

Suy luận thông minh cho Neural đa chữ số

Giải mã mã vạch dựa trên mạng

Thao Do
KAIST
Daejeon, Hàn Quốc
thaodo@kaist.ac.kr

Yalew Tolcha
KAIST
Daejeon, Hàn Quốc
yalewkidane@kaist.ac.kr

Tae Joon Jun
Trung tâm y tế Asan
Seoul, Hàn Quốc
taejoon@amc.seoul.kr

Daeyoung Kim
KAIST
Daejeon, Hàn Quốc
kimd@kaist.ac.kr

trường tượng—Các mã vạch có mặt khắp nơi và đã được sử dụng trong hầu hết các hoạt động hàng ngày quan trọng trong nhiều thập kỷ. Tuy nhiên, hầu hết các bộ giải mã truyền thống yêu cầu mã vạch có nền tảng tốt trong một điều kiện tương đối tiêu chuẩn. Trong khi các mã vạch có điều kiện cũ hơn như thiếu sáng, bị tắc, mờ, nhân và xoay thường được ghi lại trong thực tế, các bộ giải mã truyền thống đó cho thấy sự yếu kém trong nhận dạng. Một số công trình đã cố gắng giải quyết những mã vạch đầy thách thức đó, nhưng nhiều hạn chế vẫn còn tồn tại. Công việc này nhằm giải quyết vấn đề giải mã bằng cách sử dụng mạng nơ-ron tích tụ sâu với khả năng chạy trên các thiết bị di động. Đầu tiên, chúng tôi đề xuất một sửa đổi đặc biệt của suy luận dựa trên tính năng có tổng kiểm tra và tăng thời gian thử nghiệm, được đặt tên là Suy luận thông minh (SI), trong giai đoạn dự đoán của một mô hình được đào tạo. SI tăng đáng kể độ chính xác và giảm dự đoán sai cho các mô hình được đào tạo. Thứ hai, chúng tôi đã tạo ra một bộ dữ liệu đánh giá thực tế lớn về mã vạch 1D được chụp thực trong các điều kiện thử thách khác nhau để kiểm tra các phương pháp của chúng tôi một cách mạnh mẽ, công khai cho các nhà nghiên cứu khác. Kết quả thí nghiệm đã chứng minh hiệu quả của SI với độ chính xác cao nhất là 95,85%, vượt trội hơn nhiều bộ giải mã hiện có trên bộ đánh giá. Cuối cùng, chúng tôi đã thu nhỏ thành công mô hình tốt nhất bằng cách chất lọc kiến thức thành một mô hình nông có độ chính xác cao (90,85%) với tốc độ suy luận tốt là 34,2 ms cho mỗi hình ảnh trên thiết bị cạnh thực. công khai cho các nhà nghiên cứu khác. Kết quả thí nghiệm đã chứng minh hiệu quả của SI với độ chính xác cao nhất là 95,85%, vượt trội hơn nhiều bộ giải mã hiện có trên bộ đánh giá. Cuối cùng, chúng tôi đã thu nhỏ thành công mô hình tốt nhất bằng cách chất lọc kiến thức thành một mô hình nông có độ chính xác cao (90,85%) với tốc độ suy luận tốt là 34,2 ms cho mỗi hình ảnh trên thiết bị cạnh thực.

Điều khoản lập chỉ mục—Barcode, mạng nơ-ron phức hợp

Tôi tôiGIỚI THIỆU

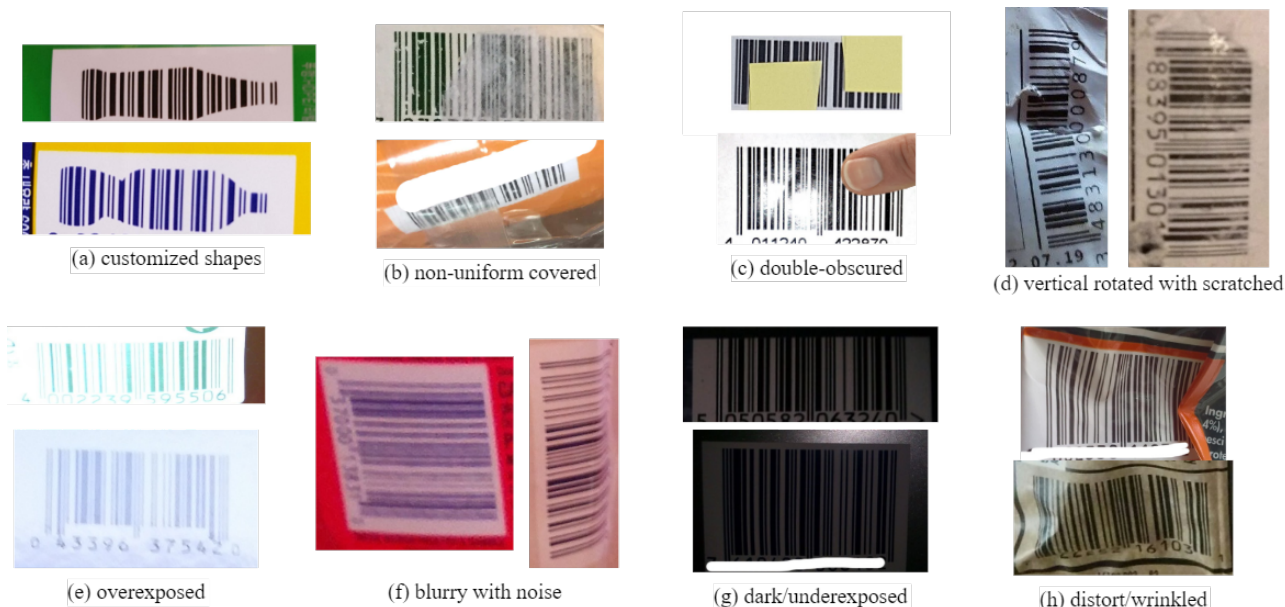
Mã vạch 1D tuyến tính xuất hiện vào những năm 1970 và hiện nay đã trở nên phổ biến trên hầu hết các sản phẩm tiêu dùng và phục vụ hậu cần do tính dễ nhận dạng của nó. Một số công nghệ gắn thẻ mới hơn đã xuất hiện trong những thập kỷ qua cho phép lưu trữ nhiều thông tin hơn (ví dụ như RFID, NFC). Tuy nhiên, không ai trong số họ thay thế hoàn toàn vai trò của nó trong ngành vì di sản và nền kinh tế của nó. Chi phí in mã vạch thấp và độ bền của thẻ dưới những hư hỏng nhỏ khiến nó vẫn là một tiêu chuẩn công nghiệp (được tiêu chuẩn hóa bởi GS1) trong những thập kỷ tới.

Một thuộc tính thiết yếu của công nghệ gắn thẻ là nó phải được đọc nhanh chóng, mạnh mẽ và chính xác bằng cách sử dụng trình đọc. Đối với trường hợp mã vạch, đầu đọc (hoặc máy quét) được phân loại thành 3 loại: dựa trên laser, dựa trên đèn LED và dựa trên camera. Trong 2 loại đầu tiên, tia laser / tia LED cần phải gần với mã vạch, không có sọc bị che khuất trên đường tia và gặp phải vấn đề phát quá nóng. Đầu đọc Camerabased có một số lợi thế so với các giải pháp dựa trên laser / LED. Lợi thế đầu tiên được xây dựng dựa trên thực tế là nhiều điện thoại thông minh được tích hợp camera chất lượng cao đã

đang được sử dụng. Với kết nối Internet, các ứng dụng di động hữu ích ra đời bằng cách truy xuất trực tuyến thông tin sản phẩm và đưa ra thông tin thành phần, cảnh báo dị ứng, lượng calo, so sánh giá giữa người bán; hoặc đối với các nhà bán lẻ, họ tìm hiểu các sản phẩm bắt mắt, có phản hồi của người tiêu dùng, v.v. (ví dụ: trong [1]). Một ưu điểm khác của giải pháp dựa trên camera là khả năng nhận dạng nhiều và tầm xa nhờ sự hỗ trợ của các thuật toán thị giác máy tính.

