông thường, quy trình phân đoạn bắt đầu với một số pixel được đặt làm pixel gốc (seed pixel). Thuật toán sẽ phát hiện ranh giới trực tiếp của các pixel gốc, từ đó phân loại chúng là tương tự hoặc khác nhau. Các vùng lân cận sau đó được coi là seeds và các bước được lặp lại cho đến khi toàn bộ hình ảnh được phân đoạn.

* Phân đoạn Edge

Phân đoạn cạnh, còn được gọi là phát hiện cạnh, thực hiện tác vụ phát hiện các cạnh trong ảnh. Việc này có thể hiểu đơn giản là: phân loại pixel nào trong hình ảnh là pixel cạnh và chọn các pixel cạnh đó theo một lớp riêng biệt tương ứng.

Phát hiện cạnh thường được thực hiện bằng cách sử dụng các bộ lọc đặc biệt cung cấp cho ta biết các cạnh của hình ảnh khi tích chập. Các bộ lọc này được tính toán bằng các thuật toán chuyên dụng hoạt động dựa trên ước tính gradients của hình ảnh theo tọa độ x và y của mặt phẳng không gian. Ví dụ Canny là một trong những thuật toán phát hiện cạnh phổ biến nhất được hiển thị bên dưới.



Hình 1.4. Ảnh trước và sau khi phân đoạn cạnh

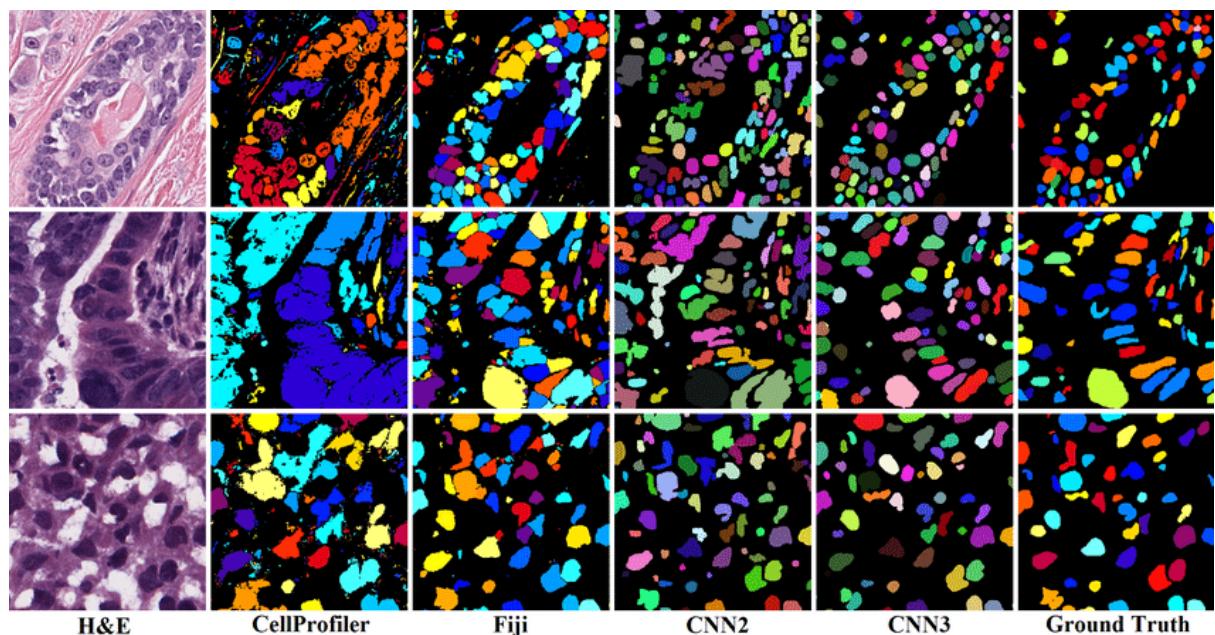
* Phân đoạn Clustering-Based

Các quy trình phân đoạn hiện đại phụ thuộc vào kỹ thuật xử lý ảnh thường sử dụng thuật toán phân cụm. Những thuật toán này có thể cung cấp các phân đoạn hợp lý trong một khoảng thời gian ngắn. Các thuật toán phổ biến như phân cụm K-mean (K-means clustering) là các thuật toán không giám sát hoạt động bằng cách phân cụm các pixel có thuộc tính chung lại với nhau.

Đặc biệt, K-means clustering xem xét tất cả các pixel và phân cụm chúng thành các lớp “k”. Khác với phương pháp phân đoạn theo vùng, các phương pháp dựa trên phân cụm không cần điểm gốc để bắt đầu phân đoạn từ đó.

1.3. Ứng dụng trong nghiên cứu và công nghiệp

Ứng dụng trong nghiên cứu sinh học, y tế:



Hình 1.5. Phân đoạn hình ảnh trong y tế, sinh học

Nghiên cứu về bệnh lý: Phân đoạn hình ảnh giúp nhận diện và phân loại các biểu hiện của bệnh lý tế bào, từ đó hỗ trợ các nhóm nghiên cứu trong việc hiểu rõ hơn về cơ chế của các bệnh lý.

Phát triển thuốc: Phân đoạn hình ảnh có thể giúp định rõ ảnh hưởng của các hợp chất và phương pháp điều trị lên tế bào.

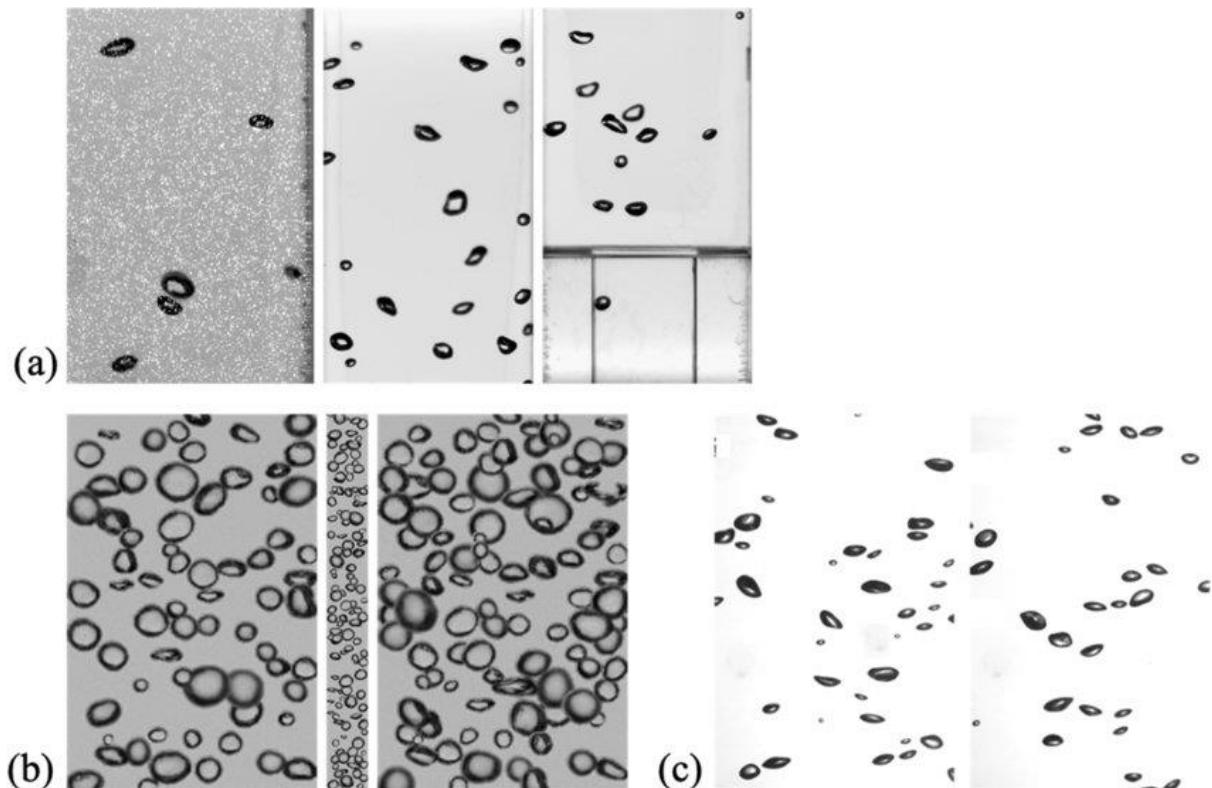
Chẩn đoán ung thư: Phân đoạn hình ảnh, tế bào giúp xác định các biểu hiện của tế bào ung thư và định vị chính xác vùng tổn thương. Điều này giúp ích trong quá trình chẩn đoán và quyết định phương pháp điều trị.

Phân tích tế bào gen: Phân đoạn ảnh tế bào giúp xác định vị trí của gen trên tế bào, từ đó cung cấp thông tin quan trọng cho nghiên cứu gen và genomics().

Kiểm tra chất lượng vắc xin, dược phẩm: Trong quá trình sản xuất dược phẩm và vắc xin, phân đoạn ảnh tế bào có thể được sử dụng để kiểm tra chất lượng sản phẩm và đảm bảo sự đồng đều của thành phần tế bào.

Ứng dụng trong công nghiệp và đời sống:

Nhận dạng ảnh bọt khí: Phân đoạn hình ảnh bọt khí là một lĩnh vực quan trọng trong thị giác máy tính và xử lý ảnh, nơi mục tiêu chính là phân chia và định vị các đối tượng bọt khí trong một hình ảnh. Việc này là một phần quan trọng của nghiên cứu về dòng chảy bọt khí, nơi mà việc hiểu và theo dõi bọt khí có vai trò quan trọng trong nghiên cứu và ứng dụng thực tế.



Hình 1.6. Bọt khí

* Ứng dụng:

Nghiên cứu khoa học: Phân đoạn hình ảnh bọt đóng vai trò quan trọng trong việc nghiên cứu đối tượng và đặc điểm của dòng chảy bọt khí, giúp hiểu rõ hơn về động học và tính chất của các hệ thống này.

Quản lý dòng chảy bọt khí trong công nghiệp: Trong các ứng dụng công nghiệp, như trong ngành dầu khí hoặc sản xuất hóa chất, phân đoạn hình ảnh bọt khí có thể được sử dụng để theo dõi và kiểm soát dòng chảy trong các quá trình sản xuất.

Kiểm tra chất lượng nước: Trong ngành môi trường, việc phân đoạn hình ảnh bọt khí có thể giúp đánh giá chất lượng nước và theo dõi sự phát triển của các loại vi sinh vật trong môi trường nước.

Y tế và nghiên cứu sinh học: Trong lĩnh vực y tế, phân đoạn hình ảnh bọt khí có thể được sử dụng để phân đoạn và đếm tế bào trong các ảnh vi sinh học, giúp trong việc nghiên cứu và chẩn đoán các bệnh lý.

Nhận dạng ảnh kim cương: Nhận dạng ảnh kim cương và đếm chúng là một lĩnh vực quan trọng trong thị giác máy tính và xử lý ảnh. Các hệ thống nhận dạng và đếm ảnh kim cương sử dụng các phương pháp và công nghệ tiên tiến để tự động nhận diện và đếm số lượng kim cương trong một hình ảnh hoặc video. Điều này là rất cần thiết đặc biệt trong ngành công nghiệp sản xuất kim cương, việc kiểm soát chặt chẽ số lượng kim cương trong từng dây chuyền là điều tiên quyết phải làm. Các phương pháp xử lý ảnh hiện nay rất khó để có thể đưa ra 1 kết quả với độ chính xác cao. Chính vì vậy đồ án này đưa ra hướng giải quyết cho những vấn đề mà xử lý ảnh truyền thống còn tồn tại.



Hình 1.7. Kim cương

Nhận dạng ảnh trong lĩnh vực thực phẩm: Trong các quá trình sản xuất thực phẩm công nghiệp, các sản phẩm có thể gặp sự cố hoặc bị hỏng trong quá trình sản xuất. Việc kiểm tra tính nguyên vẹn của sản phẩm trước khi đóng gói là quan trọng để đảm

bảo sự hài lòng của khách hàng và bảo vệ uy tín thương hiệu. Phân đoạn, nhận dạng hình ảnh giúp xác định các khuyết điểm và loại bỏ chúng trước khi chúng đến tay khách hàng, bảo vệ uy tín của công ty trước ảnh hưởng tiêu cực của hàng hóa bị hỏng, đồng thời tránh tình trạng gián đoạn và thời gian chết máy sản xuất.



Hình 1.8. Nhận dạng khuyết điểm của thực phẩm

Trong quá trình đóng gói thực phẩm, bao bì ảnh hưởng đến nhận thức của người tiêu dùng về chất lượng sản phẩm, an toàn và giá trị. Phân đoạn hình ảnh giúp kiểm tra bao bì thực phẩm để đảm bảo rằng nó được lắp ráp đúng cách, không có khuyết điểm và hoàn chỉnh, từ đó chỉ những sản phẩm chất lượng cao nhất mới đến tay khách hàng.



Hình 1.9. Nhận dạng bao bì khi đóng gói thực phẩm

1.4. Bài toán của đồ án

Mục tiêu: Tìm hiểu phương pháp học sâu dựa trên mạng CNN áp dụng cho các bài toán nhận dạng tế bào, bọt khí... Xây dựng được 1 ứng dụng web để nhận dạng các dạng ảnh như trên và có khả năng xảy ra che khuất, chồng lấn nhau.

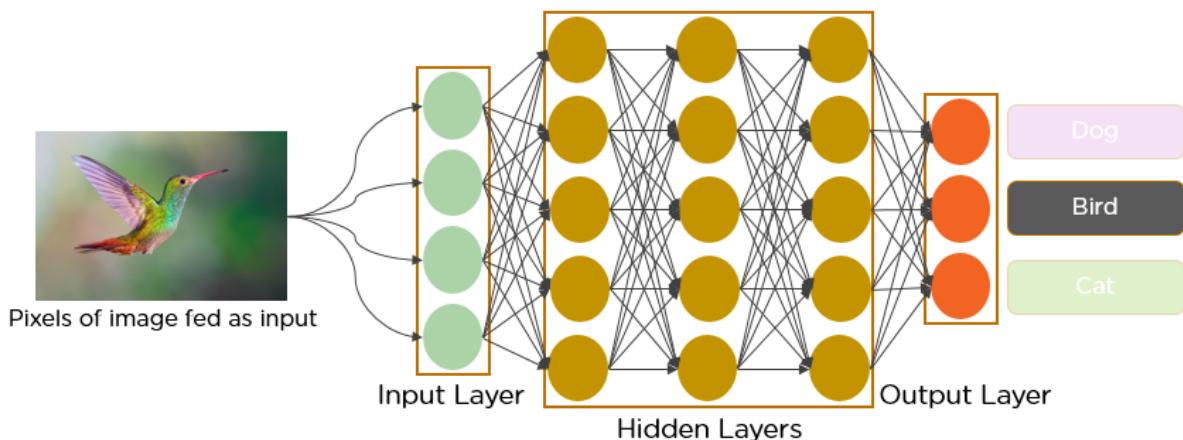
Mục đích: Tiếp cận với các bài toán công nghiệp có tính thời sự, nắm được phương pháp xử lý ảnh hiện đại, nâng cao kỹ năng lập trình phát triển ứng dụng xử lý ảnh, ứng dụng web.

Nội dung: Tìm hiểu về phân đoạn hình ảnh trong lĩnh vực thị giác máy tính, các tác vụ phân đoạn hình ảnh thông thường, các ứng dụng của phân đoạn hình ảnh trong đời sống và công nghiệp. Đi sâu vào bài toán phân đoạn, nhận dạng tế bào như kim cương và bọt khí. Đưa ra những thiếu sót của những phương pháp thông thường hiện tại từ đó mang đến một cách tiếp cận mới mang nhiều hiệu quả với độ chính xác cao. Các phương pháp tiếp cận thông thường, cùng với những thiếu sót còn tồn đọng sẽ được giới thiệu ở chương tiếp theo.

CHƯƠNG II. CÁC PHƯƠNG PHÁP TIẾP CẬN THÔNG THƯỜNG

2.1. Giới thiệu về mạng CNN (mạng nơ-ron tích chập)

Trong vài thập kỷ qua, Deep Learning đã chứng tỏ là một công cụ rất mạnh mẽ vì khả năng xử lý một lượng lớn dữ liệu. Sự quan tâm đến việc sử dụng các lớp ẩn đã vượt qua các kỹ thuật truyền thống, đặc biệt là trong nhận dạng mẫu. Một trong những mạng nơ-ron sâu phổ biến nhất là Mạng nơ-ron tích chập (còn được gọi là CNN hoặc ConvNet) trong học sâu, đặc biệt là khi nói đến các ứng dụng Thị giác máy tính.



Hình 2.1. Mạng nơ-ron tích chập

2.1.1. Mở đầu

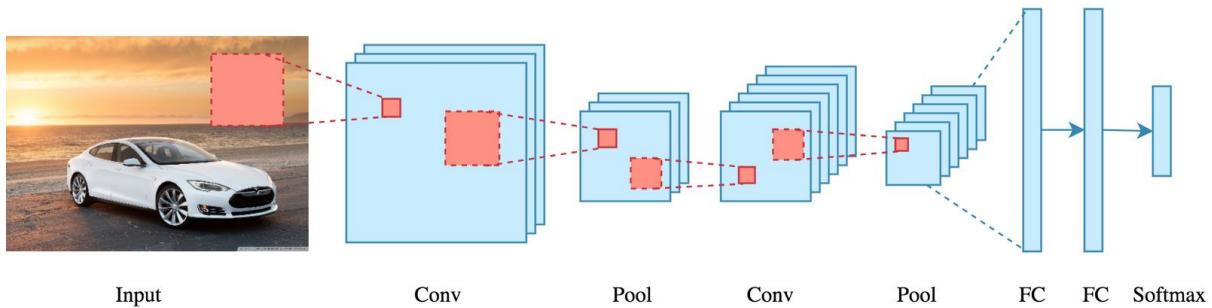
CNN lần đầu tiên được phát triển và sử dụng vào khoảng những năm 1980. Điều tốt nhất mà CNN có thể làm vào thời điểm đó là nhận dạng các chữ số viết tay. Nó chủ yếu được sử dụng trong lĩnh vực bưu chính để đọc mã zip, mã pin, v.v. Điều quan trọng cần nhớ về bất kỳ mô hình học sâu nào là nó yêu cầu một lượng lớn dữ liệu để huấn luyện và cũng cần nhiều tài nguyên máy tính. Đây là hạn chế chính của CNN vào thời kỳ đó và do đó CNN chỉ giới hạn trong lĩnh vực bưu chính và không thể thâm nhập vào thế giới học máy.

Vào năm 2012, Alex Krizhevsky nhận ra rằng đã đến lúc đưa nhánh học sâu sử dụng mạng lưới nơ-ron nhiều lớp trở lại. Sự sẵn có của các bộ dữ liệu lớn, cụ thể hơn là các bộ dữ liệu ImageNet với hàng triệu hình ảnh được gắn nhãn và sự bùng nổ năng lực

tính toán của các hệ thống máy tính đòi hỏi mới đã cho phép các nhà nghiên cứu hồi sinh CNN.

2.1.2. Khái niệm CNN?

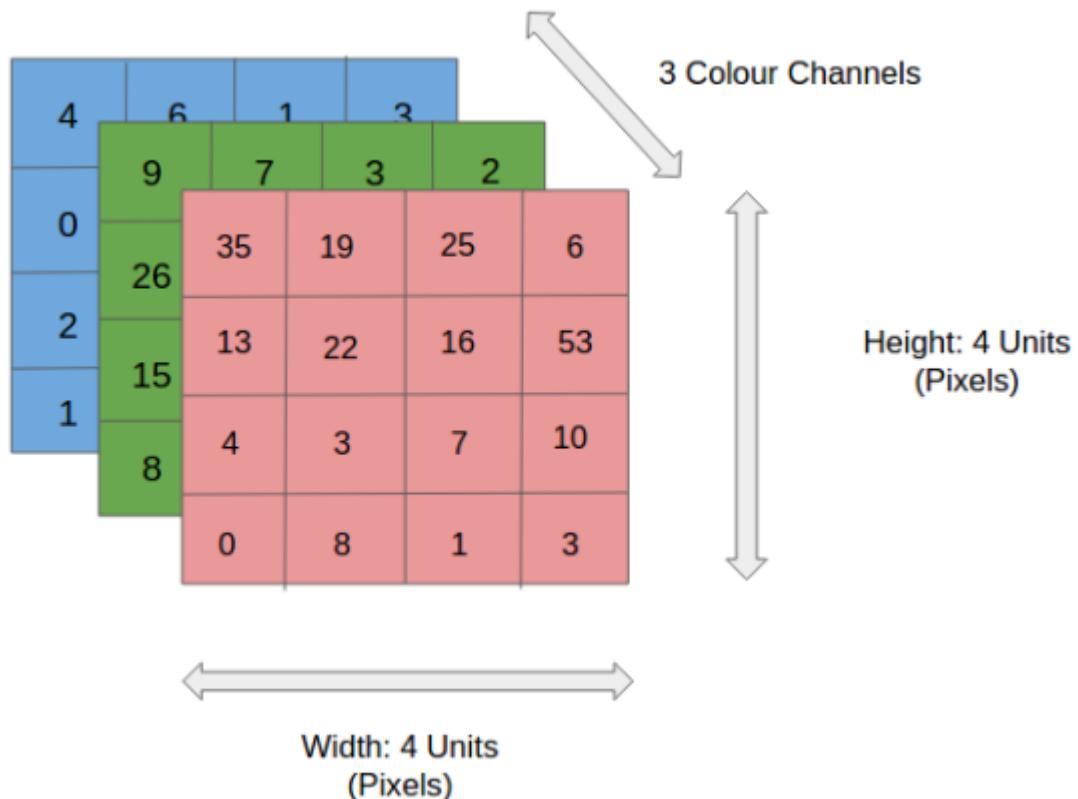
Convolutional Neural Network(mạng nơ-ron tích chập) là một trong những phương pháp quan trọng hiện nay khi thực hiện các nhận dạng ảnh. Kiến trúc mạng này xuất hiện do các phương pháp xử lý dữ liệu ảnh thường sử dụng giá trị của từng pixel. Vậy nên với một ảnh có giá trị kích thước 100×100 sử dụng kênh RGB ta có tổng cộng ta có $100 * 100 * 3$ bằng 30000 nút ở lớp đầu vào. Điều đó kéo theo việc có một số lượng lớn trọng số và độ chêch/lệch dẫn đến mạng nơ-ron trở nên quá đồ sộ, gây khó khăn cho việc tính toán. Hơn nữa, chúng ta có thể thấy rằng thông tin của các pixel thường chỉ chịu tác động bởi các pixel ngay gần nó, vậy nên việc bỏ qua một số nút ở tầng đầu vào trong mỗi lần huấn luyện sẽ không làm giảm độ chính xác của mô hình. Vì thế người ta sử dụng cửa sổ tích chập nhằm giải quyết vấn đề số lượng tham số lớn mà vẫn trích xuất được đặc trưng của ảnh.



