NGHIÊN CỨU ỨNG DỤNG HỌC SÂU XÂY DỰNG BỘ NHẬN DẠNG VẬT THỂ GIÚP THANH TOÁN HÀNG HÓA NHANH

A STUDY ON APPLICATION OF DEEP LEARNING INTO BUILDING AN OBJECT DETECTOR TO SPEED UP RETAIL CHECKOUT

Nguyễn Trí Bằng[1*], Nguyễn Đình Vinh[1], Trần Trọng Đức[1]

[1]*Trường Đại học Bách khoa – Đại học Đà Nẵng*

[*]Tác giả liên hệ: ntbang@dut.udn.vn

(Nhận bài: 22/6/2021; Chấp nhận đăng: 09/8/2021)

Tóm tắt - Hiện nay, chưa có nhiều nghiên cứu về ứng dụng học sâu vào mảng nhận dạng thanh toán hàng hóa; Hầu hết chỉ nêu ra việc sử dụng YOLO để theo dõi số lượng vật phẩm thay đổi trên kệ hàng. Bài báo này trình bày giải pháp xây dựng bộ nhận dạng vật thể thời gian thực giúp thanh toán hàng hóa nhanh. Tác giả sử dụng YOLOv4, TResNet và FAISS lần lượt ở các giai đoạn phát hiện vật thể, trích xuất đặc trưng, phân loại hình ảnh đầu ra. Điều này giúp việc thêm dữ liệu mặt hàng mới mà không phải huấn luyện lại từ đầu so với giải pháp chỉ dùng YOLO. Bộ nhận dạng có một camera được lắp bên trên bàn thanh toán và màn hình hiển thị thông tin hóa đơn. Với kết quả thử nghiệm ban đầu, bộ nhận dạng có độ chính xác trung bình 94,54%. Thời gian thanh toán nhanh gấp đôi so với quét mã vạch. Ngoài ra, tác giả giới thiệu tập dữ liệu thanh toán hàng hóa BRC, góp phần cải thiện sự thiếu hụt dữ liệu trong cộng đồng nghiên cứu học sâu.

Từ khóa - Học sâu; YOLO; TResNet; FAISS; nhận dạng vật thể

Abstract - Currently, there have not been many studies on applying deep learning to the field of goods checkout detection; most of them just point out the solution of using YOLO to track the change of number of items on shelves. This paper presents a solution to build a real-time object detector to speed up retail checkout progress. The author uses YOLOv4, TResNet and FAISS respectively in the stages of object detection, feature extraction, and image classification. Which makes it possible to add new item data without having to completely retrain the model compared to a YOLO-only solution. The detector has a camera mounted above the checkout table and a monitor to display the invoice information. Initial experiment results show that our detector has an average accuracy of 94.54%. Payment time is twice as fast as barcode scanning. In addition, the author introduces the BRC, a dataset of retail checkout, which contributes to ameliorating the data shortage in the deep learning research community.

Key words - Deep learning; YOLO; TResNet; FAISS; object detector

1. Giới thiệu

1.1. Học sâu trong nhận dạng thanh toán hàng hóa và thách thức về mặt dữ liệu

Khi thanh toán hàng hoá với phương pháp quét mã vạch, nhân viên cần thời gian điều chỉnh máy quét và tìm kiếm vị trí in mã vạch vì chúng ở các vị trí khác nhau tùy sản phẩm. Bên cạnh đó, RFID cũng thường được áp dụng khi thanh toán hàng hóa nhưng vẫn có tỉ lệ lỗi do sóng radio bị nhiễu. RFID có chi phí cao, gây ra các vấn đề về phát triển bền vững [1]. Theo kết quả khảo sát của Jupiter Research [2], chi tiêu toàn cầu cho dịch vụ bán lẻ dựa vào trí tuệ nhân tạo tăng 300% từ 3,6 tỷ $ trong năm 2019 sang 12 tỷ $ trong năm 2023. Việc sử dụng các hệ thống tự động thanh toán hàng hoá bản lẻ tại siêu thị giúp giảm chi phí nhân công và mang lại trải nghiệm mua sắm tốt hơn [3]. Trong nghiên cứu [4], [5] chỉ ra, thời gian chờ đợi thanh toán ảnh hưởng tiêu cực đến mức độ hài lòng mua sắm của khách hàng. Vì vậy, việc ứng dụng trí tuệ nhân tạo giúp cải tiến các vấn đề trong lĩnh vực thanh toán hàng hóa là cần thiết, cần được nghiên cứu ứng dụng rộng rãi. Học sâu là một nhánh của học máy, sử dụng nhiều lớp xử lý với cấu trúc phức tạp. Trong thập kỷ qua, học sâu đã trở thành một kỹ thuật quan trọng để giải quyết các bài toán liên quan đến phát hiện vật thể và phân loại hình ảnh [6], [7].

