TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

——————— * ———————

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

PHÁT HIỆN BÀN TAY TRONG VIDEO DỰA TRÊN KỸ THUẬT HỌC SÂU VÀ THEO VẾT

Sinh viên thực hiện : **Nguyễn Đình Hà**
Lớp: KSCLC HTTT&TT – K58
Giáo viên hướng dẫn: **PGS. TS. Trần Thị Thanh Hải**

HÀ NỘI 6-2018

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Đình Hà

Điện thoại liên lạc 0969538900 Email: nguyenha.pfiev@gmail.com

Lớp: KSCLC Hệ thống thông tin và truyền thông K58 Hệ đào tạo: KSCLC-TN-TT

Đồ án tốt nghiệp được thực hiện tại:

Viện nghiên cứu quốc tế MICA – Trường Đại học Bách khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 19/1/2018 đến 28/05/2018

2. Mục đích nội dung của ĐATN

Nghiên cứu kỹ thuật phát hiện và phân vùng đối tượng bàn tay trên ảnh

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu bài toán phát hiện và phân vùng đối tượng bàn tay trong video và hướng giải quyết
- Tìm hiểu và thử nghiệm kỹ thuật mạng neuron tích chập nhằm phát hiện và phân đoạn vùng bàn tay (Mask R-CNN).
- nghiên cứu và thử nghiệm kỹ thuật theo vết Mean Shift để nâng cao độ chính xác của giải thuật Mask R-CNN
- Đánh giá độ chính xác, ưu nhược điểm của kỹ thuật nghiên cứu.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Đình Hà* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS. Trần Thị Thanh Hải*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Nguyễn Đình Hà

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

Hà Nội, ngày tháng năm

Giáo viên hướng dẫn

PGS.TS. Trần Thị Thanh Hải

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Bài toán phát hiện và phân đoạn vùng bàn tay trên ảnh có rất nhiều ứng dụng trong các hệ thống thông minh hiện nay như hệ thống nhận dạng cử chỉ tay nhằm điều khiển các thiết bị điện gia dụng trong nhà thông minh. Vấn đề chính đặt ra trong bài toán này là phát hiện có hay không đối tượng bàn tay (hand detection), nếu có xác định vùng bàn tay đó trên ảnh (hand segmentation). Trong thời gian gần đây, có một số mô hình giải quyết khá tốt cho bài toán phát hiện và phân vùng đối tượng nói chung dựa trên các kỹ thuật học sâu tiên tiến như Fast R-CNN, Faster R-CNN, Mask R-CNN. Trong khuôn khổ của ĐATN, em tìm hiểu kiến trúc mạng Mask R-CNN. Đây là một kiến trúc mạng neuron sâu cho kết quả phân đoạn tốt các lớp đối tượng trên ảnh như người, xe, con vật. Tuy nhiên chưa có một công việc nào sử dụng mạng Mask R-CNN cho bài toán phát hiện và phân vùng bàn tay trong ảnh và video. ĐATN của em sẽ nghiên cứu và đánh giá tính khả thi của kiến trúc Mask R-CNN để giải quyết bài toán phát hiện và phân vùng bàn tay như một pha tiền xử lý trong một hệ thống nhận dạng cử chỉ hoàn chính. Các nghiên cứu thực nghiệm cho thấy Mask R-CNN làm việc tốt khi bàn tay xuất hiện trực diện trước khung hình của camera, không bị che khuất hoặc trùng lấp trên vùng ảnh có tính chất màu da (mặt người). Để khắc phục các thách thức này, trong ĐATN của mình, em đã nghiên cứu và áp dụng giải thuật Meanshift để theo vết bàn tay nhằm loại bỏ những phát hiện thừa hoặc bổ sung những phát hiện thiếu. Việc kết hợp Mask R-CNN với giải thuật tracking cho hiệu quả phát hiện cao hơn Mask R-CNN nguyên bản. Các đánh giá thực nghiệm đã được triển khai trên một CSDL đa thể thức đa góc nhìn các cử chỉ động của bàn tay, được thu thập tại Viện MICA.

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến cô Trần Thị Thanh Hải vì đã hướng dẫn ân cần, tận tình, và kiên nhẫn với em và tạo động lực cho em trong khoảng thời gian thực hiện đồ án tốt nghiệp này.

Em cũng xin chân thành cảm ơn Viện Nghiên cứu Quốc tế MICA đã tạo cho em một môi trường thuận lợi để học tập và nghiên cứu.

Em cũng xin gửi lời cảm ơn đến tất cả các thầy cô Viện CNTT nói riêng và các thầy cô của Trường Đại học Bách khoa Hà Nội nói chung đã truyền đạt cho em những kiến thức cần thiết trong suốt thời gian học trên giảng đường.

Em cũng xin gửi lời cảm ơn đến tất cả các bạn cùng lớp đã đồng hành cùng em trong suốt thời gian học tập và làm việc, đã giúp đỡ động viên em rất nhiều.

Em xin gửi lời cảm ơn đến gia đình đã luôn quan tâm, ủng hộ hết lòng về vật chất và tinh thần trong suốt thời gian qua.

Do thời gian và kiến thức có hạn nên không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được sự đóng góp quý báu của thầy cô và các bạn.

Cuối cùng, em xin gửi lời chúc sức khỏe, hạnh phúc tới thầy cô, gia đình và bạn bè.

Hà Nội, ngày tháng năm 2018

Sinh viên

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3
LỜI CẢM ƠN 4
MỤC LỤC 5
DANH MỤC HÌNH ẢNH 7
DANH MỤC BẢNG 8
DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ 9
MỞ ĐẦU 10
CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN VÀ PHÂN VÙNG ĐỐI TƯỢNG BÀN TAY VÀ VẤN ĐỀ ĐẶT RA VỚI ĐỒ ÁN 12
1.1. Ngữ cảnh của bài toán 12
1.2. Các thành phần chính của hệ thống phát hiện và phân vùng đối tượng 14
1.3. Một số nghiên cứu liên quan về phát hiện và phân đoạn bàn tay người trong ảnh 15
1.3.1. Hướng tiếp cận phát hiện đối tượng bàn tay trên ảnh dựa vào các đặc trưng được thiết kế bằng tay 16
1.3.2. Hướng tiếp cận biểu diễn phát hiện và phân vùng dựa trên kỹ thuật học sâu 16
1.3.3. Nhận xét chung 18
1.4. Mục tiêu của đồ án 18
CHƯƠNG 2: KỸ THUẬT HỌC SÂU MASK R-CNN VÀ THUẬT TOÁN THEO VẾT MEAN SHIFT 20
2.1. Giới thiệu chung về mạng nơ ron tích chập 20
2.1.1. Mạng nơ ron 20
2.1.2. Mạng nơ ron tích chập 21
2.1.3. Kiến trúc của CNN cho bài toán nhận dạng và phân vùng đối tượng 23
2.2. Mạng Mask R-CNN 26
2.3. Thuật toán Mean shift theo vết đối tượng 31
CHƯƠNG 3: Triển khai mô đun nhận dạng và đánh giá thử nghiệm 33
3.1. Mô hình đề xuất nghiên cứu 33
3.1.1. Quá trình huấn luyện Mask R-CNN 34
3.1.2. Cài đặt module và huấn luyện Mask R-CNN 37
3.2. Kết quả thí nghiệm 37

3.2.1. Kết quả huấn luyện với ảnh resize kích cỡ 256x256 (1) 38
3.2.2. Kết quả huấn luyện với ảnh crop kích cỡ 256x256 39
3.2.3. Kết quả huấn luyện trên ảnh gốc trên từng góc nhìn 40
3.2.4. Kết quả có áp dụng thuật toán tracking 42
CHƯƠNG 4: KẾT LUẬN 44
4.1. Kết quả đạt được 44
4.2. Những điểm còn hạn chế 44
4.3. Hướng phát triển 44
TÀI LIỆU THAM KHẢO 45

DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ bố trí các Kinect thu thập dữ liệu cử chỉ người điều khiển ở các góc nhìn khác nhau *13*
Hình 2: Minh họa đầu vào và đầu ra của bài toán phát hiện và phân vùng đối tượng bàn tay *13*
Hình 3 Sơ đồ tổng thể hệ thống áp dụng học máy (https:machinlearningcoban.com) *15*
Hình 4 Kiến trúc cơ bản của một mạng CNN [9] *17*
Hình 5: Sơ đồ khối chung của hệ thống *19*
Hình 6: Minh họa mạng nơ ron của người (từ Rob Fergus) *20*
Hình 7: Mạng nơ ron nhiều tầng *21*
Hình 8: Kết nối giữa các tầng trong mạng nơ ron truyền thống *22*
Hình 9: Kết nối giữa các tầng trong mạng nơ ron tích chập *22*
Hình 10: Các bước cơ bản trong mạng neuron tích chập *23*
Hình 11: Cấu trúc mạng R-CNN [13] *23*
Hình 12: Kiến trúc Fast R-CNN *25*
Hình 13: Hoạt động của Spatial pyramid pooling *25*
Hình 14: Kiến trúc Faster R-CNN *26*
Hình 15: Kiến trúc Mask R-CNN *27*
Hình 16: Giải thích kiến trúc Mask R-CNN (Medium.com) *27*
Hình 17: Cách hoạt động của mạng tích chập đầy đủ *28*
Hình 18: Mô tả cách hoạt động của khối deconvolution và unpooling *28*
Hình 19: Max Pooling *29*
Hình 20: Mô tả phương pháp RoIAlign *30*
Hình 21: Kiến trúc backbone Mask R-CNN *30*
Hình 22: Mạng kim tự tháp *31*
Hình 23: Ví dụ thuật toán Mean shift (opencv docs) *32*
Hình 24: Sơ đồ thuật toán mô hình đề xuất *33*
Hình 25: Chuẩn bị dữ liệu *35*
Hình 26: Giá trị hàm mất mát trong quá trình huấn luyện *36*
Hình 27: Một số trường hợp nhận dạng sai của mạng *39*
Hình 28: So sánh kết quả phân vùng của mô hình ảnh resize và ảnh gốc *42*
Hình 29: Thuật toán Mean Shift áp dụng tăng độ chính xác cho Mask R-CNN *42*

DANH MỤC BẢNG

Bảng 1: Kết quả thử nghiệm trên ảnh gốc *38*
Bảng 2: Kết quả thử nghiệm với ảnh resize (256x256) *38*
Bảng 3: Kết quả với ảnh crop kích thước 256x256 ..... *40*
Bảng 4: Kết quả huấn luyện dữ liệu trên Kinect 1 ..... *40*
Bảng 5: Kết quả huấn luyện dữ liệu trên Kinect 3 ..... *41*
Bảng 6: Kết quả huấn luyện dữ liệu trên Kinect 5 ..... *41*

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Từ viết tắt, thuật ngữ	**Từ đầy đủ**
CNN	Convolutional Neural Networks
RCNN	Region-based Convolutional Neural Networks
Mask RCNN	Mask Convolutional Neural Networks
FPN	Feature Pyramid Network
STIP	Spatio-temporal interest points
SPP	Spatial Pyramid Pooling
GPU	Graphical Processing Unit
RoI	Region of Interest
SVM	Support Vector Machine

MỞ ĐẦU

Trong ĐATN này, em tập trung giải quyết bài toán phát hiện và phân vùng đối tượng bàn tay trên ảnh hoặc video. Bài toán phát hiện và phân vùng đối tượng bàn tay là một pha quan trọng trong các hệ thống nhận dạng hoạt động của bàn tay người. Mặc dù đã được nghiên cứu rộng rãi từ vài thập kỉ gần đây tuy nhiên đây vẫn là một nhiệm vụ thách thức với rất nhiều vấn đề được đặt ra như sự thay đổi của điều kiện chiếu sáng, tốc độ chụp camera khác nhau trong khi cử chỉ bàn tay khá nhanh và đặc biệt bàn tay người có rất nhiều hình dạng, tư thế khác nhau, bàn tay có thể đang cầm nắm các vật dụng hay tương tác với tay còn lại.

Trong đồ án này em tìm hiểu và ứng dụng một kỹ thuật học sâu tiên tiến, vốn đã rất hiệu quả trong các bài toán phát hiện và phân đoạn đối tượng nói chung, nhưng chưa được kiểm chứng trên đối tượng bàn tay với những thách thức như nêu trên. Có nhiều kiến trúc học sâu đã đạt kết quả tốt cho việc giải bài toán phát hiện, phân đoạn đối tượng trong thời gian gần đây. Trong đồ án này em nghiên cứu phương pháp Mask R-CNN được đề xuất trong tài liệu tham khảo [1] do tác giả Kaiming He và đồng nghiệp đề xuất. Phương pháp này đã được đánh giá thực nghiệm tốt trên các bộ dữ liệu dùng chung của cộng đồng nghiên cứu như COCO, ImageNet. Để thực hiện trên đối tượng bàn tay, em đã tìm hiểu và huấn luyện lại mạng Mask R-CNN dựa trên tập mẫu là tập cử chỉ bàn tay. Kết quả thực nghiệm đã được đánh giá trên một CSDL đa thể thức đa góc nhìn gồm 5 loại cử chỉ thực hiện bởi 5 người khác nhau trong môi trường tự nhiên. Các kết quả thực nghiệm cho thấy giải thuật thực hiện tốt khi camera ở góc nhìn trực diện hoặc có góc lệch trong khoảng 45 độ. Khi góc nhìn lệch đến 90 độ, độ chính xác phát hiện giảm nhanh chóng do bàn tay bị che khuất nhiều, hình ảnh của bàn tay mờ đi do chuyển động (motion blur) hoặc bàn tay ở vị trí gần với khuôn mặt có tính chất màu da tương tự.

