บทความวิจัย

ระบบตรวจจับคิวอาร์โค้ดหลายรายการด้วยการเรียนรู้เชิงลึกสำหรับงานโลจิสติกส์

นัฐธริยา เหล่าประชา* และ อัยยรัชฒ์ ภูพันธ์เว่อ
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 06 2160 4783 อีเมล: natthariya@gmail.com DOI: 10.14416/j.kmutnb.2026.11.001
รับเมื่อ 11 มกราคม 2569 แก้ไขเมื่อ 26 พฤษภาคม 2569 ตอบรับเมื่อ 27 สิงหาคม 2569 เผยแพร่ออนไลน์ 5 พฤศจิกายน 2569
© 2026 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

บทคัดย่อ

การเติบโตอย่างรวดเร็วของธุรกิจพาณิชย์อิเล็กทรอนิกส์ส่งผลให้ปริมาณพัสดุในกระบวนการโลจิสติกส์เพิ่มขึ้นอย่างมีนัยสำคัญ ทำให้เกิดความต้องการระบบตรวจจับรหัส QR ที่มีความแม่นยำและสามารถทำงานได้โดยอัตโนมัติ วิธีการสแกนรหัส QR แบบดั้งเดิมมักมีข้อจำกัดในการตรวจจับรหัสเพียงหนึ่งรหัสต่อครั้ง และอาจมีประสิทธิภาพลดลงในสภาพแวดล้อมคลังสินค้าที่มีพัสดุจำนวนมากและพื้นหลังที่ซับซ้อน งานวิจัยนี้นำเสนอกรอบการทำงานสำหรับตรวจจับรหัส QR หลายรหัสด้วยเทคนิคการเรียนรู้เชิงลึกสำหรับประยุกต์ใช้ในงานโลจิสติกส์ โดยใช้โครงข่ายประสาทเทียมแบบ Region-based Convolutional Neural Network (R-CNN) ร่วมกับ VGG16 สำหรับการสกัดคุณลักษณะ และ Support Vector Machine (SVM) สำหรับการจำแนกประเภท แบบจำลองได้รับการพัฒนาจากชุดข้อมูลที่มีการกระจายของข้อมูลอย่างสมดุล และเพิ่มความหลากหลายของข้อมูลด้วยเทคนิค Data Augmentation เพื่อเพิ่มความสามารถในการเรียนรู้และความทนทานต่อสภาวะแวดล้อมที่แตกต่างกัน การประเมินประสิทธิภาพดำเนินการกับภาพพัสดุจริงที่มีรหัส QR ตั้งแต่ 1 ถึง 10 รหัสต่อภาพ ซึ่งสะท้อนสถานการณ์ที่พบได้ในการปฏิบัติงานด้านโลจิสติกส์ ผลการทดลองพบว่าแบบจำลองมีความแม่นยำในการตรวจจับรหัส QR ร้อยละ 95.82 มีค่า Precision ร้อยละ 94.11 Recall ร้อยละ 95.82 และ F1-score ร้อยละ 94.96 นอกจากนี้ แบบจำลองยังคงมีประสิทธิภาพสูงเมื่อภาพประกอบด้วยพัสดุหลายชิ้นและพื้นหลังที่ซับซ้อน ผลการวิจัยแสดงให้เห็นว่าแนวทางที่นำเสนอมีประสิทธิภาพในการตรวจจับรหัส QR หลายรหัส และสามารถสนับสนุนการระบุและติดตามพัสดุโดยอัตโนมัติในระบบจัดการคลังสินค้าและพัสดุ

คำสำคัญ: การเรียนรู้เชิงลึก การตรวจจับคิวอาร์โค้ดหลายรายการ โครงข่ายประสาทเทียมคอนโวลูชันเชิงภูมิภาค โลจิสติกส์

การอ้างอิงบทความ: นัฐธริยา เหล่าประชา และ อัยยรัชฒ์ ภูพันธ์เว่อ, "ระบบตรวจจับคิวอาร์โค้ดหลายรายการด้วยการเรียนรู้เชิงลึกสำหรับงานโลจิสติกส์," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 37, ฉบับที่ 2, หน้า 1–21, เม.ย.–มิ.ย. 2570. เลขที่บทความ 272-098372, doi: 10.14416/j.kmutnb.2026.11.001.

Research Article

A Deep Learning-Based Multi QR Code Detection System for Logistics Applications

Natthariya Laopracha* and Aiyarat Phupanver
Computer Science, Faculty of Informatics, Mahasarakham University, Maha Sarakham, Thailand

* Corresponding Author, Tel. 06 2160 4783, E-mail: natthariya@gmail.com
DOI: 10.14416/j.kmutnb.2026.11.001
Received 11 January 2026; Revised 26 May 2026; Accepted 27 August 2026; Published online: 5 November 2026
© 2026 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

Abstract

The rapid growth of e-commerce has significantly increased parcel volumes in logistics operations, creating a demand for accurate and automated QR-code detection systems. Conventional QR-code scanning methods are often limited to single-code detection and may exhibit reduced performance in complex warehouse environments containing multiple parcels and cluttered backgrounds. This study proposes a deep learning–based multi-QR-code detection framework for logistics applications using a Region-based Convolutional Neural Network (R-CNN) integrated with VGG16 for feature extraction and a Support Vector Machine (SVM) for classification. The model was developed using a balanced dataset and enhanced through data augmentation techniques to improve learning capability and robustness under diverse operating conditions. For performance evaluation, the proposed framework was tested using real parcel images containing between one and ten QR codes per image, representing practical logistics scenarios. Experimental results demonstrated that the proposed framework achieved a QR-code detection accuracy of 95.82%, with a precision of 94.11%, a recall of 95.82%, and an F1-score of 94.96%. The framework maintained high detection performance even in images containing multiple parcels and complex backgrounds. These results indicate that the proposed framework is effective for multi-QR-code detection and can support automated parcel identification and tracking in warehouse and parcel management systems.