Tuy nhiên, học sâu trong thị giác máy tính đang đối mặt với nhiều thách thức; một trong số đó là sự thiếu hụt dữ liệu. Tập dữ liệu có tầm quan trọng to lớn đối với sự hiệu quả của mô hình học sâu bởi nó yêu cầu một lượng lớn hình ảnh để huấn luyện. Điều này đặt ra một thách thức rất lớn trong bối cảnh chỉ có ít tập dữ liệu sẵn có [8]. Hiện có 2 tập dữ liệu về hình ảnh hàng hoá lúc thanh toán đã được công bố là D2S [9] và RPC [10], được tổng hợp ở Bảng 1.

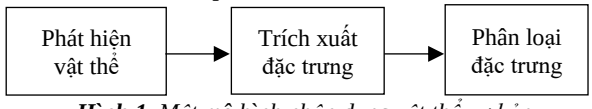
Bảng 1. *Một số thông tin về 2 tập dữ liệu D2S và RPC*

Tập dữ liệu	Tổng số hình ảnh	Số lượng chủng loại	Tập huấn luyện		Tập kiểm thử	
			Số ảnh	Vật phẩm/ hình	Số ảnh	Vật phẩm/ hình
D2S	21,000	60	4,380	1	16,620	>1
RPC	83,739	200	53,739	1	30,000	>1

Thực tế, cùng một vật phẩm nhất định nhưng dữ liệu hình ảnh thu được từ camera là khác nhau bởi góc chụp đến sản phẩm khác nhau qua mỗi lần thanh toán. Trong khi hình ảnh trong tập huấn luyện của D2S và RPC gồm các mặt hàng đơn lẻ, được xây dựng bởi các kỹ thuật cắt và xoay ảnh, khi ứng dụng thực tế sẽ gặp khó khăn.

1.2. Các nghiên cứu liên quan

Tương tự nhận dạng vật thể, bài toán nhận dạng hàng hóa bao gồm ba giai đoạn chính được trình bày ở Hình 1: (1) Phát hiện vật thể; (2) Trích xuất đặc trưng; (3) Phân loại vật thể đó thuộc sản phẩm nào [11], [7].



Hình 1. *Một mô hình nhận dạng vật thể cơ bản*

[1] The University of Danang – University of Science and Technology (Nguyen Tri Bang, Nguyen Dinh Vinh, Tran Trong Duc)

Hiện nay, có nhiều mô hình học sâu được sử dụng để tiếp cận giải quyết bài toán đặt ra. Thách thức của bài toán nhận dạng thanh toán là phải giải quyết được vấn đề cập nhật dữ liệu hàng hóa nhanh chóng khi mà chúng được phân phối về cửa hàng, thay đổi theo thời gian cả chủng loại lẫn mẫu mã. Với việc chỉ sử dụng YOLO cho cả ba giai đoạn ở Hình 1, khi thêm mới một mặt hàng vào cơ sở dữ liệu thì cần huấn luyện lại từ đầu, bởi YOLO có số lượng các lớp đầu ra là cố định. Vì vậy, bên cạnh dùng YOLO để phát hiện vật thể, cần kết hợp thêm các mô hình trích xuất và phân loại đặc trưng khác để phù hợp với bài toán. Trong mục này nhóm tác giả phân tích lựa chọn các kỹ thuật phù hợp cho mỗi giai đoạn.

1.2.1. Phát hiện vật thể với YOLOv4

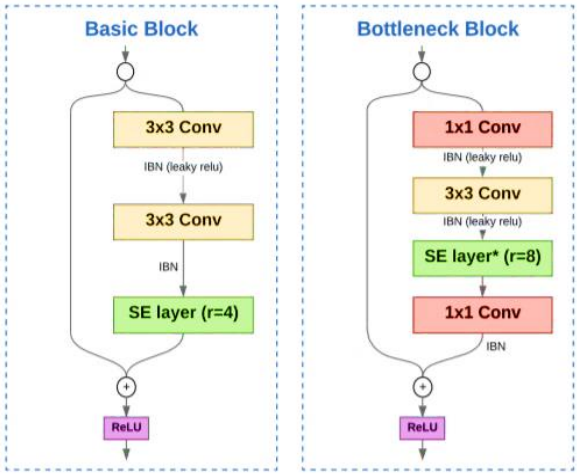
Năm 2016, YOLOv1 và YOLOv2 được xuất bản, cả hai đều trình bày cách tiếp cận khác với các thuật toán đề xuất vùng [12], [13]. Theo đó, YOLOv1 mang lại sự đột phá về tốc độ, nhưng về mặt hiệu năng thì lại kém hơn so với các thuật toán trước; YOLOv2 tốt hơn, chính xác và nhanh hơn so với các thuật toán trước đó. Độ chính xác của YOLOv1 thấp hơn so với Fast R-CNN [14] và Faster R-CNN [15] nhưng tốc độ nhận dạng nhanh hơn; Độ chính xác của YOLOv2 cũng như số khung hình trên giây (FPS) đã được cải thiện đáng kể. Được công bố năm 2020, YOLOv4 [16] đã mang lại những cải tiến đáng kể. Kết quả chỉ ra rằng, YOLOv4 là một bộ nhận dạng hàng đầu, nhanh và chính xác hơn so với các bộ nhận dạng vật thể hiện nay. YOLOv4 cải thiện độ chính xác trung bình và FPS của YOLOv3 [17] lần lượt là 10% và 20%.