Để cải thiện kết quả phát hiện trong những tình huống như vậy, em đã ứng dụng kỹ thuật theo bám đối trượng Mean Shift được đề xuất trong [2]. Giải thuật này cho phép theo vết các đối tượng (bàn tay) trên các ảnh khi không phát hiện được bởi giải thuật Mask R-CNN, vì thế khắc phục được hiện tượng phát hiện thiếu và loại bỏ một số các trường hợp phát hiện nhầm.

Đề tài này em thực hiện tại phòng Computer Vision, Viện MICA dưới sự hướng dẫn của PGS.TS. Trần Thị Thanh Hải. Trong đồ án tốt nghiệp này, em sẽ trình bày theo 4 chương:

- Chương 1: Phân tích các yêu cầu của bài toán phát hiện và phân vùng đối tượng bàn tay trong ảnh, giới thiệu các phương pháp đang được sử dụng để giải quyết, định hướng giải quyết của bài toán và tóm tắt lý thuyết và các nghiên cứu liên quan.
- Chương 2: Trình bày các tìm hiểu về kỹ thuật học sâu Mask R-CNN, và thuật toán theo vết đối tượng Mean shift.
- Chương 3: Triển khai mô đun phát hiện và phân đoạn bàn tay và đánh giá thử nghiệm
- Chương 4: Kết luận và hướng phát triển

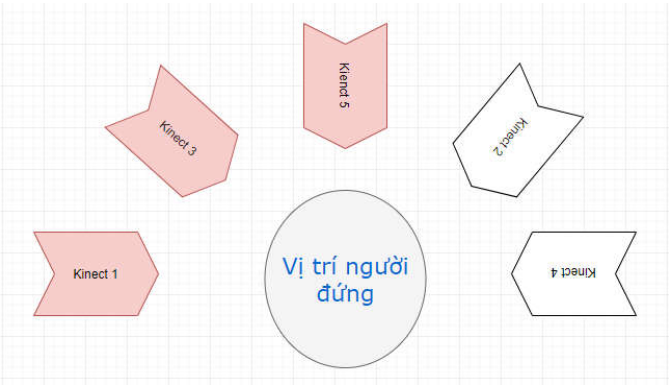
CHƯƠNG 1: TỔNG QUAN VỀ PHÁT HIỆN VÀ PHÂN VÙNG ĐỐI TƯỢNG BÀN TAY VÀ VẤN ĐỀ ĐẶT RA VỚI ĐỒ ÁN

1.1. Ngữ cảnh của bài toán

Phát hiện đối tượng bàn tay người là một bài toán cơ bản trong lĩnh vực thị giác máy tính. Ứng dụng của nó rất đa dạng, bao gồm hệ thống giám sát hoạt động tay, nhận dạng cử chỉ và điều khiển các thiết bị điện trong nhà thông minh. Một ví dụ đơn giản là dùng cử chỉ tay bật tắt quạt, điều hòa trong một căn phòng. Để các hệ thống trên hoạt động một cách nhanh chóng và chính xác thì phát hiện đối tượng bàn tay là một pha hết sức quan trọng. Đối tượng bàn tay trong môi trường bình thường có rất nhiều các tư thế, góc nhìn khác nhau như bàn tay nắm, xòe, giơ cao, cầm nắm vật thể. Vì vậy bài toàn phát hiện và phân vùng bàn tay vẫn còn là một thách thức.

Trong khuôn khổ của ĐATN, em hướng đến giải quyết bài toán phát hiện và phân vùng đối tượng bàn tay người trong một ứng dụng điều khiển thiết bị trong phòng thông minh sử dụng cử chỉ bàn tay. Môi trường thử nghiệm là một căn phòng được bài trí với các vật dụng giống như một phòng trong nhà hoặc căn hộ. Phòng được bố trí 5 Kinect nhìn theo 5 hướng khác nhau nhằm ghi lại cử chỉ điều khiển các thiết bị của một người đứng tại một vị trí cụ thể (giữa phòng) như Hình 1. Tập 5 cử chỉ đã được định nghĩa từ trước để thực hiện các lệnh điều khiển cơ bản đối với đèn như bật, tắt, tăng, giảm độ sáng, v.v đã được trình bày trong một LATS về nhận dạng cử chỉ động của bàn tay ứng dụng điều khiển thiết bị trong phòng thông minh. Tuy nhiên, nghiên cứu mới này tập trung đánh giá độ bền vững của các giải thuật đề xuất dựa khi có sư thay đổi của góc nhìn. Vì vậy nhiều Kinect đã được thiết lập để thu cùng một lúc các hình ảnh về cử chỉ động của bàn tay.

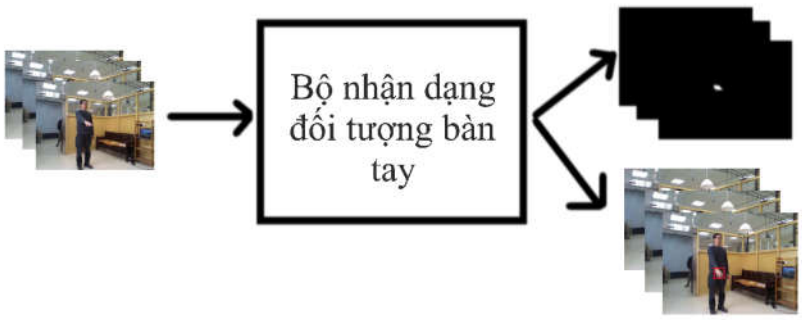
Với bài toán này, các thách thức cơ bản đặt ra đó chính là sự thay đổi góc nhìn khi cùng quan sát một cử chỉ. Có những góc nhìn trực diện (Kinect 5) nên khá thuận lợi để quan sát trong khi lại có những góc nhìn nghiêng khó hơn (Kinect 3, Kinect 2) hoặc có những góc nhìn thách thức do bàn tay bị che khuất một phần hoặc hoàn toàn (Kinect 1, Kinect 4).



Hình 1: Sơ đồ bố trí các Kinect thu thập dữ liệu cử chỉ người điều khiển ở các góc nhìn khác nhau.

Bài toán phát hiện và phân vùng đối tượng bàn tay người trong một video, hoặc một ảnh được định nghĩa như sau. Hình 2 minh họa đầu vào của hệ thống là các ảnh liên tiếp thu được và đầu ra là các vùng bàn tay đã được phân đoạn và bao đóng của bàn tay được đóng khung ở trên ảnh.

Bài toán phát hiện và phân vùng bàn tay người trên ảnh
Đầu vào: Ảnh của người được thu bằng cảm biến hình ảnh
Đầu ra: Bao đóng và phân vùng đối tượng bàn tay người trên ảnh



Hình 2: Minh họa đầu vào và đầu ra của bài toán phát hiện và phân vùng đối tượng bàn tay

Phát hiện và phân vùng đối tượng là một bài toán có nhiều thách thức đối với các nhà khoa học do nhiều nguyên nhân như nhiễu nền, góc nhìn thay đổi, đa dạng trong thực hiện

hoạt động của từng người. Để việc phát hiện được chính xác cần có phương pháp biểu diễn tốt cũng như cần có một bộ dữ liệu đủ lớn và đa dạng để cho việc học có hiệu quả cao. Trong đồ án này em tìm hiểu các nghiên cứu liên quan, từ đó đề xuất giải pháp cho bài toán phát hiện và phân vùng đối tượng bàn tay trên 3 góc nhìn khác nhau (K1, K3, K5) như trên Hình 1. Các kết quả phân tích trên ba góc nhìn này có thể phần nào đánh giá được tính bền vững của phương pháp lựa chọn nghiên cứu đối với sự thay đổi của góc nhìn.

Kinect là thiết bị có chức năng thu ảnh RGB như camera bình thường kèm theo cảm biến độ sâu đo khoảng cách từ các điểm ảnh thu được tới Kinect với tốc độ khoảng 30 hình/giây với ảnh có độ phân giải 480x640 và 10 hình /giây với ảnh có độ phân giải 720x1080. Mặc dù có dữ liệu về ảnh độ sâu tuy nhiên trong đồ án này em mới thực hiện trên ảnh RGB thông thường 480x640. Thông tin độ sâu sẽ được khai thác và nghiên cứu trong các công việc tiếp theo.

1.2. Các thành phần chính của hệ thống phát hiện và phân vùng đối tượng

Một hệ thống phát hiện và phân vùng đối tượng ảnh dựa trên các kỹ thuật học có giám sát (để phân lớp đối tượng quan tâm với các đối tượng ảnh còn lại) thông thường được thực hiện thông qua 2 pha sau:

- Pha huấn luyện: Sử dụng bộ dữ liệu học đưa vào huấn luyện để đưa ra mô hình nhận dạng.
- Pha thử nghiệm: Sử dụng mô hình vừa huấn luyện được ở trên để thực hiện phân lớp trên dữ liệu mới.

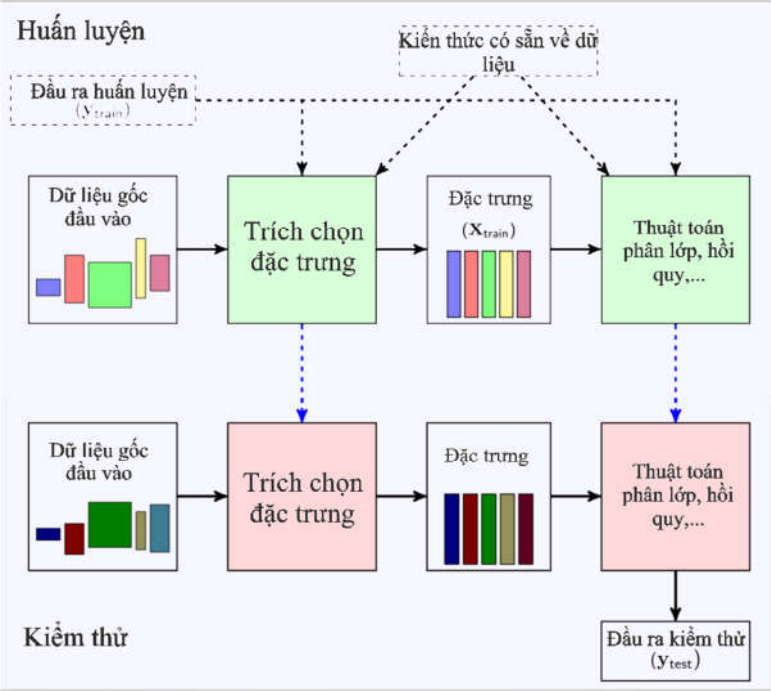
Hình 3 mô tả mô hình chung cho một bài toán áp dụng kĩ thuật học máy.

Áp dụng cho bài toán phát hiện, phân vùng đối tượng mỗi pha đều có các khối xử lý cơ bản như sau:

- Tiền xử lý dữ liệu: Chuyển đổi dữ liệu đầu vào sang định dạng chuẩn với mô hình.
- Trích chọn đặc trưng: Từ dữ liệu đã được tiền xử lý, thực hiện trích rút các đặc trưng biểu diễn đối tượng quan tâm.
- Huấn luyện: sử dụng các đặc trưng được trích chọn để làm dữ liệu đầu vào cho việc huấn luyện các khối trong hệ thống có khả năng huấn luyện được như khối trích trọn đặc trưng, khối phân lớp,...
- Dự đoán bao đóng đối tượng: dữ liệu cần phát hiện được chuyển qua các bước tiền xử lý, trích chọn đặc trưng, sau đó dùng bộ phân lớp đã được huấn luyện để dự đoán bao đóng có chứa đối tượng trên ảnh

- Phân vùng: Từ ánh xạ đặc trưng trích đưa các vùng quan tâm vào mạng để phân lớp từng pixel ảnh để dự đoán mặt nạ trên ảnh. Phân vùng và dự đoán bao đóng đối tượng có thể tách rời hoặc nằm chúng trong một khối tùy hệ thống.

Hình 3 minh họa sơ đồ tổng thể hệ thống với các thành phần như phân tích ở trên.



Hình 3 Sơ đồ tổng thể hệ thống áp dụng học máy (https:machinlearningcoban.com)

1.3. Một số nghiên cứu liên quan về phát hiện và phân đoạn bàn tay người trong ảnh

Có nhiều phương pháp đã được đề xuất cho bài toán phát hiện và phân đoạn bàn tay trong ảnh. Các phương pháp này có thể được chia thành hai loại: nhóm các phương pháp dựa trên các đặc trưng trích chọn biểu diễn bàn tay được thiết kế từ trước và nhóm các phương pháp dựa trên các đặc trưng học được từ dữ liệu. Phần dưới đây sẽ trình bày tóm lược những tìm hiểu của em về một số phương pháp thuộc từng nhóm tiếp cận này.

1.3.1. Hướng tiếp cận phát hiện đối tượng bàn tay trên ảnh dựa vào các đặc trưng được thiết kế bằng tay

Đặc trưng thiết kế bằng tay (hand crafted feature) là các đặc trưng được thiết kế từ trước, nhằm đưa ra cấu trúc đặc trưng mới phù hợp nhất với từng đối tượng hoặc hoạt động. Nhờ vậy mà các mô hình cải thiện được độ chính xác của mình. Đây là công việc đòi hỏi sự sáng tạo và thời gian của các nhà khoa học máy tính. Các đặc trưng giúp cho việc chuyển đổi dữ liệu thô ban đầu thành tập các thuộc tính giúp biểu diễn dữ liệu tốt hơn, giúp tương thích với từng mô hình dự đoán cụ thể, cũng như cải thiện độ chính xác của mô hình hiện tại.