Tuy nhiên, hầu hết các kỹ thuật hiện tại (xử lý hình ảnh tĩnh và đối sánh mẫu) đang được sử dụng trong các đầu đọc dựa trên máy ảnh đều có những sai sót làm hạn chế khả năng sử dụng của chúng. Vấn đề chính với chúng là nhu cầu nhập liệu kiểu quét phẳng được đóng khung tốt hơn so với chụp bình thường. Không thể nhận biết được các điều kiện hoang dã nhưng được chụp phổ biến như thiếu sáng, bị che khuất, mờ hoặc cong, không nằm ngang (như trong Hình 1). Điều này yêu cầu người dùng sửa lỗi không hợp thời và làm chậm quá trình quét. Có 2 nhiệm vụ riêng biệt trong quá trình quét mã vạch: phát hiện (tức là định vị) nơi có vùng mã vạch trong hình ảnh và giải mã vùng được phát hiện thành chuỗi mã vạch. Các công trình gần đây cho thấy rằng nhiệm vụ đầu tiên gần như được giải quyết ngay cả trong những điều kiện đầy thử thách.

Mặt khác, việc giải mã những mã vạch đầy thách thức đó vẫn cần được cải thiện vì các công trình hiện tại vẫn còn nhiều hạn chế. Các phương pháp truyền thống được trình bày trong [2], [3], [4] áp dụng các kỹ thuật truyền thống như phép biến đổi Hough, phương pháp tiếp cận theo đường quét với các ngưỡng để mã hóa dựa trên các giả định nhất định về đặc điểm mã vạch trong khi chúng không phải lúc nào cũng đúng. Nhiều người đánh giá công cụ của họ trên những bộ chưa được xuất bản, một số đã xuất bản những bộ của họ nhưng nhỏ và không đủ điều kiện thử thách. Với những hạn chế đó và những thành công của mạng nơ-ron tích chập (CNN) trong nhiều ứng dụng, [5] là công trình đầu tiên được đề xuất sử dụng CNN để giải mã những mã khó này. Tuy nhiên, công việc của họ có một số điểm yếu khiến hiệu quả hoạt động thấp hơn nhiều so với tiềm năng của CNN. Không chỉ các trình trích xuất tính năng CNN của họ đơn giản mà cả giả định đầu vào của họ cũng được đơn giản hóa quá mức. Họ chỉ giả định mã vạch ngang là đầu vào; Bộ thử nghiệm của họ được làm từ mã vạch được tạo hình chữ nhật in trên giấy thường trong khi mã vạch ngoài đời thực được in theo hình dạng tùy chỉnh (ví dụ: hình biểu tượng coca) trên các vật liệu khác nhau với nhiều loại biến dạng và đôi khi bị che phủ bởi màng. Họ cũng không xem xét khả năng chạy tác vụ trên các thiết bị cạnh vì các mô hình chưa được tối ưu hóa.



Hình 1. Điều kiện thách thức

Do đó, trong nghiên cứu này, chúng tôi đã đề xuất một phương pháp dựa trên CNN để giải quyết nhiệm vụ giải mã với các điểm đóng góp sau: (i) chúng tôi đề xuất Suy luận thông minh - 3 thuật toán tận dụng tính năng có tổng kiểm tra và tăng thời gian thử nghiệm được xây dựng trên nền tảng sâu được đào tạo CNN mô hình giúp tăng đáng kể độ chính xác của mô hình và giảm dự đoán sai; (ii) chúng tôi đã tạo tập dữ liệu mã vạch EAN13 cắt 2500 mẫu thử thách (UPC-A, ISBN13 là các tập con của nó) từ các hình ảnh được chụp thực trong các điều kiện khác nhau (bao gồm cả khắc nghiệt) trên nhiều sản phẩm - tập dữ liệu này được xuất bản để các nhà nghiên cứu khác đánh giá mô hình của họ và khuyến khích đóng góp nhiều hơn cho nhiệm vụ này; (iii) cuối cùng, chúng tôi áp dụng kỹ thuật chất lọc kiến thức với mục tiêu là có một mô hình nhẹ từ mô hình tốt nhất phù hợp trên các thiết bị cầm tay, kết quả thử nghiệm do đó khẳng định khả năng xảy ra bằng tốc độ suy luận tốt trên bảng cạnh thực.

II. CÔNG VIỆC ĐÃ ĐẠT ĐƯỢC

Về nhiệm vụ định vị mã vạch, có một số phương pháp được trình bày với việc cải thiện hiệu suất trong thập kỷ qua. Năm 2011, Lin và cộng sự [6] đã trình bày các phương pháp nhận dạng mã vạch dung nạp nhiễu và xoay vòng đầu tiên. Công việc này tập trung nhiều hơn vào việc phát hiện vấn đề bằng cách sử dụng một số sơ đồ xử lý hình ảnh như lọc làm mịn Gaussian để phân đoạn các vùng mã vạch, tăng cường các đường sọc, xoay các vùng theo góc ngang và đưa vào bộ giải mã có biểu quyết. Mặc dù phương pháp này hoạt động tốt trên mã vạch xổ số (in trên giấy thường), nhưng nó vẫn còn chậm và không có độ chính xác cao trên tập dữ liệu về các sản phẩm hàng hóa không rõ ràng về mức độ thử thách. Katona và cộng sự [7] vào năm 2012 đã đề xuất một phương pháp sử dụng các phép toán hình thái học để phân đoạn mã vạch 1D và 2D dưới dạng mờ, nhiễu, cắt và các điều kiện quay khác nhau với hiệu suất tốt. Soros và cộng sự [8] tiếp tục

xử lý hiện tượng mờ bằng cách sử dụng ma trận cấu trúc và độ bão hòa từ hệ màu HSV để phát hiện mã vạch mờ tốt hơn nhưng với chi phí là tốc độ giảm vào năm 2013. Gần đây, Creusot et al. [9] đã đề xuất một phương pháp nhanh hơn cho mã vạch mờ dựa trên Máy dò phân đoạn đường sau công trình trước đó của họ [10] bằng cách sử dụng Vùng cực trị ổn định tối đa cho thấy nhạy với mờ. Theo một cách khác, Hansen [11] lần đầu tiên thử áp dụng mô hình học sâu phát hiện đối tượng (YOLO) trên cả mã 1D và 2D với tỷ lệ phát hiện hộp giới hạn tốt nhất.