Trong nghiên cứu về nhận dạng sản phẩm được đặt ở trên kệ hàng [18], nhóm nghiên cứu đã sử dụng YOLO để thực nghiệm trên tập Grocery và Imagenet. Trong [19] làm về bộ nhận dạng thanh toán bán lẻ với một camera được đặt ở phía trên bàn thanh toán, YOLO và CaffeNet được sử dụng để nhận dạng sản phẩm. Trong [20] nói về hệ thống giám sát tình trạng hàng hóa ở siêu thị với các camera IP, YOLO được lựa chọn bởi khả năng phát hiện với độ chính xác và hiệu năng theo thời gian thực cao.

Độ chính xác và tốc độ là hai yếu tố quan trọng khi xây dựng một bộ phát hiện vật thể. Theo như phân tích các nghiên cứu ở trên thì YOLOv4 là một lựa chọn hàng đầu.

1.2.2. Trích xuất đặc trưng với TResNet

Giai đoạn trích xuất đặc trưng đóng vai trò quan trọng trong bài toán thị giác máy tính [21], [22], [23]. Trong các bài toán liên quan đến phát hiện vật thể, mạng học sâu ResNet được sử dụng để trích xuất đặc trưng [24], [25]. Ý tưởng chính của ResNet là sử dụng kết nối tắt đồng nhất để xuyên qua một hay nhiều lớp, được thể hiện ở Hình 3. Phát triển dựa trên kiến trúc của mạng ResNet, TResNet [26] ra đời với 3 biến thể: TResNet-M, TResNet-L và TResNet-XL; Khác nhau về chiều sâu và số lượng kênh. Nghiên cứu [26] chỉ ra một số điểm nổi bật: (1) TResNet cải thiện sự cân bằng về độ chính xác và tốc độ; cho hiệu năng vượt trội hơn các mô hình học sâu hàng đầu trong tác vụ phát hiện vật thể và phân loại đa nhãn; (2) TResNet đã thay thế các lớp BatchNorm bằng InPlace-ABN [27] nhằm cải thiện việc sử dụng nguồn tài nguyên tối ưu của GPU – điều đóng vai trò quan trọng trong bài toán cần tốc độ thời gian thực; (3) Hàm kích hoạt thuần ReLu của ResNet50 được thay thế bằng hàm Leaky-Relu, cho độ chính xác cao hơn; (4) Cấu trúc mạng lưới (Hình 2), kết hợp khối cơ bản của ResNet34 và khối cổ chai của ResNet50. Ở khối cơ bản, lớp SE [28] được thêm vào trước khối cộng dư với hệ số duy giảm r = 4. Ở khối cổ chai, lớp SE được thêm vào sau khối tích chập 3x3 với r = 8, dấu * nghĩa là chỉ dùng ở giai đoạn 3.



Hình 2. *Khối cơ bản và khối cổ chai của mạng TresNet [26]*

1.2.3. Phân loại hình ảnh với thư viện tìm kiếm tương tự FAISS

Truy vấn hình ảnh là tìm kiếm những mẫu thông tin hình ảnh liên quan nhất đến dữ liệu truy vấn đầu vào. Về bản chất, truy vấn hình ảnh giống với phân loại hình ảnh [29]. Phương pháp quan trọng thường được sử dụng trong truy vấn hình ảnh là tìm kiếm tương tự [30], [31], phù hợp với những bài toán có cơ sở dữ liệu phức tạp như video hoặc hình ảnh được biểu diễn bởi các vector đặc trưng đa chiều [32]. Bài toán truy vấn hình ảnh được mô tả như sau: Đầu vào là một vector truy vấn; Kết quả trả về là danh sách gồm các vector trong cơ sở dữ liệu cho trước có khoảng cách Euclid gần nhất với vector truy vấn.

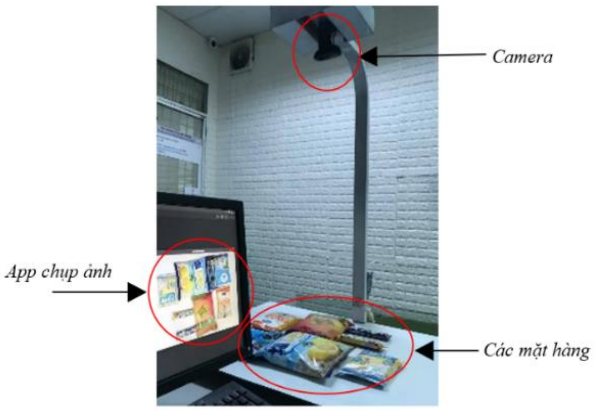
Hnsw [33] và Faiss [32] là hai thư viện hỗ trợ tìm kiếm tương tự được sử dụng phổ biến. Với tìm kiếm tương tự, cách tiếp cận bằng khoảng cách Euclid L2 thường được dùng, được định nghĩa như sau: Giả sử 2 vector X và Y được đại diện bởi 2 điểm $x = (x_1, x_2, \dots x_n)$, $y = (y_1, y_2, \dots y_n)$ trong không gian Euclid n chiều, khi đó khoảng cách L2 giữa 2 điểm x và y là d, được tính bởi:

$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

Với những trình bày về xu thế ứng dụng học sâu vào mảng thanh toán hàng hoá cũng như từ các phân tích và đánh giá ở trên, nhóm tác giả chọn ra các mô hình và thư viện phù hợp để xây dựng một bộ nhận dạng thanh toán hàng hoá. Các bước thực hiện được trình bày ở phần 1.3 và 1.4 của bài báo.