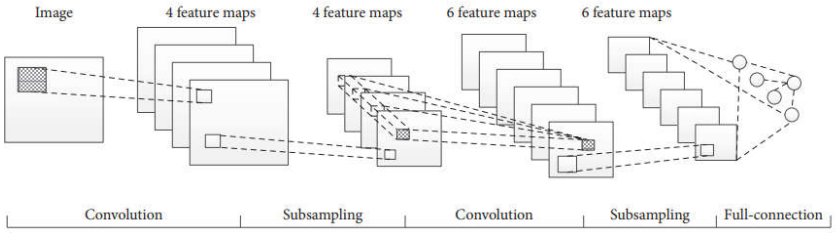
Một phương pháp cơ bản nhất được đề xuất trong [3] dựa trên phát hiện vùng màu da. Tuy nhiên phương pháp này không hiệu quả khi vùng màu da của bàn tay bị trùng với các các vùng như mặt, cánh tay. Bên cạnh đó phương pháp này còn bị ảnh hưởng nhiều bởi điều kiện chiếu sáng. Một phương pháp phổ biến để phát hiện rất nhiều các đối tượng khác nhau trong ảnh là sử dụng đặc trưng Haar-like và bộ phân lớp Adaboost được đề xuất trong [4] và [5]. Phương pháp này được áp dụng rộng rãi trọng việc phát hiện người, mặt, động vật,.. Tuy nhiên vấn đề sử dụng với đối tượng bàn tay gặp phải là hình dạng bàn tay trong môi trường thực có khá nhiều, thay đổi theo các góc nhìn khác nhau, do vậy sử dụng đặc trưng Haar-like là không đủ mạnh cho bài toán này.

Hiện nay, trên thế giới đã có rất nhiều đặc trưng khác được thiết kế để giải quyết bài toán nhận dạng đối tượng. Trong [6], các tác giả đề xuất sử dụng đặc trưng HOG (Histogram of Oriented Gradient) trên ảnh RGB để nhận dạng đối tượng mặt và hai mắt người. Felzenszwalb và đồng nghiệp đã đề xuất mô hình Deformable Part Model (Mô hình phần biến dạng) trong [7] DPM. Các tác giả đã tính toán đặc trưng HOG trên các thành phần của hình ảnh tương ứng với các bộ phận của đối tượng quan tâm từ đó đưa vào phát hiện đối tượng. DPM cho kết quả tốt trong cuộc thi Visual Object Classes (VOC) về phát hiện đối tượng được tổ chức từ năm 2007 đến năm 2009. Gần đây Mittal và đồng nghiệp đã áp dụng DPM trong [8] để phát hiện vùng tay dựa trên các bộ phát hiện hình dạng, bộ phát hiện màu da, và dò theo ngữ cảnh.

1.3.2. Hướng tiếp cận biểu diễn phát hiện và phân vùng dựa trên kỹ thuật học sâu

Kỹ thuật học sâu (Deep learning) là một thuật toán học máy được xây dựng dựa trên một số ý tưởng mô phỏng hệ thống não bộ của con người. Nó biểu diễn dữ liệu thông qua nhiều tầng từ cụ thể đến trừu tượng qua đó trích rút được các đặc trưng có ý nghĩa trong nhận

dạng đối tượng ảnh. Thuật toán học sâu đã đạt được nhiều thành công trong bài toán xử lý ảnh hay nhận dạng giọng nói. Theo phương pháp này các đặc trưng không cần phải thiết kể một cách thủ công mà sẽ được học một cách tự động bởi các mạng neuron sâu thông qua các bộ dữ liệu. Hình 4 minh họa một kiến trúc mạng neuron sâu gồm 2 tầng tích chập, 2 tầng lấy mẫu và 2 tầng kết nối đầy đủ. Mô hình này đã được sử dụng để giải quyết cho bài toán nhận dạng chữ viết tay được đề xuất trong [9] bởi tác giả Lecun và đồng nghiệp.



Hình 4 Kiến trúc cơ bản của một mạng CNN [9]

Szegedy và đồng nghiệp đã áp dụng riêng biệ mạng CNN cho việc phát hiện bao đóng và phân lớp đối tượng để kiểm tra liệu bao đóng có chứa đối tượng không và đối tượng đó là gì. Trong bài báo [10], các tác giả đã đề xuất một kiến trúc mạng neuron tích chập R-CNN (Region-based Convolutional Neural Networks). Ý tưởng chính của phương pháp này là sử dụng một thuật toán dự đoán vùng chứa đối tượng (selective-search [11]) để sinh ra các vùng đề xuất và CNN sẽ điều chỉnh vùng đề xuất đó để tạo ra các vùng chứa đối tượng quan tâm. Đặc trưng ảnh được trích xuất bởi mô hình CNN để hướng tới nhận dạng bằng giải thuật SVM (Support Vector Machine). Mạng R-CNN cho kết quả khá tốt về độ chính xác cũng như thời gian tính toán so với các phương pháp đã có. Trong những năm gần đây R-CNN có nhiều ứng dụng trong việc phát hiện nhiều các đối tượng khác nhau, trong đó có một nghiên cứu áp dụng R-CNN cho việc phát hiện đối tượng bàn tay như trong [12] được tác giả Shiyang Yan và đồng nghiệp nghiên cứu cho kết quả khá tốt trên các tập dữ liệu bàn tay như tập Oxford Hand Dataset, VIVA Hand Detection. Trong [13] Tác giả T. Hoang Ngan Le và đồng nghiệp có đề xuất mạng Multi-scale Region-base Fully Convolution Networks cho bài toán phát hiện tay trên vô lăng ô tô – một bài toán có ứng dụng thực tế khá thú vị đã cho kết quả khá tốt AP: 86.0% trên tập VIVA và 75.1% trên tập Oxford.

Bên cạnh việc phát hiện đối tượng bàn tay vấn đề đặt ra trong đồ án này của em là phân vùng đối tượng trên ảnh. Đây cũng là một vẫn đề thách thức không kém trong thị giác máy tính. Có 2 hướng tiếp cận chính cho việc giải quyết bài toán này là dựa vào kết quả dự đoán bao đóng đối tượng và phân vùng đối tượng trên bao đóng. Dai và đồng nghiệp đã đề xuất

trong [14] một mô hình phức tạp nhiều tầng theo kiểu thác nước dự đoán phân vùng đối tượng dựa trên hộp bao theo sau bằng một khối phân lớp. Đối với đối tượng bàn tay gần đây cũng có khá nhiều nghiên cứu liên quan, trong đó nổi bật là nghiên cứu của Kankana Roy và đồng nghiệp trong [15] áp dụng Faster R-CNN phát hiện đối tượng và kết hợp một mạng phân vùng vùng màu da trên bao đóng vùng bàn tay cho kết quả khá tốt trên các tập dữ liệu bàn tay đã đề cập ở trên đồng thời trên tập ICD do họ chuẩn bị. Đây là tập dữ liệu dựa trên hình ảnh, video múa cổ truyền của Ấn Độ (Indian Classical Dance). Trong tập ICD bàn tay có rất nhiều hình dạng khác nhau kèm theo đó là có rất nhiều trang sức được đeo trên tay nghệ sĩ múa. Tuy nhiên mô hình đề xuất vẫn cho kết quả khá tốt. Vấn đề của nghiên cứu này đó là việc sử dụng kết hợp 2 mạng neuron làm cho hệ thống hoạt động chậm, thời gian huấn luyện lâu.

Trong những nghiên cứu gần đây cho bài toán phát hiện và phân vùng đối tượng có một nghiên cứu của tác giả Kaiming He và đồng nghiệp tại Facebook Research dựa trên kiến trúc R-CNN là Mask R-CNN phát triển mạng end-to-end cho việc phát hiện và phân vùng đối tượng cho kết quả rất tốt trên tập dữ liệu COCO và Imagenet.

1.3.3. Nhận xét chung

Hướng biểu diễn dựa trên đặc trưng được trích xuất bằng tay cho kết quả rất tốt trên một số tập dữ liệu tuy nhiên với các điều kiện thay đổi phương pháp này không còn giữ được độ chính xác. Thêm nữa để thiết kế ra các đặc trưng tùy thuộc rất nhiều vào dữ liệu sử dụng cho bài toán cụ thể.

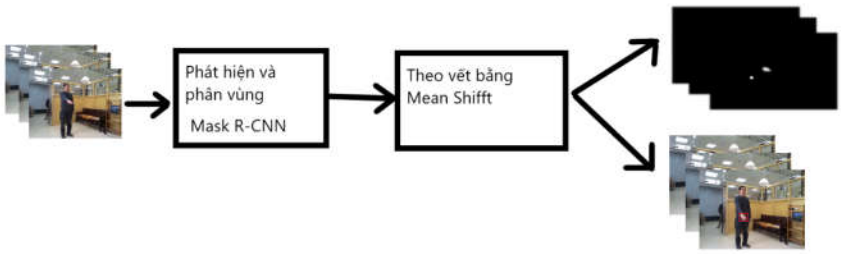
Kỹ thuật học sâu có thể cải thiện phần nào các thách thức này. Khác với đặc trưng trích xuất bằng tay, kỹ thuật học sâu không phụ thuộc vào sự đa dạng của bộ dữ liệu. Kỹ thuật này có thể xây dựng bộ trích xuất đặc trưng dựa trên việc học cách biểu diễn chúng. Tuy nhiên, kỹ thuật này đòi hỏi một lượng dữ liệu đủ lớn để đạt độ chính xác cho mô hình và dữ liệu phải đủ tốt và bao quát toàn bộ các tình huống thực tế. Hơn nữa, việc tìm ra các tham số và kiến trúc phù hợp cần rất nhiều thời gian để thử nghiệm. Quá trình huấn luyện cũng như thử nghiệm yêu cầu khá cao về cấu hình phần cứng cụ thể là CPU, GPU do quá trình tính toán rất nhiều.

1.4. Mục tiêu của đồ án

Kỹ thuật học sâu là kỹ thuật có nhiều tiềm năng trong tương lai. Rất nhiều nhà nghiên cứu đã thử nghiệm trên các mô hình khác nhau để tìm ra một kiến trúc phù hợp. Hơn nữa, với sự phát triển nhanh chóng của khoa học và kỹ thuật, máy tính, siêu máy tính với card đồ

họa cho phép giải các bài toán dữ liệu lớn. Nhờ đó việc thời gian tính toán được giảm đi hàng chục lần so với ban đầu.

Dựa vào các yêu cầu bài toán cụ thể là phát hiện và phân vùng đối bàn tay, dữ liệu đã được chuẩn bị sẵn với số lượng ảnh đủ lớn. Đặc biệt là trong thời gian gần đây việc áp dụng kĩ thuật học sâu để giải quyết các bài toán nhận dạng và phân vùng được nghiên cứu thử nghiệm khá nhiều và cho kết quả khá tốt như em đã trình bày ở trên. Trong ĐATN này, em lựa chọn kỹ thuật học sâu để giải quyết bài toán này. Qua thời gian tham khảo tài liệu và nghiên cứu các công việc liên quan em lựa chọn kiến trúc Mask R-CNN được trình bày trong [1] để tìm hiểu và thử nghiệm cho bài toán phát hiện và phân vùng đối tượng bàn tay người . Em cho rằng đây là một kỹ thuật đã được đánh giá thử nghiệm với nhiều bộ tham số khác nhau và đã cho kết quả khá cao với bộ dữ liệu có rất nhiều đối tượng có kích cỡ rất khác nhau. Các nghiên cứu sẽ được thử nghiệm trên tập dữ liệu cử chỉ bàn tay thu thập tại Viện MICA. Tập dữ liệu này được gán nhãn bằng công cụ bán tự động sẽ được sử dụng một phần để huấn luyện và một phần đánh giá. Do kích thước bàn tay nhỏ nên việc gán nhãn bằng tay đôi khi vẫn còn sai sót. Hơn nữa thiết bị thu dữ liệu còn nhiều hạn chế về tốc độ chụp gây ra các hiệu ứng ảnh mờ khó nhận biết làm cho kết quả phát hiện bởi Mask R-CNN. Vì vậy em áp dụng thuật toán tracking đối tượng trên ảnh kết hợp để tăng độ chính xác cho mô hình.



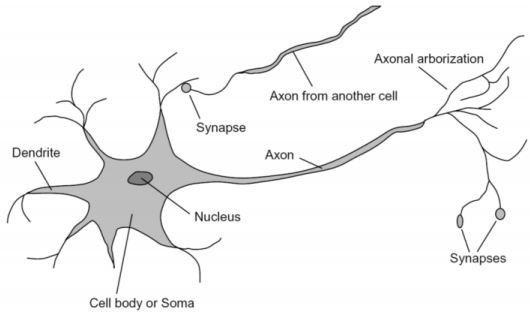
Hình 5: Sơ đồ khối chung của hệ thống

CHƯƠNG 2: KỸ THUẬT HỌC SÂU MASK R-CNN VÀ THUẬT TOÁN THEO VẾT MEAN SHIFT

2.1. Giới thiệu chung về mạng nơ ron tích chập

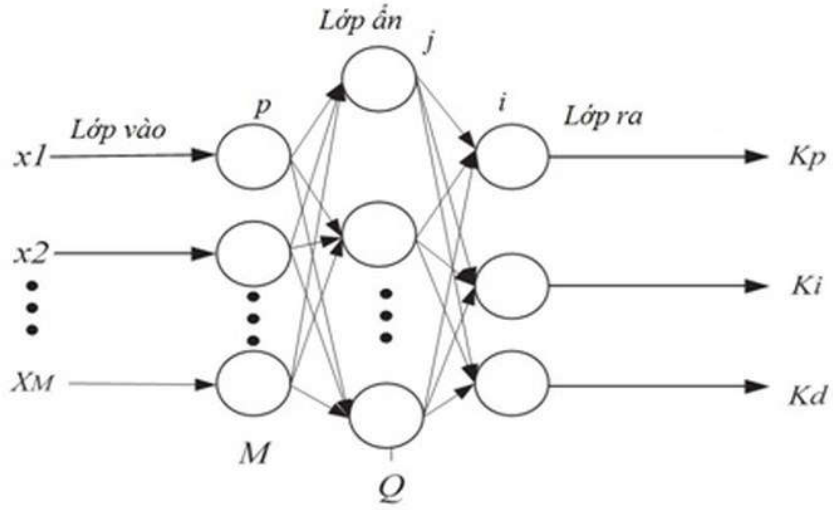
2.1.1. Mạng nơ ron

Mạng nơ ron nhân tạo được thiết kế nhằm mô phỏng mạng neuron của bộ não người Mạng neuron nhân tạo được cấu thành từ một tập các phần tử xử lý đơn giản được kết nối với nhau. Mỗi phần tử xử lý này chỉ có thể thực hiện được một thao tác tính toán nhỏ, nhưng một mạng lưới các phần tử như vậy có một khả năng tính toán lớn hơn rất nhiều. Phần tử tính toán cơ bản của mạng nơ ron là một perceptron hay một nơ ron.



