VẬT LIỆU & XÂY DỰNG
Tạp chí Vật liệu và Xây dựng
ISSN 1859 - 381X
Website: www. jomc.vn

Mô hình học sâu phát hiện và nhận diện mã container áp dụng trong vận hành cảng thông minh

Mã Chí Hiếu[1,2*], Trần Quang Trường[1,2], Lê Tuấn Anh[1,2]

[1] Khoa Kỹ thuật Xây dựng, Trường Đại học Bách Khoa TP.HCM, Việt Nam

[2] Đại học Quốc Gia Thành phố Hồ Chí Minh, Việt Nam

TỪ KHOÁ

Yolov11
EasyOCR
mã code container
thị giác máy tính
cảng thông minh
nhận diện ký tự quang học (OCR)
phát hiện đối tượng

TÓM TẮT

Thị giác máy tính, một lĩnh vực quan trọng trong trí tuệ nhân tạo, đang ngày càng phát triển mạnh mẽ và được ứng dụng rộng rãi trong nhiều ngành công nghiệp. Dựa trên kiến trúc mạng nơ-ron tích chập (CNN), nhiều mô hình tiên tiến đã được xây dựng để giải quyết các vấn đề như phát hiện đối tượng, phân đoạn hình ảnh, nhận diện ký tự quang học (OCR)... Trong số đó, YOLO nổi bật với khả năng phát hiện đối tượng nhanh và chính xác; và EasyOCR là một công cụ hiệu quả trong nhận dạng ký tự với độ chính xác cao. Nghiên cứu hiện tại tập trung vào việc phát hiện và nhận diện mã thông qua sự kết hợp giữa mô hình YOLOv11 và EasyOCR. Nội dung nghiên cứu bao gồm xây dựng tập dữ liệu, huấn luyện mô hình và đánh giá hiệu suất của mô hình. Kết quả thực nghiệm cho thấy mô hình đề xuất đạt độ chính xác trên 90 %, chứng tỏ tính khả thi và tiềm năng ứng dụng trong các hệ thống thực tế trong các cảng thông mình.

KEYWORDS

Yolov11
EasyOCR
Container codes
Computer vision
Smart port
Optical character recognition
Object detection

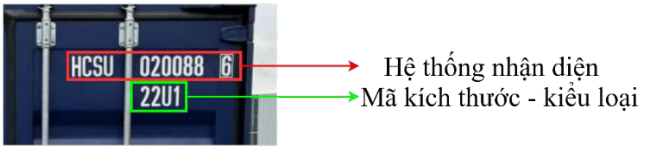
ABSTRACT

Computer vision, a key area within artificial intelligence, has been rapidly advancing and is increasingly applied across various industrial domains. Based on the architecture of Convolutional Neural Networks (CNNs), numerous state-of-the-art models have been developed to address a range of tasks, including object detection, image segmentation, and optical character recognition (OCR), etc. Among these, YOLO (You Only Look Once) stands out for its high-speed and accurate object detection capabilities, while EasyOCR has proven to be an effective tool, offering high character recognition accuracy. The present study focuses on the detection and recognition of container codes by integrating the YOLOv11 model with EasyOCR. The research encompasses the construction of a training dataset, model training, and model performance evaluation. Output results indicate that the proposed model achieves an accuracy of over 90%, demonstrating its feasibility and strong potential for real-world applications in the smart ports.

1. Giới thiệu

Hiện nay, mô hình cảng thông minh đang trở thành xu hướng phát triển chủ đạo của các cảng biển trên toàn cầu. Cảng thông minh được hiểu là một hệ thống vận hành tích hợp, trong đó các bên liên quan như khách hàng, cảng và các đơn vị logistics cùng tham gia vào quy trình thu thập, phân phối và vận chuyển hàng hóa thông qua việc ứng dụng các công nghệ hiện đại. Mục tiêu chính là tối ưu hóa việc sử dụng các nguồn lực. Để đạt được điều đó, cảng thông minh cần đáp ứng các yêu cầu như giám sát thông minh, cung cấp dịch vụ thông minh và khả năng xử lý tự động. Những yếu tố này giúp nâng cao mức độ an toàn, hiệu quả và chất lượng trong các dịch vụ logistics. Có thể khẳng định rằng, cảng thông minh được xây dựng trên nền tảng cơ sở hạ tầng hiện đại, với sự tích hợp của các công nghệ tiên tiến như mạng 5G, Internet vạn vật (IoT), dữ liệu lớn (Big Data), trí tuệ nhân tạo (AI), và công nghệ chuỗi khối (Blockchain), nhằm phù hợp với chức năng vận hành của cảng [1].

Hiện nay, việc vận chuyển hàng hóa bằng container đang được các công ty vận tải và thương mại sử dụng phổ biến. Container là những thùng thép tiêu chuẩn chuyên dụng, được gắn mã định danh bao gồm các ký tự và chữ số theo quy chuẩn quốc tế ISO 6346:2022 [2], hoặc tiêu chuẩn tương đương tại Việt Nam là TCVN 7623:2023 [3]. Mỗi mã container bao gồm hai phần chính: hệ thống nhận diện (Identification system) và mã kích thước – kiểu loại (Size and Type codes), được minh họa trong Hình 1.