Hình 2.2. Quá trình hoạt động của mạng nơ-ron tích chập

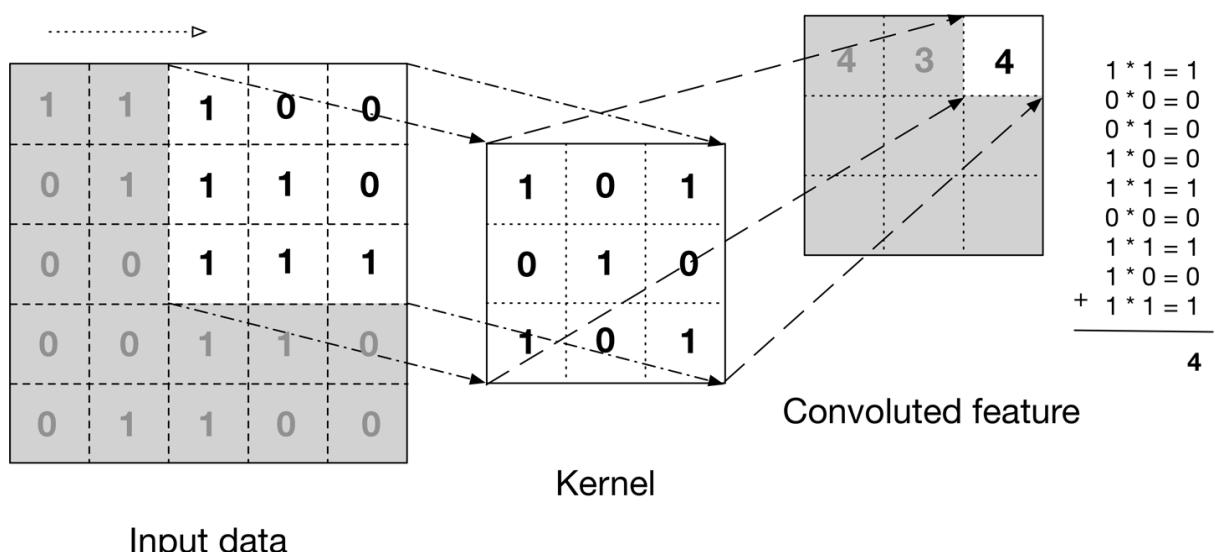
2.1.3. Cách hoạt động của CNN

Trước khi bắt đầu làm việc với CNN, chúng ta hãy tìm hiểu những điều cơ bản như hình ảnh là gì và nó được thể hiện như thế nào. Hình ảnh RGB là các ma trận chứa các giá trị pixel (điểm ảnh) có ba ma trận như vậy ứng với 3 màu R (Red - Đỏ), G (Green - Xanh lá cây) và B (Blue - Xanh da trời) trong khi ảnh xám chỉ có 1 ma trận duy nhất là cường độ sáng tại mỗi điểm ảnh. Hãy nhìn vào hình ảnh này để hiểu thêm.



Hình 2.3. Ảnh RGB

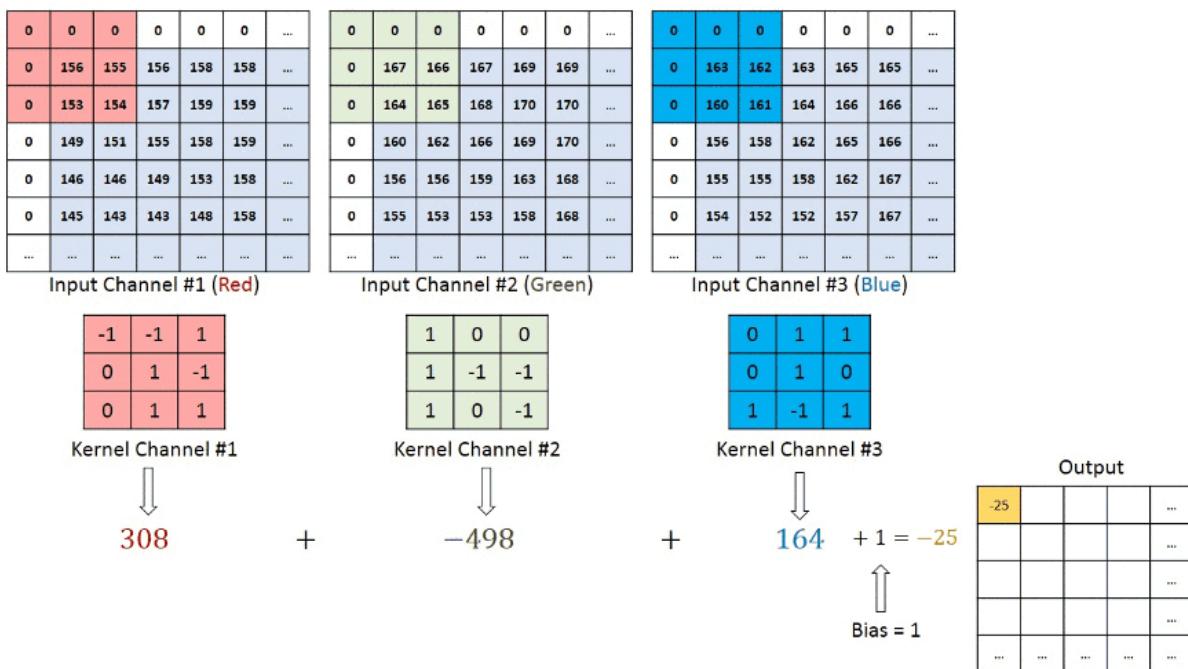
Để đơn giản, chúng ta chỉ cần sử dụng ảnh xám để tìm hiểu cách thức hoạt động của CNN.



Hình 2.4. Cách thức hoạt động của CNN

Hình ảnh trên cho định nghĩa tích chập. Chúng tôi lấy bộ lọc/hạt nhân (kernel) (ma trận 3×3) và áp dụng nó cho hình ảnh đầu vào để có được đặc trưng tích chập. Đặc trưng phức tạp này được chuyển sang lớp tiếp theo.

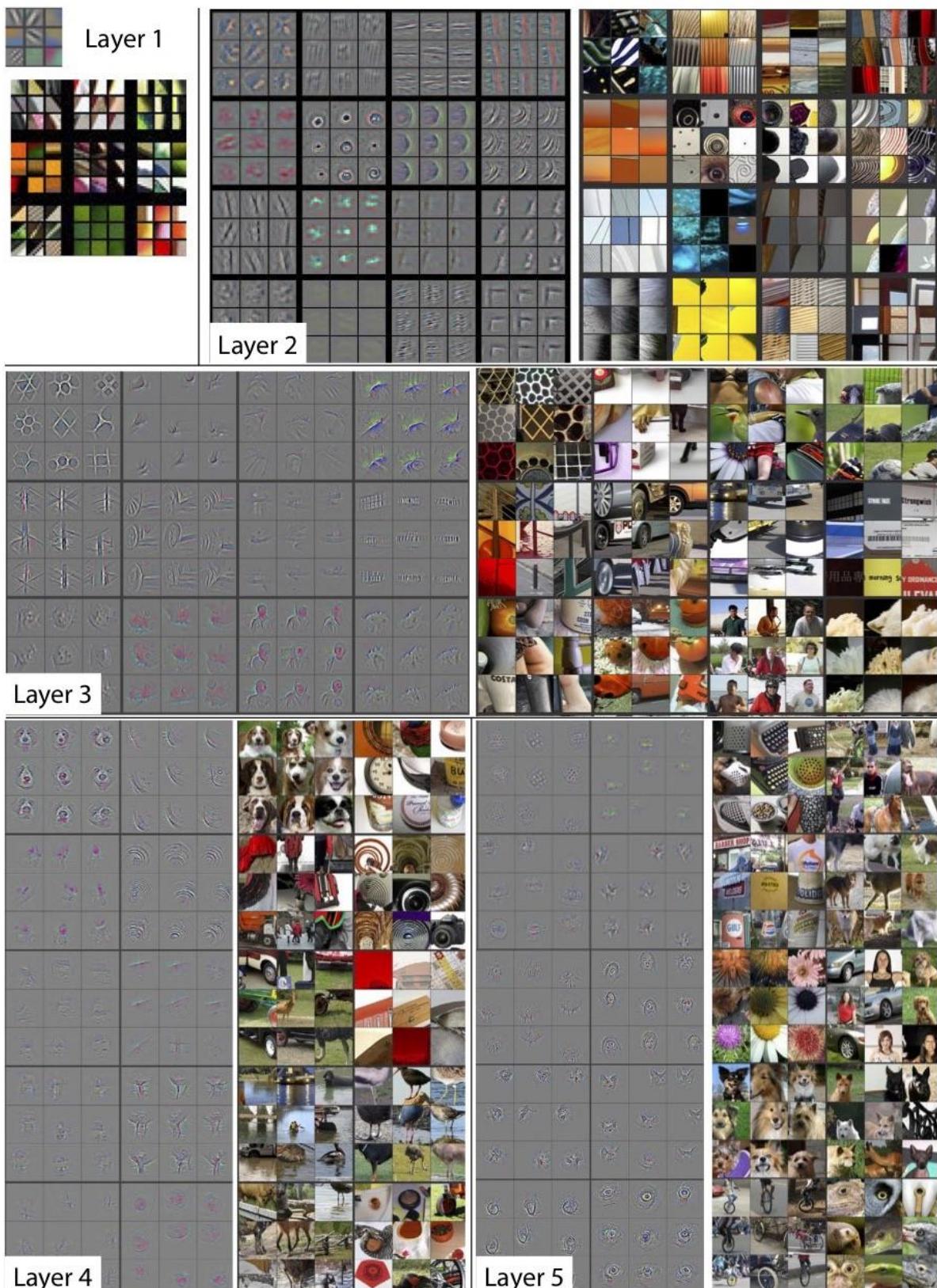
Trong trường hợp màu RGB, quan sát ảnh bên dưới để có thể hiểu cách thức hoạt động của nó.

*Hình 2.5. Cách hoạt động của CNN với ảnh RGB*

Mạng nơ-ron tích chập bao gồm nhiều lớp nơ-ron nhân tạo. Tế bào nơ-ron nhân tạo, mô phỏng sơ bộ các tế bào nơ-ron sinh học của chúng, là các hàm toán học tính toán tổng trọng số của nhiều đầu vào và đưa ra giá trị kích hoạt. Khi bạn nhập một hình ảnh vào ConvNet¹, mỗi lớp sẽ tạo ra một số chức năng kích hoạt(activation functions) được chuyển cho lớp tiếp theo.

Lớp đầu tiên thường trích xuất các đặc điểm cơ bản như các cạnh ngang hoặc chéo. Đầu ra này được chuyển sang lớp tiếp theo để phát hiện các đặc trưng phức tạp hơn như các góc hoặc các cạnh tổ hợp. Khi chúng ta đi sâu hơn vào các tầng mạng tiếp theo, nó có thể xác định các đặc trưng phức tạp hơn nữa như vật thể, khuôn mặt, v.v.

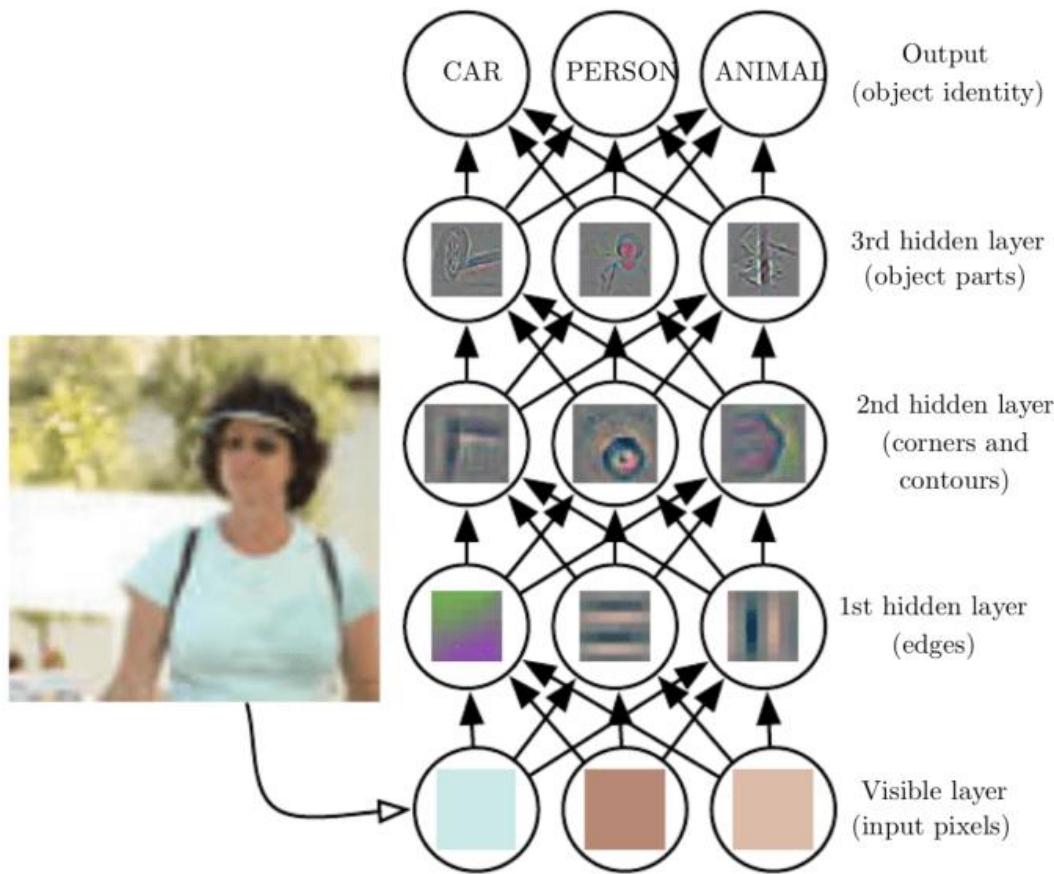
¹ ConvNet: Mạng nơ-ron tích chập hay CNN



Hình 2.6. Các lớp trong mạng nơ-ron tích chập

Dựa trên bản đồ kích hoạt của lớp chập cuối cùng, lớp phân loại đưa ra một tập hợp điểm tin cậy (giá trị từ 0 đến 1) xác định khả năng hình ảnh thuộc về một "lớp". Ví

đụ: nếu bạn có ConvNet phát hiện mèo, chó và ngựa thì đầu ra của lớp cuối cùng là khả năng hình ảnh đầu vào có chứa bất kỳ động vật nào trong số đó.



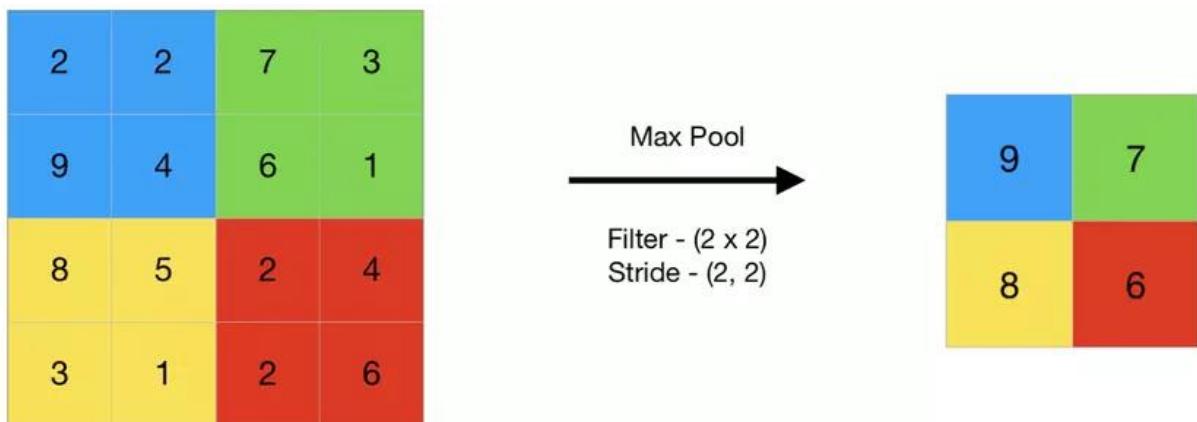
Hình 2.7. Quá trình chuyển đổi qua từng lớp

2.1.4. Tầng tổng hợp - Pooling Layer

Tầng tổng hợp thường được dùng giữa các tầng tích chập - convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model. Trong quá trình này, quy tắc về stride²(bước nhảy) và padding³(đệm) áp dụng như phép tính convolution trên ảnh.

² *stride*: stride là bước nhảy giữa các vị trí mà kernel (hoặc filter) của phép tích chập di chuyển trên đầu vào.

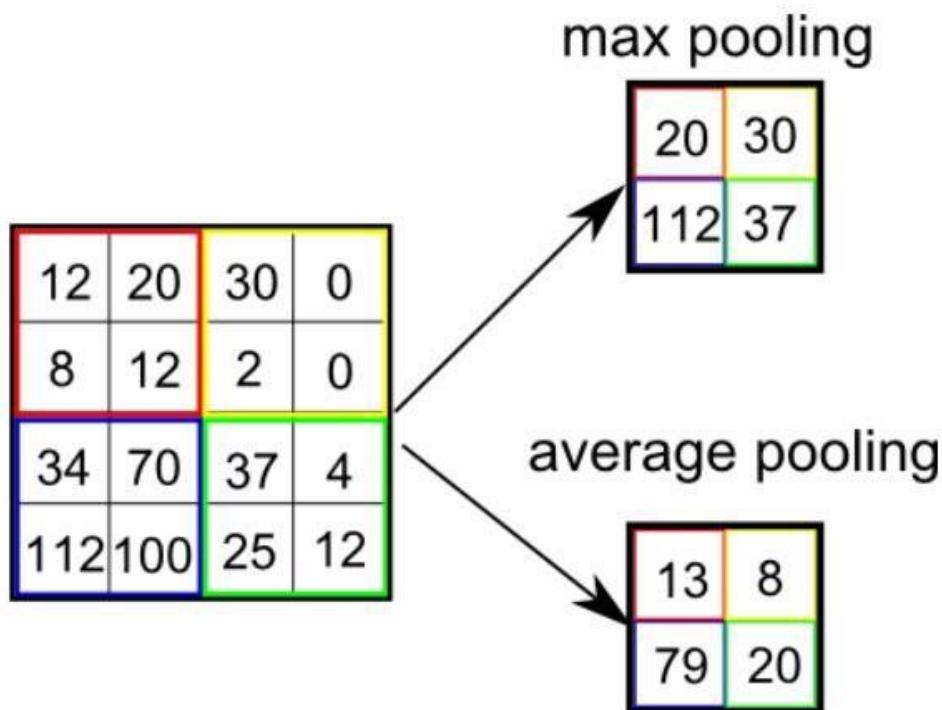
³ *padding*: padding là quá trình thêm các giá trị 0 (hoặc giá trị cụ thể khác) xung quanh biên của đầu vào trước khi thực hiện phép tích chập.



Hình 2.8. Ví dụ về một hàm pooling là Max Pooling của tầng tổng hợp - pooling layer

Vì vậy, những gì chúng ta làm trong Max Pooling là chúng ta tìm giá trị tối đa của pixel từ một phần hình ảnh được bao phủ bởi hạt nhân. Nó loại bỏ hoàn toàn các kích hoạt nhiễu và cũng thực hiện khử nhiễu cùng với việc giảm kích thước.

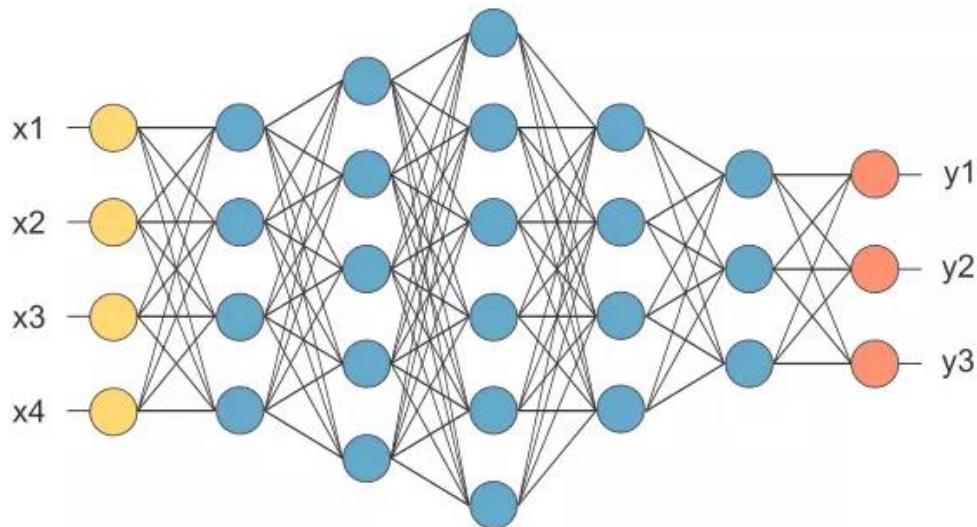
Một hàm pooling khác, Average Pooling trả về giá trị trung bình của tất cả các giá trị từ phần hình ảnh được Kernel bao phủ. Tổng hợp trung bình chỉ đơn giản thực hiện giảm kích thước như một cơ chế khử nhiễu. Do đó, chúng ta có thể nói rằng Max Pooling hoạt động tốt hơn rất nhiều so với Average Pooling.



Hình 2.9. Hàm max pooling và average pooling

2.1.5. Tầng kết nối đầy đủ - Fully Connected Layer

Sau khi ảnh được truyền qua nhiều tầng tích chập và tầng tổng hợp mô hình CNN khi đó đã học được các đặc trưng của ảnh thì tensor của đầu ra của tầng cuối cùng sẽ được là phẳng thành vector và đưa vào một lớp được kết nối như một mạng nơ-ron cơ bản. Với FC tầng được kết hợp với các đặc trưng lại với nhau để tạo ra một mô hình. Cuối cùng sử dụng softmax hoặc sigmoid để phân loại đầu ra.

