Hand Detection For Grab-and-Go Groceries

Phạm Thế Hiển and Phạm Tài Đức Phú Đa

Ngày 24 tháng 12 năm 2022

Tóm tắt nội dung

Hệ thống nhận diện bàn tay là một thành phần rất quan trọng trong việc thực hiện các hoạt động mua bán tạp hóa hoàn toàn tự động. Trong dự án này, chúng tôi đề xuất triển khai mô hình nhận diện tay trong thời gian thực bằng cách sử dụngYou only look once(YOLO) network. Mô hình mà chúng tôi đào tạo trên tập dữ liệu có thể đạt được mAP thử nghiệm là 0.975.

1 Introduction

Những tiến bộ gần đây trong thị giác máy tính đã biến các cửa hàng tạp hóa mua mang về như Amazon Go trở thành hiện thực.

Người mua sắm có thể chỉ cần đi bộ trong cửa hàng, lấy các món đồ và đi ra ngoài mà không phải xếp hàng dài chờ thanh toán.

Việc tự động hóa quy trình thanh toán chủ yếu dựa vào hệ thống thị giác máy tính có khả năng theo dõi các mặt hàng được một khách hàng nhất định lấy

Nhận diện bàn tay là một thành phần quan trọng trong một hệ thống ligent intel như vậy. Các hành vi của khách hàng trong quá trình mua sắm rất phức tạp và khó có thể đoán trước được. Ví dụ, người mua sắm thường lấy các mặt hàng từ kệ và sau đó đặt chúng trở lại. Để hiểu rõ hơn về hành vi của khách hàng, điều cực kỳ quan trọng là phát hiện và theo dõi bàn tay của họ trong các luồng video.

Để giải quyết những thách thức từ môi trường mua sắm phức tạp trong hiện thực, một máy dò tay lý tưởng không chỉ cung cấp dự đoán chính xác cao, dự đoán như vậy cũng phải được thực hiện trong thời gian ngắn hợp lý để đáp ứng yêu cầu thời gian thực. Bên cạnh yêu cầu về độ chính xác và tốc độ, cũng thực tế khi tính đến hiệu quả tính toán. Thuật toán phát hiện bàn tay phải nhẹ để ngay cả một hệ thống nhúng chi phí thấp cũng có thể hoàn thành nhiệm vụ phát hiện bằng cách áp dụng thuật toán.

2 Công việc liên quan

Nhiều phương pháp thị giác máy tính truyền thống đã được đề xuất để phát hiện bàn tay trong một hình ảnh. Phân tích tần số xuất hiện các bàn tay bằng cách phân loại các đặc điểm tần số trông giống như bàn tay. Một cách tiếp cận khác là sử dụng phân đoạn thông qua thông tin cấp pixel. Tuy nhiên, hầu hết các phương pháp này đều dựa vào việc phát hiện pixel tông màu da, điều này có thể không rõ ràng nếu khuôn mặt hoặc các vùng da khác cũng xuất hiện trong hình ảnh

Gần đây, việc áp dụng các mạng tích chập đã cải thiện phần lớn hiệu suất của hình ảnh phân loại và phát hiện đối tượng. Phát hiện đối tượng nhiều hơn thách thức vì nó liên quan đến việc đề xuất các hộp giới hạn cho đối tượng tương ứng.

2.1 Region Proposal Based Detectors

Khu vực có CNN (R-CNN) là một phát hiện đối tượng mạng lưới dựa trên hệ thống đề xuất khu vực bên ngoài.