Hình 6: Minh họa mạng nơ ron của người (từ Rob Fergus)

Một nơ ron mô phỏng quá trình tính toán của bộ não con người. Dữ liệu được đưa tới các nơ ron thông qua các Dendrite vào Nucleus để tính toán. Tín hiệu ra được xuất ra ở dây Axon. Các nơ ron được liên kết với nhau thông qua các dây Synapse. Mạng nơ ron bao gồm rất nhiều phần tử như vậy liên kết với nhau.



Hình 7: Mạng nơ ron nhiều tầng

Hình 7 minh họa một mạng nơ ron truyền thẳng với 3 loại node sau:

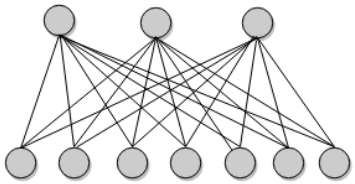
- Input nodes (Node đầu vào): Chứa dữ liệu đầu vào từ bên ngoài và đưa trực tiếp vào các Hidden nodes.
- Hidden nodes (Node ẩn): Nó không chứa kết nối trực tiếp đến dữ liệu từ bên ngoài. Nó thực hiện tính toán các dữ liệu nhận được từ các input nodes, thực hiện tính toán và đưa ra các output nodes. Tập hợp các node ẩn trong mạng tạo thành tầng ẩn. Một mạng nơ ron truyền thẳng có thể có hoặc không có tầng ẩn.
- Output nodes (Node đầu ra): Có nhiệm vụ tính toán và đưa dữ liệu từ trong mạng ra bên ngoài.

2.1.2. Mạng nơ ron tích chập

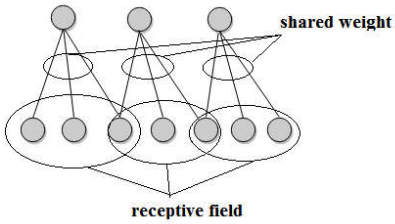
Mạng nơ ron tích chập là một mạng neuron nhân tạo với các toán tử tích chập. Nó có khả năng học một lượng lớn các dữ liệu trong khoảng thời gian ngắn hơn nhiều so với mạng nơ ron thông thường. Lý do là nó sử dụng ít trọng số hơn trong khi độ chính xác chỉ kém hơn một phần nhỏ so với kiến trúc truyền thống. Mô hình này sử dụng trong [16] và đã đạt kết quả khá tốt trong bài toán phân loại ảnh.

Trong mạng nơ ron truyền thống, các node ở các tầng phía sau sẽ liên kết với toàn bộ các node ở layer phía dưới thông qua một tập các trọng số. Với mỗi nơ ron khác nhau, chúng ta cần một tập trọng số hoàn toàn độc lập để liên kết với các nơ ron ở tầng trước đó.

Điểm khác biệt của mạng nơ ron tích chập so với mạng nơ ron truyền thống đó là trong liên kết giữa 2 tầng liên tiếp nhau việc các node ở các tầng phía sau chỉ liên kết với một bộ phận các node ở tầng phía trước đó gọi là receptive field thông qua một tập các trọng số. Hơn nữa tập trọng số này là như nhau đối với mỗi nơ ron ở tầng sau. Do đó số lượng tham số cần huấn luyện ít hơn trong khi vẫn giữ được lượng thông tin cần thiết.



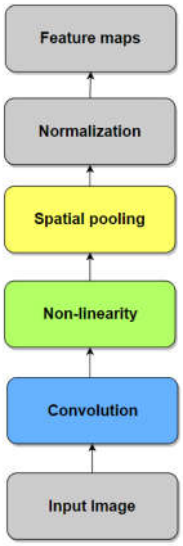
Hình 8: Kết nối giữa các tầng trong mạng nơ ron truyền thống



Hình 9: Kết nối giữa các tầng trong mạng nơ ron tích chập

Một mạng nơ ron tích chập thường được thực hiện thông qua các bước sau:

- Convolutional layer
- Pooling layer
- Non-linearity layer
- Fully-connected layer

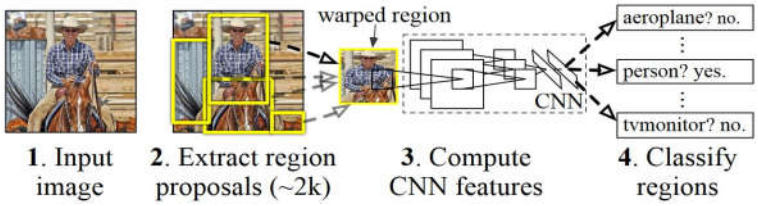


Hình 10: Các bước cơ bản trong mạng neuron tích chập

Trên đây là các khối cơ bản trong một mạng neuron tích chập, mạng neuron tích chập được cấu thành bằng việc xếp trồng nhiều các lớp thành một cấu trúc chặt chẽ. Trong nội dung đồ án này em sẽ không đề cập đến việc thiết kế mạng chỉ áp dụng những cấu trúc đã được nghiên cứu và cho hiệu quả tốt.

2.1.3. Kiến trúc của CNN cho bài toán nhận dạng và phân vùng đối tượng

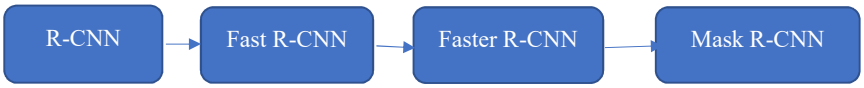
R-CNN



Hình 11: Cấu trúc mạng R-CNN [13]

Ở phần trên em đã giải thích về các kĩ thuật đã sử dụng để tạo ra các mạng nơ ron tích chập. Trong phần này em sẽ trình bày kiến trúc cụ thể sử dụng cho bài toán phát hiện và phân vùng đối tượng bàn tay em sử dụng trong đồ án này.

Để hiểu được kiến trúc Mask R-CNN em đi theo trật tự như sau:



Đây là trật tự phát triển của họ các thuật toán mạng neuron tích chập trên vùng được nghiên cứu và phát triển trong những năm gần đây.

R-CNN được đề xuất từ năm 2013 thể hiện 3 bước để nhận dạng đối tượng:

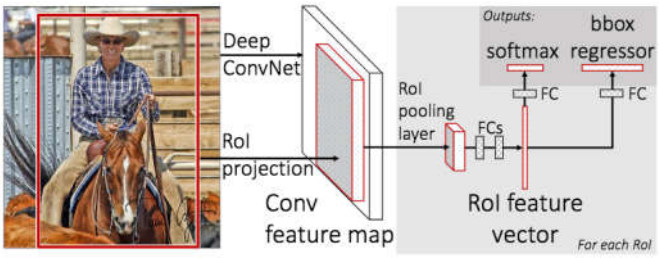
- Tạo ra các vùng bao đóng giới hạn trên ảnh bằng một thuật toán đề xuất vùng bên ngoài mạng như selective search
- Đóng gói dữ liệu từ vùng đề xuất (các bước tiền xử lí) để đưa vào mạng đã được học sẵn để trích dẫn ra các đặc trưng
- Truyền đặc trưng trích dẫn được từ bước trên đưa qua bộ phân lớp tách rời có thể là một mạng kết nối đầy đủ hoặc SVM để chọn xem bao đóng có thực sự chứa đối tượng quan tâm hay không.

Kiến trúc R-CNN đạt được hiệu quả độ chính xác khoảng 40% trên tập dữ liệu Pascal VOC (2010).

Hình 11 từ bài báo [17] mô tả chi tiết các khối của kiến trúc R-CNN. Tuy vậy mạng R-CNN vẫn gặp một số vấn đề sau Tăng độ chính xác của R-CNN chỉ là một phần của R-CNN, vẫn đề chủ yếu đặt ra với R-CNN đó là nó chạy khá chậm và qua một số thí nghiệm cho thấy vấn đề tốc độ bị giới hạn là do R-CNN chạy CNN độc lập cho mỗi bao đóng và việc sinh ra hộp bao bởi các thuật toán bên ngoài là rất nhiều. Cụ thể R-CNN chạy mất khoảng 40s cho mỗi ảnh trên GPU. Mô hình này rất khó để có thể học do có 2 phần mô hình học máy (CNN và SVM) cần học độc lập. Thêm nữa R-CNN yêu cầu một thuật toán bên ngoài như selective search.

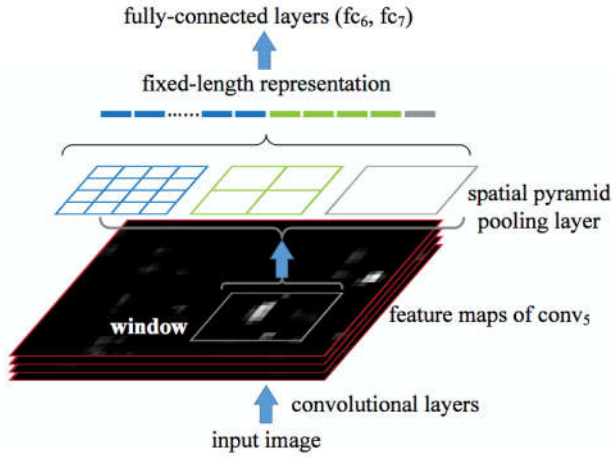
Fast R-CNN

Để khắc phục những hạn chế của R-CNN, một thuật toán mới cải tiến từ R-CNN đã được đề xuất với tên gọi Fast R-CNN. Cải tiến chính của thuật toán này so với R-CNN là cải tiến về tốc độ. Thời gian phát hiện đối tượng trên một ảnh giảm từ 40s xuống còn 0.2s tức là nhanh hơn khoảng 200 lần so với R-CNN. Ý tưởng chính của Fast R-CNN đó là sử dụng một mạng nơ ron duy nhất cho trích xuất đặc trưng và phân lớp thay thế cho mạng SVM độc lập. Kiến trúc cụ thể được mô tả trong hình 12 dưới đây:



Hình 12: Kiến trúc Fast R-CNN

Để phát hiện đối tượng một cách độc lập với kích cỡ ảnh, Fast R-CNN sử dụng một tầng Spatial Pyramid Pooling được giới thiệu trong [18]. Ý tưởng là thay vì cắt ảnh đầu vào thành nhiều các phần khác nhau, Fast R-CNN tính toán đặc trưng một lần qua mạng trích xuất đặc trưng và ánh xạ vùng quan tâm trên ảnh (được sinh ra nhờ thuật toán đề xuất selective search), mỗi vùng quan tâm trên đặc trưng được pooling theo các kích cỡ khác nhau rồi ghép lại thành một đặc trưng có kích cỡ cố định và đưa vào tầng tiếp theo, nhờ vậy mà đặc trưng tạo ra độc lập với kích cỡ vùng quan tâm. Bên cạnh đó dễ thấy kiến trúc này có thể tái sử dụng những lớp CNN ở tầng trên thay vì phải tính hàng nghìn lần như R-CNN. Spatial Pyramid Pooling được sử dụng để chuẩn hóa đầu vào cho tầng kết nối đầy đủ khi mà kích cỡ RoI là biến đổi. Hình 13 mô tả chi tiết cách Spatial Pyramid Pooling hoạt động:

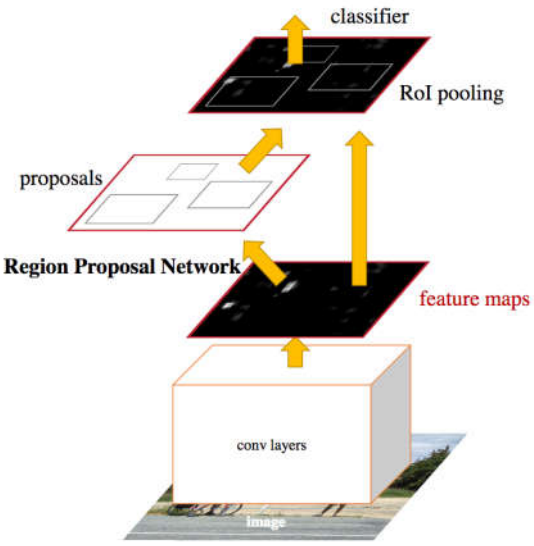


Hình 13: Hoạt động của Spatial pyramid pooling

Mặc dù nhanh hơn khoảng 200 lần so với R-CNN tuy nhiên Fast R-CNN vẫn chưa đủ nhanh để hoạt động trong các hệ thống yêu cầu thời gian thực.