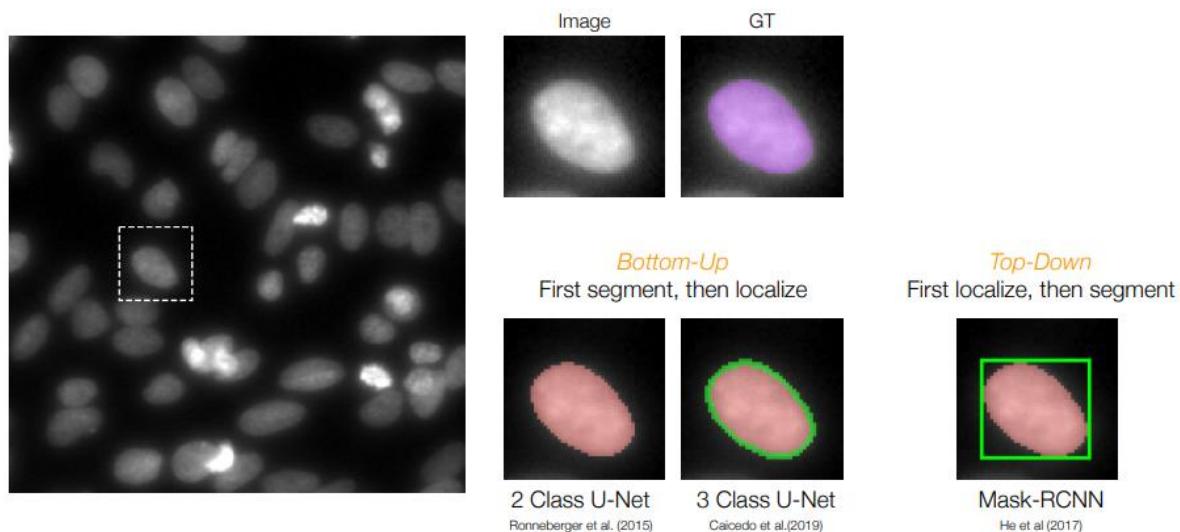


Hình 2.10. Tầng kết nối đầy đủ trong mạng nơ-ron tích chập

Sử dụng mạng nơ tron tích chập như là một công cụ trong nhận dạng đối tượng, có 2 cách tiếp cận dưới đây để tiến hành nhận dạng.

2.2. Phương pháp tiếp cận Bottom-Up (từ dưới lên)

2.2.1. Giới thiệu về cách tiếp cận Bottom-Up



Hình 2.11. Các tiếp cận từ Bottom-up(từ dưới lên)

Phương pháp tiếp cận Bottom-Up(từ dưới lên) trong phân vùng các phần tử bắt đầu từ mức độ thấp nhất của đặc trưng hình ảnh và dần dần xây dựng để nhận diện và phân đoạn từng phần tử riêng lẻ. Phương pháp này thường dựa vào thông tin pixel và sử dụng nhiều kỹ thuật khác nhau để nhóm pixel thành các vùng có ý nghĩa tương ứng với các phần tử.

Dưới đây là tóm tắt sơ lược về phương pháp tiếp cận Bottom-Up để phân đoạn các đối tượng:

Tiền xử lý:

Nâng cao hình ảnh: Áp dụng các kỹ thuật tiền xử lý để nâng cao chất lượng hình ảnh, kéo dãn độ tương phản, cân bằng lược đồ xám hoặc lọc.

Giảm nhiễu: Sử dụng các phương pháp giảm nhiễu, như làm mờ Gaussian hoặc lọc trung bình, để cải thiện tỷ lệ tín hiệu và nhiễu.

Phân đoạn hình ảnh (Segmentation Image):

Nguõng hóa: Sử dụng các phương pháp nguõng dựa trên độ sáng để phân chia hình ảnh thành các khu vực nền và khu vực chính. Điều này giúp làm nổi bật các khu vực có khả năng chứa hạt nhân.

Phép toán hình thái: Áp dụng các phép toán hình thái (ví dụ: co, mở rộng) để làm sạch các khu vực phân đoạn và tách các hạt nhân gần nhau.

Trích xuất đặc trưng:

Đặc trưng hình ảnh và kích thước: Trích xuất các đặc trưng như diện tích, chu vi, độ tròn và độ lệch để mô tả hình dạng và kích thước của các khu vực đã phân đoạn.

Đặc trưng độ sáng: Thu thập thông tin liên quan đến độ sáng trong mỗi khu vực, như độ sáng trung bình hoặc đặc trưng cấu trúc.

Hợp nhất và chia nhỏ khu vực:

Hợp nhất khu vực: Kết hợp các khu vực liền kề dựa trên các tiêu chí nhất định, như tương đồng về độ sáng hoặc hình dạng, để tạo ra các hạt nhân lớn, đầy đủ hơn.

Chia nhỏ khu vực: Chia nhỏ các khu vực có khả năng đại diện cho nhiều hạt nhân bằng cách xem xét các đặc trưng như biến độ sáng hoặc không đồng đều về hình dạng.

Sau xử lý:

Lọc: Áp dụng bộ lọc bổ sung để loại bỏ các kết quả dương giả hay nhiễu.

Sửa chữa: Sử dụng các phương pháp học máy hoặc dựa trên quy tắc để tinh chỉnh kết quả phân đoạn thêm nữa.

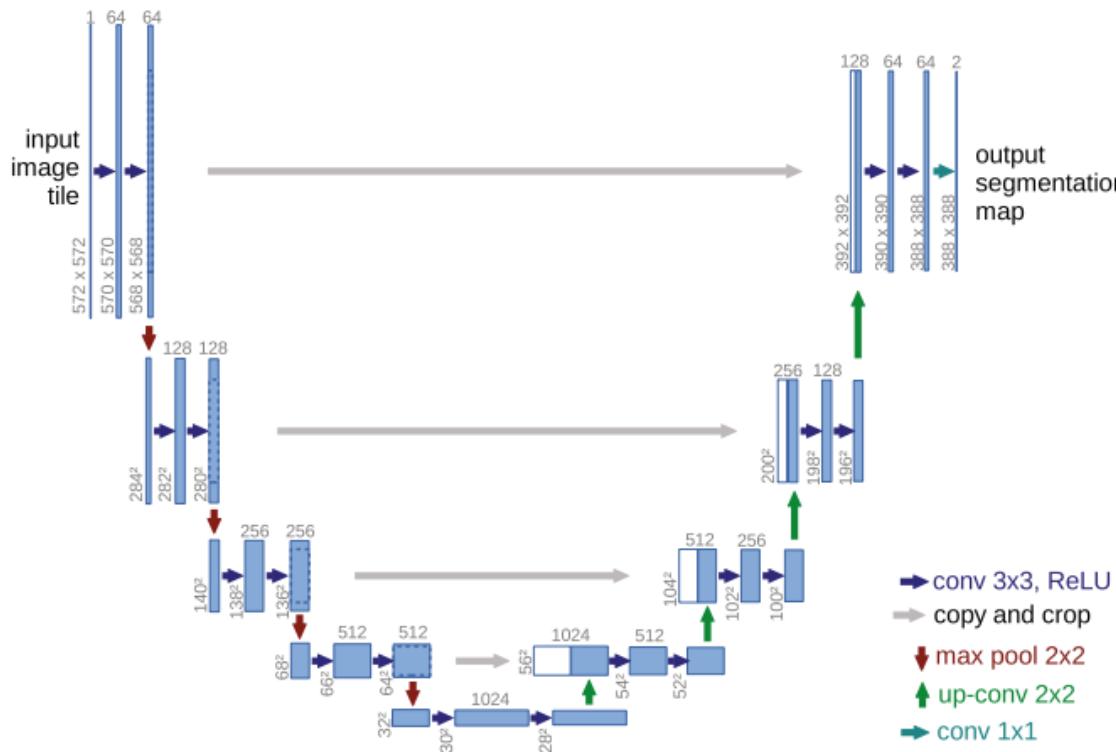
Xác nhận:

So sánh với dữ liệu chuẩn - Ground Truth: So sánh hạt nhân đã phân đoạn với dữ liệu chuẩn để đánh giá độ chính xác của quá trình phân đoạn.

Tối ưu hóa lặp lại: Nếu cần, lặp lại quá trình, điều chỉnh các tham số và phương pháp dựa trên kết quả xác nhận.

Rất phổ biến, người ta thường hay sử dụng mạng U-Net trong để phân vùng ảnh. Dưới đây là cơ bản về mạng U-Net.

2.2.2 Mạng U-Net



Hình 2.12. Mạng nơ-ron U-net

Cơ bản về mạng U-Net

Trong những năm gần đây, mạng nơ-ron đã đóng một vai trò quan trọng trong việc giải quyết các vấn đề phức tạp trong lĩnh vực xử lý hình ảnh. Trong ngữ cảnh của các ứng dụng y sinh học, đặc biệt là trong phân đoạn hình ảnh vi sinh học và y học, việc nhận diện và phân loại các cấu trúc tế bào là một thách thức lớn. Mô hình phân đoạn cần phải hiệu quả trong việc giữ lại thông tin vị trí cụ thể của các cấu trúc tế bào và đồng thời xử lý những hình ảnh có độ phân giải cao và chi tiết.

Mạng nơ-ron U-Net là một kiến trúc mạng sử dụng trong lĩnh vực xử lý hình ảnh và phân đoạn hình ảnh, đặc biệt phổ biến trong các ứng dụng y sinh học như phân đoạn tế bào trong hình ảnh vi sinh học. Được giới thiệu bởi Olaf Ronneberger, Philipp Fischer và Thomas Brox vào năm 2015, U-Net nhanh chóng trở thành một trong những kiến trúc phổ biến cho các nhiệm vụ phân đoạn.

Trong quá khứ, các mô hình truyền thống thường gặp khó khăn khi đối mặt với các thách thức này. Các mô hình phổ biến như mạng tích chập không gian (CNN) thường

chỉ đưa ra đầu ra phân đoạn tổng quát mà không giữ lại thông tin về vị trí cụ thể của các đối tượng trong hình ảnh. Đồng thời, kích thước lớn của các hình ảnh y sinh học và số lượng hạn chế của dữ liệu huấn luyện cũng làm giảm độ chính xác của các mô hình truyền thống.

Kiến trúc mạng U-net:

U-Net, giới thiệu bởi Olaf Ronneberger và đồng nghiệp vào năm 2015, là một kiến trúc mạng nơ-ron đặc biệt được thiết kế để giải quyết những thách thức trong phân đoạn hình ảnh y sinh học. Cấu trúc chữ U đặc biệt của U-Net bao gồm hai phần chính: Encoder (Thu nhỏ) và Decoder (Mở rộng). Sự kết hợp giữa các lớp thu nhỏ và mở rộng giúp U-Net duy trì thông tin chi tiết cấu trúc và vị trí của các cấu trúc tế bào.

Đặc điểm nổi bật:

Skip Connections: Một điểm đặc đáo của U-Net là việc sử dụng các "skip connections" giữa các lớp Encoder và Decoder. Các kết nối này cho phép thông tin chi tiết từ các lớp thu nhỏ được chuyển đến các lớp mở rộng, giúp duy trì độ chính xác về vị trí và cấu trúc.

Tích chập 1x1: U-Net sử dụng các lớp tích chập với kernel size 1x1 để giảm số lượng kênh đặc trưng mà không làm mất thông tin quan trọng.

Ứng dụng và hiệu suất:

U-Net đã được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

Phân đoạn tế bào: Nhận diện và phân loại cấu trúc tế bào trong hình ảnh y sinh học.

Nhận dạng vật thể: Phân đoạn và nhận dạng đối tượng trong hình ảnh.

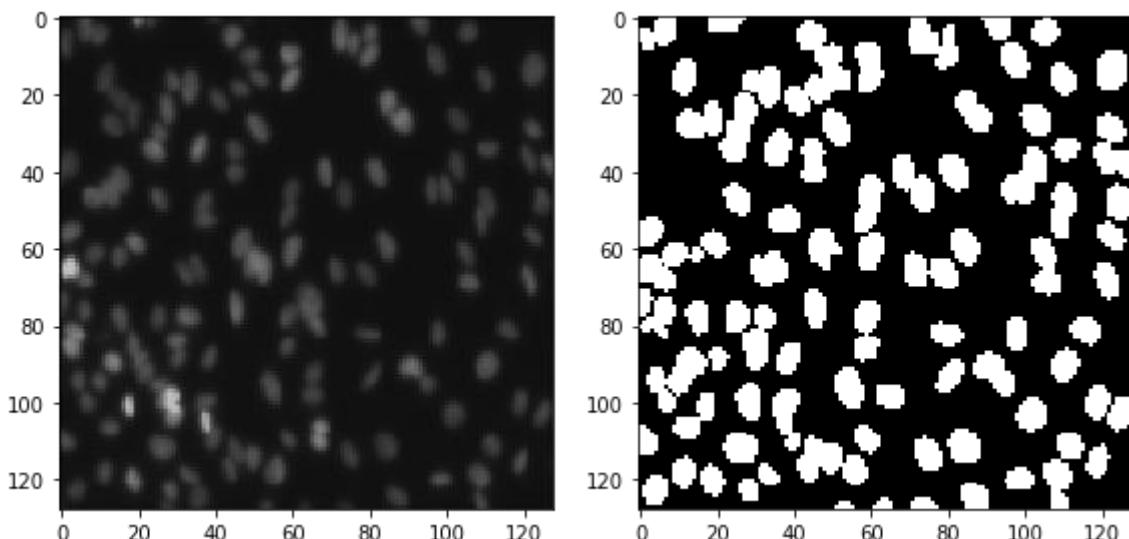
Xử lý hình ảnh y tế: Ứng dụng trong việc phân đoạn cơ quan và cấu trúc trong cơ thể.

U-Net thường xuất sắc trong việc giữ lại thông tin vị trí chi tiết của các cấu trúc tế bào, đồng thời có khả năng xử lý hiệu quả với những hình ảnh có độ phân giải lớn. Các "skip connections" và thiết kế đặc biệt của U-Net đã giúp nó trở thành một trong những lựa chọn hàng đầu cho các nhiệm vụ phân đoạn hình ảnh y sinh học.

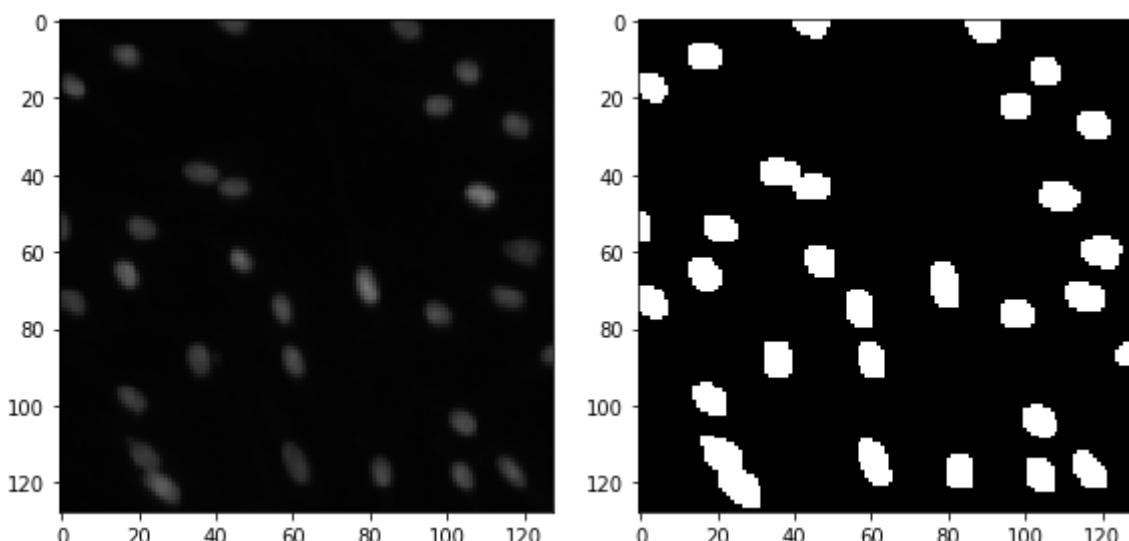
Có thể nói mạng nơ-ron U-Net đại diện cho một đột phá quan trọng trong lĩnh vực xử lý hình ảnh y sinh học. Với kiến trúc linh hoạt và hiệu suất xuất sắc, U-Net không chỉ giải quyết những thách thức cụ thể trong việc phân đoạn hình ảnh y sinh học mà còn có ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

2.2.3. Ví dụ kết quả nhận dạng đối tượng với cách tiếp cận Bottom-Up

Phân đoạn hạt nhân với mô hình mạng U-net, với tập dữ liệu Kaggle's Data Science Bowl 2018⁴. Mô hình với kiến trúc mạng bao gồm việc lặp 2 phép tích chập 3×3 (phép tích chập không đệm), mỗi phép tích chập được theo sau một hàm kích hoạt tuyến tính(ReLU) và một phép lấy giá trị lớn nhất 2×2 với bước nhảy 2 để thu nhỏ. Tại mỗi bước thu nhỏ, mô hình tăng gấp đôi số kênh đặc trưng. Tại lớp cuối cùng một phép tích chập 1×1 được sử dụng để ánh xạ mỗi vectơ đặc trưng thành số lớp mong muốn.



Hình 2.13. Dữ liệu gốc và mask



Hình 2.14. Dữ liệu gốc và dữ liệu dự đoán

⁴ <https://www.kaggle.com/code/advaitasave/tensorflow-2-nuclei-segmentation-unet/input>

2.3. Phương pháp tiếp cận Top-Down

2.3.1. Giới thiệu về cách tiếp cận Top-Down

Phương pháp tiếp cận từ trên xuống (Top-Down) trong phân đoạn đối tượng tập trung vào việc xác định các đặc điểm cấp cao và sử dụng chúng để hướng dẫn quá trình phân đoạn. Đây thường là các phương pháp dựa trên học máy, đặc biệt là mô hình học sâu (deep learning). Dưới đây là một phác thảo về cách phương pháp Top-Down có thể được triển khai:

Thu thập dữ liệu và chuẩn bị dữ liệu so sánh chuẩn

Huấn luyện dữ liệu: Xây dựng tập dữ liệu huấn luyện có nhãn chính xác cho việc học mô hình.

Chuẩn bị dữ liệu so sánh chuẩn: Tạo bản đồ cơ sở dữ liệu với thông tin về vị trí và biên của các đối tượng trong ảnh.

Học mô hình

Mô hình học sâu (deep learning): Sử dụng mô hình học sâu như Convolutional Neural Network (CNN) hoặc U-Net để học cách phân biệt giữa các loại cấu trúc trong hình ảnh, đặc biệt là các loại đối tượng cần quan tâm.

Học chuyển giao: Nếu có sẵn dữ liệu huấn luyện lớn, có thể sử dụng học chuyển giao từ mô hình đã được huấn luyện trước đó trên dữ liệu lớn hơn.

Áp dụng mô hình học sâu vào hình ảnh mới

Dự đoán hạt nhân: Sử dụng mô hình đã được huấn luyện để dự đoán vị trí và đường biên của các hạt nhân trong hình ảnh mới.

Hợp nhất và tối ưu hóa

Hợp nhất đối tượng: Sử dụng kỹ thuật như kết hợp kết quả từ mô hình với các kỹ thuật phân đoạn truyền thống để tạo ra kết quả cuối cùng.

Tối ưu hóa: Làm sạch kết quả, loại bỏ các dự đoán giả mạo và thực hiện các bước tối ưu hóa để cải thiện chính xác.

Xác nhận và đánh giá

So sánh với dữ liệu chuẩn: Đánh giá kết quả phân đoạn bằng cách so sánh với dữ liệu chuẩn hoặc các dữ liệu đánh giá được nhận định trước.

Tối ưu hóa thêm khi cần thiết: Nếu kết quả chưa đạt được mong muốn, có thể điều chỉnh mô hình hoặc thực hiện các bước tối ưu hóa thêm.

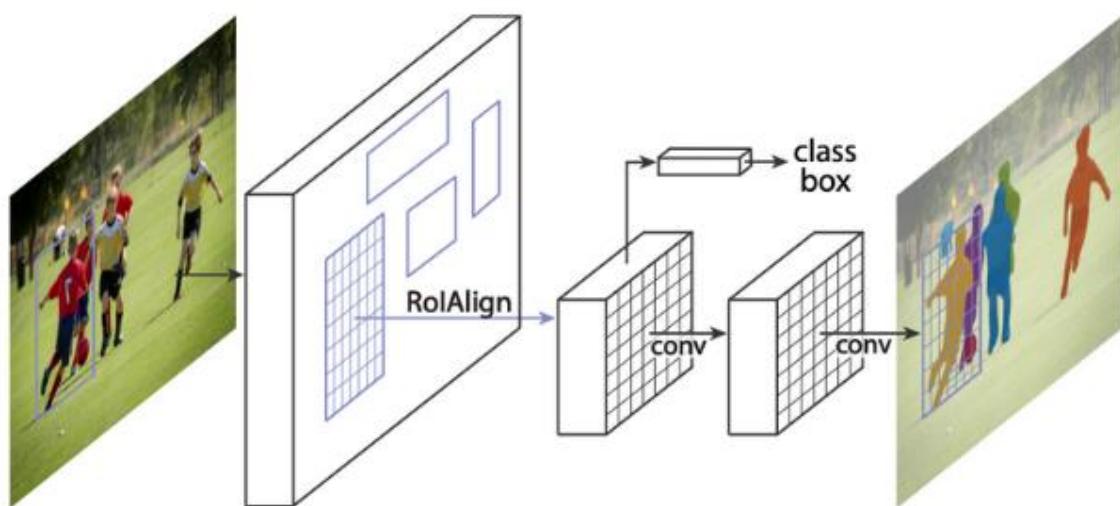
Triển khai và sử dụng

Triển khai mô hình: Áp dụng mô hình đã được huấn luyện vào các hình ảnh mới để thực hiện phân đoạn hạt nhân.

Kiểm soát và cập nhật: Theo dõi hiệu suất của mô hình trong thời gian và cập nhật nó khi cần thiết.

Trong các bài toán phân loại các đối tượng dạng tế bào hay bọt khí theo cách tiếp cận Top-Down, mạng Mask R-CNN (Mask Region-Based CNN) được sử dụng rộng rãi. Cơ bản về mạng này được trình bày ngắn gọn dưới đây.

2.3.2. Mạng Mask-RCNN



Hình 2.15. Mạng nơ-ron Mask R-CNN

Trong lĩnh vực thị giác máy tính và xử lý hình ảnh, việc nhận diện và phân loại đối tượng cùng với việc xác định vị trí của chúng là một thách thức quan trọng. Mạng Mask R-CNN xuất hiện như một đối tượng nổi bật trong việc giải quyết vấn đề này, đặc biệt là trong các ứng dụng như nhận diện đối tượng và phân đoạn hình ảnh.