Hình 1. Minh họa cho một mã container tiêu chuẩn.

Để theo dõi và quản lý các container một cách hiệu quả, cần có hệ thống nhận dạng mã container. Hiện có ba phương pháp chính: nhận

*Liên hệ tác giả: chihieuma@hcmut.edu.vn
Nhận ngày 22/05/2025, sửa xong ngày 12/06/2025, chấp nhận đăng ngày 13/06/2025
Link DOI: https://doi.org/10.54772/jomc.03.2025.995

diện thủ công, nhận diện bằng sóng vô tuyến (RFID – Radio Frequency Identification), hoặc sử dụng thị giác máy tính [4].

Mỗi phương pháp có những lợi thế và hạn chế riêng. Nhận dạng thủ công không yêu cầu hệ thống thiết bị phức tạp nhưng năng suất thấp và dễ xảy ra sai sót. Công nghệ RFID mang lại độ chính xác gần như tuyệt đối, tuy nhiên chi phí lắp đặt và bảo trì cao do yêu cầu gắn thẻ từ cho từng container. Hơn nữa, hệ thống này vẫn chưa được triển khai đồng bộ trên toàn cầu. Thị giác máy tính là một giải pháp kinh tế hơn, song gặp phải nhiều thách thức về độ chính xác và tốc độ nhận diện do chịu ảnh hưởng từ các yếu tố như điều kiện ánh sáng, góc chụp, độ mờ hình ảnh, kích thước và kiểu chữ của mã container.

Việc trích xuất và phân tích dữ liệu từ hình ảnh chứa mã số container đòi hỏi sự hỗ trợ của các thuật toán xử lý phức tạp và cần được tối ưu để đảm bảo độ tin cậy và độ chính xác cao trong nhận diện.

Mặc dù đã có nhiều tiến bộ đáng kể trong công nghệ nhận diện hình ảnh nhưng đối với vấn đề nhận diện mã container vẫn là một thách thức khá lớn đối với các hệ thống quan sát và kiểm soát trong các cảng tự động. Bài báo này tập trung nghiên cứu và phát triển hệ thống nhận diện mã container tự động dựa theo kiến trúc học sâu mới đó là YOLOv11 [5] và EasyOCR [6]. Trong bài báo hiện tại, mô hình YOLOv11 dùng để phát hiện và khoanh vùng vị trí của mã container. Sau đó, mô hình EasyOCR [6] được sử dụng để trích xuất các ký tự có trong vùng vị trí đã được tìm ra trước đó. Kết quả cho thấy mô hình được đề xuất có khả năng ứng dụng rộng rãi và hiệu quả trong hệ thống vận hành của các cảng tự động.

2. Phương pháp

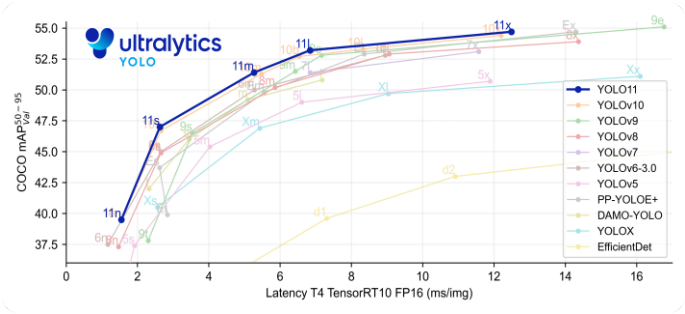
2.1. Mô hình đề xuất

Mô hình đề xuất được chia thành 2 giai đoạn : Nhận diện và Phát hiện. Sơ đồ khối của mô hình đề xuất được minh họa như Hình 3.

Hình 2. Sơ đồ khối mô hình đề xuất.

2.1.1. Giai đoạn 1: Phát hiện vị trí của mã container

YOLO (You Only Look Once) là một trong những kiến trúc phổ biến và được nhiều người biết đến trong lĩnh vực thị giác máy tính. YOLOv11 là YOLO phiên bản thứ 11 được ra mắt vào năm 2024 với độ chính xác, tốc độ xử lý và hiệu suất được cải thiện so với các phiên bản trước đó [5]. Hiệu năng của YOLOv11 so với các phiên bản trước được thể hiện ở Hình 3. Ở phiên bản này sử dụng kiến trúc được cải tiến ở phần Backbone và Neck, nhờ đó giúp tăng cường khả năng trích xuất đặc trưng để phát hiện đối tượng chính xác hơn và thực hiện được các tác vụ phức tạp với số lượng tham số ít hơn nên tốc độ xử lý cũng được cải thiện hơn các phiên bản trước.



Hình 3. Biểu đồ so sánh hiện năng các phiên bản YOLO [5].