Faster R-CNN

Nhanh hơn Fast R-CNN một mô hình có tên là Faster R-CNN được đề xuất trong [19]. Faster R-CNN cải tiến hơn so với Fast R-CNN bằng việc thay thế thuật toán sinh vùng độc lập với mô hình. Mạng Faster R-CNN chỉ cần sử dụng một mạng nơ ron duy nhất để thực hiện nhiệm vụ sinh vùng đặc trưng bằng việc thêm vào mạng một số tầng mới làm nhiệm vụ đề xuất vùng đặc trưng khối này có tên Region Proposal Network (RPN). Vì vậy kiến trúc Faster R-CNN được mô tả như hình 14:



Hình 14: Kiến trúc Faster R-CNN

RPN là một mạng tích chập đầy đủ (Fully convolution network) ko có các tầng kết nối đầy đủ do đó chi phí tính toán không đáng kể khoảng 10ms cho mỗi ảnh. Ngoài ra Faster R-CNN còn thay thế các Spatial Pyramid bằng các Anchor với các tỉ lệ khung khác nhau.

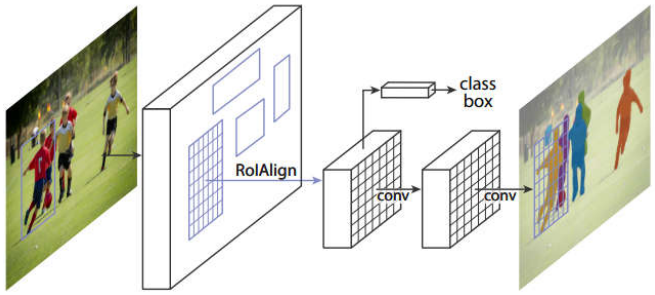
2.2. Mạng Mask R-CNN

Mask R-CNN được phát triển bởi Kaiming He và đồng nghiệp. Mask R-CNN kế thừa từ kiến trúc Faster R-CNN bằng cách thêm một nhánh làm nhiệm vụ dự đoán phân vùng đối tượng song song với nhánh dự đoán lớp đối tượng và hộp bao. Tốc độ Mask RCNN đạt được khá nhanh khoảng 5 frame trên giây. Mask R-CNN là mô hình tốt nhất trên tập dữ liệu COCO năm 2016. Bên cạnh đó việc cài đặt và sử dụng cũng tương đối đơn giản. Vì vậy em đã chọn Mask R-CNN là để nghiên cứu cho bài toán cụ thể là phát hiện và phân vùng bàn tay trong

ảnh và video, một pha tiền xử lý cho hệ thống nhận dạng cử chỉ động của bàn tay trong ứng dụng điều khiển thiết bị trong phòng thông minh.

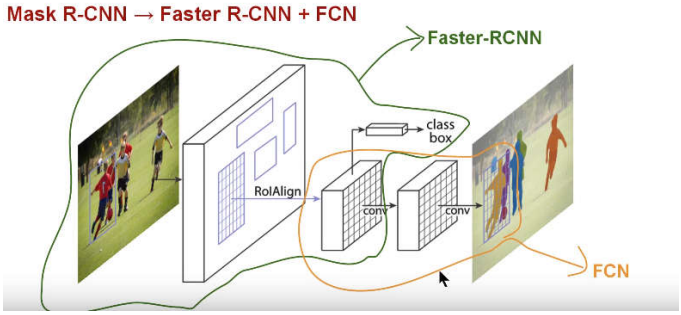
Kiến trúc mạng Mask R-CNN

Mask R-CNN mở rộng Faster R-CNN bằng cách thêm nhánh dự đoán mặt nạ đối tượng trên mỗi vùng đề xuất từ RPN. Kiến trúc các phần còn lại tương đối giống với Faster R-CNN. Hình 15 mô tả kiến trúc Faster R-CNN:



Hình 15: Kiến trúc Mask R-CNN

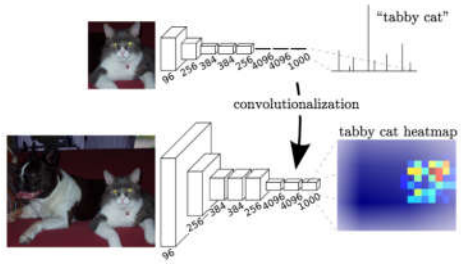
Nhánh phía trên dùng dự đoán hộp bao và lớp đối tượng trên ảnh, và nhánh dưới là nhánh làm nhiệm vụ đánh nahnx cho mỗi pixel trong vùng quan tâm (RoI) để xây dựng mặt nạ đối tượng. Hình 16 giải thích lại kiến trúc Mask R-CNN như sau:



Hình 16: Giải thích kiến trúc Mask R-CNN (Medium.com)

Như đã đề cập ở trên FCN được sử dụng ở cả 2 nhiệm vụ phát hiện đối tượng và phân vùng đối tượng. Một mạng nơ ron bình thường được sử dụng cho phát hiện và nhận dạng đối tượng ở tầng cuối thường là một vector có cùng kích thước với số lớp và cho biết điểm số dự đoán của mỗi lớp. Nếu chúng ta dừng lại ở một số lớp trung gian của mạng và thay thế vector bởi một số bước tích chập, và thay vì một vector có kích thước bằng số lớp ta sẽ có số lượng đặc trưng cùng kích thước với số lớp. Sau quá trình học thích hợp chúng ta có được điểm số dự đoán lớp cho tất cả các điểm ảnh của lớp cuối cùng. Mỗi lớp sẽ nhận được một

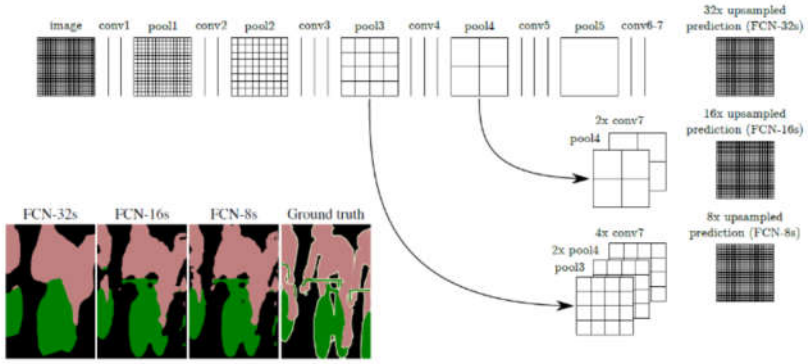
"bản đồ nhiệt". Đây chính là cách hoạt động của bộ phân loại trên top và tích chập đầy đủ bên dưới hoạt động.



Hình 17: Cách hoạt động của mạng tích chập đầy đủ

Phần phân đoạn của Mask R-CNN sẽ sử dụng kết quả "bản đồ nhiệt" như trên và từ đây ta deconvolution và unpooling để thu được mặt nạ trên ảnh gốc. Cụ thể cách thức hoạt động của 2 bước này như sau:

- Deconvolution (giải tích chập): thực chất chỉ là tích chập với ma trận chuyển vị
- Unpooling: để hiểu về unpooling ta cùng xem lại về pooling. Với max-pooling ta lấy giá trị max của khối vì vậy thông tin sẽ bị mất mát đi trong quá trình pooling. Unpooling là quá trình ta xây dựng lại ma trận bằng cách ghi nhớ tọa độ của điểm max và điền lại chính xác điểm max còn lại các điểm khác trong khối sẽ được sắp xỉ từ điểm đã có giá trị. Thông tin mất mát nhưng trong trường hợp này chúng ta có thể chấp nhận được.



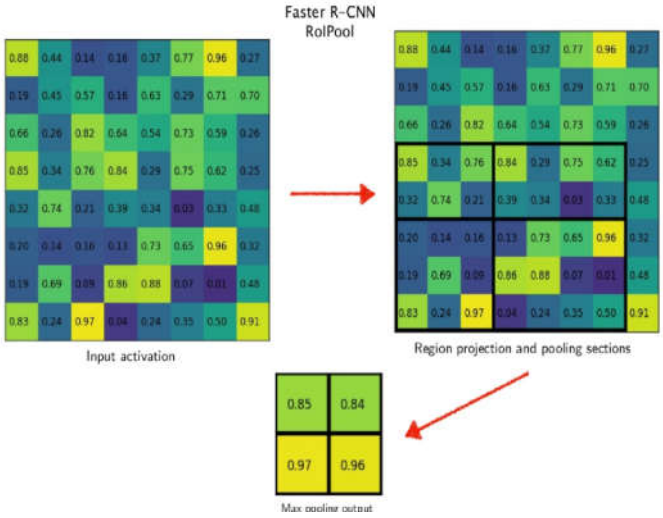
Hình 18: Mô tả cách hoạt động của khối deconvolution và unpooling

Qua việc deconvolution và unpooling, chúng ta có thể xây dựng một phân vùng dự đoán trên ảnh gốc cho tất cả các lớp đối tượng. Đây cũng chính là đầu ra cho khối phân vùng đối tượng.

RoIAlign

Ngoài ra so với Faster R-CNN Mask RCNN có một cải tiến đó là thay thế việc sử dụng khôi RoI Pooling bằng một khối có tên là RoIAlign. Theo tác giả đây là phần rất quan trọng để cải thiện độ chính xác cho Mask RCNN.

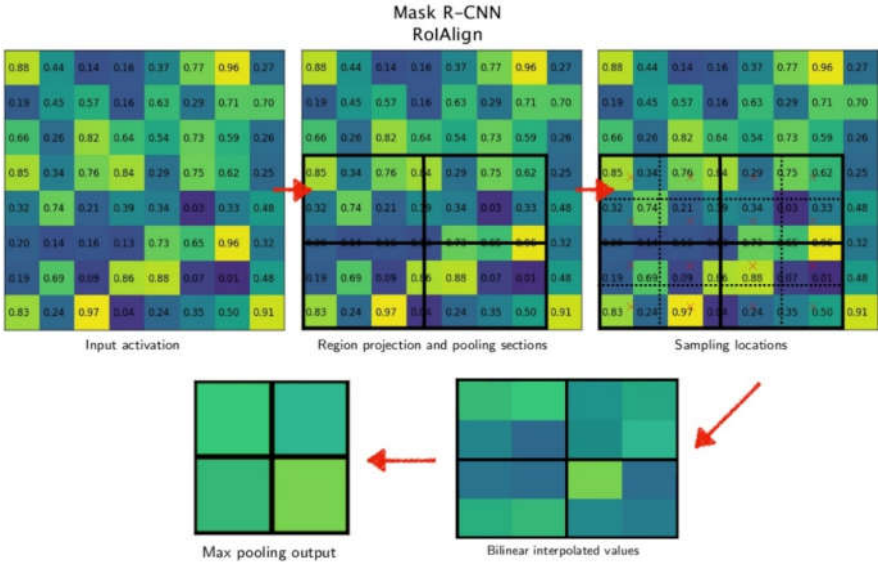
RoIPooling hay RoIAlign đều có nhiệm vụ chính là chuẩn hóa kích cỡ của vùng quan tâm đưa vào tầng kế tiếp. Ví dụ sau đây sẽ cho thấy RoIPooling làm cho mô hình mất đi độ chính xác:



Hình 19: Max Pooling

Khá đơn giản và dễ sử dụng tuy nhiên một phần thông tin cục bộ đã bị mất đi khá nhiều.

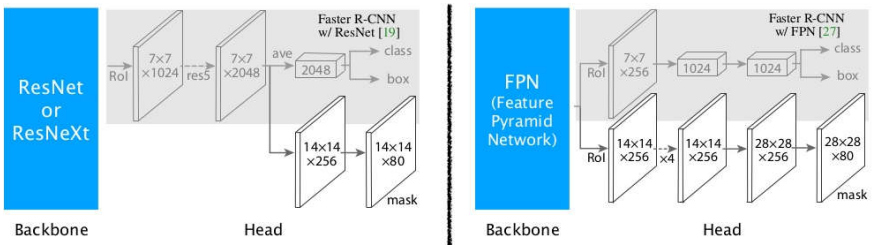
Sau đây là cách RoIAlign cải tiến so với RoIPooling được minh họa trong hình 19. Thay vì việc tính mỗi điểm trên đặc trưng thông qua việc lượng tử hóa từng khối nhỏ trên ma trận để thu được kích cỡ cố đihj thì RoIAlign thực hiện một phép nội suy phi tuyến để tính ra từ đặc trưng của mỗi vùng như hình dưới. Theo tác giả việc áp dụng RoIAlign cải thiện khá nhiều độ chính xác cho mạng.



Hình 20: Mô tả phương pháp RoIAlign

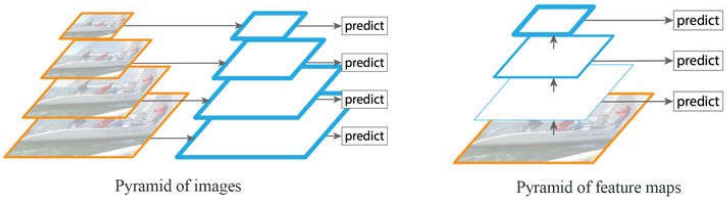
Kiến trúc Mask R-CNN được dùng trong các thí nghiệm

Mask R-CNN được chia làm 2 phần: phần đầu và phần trích xuất đặc trưng. Tác giả đề xuất 2 kiến trúc mạng cho Mask R-CNN dự vào cách chọn mạng cho phần trích chọn đặc trưng (Backbone) tương ứng với phần đầu mạng tương ứng như hình 21 dưới đây.