1.3. Xây dựng tập dữ liệu

Nhóm tác giả viết một ứng dụng bằng Python để chụp ảnh. Cách bố trí được thể hiện ở Hình 3.



Hình 3. *Camera chụp bao quát hàng hóa bên dưới*

1.3.1. Tập huấn luyện

Nhóm tác giả thử nghiệm trên 120 mặt hàng khác nhau được mua ở chuỗi cửa hàng Vinmart Việt Nam. Bộ dữ liệu huấn luyện gồm 7500 bức ảnh. Mỗi bức ảnh chứa 8 mặt hàng khác nhau được chụp bởi 1 camera đặt cố định ở bên trên như ở Hình 3. Khoảng cách từ camera đến mặt bàn thanh toán là 70 cm. Hình ảnh sau đó được gán nhãn bởi công cụ LabelImg [34]. Một ví dụ được trình bày trong Hình 4. Chi tiết các thông số được thể hiện ở Bảng 2.

Hình 4. *Một hình ảnh trong tập huấn luyện được đánh nhãn*

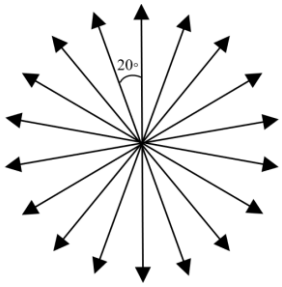
Bảng 2. *Mô tả về tập huấn luyện của BRC*

Thông số	Mô tả
Số mặt hàng	120
Số mặt hàng/ảnh	8
Số ảnh/mặt hàng	500
Số ảnh dùng để huấn luyện/mặt hàng	400
Số hình dùng để đánh giá quá trình huấn luyện/ mặt hàng	100
Tổng số hình ảnh	7500
Độ phân giải ảnh (pixel)	640x480
Ánh sáng môi trường (lux)	170

Nhóm tác giả huấn luyện 2 mô hình YOLOv4 và TResNet trên tập huấn luyện của BRC.

1.3.2. Tập kiểm thử

Để xây dựng tập kiểm thử, nhóm tác giả sử dụng camera, góc chụp, điều kiện ánh sáng môi trường, nền ảnh giống với lúc xây dựng tập huấn luyện. Mỗi mặt hàng được chụp 18 lần tương ứng 18 hướng khác nhau với góc chụp sai khác 20 độ như trong Hình 5.



Hình 5. *Mỗi mặt hàng có 18 hướng chụp khác nhau*

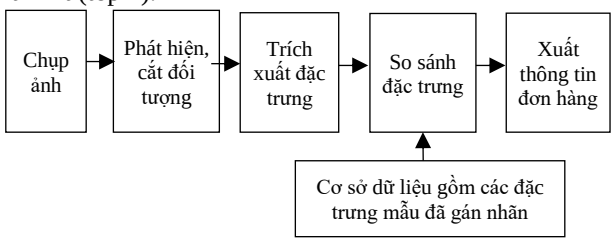
Các mô tả chi tiết được trình bày ở Bảng 3.

Bảng 3. *Mô tả về tập kiểm thử của BRC*

Thông số	Mô tả
Tổng số hình	5440
Số mặt hàng	80
Số hình/mặt hàng	68
Số hình dùng để trích xuất làm vector đặc trưng mẫu	18
Số hình dùng để trích xuất làm vector truy vấn	50
Độ phân giải ảnh (pixel)	640x480
Ánh sáng môi trường	170 lux

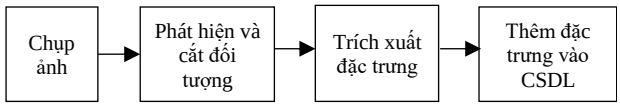
1.4. Xây dựng mô hình bộ nhận dạng thanh toán

Mô hình bộ nhận dạng thanh toán hàng hóa BRC được trình bày ở Hình 6. Đầu tiên, hình ảnh các mặt hàng cần thanh toán được ghi lại thông qua camera và phát hiện bởi YOLOv4. Sau đó, các đặc trưng của đối tượng được trích xuất bởi TResNet-M. Vector đặc trưng được truy vấn với thư viện tìm kiếm tương tự Faiss trong cơ sở dữ liệu nhằm lấy kết quả đầu ra và xuất thông tin đơn hàng lên màn hình. Ở giai đoạn truy vấn, nhóm tác giả lấy một kết quả trả về tốt nhất (top 1).



Hình 6. *Các công đoạn nhận dạng thanh toán sản phẩm*

Khi cần bổ sung một sản phẩm mới vào cơ sở dữ liệu, các công đoạn tiến hành được trình bày ở Hình 7.