Keywords: Deep Learning, Multi QR Code Detection, Region-based Convolution Neural Network, Logistics

Please cite this article as: N. Laopracha and A. Phupanver, "A deep learning-based multi qr code detection system for logistics applications," *The Journal of KMUTNB*, vol. 37, no. 2, pp. 1–21, Apr.–Jun. 2027 (in Thai), Art. no. 272-098372, doi: 10.14416/j.kmutnb.2026.11.001.

1. Introduction

Computer vision techniques have been widely adopted to automate visual perception tasks in real world systems, including object detection, recognition, and tracking [1]. Among various visual encoding schemes, Quick Response (QR) codes are extensively utilized due to their high data capacity, rapid readability, and strong error correction capability. Consequently, QR codes have become fundamental components in numerous applications such as logistics management, inventory control, mobile payment systems, ticketing, and access control [2].

Despite these advantages, reliable QR-code detection in unconstrained environments remains a challenging problem. In real world scenarios, QR codes are often captured under complex backgrounds, non-uniform illumination, geometric distortion, occlusion, and motion blur. The challenge becomes more critical when multiple QR codes appear simultaneously within a single image, as commonly observed in warehouse and logistics environments. Conventional QR-code scanners and classical image processing–based methods are typically designed for single-code detection and exhibit limited robustness under such conditions [3].

Early research on QR-code detection focused on improving robustness under challenging imaging conditions through adaptive filtering, geometric correction, and neural network–based distortion compensation techniques [4]. Machine learning–based approaches were subsequently introduced to enhance preprocessing, feature extraction, and pattern recognition mechanisms [5]. A systematic literature review by Jain *et al.* [6] summarized a typical multi-stage processing pipeline comprising image acquisition, preprocessing, feature extraction, detection, and decoding and emphasized the critical role of preprocessing in improving detection accuracy in real world applications.

With the advancement of deep learning, QR-code detection has increasingly incorporated modern object detection frameworks. Deep learning–based detectors such as YOLO and various R-CNN variants have demonstrated strong capability in localizing multiple QR codes within complex scenes [7]. In particular, Region-based Convolutional Neural Networks (R-CNN) integrate region proposal mechanisms with deep feature learning, enabling precise localization and improved discrimination between visually similar objects. Improved variants such as Faster R-CNN have achieved high detection accuracy under challenging conditions, including scale variation, background clutter, and partial occlusion

[8]. Compared with single-shot detectors that prioritize real-time performance, R-CNN–based approaches emphasize detection precision and robustness, making them suitable for applications where accuracy is critical [9]. Comparative studies further indicate that QR-code detection methods exhibit strong robustness to image distortions such as skew, rotation, and low illumination, owing to the inherent error correction capability and distinctive structural patterns of QR codes [10].

Recent research has also addressed specialized scenarios, including QR-code detection on dynamically moving objects and real time deployment environments, highlighting the need for detection frameworks that balance accuracy and computational complexity for practical use [11]. In industrial contexts, region-based detection approaches have demonstrated superior reliability. For example, Alsulami and Jhanjhi [12] proposed a deep learning–based QR-code detection system for automated warehouse management, reporting improved robustness under varying illumination and occlusion. Similarly, Yang *et al.* [13] investigated an industrial-oriented QR-code recognition system using Unmanned Aerial Vehicles (UAVs) in warehouse environments, where accurate localization and recognition under real-world conditions are essential.

The rapid expansion of e-commerce and online shopping has significantly increased parcel volumes processed by logistics providers. Manual inspection and sorting are inefficient and prone to error, while existing commercial QR-code scanning systems often struggle to detect multiple QR codes simultaneously, leading to increased operational costs. From a computer vision perspective, this problem requires accurate multi-object detection and robust QR-code localization under real-world industrial conditions.

In practical logistics operations, QR-code recognition requires not only accurate object detection but also robustness to diverse warehouse conditions, including multiple parcels, varying illumination, different viewing angles, complex package labels, and background clutter. Furthermore, logistics systems commonly employ standardized QR codes for parcel identification, shipment tracking, and information retrieval, requiring reliable detection and recognition under real operating conditions. Therefore, the contribution of this study is twofold. First, an R-CNN–based deep learning framework integrating VGG16 and SVM is proposed for accurate multi-QR-code detection. Second, the framework is evaluated using real-world package images collected from warehouse and

logistics environments to assess its applicability to parcel identification and tracking tasks.

R-CNN was selected due to its superior localization accuracy and robustness under complex visual conditions compared to single-stage detectors. By integrating region proposal generation with deep feature extraction, R-CNN enables more accurate localization of small QR codes embedded in overlapping objects, scale variations, cluttered package labels, and partially occluded scenes. These characteristics make R-CNN particularly suitable for industrial logistics environments where detection accuracy is more critical than processing speed alone.