Hình 21: Kiến trúc backbone Mask R-CNN

Kiến trúc mạng bên trái phần backbone sử dụng mạng ResNet hoặc ResNeXt với số lớp convolution là 50 hoặc 101 lớp được đề xuất trong [20]. Đầu ra đặc trưng được trích dẫn ở tầng thứ 4 của mạng gọi là C4. Bên phải là áp dụng Mask RCNN với phần Backbone là FPN (Feature Pyramid Network) được nghiên cứu và đề xuất bởi Li và đồng nghiệp tại [21]. FPN là một bộ trích chọn đặc trưng được thiết kế có cấu trúc kim tự tháp cho tốc độ tính toán và độ chính xác cao với đối tượng ở nhiều kích cỡ khác nhau.



Hình 22: Mạng kim tự tháp

Kiến trúc đầu mạng được mở rộng từ Faster R-CNN với phần thêm vào là mạng tích chập đầy đủ cho dự đoán phân vùng đối tượng. Phần đầu với mạng ResNet-C4 được thêm vào tầng convolution thứ 5 – res5 (mất thời gian tính toán), với FPN backbone kiến trúc đã bao gồm res5 vì vậy mà mạng ít bộ lọc và hiệu quả hơn.

Trên đây em đã trình bày kiến trúc chung của Mask R-CNN tham khảo từ bài báo chính của tác giả Kaiming He tại [1]. Trong nội dung đồ án em sử dụng Mask R-CNN với kiến trúc Backbone là FPN như hình 22 bên phải cho việc phát hiện và phân vùng vùng bàn tay.

2.3. Thuật toán Mean shift theo vết đối tượng

Như đã nói tại chương 1 việc sử dụng Mask R-CNN còn có nhiều trường hợp sai sót, thiếu vì vậy để tăng lên độ chính xác cho hệ thống em sử dụng kết hợp thêm với thuật toán theo vết đối tượng để loại bỏ các trường hợp sai khác của Mask R-CNN. Đầu vào và đầu ra của thuật toán theo vết như sau:

Đầu vào: Frame đầu và vị trí đối tượng cần theo dõi
Đầu ra: Vị trí đối tượng trên các frame tiếp theo
Vị trí đối tượng được biểu diễn bởi một hộp bao bao quanh nó

Có nhiều thuật toán theo vết cho hiệu quả khá tốt như Cam-Shift, Mean-Shift, Kalman Filter,.. cho hiệu quả khá tốt trong bài toán này. Trong đồ án này em nghiên cứu và áp dụng thuật toán mean shift để kết hợp với Mask R-CNN trong việc phát hiện vùng đối tượng bàn tay trên ảnh. Sau đây em sẽ trình bày về thuật toán Mean Shift.

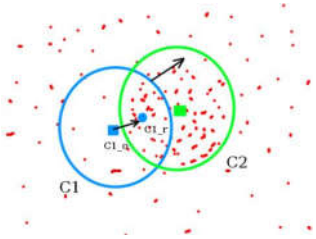
- Superpixels: là điểm ảnh lớn hiểu là một phân vùng của ảnh tức là nhóm một vùng pixel có cùng chung một đặc điểm nào đó.
- Gradient: của một ảnh cho biết ảnh được thay đổi như thế nào. Gradient của ảnh được sử dụng cho nhiều bài toán phân tích ảnh, trong đó có các phương pháp liên quan đến superpixels. Gradient của ảnh cung cấp 2 loại thông tin: Độ lớn (magnitude), cho biết hình ảnh đang thay đổi nhanh như thế nào, hướng gradient cho biết hướng mà ảnh

thay đổi nhiều nhất. Gradient có một hướng đi và có một độ lớn vì vậy mà mã hóa phù hợp nhất sử dụng cho gradient là dùng vector. Chiều dài vector cho ta biết độ lớn gradient và hướng vector cho ta biết hướng gradient.

Có nhiều các thuật toán tính toán các superpixels với các thông số khác nhau tuy nhiên dáng chú ý là các thuật toán dự vào gradient đi lên (Gradient Ascent-Based). Mean Shift là một thuật toán áp dụng gradient đi lên để tính ra superpixels.

Mean shift bắt đầu từ một nhóm điểm ảnh thô, phương pháp gradient đi lên được lặp cho đến khi đạt được một tiêu chuẩn hội tụ nào đó của superpixels. Trong [2], mô tả mean shift là một quá trình lặp lại tìm cực đại địa phương được áp dụng trong tìm trạng thái trong không gian màu và hoặc trong không gian cường độ hình ảnh. Các điểm ảnh lân cận có cùng một trạng thái được nhóm lại và tạo nên superpixel.

Một cách đơn giản ta có thể hiểu là ta chạy một vòng lặp tìm trọng tâm phân vùng (có thể là phân bố pixels), update trọng tâm đó cho đến khi đạt được một tiêu chuẩn hội tụ hội tụ theo một tiêu chuẩn nào đó ta thu được vùng mới có vị trí chính xác của vật thể quan tâm. Một ví dụ ta cho một phân vùng nhỏ giả sử một vòng tròn và thuật toán sẽ làm là dịch chuyển vòng tròn đó đến một vị trí mới có mật độ điểm ảnh tối đa như hình 23:



Hình 23: Ví dụ thuật toán Mean shift (opencv docs)

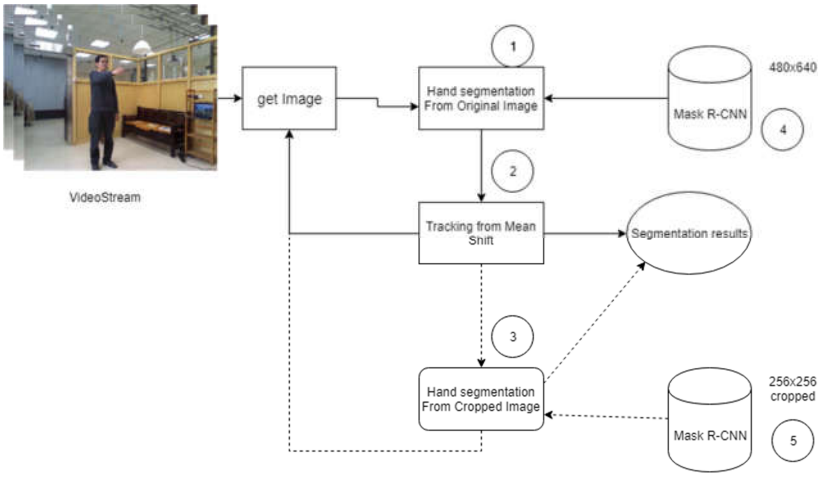
Ban đầu cửa sổ được cho bởi vòng tròn C1, tâm vòng tròn thực sự ở vị trí hình vuông mà xanh, tuy nhiên khi ta tìm trọng tâm của các điểm ảnh màu đỏ bên trong hình thì ta tính toán được trọng tâm nằm tại điểm vòng tròn c1_r đó là điểm trọng tâm thực sự của phân vùng, từ đây ta dịch chuyển phân vùng tới điểm tâm mới là trọng tâm của nó và lặp lại quá trình tìm trọng tâm cho đến khi hội tụ. Mục đích chính của việc sử dụng thuật toán Mean shift trong đồ án này của em là tăng độ chính xác cho việc phát hiện đối tượng tay, loại bỏ các kết quả phát hiện sai do nhiễu môi trường.

Việc cài đặt và sử dụng thuật toán mean shift này em sử dụng FrameWork Opencv. Do tính tiện lợi và phù hợp với ngôn ngữ lập trình em chọn sử dụng trong đồ án này là Python.

CHƯƠNG 3: TRIỂN KHAI MÔ ĐUN PHÁT HIỆN VÀ PHÂN ĐOẠN BÀN TAY

3.1. Mô hình đề xuất nghiên cứu

Mô hình đề xuất của em là sử dụng Mask R-CNN với kiến trúc backbone là FPN và kết hợp với thuật toán tracking đối tương Mean-Shift để phát hiện vùng bàn tay trên ảnh. Sở đồ khối hệ thống như sau:



Hình 24: Sơ đồ thuật toán mô hình đề xuất

Giải thích các khối:

1- Phát hiện đối tượng bàn tay bằng mô hình (4)
2- Khởi tạo cửa sổ theo vết dựa vào kết quả của Mask R-CNN hoặc từ ảnh trước đó đã phát hiện được đối tượng
3- Mở rộng cửa sổ theo vết và phát hiện bằng mô hình (5)
4- Mô hình Mask R-CNN được huấn luyện lại từ bộ trọng số tập dữ liệu COCO với đối tượng bàn tay trên ảnh kích cỡ gốc (480x640)
5- Mô hình Mask R-CNN được huấn luyện lại từ bộ trọng số tập dữ liệu COCO với đối tượng bàn tay trên ảnh kích cỡ resize (256x256)

Giải thích mô hình: Với những công cụ được trình bày tại các phần trên em xin đề xuất một mô hình kết hợp giữa Mask R-CNN và thuật toán theo vết đối tượng Mean shift để giải quyết bài toán phát hiện và phân vùng đối tượng bàn tay trên ảnh:

- Em chuẩn bị 2 mô hình Mask R-CNN đã được huấn luyện với đối tượng bàn tay: - Một mô hình được coi là mô hình phụ em huấn luyện bằng cách crop vùng bàn tay lại kích cỡ 256x256 trên tập huấn luyện và đưa vào mô hình huấn luyện trộn lẫn (3 Kinect) . Mô hình thứ 2 được coi là mô hình đánh giá chính theo từng thí nghiệm cụ thể. Trong mô hình này, các dữ liệu huấn luyện được thực hiện với độ phân giải gốc.
- Dữ liệu ảnh để kiểm thử hoàn toàn độc lập với tập huận luyện và được đọc vào một cách tuần tự để việc theo vết có thể hoạt động
- Thuật toán mean shift hoạt động được cần khởi tạo cửa số đối tượng. Nếu ảnh ban đầu chưa có thì có thể khởi tạo bằng cách lấy vào đối tượng đầu tiên có điểm số dự đoán được cao nhất và coi là đối tượng chính xác mà ta cần theo vết trong suốt video (hoặc chuỗi ảnh tuần tự). Việc khởi tạo dựa trên điểm số mạng dự đoán ra đôi khi sai vì vậy em áp dụng một ngưỡng để đảm bảo độ tin vậy cho kết quả (trong trường hợp này em sử dụng ngưỡng là 0.96). Trong quá trình chạy cửa sổ sẽ bị xóa đi khi mà đối tượng không còn được tìm thấy trên ảnh và ở frame tiếp theo em sẽ tiến hành khởi tạo lại.
- Do trong quá trình đánh giá kết quả phát hiện của riêng Mask R-CNN, một số trường hợp Mask R-CNN không phát hiện được đối tượng tuy nhiên nếu crop vùng đối tượng với kích cỡ 256x256 và cho vào mạng huấn luyện với ảnh này thì cho kết quả khá chính xác vì vậy em có sử dụng model crop cho việc phát hiện những frame thiếu này.
- Để đánh giá mức độ sai khác của cửa sổ theo vết và các ứng viên dự đoán bởi Mask R-CNN em so sánh 3 tiêu chí là khoảng cách giữa 2 tâm, và độ sai khác màu sắc giữa 2 vùng thông quá độ sai khác giữa 2 giá trị màu H và V trong không gian màu HSV và điểm số dự đoán của Mask R-CNN. Do việc sai khác về mặt đơn vị nên em chuẩn hóa chúng về dạng tỉ lệ [0, 1]:

$$D = d + \Delta H + \Delta S + Score$$

3.1.1. Quá trình huấn luyện Mask R-CNN

Để huấn luyện mạng Mask R-CNN em có tham gia vào quá trình chuẩn bị dữ liệu trên tập dữ liệu MICA với khoảng 3000 ảnh trên tổng số khoảng 15000 trên 3 góc nhìn thu từ Kinect ảnh với công cụ (Interactive Segmentation Tool) một ví dụ cho việc chuẩn bị dữ liệu:

Hình 25: Chuẩn bị dữ liệu. Hình bên trái là ảnh gốc. Hình bên phải là vùng bàn tay đã được tách khỏi nền.

Em sử dụng mô hình pre-train của tác giả trên tập dữ liệu COCO để khởi tạo các tham số mô hình cho mạng. Các bước huấn luyện như sau:

Bước 1: Chuẩn bị dữ liệu ảnh màu và ảnh nhị phân chứa vùng đối tượng trên ảnh màu tương ứng

Bước 2: Tải bộ trọng số Pre-train và cấu hình các tham số mạng

Bước 3: Tải dữ liệu lên và tiến hành huấn luyện

Bước 4: Lưu lại các checkpoint để huấn luyện lại nếu xảy ra lỗi

Từ việc chuẩn bị dữ liệu em có khoảng 15 000 ảnh

Quá trình huấn luyện mạng theo các bước như sau:

- Huấn luyện các lớp đầu với khoảng 40 000 bước lặp với giá trị learning rate 0.001
- Tiếp theo là 80 000 bước lặp trên toàn bộ mạng với giá trị learning rate 0.001
- 40 000 bước lặp tiếp theo huấn luyện trên toàn bộ mạng với learning rate 0.0001

Trong quá trình huấn luyện em có thử trên 2 kích thước với ảnh đầu vào là kích thước gốc 480x640 và kích cỡ 256x256 sau khi resize. Em thực hiện các thí nghiệm trên GPU GTX 1080 Ti với dung lượng Ram 12gb. Với số lượng ảnh lớn sẽ không thể huấn luyện được mạng do giới hạn bởi dung lượng Ram vì vậy em chọn cách thực trao đổi giữa kích cỡ ảnh huấn luyện và số lượng ảnh cho vào huấn luyện. Trong quá trình thực hiện đồ án em có thực hiện các thí nghiệm sau:

- Thí nghiệm 1: Huấn luyện Mask R-CNN với 9723 ảnh (dữ liệu của 5 người với 11 hành động khác nhau) tại 3 góc nhìn khác nhau được trộn lẫn. Đầu vào ảnh được resize về kích cỡ 256x256. Dữ liệu test: 1403 ảnh chia đều 3 góc nhìn.
- Thí nghiệm 2: Huấn luyện với số lượng ảnh như trên nhưng crop lại kích với kích thước 256x256 lan từ tâm vùng bàn tay.
- Thí nghiệm 3: Huấn luyện Mask R-CNN trên với khoảng 2200 ảnh (4 người với 5 hành động khác nhau trên từng góc nhìn K1, K3, K5) với độ phân giải gốc

(480x640). Dữ liệu từ 1 người 5 hành động độc lập tập huấn luyện sẽ được sử dụng cho việc test.