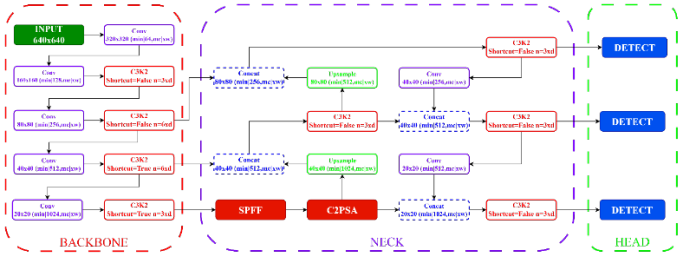
Trong giai đoạn này, dữ liệu đầu vào (ảnh hoặc video) sẽ được huấn luyện bằng mô hình YOLOv11 để phát hiện vị trí của mã container. Mô hình sẽ tạo ra một khung viền bao xung quanh mã container.

2.1.2. Giai đoạn 2: Nhận diện ký tự mã container

Trong giai đoạn này, hệ thống sử dụng mô hình EasyOCR để trích xuất các ký tự bên trong khung viền đã được xác định ở giai đoạn 1. Mã container được trích xuất có thể xuất ra với nhiều định dạng khác nhau như đè lên dữ liệu gốc hoặc định dạng tệp.

2.2. Mô hình YOLOv11

Trong nghiên cứu này, mô hình YOLOv11 được lựa chọn để thực hiện giai đoạn đầu tiên – phát hiện vị trí mã container trong ảnh hoặc video. Kiến trúc của YOLOv11 được tổ chức thành ba thành phần chính: Backbone, Neck, và Head. Mỗi thành phần đảm nhiệm một vai trò cụ thể trong quá trình trích xuất đặc trưng và phát hiện đối tượng. Kiến trúc tổng quan của mô hình YOLOv11 được minh họa trong Hình 4.



Hình 4. Kiến trúc mô hình YOLOv11 [7].

2.2.1 Backbone

Backbone chịu trách nhiệm trích xuất các đặc trưng từ hình ảnh đầu vào thông qua một chuỗi các tầng tích chập [7]. Dữ liệu đầu vào là hình ảnh có độ phân giải 640x640 pixel với 3 kênh màu (RGB).

Các lớp tích chập (Conv) chịu trách nhiệm giảm kích thước ảnh từ 640x640x3 xuống còn 320x320x64, 160x160x128, 80x80x265, 40x40x512 và 20x20x1024.

Khối C3k2 là một điểm cải tiến mới trong kiến trúc của YOLOv11

so với các phiên bản trước đó, được thiết kế để tối ưu việc trích xuất các đặc trưng của ảnh ở các mức độ phân giải 80x80, 40x40 và 20x20.

Backbone của YOLOv11 tạo ra các đặc trưng đa tỷ lệ tại ba kích thước chính là 20x20, 40x40 và 80x80, và sau đó chuyển tiếp chúng sang phần Neck để tiếp tục xử lý và tổng hợp thông tin.

2.2.2 Neck

Thành phần Neck trong kiến trúc YOLOv11 có nhiệm vụ tổng hợp và kết hợp các đặc trưng được trích xuất từ nhiều tầng khác nhau của Backbone, đặc biệt là các đặc trưng ở nhiều mức độ phân giải. Việc tích hợp thông tin theo cách này giúp tăng cường khả năng phát hiện các đối tượng có kích thước đa dạng trước khi chuyển sang giai đoạn dự đoán tại phần Head [7].

Neck của YOLOv11 sử dụng một số thành phần quan trọng như sau:

- SPPF (Spatial Pyramid Pooling – Fast): Là kỹ thuật gộp đặc trưng theo nhiều tỷ lệ khác nhau, cho phép mô hình thu nhận thông tin đa tỷ lệ một cách hiệu quả. Đây là một trong những cải tiến tiêu biểu của các phiên bản YOLO gần đây, giúp đạt được sự cân bằng giữa độ chính xác và tốc độ xử lý.
- C2PSA (Convolutional Block with Parallel Spatial Attention): Là khối tích chập được tích hợp cơ chế chú ý không gian song song, giúp tăng cường khả năng nhận biết đặc trưng không gian và toàn cục. Thành phần này góp phần cải thiện độ chính xác của mô hình trong việc phát hiện các đối tượng, kể cả trong điều kiện phức tạp. Đây là bước tiến đáng kể so với các phiên bản trước, đặc biệt phù hợp với các ứng dụng thị giác máy tính thời gian thực nhờ vào hiệu quả tính toán cao.
- Upsample: Có chức năng tăng độ phân giải không gian của các bản đồ đặc trưng ở mức thấp, nhằm tránh bỏ sót các vật thể nhỏ và cải thiện khả năng nhận diện chi tiết.
- Concat (Concatenation): Thực hiện việc nối các bản đồ đặc trưng được trích xuất từ Backbone với các bản đồ đặc trưng đã được tăng độ phân giải từ khối Upsample. Sự kết hợp này giúp mô hình tận dụng đầy đủ thông tin ở cả cấp độ thấp và cao, từ đó nâng cao độ chính xác trong quá trình dự đoán.