Các mô hình truyền thống trước đây thường chỉ tập trung vào việc phân loại đối tượng và xác định vị trí của chúng, bỏ qua khả năng phân đoạn chi tiết từng phần của đối tượng. Mask R-CNN được phát triển để đối mặt với thách thức này bằng cách kết hợp khả năng phân loại, xác định vị trí và phân đoạn mặt của đối tượng trong một lần chạy.

Kiến Trúc Mask R-CNN

Mask R-CNN xây dựng trên cơ sở của Faster R-CNN, mô hình nổi tiếng về phát hiện đối tượng. Mô hình chia thành ba phần chính:

Backbone Network: Sử dụng một mạng tích chập để trích xuất đặc trưng từ hình ảnh đầu vào. Thông thường, các mô hình như ResNet hoặc ResNeXt được sử dụng làm backbone.

Region Proposal Network (RPN): Tạo ra các ô đề xuất (proposals) cho các vùng chứa đối tượng trong hình ảnh.

ROI Align và Heads: Gồm các phần xử lý vùng quan tâm (ROI) để thực hiện nhiệm vụ phân loại, xác định vị trí và phân đoạn.

Mask Head

Phần quan trọng nhất của Mask R-CNN là Mask Head, nơi mà mạng tạo ra các mặt nạ chi tiết cho từng đối tượng. Nó sử dụng một kiến trúc chuyển giao (transferring) từ đặc trưng của ROI để dự đoán mặt nạ pixel-wise.

Quy Trình Hoạt Động

Đề Xuất Vùng: RPN được sử dụng để đề xuất vùng chứa đối tượng trong hình ảnh. Những đề xuất này sau đó được sử dụng để tạo ra các vùng quan tâm (ROIs) cho việc xác định vị trí và phân loại.

Xác Định Vị Trí và Phân Loại: Sau khi có ROIs, mô hình sử dụng các đầu phân loại và xác định vị trí để xác định lớp của đối tượng và vị trí của chúng trong hình ảnh.

Phân Đoạn Mặt: Cuối cùng, Mask Head được áp dụng để tạo ra mặt nạ chi tiết cho từng đối tượng, phân đoạn chính xác các phần khác nhau của đối tượng.

Ứng Dụng và Hiệu Suất

Các Ứng Dụng: Mask R-CNN đã được triển khai rộng rãi trong nhiều ứng dụng, bao gồm nhận diện đối tượng trong hình ảnh, phân đoạn hình ảnh y sinh học, nhận dạng vật thể trong video, và nhiều ứng dụng thị giác máy tính khác.

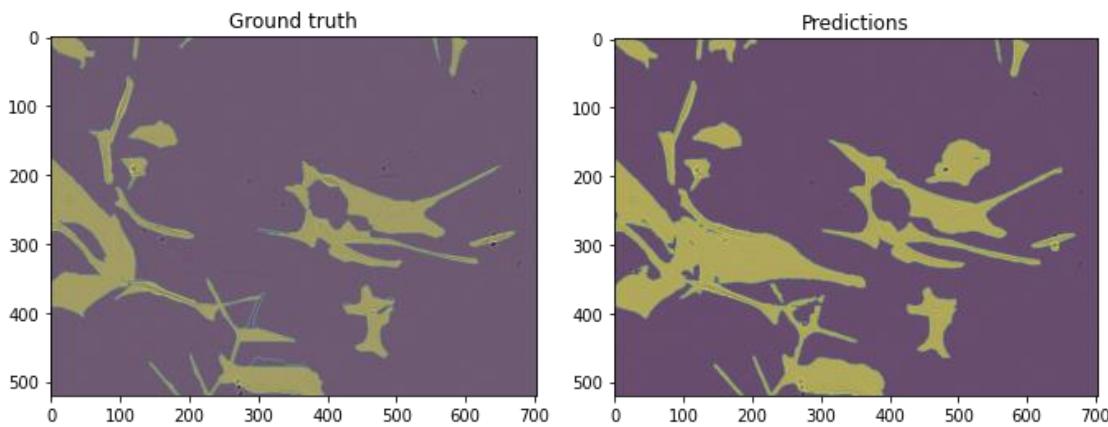
Hiệu Suất: Mask R-CNN thường xuất sắc trong việc đạt được độ chính xác cao, đặc biệt là trong các nhiệm vụ đòi hỏi phân đoạn chi tiết đối tượng. Sự kết hợp linh hoạt của nó giữa phân loại, xác định vị trí và phân đoạn mặt làm cho nó trở thành một trong những mô hình đáng chú ý nhất trong lĩnh vực thị giác máy tính.

Mạng Mask R-CNN đánh dấu một bước tiến lớn trong phát triển mô hình đa nhiệm cho việc nhận diện đối tượng và phân đoạn hình ảnh. Với khả năng tạo ra mặt nạ

chi tiết, nó đáp ứng đòi hỏi ngày càng tăng về độ chính xác trong các ứng dụng như y sinh học, thị giác máy tính, và nhiều lĩnh vực khác.

2.3.3. Ví dụ về nhận dạng đối tượng theo cách tiếp cận Top-Down

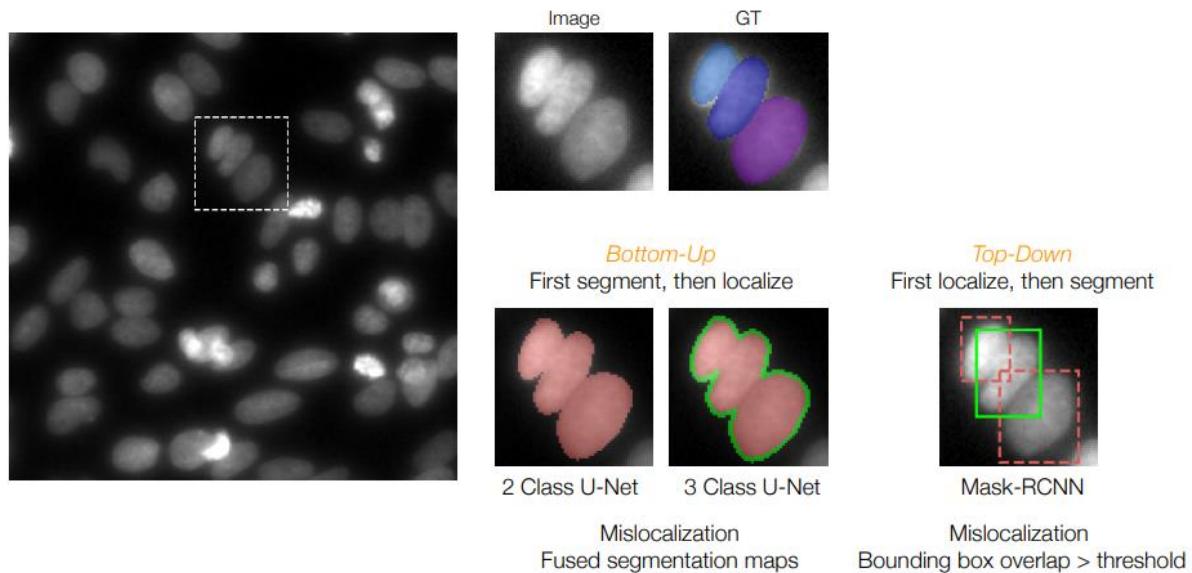
Nhận dạng tế bào thần kinh, phục vụ mục đích sinh học và y tế, giúp phát hiện và điều trị các bệnh lý về thần kinh như Alzheimer... Sử dụng mô hình mạng Mask R-CNN với bộ dữ liệu Satorius - Cell Instance Segmentation⁵.



Hình 2.16. Dữ liệu thực và dự đoán(Mask R-CNN)

⁵ <https://www.kaggle.com/code/nikmarker/sartorius-starter-torch-mask-r-cnn-lb-0-273/input>

2.4. So sánh 2 cách tiếp cận



Hình 2.17. So sánh 2 cách tiếp cận Top-down và Bottom-up

2.4.1. Top-down

Khó khăn trong việc xử lý tắc nghẽn: Phương pháp tiếp cận Top-down thường gặp khó khăn trong việc xử lý các điểm tắc khi các đối tượng chồng lên nhau. Khi các đối tượng được đóng gói dày đặc hoặc chồng lên nhau đáng kể, việc xác định và phân chia chính xác các đối tượng riêng lẻ có thể gặp khó khăn.

Tính toán chuyên sâu: Phương pháp tiếp cận Top-down thường liên quan đến việc xử lý toàn bộ hình ảnh để xác định và định vị các đối tượng, có thể cần nhiều tính toán. Điều này có thể trở thành nút thắt cổ chai, đặc biệt với các tập dữ liệu lớn hoặc hình ảnh có độ phân giải cao.

Khó khăn trong việc khởi tạo: Nhiều phương pháp từ trên xuống dựa vào các khởi tạo hoặc giả định về cảnh và những giả định này có thể không phải lúc nào cũng đúng, đặc biệt là trong các ảnh dày đặc. Độ nhạy với các điều kiện ban đầu có thể dẫn đến lỗi phân đoạn.

2.4.2. Bottom-up

Khó khăn trong việc hiểu toàn bộ bối cảnh: Cách tiếp cận từ dưới lên tập trung vào các đặc điểm địa phương và có thể gặp khó khăn trong việc nắm bắt bối cảnh toàn

cục của một cảnh. Trong các kịch bản mật độ cao, việc hiểu mối quan hệ giữa các đối tượng là rất quan trọng và các phương pháp thuần túy từ dưới lên có thể gặp khó khăn này.

Phân đoạn quá mức: Các phương pháp từ dưới lên có thể dễ bị phân đoạn quá mức, đặc biệt khi xử lý các cảnh phức tạp. Điều này có thể dẫn đến việc chia các đối tượng thành các phân đoạn nhỏ hơn, khiến việc hợp nhất chúng một cách chính xác thành các đối tượng có ý nghĩa trở nên khó khăn.

Nhận dạng đối tượng hạn chế: Cách tiếp cận từ dưới lên có thể có những hạn chế trong việc nhận dạng các lớp đối tượng cụ thể. Trong các kịch bản mật độ cao, nơi có sự đa dạng của các đối tượng, việc xác định và phân loại chính xác từng đối tượng trở nên quan trọng và các phương pháp từ dưới lên có thể thiếu sót ở khía cạnh này.

Khó khăn trong việc xử lý các kích cỡ đa dạng: Ảnh mật độ cao thường liên quan đến các vật thể có kích thước và quy mô khác nhau. Các phương pháp tiếp cận từ dưới lên có thể gặp khó khăn trong việc xử lý những biến thể này một cách hiệu quả, dẫn đến sự thiếu chính xác trong phân khúc.

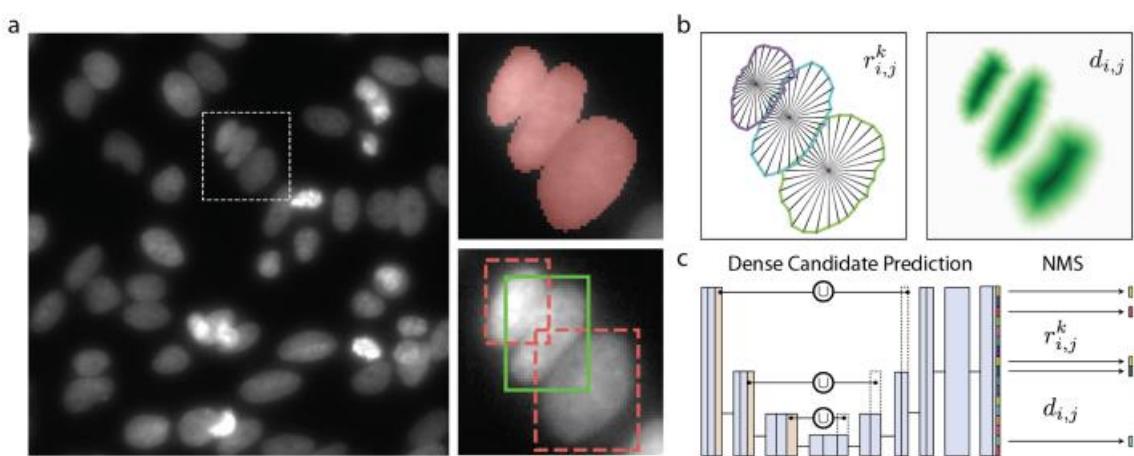
2.4.3. Lựa chọn cách tiếp cận/mô hình

Với những bài toán cần phân đoạn, nhận diện tế bào, việc cần nhận diện chính xác từng tế bào trong 1 ảnh, với những ảnh có mật độ dày đặc là thực sự cần thiết. Với cách tiếp cận từ dưới lên, với ảnh mật độ dày đặc, rất dễ xảy ra xót tế bào khi các tế bào chồng chéo nhau. Vì vậy cách tiếp cận hợp lý với phân đoạn tế bào cho ảnh mật độ dày đặc là cách tiếp cận từ trên xuống (top-down), tuy nhiên cách tiếp cận từ trên xuống vẫn còn một vài điểm thiếu xót. Cho nên cách tiếp cận top-down với mô hình CNN cải tiến Stardist và sẽ được trình bày tiếp theo đây

CHƯƠNG III. PHƯƠNG PHÁP TIẾP CẬN CỦA ĐỒ ÁN.

3.1. Mô hình Stardist

Về cơ bản StarDist thuộc cách tiếp cận Top-Down. Ví dụ như *Hình 3.1* ở dưới. Các đối tượng cần nhận dạng sẽ được xác định thuộc một đa giác thay vì 1 hộp giới hạn thường được dùng trong các phương pháp khác.



Hình 3.1. Mô hình StarDist

3.1.1. Tổng quan về Stardist

Nhiều công việc trong lĩnh vực sinh học đòi hỏi việc phát hiện và phân đoạn chính xác các tế bào và nhân từ hình ảnh vi kính. Các ví dụ bao gồm việc sàng lọc các biến thể trong các hiện tượng của tế bào, hoặc xác định các dòng phân tử phân chia của tế bào. Trong nhiều trường hợp, mục tiêu là có được một phân đoạn đối tượng, tức là gán một định danh thể hiện của tế bào cho mỗi điểm ảnh của hình ảnh. Với mục đích đó, một phương pháp phổ biến từ dưới lên - Bottom Up là trước hết phân loại từng điểm ảnh thành các lớp ngữ nghĩa (như tế bào hoặc nền) và sau đó nhóm các điểm ảnh thuộc cùng một lớp thành các thể hiện riêng lẻ. Bước đầu tiên thường được thực hiện với các bộ phân loại đã học, chẳng hạn như rừng ngẫu nhiên (*random forest*) hoặc mạng neural. Việc nhóm điểm ảnh có thể được thực hiện bằng cách tìm các thành phần kết nối. Mặc dù phương pháp này thường mang lại kết quả tốt, nhưng nó gặp vấn đề khi xử lý hình

ánh tế bào với mật độ dày đặc, vì chỉ cần một vài điểm ảnh bị phân loại sai có thể làm cho các thể hiện của tế bào liền kề nhưng khác nhau bị hợp nhất.

Một phương pháp thay thế từ trên xuống - Top Down là trước hết xác định vị các thể hiện tế bào cá thể với một biểu diễn hình dạng thô và sau đó làm tinh chỉnh hình dạng trong một bước bổ sung. Để làm được điều này, các phương pháp phát hiện đối tượng hiện đại nhất chủ yếu dự đoán các hộp giới hạn căn theo trục tọa độ của ảnh, có thể được làm tinh chỉnh để có được một phân đoạn thể hiện bằng cách phân loại các điểm ảnh bên trong mỗi hộp. Hầu hết các phương pháp này có điểm chung là tránh phát hiện đối tượng giống nhau nhiều lần bằng cách thực hiện một bước *non-maximum suppression*⁶(NMS) nơi các hộp với độ tin cậy thấp bị loại bỏ bởi hộp với độ tin cậy cao hơn nếu chúng chồng lấn đáng kể. NMS có thể gây vấn đề nếu các đối tượng quan tâm được biểu diễn kém bằng các hộp giới hạn căn theo trực, điều này có thể xảy ra với nhân tế bào. Mặc dù điều này có thể được giảm nhẹ bằng cách sử dụng các hộp giới hạn xoay, vẫn cần phải làm tinh chỉnh hình dạng hộp để mô tả chính xác các đối tượng như nhân tế bào.

Để giảm nhẹ các vấn đề đã đề cập, StarDist đã được đề xuất, một phương pháp phát hiện tế bào dự đoán một biểu diễn hình dạng linh hoạt đủ để - mà không cần làm tinh chỉnh - độ chính xác của việc định vị có thể cạnh tranh với các phương pháp phân đoạn thể hiện. Để làm được điều này, StarDist sử dụng đa giác sao lồi - star-convex polygon⁷ phù hợp với xấp xỉ hình dạng thường tròn của nhân tế bào trong hình ảnh vi kính, cũng có thể áp dụng cho các hình dạng của các hạt nhân như kim cương, bọt khí.

StarDist là một mô hình học máy được thiết kế để thực hiện nhiệm vụ phân đoạn tế bào trong ảnh vi sinh học, đặc biệt là trong lĩnh vực fluorescence microscopy images (ảnh vi sinh nhiều màu sắc). Mô hình này sử dụng đa giác lồi hình ngôi sao để biểu diễn hình dạng của các tế bào, thay vì sử dụng hộp giới hạn truyền thống.

⁶ *non-maximum suppression*: là một kỹ thuật quan trọng, được sử dụng để loại bỏ các hộp giới hạn dư thừa hoặc không chính xác sau khi các dự đoán đã được thực hiện. NMS giúp xác định hộp giới hạn duy nhất cho mỗi đối tượng trong hình ảnh, cải thiện độ chính xác và hiệu quả của việc phát hiện đối tượng.

⁷ *star-convex polygon*: là một đa giác mà tồn tại 1 tâm(kernel) có thể nối tất cả các biên của đa giác

Một số đặc điểm quan trọng về StarDist

Biểu diễn hình dạng đa giác sao-lồi: StarDist sử dụng đa giác sao-lồi để biểu diễn hình dạng của tế bào. Đa giác sao-lồi này được dự đoán cho từng pixel trong ảnh, thay vì sử dụng hộp giới hạn, giúp mô hình mô phỏng chính xác hình dạng của tế bào.

Mạng nơ ron tích chập (CNN): Mô hình sử dụng kiến trúc mạng nơ-ron tích chập (CNN) để học các đặc trưng từ ảnh và dự đoán đa giác lồi tương ứng cho từng pixel. CNN giúp mô hình tự động học được các đặc trưng phức tạp từ dữ liệu huấn luyện.

Hàm mất mát đa giác sao lồi (Star-convex Loss): Để huấn luyện mô hình, StarDist sử dụng một hàm mất mát được thiết kế đặc biệt cho các đa giác lồi, giúp tối ưu hóa việc dự đoán đa giác lồi cho các tế bào.

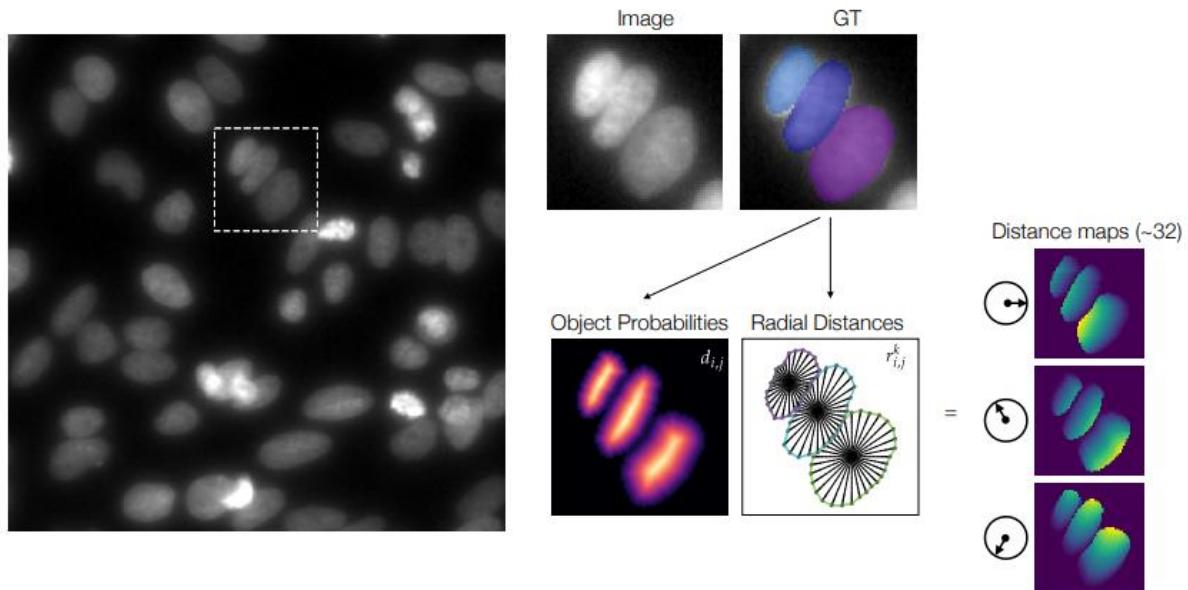
Ứng dụng trong ảnh vi sinh học: StarDist được phát triển để giải quyết các thách thức trong việc phân đoạn tế bào trong ảnh vi sinh học, đặc biệt là khi tế bào đang đông đúc và có hình dạng phức tạp.

Hiệu suất trên nhiều datasets: Mô hình đã được kiểm tra trên nhiều bộ dữ liệu, bao gồm cả các bộ dữ liệu tổng hợp và các bộ dữ liệu thực tế từ ảnh vi sinh nhiều màu sắc. StarDist đã cho thấy hiệu suất tốt trong việc giải quyết các bài toán phân đoạn tế bào phức tạp.

Đặc biệt hiệu quả với ảnh có mật độ cao: Trong các tình huống tế bào đông đúc, StarDist giúp giảm thiểu lỗi phân đoạn, như việc hợp nhất sai giữa các tế bào lân cận hoặc làm mất thông tin về hình dạng.

Không yêu cầu hình dạng: Mô hình không yêu cầu quá trình rèn luyện hình dạng riêng biệt do đa giác lồi đã chứa đựng đủ thông tin về hình dạng của tế bào.