Thời gian huấn luyện cho mỗi thí nghiệm khoảng 30 tiếng trên GPU với các bước huấn luyện như nhau đã đề cập ở trên.

Giá trị hàm mất mất trong quá trình huấn luyện:

Như đã nói trong phần cơ bản mạng neuron quá trình huấn luyện thực chất là quá trình tối ưu hàm mất mát sao cho giá trị nó đi tới cực tiểu. Trong Mask R-CNN giá trị hàm loss được định nghĩa là tổng của 3 giá trị loss khác nhau:

L_{cls}: hàm mất mát trong dự đoán lớp

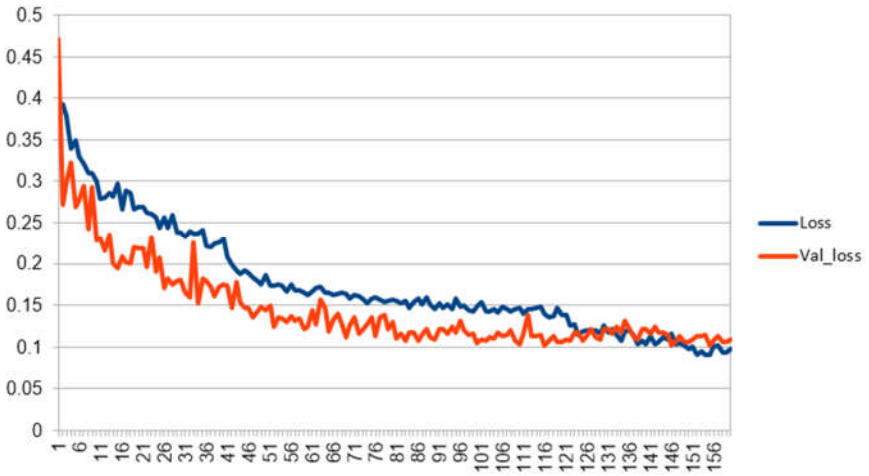
L_{bbox}: giá trị mất mát trong dự đoán bao đóng

L_{mask}: giá trị mất mát trong dự đoán phân vùng

$Loss = L_{cls} + L_{bbox} + L_{mask}$

Trong quá trình huấn luyện em thấy giá trinh hàm Loss hội tụ sau 160.000 bước lặp. Minh họa dưới đây cho biểu thị hàm loss trong quá trình huấn luyện mạng Mask R-CNN với thí nghiệm huấn luyện ảnh resize về kích cỡ 256x256 sẽ được em trình bày sau đây. Các thí nghiệm khác đồ thị tương tự vì vậy em không report thêm.

Commented [T1]: Mạng nào ? có nhiều thí nghiệm



Hình 26: Giá trị hàm mất mát trong quá trình huấn luyện

3.1.2. Cài đặt module và huấn luyện Mask R-CNN

Ngôn ngữ lập trình dùng cho việc thao tác dữ liệu: Python là một ngôn ngữ lập trình scripting phổ biến và rất thú vị. Hơn nữa nó rất hiệu quả trong việc xử lý dữ liệu, dễ học, dễ làm và dễ cài đặt. Python đang là ngôn ngữ phổ biến nhất hiện tại.

Công cụ xử lí ảnh và Video em sử dụng là Opencv do việc xử dụng khá đơn giản và Opencv được hỗ trợ trên python trên nền numpy

Cấu hình máy tính sử dụng để chạy thử nghiệm:

12 core CPU 2.5GHz, GPU GeForce GTX 1080 Ti, 12GB VRam

Môi trường lập trình là python vì vậy cần cài đặt python và các package cần thiết là: tensorflow, keras, opencv2, numpy

Để sử dụng Mask R-CNN ta có 2 lựa chọn framework : Tensorflow và Caffe tại: http://caffe.berkeleyvision.org và https://www.tensorflow.org/ .

Source code của Mask R-CNN được public tại: https://github.com/matterport/Mask_RCNN

Hoặc với Caffe tại : https://github.com/facebookresearch/Detectron

Trong đồ án em sử dụng framework Tensorflow.

Sau khi cài đặt các gói python cần thiết và framework ta tiến hành clone source từ github bằng lệnh:

```
git clone https://github.com/matterport/Mask_RCNN
```

Tải bộ trọng số mô hình pre-train tại:

```
https://github.com/matterport/Mask_RCNN/releases
```

Chuẩn bị dữ liệu ảnh màu và nhị phân load toàn bộ lên mạng và tiến hành train.

Quá trình test làm tương tự với train thay đổi mode mô hình từ training -> inference

Thuật toán Tracking MeanShift ta sử dụng thư viện OpenCV3 để chạy.

3.2. Kết quả thí nghiệm

Các thí nghiệm của em có 4 kết quả chính gồm

- Huấn luyện trộn lẫn các góc nhìn ảnh resize kích cỡ 256x256, kiểm thử trộn lẫn trên ảnh gốc và ảnh resize kích cỡ 256x256.

- Huấn luyện trộn lẫn các góc nhìn với ảnh crop kích cỡ 256x256 và test trên ảnh crop kích cỡ 256x256 và ảnh gốc.

- Huấn luyện trên tập dữ liệu nhỏ hơn với kích cỡ ảnh gốc với từng góc nhìn và kiểm thử trên tất cả các góc nhìn.

- Kết hợp thuật toán theo vết đối tượng để tăng độ chính xác đối với mô hình.

Trong các thí nghiệm này, AP là tỷ lệ giữa vùng giao và vùng hợp tạo bởi vùng phát hiện được với groundtruth.

Thời gian tính toán:

Đối với các thì nghiệm dưới đây thời gian thử nghiệm trên mỗi ảnh đối với GPU GTX 1080Ti Vram 12Gb:

- Thử nghiệm bộ trọng số được huấn luyện từ Mask R-CNN tốc độ đạt khoảng 5fps.
- Thử nghiệm kết hợp thuật toán theo vết theo mô hình đề xuất tốc độ khoảng 3fp.

3.2.1. Kết quả huấn luyện với ảnh resize kích cỡ 256x256

Như đề cập ở trên phần này em huấn luyện với số lượng ảnh là 11339 ảnh hỗn hợp trên 3 Kinect 1, 3, 5 và test trên tập dữ liệu 1599 ảnh cũng thuộc 3 góc nhìn 1, 3, 5 có số lượng gần tương đương nhau.

Có 2 kết quả là test trên ảnh đã resize về kích cỡ 256x256 và trên ảnh gốc :

Kết quả thí nghiệm :

Bảng 1: Kết quả thử nghiệm trên ảnh gốc

	K1 (458)	K3 (458)	K5 (487)	K1 + K3 + K5 (1403)
mAP	0.075	0.292	0.201	0.192
AP(0.5)	0.261	**0.826**	0.725	**0.610**
AP(0.6)	0.138	0.664	0.382	0.402
AP(0.7)	0.046	0.181	0.076	0.103

Bảng 2: Kết quả thử nghiệm với ảnh resize (256x256)

	K1 (458)	K3 (458)	K5 (487)	K1 + K3 + K5 (1403)
mAP	0.082	0.362	0.252	0.232
AP(0.5)	0.276	**0.852**	0.742	0.623
AP(0.6)	0.161	0.727	0.532	0.473
AP(0.7)	0.036	0.401	0.146	0.194

Nhận xét: Từ kết quả trên ta thấy được độ chính xác trên trên góc nhìn thứ 1 là rất thấp , góc nhìn thứ 3 và thứ 5 có độ chính xác tại giá trị AP=0.5 là chấp nhận được. Ta thu được kết quả như vậy là do tại góc nhìn thứ nhất bàn tay bị lẫn nhiều bởi màu nền, thêm nữa cử chỉ khá nhanh làm cho hình ảnh bị hiệu ứng mờ (motion blur) làm hệ thống không nhận dạng được.

Một nguyên nhân nữa là do ta đã resize ảnh huấn luyện đầu vào về kích thước 256x256 làm cho thông tin bị mất mát. Thí nghiệm này là huấn luyện trên tất cả các góc nhìn, sau 160 bước giá trị hàm loss vẫn khá cao cũng là nguyên nhân gây nên kết quả kém.

Với Mask R-CNN kết quả thử nghiệm trên cho thấy việc resize ảnh trong lúc thử nghiệm không gây ra nhiều sai khác do kiến trúc có sử dụng mạng FPN. Mặc dù vậy kết quả khi thử nghiệm với kích thước ảnh đã resize về kích cỡ 256x256 bằng kích cỡ ảnh huấn luyện vẫn cho kết quả cao hơn khoảng 4%

Hình 27: Một số trường hợp nhận dạng sai của mạng.

Trên hình là một số trường hợp nhận dạng sai của mạng được huấn luyện với kích cỡ ảnh crop 256x256. Màu vàng thể hiện ground truth màu còn lại thể hiện các đối bàn tay phát hiện được của mạng. Nguyên nhân là do bàn tay bị lẫn bởi mặt, nhận dạng nhầm tay còn lại và bàn tay di chuyển nhanh gây nên hiệu ứng mờ (motion blur)

3.2.2. Kết quả huấn luyện với ảnh crop kích cỡ 256x256

Em tiến hành với tập dữ liệu như trên và kết quả em đánh giá trên ảnh crop kích cớ 256x256 và ảnh gốc. Trong thí nghiệm này em huấn luyện ảnh đã crop kích cỡ 256x256 với tâm là trọng tâm của vùng bàn tay đã được phân vùng thủ công. Trong trường hợp này em chỉ báo cáo kết quả thử nghiệm với tập ảnh thuộc cả 3 góc nhìn với 2 trường hợp là ảnh thử nghiệm cũng được crop kích thước 256x256 và ảnh thử nghiệm giữ nguyên kích thước gốc.

Bảng 3:Kết quả với ảnh crop kích thước 256x256

	Kích thước gốc	Crop kích thước 256x256
mAP	0.04	0.756
AP(0.5)	0.18	**0.993**
AP(0.6)	0.08	0.986
AP(0.7)	0.018	0.940

Dễ nhận thấy ảnh kích cỡ ban đầu 480x640 so với ảnh đã crop thông tin bị mất đi nhiều so với ảnh gốc vì vậy mà mô hình cho kết quả rất tệ với ảnh kích cỡ gốc. Tuy nhiên với ảnh được crop lại kích cỡ 256x256 chứa vùng bàn tay thì hầu như là không bị nhận nhầm. Độ chính xác có thể đạt tới 99%.

3.2.3. Kết quả huấn luyện trên ảnh gốc trên từng góc nhìn

Với ảnh gốc em huấn luyện trên từng góc nhìn riêng lẻ với số lượng ảnh tương đối nhỏ so với 2 thí nghiệm trên khoảng 2000 ảnh và kiểm thử trên khoảng 400 ảnh. Kết quả như sau:

Bảng 4: Kết quả huấn luyện dữ liệu trên Kinect 1

	K1 (458)	K3(458)	K5(487)	Mean Shift
mAP	**0.335**	0.495	0.432	0.42177584
AP(0.5)	**0.701**	0.862	0.800	**0.78883552**
AP(0.6)	**0.584**	0.819	0.734	0.72369723
AP(0.7)	**0.421**	0.647	0.549	0.51919942
Overlap	0.473609			**0.49123127**

Bảng 5: Kết quả huấn luyện dữ liệu trên Kinect 3

	K1 (458)	**K3(458)**	K5(487)	Mean Shift
mAP	0.180	**0.5**	0.586	0.542
AP(0.5)	0.470	**0.920**	0.971	0.970
AP(0.6)	0.352	**0.889**	0.949	0.942
AP(0.7)	0.182	**0.696**	0.784	0.732
Overlap		**0.68871723**		0.71332992

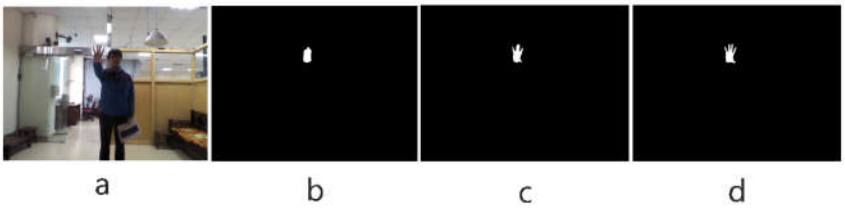
Bảng 6: Kết quả huấn luyện dữ liệu trên Kinect 5

	K1 (458)	K3(458)	**K5(487)**	Mean Shift
mAP	0.182	0.586	**0.650**	0.649
AP(0.5)	0.443	0.950	**0.981**	0.982
AP(0.6)	0.359	0.936	**0.973**	0.974
AP(0.7)	0.213	0.805	**0.901**	0.892
Overlap			**0.745**	**0.789**

Nhận xét: Từ bảng kết quả trên ta thấy việc sử dụng Mask R-CNN cho kết quả tốt nhất khi thực hiện huận luyện trên góc nhìn nào và thử nghiệm trên góc nhìn đó. Trên góc nhìn thứ nhất khi sử dụng huấn luyện một tập dữ liệu chung cho cả 3 góc nhìn cho kết quả khá kém tuy nhiên khi ta kiểm nghiệm với tập dữ liệu chỉ thuộc góc nhìn này thì cho kết quả cao hơn khá nhiều tuy nhiên vẫn còn khá hạn chế so với 2 góc nhìn còn lại.