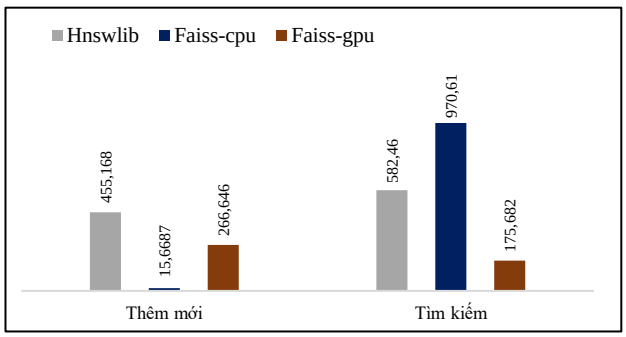
Hình 7. *Các công đoạn thêm mặt hàng mới vào cơ sở dữ liệu*

Tóm lại, các mô hình và thư viện cho mỗi giai đoạn được trình bày ở Bảng 4.

Bảng 4. *Các thư viện được sử dụng để xây dựng bộ nhận dạng*

Công đoạn	Mô hình	Thư viện
Phát hiện vật thể	YOLOv4	Darknet
Trích xuất đặc trưng	TResNet-M	Pytorch
Truy vấn hình ảnh	Tìm kiếm tương tự	FAISS

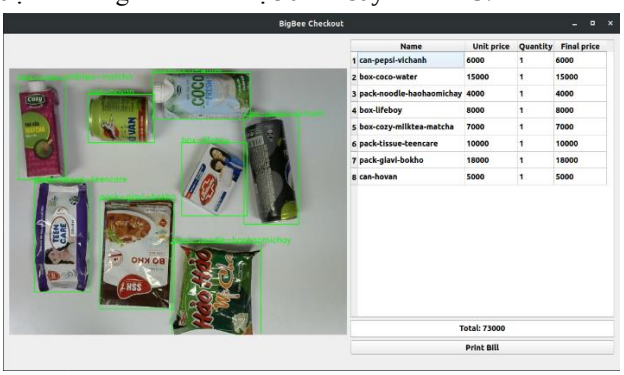
Để chọn ra thư viện phù hợp ở giai đoạn truy vấn hình ảnh, nhóm tác giả so sánh thời gian thêm mới và tìm kiếm một vector đặc trưng trong cơ sở dữ liệu sử dụng khoảng cách L2 của 3 thư viện: *Hnswlib*, *Faiss-cpu* và *Faiss-gpu*. Cấu hình sử dụng: CPU Intel Xeon 2.20 GHz 4 nhân, GPU NVIDIA Tesla P100 16GB. Tập kiểm thử lấy từ bộ dữ liệu BRC gồm 68 ảnh cho mỗi mặt hàng. Trong đó, 18 ảnh dùng để trích xuất đặc trưng mẫu và 50 ảnh để đích xuất xuất đặc trưng dùng cho truy vấn, thu được kết quả ở Hình 8.



Hình 8. *So sánh thời gian (µs) thêm mới và tìm kiếm đặc trưng theo L2 của Hnsw, Faiss-cpu và Faiss-gpu*

Theo đó, thời gian *Faiss-cpu* thêm mới một vector vào cơ sở dữ liệu nhanh nhất nhưng tìm kiếm một vector đặc trưng trong cơ sở dữ liệu lâu nhất. *Hnswlib* tốn nhiều thời gian hơn để thêm mới và tìm kiếm vector đặc trưng so với *Faiss-gpu*. Rõ ràng *Faiss-gpu* cho kết quả tốt nhất trong 3 thư viện.

Bộ nhận dạng BRC có chức năng nhận dạng thanh toán một lúc nhiều sản phẩm theo thời gian thực, giao diện được xây dựng với thư viện PyQt5 và OpenCV. Thông tin thanh toán hiển thị lên màn hình gồm có các trường: Tên mặt hàng, đơn giá, số lượng, giá tổng từng mặt hàng và tổng giá trị đơn hàng. Chi tiết được trình bày ở Hình 9.