3.1.2. Nguyên lý của Stardist



Hình 3.2. Nguyên lý của mô hình StarDist

Phương pháp của Stardist tương tự như các phương pháp phát hiện đối tượng mà trực tiếp dự đoán hình dạng cho mỗi đối tượng quan tâm. Khác với hầu hết chúng, Stardist không sử dụng các hộp giới hạn có trục phô hợp như là biểu diễn hình dạng. Thay vào đó, mô hình của chúng tôi dự đoán một đa giác sao-lồi cho mỗi pixel. Cụ thể, đối với mỗi pixel có chỉ số i, j , mô hình hồi quy các khoảng cách $\{r_{i,j}^k\}_{k=1}^n$ tới ranh giới của đối tượng mà pixel đó thuộc về, theo một tập hợp n hướng tia phân chia góc đều. Rõ ràng, điều này chỉ được định nghĩa đúng cho các pixel (không phải nền) nằm trong một đối tượng. Do đó, mô hình của Stardist cũng dự đoán riêng cho mỗi pixel liệu nó có phải là một phần của một đối tượng không, chúng ta chỉ xem xét các đề xuất đa giác từ các pixel có xác suất đối tượng đủ cao $d_{i,j}$. Với các ứng viên đa giác như vậy và xác suất đối tượng kèm theo, thuật toán thực hiện bước loại bỏ độ tin cậy không cực đại (NMS) để đến được tập hợp đa giác cuối cùng, mỗi cái đại diện cho một trường hợp đối tượng cá thể.

Xác Suất Đối Tượng của Pixel:

Đối với mỗi pixel, StarDist định nghĩa xác suất thuộc về đối tượng thông qua khoảng cách Euclidean chuẩn hóa đến pixel nền gần nhất. Xác suất này đại diện cho độ chắc chắn của mô hình rằng pixel đó thuộc về một đối tượng.

Tính Tổng Quát và Độc Lập Không Gian:

Sử dụng mạng CNN, mô hình học được các đặc trưng không gian từ ảnh, giúp nó nhận biết các đối tượng ở bất kỳ vị trí nào trên ảnh. Điều này đảm bảo tính tổng quát và độc lập không gian của mô hình.

Khoảng Cách Tới Biên Đối Tượng:

Mô hình tính toán khoảng cách Euclidean từ mỗi pixel đến biên của đối tượng theo các hướng xuyên tâm. Quá trình này giúp StarDist xác định rõ biên của đối tượng, đặc biệt là ở những nơi có hình dạng phức tạp hoặc trong các tình huống tế bào đông đúc.

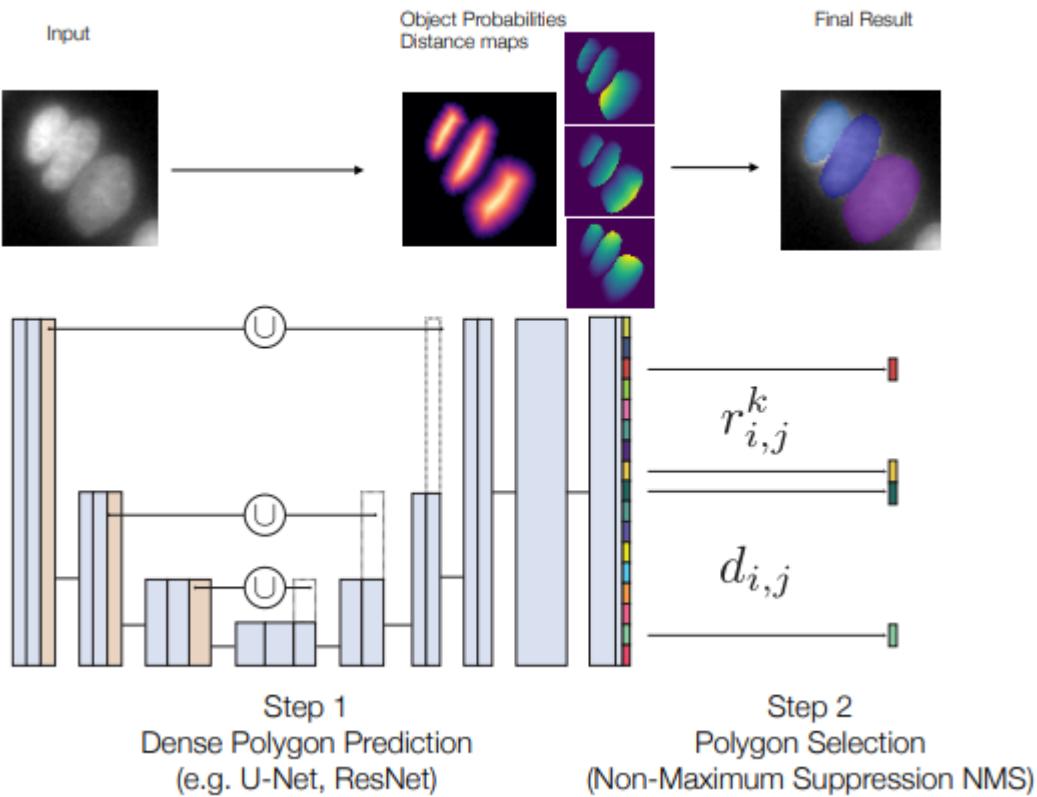
GPU Implementation:

Để đảm bảo tốc độ tính toán, mô hình sử dụng cài đặt GPU hiệu quả để tính toán các khoảng cách cần thiết trong quá trình huấn luyện. Điều này giúp tăng cường hiệu suất và làm cho quá trình huấn luyện nhanh chóng và hiệu quả.

Hiệu Suất Trong Điều Kiện Khó Khăn:

StarDist được đặc biệt thiết kế để giải quyết các thách thức trong việc phân đoạn tế bào trong ảnh vi sinh học, nơi mà các tế bào thường có hình dạng phức tạp và có thể đông đúc. Mô hình này thường cho hiệu suất tốt, đặc biệt là trong những tình huống phức tạp như vậy.

3.1.3. Quá trình xử lý



Hình 3.3. Quá trình xử lý của mô hình StarDist

Khi mô hình StarDist đọc một ảnh đầu vào, nó thực hiện một loạt các bước để dự đoán các đa giác lồi tương ứng với mỗi pixel trên ảnh.

Step 1: Dense Polygon Prediction (tính toán đa giác với độ chi tiết cao)

Mô hình StarDist sử dụng một kiến trúc mạng neural network để thực hiện tính toán đa giác với độ chi tiết cao. Cụ thể, nó sử dụng một mạng nơ-ron tích chập (CNN) để học các đặc trưng và dự đoán đa giác lồi tương ứng với mỗi pixel trên ảnh.

Kiến trúc của mạng nơ-ron trong StarDist có thể được tùy chỉnh tùy thuộc vào nhu cầu và cấu hình của người sử dụng, nhưng thường thì nó sẽ bao gồm các lớp tích chập để trích xuất đặc trưng, lớp kích hoạt để tạo ra đầu ra xác suất, và các lớp chuyển đổi để chuyển đổi xác suất thành đa giác sao-lồi.

Mục tiêu của mạng này là học mối quan hệ giữa các điểm trên đa giác và các đặc trưng của đối tượng trong không gian ảnh. Qua quá trình huấn luyện, mô hình học cách

tối ưu hóa mỗi quan hệ này để dự đoán các đa giác lồi một cách chính xác cho mỗi pixel trên ảnh đầu vào.

Tính toán đa giác với độ chi tiết cao là một phần quan trọng của mô hình StarDist và liên quan đến khả năng của mô hình dự đoán các đa giác sao-lồi tương ứng với mỗi pixel trên ảnh đầu vào một cách chính xác. Dưới đây là chi tiết về tính toán đa giác trong mô hình StarDist:

Dự đoán đa giác sao-lồi tại mỗi điểm pixel: Mô hình StarDist dự đoán một đa giác lồi tại mỗi pixel trên ảnh. Đa giác này là biểu diễn cho hình dạng của đối tượng tại vị trí đó. Đa giác có thể có nhiều cạnh và hình dạng phức tạp, tùy thuộc vào đặc trưng của đối tượng trong khu vực đó.

Sử dụng các hướng xuyên tâm để đo khoảng cách: Để dự đoán đa giác, mô hình sử dụng các hướng xuyên tâm để đo khoảng cách từ mỗi pixel đến biên của đối tượng. Các hướng xuyên tâm này giúp mô hình lấy thông tin về hình dạng của đối tượng từ nhiều phía khác nhau, tăng cường khả năng dự đoán chính xác của nó.

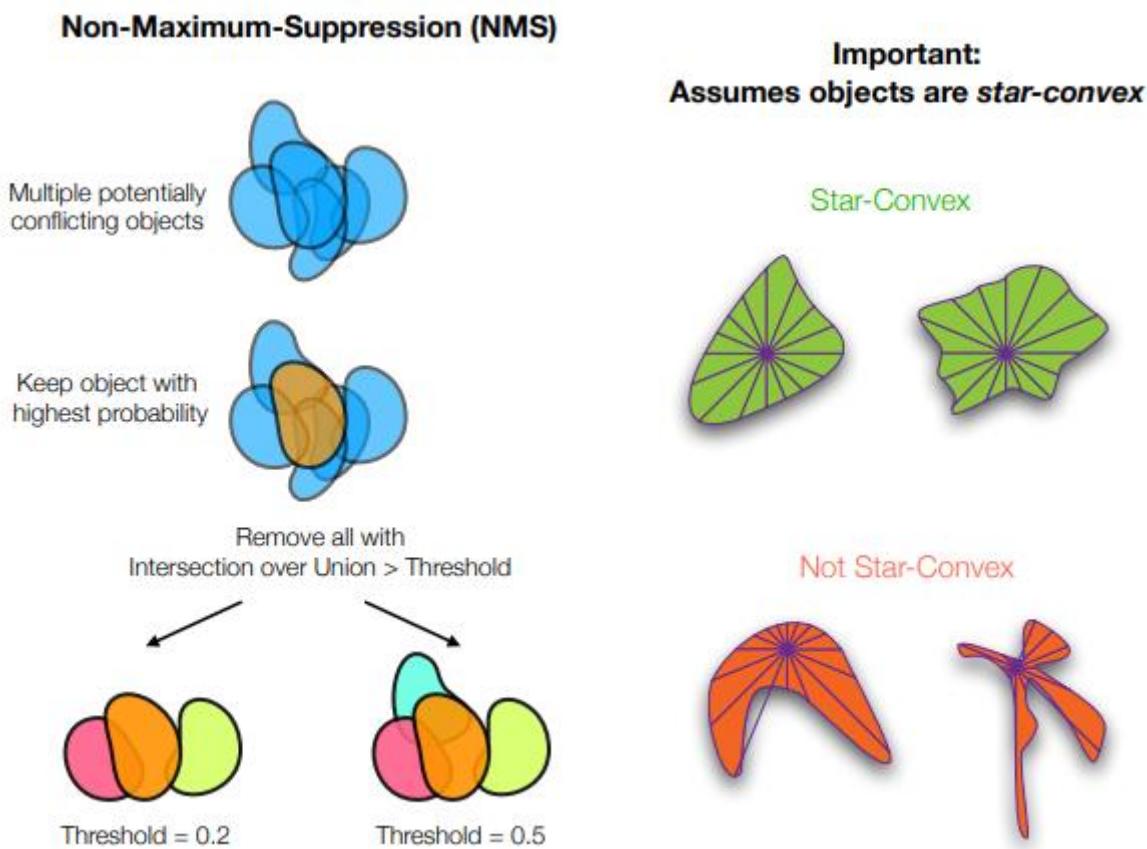
Tính chính xác và đồng nhất: Tính toán đa giác trong StarDist tập trung vào việc tạo ra đa giác lồi chính xác và đồng nhất với hình dạng thực tế của đối tượng. Điều này giúp mô hình phân đoạn tế bào một cách chính xác, đặc biệt là trong các kịch bản tế bào đồng đúc và có hình dạng phức tạp.

Học tự động từ quá trình huấn luyện: Trong quá trình huấn luyện, mô hình học tự động từ dữ liệu huấn luyện, điều này bao gồm cả việc học cách dự đoán các điểm chính xác trên đa giác và cách xử lý đa giác trong các điều kiện khác nhau.

Tích hợp với các đặc trưng của đối tượng: Mô hình tích hợp các đặc trưng của đối tượng từ các hướng xuyên tâm khác nhau để tạo ra một đa giác lồi phản ánh chính xác và đầy đủ hình dạng của đối tượng.

Hiệu quả trên các đối tượng đa dạng: Dense Polygon Prediction được thiết kế để hoạt động hiệu quả trên nhiều loại đối tượng và trong các điều kiện khác nhau, từ các tế bào đơn lẻ đến các khu vực tế bào đồng đúc và có hình dạng phức tạp.

Step 2: Polygon Selection(Non-Maximum Supresion NMS)



Hình 3.4. Non-Maximum Suppression

Hình 3.5. Đa giác sao-lồi

Trong mô hình StarDist, quá trình Polygon Selection(chọn đa giác) được thực hiện bằng cách sử dụng một kỹ thuật gọi là Non-Maximum Suppression (NMS). NMS được áp dụng để giảm thiểu sự chồng lấn và chọn ra các đa giác lồi đại diện cho các đối tượng trên ảnh một cách hiệu quả. Dưới đây là chi tiết về cách NMS được thực hiện trong StarDist:

Non-Maximum Suppression (NMS):

Xác định điểm cực đại cục bộ: Sau khi mô hình đã dự đoán các đa giác lồi tại mỗi pixel, mỗi đa giác được biểu diễn bởi các điểm chính xác cục bộ, thường là các điểm cực đại trên heatmap xác suất.

Sắp xếp các đa giác theo xác suất: Các đa giác được sắp xếp theo xác suất tương ứng với các điểm cực đại. Các đa giác có xác suất cao hơn sẽ được đặt ở phía trước trong danh sách.

Lặp qua các đa giác: Bắt đầu từ đa giác có xác suất cao nhất, lặp qua danh sách các đa giác. Nếu đa giác hiện tại có xác suất lớn hơn một ngưỡng được xác định,

nó được chọn làm một trong các đa giác cuối cùng. Các đa giác khác mà có IoU (Intersection over Union) lớn hơn một ngưỡng với đa giác đã chọn sẽ bị loại bỏ. *Lặp đến khi hết đa giác:* Quá trình lặp được tiếp tục cho đến khi tất cả các đa giác được xem xét.

IoU (Intersection over Union):

IoU, hay Intersection over Union, là một “độ đo” được sử dụng rộng rãi trong lĩnh vực nhận dạng đối tượng và phân đoạn ảnh. Nó đo lường mức độ trùng lặp giữa hai khu vực được dự đoán và thực tế trên ảnh. IoU được tính theo công thức sau:

$$IoU = \frac{\text{diện tích giao nhau}}{\text{diện tích hợp nhau}}$$

Diện tích giao nhau (Intersection): Là diện tích của khu vực mà cả hai khu vực dự đoán và thực tế đều chứa.

Diện tích hợp nhau (Union): Là diện tích của khu vực mà cả hai khu vực dự đoán và thực tế đều chứa, bao gồm cả diện tích giao nhau.

Kết quả của IoU nằm trong khoảng từ 0 đến 1, với giá trị càng gần 1 thì mức độ trùng lặp càng cao. Cụ thể:

IoU = 0: Không có sự trùng lặp giữa khu vực dự đoán và thực tế.

IoU = 1: Khu vực dự đoán và thực tế trùng khớp hoàn toàn.

IoU thường được sử dụng trong các nhiệm vụ đánh giá hiệu suất của mô hình machine learning, đặc biệt là trong bài toán phân đoạn ảnh, như việc đánh giá chất lượng của các đối tượng dự đoán so với đối tượng thực tế trên ảnh.

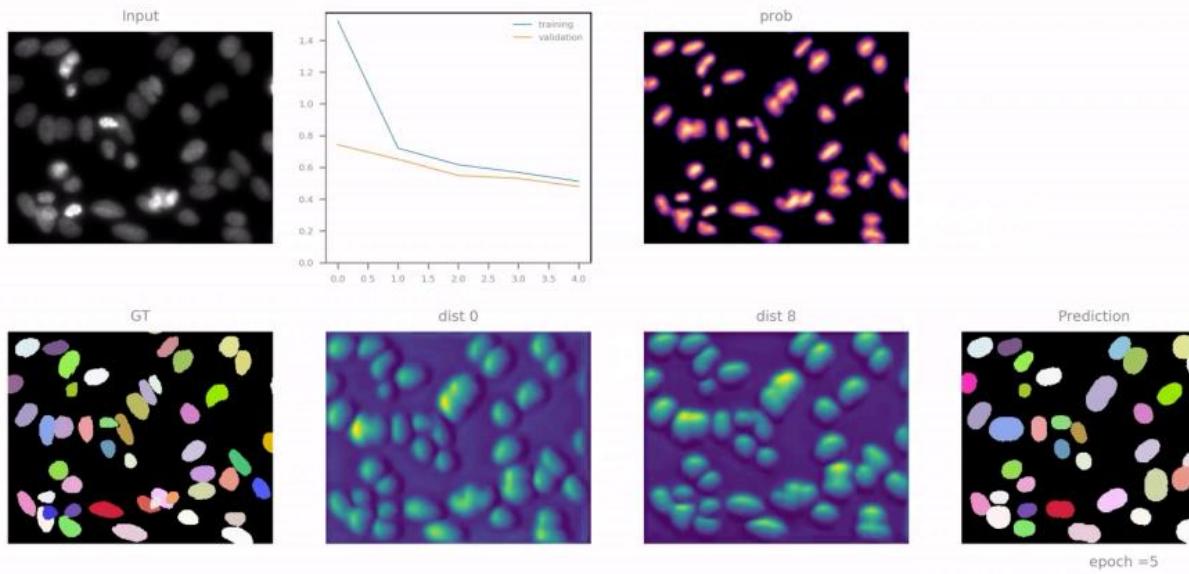
Kết Quả:

Kết quả cuối cùng của NMS là một tập hợp các đa giác được chọn một cách có ý nghĩa, không có sự chồng lấn lớn và đại diện cho các đối tượng trên ảnh.

Lưu ý quan trọng:

Mô hình StarDist chỉ áp dụng tốt cho những đối tượng hay tế bào có dạng đa giác sao-lồi, việc áp dụng cho những hình dạng khác có thể mang lại kết quả không như kỳ vọng.

3.2. Quá trình huấn luyện (Training process)



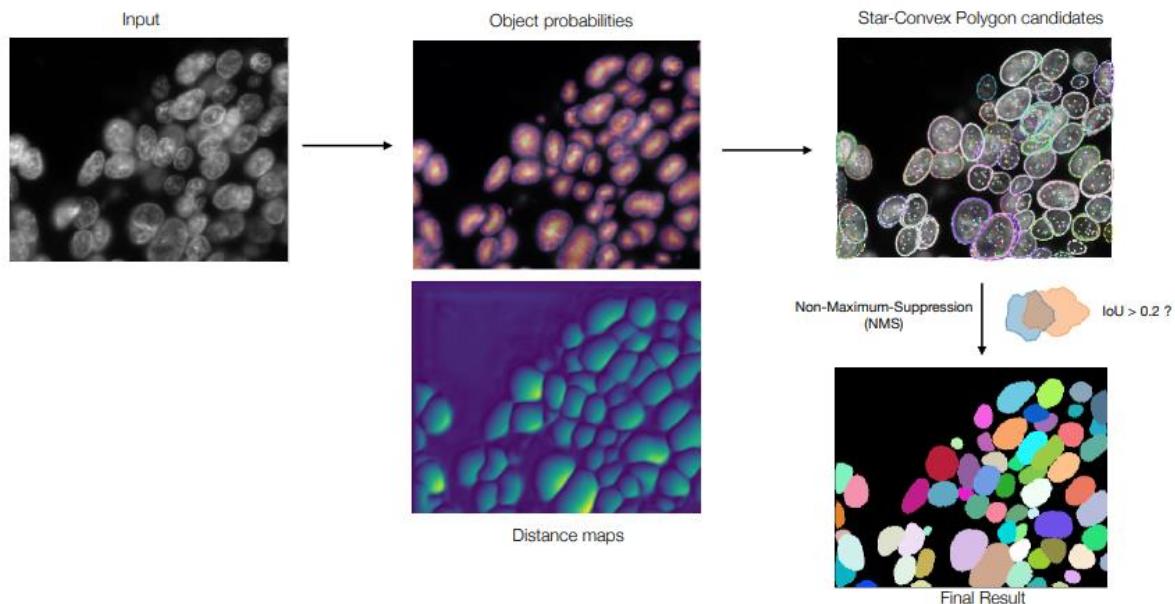
Hình 3.6. Quá trình huấn luyện của mô hình StarDist

Mặc dù phương pháp tổng quát của Stardist không ràng buộc vào một phương pháp hồi quy hoặc phân loại cụ thể nào, nhưng Stardist chọn mô hình mạng U-Net phổ biến làm cơ sở cho mô hình. Sau lớp đặc trưng U-Net cuối cùng, Stardist cần thêm một lớp convolutional(tích chập) 3×3 với 128 kênh (và hàm kích hoạt Relu) để tránh cho hai lớp đầu ra tiếp theo phải "cạnh tranh với nhau về đặc trưng". Cụ thể, Stardist sử dụng một lớp convolutional một kênh với hàm kích hoạt sigmoid cho đầu ra xác suất đối tượng. Lớp đầu ra khoảng cách đa giác có số kênh bằng số hướng tia xuyên tâm n và không sử dụng hàm kích hoạt bổ sung.

Huấn luyện. Stardist làm cực tiểu một hàm mất mát tiêu chuẩn cross-entropy nhị phân cho xác suất đối tượng được dự đoán. Đối với khoảng cách đa giác, Stardist sử dụng một hàm mất mát trung bình tuyệt đối được trọng số bởi xác suất đối tượng đúng của mỗi điểm ảnh, tức là sai số từng điểm ảnh được nhân với xác suất đối tượng trước khi lấy trung bình. Do đó, các điểm ảnh thuộc về nền sẽ không đóng góp vào hàm mất mát, vì xác suất đối tượng của chúng là không. Hơn nữa, các dự đoán cho các điểm ảnh gần trung tâm của mỗi đối tượng được đánh trọng số cao hơn, điều này là phù hợp vì chúng sẽ được ưu chuộng trong quá trình Non-Maximum Suppression.

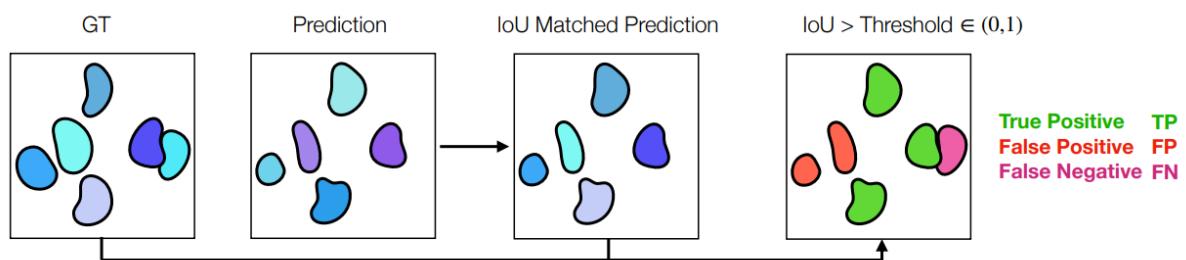
Non-maximum Suppression (NMS). Stardist thực hiện phương pháp non-maximum suppression (NMS) thông thường và tham lam để chỉ giữ lại những đa giác

có xác suất đối tượng cao nhất trong một khu vực nhất định. Stardist chỉ xem xét các đa giác liên quan đến các điểm ảnh có xác suất đối tượng cao hơn ngưỡng nào đó và tính toán sự giao nhau của chúng bằng một phương pháp cắt đa giác tiêu chuẩn.