Mặc dù nhanh hơn và chính xác hơn đáng kể so với các phương pháp dựa trên các tính năng giống như HOG truyền thống,RCNN vẫn còn bị vấn đề hiệu suất tốc độ kế thừa từ hệ thống đề xuất khu vực bên ngoài. Theo đánh giá của chúng tôi, để đề xuất các vùng cho một hình ảnh duy nhất, tìm kiếm có chọn lọc mất 5 giây trên CPU 8 nhân. Fast R-CNN được sử dụng lại bản đồ đặc trưng từ

đầu ra tích chập theo vùng chiếu đề xuất cho lớp Roi Pooling. So với R-CNN, điều này cách tiếp cận đạt được hiệu suất nhanh hơn 25 lần bằng cách tránh tính toán lặp lại cho các bản đồ đối tượng địa lý. Tuy nhiên, Fast R-CNN vẫn phụ thuộc vào các đề xuất khu vực bên ngoài, là điểm nghẽn của tốc độ tàu và dự đoán. Mới nhất Faster R-CNN [12] đã loại bỏ đề xuất khu vực bên ngoài bởi chèn Mạng Đề xuất Khu vực sau khi tích hợp sâu các lớp. Cải tiến này đã giúp Faster RCNN đạt được 10 tốc độ nhanh hơn Fast RCNN

Tất cả các thiết bị phát hiện dựa trên khu vực được mô tả ở trên có chung một đặc điểm chung: họ có một phần trong mạng lưới của họ dành riêng để cung cấp các đề xuất khu vực, theo sau là phân loại chất lượng để phân loại các đề xuất này. Những phương pháp đó rất chính xác nhưng có chi phí là công việc ròng quá phức tạp và tính toán cao. Nói cách khác, chúng không phù hợp để sử dụng trong các hệ thống nhúng chi phí thấp.

3 Single Shot Detectors

Một loại trình phát hiện khác tấn công vấn đề phát hiện đối tượng từ một góc độ khác, hợp nhất các điểm phát hiện đối tượng riêng biệt thành một mạng nơ-ron duy nhất Các thuật toán này chỉ chụp một bức ảnh duy nhất và có thể đạt được tốc độ cao hơn so với các máy dò dựa trên đề xuất khu vực

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

Yolov3,Yolov5 và Yolov7 là phiên bản nâng cấp của YOLO, có thể phát hiện nhanh hơn và có thể đạt 0.97,4 mAP trên tập dữ liệu Hand dectection so với mô hình Yolo ban đầu . Máy dò MultiBox Single Shot (SSD) [9] đã xem xét các bản đồ tính năng khác nhau từ các lớp phức hợp để dự đoán các hộp giới hạn với các thang điểm khác nhau. SDD phù hợp nhất cho các mục đích sử dụng trong đó quy mô của các hộp giới hạn khác nhau từ một phạm vi lớn.

Bảng 1 tóm tắt độ chính xác và hiệu suất tốc độ của khung phát hiện mà chúng tôi đã mô tả. Như chúng ta có thể thấy, các máy dò ảnh đơn có xu hướng hoạt động tốt hơn về tốc độ và độ chính xác ac.

Detection Framework	mAP	FPS
R-CNN O-Net BB [6]	66.0	0.02
Fast R-CNN	70.0	0.5
Faster R-CNN VGG-16	73.2	7
YOLO	63.4	45
SSD300	74.3	46
YOLOv2 480×480	77.8	59

Table 1. Accuracy and Speed Comparison for different models [11]. Models are trained on PASCAL VOC 2007 + 2012 datasets.

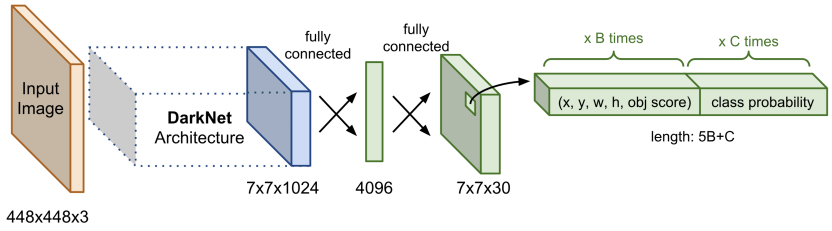
3.1 Phương Pháp

Để giải quyết bài toán phát hiện tay trong cửa hàng chúng tôi đã sử dụng đến mô hình Yolo lần lượt là các phiên bản nâng cấp của nó là Yolov3,Yolov5 và Yolov7

Chúng tôi sẽ mô tả nguyên tắc của mô hình YOLO và một số tối ưu hóa được thực hiện bởi YOLO9000 trong các phần sau.