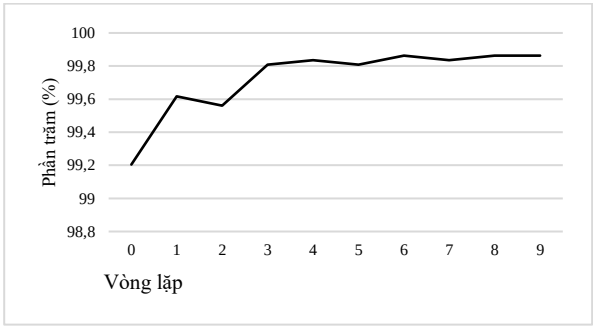


Hình 9. *Giao diện của ứng dụng BigBee Retail Checkout*

2. Kết quả

2.1. Quá trình huấn luyện

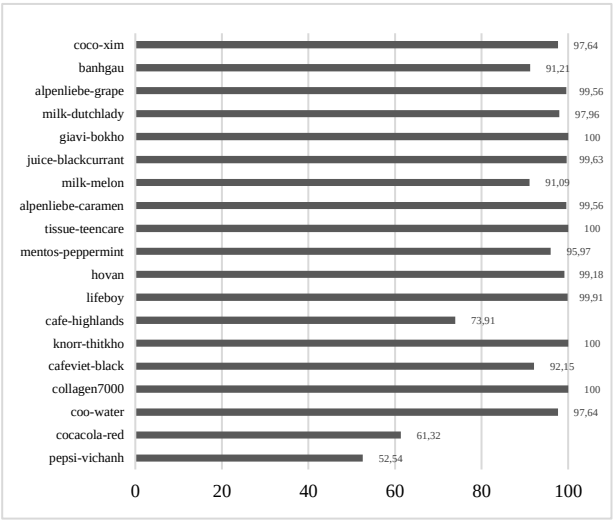
Nhóm tác giả sử dụng tập huấn luyện BRC để huấn luyện 2 mô hình YOLOv4 và TResNet-M trên máy tính có cấu hình Intel Xeon CPU 2.00GHz - 4 nhân 8 luồng, GPU NVIDIA Tesla P100 16GB và RAM 26GB. Với mô hình YOLOv4, thu được độ chính xác trung bình 99,8% sau 1000 vòng lặp huấn luyện đầu tiên. Đối với TResNet-M, độ chính xác trung bình top 1 sau 10 vòng lặp đầu tiên đều ở mức trên 99%, được thể hiện ở Hình 10.



Hình 10. *Độ chính xác trung bình top 1 của TResNet-M qua 10 vòng lặp đầu tiên*

2.2. Quá trình kiểm thử

Đối với mô hình TResNet-M, nhóm tác giả thu được độ chính xác trung bình dự đoán top 1 của các lớp hàng hóa là 92,18%. Độ chính xác của mô hình YOLOv4 đạt được là 99,25%. Nhóm tác giả trích chọn và trình bày độ chính xác của 19 lớp tương ứng với 19 mặt hàng phổ biến ở Hình 11:



Hình 11. *Độ chính xác khi kiểm thử của một số lớp hàng hóa*

Kết quả từ Hình 11 cho thấy:

(1) Các mặt hàng dạng lon có hình trụ tròn cho độ chính xác thấp: *pepsi-vichanh*, *cocacola-red*, *cafe-highlands* có độ chính xác lần lượt 52,54%; 61,32%; 73,91%;

(2) Các mặt hàng dạng hộp như: *milk-dutchlady*, *milk-melon*, *banhgau*, *coco-water*, *coco-xim* cho độ chính xác ở quanh mức 95%;

(3) Các mặt hàng dạng gói phẳng cho độ chính xác tuyệt đối 100%: *collagen700*, *tissue-teencare*, *giavi-bokho*, *giavi-thitkho*.

Nhận xét:

Vì tính chất cố hữu của vật phẩm trụ tròn là dễ dàng lăn trên bàn thanh toán nên dữ liệu camera thu được từ chúng sẽ khác nhau đáng kể qua mỗi lần thanh toán. Với cách xây dựng bộ dữ liệu huấn luyện BRC của nhóm tác giả, toàn bộ các mặt hàng đều được chụp cùng một số lượng hình ảnh, dẫn đến có sự 'không công bằng' đối với loại hình trụ tròn. Bởi vậy, cần nhiều dữ liệu huấn luyện hơn cho các loại hàng này. Ví dụ, một lon café cần nhiều dữ liệu hơn một gói café để mạng TResNet được huấn luyện tốt hơn.

2.3. Thử nghiệm, so sánh với phương pháp quét mã vạch

Nhóm tác giả tiến hành mua và thanh toán sản phẩm tại 5 cửa hàng Vinmart – tại đó phương pháp quét mã vạch đang được sử dụng. Cũng chính lượng vật phẩm được mua từ mỗi cửa hàng trên, nhóm tác giả tiến hành thanh toán bằng bộ nhận dạng BRC. Dữ liệu thu được của 2 phương pháp trên gồm số lượng mặt hàng, thời lượng thanh toán ở 5 cửa hàng (CH) được so sánh và thể hiện ở Bảng 5. Thời lượng thanh toán được tính từ lúc đặt sản phẩm đầu tiên lên bàn đến lúc tổng giá trị hóa đơn được xuất ra.

Bảng 5. *So sánh thời gian thanh toán trung bình bởi nhân viên và bộ nhận dạng BigBee Retail Checkout*

Cửa hàng	Số lượng mặt hàng thanh toán	Thời gian thanh toán (giây)	
		Nhân viên	BigBee Retail Checkout
CH 1	8	31,12	12,35
CH 2	9	20,15	12,50
CH 3	10	25,38	13,10
CH 4	11	29,45	14,01
CH 5	12	33,23	16,45
Trung bình	**10**	**27,87**	**13,68**

Độ chính xác trung bình của bộ nhận dạng BRC được ghi lại ở Bảng 6. Ví dụ, với 8 vật phẩm mua ở cửa hàng 1, BRC nhận dạng 8 vật phẩm với 8 mức chính xác khác nhau, tính trung bình là 96,5.