Hình 3.7. Hai bước trong quá trình huấn luyện của StarDist

3.3. Đánh giá các chỉ số



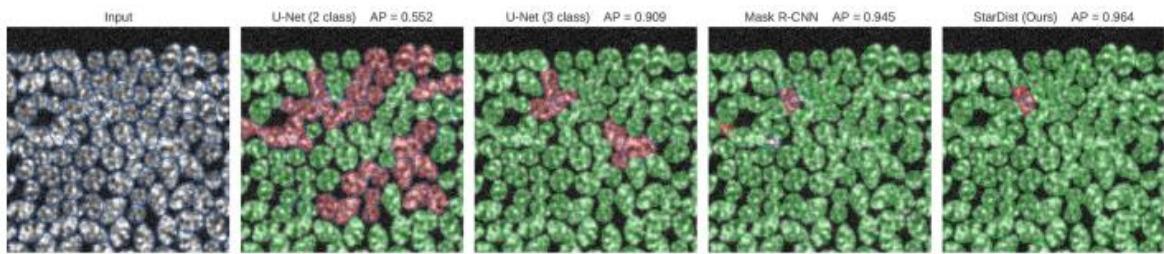
Hình 3.8. Minh họa các chỉ số đánh giá

Stardist áp dụng một phương pháp đánh giá tiêu biểu cho nhiệm vụ phát hiện đối tượng: Mỗi đối tượng được phát hiện I_{pred} được coi là một kết quả trùng khớp (*true positive* TP_τ) nếu tồn tại một đối tượng thực tế I_{gt} có diện tích giao nhau liên hiệp

$$IoU = \frac{I_{pred} \cap I_{gt}}{I_{pred} \cup I_{gt}}$$

lớn hơn một ngưỡng cố định $\tau \in [0,1]$.

Các đối tượng dự đoán không khớp được tính là *false positives* (FP_τ), còn các đối tượng thực tế không khớp được tính là *false negatives* (FN_τ). Stardist sử dụng độ chính xác trung bình *average precision* $AP_\tau = \frac{TP_\tau}{TP_\tau + FN_\tau + FP_\tau}$ để đánh giá trên toàn bộ các hình ảnh để làm điểm số cuối cùng.



Hình 3.9. Độ chính xác trung bình của các mô hình khác nhau với cùng ảnh đầu vào

3.4. So sánh các phương pháp

U-net (2 lớp): Các nhà nghiên cứu sử dụng kiến trúc U-net như là một cơ sở để dự đoán 2 lớp đầu ra (tế bào, nền). Sử dụng 3 khối lớn/ thấp mẫu (down/up-sampling), mỗi khối bao gồm 2 lớp tích chập với 32.2^k ($k = 0,1,2$) bộ lọc có kích thước 3 x 3 (Tổng cộng có 1,4 triệu tham số). Áp dụng ngưỡng σ lên bản đồ xác suất của tế bào và giữ lại các thành phần liên thông như kết quả cuối cùng (σ được tối ưu hóa trên tập kiểm tra cho mỗi tập dữ liệu).

U-net (3 lớp): Tương tự như U-net (2 lớp) nhưng các nhà nghiên cứu bổ sung việc dự đoán các pixel biên của tế bào như là một lớp bổ sung. Mục đích của việc này là phân biệt giữa các tế bào chật chội có biên liền nhau. Mỗi lần nữa, sử dụng các thành phần liên thông của lớp tế bào được ngưỡng để làm kết quả cuối cùng.

Mask R-CNN: Một phương pháp phân đoạn ví dụ hàng đầu, kết hợp một mạng để xuất vùng dựa trên bounding-box, thuật toán non-maximum-suppression (NMS), và phân đoạn mặt nạ cuối cùng (tổng cộng khoảng 45 triệu tham số). Các nhà nghiên cứu sử dụng một mã nguồn thật phổ biến. Đối với mỗi tập dữ liệu, thực hiện tìm kiếm trên lưới (grid-search) qua các siêu tham số thông thường, như ngưỡng NMS cho phần nhận diện, ngưỡng NMS cho mạng để xuất vùng và số lượng anchors.

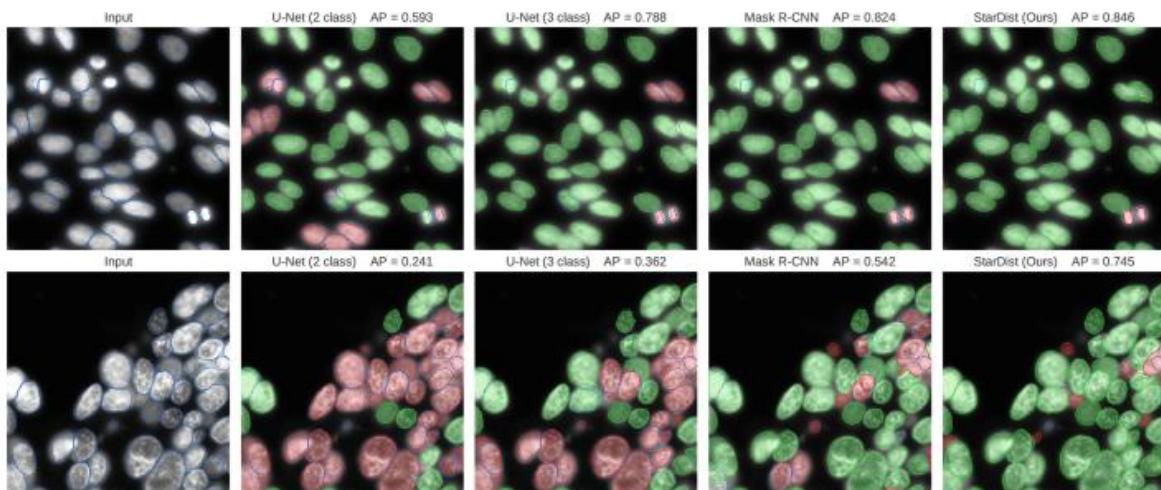
StarDist: Phương pháp mà đồ án đang đề xuất như mô tả ở các phần trước. Sử dụng n = 32 hướng tuyến tính và sử dụng một nền tảng U-net như 2 phương pháp cơ sở đầu tiên được mô tả ở trên.

low threshold: already slightly overlaps are counted as TP (high AP)
high threshold: only almost complete overlaps are counted as TP (low AP)



	Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$	
Average Precision (AP)	$\frac{TP}{TP + FP + FN}$	
Accuracy		$\frac{TP + TN}{TP + FP + FN}$

Hình 3.10. Ngưỡng IoU của từng mô hình với cùng một tập dữ liệu



Hình 3.11. So sánh các mô hình với dữ liệu bình thường và dữ liệu dày đặc

CHƯƠNG IV. ÁP DỤNG VÀ TRIỂN KHAI MÔ HÌNH STARDIST VÀO BÀI TOÁN CỦA ĐỒ ÁN

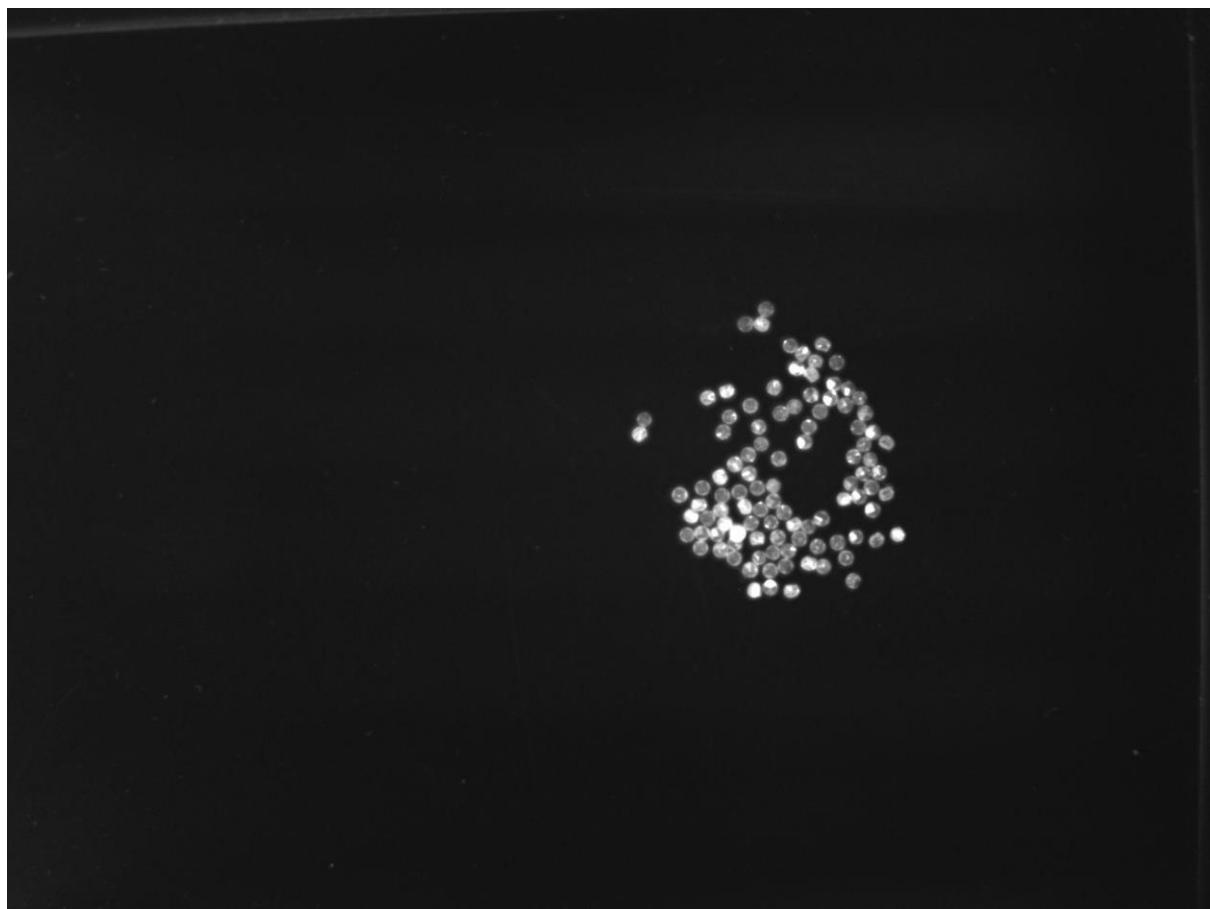
4.1. Chuẩn bị dữ liệu và huấn luyện mô hình

* Chuẩn bị và đánh nhãn dữ liệu:

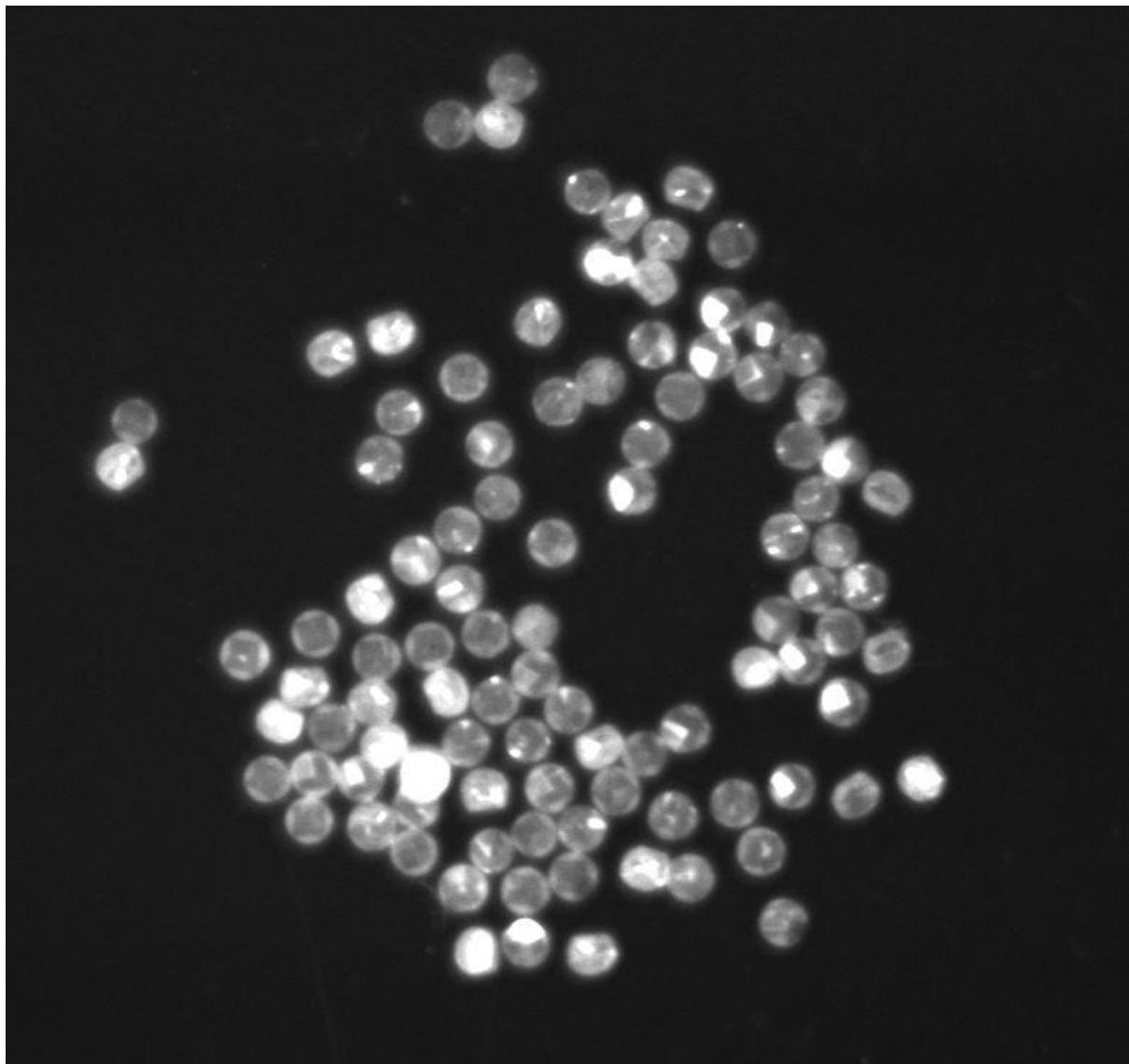
Với hình ảnh kim cương, bộ dữ liệu bao gồm các hình ảnh của kim cương với kích thước 1mm, được chia làm 2 phần: 33 ảnh cho dữ liệu train và 3 ảnh cho dữ liệu test. Tỷ lệ mỗi ảnh được chia khác nhau nhằm tăng tính chính xác cho dữ liệu học.

Với hình ảnh bọt khí, bộ dữ liệu bao gồm các hình ảnh bọt khí trong ống thủy tinh với các kích thước khác nhau, bộ dữ liệu được chia làm 2 phần: 49 ảnh cho dữ liệu train và 8 ảnh cho dữ liệu test. Tương tự như dữ liệu kim cương, ảnh bọt khí cũng được chia với tỷ lệ khác nhau nhằm tăng tính chính xác cho mô hình.

Ảnh kim cương:



Hình 4.1. Ảnh kim cương 28mm



Hình 4.2. Ảnh phóng to kim cương

Ảnh bọt khí (bóng bóng):



Hình 4.3. Ảnh bọt khí



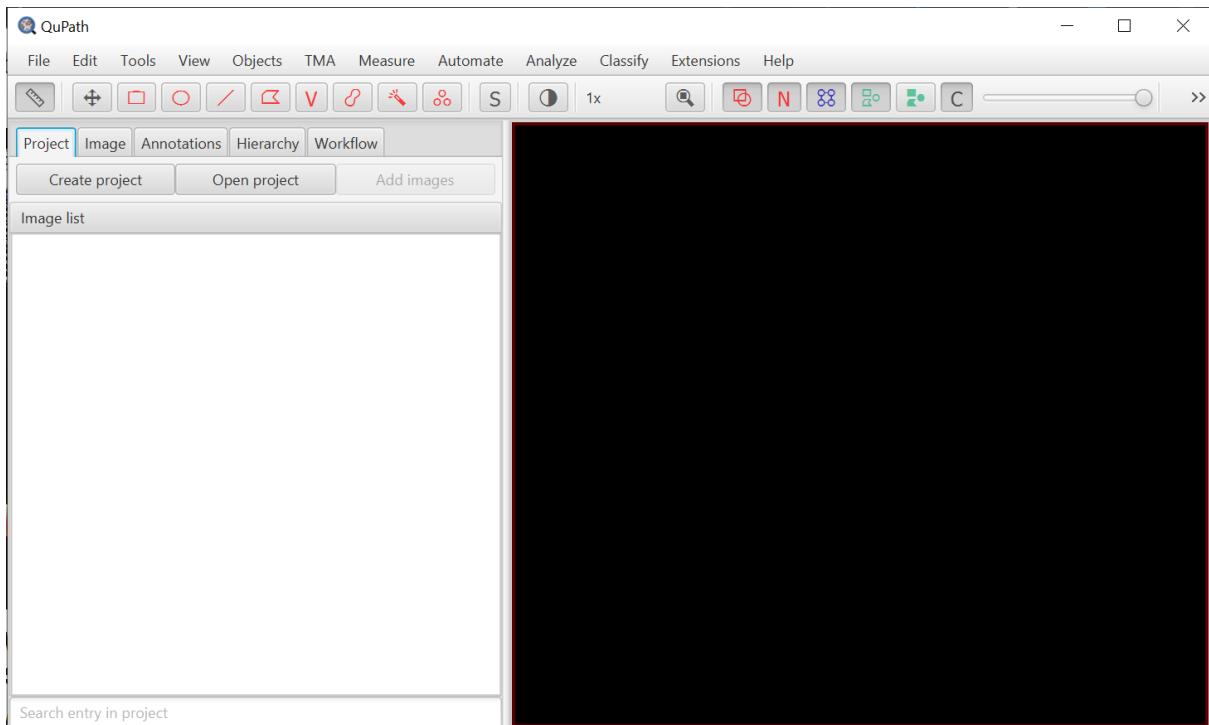
Hình 4.4. Ảnh phóng to bọt khí

Đánh nhãn dữ liệu:

Để đánh nhãn cho các dữ liệu hình ảnh kim cương, bọt khí đồ án sử dụng ứng dụng QuPath-0.4.4 để đảm nhiệm công việc này. QuPath là một phần mềm miễn phí và mã nguồn mở được thiết kế để hỗ trợ việc quản lý và phân tích hình ảnh y học, đặc biệt là trong lĩnh vực patô học (pathology). Được phát triển bởi Trung tâm Nghiên cứu Công

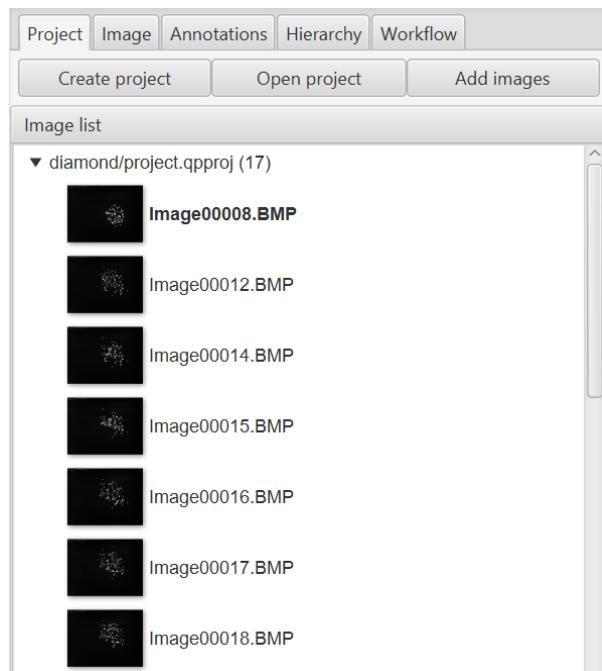
nghệ Ảnh và Thông tin Y học tại Đại học Queen's ở Belfast, QuPath cung cấp một loạt các tính năng mạnh mẽ để xử lý dữ liệu hình ảnh và đánh nhãn cho mục đích nghiên cứu y học và thực hành lâm sàng.

Kỹ thuật đánh nhãn:

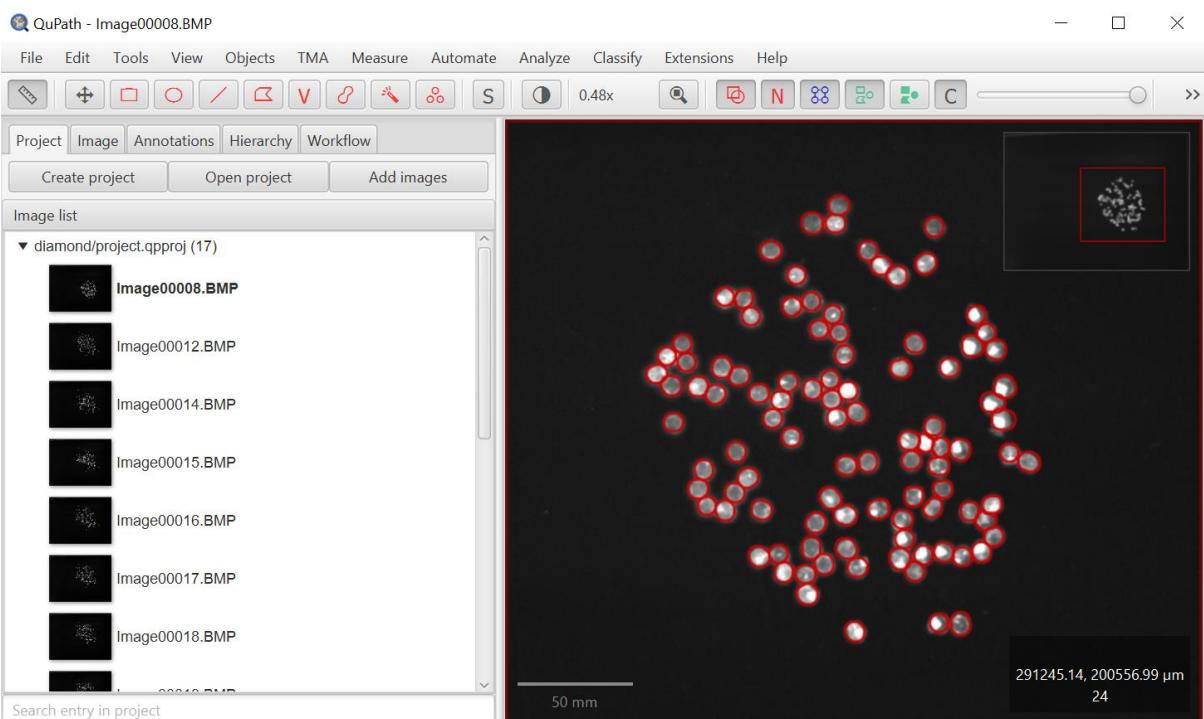


Hình 4.5. Giao diện trang chủ QuPath

Tại cửa sổ ứng dụng QuPath, ta chọn *Create Project*, sau khi hoàn tất các thông tin của project, ta import các dữ liệu ảnh cần đánh nhãn vào project vừa tạo.