3.2 Nguyên Lý hoạt động của Yolo

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp các convolutional layers



Hình 1: Giải thích Quy trình làm việc của YOLO.

sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.. Như Hình 1 cho thấy, đầu ra cuối cùng của mạng là tensor S × S × X, trong đó X phụ thuộc vào num ber của các lớp và phiên bản YOLO. Mỗi điểm trong tensor đầu ra cuối cùng nhìn vào ô trong hình ảnh gốc với cùng một vị trí không gian và dự đoán hộp giới hạn B. Đối với mỗi hộp giới hạn, YOLO không chỉ dự đoán x, y, chiều rộng và chiều cao, mà còn là điểm tin cậy đại diện cho IOU với hộp giới hạn chân lý cơ bản và xác suất của một đối tượng có mặt trong ô đó:

$$Confidence_{b-box} = Pr(Object) \times IOU_{pred}^{truth} \quad (1)$$

Mỗi ô cũng dự đoán xác suất lớp có điều kiện C, P r (Classi | Object). YOLO chỉ dự đoán một xác suất lớp đã đặt cho mỗi ô trong khi YOLO v2 dự đoán cho mỗi hộp giới hạn .Điểm tin cậy cho lớp có thể được đặt là:

$$\begin{aligned} Confidence_{class_i} &= Pr(Class_i|Object) \times Pr(Object) \\ &\quad \times IOU_{pred}^{truth} \\ &= Pr(Class_i) \times IOU_{pred}^{truth} \end{aligned} \quad (2)$$

Từ Công thức 2, chúng ta có thể thấy điểm số của lớp mã hóa cả xác suất xuất hiện của lớp đó trong hộp và mức độ phù hợp của hộp được dự đoán với đối tượng. Dự đoán cuối cùng sẽ được thực hiện bởi người dự đoán có IOU cao nhất với hộp giới hạn sự thật cơ bản. Trong quá trình đào tạo, YOLO op obj tối thiểu hóa hàm mất mát nhiều phần như Phương trình 3 cho thấy. 1 i obj biểu thị nếu một đối tượng xuất hiện trong ô i và 1 biểu thị rằng ij

Trình dự đoán hộp giới hạn thứ j trong ô thứ i chịu trách nhiệm cho dự đoán đó. 2 phần đầu tiên trong phương trình xử lý sự mất mát từ các hộp giới hạn phối hợp các dự đoán Phần thứ 3, 4 và 5 trong phương trình xử lý sự mất mát từ cation phân loại

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ &+\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ &+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\ &+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\ &+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (3)$$

3.3 Yolov3

Backbone YOLOv3 là một bản nâng cấp đáng giá của YOLOv2.Về cấu trúc,ở YOLOv1 thì sử dụng 24 lớp chập, sang YOLOv2 thì sử dụng backbone là darknet19 cộng với 11 lớp chập nữa để nhận dạng.Còn YOLOv3 xây dựng một backbone mới, gọi là Darknet-53.Backbone của YOLOv1 thì sử dụng 1×1 Convolution (gọi là Bottleneck) của Inception Network, lên YOLOv2 thì áp dụng thêm BatchNorm, sang YOLOv3 thì áp dụng thêm skip-connection từ ResNet, gọi là một Residual Block.

3.4 Yolov5

YOLOv5 không có quá nhiều thay đổi so với YOLOv4. YOLOv5 tập trung vào tốc độ và độ dễ sử dụng

1.BackBone : YOLOv5 cải tiến CSPResBlock của YOLOv4 thành một module mới, ít hơn một lớp Convolution gọi là C3 module. Chi tiết được thể hiện ở hình bên dưới.

2.Activation function: YOLOv4 sử dụng Mish hoặc LeakyReLU cho phiên bản nhẹ, còn sang YOLOv5, activation function được sử dụng là SiLU.