2.2.3. Head

Phần Head trong kiến trúc YOLOv11 chịu trách nhiệm tạo ra các dự đoán đầu ra cuối cùng của mô hình. Thành phần này được chia thành hai nhánh chính:

- Nhánh định vị (Bounding Box): Dự đoán tọa độ và kích thước của các khung giới hạn bao quanh đối tượng trong ảnh.
- Nhánh phân loại (Class Prediction): Xác định nhãn lớp của đối tượng trong khung giới hạn, đồng thời gán kèm một giá trị độ tin cậy (confidence score) thể hiện mức độ chắc chắn của mô hình đối với dự đoán đó [7].

Cơ chế dự đoán hai nhánh cho phép mô hình vừa phát hiện vị trí của đối tượng, vừa nhận diện được bản chất của đối tượng đó, từ đó tối ưu hiệu quả trong các bài toán phát hiện và phân loại đối tượng đồng thời.

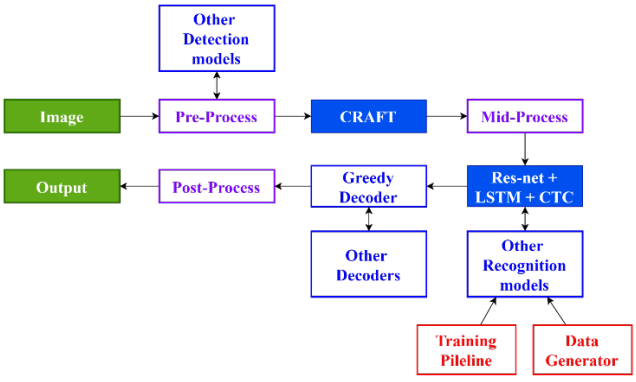
2.3 Mô hình EasyOCR

Trong giai đoạn nhận dạng ký tự, EasyOCR sử dụng thuật toán CRAFT (Character Region Awareness for Text Detection) để phát hiện vùng chứa ký tự trong ảnh. Đây là một phương pháp phát hiện văn bản hiệu quả, có khả năng xác định chính xác vị trí của từng ký tự, kể cả trong các bố cục phức tạp.

Phần nhận dạng ký tự được thực hiện thông qua mô hình CRNN (Convolutional Recurrent Neural Network), bao gồm ba thành phần chính:

- Trích xuất đặc trưng: Sử dụng các kiến trúc CNN nổi bật như ResNet và VGG để rút trích các đặc trưng hình ảnh đầu vào.
- Mã hóa chuỗi: Áp dụng mạng hồi tiếp LSTM (Long Short-Term Memory) để xử lý trình tự các đặc trưng, nhằm ghi nhận mối liên kết theo chiều ngang giữa các ký tự.
- Giải mã đầu ra: Sử dụng thuật toán CTC (Connectionist Temporal Classification) để chuyển đổi chuỗi đặc trưng thành chuỗi ký tự văn bản tương ứng.

Quy trình nhận diện trong EasyOCR là một phiên bản cải tiến từ các mô hình nhận diện văn bản sâu truyền thống [6]. Tổng quan kiến trúc của mô hình được minh họa trong Hình 5.



Hình 5. Kiến trúc mô hình EasyOCR [6].

3. Phương thức đánh giá

3.1. Phương thức đánh giá mô hình trong giai đoạn 1

Giai đoạn đầu tiên trong mô hình đề xuất là phát hiện đối tượng. Để xác định chất lượng của mô hình trong giai đoạn này, việc đánh giá hiệu năng là bước không thể thiếu. Việc đánh giá mô hình không chỉ giúp xác định mức độ chính xác mà còn hỗ trợ trong việc lựa chọn mô hình phù hợp nhất cho bài toán cụ thể.

Một trong những tiêu chí quan trọng để đánh giá hiệu năng các mô hình phát hiện đối tượng là mAP (mean Average Precision), tức là giá trị trung bình của chỉ số AP (Average Precision) trên tất cả các lớp đối tượng. Giá trị mAP càng cao chứng tỏ mô hình có khả năng phát hiện

đúng các đối tượng với sai số thấp, đảm bảo độ tin cậy trong quá trình nhận diện [8]. Giá trị mAP được tính toán như sau:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad (1)$$

trong đó AP là Average Precision; n là tổng số lớp. Để xác định giá trị AP, trước hết phải tính toán các giá trị Precision và Recall cho từng lớp. Sau đó, lấy diện tích bên dưới đường cong Precision-Recall cho từng đối tượng bằng công thức:

$$AP = \sum_{i=1}^{n}(R_i - R_{i-1}) \times P_i \quad (2)$$

trong đó P_i và R_i lần lượt là các giá trị Precision và Recall tại điểm thứ i trên đường cong P-R (Precision-Recall). Đường cong này bắt đầu tại (0,1) và kết thúc tại (1,0).

Giá trị của Precision và Recall tại mỗi điểm thứ i trên đường cong P-R được tính như sau:

$$Precision = \frac{TruePositive}{TruePositive+FalsePositive} \quad (3)$$

$$Recall = \frac{TruePositive}{TruePositive+FalseNegative} \quad (4)$$

Mặc dù Precision và Recall phản ánh tính chính xác của các dự đoán và khả năng bao quát của mô hình, chúng không phản ánh trực tiếp chất lượng định vị. Vì vậy, chỉ số IoU (Intersection over Union) được sử dụng để đánh giá độ chính xác của khung dự đoán thông qua việc đo lường sự chồng chéo giữa các khung viền dự đoán và khung viền thực tế:

$$IoU = \frac{Area of Inter sec tion}{Area of Union} \quad (5)$$

Giá trị IoU nằm trong khoảng từ 0 đến 1; IoU càng cao đồng nghĩa với khả năng định vị càng chính xác. Một dự đoán chỉ được xem là True Positive nếu IoU vượt qua ngưỡng xác định trước (ví dụ: 0,5).