Trong khi nhiệm vụ phát hiện mã vạch gần đạt đến mức bão hòa, một số công trình về giải mã đã được đề xuất một cách thưa thớt kể từ những năm 1990. Các tác phẩm ban đầu [2], [3] đã đạt được mục tiêu của mình bằng các kỹ thuật như biến đổi Hough, vị trí đỉnh dựa trên wavelet trên các đầu vào đơn giản (kiểu quét) của chúng. Wachenfeld và cộng sự [4] đã đề xuất phương pháp tiếp cận dựa trên dòng quét kèm theo tập dữ liệu EAN13 (còn được gọi là *MuensterDB*). Tuy nhiên, vì phương pháp của họ dựa trên cách tiếp cận dòng quét vào thời điểm đó, nó chỉ hoạt động tốt trên mã vạch xoay nhẹ ($\pm 15^\circ$), xoay mạnh hơn hoặc bị bóp méo sẽ là vấn đề. Tương tự như [7], Zamberletti và cộng sự [12] cũng giải quyết vấn đề mã vạch mất nét (mờ) bằng cách sử dụng mô hình perceptron nhiều lớp để tìm các thông số của ngưỡng thích ứng (thay vì nhị phân tiêu chuẩn) để khôi phục hình ảnh mờ thành hình ảnh rõ ràng hơn và đưa nó vào Zxing [13] để giải mã. Tuy nhiên, cách tiếp cận này rất đơn giản với khả năng thu hồi thấp và tốn ít thời gian (2 bước). Gần đây, Yang và cộng sự [14] đã cố gắng giải quyết 2 nhiệm vụ trên 5 bộ dữ liệu nghiêm ngặt. Công việc vượt trội hơn tất cả các phương pháp khác rất tốt trên mã vạch EAN13, nhưng vì phương pháp này chủ yếu dựa trên các tính năng và phân tích của cơ sở scanline và thủ công cho từng điều kiện khó khăn,

điều kiện bị che khuất (Hình 1) sẽ không thể áp dụng được. Cuối cùng, Fridborn [5] vào năm 2017 lần đầu tiên tận dụng sức mạnh của CNN để trích xuất trực tiếp các tính năng và dự đoán tới 13 kết quả đầu ra (tương ứng với 13 chữ số) đồng thời (tương tự như [15] trong bài toán Số nhà ở Chế độ xem phố). So với các phương pháp làm nổi bật truyền thống và thủ công, phương pháp dựa trên CNN tương đối đơn giản hơn và theo hướng dữ liệu hơn là phân tích từng trường hợp cụ thể. Một ví dụ rõ ràng là điều kiện được thu thập kép là vấn đề đối với các phương pháp tiếp cận dựa trên đường quét nhưng có thể dễ dàng học được và khắc phục bằng bộ phân loại CNN. Do đó, công việc của chúng tôi cũng dựa trên CNN, tuy nhiên, khác với [5] ở các điểm sau: (i) chúng tôi sử dụng các mô hình trích xuất tính năng CNN tiên tiến hơn; (ii) giả định đầu vào của chúng tôi thực tế hơn cũng như bộ đào tạo và bộ đánh giá của chúng tôi bao gồm nhiều trường hợp hơn; (iii) chúng tôi đề xuất Suy luận Thông minh khai thác thuộc tính tổng kiểm tra của chuỗi mã vạch để nâng cao độ chính xác của mô hình; (iv) chúng tôi đã xem xét giảm thiểu và xác minh khả năng của phương pháp tiếp cận dựa trên CNN trên một thiết bị cạnh thực.

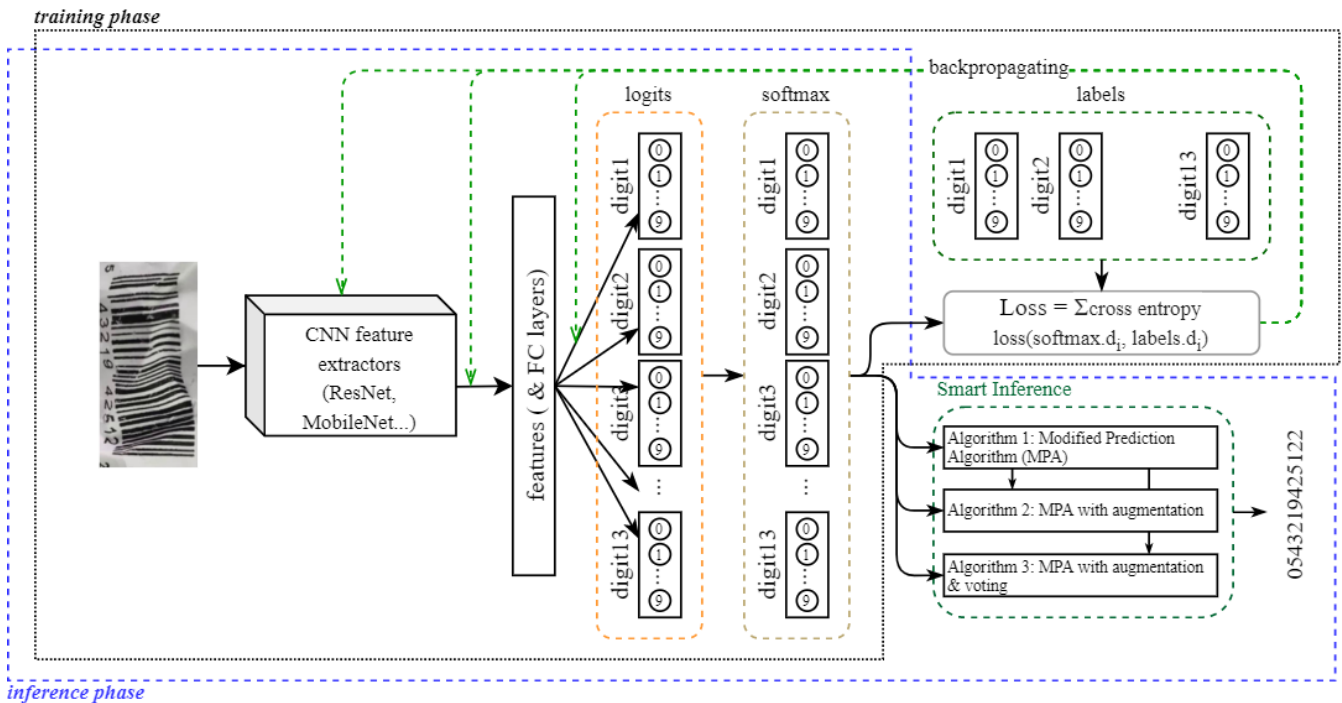
Đề cập đến việc tăng thời gian thử nghiệm mà chúng tôi đã sử dụng trong công trình này để nâng cao độ chính xác suy luận của một mô hình, kỹ thuật này thường được sử dụng trong học sâu như khảo sát này [16] và có thể được tìm thấy trong *AlexNet* [17], *ResNet* [18]. Trong khi tăng cường dữ liệu thời gian đào tạo cung cấp nhiều biến thể hơn của tập dữ liệu để cho phép mô hình cũng tìm hiểu tất cả các biến thể có thể có; tăng thời gian thử nghiệm cũng áp dụng một số sửa đổi thích hợp cho các mẫu ban đầu, để mô hình đưa ra nhiều dự đoán về các phiên bản sửa đổi đó và chọn một dự đoán phù hợp nhất trong số các dự đoán này bằng cơ chế bỏ phiếu. Cách tăng cường dữ liệu để có hiệu suất tốt hơn cũng là một chủ đề thời thượng trong học sâu hiện nay với các bài báo như *AutoAugment* [19], *Augment thông minh*-

tation [20]. Trong công việc của mình, chúng tôi đã tích hợp tăng thời gian thử nghiệm vào Suy luận thông minh khá hiệu quả.

Về chủ đề nén mô hình và khả năng ứng dụng học sâu trên thiết bị di động, Cheng và cộng sự [21] đã phân loại các phương pháp thành 4 loại: Cắt và chia sẻ tham số, phân tích nhân tử hạng thấp, bộ lọc tích chập và chốt lọc kiến trúc. Cái đầu tiên giảm các tham số dư thừa không nhạy cảm với hiệu suất trong khi cái thứ hai sử dụng phân rã ma trận để ước tính các tham số thông tin. Kỹ thuật thứ ba xây dựng các bộ lọc đặc biệt để lưu các tham số chỉ cho các lớp chập, trong khi kỹ thuật cuối cùng đào tạo một mạng nơ-ron nhỏ gọn với kiến trúc được chốt lọc từ một mô hình lớn được gọi là mô hình giáo viên. Để đơn giản, trong bài báo này, chúng tôi sử dụng phương pháp chưng cất tri thức ban đầu (KD) được đề xuất bởi Hinton và cộng sự [22].