3.Loss function: Thêm hệ số scale cho Objectness Loss

4.Neck: YOLOv5 áp dụng một module giống với SPP, nhưng nhanh hơn gấp đôi và gọi đó là SPP - Fast (SPPF). Kiến trúc của SPPF được thể hiện ở Hình 12.

5.Anchor Box : Anchor Box trong YOLOv5 nhận được 2 sự thay đổi lớn. Đầu tiên là sử dụng Auto Anchor, một kĩ thuật áp dụng giải thuật di truyền (GA) vào Anchor Box ở sau bước k-means để Anchor Box hoạt động tốt hơn với những custom dataset của người dùng,

4 Dataset

4.1 Data EgoHands

4.2 Xử lý dữ liệu

Dùng ứng dụng hỗ trợ roboflow để dán label cho dữ liệu đầu vào gồm 1000 ảnh và ứng dụng hỗ trợ chia dữ liệu các các tập train, test thuận tiện cho quá trình chạy mô hình.

5 Huấn luyện và kết quả

Chúng tôi huấn luyện mô hình yolov3 và yolov5 với các thông số sau:

Model	Map	loss box	loss object
Yolov3	0.975	0.01096	0.02469
Yolov5	0.961	0.05477	0.03756

5.1 Kết quả

kết quả đạt được sau khi huấn luyện :

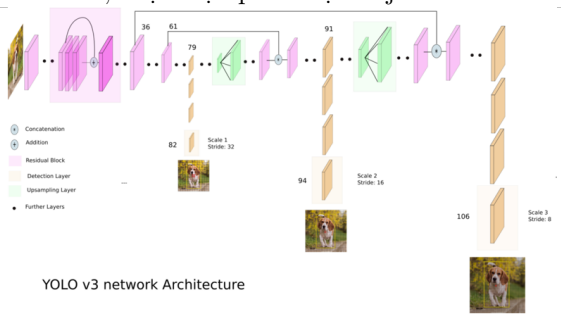
Model	Map	loss box	loss object
Yolov3	0.975	0.01096	0.02469
Yolov5	0.961	0.05477	0.03756

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

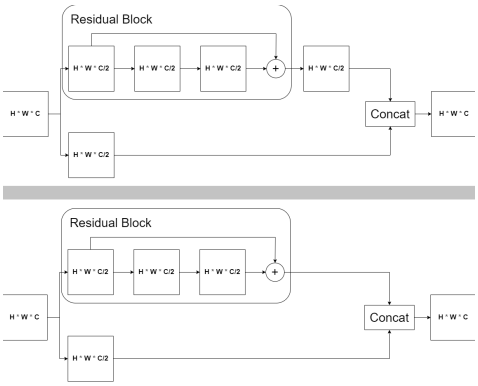
Bảng 2. Kiến trúc backbone của YOLOv3

Darknet-53 chủ yếu bao gồm các bộ lọc 3 x 3 và 1 x 1 với các skip connections giống như mạng còn lại trong ResNet.

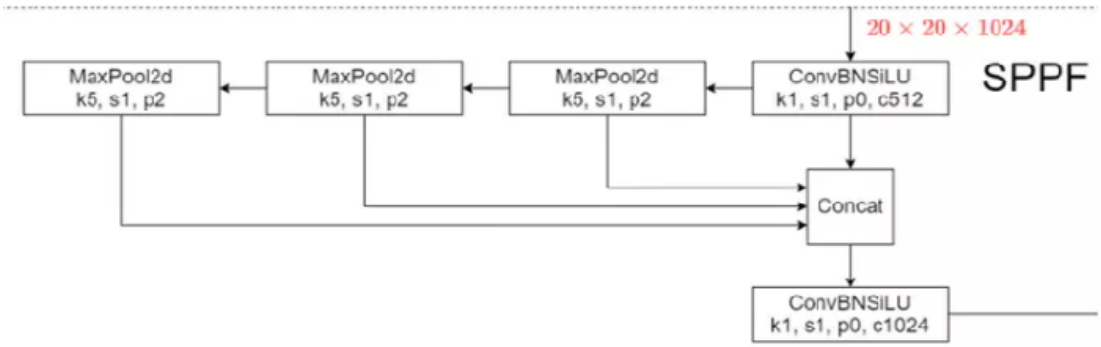
Neck Từ các phiên bản YOLO trước, phát hiện vật thể nhỏ luôn là một điểm yếu. Dù trong YOLOv2 đã sử dụng skip-connection từ layer trước đó để đưa thông tin từ feature map có kích thước lớn hơn vào feature map đằng sau, nhưng điều đó là không đủ. YOLOv3 là một sự nâng cấp cho vấn đề này, áp dụng Feature Pyramid Network, thực hiện phát hiện object ở 3 scale khác nhau của feature map