Để có cái nhìn toàn diện hơn, có hai cách đánh giá phổ biến:

- mAP@50: Tính mAP tại ngưỡng IoU = 0,50, thường được sử dụng như một chuẩn cơ bản.
- mAP@50:95: Tính trung bình mAP trên các ngưỡng IoU từ 0,50 đến 0,95 với bước nhảy 0,05, cung cấp đánh giá khắt khe hơn và phản ánh toàn diện hiệu suất mô hình ở nhiều mức độ chính xác định vị khác nhau [8].

3.2. Phương thức đánh giá mô hình trong giai đoạn 2

Để đánh giá hiệu quả của mô hình trong việc nhận dạng ký tự, văn bản được trích xuất bởi mô hình sẽ được so sánh với văn bản gốc thông qua Khoảng cách Levenshtein, còn được gọi là Khoảng cách chỉnh sửa [9].

Chỉ số này đo lường số lượng thao tác tối thiểu cần thực hiện để chuyển đổi chuỗi ký tự dự đoán thành chuỗi ký tự đúng, bao gồm: thêm, xóa hoặc thay thế một ký tự. Giá trị càng nhỏ đồng nghĩa với độ tương đồng càng cao giữa văn bản nhận dạng và văn bản thực tế. Mức đánh giá cụ thể dựa trên chỉ số này được trình bày trong Bảng 1.

Thông thường, Khoảng cách Levenshtein được tính cho từng từ bằng cách xác định số chỉnh sửa cần thiết cho từng ký tự, sau đó lấy trung bình cộng. Tuy nhiên, trong trường hợp khoảng cách lớn hơn 2, từ đó được xem là nhận dạng sai hoàn toàn và được gán độ chính xác bằng 0.

Bảng 1. Điểm khoảng cách Levenshtein [9].

Điều kiện Levenshtein	Điểm
Chính xác Khoảng cách Levenshtein = 0	1
Thêm/Xóa 1 ký tự Khoảng cách Levenshtein = 1	0,9
Thay thế 1 ký tự Thêm/Xóa 2 ký tự Khoảng cách Levenshtein = 2	0,8
Khoảng cách Levenshtein > 2	0

Trong nghiên cứu này, để đánh giá toàn diện một mã container, Khoảng cách Levenshtein được tính riêng cho từng class (ký tự hoặc nhóm ký tự trong mã). Độ chính xác tổng thể của mã container được xác định bằng công thức sau:

$$Accuracy = \frac{Totalscores}{Totalclasses} \times 100 \quad (6)$$

trong đó: *Tổng điểm* là tổng số điểm nhận được từ tất cả các lớp sau khi áp dụng đánh giá bằng khoảng cách Levenshtein.

4. Thực nghiệm

4.1. Chuẩn bị bộ dữ liệu và tinh chỉnh mô hình

4.1.1. Giai đoạn 1

Tập dữ liệu được sử dụng để huấn luyện mô hình YOLOv11 bao gồm 387 hình ảnh với độ phân giải đa dạng. Quá trình xử lý dữ liệu bắt đầu bằng việc phân loại ảnh thành hai lớp đối tượng: "Hệ thống nhận diện" và "Mã kích thước – kiểu loại". Việc gán nhãn cho các lớp này được thực hiện thông qua các nền tảng chuyên dụng trong lĩnh vực thị giác máy tính. Sau đó, tập dữ liệu được chia thành ba phần: huấn luyện, kiểm định và kiểm tra với tỷ lệ tương ứng 80%, 10% và 10% (tương đương 309, 39 và 39 hình ảnh).

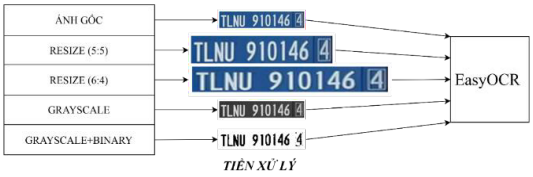
Để chuẩn hóa đầu vào và tăng tính đa dạng cho tập dữ liệu, một số kỹ thuật tiền xử lý và tăng cường dữ liệu đã được áp dụng. Trong đó, các bước tiền xử lý bao gồm tự động điều chỉnh hướng (auto-orientation) và tự động hiệu chỉnh độ tương phản (auto contrast adjustment). Bên cạnh đó, nhằm mở rộng và đa dạng hóa dữ liệu huấn luyện, các kỹ thuật tăng cường như cắt ảnh (cropping), xoay ảnh (shearing), điều chỉnh độ sáng (brightness adjustment), thay đổi mức phơi sáng (exposure variation) và làm mờ (blurring) cũng được triển khai. Sau khi áp dụng các kỹ thuật tăng cường, số lượng hình ảnh trong tập huấn luyện tăng từ 309 lên 927 ảnh.