Motivated by these challenges, this paper proposes an industrial-oriented R-CNN–based multi-QR code detection system for logistics applications. The proposed system captures images using a mobile phone camera or webcam, enabling flexible deployment in warehouse and distribution center environments. By leveraging the strong localization capability of R-CNN, the system accurately detects multiple QR codes within a single image and extracts sender and recipient information. The proposed approach aims to improve detection accuracy, reduce misclassification errors, and enhance the efficiency and reliability of QR-code–based logistics workflows in industrial settings. The effectiveness of the proposed framework is validated using real-world package images collected from warehouse and logistics environments, providing insights into its practical applicability for parcel identification and tracking.

2. Materials and Methods

2.1 Data Preparation

The dataset consists of 1,000 images, comprising 500 QR-Code images and 500 Non-QR-Code images, resulting in a balanced dataset for binary classification. The original images were captured using a smartphone camera under practical logistics conditions and subsequently resized to 224 × 224 pixels to satisfy the input requirements of the VGG16 network for feature extraction and model training. To determine the optimal model performance, the dataset was divided into two experimental configurations. In the first configuration, 70% of the dataset was used for training and 30% for validation. In the second configuration, 80% of the dataset was allocated for training and the remaining 20% for validation.

The preparation of the training and validation datasets is illustrated in Figure 1, which presents representative package images



(a) The dataset image contains two QR-codes



(b) The dataset image contains four QR-codes



(c) The dataset image contains eight QR-codes

Figure 1 Representative images from the dataset used in this study.

used for training and validation. Figures 1(a), 1(b), and 1(c) show dataset images containing two, four, and eight QR codes, respectively. QR code and Non-QR-code regions were manually cropped and annotated as ground-truth samples. To enhance model robustness, data augmentation techniques including rotation, skewing, scaling, blurring, and noise addition were applied to simulate challenging logistics conditions. The ground-truth labeling process is illustrated in Figure 2 (a)–(c), which is the ground-truth annotation process used for QR-code localization. QR-code regions are manually annotated using bounding boxes, where each object is represented by four coordinates: Xmin, Ymin, Xmax, and Ymax. These parameters correspond to the minimum x-coordinate, minimum y-coordinate, maximum x-coordinate, and maximum y-coordinate of the bounding box, respectively. The resulting annotation file records the location of each QR-code object and is subsequently used for training and validation of the proposed detection model.

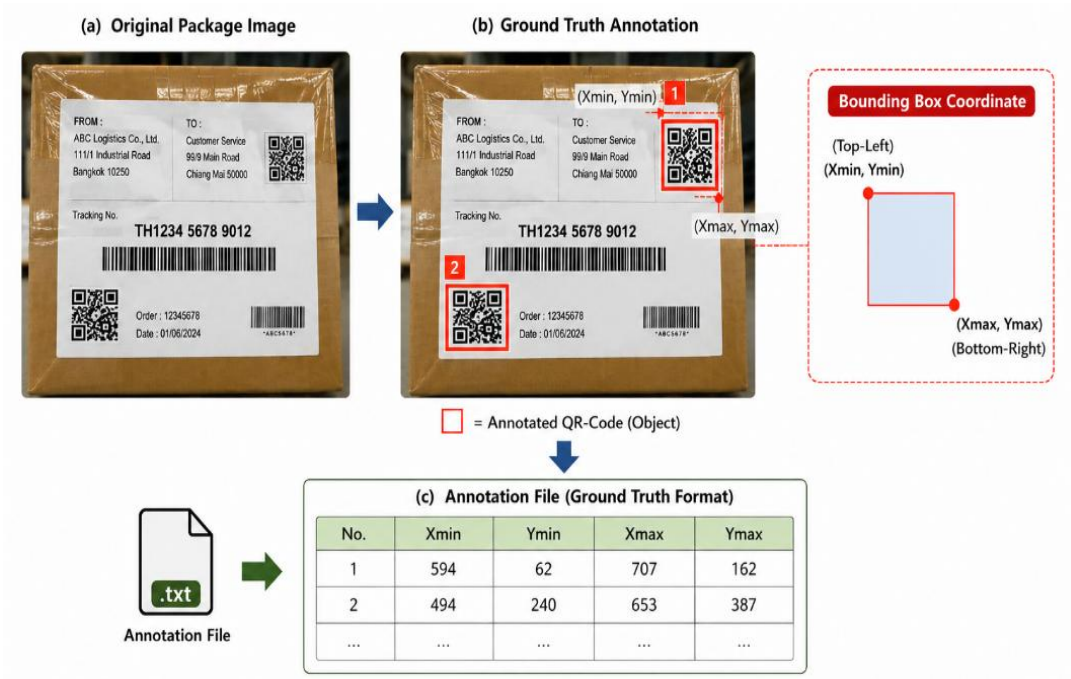


Figure 2 Ground-truth annotation process for QR-code localization. (a) Original package image. (b) Manual annotation of QR-code objects using bounding boxes. (c) The annotation file stores the bounding-box coordinates in the format (Xmin, Ymin, Xmax, Ymax), which are used for training and validation of the proposed detection model.

2.2 Research Methods

This study consists of two main stages. In the first stage, a multi-QR-code detection model is developed and evaluated. In the second stage, the selected model is integrated into a web-based application that can be accessed through both mobile devices and desktop computers. The entire system, including the detection model and the web application, is implemented using the Python programming language. The overall workflow of the proposed system is illustrated in Figure 3.