III. MẠO ĐỨC

Phương pháp tiếp cận cơ sở mà chúng tôi sử dụng trong công việc này là đào tạo mô hình xác suất giải mã chuỗi mã vạch cho hình ảnh mã vạch như [5]. Để cho Đại diện cho trình tự mã vạch và X đại diện cho hình ảnh mã vạch đầu vào. Mục tiêu là tìm hiểu một mô hình $P(D/X)$ bằng cách tối đa hóa hàm log-likelihood $\log P(D/X)$ trên tập huấn luyện. D được mô hình hóa như một bộ sưu tập của M biến ngẫu nhiên D_1, \dots, D_M đại diện cho các chữ số của dãy đã giải mã. Để đơn giản hóa, giả sử rằng giá trị của mỗi chữ số là độc lập với nhau, do đó xác suất của một chuỗi $d = d_1, \dots, d_M$ được đưa ra bởi $P(D = d/X) = \prod_{i=1}^M P(D_i = d_i/X)$. Mỗi của các chữ số là rời rạc và có 10 giá trị có thể có (0 đến 9). Điều này có nghĩa là mỗi chữ số có thể được biểu diễn bằng một bộ phân loại softmax nhận dưới dạng các tính năng đầu vào được trích xuất từ X qua



Hình 2. Kiến trúc tổng thể

một CNN. Loại mô hình này ban đầu được đề xuất bởi [15], vì vậy chúng tôi gọi nó là Multidigit CNN. Trong giai đoạn huấn luyện, tổn thất được tính bằng tổng của tất cả các mất mát entropy chéo của các chữ số như bình thường. Tuy nhiên, trong giai đoạn suy luận, thay vì suy luận thông thường, chúng tôi đề xuất một sửa đổi có tên là Suy luận thông minh (SI), là một trong những đóng góp chính trong công việc này. Chi tiết của SI được mô tả trong đoạn tiếp theo. Mô hình tổng thể được thể hiện trong Hình 2.

A. Suy luận thông minh

Thông thường sau khi nhận được nhật ký từ hình ảnh mã vạch đã cho của mô hình, chúng tôi áp dụng hàm softmax để nhận xác suất của từng giá trị (0 đến 9) cho tất cả N các chữ số; sau đó, chúng tôi chọn giá trị có xác suất cao nhất. Bằng cách này, cuối cùng chúng tôi có N -dãy chữ số từ các giá trị có xác suất cao nhất. Tuy nhiên, giá trị có xác suất cao nhất không phải lúc nào cũng là giá trị chính xác. Thay vào đó, giá trị đúng có thể là giá trị có xác suất cao thứ hai hoặc cao thứ ba. Bên cạnh đó, vì hầu hết các mã vạch 1D đều có đặc điểm về sự hài lòng của tổng kiểm tra là [23]. Để cho D là chuỗi mã vạch, $D[tô]$ là chữ số $tô$ thứ $tô$ của chuỗi từ trái sang phải, L là độ dài của chuỗi mã vạch (ví dụ độ dài của EAN13 là 13) (vì vậy chữ số đầu tiên là $D[1]$), thuộc tính tổng kiểm tra có thể được tóm tắt dưới dạng phương trình sau:

$$\begin{aligned} & (D[L-0]*1 + D[L-1]*3 + D[L-2]* \\ & 1 + D[L-3]*3 + ... + D[L-2tô]*1 + D[L \\ & -2tô-1]*3 + ... + D[1]*1 \text{ nếu } L \bmod 2 == 0 \\ & \text{khác } 3) \bmod 10 = 0 \end{aligned} \tag{1}$$

Tận dụng đặc điểm này, ý tưởng ban đầu của chúng tôi là tạo ra nhiều hơn một chuỗi dự đoán không chỉ từ giá trị có xác suất cao nhất mà còn từ giá trị có 2^{nd} xác suất (hoặc 3^{rd}) cao nhất cho mỗi chữ số của N các chữ số; sau đó, chúng tôi xác minh các kết hợp đó bằng phương trình (1). Theo trực quan, khoảng cách lớn hơn giữa giá trị có xác suất cao nhất và giá trị có 2^{nd} (hoặc 3^{rd}) xác suất cao nhất thì mô hình càng tự tin dự đoán giá trị có xác suất cao nhất và ngược lại. Do đó, ưu tiên xem xét những chữ số có khoảng trống nhỏ nhất mà mô hình dễ bị nhầm lẫn hơn và ít chắc chắn hơn ở giá trị duy nhất có xác suất cao nhất. Để cho V là số giá trị có xác suất cao nhất và L là L lập lại tối đa là số chữ số được coi là nhiều hơn một giá trị (như trong Thuật toán 1). Trong công việc này, vì càng lớn V hoặc L lập lại tối đa là, số lượng kết hợp lớn hơn được tạo ra gây ra thời gian ngừng hoạt động suy luận, chúng tôi chỉ chọn $V=2$ (tức là chúng tôi chỉ xem xét 2 giá trị có 2 xác suất cao nhất) và tiến hành thử nghiệm với L lập lại tối đa từ 1 đến 4 (tức là giá trị của mỗi $(n-L)$ lập lại tối đa) các chữ số khác là giá trị có xác suất cao nhất, chúng ta có $2L$ lập lại tối đa kết hợp). Cuối cùng, chúng tôi sắp xếp các kết hợp ứng cử viên theo thứ tự xác suất lớn hơn đến nhỏ hơn và lần lượt kiểm tra sự thỏa mãn của phương trình (1), dừng lại ở kết hợp thỏa mãn đầu tiên để suy luận nhanh. Quá trình này được mô tả rõ ràng trong Thuật toán 1.

Thuật toán 1: Thuật toán dự đoán sửa đổi (MPA)

Đầu vào :Đã đào tạo CNN Multidigit (MDCNN), Mã vạch Hình ảnh (BI), Lập lại tối đa (Tối đa), Trạng thái biểu quyết (Bỏ phiếu)

Đầu ra: Kết hợp số mã vạch / s (BDC) Tính logit [] [] sử dụng MDCNN cho BI; vì $K \leftarrow 1$ đến M làm

```
thăm dò  $k[] \leftarrow \text{softmax}(\text{logit}_k[])$ ;  
Sắp xếp giảm dần  $\text{Prob}_k[]$ ;  
 $\text{khác biệt}_k \leftarrow \text{Prob}_k[0].\text{val} - \text{Prob}_k[1].\text{val}$ ;  
 $\text{chữ số}_k \leftarrow \text{Prob}_k[0].\text{mục lục}$ ;  
nối thêm  $\text{khoảng cách}_k = \text{khác biệt}_k \times N$  và  $\text{Chức vụ}_k = K$  đến  
danh sách khoảng cách;
```

chấm dứt

Sắp xếp tăng dần $\text{danh sách khoảng cách}$ với phần tử khoảng cách ; sự kết hợp ban đầu ← sự kết hợp mới với chữ số_k ; nối thêm sự kết hợp ban đầu đến danh sách kết hợp ; khởi tạo iter bằng không vì $\text{mỗi khoảng cách} \in \text{danh sách khoảng cách}$ làm

tăng iter bởi một;
nếu iter lớn hơn Max sau đó

```
nếu  $\text{biểu quyết}$  sau đó  
| trở về  $\text{kết hợp biểu quyết}$ ;  
 $\text{khác}$   
| trở về vô giá trị;  
chấm dứt
```