Hình 4.6. Giao diện project trong QuPath



Hình 4.7. Giao diện sau khi đánh nhãn cho dữ liệu

Ta có thể chọn các công cụ trên thanh công cụ để đánh nhãn cho dữ liệu, hình ảnh sau khi được đánh nhãn dữ liệu được hiển thị ở cửa sổ *image*. Ta cũng có thể theo dõi những nhãn đã tạo tại cửa sổ *Annotations*.

Sau khi đánh nhãn cho dữ liệu hoàn thành, chúng ta có thể sử dụng cửa sổ *Automate* để chạy các *script* để lưu dữ liệu hình ảnh đã được đánh nhãn về máy tính. Dữ liệu sau khi được đánh nhãn sẽ được lưu trong thư mục *ground_truth*, bao gồm 2 thư mục nhỏ *images* và *masks*, chính là dữ liệu ảnh gốc và ảnh đã được đánh nhãn thành công.

Huấn luyện mô hình:

Vì mô hình StarDist có sử dụng GPU, vì vậy để thuận lợi, chúng ta sẽ sử dụng Google Colab để huấn luyện mô hình và xuất mô hình.

Các bước huấn luyện mô hình:

Import các thư viện cần thiết cho quá trình huấn luyện.

```
!pip install stardist
```

```
from __future__ import print_function, unicode_literals, absolute_import, division
import sys
import numpy as np
import matplotlib
from glob import glob
from tqdm import tqdm
from tifffile import imread
from csbdeep.utils import Path, normalize

from stardist import fill_label_holes, random_label_cmap, calculate_extents, gputools_available
from stardist.matching import matching, matching_dataset
from stardist.models import Config2D, StarDist2D, StarDistData2D
```

Thêm color map để hiển thị các dữ liệu nhãn.

```
#Random color map labels
np.random.seed(42)
lbl_cmap = random_label_cmap()
```

Vì các dữ liệu ảnh được lưu trữ tại Google Drive, nên cần kết nối Colab tới Drive để có thể sử dụng chúng.

```
from google.colab import drive
drive.mount('/content/drive')
```

Đọc các dữ liệu ảnh và dữ liệu nhãn đi kèm.

```
#Read input image and corresponding mask names
X = sorted(Path('/content/drive/MyDrive/datn/train/images/').glob('*.tif'))
Y = sorted(Path('/content/drive/MyDrive/datn/train/masks').glob('*.tif'))
```

Do các ảnh huấn luyện được sử dụng có đuôi `.tif` nên chúng ta cần sử dụng `imread` để đọc chúng.

#Read images and masks using their names.

#We are using tifffile library to read images as we have tif images.

```
X = list(map(imread,X))
Y = list(map(imread,Y))

n_channel = 1 if X[0].ndim == 2 else X[0].shape[-1] #If no third dim. then number of channels = 1.
Otherwise get the num channels from the last dim.
```

#Read images and masks using their names.

#We are using tifffile library to read images as we have tif images.

```
X_Test = list(map(imread,X_Test))
Y_Test = list(map(imread,Y_Test))
```

Sau khi đọc được các dữ liệu hình ảnh, chúng ta cần bình thường hóa chúng, sử dụng phương thức `normalize` của thư viện `csbdeep`.

```
#Normalize input images and fill holes in masks
axis_norm = (0,1) # normalize channels independently
# axis_norm = (0,1,2) # normalize channels jointly
if n_channel > 1:
    print("Normalizing image channels %s." % ('jointly' if axis_norm is None or 2 in axis_norm else 'independently'))
    sys.stdout.flush()
```

```
X = [normalize(x,1,99.8,axis=axis_norm) for x in tqdm(X)]
Y = [fill_label_holes(y) for y in tqdm(Y)]
X_Test = [normalize(x,1,99.8,axis=axis_norm) for x in tqdm(X_Test)]
Y_Test = [fill_label_holes(y) for y in tqdm(Y_Test)]
```

Sau khi bình thường hóa, để chuẩn bị cho quá trình huấn luyện, ta sẽ chia bộ dữ liệu train thành 2 phần: training và validation.

#Split to train and val

```
#You can use any method to split. I am following the method used in StarDist documentation example
assert len(X) > 1, "not enough training data"
rng = np.random.RandomState(42)
ind = rng.permutation(len(X))
n_val = max(1, int(round(0.15 * len(ind))))
ind_val, ind_train = ind[:n_val], ind[n_val:]
X_val, Y_val = [X[i] for i in ind_val], [Y[i] for i in ind_val]
X_trn, Y_trn = [X[i] for i in ind_train], [Y[i] for i in ind_train]
```

Chuẩn bị các cài đặt cho mô hình để huấn luyện. StarDist cung cấp cho chúng ta rất nhiều tùy chọn để có thể huấn luyện cho mô hình.

Ví dụ:

Về tham số:

axes: trục của ảnh đầu vào

n_rays: Số các hướng xuyên tâm của đa giác lồi ngôi sao, thường được khuyên chọn là n = 32.

n_channel_in: Số kênh của dữ liệu đầu vào (mặc định: 1)

backbone: Tên của kiến trúc mạng nơ-ron được dùng như backbone.

Về thuộc tính (có thể ghi đè)

unet_n_depth: Số mức độ phân giải của U-net

unet_kernel_size: Kích thước hạt nhân - kernel chập cho tất cả các lớp chập U-net

....

Cài đặt mô hình:

```
#Define the config by setting some parameter values
```

```
n_rays = 64 #Number of radial directions for the star-convex polygon.
```

```
# Use OpenCL-based computations for data generator during training (requires 'gputools')
```

```
use_gpu = False and gputools_available()
```

```
# Predict on subsampled grid for increased efficiency and larger field of view
```

```
grid = (2,2)
```

```
conf = Config2D (
```

```
    n_rays      = n_rays,
```

```
    grid       = grid,
```

```
    use_gpu     = use_gpu,
```

```
    n_channel_in = n_channel,
```

)

Ta sử dụng $n_rays = 64$ để có thể tăng độ chi tiết cho hình dạng của ảnh kim cương, với ảnh bọt khí ta sử dụng n_rays mặc định là 32

Lưu mô hình vào thư mục lưu trữ

```
#Save model to the specified directory
```

```
model = StarDist2D(conf, name='diamond', basedir='/content/drive/MyDrive/Colab Notebooks/models')
```

Định nghĩa một số hàm cần thiết cho quá trình huấn luyện:

```
#Define a few augmentation methods
```

```
def random_fliprot(img, mask):
```

```
    assert img.ndim >= mask.ndim
```

```
    axes = tuple(range(mask.ndim))
```

```
    perm = tuple(np.random.permutation(axes))
```

```
    img = img.transpose(perm + tuple(range(mask.ndim, img.ndim)))
```

```
    mask = mask.transpose(perm)
```

```
    for ax in axes:
```

```
        if np.random.rand() > 0.5:
```

```
            img = np.flip(img, axis=ax)
```

```
            mask = np.flip(mask, axis=ax)
```

```
    return img, mask
```

```
def random_intensity_change(img):
```

```
    img = img * np.random.uniform(0.6, 2) + np.random.uniform(-0.2, 0.2)
```

```
    return img
```

```
def augmenter(x, y):
```

```
    """Augmentation of a single input/label image pair.
```

```
    x is an input image
```

```
    y is the corresponding ground-truth label image
```

```
    """
```

```
    x, y = random_fliprot(x, y)
```

```
    x = random_intensity_change(x)
```

```
    # add some gaussian noise
```

```
    sig = 0.02 * np.random.uniform(0, 1)
```

```
    x = x + sig * np.random.normal(0, 1, x.shape)
```

```
    return x, y
```

Huấn luyện mô hình:

```
model.train(X_trn, Y_trn, validation_data=(X_val,Y_val), augmenter=augmenter, epochs=10, steps_per_epoch=100)
```

Sử dụng hàm train được cung cấp sẵn bởi StarDist, với số *epochs* là 10 và *steps_per_epoch* là 100.

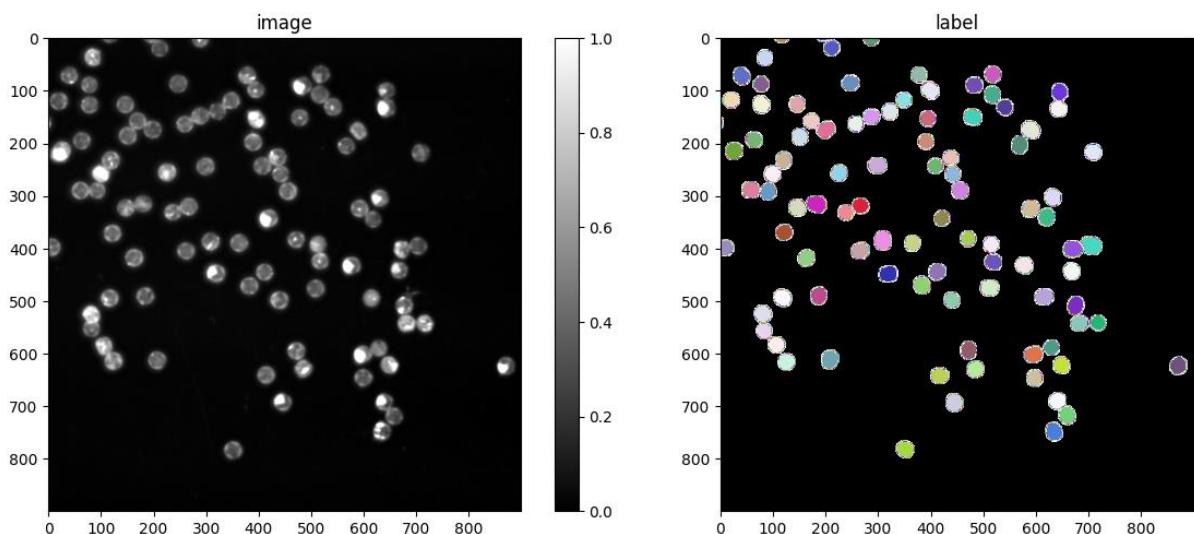
Sau khi mô hình huấn luyện thành công, ta cần tối ưu hóa ngưỡng sử dụng mô hình đã được huấn luyện

```
#Optimize the thresholds using the trained model
```

```
model.optimize_thresholds(X_val, Y_val)
```

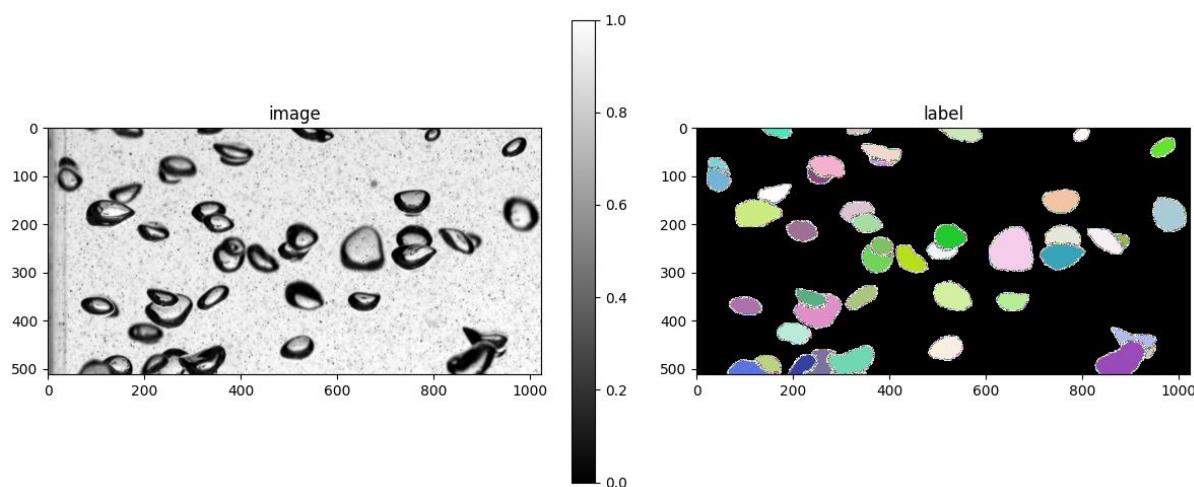
Biểu diễn hình ảnh trước khi huấn luyện:

Ảnh kim cương:



Hình 4.8. *Ảnh kim cương gốc và nhãn trước khi huấn luyện*

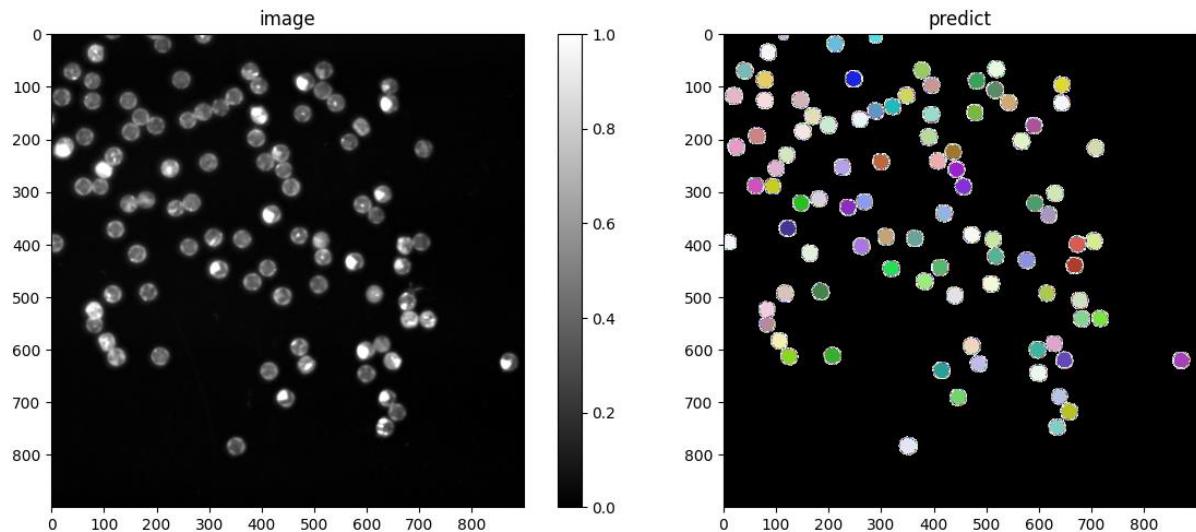
Ảnh bọt khí:



Hình 4.9. *Ảnh bọt khí gốc và nhãn trước khi huấn luyện*

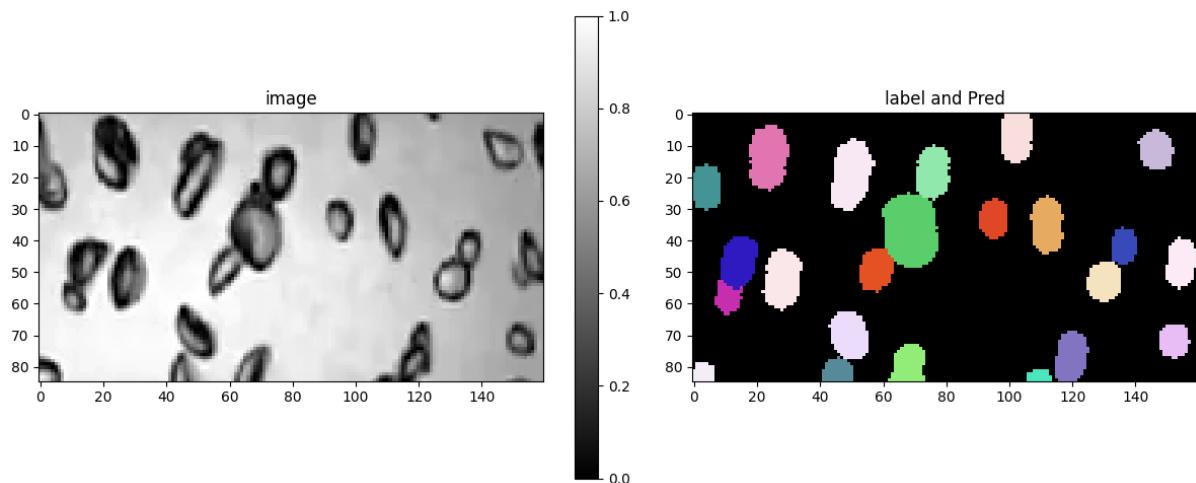
Sau khi mô hình được huấn luyện, ta thu được kết quả:

Ảnh kim cương:



Hình 4.10. Ảnh kim cương gốc và dự đoán sau khi huấn luyện

Ảnh bọt khí:



Hình 4.11. Ảnh bọt khí gốc và dự đoán sau khi huấn luyện

Đánh giá độ chính xác sau khi huấn luyện

Với mô hình kim cương:

```
DatasetMatching(criterion='iou', thresh=0.5, fp=186, tp=2667, fn=288, precision=0.9348054679284963,
recall=0.9025380710659898, accuracy=0.8490926456542502, f1=0.9183884297520661, n_true=2955,
n_pred=2853, mean_true_score=0.779269817012619, mean_matched_score=0.8634204384223056,
panoptic_quality=0.7929553406585018, by_image=False)
```

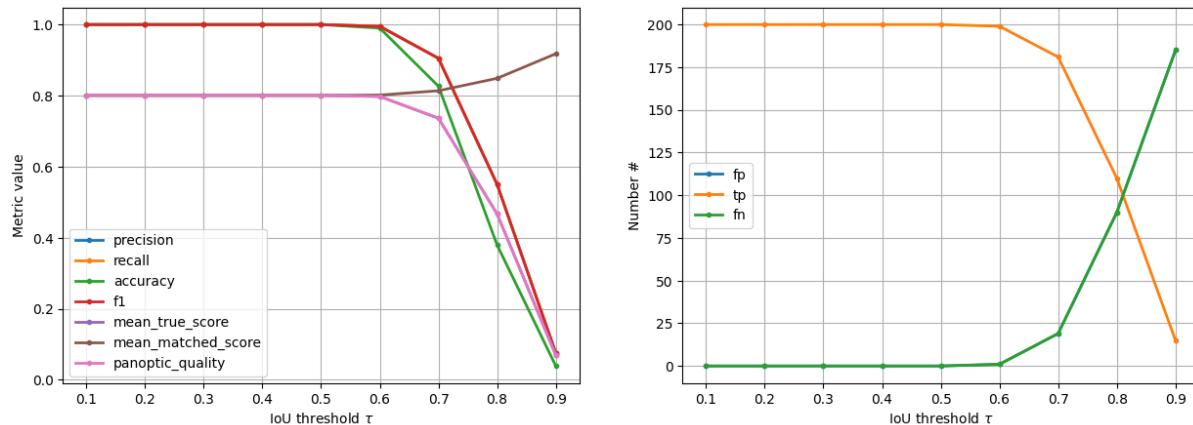
Ta có thể thấy với mô hình huấn luyện kim cương, ta thu được các chỉ số:

$precision = 0.935$

$recall = 0.902$

$accuracy = 0.849$

$f1_score = 0.918$



Hình 4.12. Biểu đồ các chỉ số của mô hình với dữ liệu kim cương

Với mô hình bọt khí:

```
DatasetMatching(criterion='iou', thresh=0.5, fp=45, tp=293, fn=106, precision=0.8668639053254438,
recall=0.7343358395989975, accuracy=0.6599099099099099, f1=0.7951153324287653, n_true=399,
n_pred=338, mean_true_score=0.5948370361985419, mean_matched_score=0.8100340527072295,
panoptic_quality=0.6440704950969287, by_image=False)
```

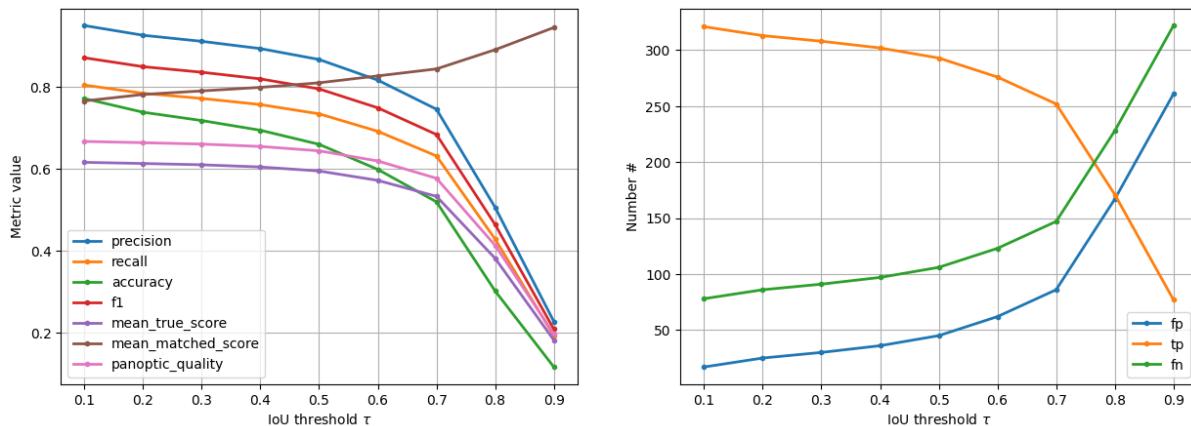
Ta có thể thấy với mô hình huấn luyện kim cương, ta thu được các chỉ số:

$precision = 0.866$

$recall = 0.734$

$accuracy = 0.66$

$f1_score = 0.795$



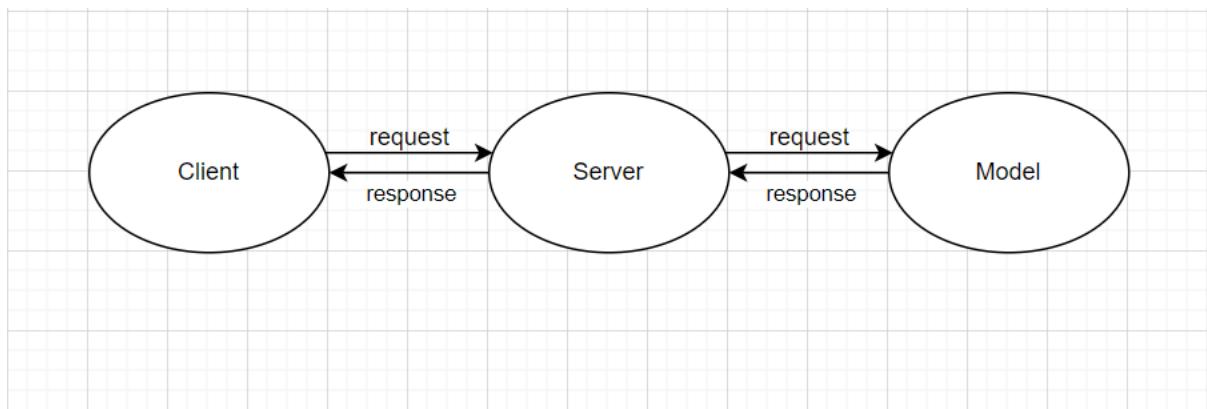
Hình 4.13. Biểu đồ các chỉ số của mô hình với dữ liệu bọt khí

Có thể thấy với dữ liệu ảnh kim cương, mô hình đang cho kết quả rất tốt, độ chính xác cao. Với dữ liệu khó như bọt khí, mô hình đang chỉ cho kết quả ở mức trung bình, chưa quá ấn tượng.