As shown in Figure 3, the first stage involves a comparative evaluation of the VGG16 and ResNet50 models. During the training phase, QR-code and Non-QR-code samples are used to train and validate the classification models under different experimental configurations. The model achieving the highest validation performance is selected for subsequent deployment. Based on the experimental results, VGG16 demonstrates superior performance and is therefore adopted as the feature extraction network in the proposed framework.

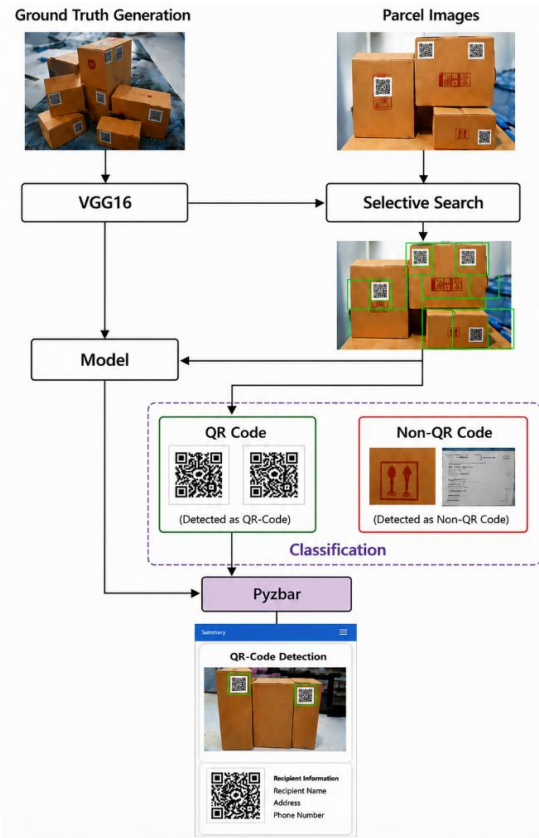


Figure 3 Workflow of the proposed QR code detection system.

In the second stage, the selected VGG16-SVM model is integrated into an R-CNN framework for multi-QR-code detection. Selective search is first employed to generate candidate object regions from the input image. Each candidate region is then processed by the VGG16 feature extractor, and the resulting feature vectors are classified by a trained SVM classifier into QR-code and Non-QR-code categories. Regions classified as QR codes are subsequently localized and forwarded for information decoding. The computational details of the R-CNN, VGG16, and SVM components are presented in sections 2.3 and 2.4, respectively.

2.3 Region-based Convolutional Neural Network (R-CNN)

A region-based convolutional neural network [14] is a deep learning framework widely used for object detection in computer vision applications. The R-CNN approach begins by generating candidate object regions, referred to as region proposals, using the Selective Search algorithm. These region proposals are subsequently resized to a fixed input size and processed by a Convolutional Neural Network (CNN) to extract discriminative visual features. The extracted features are then classified using a Support Vector Machine (SVM) classifier to determine whether the region corresponds to a QR-code or a Non-QR-code object. In addition, bounding-box regression can be applied to refine the predicted object locations and improve localization accuracy.

Figure 4 illustrates the workflow of the proposed R-CNN– based QR-code detection framework. First, an input image containing one or more parcels is acquired. The Selective Search algorithm is then employed to generate candidate object regions. Each candidate region is resized and processed by the VGG16 network to extract deep feature representations. The resulting

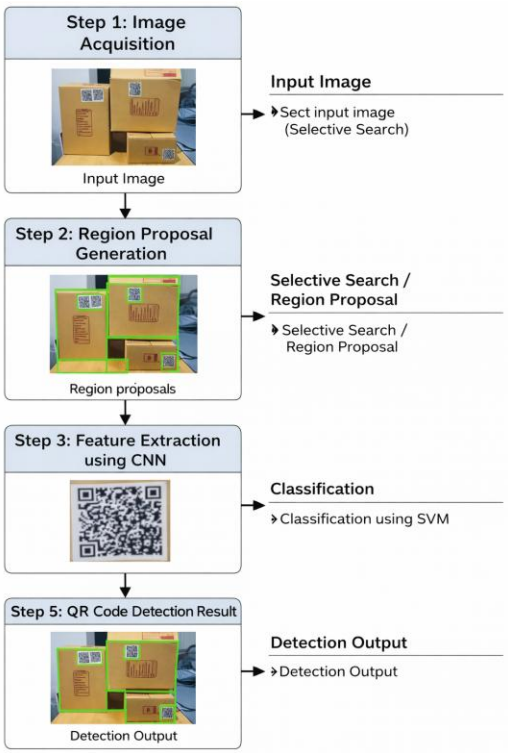


Figure 4 The workflow of the R-CNN-based QR code detection.

feature vectors are subsequently classified by the trained SVM model into QR-code and Non-QR-code categories. Regions classified as QR codes are finally localized and forwarded to the decoding stage for information extraction. This framework enables accurate multi-QR-code detection in logistics environments.

2.4 VGG16-SVM Classification Framework

In the classification stage, the VGG16 convolutional neural network was employed as a feature extractor to learn discriminative visual representations from the input images. The extracted feature vectors obtained from the VGG16 network were subsequently forwarded to the Support Vector Machine classifier for object classification. The SVM classifier was used to determine whether the detected object corresponded to a QR-code or a non–QR-code object based on the extracted visual features.