chấm dứt

$K \leftarrow \text{khoảng cách.position}$;
 $\text{chữ số mới} \leftarrow \text{Prob}_k[1].\text{mục lục}$;
vì $\text{mỗi sự kết hợp} \in \text{danh sách kết hợp}$ làm
sự kết hợp mới ← biến đổi sự kết hợp tại
Chức vụ K với chữ số mới ; nối
thêm sự kết hợp mới đến
 $\text{danh sách kết hợp mới}$;

chấm dứt

vì $\text{mỗi sự kết hợp} \in \text{danh sách kết hợp mới}$ làm

$\text{trạng thái} \leftarrow$ Tính toán kiểm tra tổng kiểm tra cho
 sự kết hợp ;

nếu trạng thái sau đó

```
nếu  $\text{biểu quyết}$  sau đó  
| nối thêm  $\text{sự kết hợp}$  đến  
|  $\text{kết hợp biểu quyết}$ ;  
 $\text{khác}$   
| trở về  $\text{sự kết hợp}$ ;  
chấm dứt
```

chấm dứt

chấm dứt

chấm dứt

trở về vô giá trị

Thuật toán 2:MPA với tăng cường

Đầu vào :Đã đào tạo CNN Multidigit (MDCNN), Mã vạch Hình ảnh (BI), Lập lại tối đa (Tối đa) Đầu ra: Kết hợp chữ số mã vạch (BDC) danh sách bằng cấp←nối thêm độ [90,180,270]; danh sách hình ảnh←xoay hình ảnh BI vớidanh sách bằng cấp; —

vìmỗihình ảnh∈danh sách hình ảnhlàm

sự kết hợp←Dự đoán bằng cách sử dụngThuật toán 1

MDCNN đã cho,hình ảnh, và biểu quyết = Sai; nếu

sự kết hợpkhông rỗngsau đó

trở vềsự kết hợp;

chấm dứt

trở vềvô giá trị

Thuật toán 3:KBTB với Tăng cường và Bỏ phiếu

Đầu vào :Đã đào tạo CNN Multidigit (MDCNN), Mã vạch Hình ảnh (BI), Lập lại tối đa (Tối đa) Đầu ra: Kết hợp chữ số mã vạch (BDC) danh sách bằng cấp←nối thêm độ [90,180,270]; danh sách hình ảnh←xoay hình ảnh BI vớidanh sách bằng cấp; —

vìmỗiHình ảnh∈Danh sách hình ảnhlàm

sự kết hợp←Dự đoán vớiThuật toán 1

MDCNN đã cho,hình ảnh, và biểu quyết = Đúng;

nếusự kết hợpkhông rỗngsau đó

nối thêmsự kết hợpđến danh sách kết hợp;

chấm dứt

nếudanh sách kết hợpkhông có sản phẩm nàosau đó

các kết hợp nhóm tương tự nhau; sự kết hợp

←chọn kết hợp với cao nhất

đếm;

trở vềsự kết hợp;

khác

trở vềvô giá trị;

chấm dứt

Thuật toán 1 được cải tiến bằng cách áp dụng tăng thời gian thử nghiệm theo 2 cách: theo dõi nhanh như trong Thuật toán 2 và bỏ phiếu như Thuật toán 3. Để đơn giản và suy luận nhanh, điều quan trọng trong ứng dụng này, chúng tôi chỉ sử dụng 3 thao tác xoay để tăng thêm mỗi đầu vào hình ảnh. Thuật toán 2 lặp lại thông qua đầu vào ban đầu và 3 biến thể của nó gọi từng bước Thuật toán 1 và dừng ngay khi Thuật toán 1 nhận được phương trình đầu tiên thỏa mãn tổ hợp, nếu không, không có chuỗi đã giải mã nào được trả về. Mặt khác, Thuật toán 3 thu thập các kết hợp thỏa mãn từ tất cả các lần lặp (đầu vào và biến thể ban đầu) và chọn các kết hợp thường xuyên nhất.

Một điều chúng ta cần nhấn mạnh ý tưởng này so với [14] là các kỹ thuật được đề xuất của chúng tôi có thể mở rộng hơn cho các loại mã vạch 1D khác (họ mã vạch EAN, UPC, ITF) với một vài thay đổi. Kỹ thuật này có thể áp dụng cho nhiều loại mã vạch trong một mô hình, chúng ta chỉ cần thêm một vài nút nữa, một số để phân loại các loại mã vạch, một số để điền vào

độ dài của các loại mã vạch dài nhất (mỗi chữ số hiện có 11 giá trị, 0-9 và NA), phương trình 1 vẫn áp dụng cho tất cả các mã EAN, UPC khác.

B. Giảm thiểu mô hình sâu

Để giảm thiểu các mô hình sâu để có một mô hình phù hợp hơn cho các thiết bị cạnh, chúng tôi sử dụng kỹ thuật chất lọc kiến thức ban đầu trong [22] để chất lọc kiến thức từ mô hình tốt nhất (sâu) đến các mô hình nông nhỏ bằng cách thay thế hàm tổn thất ban đầu bằng tổn hao kết hợp:

$$L = (1 - \alpha) * L_H + \alpha * L_{KL}$$

Ở đây L_H là sự mất mát entropy chéo từ các nhãn cứng, L_{KL} là tổn thất phân kỳ Kullback-Leibler từ các nhãn giáo viên (nhãn mềm) và α là siêu tham số.

IV. EXÁC NHẬN

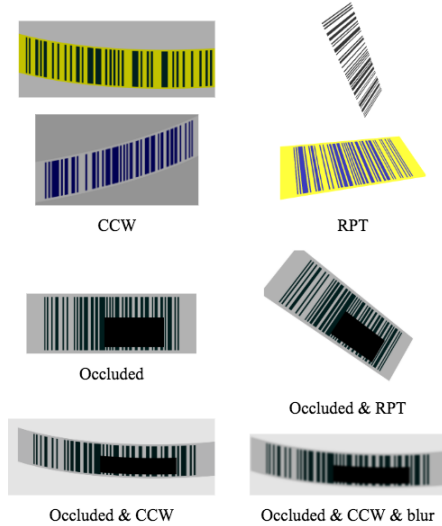
A. Tập dữ liệu

Bộ sưu tập thực sự của chúng tôi bao gồm 1055 mẫu từ mở rộng *MuensterDB*, 408 mẫu từ [12] và 1037 do chúng tôi tự thu thập. Tổng cộng chúng tôi có 2500 mẫu sau khi chúng tôi vẽ hộp giới hạn, gắn nhãn các trình tự đã được giải mã ([4] và [12] chưa hoàn thành cả hai nhiệm vụ trong bộ dữ liệu của chúng). Các mẫu tự thu thập của chúng tôi được lấy từ 5 siêu thị (1 ở Pháp, 2 ở Hàn Quốc, 2 ở Việt Nam) cả trong nhà và ngoài trời trong 2 tuần. Một loạt các sản phẩm từ thực phẩm và gói sản phẩm ăn được, sách, đồ dùng nhà bếp, văn phòng phẩm, thẻ quần áo trên nhiều chất liệu khác nhau như lon kim loại, chai rượu, túi nhựa đựng thực phẩm, hộp các tông dưới nhiều nguồn sáng khác nhau (đèn huỳnh quang, bóng đèn sợi đốt, buổi sáng và ánh sáng mặt trời buổi chiều) và các điều kiện (tự động tắt lấy nét, bắt tay, khoảng cách xa, bị che khuất bởi các ngón tay, nhãn nếp, méo mó, dồn vào đường cùng); cũng có 195 mã vạch được in trên giấy thường với các điều kiện bị tắc và nhãn. resl.kaist.ac.kr/doc/datasets