4.2. Triển khai mô hình lên ứng dụng web

Ứng dụng được triển khai lên môi trường website nhằm giúp người dùng có thể dễ dàng sử dụng mô hình, để có thể phân đoạn các hình ảnh kim cương cũng như bọt khí khi cần thiết. Ứng dụng cung cấp chức năng cho phép người dùng có thể tải ảnh lên từ máy tính, sau đó phân đoạn và đếm số lượng object có trong ảnh. Người dùng cũng có thể tải hình ảnh sau khi phân đoạn về máy tính để lưu trữ và sử dụng trong công việc của mình. Sau khi hình ảnh được phân đoạn, người dùng cũng có thể xem từng object mà mô hình đã dự đoán.

* Mô hình ứng dụng:



Hình 4.14. Mô hình của ứng dụng

Khi người dùng gửi yêu cầu, cũng như dữ liệu hình ảnh cần phân đoạn lên server, server tiếp nhận yêu cầu và sử dụng model đã được huấn luyện để phân đoạn hình ảnh, sau đó server nhận kết quả trả về từ model và trả kết quả cho phía client.

Ứng dụng được xây dựng với 2 phần:

Về giao diện (front-end), ứng dụng sử dụng Vue Js. Vue js là một thư viện mã nguồn mở JavaScript được thiết kế để xây dựng giao diện người dùng (UI) hiện đại và linh động trên trình duyệt web. Vue.js chú trọng vào việc đơn giản hóa quá trình phát triển ứng dụng và cung cấp một cách linh hoạt để quản lý và tái sử dụng các thành phần trong ứng dụng.

Về phần back-end, vì ứng dụng không yêu cầu nhiều tính năng phức tạp, Flask Python đã được áp dụng. Flask là một framework web Python nhẹ và dễ sử dụng, được thiết kế để xây dựng ứng dụng web nhanh chóng và đơn giản. Flask tập trung vào sự đơn giản, linh hoạt và có hiệu suất tốt, làm cho nó trở thành một trong những lựa chọn phổ biến cho việc phát triển các ứng dụng web và API.

* Luồng ứng dụng:

Sau khi người dùng tải hình ảnh lên giao diện ứng dụng, chọn các tùy chọn mô hình và submit. Yêu cầu được gửi tới server.

End point: `localhost:5002/segmentation`, Method: POST

Ở phía server:

Open image from request:

```

imageUpload = request.files['image']
modelType = request.form['type']
  
```

```

modelColorMap = request.form['colorMap']

Lấy dữ liệu hình ảnh

# Open image by pilow and normalize image

axis_norm = (0,1)

image = Image.open(imageUpload)

image = image.convert('L')

image = np.array(image)

image = normalize(image,1,99.8,axis=axis_norm)

```

Mở ảnh và bình thường hóa hình ảnh.

```

# Open trained model

modelName = 'stardist'

if modelType == "bubble":

    modelName = "bubble"

model = StarDist2D(None, name = modelName, basedir = 'models')

```

```

# Predict the image

y_test=model.predict_instances(image, n_tiles=model._guess_n_tiles(image), show_tile_progress=False)

```

Dựa vào dữ liệu người dùng gửi lên, server sẽ chọn loại mô hình phù hợp sau đó gọi mô hình để dự đoán dữ liệu

```

# Define a colormap

cmap = random_label_cmap()

if modelColorMap != "random_label_cmap":

    cmap = plt.get_cmap(modelColorMap)

# Normalize the grayscale image values to the range [0, 1]

normalized_image = (y_test[0] - y_test[0].min()) / (y_test[0].max() - y_test[0].min())

# Apply the colormap to the normalized image

image = (cmap(normalized_image) * 255).astype(np.uint8)

image = Image.fromarray(image)

```

Sau khi dự đoán hình ảnh hoàn thành, server sẽ định nghĩa color_map trả về dựa trên dữ liệu người dùng gửi lên, đồng thời server sẽ đưa hình ảnh về dạng grey scale và áp dụng color_map vừa định nghĩa.

```
# Save the image to a BytesIO object
```

```

image_bytes = BytesIO()
image.save(image_bytes, format='PNG')
image_bytes.seek(0)

# Return the image as a response with the appropriate content type
# Create a dictionary to hold the image and text data
response_data = {
    'imageBytes': image_base64,
    'objectsCount': len(y_test[1]['points']),
    'points': y_test[1]['points'].tolist(),
    'coord': y_test[1]['coord'].tolist(),
}

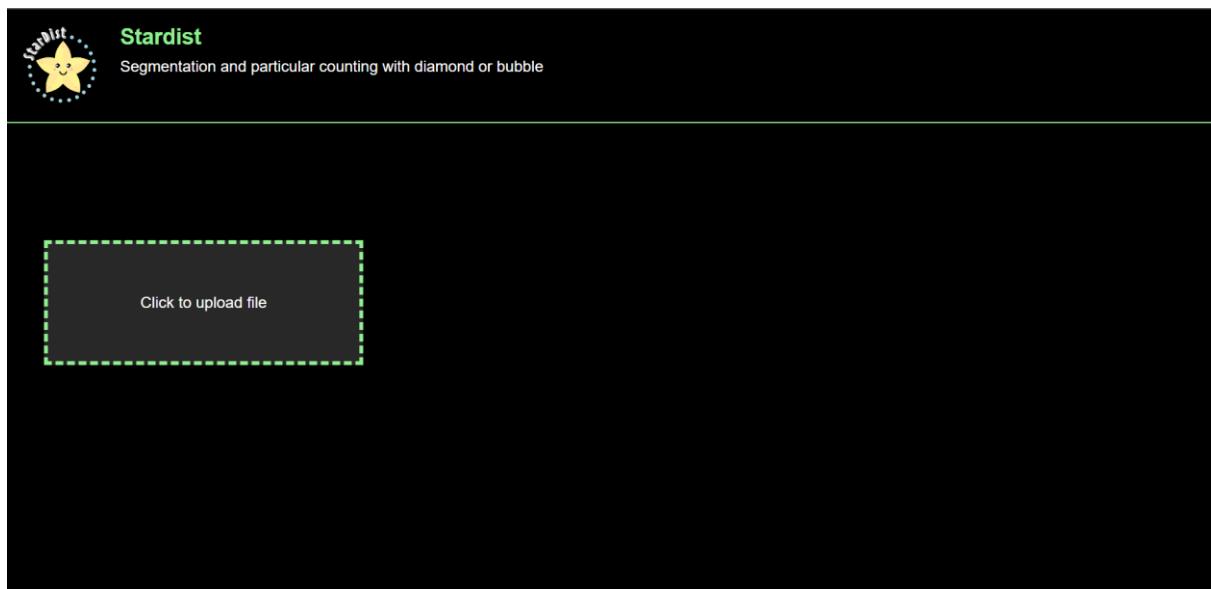
# Return the response as JSON with image and text data
return jsonify(response_data)

```

Sau quá trình dự đoán, server sẽ chuyển ảnh dự đoán sang dạng byte, trả về cho phía client dữ liệu dưới dạng JSON.

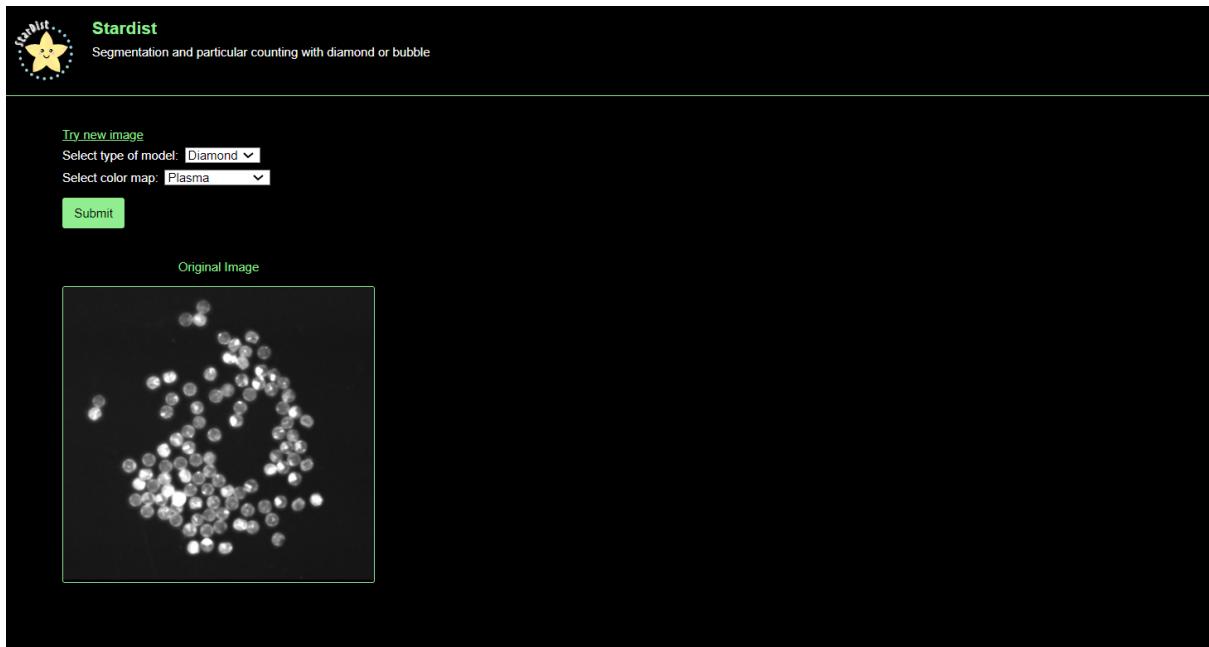
* Giao diện của ứng dụng:

Trang chủ:

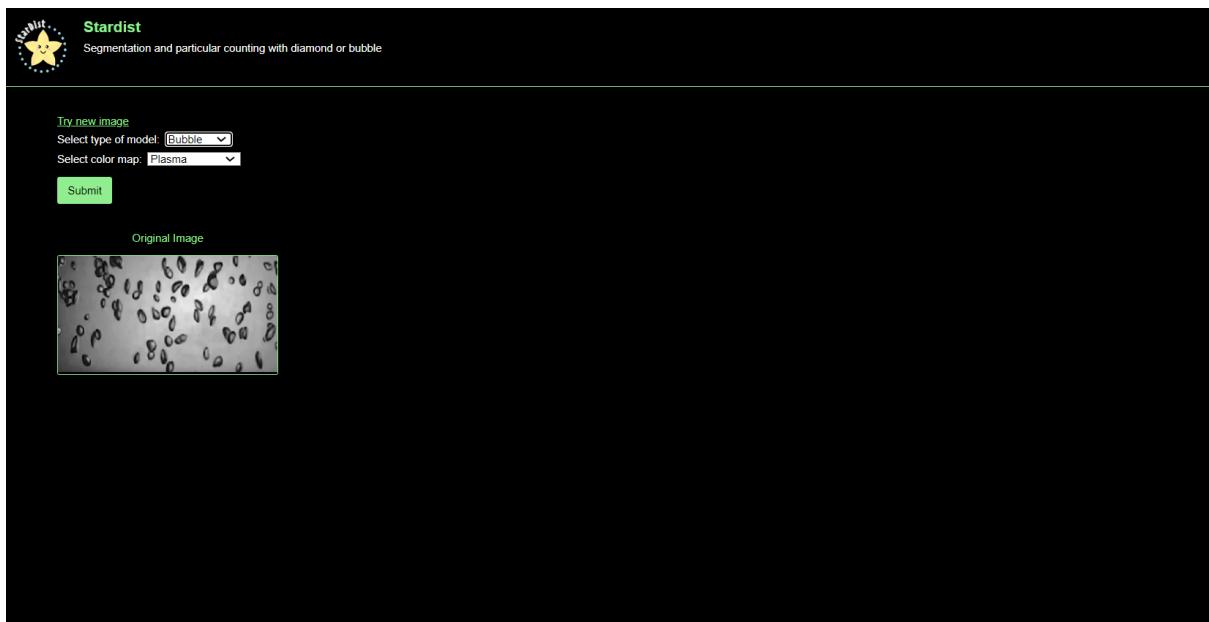


Hình 4.15. Giao diện trang chủ của ứng dụng

Giao diện khi tải dữ liệu ảnh cần phân đoạn:



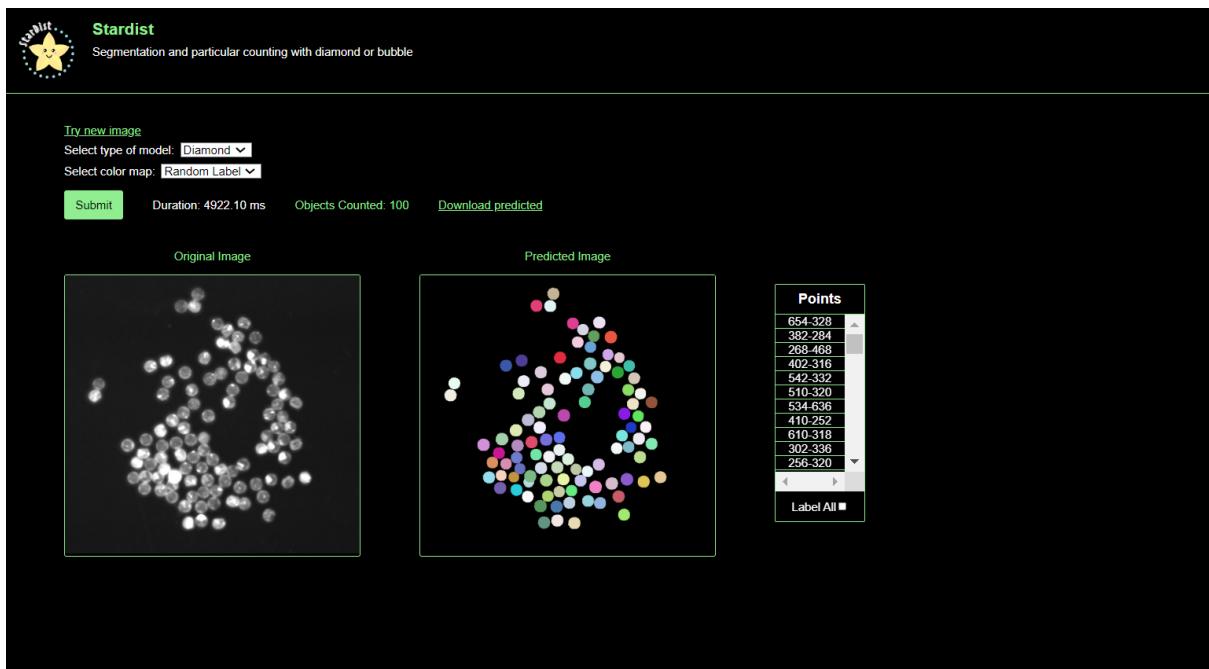
Hình 4.16. Giao diện sau khi tải dữ liệu ảnh kim cương



Hình 4.17. Giao diện sau khi tải dữ liệu ảnh bọt khí

Ở giao diện này, người dùng có thể chọn loại mô hình phù hợp, cũng như có thể chọn các color_map hiển thị cho ảnh phân đoạn.

Giao diện sau khi phân đoạn hoàn thành:



Hình 4.18. Giao diện sau khi ứng dụng hoàn thành phân đoạn

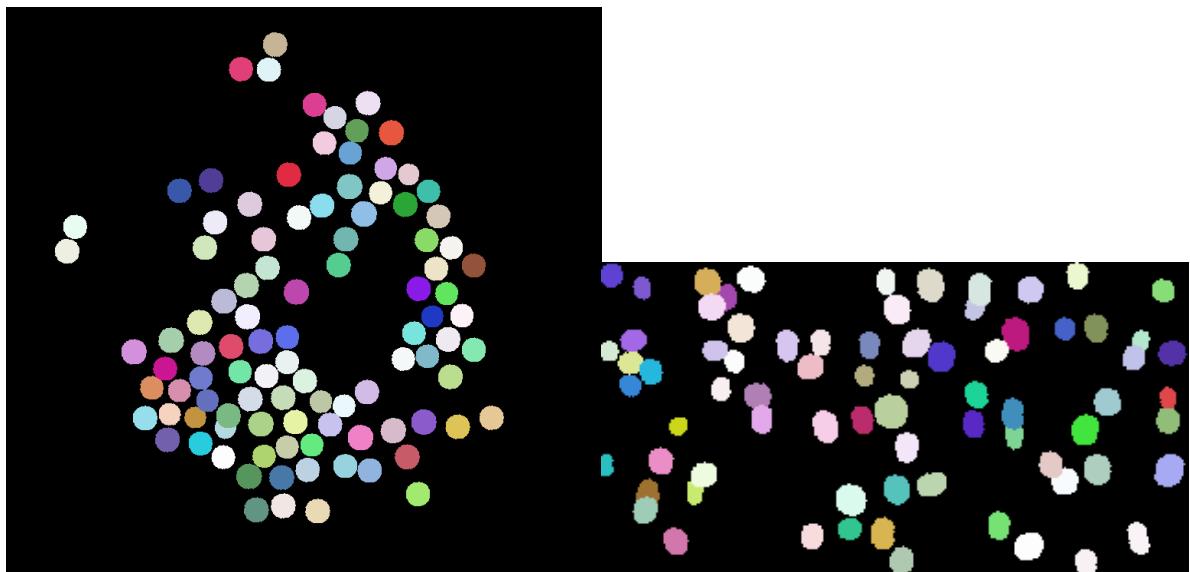


Hình 4.19. Giao diện sau khi ứng dụng hoàn thành phân đoạn

Ở giao diện sau khi phân đoạn, ứng dụng hiển thị ảnh dự đoán của mô hình, các thông tin liên quan đến các object được mô hình dự đoán. Người dùng cũng có thể tải xuống hình ảnh dự đoán của mô hình.

Ảnh dự đoán của mô hình:

Sau khi áp dụng mô hình, ảnh sẽ được tải về máy tính của người dùng với định dạng PNG.



Hình 4.20. Ảnh sau khi dự đoán từ ứng dụng

CHƯƠNG V. KẾT LUẬN

Trong quá trình nghiên cứu và triển khai mô hình StarDist với sự chuyên sâu vào các hình ảnh chứa hạt nhân được biểu diễn dưới dạng kim cương và bọt khí, đồ án đã đạt được những kết quả quan trọng và có ý nghĩa. Đồ án đã đạt được mục tiêu đặt ra là phân đoạn, nhận dạng được hình ảnh kim cương cũng như bọt khí. Đồ án đã tổng hợp được các cách tiếp cận thông thường, từ phương pháp Bottom-Up đến Top-Down và đặc biệt chú ý đến những vấn đề phô biến khi phân đoạn hình ảnh với lượng đối tượng dày đặc và chồng chéo nhau. Các kiến trúc mạng CNN như U-net, Mask R-CNN đã được đề như nền tảng của phương pháp sau này. Đồ án giới thiệu phương pháp Stardist, một phương pháp mới tiếp cận vấn đề Nuclei Segmentation. Nguyên tắc của Stardist đã được trình bày cùng với ví dụ minh họa và quá trình huấn luyện. Đồ án đã so sánh phương pháp này với các phương pháp khác và thực hiện tính toán về độ chính xác để đánh giá hiệu suất. Quá trình áp dụng mô hình Stardist vào bài toán thực tế được mô tả chi tiết, bao gồm việc thu thập và tiền xử lý dữ liệu, huấn luyện mô hình và đánh giá độ chính xác. Triển khai mô hình lên web server để sử dụng một cách thuận tiện và hiệu quả cũng được thảo luận.

Trong tương lai sắp tới, phân đoạn hình ảnh sẽ vẫn tiếp tục là một lĩnh vực quan trọng trong computer vision với nhiều ứng dụng trong thực tế, đặc biệt là trong những lĩnh vực về công nghiệp. Với tiềm năng, cũng như những cơ hội to lớn để phát triển, phân đoạn hình ảnh hứa hẹn sẽ là một lĩnh vực cực kỳ thu hút đối với thế giới của chúng ta.

TÀI LIỆU THAM KHẢO

- [1] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers.
“*Cell Detection with Star-convex Polygons.*”
- [2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox.
“*U-Net: Convolutional Networks for Biomedical Image Segmentation*”
- [3] H. Hessenkemper, S. Starke, Y. Atassi, T. Ziegenhein, D. Lucas.
“*Bubble identification from images with machine learning methods*”
- [4] Kaiming He Georgia Gkioxari Piotr Dollar Ross Girshick.
“*Mask R-CNN*”
- [5] Frédéric Cao, José Luis Lisani, Jean-Michel Morel, Pablo Musé, Frédéric Sur
“*A Theory of Shape Identification*”
- [6] VinBigData, “Phân đoạn hình ảnh: Các kỹ thuật truyền thống và phương pháp học sâu”.<https://vinbigdata.com/kham-pha/phan-doan-hinh-anhcac-ky-thuat-truyen-thong-phuong-phap-hoc-sau.html>