Figure 5 illustrates the training process of the proposed VGG16-SVM model for QR-code recognition. The training dataset is composed of QR-code images and cropped Non-QR-code background images extracted from package surfaces, shipping labels, sender and receiver information, and other irrelevant regions. Prior to training, all images are resized to 224 × 224 × 3 pixels and provided as input to the VGG16 network. Through five convolutional blocks consisting of convolution, ReLU activation, and max-pooling layers, VGG16 extracts hierarchical feature representations from the input images. The resulting feature maps are flattened and transformed into a 4096-dimensional feature vector through fully connected layers. Subsequently, the extracted features are used to train an SVM classifier for binary classification.

Hyperparameter optimization was performed using a grid-search strategy. Different combinations of learning rate, batch size, optimizer, kernel type, C value, and gamma value were systematically

evaluated. The configuration yielding the highest validation accuracy was selected as the optimal parameter set. The final hyperparameter settings are summarized in Table 1 and were subsequently used to train the final VGG16- SVM model for QR- code detection and classification.

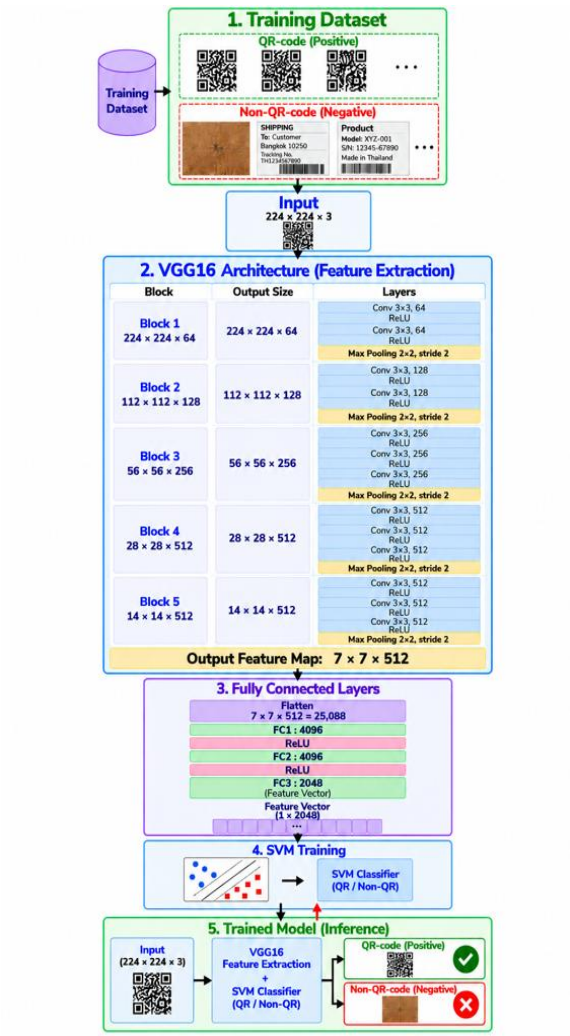


Figure 5 Training Pipeline of the Proposed VGG16-SVM Model for QR-Code Classification.

Table 1 Hyperparameter search space and optimal parameter settings for VGG16 and SVM models.

VGG16		
Hyperparameter	**Tested Values**	**Final Value**
Learning Rate	0.01, 0.001	0.001
Batch Size	16, 32, 64	32
Optimizer	SGD, Adam	Adam
SVM		
Hyperparameter	**Tested Values**	**Final Value**
Kernel	Linear, RBF	RBF
C	0.2, 1.0, 10	1.0
Gamma	Auto, scale	scale

After the best-performing VGG16-SVM model has been obtained, it is deployed for QR- code detection as illustrated in Figure 6. Images containing multiple package boxes are processed by the R-CNN framework, where selective search generates candidate regions from each package. The candidate regions are resized and passed through the trained VGG16 network for feature extraction. The resulting feature vectors are then classified by the trained SVM model as either QR-code or Non-QR-code regions. Finally, the detected QR- codes are localized using bounding boxes, enabling automatic QR- code detection from multiple package boxes in a single image.

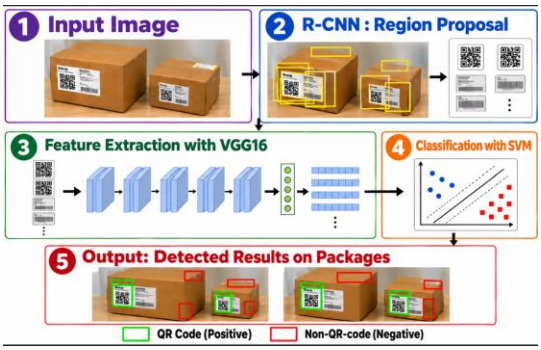


Figure 6 Inference Workflow of the Proposed R-CNN- Based Multi- QR- Code Detection System Using VGG16 and SVM.

After the QR- code regions have been successfully detected and classified by the trained VGG16- SVM model, the detected QR-code images are forwarded to the Pyzbar library for decoding. Pyzbar is an open-source barcode and QR- code decoding library that extracts encoded information directly from the detected QR- code region. The library analyzes the QR-code pattern and converts the embedded data into readable text. In this study, the decoded information includes logistics- related details such as recipient name, address, phone number, tracking information, and other encoded shipment data. By processing only the regions classified as QR- codes, the system reduces unnecessary computations and improves the reliability of the decoding process. The extracted information is subsequently displayed to the user through the application interface for further verification and utilization.