Bảng 6. *Độ chính xác trung bình của BRC ở 5 cửa hàng*

Cửa hàng	CH1	CH2	CH3	CH4	CH5
Độ chính xác	96,5	95,3	91,7	93,6	95,6

Từ số liệu ở Bảng 6, nếu xem độ chính xác khi thanh toán bằng quét mã vạch bởi nhân viên là 100% thì giải pháp BRC đạt độ chính xác trung bình 94,54% khi thử nghiệm với 5 lần thanh toán. Trong phạm vi nghiên cứu, dữ liệu huấn luyện và kiểm thử của BRC còn hạn chế bởi việc xây dựng một tập dữ liệu tốn nhiều công sức và thời gian; Vì thế nhóm nghiên cứu vẫn tiếp tục bổ sung, phát triển bộ dữ liệu ở các phiên bản tiếp theo để cải thiện độ chính xác.

Về mặt thời gian, giải pháp của nhóm tác giả cải thiện tốc độ thanh toán nhanh đáng kể. Số liệu ở Bảng 5 cho thấy, thời gian trung bình thực hiện bởi bộ nhận dạng BRC là 13,68 giây, nhanh gần gấp đôi so với giải pháp quét mã vạch với 27,87 giây. Tuy nhiên, đây chỉ là kết quả thử nghiệm ban đầu, cần có nhiều nghiên cứu hơn để kết luận. Thời gian tiến hành thanh toán còn phụ thuộc vào nhiều yếu tố khác chẳng hạn như số lượng mặt hàng, vị trí in mã vạch, hiệu năng máy quét mã vạch, kĩ năng và kinh nghiệm của nhân viên tại quầy.

3. Kết luận

Đầu tiên, bài báo đã nêu ra xu thế cũng như thách thức thiếu hụt dữ liệu của việc ứng dụng kỹ thuật học sâu vào mảng nhận dạng thanh toán hàng hóa. Tiếp đó, nhóm tác giả phân tích và lựa chọn các kỹ thuật và mô hình phù hợp để ứng dụng vào việc xây dựng một bộ nhận dạng thanh toán: YOLOv4 cho tác vụ phát hiện vật thể; Mô hình TResNet cho giai đoạn trích xuất đặc trưng; Thư viện tìm kiếm tương tự Faiss để truy vấn hình ảnh để tìm đầu ra của bài toán. Giải pháp của nhóm tác giả bước đầu thử nghiệm trên tập dữ liệu gồm các hình ảnh của các mặt hàng được mua ở cửa hàng Vinmart, chưa được triển khai ứng dụng vào thực tiễn. Kết quả thử nghiệm ban đầu chỉ ra rằng, bộ nhận dạng thanh toán BigBee Retail Checkout cho kết quả nhanh hơn đáng kể so với phương pháp quét mã vạch. Tuy nhiên, cần thực hiện thêm nhiều nghiên cứu sâu khác để đánh giá chi tiết và tính khả thi khi áp dụng trên số lượng lớn mặt hàng.

Bên cạnh đó, hiểu được việc ứng dụng kỹ thuật học sâu và thị giác máy tính vào các lĩnh vực thanh toán hàng hóa là cần thiết, nhưng trong bối cảnh chỉ có một số ít tập dữ liệu có sẵn, nhóm tác giả đã giới thiệu bộ dữ liệu hàng hóa thanh toán BRC, góp phần giải quyết thách thức về sự thiếu hụt dữ liệu. Hơn nữa, nghiên cứu cũng đã chỉ ra một số khó khăn cụ thể khi triển khai đối với các mặt hàng dạng hình trụ, hình hộp. Bổ sung thêm dữ liệu huấn luyện cho các loại hàng này là một trong những giải pháp cần thực hiện.

Tóm lại, bài báo đã có 4 đóng góp chính: (1) Phân tích lựa chọn các mô hình và thư viện phù hợp sử dụng cho bài toán nhận dạng thanh toán hàng hóa; (2) Đề xuất giải pháp xây dựng một bộ nhận dạng thanh toán sản phẩm, bước đầu thử nghiệm có hiệu quả về mặt thời gian thanh toán; (3) Giới thiệu bộ dữ liệu hàng hóa thanh toán BRC góp phần phục vụ cộng đồng nghiên cứu học sâu; (4) Nêu ra những khó khăn và giải pháp khi triển khai xây dựng bộ nhận dạng thanh toán hàng hóa.

Lời cảm ơn: Bài báo này được tài trợ bởi Quỹ Khoa học Công nghệ Murata và Trường Đại học Bách khoa – Đại học Đà Nẵng với đề tài có mã số T2020-02-09MSF.

TÀI LIỆU THAM KHẢO

[1] B. Santra and D. P. Mukherjee, “A comprehensive survey on computer vision-based approaches for automatic identification of products in retail store”, *Image and Vision Computing*, 2019, vol. 86, 45–63.

[2] Jupiter Research, “AI spending by retailers to reach $12 billion by 2023, driven by the promise of improved margins”, *Jupiter Press Release*, 2019.