Ngoài việc chuẩn bị dữ liệu đầu vào, một số tinh chỉnh tham số cho mô hình hiện tại cũng được thực hiện. Batch size (batch) là một trong những siêu tham số quan trọng quy định số lượng dữ liệu được sử dụng trong một lần cập nhật tham số. Khi tăng batch size thì thời gian huấn luyện dữ liệu sẽ được rút ngắn; từ đó mô hình sẽ ổn định và ít bị nhiễu hơn. Tuy nhiên, điều này cũng sẽ tiêu tốn nhiều tài nguyên tính toán hơn và dễ dẫn tới bị lỗi "Out of memory". Đối với cấu hình máy tính hiện có và áp dụng cho mô hình đề xuất, nhóm tác giả đã thử

nghiệm và chọn thông số batch bằng 16 để tạo sự ổn định khi huấn luyện mô hình. Ngoài batch size, Learning rate cũng là một siêu tham số quan trọng, giúp điều chỉnh bước nhảy khi cập nhật trọng số trong quá trình huấn luyện. Nếu Learning rate quá lớn thì có thể ảnh hưởng tới độ chính xác của mô hình (gây dao động lớn ở hàm Loss). Ngược lại, nếu Learning rate quá bé thì sẽ ảnh hưởng đến thời gian huấn luyện mô hình và có thể khiến cho mô hình bị không tối ưu được hàm Loss. Batch size càng lớn thì Learning rate có thể đặt càng cao để mô hình hội tụ nhanh, còn Batch size nhỏ thì cần Learning rate thấp để tránh mất ổn định cho mô hình. Trong nghiên cứu hiện tại, Learning rate được chọn bằng 0.01 để phù hợp với giá trị batch size bằng 16. Với bộ dữ liệu không quá lớn gồm 1005 tấm ảnh và không quá phức tạp với hai lớp đối tượng như trong nghiên cứu hiện tại, thì số vòng lặp của mô hình được chọn là 100 vòng lặp (epochs = 100) để tránh mô hình gặp vấn đề "Overfitting".

4.1.2. Giai đoạn 2

Sau quá trình huấn luyện, mô hình YOLOv11 có khả năng phát hiện và tạo khung bao quanh các mã số container trong ảnh đầu vào. Dựa trên các khung giới hạn được tạo bởi mô hình, ảnh sẽ được cắt để trích xuất vùng chứa mã, qua đó loại bỏ các ký tự và đối tượng gây nhiễu không liên quan. Nhằm nâng cao độ chính xác trong bước nhận diện ký tự bằng EasyOCR, một số kỹ thuật tiền xử lý ảnh đã được áp dụng lên vùng ảnh chứa mã container sau khi cắt. Các phương pháp tiền xử lý này được minh họa trong Hình 6.



Hình 6. Các kỹ thuật tiền xử lý ảnh.

4.2 Kết quả và thảo luận

4.2.1. Giai đoạn 1

YOLOv11 có một số phiên bản khác nhau như n (nano), s (small), m (medium), l (large) với các tốc độ xử lý và độ chính xác khác nhau. Trong nghiên cứu này, hiệu suất mô hình của các phiên bản trên sẽ được khảo sát để tìm ra phiên bản phù hợp với bài toán hiện tại. Bảng 2 trình bày các kết quả huấn luyện bằng các phiên bản khác nhau của YOLOv11 cho cùng chung một bộ dữ liệu trong vòng 100 epochs.

Quan sát từ Bảng 2 cho thấy rằng khả năng nhận diện của các phiên bản mô hình đã huấn luyện không khác nhau đáng kể ở phần hiệu suất, các thông số Precision, Recall và mAP không có nhiều khác biệt. Mặc dù có các chỉ số hiệu suất cao nhất, nhưng phiên bản nano lại có mức độ ổn định không bằng các mô hình cấp cao hơn. Thời gian huấn luyện mô hình, kích thước mô hình có sự khác nhau đáng kể giữa các phiên bản và tăng dần từ n đến l. Tiếp tục thử nghiệm trên cùng một tấm ảnh, cả bốn phiên bản đều cho kết quả chính xác khi nhận diện đủ cả hai lớp nhưng tốc độ nhận diện lại khác nhau như trong Bảng 2. Trong điều kiện của nghiên cứu hiện tại, để tiết kiệm chi phí tính toán cũng như tối ưu về mặt thời gian tính toán, phiên bản nano sẽ được chọn để sử dụng cho các giai đoạn tiếp theo.