3. Result and Discussions

3.1 QR Code detection result

This study evaluates the performance of two deep learning models, VGG16 and ResNet50, to identify the optimal model for QR code detection. The experimental results are summarized in Table 2. Although both VGG16 configurations achieved identical validation accuracy, the 70:30 training–validation split was selected because it required less training time than the 80:20 configuration. Therefore, this model was used for subsequent system development.

Based on the experimental results presented in Table 2, the VGG16 model was selected to develop the proposed system. The model was evaluated using images of parcel boxes containing QR codes.

Table 2 Performance comparison of VGG16 and ResNet50 in terms of accuracy and processing time.

Model	VGG16 70:30	Restnet50 70:30	VGG16 80:20	Restnet50 80:20
Training Accuracy (%)	100	100	100	80
Validation Accuracy (%)	100	50	100	60
Learning Time (second)	105	47	114	50

Table 3 Confusion Matrix of the VGG16 Model on the Validation Set (70:30 Split).

Actual/Predicted	QR-Code	Non-QR code
QR-Code	150	0
Non-QR code	0	150

Table 4 Confusion Matrix of the VGG16 Model on the Validation Set (80:20 Split).

Actual/Predicted	QR-Code	Non-QR code
QR-Code	100	0
Non-QR code	0	100

Tables 3 and 4 present the confusion matrices of the VGG16 model evaluated on the validation datasets using the 70:30 and 80:20 training–validation splits, respectively. In both experimental configurations, all QR-Code and Non-QR-Code samples were classified correctly, resulting in zero false positives and zero false negatives. Consequently, the model achieved 100% accuracy, precision, recall, and F1-score on both validation sets. These results indicate that the VGG16 model effectively learned discriminative visual features for distinguishing QR-Code and Non-QR-Code images within the prepared dataset. The consistent performance across both validation configurations demonstrates the effectiveness of the proposed VGG16-SVM classification framework. However, further evaluation using larger and more diverse logistics datasets is necessary to assess the generalization capability of the proposed model under real-world operational conditions.

Table 5 Sample images for evaluating QR code detection with varying numbers of boxes.

Number of boxes	Sample 1	Sample 2
1		
2		
3		
4		
5		

Table 5 Sample images for evaluating QR code detection with varying numbers of boxes. (*continued*)

Number of boxes	Sample 1	Sample 2
6		
7		
8		
9		
10		

In the QR-code detection experiments, the evaluation was conducted using images containing between one and ten boxes per image. Each image includes different box alignments. The experimental results for images containing one to ten boxes are presented in Table 5. Correct detections are represented by green bounding boxes. QR code detection is performed using the R-CNN approach. The experimental results are summarized in Table 6.

Table 6 Results of QR code detection using R-CNN with 10 QR Code images per box.

Number of Boxes per Image	Number of QR codes	Number of correctly detected QR Code	Number of incorrectly detected QR Code
1	10	9	1
2	20	20	0
3	30	30	0
4	40	40	0
5	50	50	0
6	60	59	1
7	70	66	4
8	80	76	4
9	90	83	7
10	100	94	6
Total		527	23

Table 6 presents the QR-code detection results obtained using the proposed R-CNN framework under different numbers of parcel boxes per image. Each parcel box contains one QR code, resulting in a total of 550 QR codes evaluated across all experimental scenarios. The proposed system correctly detected 527 QR codes, while 23 QR codes were missed, corresponding to an overall detection accuracy of 95.82%. The results indicate that the proposed method achieves excellent detection performance

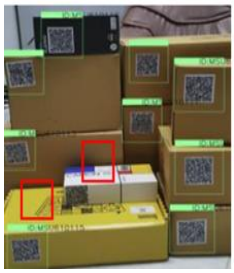
when the number of boxes per image is relatively small. In particular, images containing 2–5 boxes achieved 100% detection accuracy, with all QR codes correctly detected. As the number of boxes increased beyond five, a slight decrease in detection performance was observed. For example, images containing six boxes achieved 98.33% accuracy (59/60), while images containing seven and eight boxes achieved 94.29% (66/70) and 95.00% (76/80), respectively. The most challenging scenario was observed when the image contained nine boxes, where 83 of 90 QR codes were correctly detected, resulting in an accuracy of 92.22%. This reduction in performance is mainly attributed to increased object density, smaller QR-code regions, partial occlusion, and greater visual complexity within the image. Nevertheless, even in the most challenging cases, the proposed R-CNN–based framework maintained a detection accuracy exceeding 90%, demonstrating its effectiveness for multi-QR-code detection in logistics applications. However, the observed performance degradation at higher object densities suggests that further optimization is required to improve detection reliability in large-scale warehouse environments containing densely packed parcels.

Table 7 summarizes the overall detection performance of the proposed R-CNN–based QR-code detection framework on multiple package images. The proposed system achieved a detection accuracy of 95.82%, indicating that the majority of QR-code regions were successfully identified and localized. The precision value of 94.11% demonstrates that most regions classified as QR codes were correct detections, with only a small number of false-positive predictions. In addition, the recall value of 95.82% indicates that the proposed framework was able to detect most of the QR codes present in the test images, resulting in a low number of missed detections. The F1-score of 94.96% further confirms the balanced performance between precision and recall. Overall, these results demonstrate that the proposed framework provides reliable and effective multi-QR-code detection in logistics environments, even when multiple package boxes are present within a single image.

Table 7 Detection Performance of the Proposed Framework on Multiple Package Images.