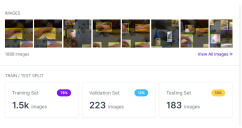
Hình 3. Kiến trúc của YOLOv3 với Feature Pyramid Network



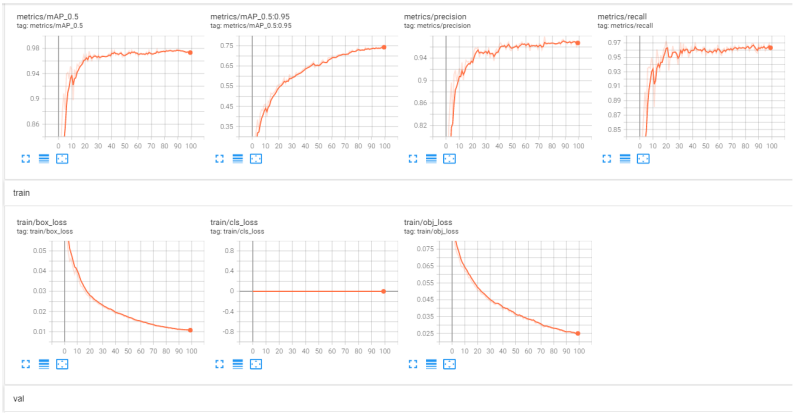
Hình 2: Sự khác biệt giữa CSPResBlock trong YOLOv4 (trên) và C3 Module (dưới)



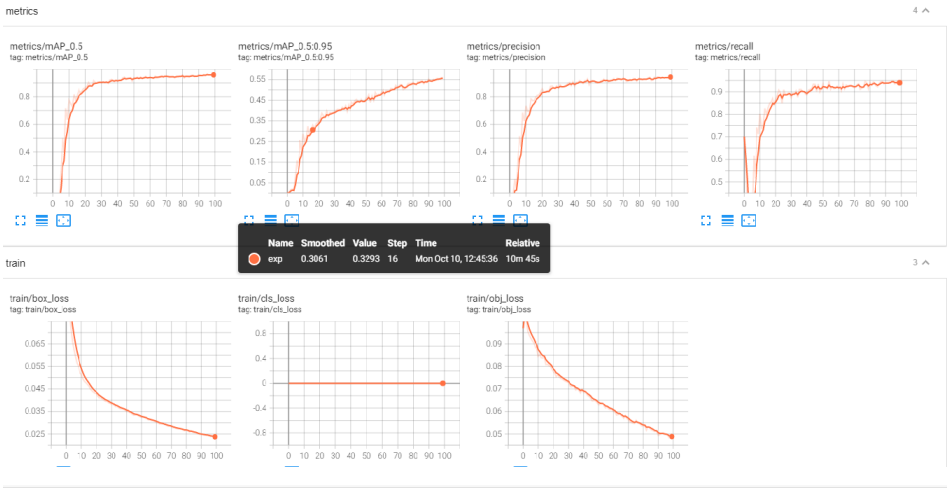
Hình 12. Kiến trúc của module SPPF



Hình 3: Dữ liệu sau khi xử lý



Hình 4: Kết quả quá trình huấn luyện yolov3



Hình 5: Kết quả quá trình huấn luyện yolov5

Hình 6: Ảnh sau khi dự đoán