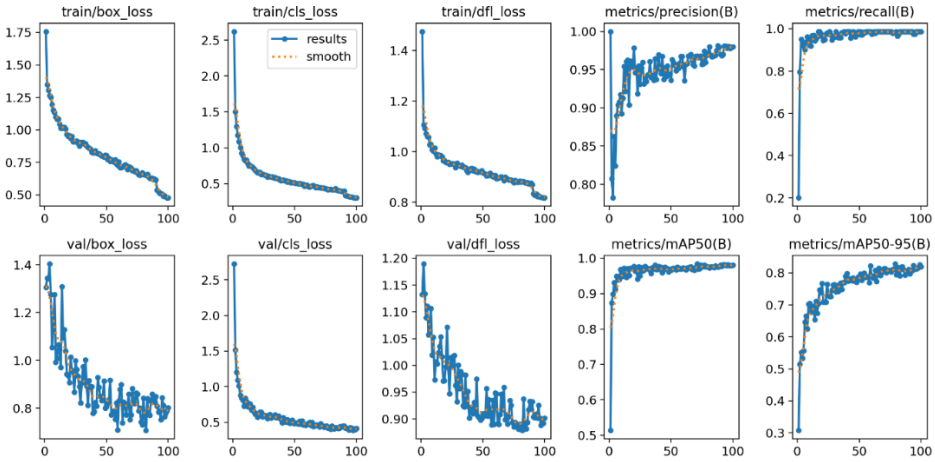
Kết quả huấn luyện mô hình với tập dữ liệu gồm 387 hình ảnh trong 100 vòng lặp (epochs) cho thấy hiệu suất nhận diện ở mức cao, như được thể hiện trong Hình 7. Cụ thể, mô hình có Precision đạt 98,14 %, Recall là 98,75 %, mAP@50 là 98,38 % và mAP@50 :95 đạt 82,85 %. Các chỉ số này cho thấy mô hình được huấn luyện có khả năng phát hiện đối tượng đạt độ chính xác cao và ổn định trên tập dữ liệu đã xây dựng.

4.2.2. Giai đoạn 2

Các kết quả của việc tiền xử lý ảnh cho giai đoạn 2 được thể hiện trong Bảng 3. Dựa vào Bảng 3, có thể nhận thấy rằng đặc điểm bề mặt container – thường được sơn với nhiều màu sắc khác nhau và dễ bám bụi bẩn – ảnh hưởng đáng kể đến hiệu quả nhận diện mã container. Các kỹ thuật tiền xử lý như chuyển sang ảnh xám (Grayscale) hoặc kết hợp Grayscale với nhị phân hóa (Grayscale + Binary) không mang lại hiệu quả cao trong điều kiện này. Ngược lại, phương pháp thay đổi kích thước ảnh (Resize) cho thấy hiệu suất nhận dạng vượt trội. Đáng chú ý, việc điều chỉnh tỷ lệ khung hình về 6:4 (chiều ngang: chiều cao) cho kết quả tốt hơn so với tỷ lệ 5:5, do đặc điểm hình dạng mã container thường có chiều ngang của ký tự hẹp và kéo dài.

Bảng 2. Kết quả huấn luyện các phiên bản YOLOv11.

Phiên bản	Precision lớn nhất	Recall lớn nhất	mAP@50 lớn nhất	mAP@50-95 lớn nhất	Thời gian train 100 epochs	Kích thước mô hình	Tốc độ xử lý 1 tấm ảnh
Nano	0,9814	0,9875	0,9838	0,8285	0,608 giờ	5,2 MB	440,7 ms
Small	0,9832	0,9487	0,9663	0,7947	0,671 giờ	18,3 MB	575,8 ms
Medium	0,9857	0,9512	0,9601	0,7819	1,086 giờ	38,6 MB	1441,6 ms
Large	0,9855	0,9512	0,9531	0,7824	1,462 giờ	48,8 MB	1802,8 ms

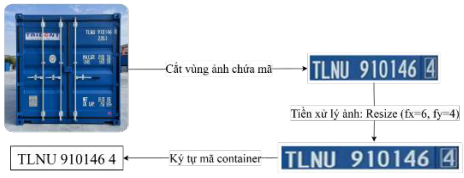


Hình 7. Kết quả huấn luyện mô hình YOLOv11n cho 100 epochs.

Bảng 3. Kết quả tiền xử lý ảnh.

Nội dung	Lớp 1: TLNU 910146 4 Lớp 2: 22G1		Lớp 1: TSLU 053669 3 Lớp 2: 45G1		Lớp 1: MBGU 020250 3 Lớp 2: 25G1	
	Ký tự được nhận diện	Độ chính xác	Ký tự được nhận diện	Độ chính xác	Ký tự được nhận diện	Độ chính xác
Ảnh gốc	TLNU 910146 22G1	95%	TSLU 053669 6 45G1	90%	MBGU 020250 81 256	0%
Resize (fx = fy = 5)	TLNU 910146 4 22G1	100%	TSLU 053669 3 45G1	100%	MBCU 020250 3 25G1	90%
Resize (fx = 6, fy = 4)	TLNU 910146 4 22G1	100%	TSLU 053669 3 45G1	100%	MBGU 020250 13 25G1	95%
GrayScale	TLNU 910146 0 22G1	90%	TSLU 053669 63 45G1	95%	MBGU 020250 81 256	0%
GrayScale + Binary	TLNU 910146 4 22G1	100%	TSLU 053669 45G1	95%	MBGU 020250 B 25G1	90%