Metric	Value(%)
Detection Accuracy	95.82
Precision	94.11
Recall	95.82
F1-Score	94.96

Figure 7(a)–(b) presents representative examples of multi-QR-code detection results obtained using the proposed framework. Green



(a) 10 QR-Codes/image



(b) 9 QR-Codes/image

Figure 7 Representative examples of (a) 10 QR-Code/image and (b) 9 QR-Code/image QR code detection outcomes.

bounding boxes indicate correctly detected QR codes, whereas red bounding boxes represent missed or incorrectly detected QR codes. Detection errors are more likely to occur when QR codes are blurred, partially occluded, or exhibit visual characteristics similar to the surrounding package surfaces and labels. These conditions reduce the distinct visual features available for feature extraction and classification, thereby increasing the difficulty of accurate QR-code localization and detection. The missed detections were primarily observed in densely packed package arrangements where QR codes appeared at small scales or were partially occluded.

Table 8 presents a comparative evaluation of several QR-code detection methods reported in previous studies. The image-processing and ZBar-based detection approach achieved an accuracy ranging from 80% to 90% when evaluated under varying environmental conditions. However, its performance was sensitive to image quality, illumination variation, and background complexity. The YOLOv5-640 method reported the highest accuracy of 97.6%, demonstrating excellent detection capability under the experimental conditions considered in the study. The Faster R-CNN approach achieved an accuracy of 89.00% on datasets containing QR codes with diverse orientations and visual characteristics. Compared with these methods, the proposed R-CNN–based framework achieved a detection accuracy of 95.82%. The results indicate that the proposed method provides competitive performance for multi-QR-code detection while maintaining robustness in logistics scenarios involving multiple package boxes, varying QR-code positions, and complex background information. These findings demonstrate the effectiveness of combining R-CNN, VGG16 feature extraction, and SVM classification for QR-code localization and detection in practical logistics applications.

Table 8 Performance comparison of QR-code detection methods in terms of accuracy.

Method	Accuracy (%)
Image processing+ Zbar detection [14]	80–90
YOLOv5-640 [15]	97.6
Faster R-CNN [16]	89.00
Proposed Method	**95.82**

3.2 Application implementation

In this study, a web-based application was developed using the Flask framework. The backend was implemented in Python and integrated with the trained R-CNN inference engine, while the frontend accessed the detection service through Flask-based web APIs. This architecture supports both desktop and mobile devices and enables cross-platform QR-code detection through standard web browsers.

The operational workflow of the proposed system is illustrated in Figure 8. First, users access the web application and capture an image using either a mobile device camera or a webcam before uploading the image to the system. Subsequently, the proposed model detects multiple QR codes within the uploaded image and generates bounding boxes around the detected regions. The system then displays the recipient information and indicates the order of each detected parcel, enabling warehouse staff to accurately identify and retrieve the corresponding packages, as shown in Figure 9.

The proposed system was designed to support practical logistics and parcel management applications. Nevertheless, system performance may be affected under challenging environmental conditions, such as poor illumination, motion blur, partial occlusion, and densely overlapping QR codes. Therefore, further optimization and large-scale deployment testing are necessary to improve system robustness and operational reliability in real warehouse environments.

Table 9 summarizes the overall performance of the proposed framework in terms of QR-code detection and QR-code decoding. The proposed R-CNN–based detection framework achieved a detection accuracy of 95.82%, indicating that most QR-code regions were successfully localized within the package images. However, the QR-code decoding accuracy obtained using the Pyzbar library was 55.03%, which is considerably lower than the detection accuracy.



Figure 8 Deep learning–based multi QR code detection system for parcel logistics.



Figure 9 Multi QR code detection result.

Table 9 Detection and Decoding Performance of the Proposed Framework.

Method	Value(%)
QR-Code Detection Accuracy	95.82
QR-Code Decoding Accuracy (Pyzbar)	55.03

The discrepancy between detection and decoding performance can be attributed to several factors. Although the proposed framework successfully localized QR-code regions, some detected QR codes occupied only a small portion of the original image. Consequently, the cropped QR- code regions contained limited visual information and reduced effective resolution, which adversely affected the decoding capability of the Pyzbar library. In addition, image blur, illumination variation, partial occlusion, and perspective distortion further reduced decoding reliability.

Previous studies have reported that traditional QR- code decoding methods, including ZBar- based approaches, often experience performance degradation under challenging real-world conditions, particularly in the presence of image blur, perspective distortion, occlusion, and non- uniform illumination [7], [17]. These factors can significantly reduce decoding reliability and increase the likelihood of decoding failures. Therefore, the decoding performance observed in this study is consistent with the known limitations of conventional QR-code

decoding libraries, especially when small or low-quality QR-code regions are extracted from complex logistics images.

In addition, the overall system performance may be affected by poor lighting conditions, severe image blur, partial occlusion, and complex background information commonly encountered in logistics environments. These factors may reduce both QR-code detection and decoding accuracy, particularly when multiple parcels are densely arranged within a single image.

4. Conclusions

This study presented a deep learning–based multi-QR-code detection framework for logistics applications using a Region-based Convolutional Neural Network integrated with VGG16 feature extraction and SVM classification. The proposed framework was developed to support automated parcel identification and information retrieval in logistics operations. Experimental results demonstrated that the proposed system achieved a QR-code detection accuracy of 95.82%, with a precision of 94.11%, a recall of 95.82%, and an F1-score of 94.96%. The results indicate that the proposed framework can effectively detect multiple QR codes within a single image, even in scenarios involving multiple parcels and complex background information commonly encountered in logistics environments.