BẢNG I
SỐ ĐIỀU KIỆN ĐƯỢC HÓA

Các điều kiện)	Số lượng mẫu
đỉnh mức	30000
tối	30000
bị tắc nghiêm	20000
tắc + tối	20000
đã xoay & chuyển đổi phối cảnh (RPT)	20000
RPT + tối	20000
hình trụ & cong vênh (CCW)	20000
+ tối	20000
bị tắc + RPT	5000
mơ hồ	5000
RPT + mờ	5000
CCW + mờ	5000
lộn ngược	6000
lộn ngược + lộn ngược	6000
tối + lộn ngược mờ +	6000
lộn ngược CCW +	6000
nhiều nặng + xoay	6000
	2000
thừa sáng + bị che khuất + RPT + CCW	6000
tối + bị che khuất + RPT + CCW bị che	6000
khuất + RPT + CCW	6000

Bộ đào tạo của chúng tôi bao gồm 250000 mẫu tổng hợp (không có văn bản được giải mã dưới các sọc) với điều kiện được mô tả trong Bảng I (lưu ý rằng 40000 mẫu là nhiễu được thêm vào ngẫu nhiên) và 20000 mẫu được tăng cường từ 500 mẫu thực được chọn ngẫu nhiên từ bộ thực được thu thập. Một số mẫu tổng hợp được thể hiện trong Hình 3. 2000 mẫu còn lại của bộ thu thập thực được sử dụng làm bộ thử nghiệm.



Hình 3. Một số mẫu tổng hợp

B. Thiết lập Thử nghiệm

Để chứng minh sự cải thiện hiệu suất của mô hình được đề xuất của chúng tôi, chúng tôi đã thực hiện các thử nghiệm về các giải pháp doanh nghiệp và mô hình cơ sở với học sâu. Zxing [13] là một công cụ mã nguồn mở được hầu hết các nhà phát triển sử dụng, trong khi Google Barcode API là phiên bản thương mại hóa của Zxing. Mặt khác, Cognex và Dynamsoft là hai tập đoàn lớn có lịch sử lâu đời trong việc phát triển các sản phẩm sử dụng thị giác máy dùng trong công nghiệp. Đối với 4 công cụ này, để đưa ra đánh giá công bằng, tất cả chúng tôi đều áp dụng cách tăng thời gian thử nghiệm giống như cách với các mô hình học sâu của chúng tôi trong Thuật toán 2 và đặt chúng hoạt động cụ thể cho EAN13. Lưu ý rằng chúng tôi đã sử dụng nguồn Zxing và API Google phiên bản mới nhất, API dựa trên web demo của Dynamsoft và Cognex được sử dụng để đánh giá, vì vậy chúng tôi cũng suy ra thời lượng thông báo khứ hồi trong việc đo lường thời gian suy luận.

Về mô hình mạng nơon sâu, mô hình tương tự Fridborn tương tự như những gì được mô tả trong công việc của họ [5], (vì đầu vào của chúng là 196x100x1 trong khi của chúng tôi là 285x285x3, do đó 4 khối phức hợp dẫn đến vượt quá tài nguyên GPU) những gì chúng tôi phải thay đổi là 32 hạt nhân thay vì 256 hạt nhân cho lớp chập cuối cùng và 2048 nút thay vì 4096 nút cho mỗi lớp trong số 2 lớp FC trên cùng). Tiếp theo, chúng tôi lập mô hình không theo chu kỳ với 8 khối chập và 2 lớp FC có ít tham số hơn so với mô hình tương tự Fridborn. Các mô hình khác sử dụng trích xuất tính năng SOTA CNN như

ResNet50, ResNet34 [18], MobileNetV2 [24], DenseNet169 [25] chỉ có các bộ phận trích xuất tính năng ban đầu trực tiếp trước 13 nút đầu ra như trong Hình 2. Các kích thước lô khác nhau đã được thử nhưng theo quan sát thực nghiệm của chúng tôi, 32 có thể là con số tốt nhất. Tất cả các mô hình đều được đào tạo từ đầu mà không cần đào tạo trước kiến trúc. Lưu ý rằng chúng tôi phải đào tạo các mô hình chỉ với bộ tổng hợp trước cho đến khi tổn thất giảm xuống còn khoảng 1 (tức là các mô hình hội tụ đến một mức nhất định) trước khi đào tạo trên bộ đào tạo đầy đủ (với 20000 mẫu tăng cường được thu thập thực tế) vì đào tạo trực tiếp trên bộ đào tạo đầy đủ dẫn đến thất thoát rất cao (thậm chí cả NaN). Quá trình đào tạo được thực hiện bằng NVIDIA Titan RTX với 24 GB VRAM. Các thí nghiệm đánh giá CPU của chúng tôi được thực hiện trên máy tính để bàn sử dụng bộ xử lý Intel Core i9 9900KF,

C. Đánh giá

Chúng tôi có 2 số liệu cần làm rõ ở đây: độ chính xác và sai số. Về cơ bản, một công cụ sẽ có 3 trạng thái kết quả cho hình ảnh mã vạch: trình tự được giải mã chính xác (tức là khớp với sự thật cơ bản), trình tự được giải mã không chính xác và không tồn tại mã vạch (hoặc không có trình tự thỏa mãn tổng kiểm tra cho các mô hình được đề xuất của chúng tôi). Chỉ số độ chính xác trong công việc này được tính bằng

$$\text{Sự chính xác} = \frac{\# \text{ trong số các trình tự được giải mã chính xác}}{\# \text{ trong tổng số hình ảnh mã vạch}}$$

trong khi số lỗi = # of the incorrects. Điều này có nghĩa là một mô hình tốt là một mô hình đạt được độ chính xác cao hơn và ít lỗi hơn. Một con số khác cần được đề cập trong phần này là thời gian suy luận là thời gian suy luận trung bình trên một hình ảnh vì mỗi hình ảnh mất một khoảng thời gian khác nhau cho bộ giải mã.

BẢNG II
TOOL&MÔ HÌNH KHÔNG CÓ KẾT QUẢ HIỆU SUẤT

Người mẫu	Sự chính xác	CPU (mili giây)	# trong số các tham số (M)
Zxing	58,25%	7.65	NA
Dynamsoft	93,10%	978,8	NA
API Google	82,45%	211,9	NA
Cognex	84,60%	111,9	NA
ResNet50	93,35%	66,5	99,5
MobileNetV2	72,25%	32,4	15,7
MobileNetV2_kd	83,45%	32,4	15,7
DenseNet169	84,90%	75,65	30
ResNet34	88,70%	38,3	40,7
ResNet34_kd	89,20%	38,3	40,7
Fridborn-tương tự	31,85%	104,9	403,3
Không dư	80,80%	103,2	78,5

Về thời gian suy luận, chúng tôi cần lưu ý rằng khó có thể có đánh giá hoàn hảo tốt vì Dynamsoft, Google API và Cognex đã được kiểm tra thông qua các API chạy trên máy chủ của chính chúng không khớp với máy tính để bàn đã định cấu hình của chúng tôi. Vì một số công cụ không sử dụng máy học (ngoại trừ Dynamsoft, Cognex sử dụng DNN trên nhiều sản phẩm khác của họ nên chúng cũng có thể là mô hình DNN), thời gian xử lý của chúng tương đối nhỏ hơn so với các kỹ thuật dựa trên học sâu nhưng với