Mask R-CNN cho độ chính xác rất tốt với đối tượng bàn tay ở góc nhìn trực diện (Kinect 5) lên đến 98%.

So sánh kết quả giữa mô hình huấn luyện với ảnh resize kích cỡ 256x256 ở trên và mô hình huấn luyện với ảnh gốc:



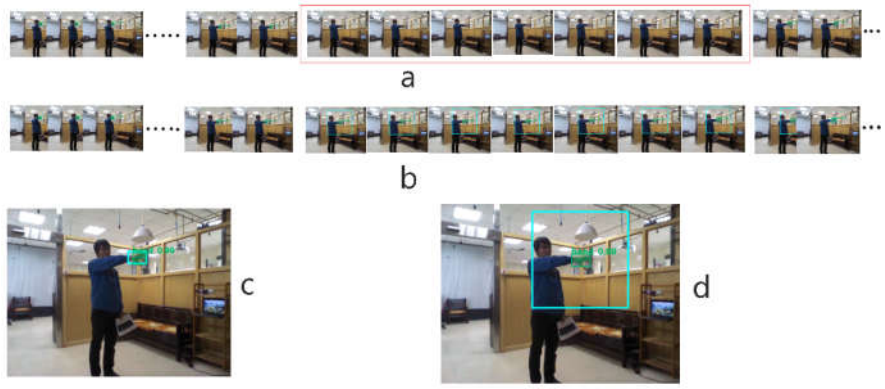
Hình 28: So sánh kết quả phân vùng của mô hình ảnh resize và ảnh gốc

Hình 28a là ảnh gốc, 28b là kết quả của mô hình huấn luyện với ảnh đã resize, 28c là kết quả với mô hình huấn luyện ảnh gốc góc nhìn 5. Em nhận thấy với ảnh gốc kích cỡ 480x640 huấn luyện trên góc nhìn thứ 5 gần với ground truth hơn so với mô hình ảnh đã resize. Ảnh đã resize thường bị thiếu các ngón tay. Điều này cho thấy mô hình huấn luyện trên ảnh kích thước gốc là hiệu quả hơn so với mô hình huấn luyện trên kích cỡ ảnh resize về độ chính xác phân vùng.

3.2.4. Kết quả có áp dụng thuật toán tracking

Ở phần 3.2.3. em có thêm vào kết quả thử nghiệm khi áp dụng thuật toán theo vết đối tượng Mean Shift và dễ thấy kết quả được nâng lên so với kết quả bình thường. Như vậy cho thấy mô hình đề xuất là hiệu quả đối với bài toán phát hiện và phân vùng đối tượng.

Bằng cách sử dụng kết hợp với thuật toán theo vết đối tượng em đã giải quyết được các trường hợp không phát hiện được, phát hiện sai của mô hình. Như ở một số trường hợp hệ thống nhận diện nhầm vùng mặt vào đối tượng bàn tay, hệ thống nhận diện thiếu khi bàn tay ở vị trí đưa sang phải, sang trái.



Hình 29: Thuật toán Mean Shift áp dụng tăng độ chính xác cho Mask R-CNN

Trên hình 29a và 29b là kết quả phát hiện trên video liên tiếp của một người thực hiện hành động đưa tay sang trái được ghi lại bởi góc nhìn thứ 3 với mô hình Mask R-CNN huấn luyện với ảnh kích thước gốc (29a) và mô hình đề xuất kết hợp thuật toán theo vết (29b). Hình 29c là mô tả cửa sổ theo vết đối tượng bàn tay (hình chữ nhật màu xanh dương). Hình 29d thể hiện trường hợp bàn tay không được phát hiện bởi Mask R-CNN, tuy nhiên thuật toán theo vết vẫn tìm được vị trí đối tượng bàn tay, em tiến hành mở rộng ra kích cỡ 256x256 (hình vuông màu xanh dương) và sử dụng mô hình huấn luyện với ảnh Crop (mô hình cho kết quả rất tốt với ảnh đối tượng được crop lại kích cỡ 256x256) phân vùng bàn tay đã được phát hiện một cách chính xác, nhược điểm đó là ta phải thực hiện 2 lần chạy thuật toán phát hiện cho một ảnh làm tăng thời gian chạy hệ thống.

Giá trị Overlap như trên bảng laf giá trị trung bình mức độ phủ của vùng bàn tay phân vùng và Ground truth. Em nhận thấy với mô hình đề xuất độ phủ trung bình vùng bàn tay tăng lên so với thí nghiệm chỉ sử dụng Mask R-CNN.

Kết quả cho thấy độ chính xác tăng lên nhờ việc áp dụng thuật toán theo vết đồng thời loại bỏ được hầu hết các ứng viên không chính xác.

Thêm nữa đầu ra của mô hình đề xuất chỉ có nhiều nhất một đối tượng bàn tay trên mỗi ảnh do yêu cầu bài toán đặt ra là nhận dạng duy nhất bàn tay đang thực hiện cử chỉ. Vì vậy nên mô hình này vẫn còn hạn chế khi nhận diện đối tượng ở góc nhìn thứ nhất do kết quả thường xuyên có sự xuất hiện của bàn tay còn lại không thực hiện hành động ảnh hưởng nhiều đến kết quả phát hiện cũng như theo vết.

CHƯƠNG 4: KẾT LUẬN

4.1. Kết quả đạt được

Trong quá trình thực hiện ĐATN, em đã tìm hiểu và làm chủ được một số kỹ thuật trong lĩnh vực thị giác máy tính và học máy, cụ thể là mạng nơ ron tích chập. Em đã hiểu về quy trình triển khai của một mô hình mạng nơ ron trên máy chủ. Bên cạnh đó, em đã tìm hiểu thêm về một lĩnh vực mới là Thị Giác Máy Tính cụ thể là lý thuyết cơ bản và một thuật toán để tính toán luồng quang học. Ngoài ra, em đã học được thêm một ngôn ngữ lập trình mới là python và sử dụng để thao tác với dữ liệu và sử dụng các thuật toán trong thư viện opencv, numpy. Kết quả chạy tương đối tốt so với yêu cầu phát hiện vùng đối tượng bàn tay trên ảnh

4.2. Những điểm còn hạn chế

- Thời gian chạy chương trình chậm chưa thể đạt đến thời gian thực. Với mô hình đề xuất em chưa nghiên cứu tối ưu thời gian chạy.
- Kết quả đạt được trên góc nhìn thứ nhất còn chưa tốt.
- Một số trường hợp khi áp dụng thuật toán theo vết đối tượng bàn tay bị loại bỏ nhầm ở góc nhìn thứ 1.

4.3. Hướng phát triển

Như đã nói trong chương 1 bài toán phát hiện và phân vùng đối tượng bàn tay đóng vai trò khá quan trọng vấn đề thì giác máy tính do vậy mà hướng phát triển cho bài toán khá cũng có rất nhiều:

- Cải thiện độ chĩnh xác cho phát hiện và phân vùng đối tượng trên các góc nhìn khó như góc nhìn thứ 1, đánh giá thủ nghiệm trên nhiều các tập dữ liệu khác nhau
- Cải thiện tốc độ tính toán cho thuật toán nhằm ứng dụng cho các bài toán đòi hỏi thời gian thực
- Từ kết quả phân vùng đối tượng áp dụng cho các pha tiếp theo như nhận diện chuyển động, cử chỉ, …

TÀI LIỆU THAM KHẢO

1. He K., Gkioxari G., Dollár P., et al. (2017). Mask R-CNN. *ArXiv170306870 Cs*.
2. Carreira-Perpiñán M.Á. (2015). A review of mean-shift algorithms for clustering. *ArXiv150300687 Cs Stat*.
3. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques - IEEE Journals & Magazine. <https://ieeexplore.ieee.org/document/5983442>, accessed: 05/20/2018.
4. Oualla M., Sadiq A., and Mbarki S. (2014). A survey of Haar-Like feature representation. *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 1101–1106, 1101–1106.
5. Chouvatut V., Yotsombat C., Sriwichai R., et al. (2015). Multi-view hand detection applying viola-jones framework using SAMME AdaBoost. *2015 7th International Conference on Knowledge and Smart Technology (KST)*, 30–35, 30–35.
6. Object detection based on HOG features: Faces and dual-eyes augmented reality - IEEE Conference Publication. <https://ieeexplore.ieee.org/document/6618716/>, accessed: 05/20/2018.
7. Divvala S.K., Efros A.A., and Hebert M. (2012). How important are Deformable Parts in the Deformable Parts Model?. *ArXiv12063714 Cs*.
8. Hand detection using multiple proposals. <http://www.robots.ox.ac.uk/~vgg/research/hands/>, accessed: 05/20/2018.
9. Backpropagation Applied to Handwritten Zip Code Recognition - MITP Journals & Magazine. <https://ieeexplore.ieee.org/document/6795724/>, accessed: 05/26/2018.
10. Girshick R., Donahue J., Darrell T., et al. (2016). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans Pattern Anal Mach Intell*, **38**(**1**), 142–158.
11. Segmentation as selective search for object recognition - IEEE Conference Publication. accessed: 05/20/2018.
12. Yan S., Xia Y., Smith J.S., et al. (2017). Multiscale Convolutional Neural Networks for Hand Detection. *Appl Comput Intell Soft Comput*, **2017**, 1–13.
13. Le T.H.N., Quach K.G., Zhu C., et al. (2017). Robust Hand Detection and Classification in Vehicles and in the Wild. IEEE, 1203–1210, 1203–1210.
14. Dai J., He K., Li Y., et al. (2016). Instance-sensitive Fully Convolutional Networks. *ArXiv160308678 Cs*.

15. Roy K., Mohanty A., and Sahay R.R. (2017). Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation. IEEE, 640–649, 640–649.
16. Krizhevsky A., Sutskever I., and Hinton G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105.
17. Girshick R., Donahue J., Darrell T., et al. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *ArXiv13112524 Cs*.
18. He K., Zhang X., Ren S., et al. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *ArXiv14064729 Cs*, **8691**, 346–361.
19. Ren S., He K., Girshick R., et al. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv150601497 Cs*.
20. He K., Zhang X., Ren S., et al. (2015). Deep Residual Learning for Image Recognition. *ArXiv151203385 Cs*.
21. Lin T.-Y., Dollár P., Girshick R., et al. (2016). Feature Pyramid Networks for Object Detection. *ArXiv161203144 Cs*.

2. Carreira-Perpiñán M.Á. (2015). A review of mean-shift algorithms for clustering. *ArXiv150300687 Cs Stat*.
3. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques - IEEE Journals & Magazine. <https://ieeexplore.ieee.org/document/5983442>, accessed: 05/20/2018.
4. Oualla M., Sadiq A., and Mbarki S. (2014). A survey of Haar-Like feature representation. *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 1101–1106, 1101–1106.
5. Chouvatut V., Yotsombat C., Sriwichai R., et al. (2015). Multi-view hand detection applying viola-jones framework using SAMME AdaBoost. *2015 7th International Conference on Knowledge and Smart Technology (KST)*, 30–35, 30–35.
6. Object detection based on HOG features: Faces and dual-eyes augmented reality - IEEE Conference Publication. <https://ieeexplore.ieee.org/document/6618716/>, accessed: 05/20/2018.
7. Divvala S.K., Efros A.A., and Hebert M. (2012). How important are Deformable Parts in the Deformable Parts Model?. *ArXiv12063714 Cs*.

8. Hand detection using multiple proposals. <http://www.robots.ox.ac.uk/~vgg/research/hands/>, accessed: 05/20/2018.
9. Backpropagation Applied to Handwritten Zip Code Recognition - MITP Journals & Magazine. <https://ieeexplore.ieee.org/document/6795724/>, accessed: 05/26/2018.
10. Girshick R., Donahue J., Darrell T., et al. (2016). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans Pattern Anal Mach Intell*, **38**(**1**), 142–158.
11. Segmentation as selective search for object recognition - IEEE Conference Publication. accessed: 05/20/2018.
12. Yan S., Xia Y., Smith J.S., et al. (2017). Multiscale Convolutional Neural Networks for Hand Detection. *Appl Comput Intell Soft Comput*, **2017**, 1–13.
13. Le T.H.N., Quach K.G., Zhu C., et al. (2017). Robust Hand Detection and Classification in Vehicles and in the Wild. IEEE, 1203–1210, 1203–1210.
14. Dai J., He K., Li Y., et al. (2016). Instance-sensitive Fully Convolutional Networks. *ArXiv160308678 Cs*.
15. Roy K., Mohanty A., and Sahay R.R. (2017). Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation. IEEE, 640–649, 640–649.
16. Krizhevsky A., Sutskever I., and Hinton G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105.
17. Girshick R., Donahue J., Darrell T., et al. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *ArXiv13112524 Cs*.
18. He K., Zhang X., Ren S., et al. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *ArXiv14064729 Cs*, **8691**, 346–361.
19. Ren S., He K., Girshick R., et al. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv150601497 Cs*.
20. He K., Zhang X., Ren S., et al. (2015). Deep Residual Learning for Image Recognition. *ArXiv151203385 Cs*.
21. Lin T.-Y., Dollár P., Girshick R., et al. (2016). Feature Pyramid Networks for Object Detection. *ArXiv161203144 Cs*.