[3] F. D. Orel and A. Kara, “Supermarket self-checkout service quality, customer satisfaction, and loyalty: empirical evidence from an emerging market”, *Journal of Retailing and Consumer Services*, 2014, vol. 21, 118–129.

[4] A. C. R. Van Riel, J. Semeijn, D. Ribbink, and Y. BomertPeters, “Waiting for service at the checkout: negative emotional responses, store image and overall satisfaction”, *Journal of Service Management*, 2012, vol. 23, số 2, 144-169.

[5] F. Morimura and K. Nishioka, “Waiting in exit-stage operations: expectation for self-checkout systems and overall satisfaction”, *Journal of Marketing Channels*, 2016, vol. 23, no. 4, 241–254.

[6] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, "Deep Learning for Computer Vision: A Brief Review", *Computational Intelligence and Neuroscience*, vol. 2018, ID 7068349, 13 trang, 2018, https://doi.org/10.1155/2018/7068349.

[7] Yuchen Wei, Son Tran, Shuxiang Xu, Byeong Kang, Matthew Springer, "Deep Learning for Retail Product Recognition: Challenges and Techniques", *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8875910, 23 pages, 2020. https://doi.org/10.1155/2020/8875910.

[8] L. Karlinsky, J. Shtok, Y. Tzur, and A. Tzadok, “Fine-grained recognition of thousands of object categories with singleexample training”, *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 4113–4122.

[9] P. Follmann, T. Bottger, P. Hartinger, R. Konig, and M. Ulrich,

MVTec "D2S: densely segmented supermarket dataset", *Proceedings of the 2018 European Conference on Computer Vision (ECCV)*, 2018.

[10] X. S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu, "RPC: a large-scale retail product checkout dataset", 2019.

[11] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review", in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 779-788.

[13] Joseph Redmon, Ali Farhadi, "YOLO9000: better, faster, stronger", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263-7271.

[14] R. Girshick, "Fast R-CNN", *Proceedings of the IEE.E international conference on computer vision*, 2015, 1440-1448.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks", *Advances in neural information processing systems*, 2015, 91-99.

[16] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", *arXiv preprint arXiv:2004.10934*, 2020.

[17] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement", *arXiv preprint arXiv:1804.02767*, 2018.

[18] C. G. Melek, E. B. Sonmez and S. Albayrak, "Object Detection in Shelf Images with YOLO", *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*, 2019, pp. 1-5, doi: 10.1109/EUROCON.2019.8861817.

[19] Bing-Fei Wu, Wan-Ju Tseng, Yung-Shin Chen, Shih-Jhe Yao, Po-Ju Chang, "An Intelligent Self-Checkout System for Smart Retail", *International Conference on System Science and Engineering (ICSSE)*, 2016.

[20] Sandeep Kumar Yedla, V. M. Manikandan, Panchami V, "Real-time Scene Change Detection with Object Detection for Automated Stock Verification", *5th International Conference on Devices, Circuits and Systems*, 2020.

[21] G. Kumar and P. K. Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems", *Fourth International Conference on Advanced Computing & Communication Technologies*, 2014, 5-12.

[22] Dong ping Tian, "A Review on Image Feature Extraction and Representation Techniques", *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 8, No. 4, 2013, 385-395

[23] X. Jiang, "Feature extraction for image recognition and computer vision", *2nd IEEE International Conference on Computer Science and Information Technology*, 2009, 1-15.

[24] X. Lu, X. Kang, S. Nishide and F. Ren, "Object detection based on SSD-ResNet", *IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2019, 89-92.

[25] M. F. Haque, H. Lim and D. Kang, "Object Detection Based on VGG with ResNet Network", *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, 2019, pp. 1-3, doi: 10.23919/ELINFOCOM.2019.8706476.

[26] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, Itamar Friedman, "High Performance GPU-Dedicated Architecture" *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1400-1409.

[27] Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[28] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 7132–7141.

[29] Xie, Lingxi & Hong, Richang & Zhang, Bo & Tian, Qi, "Image Classification and Retrieval are ONE, *ICMR'15*, 2015, 3-10.

[30] M. Wang, Y. Ming, Q. Liu and J. Yin, "Similarity search for image retrieval via local-constrained linear coding", *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, 1-6.

[31] Rahman M.M., Bhattacharya P., Desai B.C, "Similarity Searching in Image Retrieval with Statistical Distance Measures and Supervised Learning", *Pattern Recognition and Data Mining ICAPR 2005: Pattern Recognition and Data Mining*, 2005, vol 3686, pp 315-324, https://doi.org/10.1007/11551188_34

[32] Jeff Johnson, Matthijs Douze, Hervé Jégou, "Billion-scale similarity search with GPUs", *arXiv preprint arXiv:1702.08734*, 2017.

[33] Github, "Hierarchical Navigable Small World", *Release v0.5.0, https://github.com/nmslib/hnswlib*, 2021.

[34] Johnson, Jeff and Douze, Matthijs and J, Tzutalin, "LabelImg", *arXiv preprint arXiv:1702.08734*, 2017.