BẢNG III
UHÁTKBTBHIỆU SUẤT

Người mẫu	nonMPA	tối đa = 1	tối đa = 2	tối đa = 3	tối đa = 4
ResNet50	0,9335	0,942	0,9435	0,9445	0,9445
MobileNetV2	0,8345	0,8595	0,869	0,868	0,8645
DenseNet169	0,849	0,8645	0,876	0,8775	0,874
ResNet34	0,892	0,9075	0,911	0,912	0,911
Không dư	0,808	0,8295	0,844	0,8455	0,841
ResNet50	133	31	48	77	106
MobileNetV2	331	59	106	164	241
DenseNet169	302	54	105	176	230
ResNet34	216	41	73	116	157
Không dư	384	55	104	197	285

BẢNG IV
UHÁTKBTB & HIỆU SUẤT THI CÔNG

Người mẫu	nonMPA	tối đa = 1	tối đa = 2	tối đa = 3	tối đa = 4
ResNet50	0,9335	0,958	0,956	0,951	0,946
MobileNetV2	0,8345	0,906	0,8975	0,8855	0,8705
DenseNet169	0,849	0,9155	0,9075	0,8915	0,877
ResNet34	0,892	0,9375	0,9315	0,9235	0,916
Không dư	0,808	0,89	0,879	0,861	0,8445
ResNet50	133	56	73	95	108
MobileNetV2	331	121	182	225	259
DenseNet169	302	113	172	216	246
ResNet34	216	95	129	152	168
Không dư	384	145	212	274	311

BẢNG V
UHÁTKBTB & AUGMENTATION VỚI HIỆU SUẤT CUỐI CÙNG

Người mẫu	nonMPA	tối đa = 1	tối đa = 2	tối đa = 3	tối đa = 4
ResNet50	0,9335	0,9585	0,9585	0,9525	0,95
MobileNetV2	0,8345	0,9085	0,906	0,899	0,8945
DenseNet169	0,849	0,9125	0,8985	0,8785	0,8545
ResNet34	0,892	0,933	0,93	0,9215	0,911
Không dư	0,808	0,8855	0,8735	0,853	0,8355
ResNet50	133	55	68	92	100
MobileNetV2	331	116	165	198	211
DenseNet169	302	119	190	242	291
ResNet34	216	104	132	156	178
Không dư	384	154	223	290	329

độ chính xác của dự đoán thấp. Đánh giá cơ bản được trình bày trong Bảng II. Các mô hình cơ sở của chúng tôi tốt hơn các mô hình khác với thời gian tính toán hợp lý để dự đoán với độ chính xác hơn 0,93.

Để cho thấy hiệu suất đạt được khi áp dụng Suy luận thông minh trong thời gian thử nghiệm, chúng tôi đã thực hiện ba thử nghiệm khác nhau. Thử nghiệm đầu tiên được thực hiện với Thuật toán 1. Như được mô tả trong Bảng III, kết quả cho thấy hiệu suất được cải thiện so với các mô hình không có nó. Kết quả cũng cho thấy rằng hiệu suất được cải thiện khi số lượng khoảng trống được xem xét tăng lên đến một mức nào đó, sau đó nó cho thấy sự suy thoái. Thử nghiệm thứ hai được thực hiện để chứng minh cách (Thuật toán 2) tăng cường theo dõi nhanh có thể cải thiện hiệu suất của KBTB. Nó cho thấy rõ ràng sự cải thiện hiệu suất so với Thuật toán 1 như được mô tả trong Bảng IV và những cải tiến đáng kể so với cách tiếp cận cơ bản. Tuy nhiên, lần này, chỉ sau khi xem xét một cặp có khoảng cách nhỏ nhất, chúng tôi đã đạt được kết quả tốt nhất. Giống như MPA không tăng (Thuật toán 1),

2 tăng khi số lượng khoảng cách đang xét tăng lên. Tuy nhiên, Thuật toán 1 vẫn cho thấy số lỗi ít hơn khi so sánh với các dự đoán với Thuật toán 2. Thử nghiệm thứ ba tương ứng với Thuật toán 3 được thực hiện với chi phí cao hơn. Như được thể hiện trong bảng V. Đôi khi nó hoạt động tốt hơn một chút so với các thử nghiệm dựa trên Thuật toán 2 với hành vi tương tự khi chúng ta thay đổi tham số *Max*. Đây

cho thấy rằng kịch bản bỏ phiếu không phải lúc nào cũng là một lựa chọn tốt cho các mô hình của chúng tôi và các mô hình có thể đã tương đối mạnh mẽ với hình ảnh đầu vào ban đầu và chỉ cần trợ giúp sau họ đã thất bại ngay từ đầu.

Như chúng tôi đã đề cập trong phần trước, để chứng minh khả năng của giải pháp trên các thiết bị di động, chúng tôi đã chốt lọc kiến thức từ mô hình tốt nhất (ResNet50) đến 2 mô hình nhỏ được coi là: ResNet34 và MobileNetV2. Kết quả trong Bảng II cho thấy rõ ràng rằng việc chốt lọc kiến thức giúp đạt được hiệu suất cao hơn so với việc luyện tập bằng hàm mất mát thông thường. MobileNetV2 nhảy vọt đáng kể từ 72,25% đến 83,45% trong khi sự cải tiến trong mô hình ResNet34 không nhiều. Điều này có thể là do ResNet34 vẫn còn sâu (40,7 triệu thông số so với 15,7 triệu thông số của MobileNetV2) và vì vậy khả năng học hỏi của họ từ chính nó vẫn là đủ mạnh để đạt được hiệu suất cao mà không cần sự hướng dẫn từ người mẫu của giáo viên. Cuối cùng, các thử nghiệm thử nghiệm của chúng tôi trên bo mạch NVIDIA Jetson Nano cho thấy rằng MobileNetV2, ResNet34 đạt được tốc độ trung bình là 34,2, 45,6 mili giây trên mỗi hình ảnh tương ứng. Tốc độ này tương đương với trải nghiệm khung hình / giây mượt mà với sự mạnh mẽ của mô hình, chúng tôi mong đợi nó là thoải mái cho người dùng.

V. KẾT LUẬN

Trong công việc này, chúng tôi đã đề xuất Suy luận Thông minh cho các mô hình dựa trên Multidigit CNN để cải thiện hiệu suất giải mã mã vạch 1D. Chúng tôi đã thu thập nhiều mã vạch thực với dữ liệu nhãn để đào tạo và kiểm tra mô hình được đề xuất. Chúng tôi cũng đã bổ sung dữ liệu tổng hợp tốt hơn để tăng cường quá trình đào tạo và kiểm tra. Các thuật toán được đề xuất trong thời gian thử nghiệm đã tăng hiệu suất so với các mô hình cơ sở. Nó không chỉ vượt trội hơn so với mô hình cơ sở về độ chính xác mà còn có thời gian suy luận nhỏ giúp nó hoạt động hiệu quả. Cách tiếp cận dựa trên Multidigit CNN với Suy luận thông minh cũng là một giải pháp có thể mở rộng vì nó có thể mở rộng để giải mã nhiều hơn một loại mã vạch.

Mặc dù hiệu suất tốt hơn về độ chính xác, nhưng mô hình đề xuất có một hạn chế trong việc dự đoán các bản ghi sai (Dynamsoft cũng dự đoán 3 lỗi). Một hạn chế khác của phương pháp này là nó không thể áp dụng cho các loại mã vạch có độ dài không cố định như Code39. Trong tương lai, vấn đề dự đoán sai có thể được giảm thiểu bằng cách áp dụng các kỹ thuật nhận dạng sản phẩm.