Bảng 4 trình bày các kết quả nhận diện 10 mã container bằng mô hình đề xuất kết hợp giữa YOLOv11 và EasyOCR, với kỹ thuật tiền xử lý ảnh Resize theo tỷ lệ 6:4. Quy trình xử lý tổng thể của mô hình được minh họa tại Hình 8. Mô hình đạt độ chính xác trung bình 97% trên tập thử nghiệm gồm 10 mã container, cho thấy hiệu quả cao trong việc phát hiện và khoanh vùng chính xác vị trí mã container. Tuy nhiên, vẫn tồn tại một số sai sót trong giai đoạn nhận dạng ký tự, đặc biệt là đối với chữ số kiểm tra – ký tự cuối cùng trong lớp "Hệ thống nhận diện". Nguyên nhân được cho là do chữ số kiểm tra thường được đặt trong một khung viền riêng biệt, dẫn đến khó khăn cho mô-đun OCR trong việc trích xuất chính xác ký tự này.



Hình 8. Quy trình xử lý mô hình đề xuất.

Bảng 4. Kết quả mô hình đề xuất.

STT	Ký tự gốc	Ký tự được trích xuất	Độ chính xác
1	HCSU 920399 4 25G1	HCSU 920399 25G1	95%
2	HCSU 020088 6 22U1	HCSU 020088 6 22U1	100%
3	TLNU 910146 4 22G1	TLNU 910146 4 22G1	100%
4	KIBU 200442 1 22G1	KIBU 200442 22G1	95%
5	MBGU 020250 3 25G1	MBGU 020250 13 25G1	95%
6	FSCU 372931 4 22G1	FSCU 372931 0 22G1	90%
7	CPIU 047130 1 22G1	CPIU 047130 22G1	95%

STT	Ký tự gốc	Ký tự được trích xuất	Độ chính xác
8	MTBU 202589 3 22G1	MTBU 202589 3 22G1	100%
9	TSLU 053669 3 45G1	TSLU 053669 3 45G1	100%
10	EISU 225201 7 22G1	EISU 225201 7 22G1	100%
Độ chính xác trung bình			97%

5. Kết luận

Việc áp dụng mô hình YOLOv11 kết hợp với EasyOCR trong hệ thống nhận diện mã container đã cho thấy kết quả khả quan, với độ chính xác mAP vượt ngưỡng 90%. Hệ thống thể hiện tiềm năng ứng dụng cao trong các hoạt động logistics và quản lý cảng. Tuy nhiên, vẫn còn tồn tại một số hạn chế, đặc biệt là sự nhạy cảm của mô hình đối với điều kiện môi trường và các góc chụp khác nhau của container.

Để cải thiện hiệu quả hệ thống trong tương lai, các hướng nghiên cứu có thể bao gồm mở rộng và tăng cường tính đa dạng của tập dữ liệu huấn luyện, xây dựng các mô hình đặc thù cho từng góc nhìn khác nhau, và tích hợp nhiều phương pháp OCR nhằm nâng cao độ chính xác trong nhận dạng ký tự. Bên cạnh đó, việc triển khai hệ thống trên nền tảng camera giám sát hoặc ứng dụng di động sẽ mở ra khả năng nhận diện theo thời gian thực, từ đó hỗ trợ tự động hóa quy trình và nâng cao hiệu suất trong các hệ thống cảng thông minh.

Lời cảm ơn

Chúng tôi xin cảm ơn Trường Đại học Bách Khoa, ĐHQG-HCM đã hỗ trợ thời gian, phương tiện và cơ sở vật chất cho nghiên cứu này.

Tuyên bố tác giả

Nhóm tác giả không có xung đột lợi ích.

Tài liệu tham khảo

[1]. M. Mi and Y. Liu, *Smart Ports*, 1st ed. Singapore: Springer Nature Singapore, 2022.

[2]. ISO 6346:2022, *Freight containers – Coding, identification and marking*.

[3]. TCVN 7623:2023, *Công-te-nơ vận chuyển – Mã hóa, nhận dạng và ghi nhãn.*

[4]. Y. Yoon, K.-D. Ban, H. Yoon, and J. Kim, *"Automatic container code recognition from multiple views,"* ETRI Journal, vol. 38, no. 4, pp. 767–775, 2016.

[5]. Ultralytics, *"Ultralytics YOLO11."* [Trực tuyến]. Địa chỉ: https://docs.ultralytics.com/models/yolov11/. [Truy cập: 28/04/2025].

[6]. M. Maithani, D. Meher, and S. Gupta, *"Multilingual Text Recognition System,"* in Lecture Notes in Electrical Engineering, vol. 992, Singapore: Springer, 2023, pp. 103–114.

[7]. P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, and R. Tubagus, *"YOLOv8 to YOLO11: A comprehensive architecture in-depth comparative review,"* arXiv, arXiv:2501.13400v2, 2025.

[8]. R. Kaur and S. Singh, *"A comprehensive review of object detection with deep learning,"* Digital Signal Processing, vol. 132, p. 103812, 2023.

[9]. P. Batra, N. Phalnikar, D. Kurmi, J. Tembhurne, P. Sahare, and T. Diwan, *"OCR-MRD: Performance analysis of different Optical Character Recognition engines for medical report digitization,"* International Journal of Information Technology, vol. 16, pp. 447–455, 2024.