VI. Một THÔNG BÁO

Công trình này được hỗ trợ và tài trợ bởi Bộ Khoa học và CNTT-TT (MSIT) thuộc Dự án Hỗ trợ Nghiên cứu Chung Hàn Quốc-EU của Quỹ Nghiên cứu Quốc gia Hàn Quốc (NRF-2016K1A3A7A0395205414), Chương trình Nghiên cứu Chính (E0162502) của Viện Nghiên cứu Thực phẩm Hàn Quốc (KFRI), và chương trình hỗ trợ của Trung tâm Nghiên cứu Công nghệ Thông tin Grand (IITP-2020-0-01489) do IITP (Viện Quy hoạch & Đánh giá Công nghệ Thông tin & Truyền thông) giám sát.

RTIỀN TỆ

- [1] WP Fernandcz, Y. Xian và Y. Tian, "Nhận dạng và phát hiện mã vạch dựa trên hình ảnh để hỗ trợ người khiếm thị," trong *Hội nghị quốc tế thường niên lần thứ 7 IEEE 2017 về Công nghệ CYBER trong Hệ thống tự động hóa, điều khiển và thông minh (CYBER)*. IEEE, 2017, trang 1241–1245.
- [2] E. Joseph và T. Pavlidis, "Nhận dạng dạng sóng mã vạch bằng cách sử dụng các vị trí cao nhất," *Giao dịch IEEE trên Phân tích mẫu và Trí tuệ máy*, quyển sách. 16, không. 6, trang 630–640, 1994.
- [3] R. Muniz, L. Junco và A. Otero, "Một máy đọc mã vạch phần mềm mạnh mẽ sử dụng biến đổi hough," trong *Kỷ yếu 1999 Hội nghị Quốc tế về Hệ thống và Trí tuệ Thông tin (Cat. Số PR00446)*. IEEE, 1999, trang 313–319.
- [4] S. Wachenfeld, S. Terlunen và X. Jiang, "Nhận dạng mạnh mẽ mã vạch 1-d bằng cách sử dụng điện thoại máy ảnh," trong *Hội nghị quốc tế lần thứ 19 về nhận dạng mẫu năm 2008*. IEEE, 2008, trang 1–4.
- [5] F. Fridborn, "Đọc mã vạch bằng mạng nơ-ron", 2017. [Trực tuyến]. Có sẵn: <http://liu.diva-portal.org/smash/record.jsf?pid=diva2:1164104>
- [6] D.-T. Lin, M.-C. Lin và K.-Y. Huang, "Nhận dạng tự động theo thời gian thực của nhiều mã vạch đa hướng và triển khai dsp," *Thị giác máy và ứng dụng*, quyển sách. 22, không. 2, trang 409–419, 2011.
- [7] M. Katona và LG Nyúl, "Một phương pháp mới để phát hiện mã vạch chính xác và hiệu quả với các phép toán hình thái học," trong *Hội nghị quốc tế lần thứ tám về Công nghệ hình ảnh tín hiệu và các hệ thống dựa trên Internet*. IEEE, 2012, trang 307–314.
- [8] G. Sörös và C. Flörkemeier, "Bản địa hóa mã vạch 1d và 2d chống mờ cho điện thoại thông minh", trong *Kỷ yếu Hội nghị Quốc tế lần thứ 12 về Di động và Đa phương tiện phổ biến*, 2013, trang 1–8.
- [9] C. Creusot và A. Munawar, "Máy dò mã vạch tập trung tính toán thấp cho người mù," trong *Hội nghị quốc tế IEEE 2016 về xử lý hình ảnh (ICIP)*. IEEE, 2016, trang 2856–2860.
- [10] —, "Phát hiện mã vạch trong thời gian thực trong tự nhiên", trong *Hội nghị mùa đông IEEE 2015 về các ứng dụng của thị giác máy tính*. IEEE, 2015, trang 239–245.
- [11] DK Hansen, K. Nasrollahi, CB Rasmussen và TB Moeslund, "Phát hiện và phân loại mã vạch trong thời gian thực bằng cách sử dụng học sâu". Trong *IJCCI*, 2017, trang 321–327.
- [12] A. Zamberletti, I. Gallo, M. Carullo và E. Binaghi, "Giải mã mã vạch 1-d từ hình ảnh bị giảm chất lượng bằng cách sử dụng mạng nơ-ron," trong *Hội nghị quốc tế về thị giác máy tính, hình ảnh và đồ họa máy tính*. Springer, 2010, trang 45–55.
- [13] S. Owen et al., "Zxing," *Vạch kẻ đường cho người đi bộ*, 2013.
- [14] H. Yang, L. Chen, Y. Chen, Y. Lee, và Z. Yin, "Phương pháp nhận dạng mã vạch tự động dựa trên phát hiện cạnh thích ứng và mô hình ảnh xạ," *Tạp chí Hình ảnh Điện tử*, quyển sách. 25, không. 5, tr. 053019, năm 2016.
- [15] IJ Goodfellow, Y. Bulatov, J. Ibarz, S. Arnaud và V. Shet, "Nhận dạng số đa chữ số từ hình ảnh ở chế độ xem phổ bằng cách sử dụng mạng nơ-ron phức hợp sâu" *arXiv bản in trước arXiv: 1312.6082*, 2013.
- [16] C. Shorten và TM Khoshgoftaar, "Một cuộc khảo sát về tăng cường dữ liệu hình ảnh cho học sâu", *Tạp chí Dữ liệu lớn*, quyển sách. 6, không. 1, tr. 60, 2019.
- [17] A. Krizhevsky, I. Sutskever, và GE Hinton, "Phân loại mạng hình ảnh với mạng nơ-ron phức hợp sâu", trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, 2012, trang 1097–1105.
- [18] K. He, X. Zhang, S. Ren, và J. Sun, "Học sâu còn sót lại để nhận dạng hình ảnh," trong *Kỷ yếu hội nghị IEEE về thị giác máy tính và nhận dạng mẫu*, 2016, trang 770–778.
- [19] ED Cubuk, B. Zoph, D. Mane, V. Vasudevan và QV Le, "Autoaugment: Học các chiến lược gia tăng từ dữ liệu," trong *Kỷ yếu hội nghị IEEE về thị giác máy tính và nhận dạng mẫu*, 2019, trang 113–123.
- [20] J. Lemley, S. Bazrafkan và P. Corcoran, "Tăng cường thông minh học một chiến lược tăng dữ liệu tối ưu," *Ieee Access*, quyển sách. 5, trang 5858–5869, năm 2017.
- [21] Y. Cheng, D. Wang, P. Zhou, và T. Zhang, "Một cuộc khảo sát về nén và tăng tốc mô hình cho các mạng nơ-ron sâu," *arXiv bản in trước arXiv: 1710.09282*, Năm 2017.
- [22] G. Hinton, O. Vinyals và J. Dean, "Chưng cất kiến thức trong mạng nơ-ron" *arXiv bản in trước arXiv: 1503.02531*, 2015.
- [23] Anonymous, "Cách tính số séc theo cách thủ công - dịch vụ," tháng 12 năm 2014. [Trực tuyến]. Có sẵn: <https://www.gs1.org/services/how-tính-kiểm-tra-số-thủ-công>
- [24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, và L.-C. Chen, "Mobilenetv2: Phần dư ngược và tắc nghẽn tuyến tính," trong *Kỷ yếu hội nghị IEEE về thị giác máy tính và nhận dạng mẫu*, 2018, trang 4510–4520.
- [25] G. Huang, Z. Liu, L. Van Der Maaten và KQ Weinberger, "Các mạng phức hợp được kết nối dày đặc," trong *Kỷ yếu hội nghị IEEE về thị giác máy tính và nhận dạng mẫu*, 2017, trang 4700–4708.