The study also evaluated QR-code decoding performance using the Pyzbar library. Although the proposed framework successfully localized QR-code regions, the decoding accuracy was lower than the detection accuracy due to factors such as small QR-code size, image blur, illumination variation, partial occlusion, and perspective distortion. These findings highlight that reliable logistics automation depends not only on accurate object detection but also on robust QR-code decoding performance under practical operating conditions.

From an industrial perspective, the proposed framework demonstrates strong potential for deployment in warehouse and parcel management systems. The ability to simultaneously detect multiple QR codes can reduce manual inspection effort and improve operational efficiency in parcel sorting and tracking processes. Furthermore, the proposed framework can be extended to other object detection and industrial inspection applications, including barcode detection, inventory management, warehouse monitoring, and intelligent transportation systems, demonstrating its broader applicability in computer vision–based automation.

Future work will focus on improving QR-code decoding performance and system robustness under challenging logistics conditions

through higher-resolution image acquisition, image enhancement techniques, super-resolution reconstruction, and more advanced decoding algorithms. In addition, larger and more diverse logistics datasets will be investigated to further evaluate the scalability and generalization capability of the proposed framework in real-world warehouse environments. To provide a more comprehensive performance evaluation, future studies will compare the proposed approach with more recent CNN architectures, including ResNet, MobileNet, and EfficientNet, considering not only classification accuracy but also computational efficiency, inference speed, memory usage, and resource requirements. Moreover, the effects of training hyperparameters, particularly the number of training epochs, will be systematically investigated to identify the optimal model configuration for practical QR-code recognition in logistics applications.

5. Acknowledgment

This research was financially supported by a grant from the Faculty of Informatics, Mahasarakham University, in 2026.

References

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.

[2] D. A. Noever and F. G. McKee, “Dueling QR codes: The hyding of Dr. Jeckyl,” *arXiv:2503.13458*, Feb. 2025, doi: 10.48550/arXiv.2503.13458.

[3] M.- J. Tsai, Y.- C. Lee, and T.- M. Chen, “Implementing deep convolutional neural networks for QR code-based printed source identification,” *Algorithms*, vol. 16, no. 3, Mar. 2023, Art. no. 160, doi: 10.3390/a16030160.

[4] L. Huo, J. Zhu, P. K. Singh, and P. A. Pavlovich, “Research on QR image code recognition system based on artificial intelligence algorithm,” *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 855–867, Jan. 2021, doi: 10.1515/jisys-2020-0143.

[5] G. H. Girisha, A. D. Kumar, A. Singh, B. K. P., and Deepak, “QR Code detection," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 357–360, Jun. 2022, doi: 10.48175/ijarsct-5353.

[6] V. Jain, Y. Jain, H. Dhingra, D. Saini, M. C. Taplamacioglu, and M. Saka, “A systematic literature review on QR code detection and pre-processing,” *International Journal on Technical and Physical Problems of Engineering*, vol. 13, no. 1, pp. 111–119,

Mar. 2021.

[7] D. J. Corpuz, L. V. D. Rosario, J. P. Cempron, P. L. Lozano, and J. Ilao, "QR code detection with perspective correction and decoding in real-world conditions using deep learning and enhanced image processing," in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, vol. 3, 2025, pp. 685–690, doi: 10.5220/00 13287200003912.

[8] C.-Q. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z.-J. Feng, Y. Liu, and Z. Wu, " An improved faster R-CNN for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019. 2932731.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, " YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020, doi: 10.48550/arXi v.2004.10934

[10] M. Rostan, " A comparative study on QR code and bar code detection techniques in modern systems," *International Journal of Synergy in Engineering and Technology*, vol. 6, no. 1, pp. 14–22, 2025.

[11] M. Rostan and S. Safei, "Enhance efficiency of QR code detection on movement object: Performance analysis of algorithms and environmental factors," *International Journal of Synergy in Engineering and Technology*, vol. 6, no. 1, pp. 51–62, 2025.

[12] F. Alsulami and N. Z. Jhanjhi, "Deep learning framework for barcode localization and decoding using simulated UAV imagery," *Scientific Reports*, vol. 16, 2026, Art. no. 399, doi: 10.1038/s41598-025-29720-w.

[13] S.-Y. Yang, H.-C. Jan, C.-Y. Chen, and M.-S. Wang, " CNN-based QR code reading of package for unmanned aerial vehicle," *Sensors*, vol. 23, no. 10, 2023, Art. no. 4707, doi: 10.3390/s23104707.

[14] A. A. Alsulami, Q. A. Al-Haija, B. Alturki, A. Yafoz, A. Alqahtani, R. Alsini, and S. S. Binyamin, "Efficient malicious QR code detection system using an advanced deep learning approach," *Computer Modeling in Engineering & Sciences*, vol. 145, no. 1, pp. 1117–1140, 2025, doi: 10.32604/cmes. 2025.070745.

[15] R. Chen, H. Huang, Y. Yu, J. Ren, P. Wang, H. Zhao, and X. Lu, "Rapid detection of multi-QR codes based on multistage stepwise discrimination and a compressed MobileNet," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 15966–15979, Sep. 2023, doi: 10.1109/JIOT.2023.3268636.

[16] A. Sharif, G.-T. Zhai, X.-K. Min, J. Jia, and K. Munir, "Enhancing decoding rate of barcode

decoders in complex scenes for IoT systems," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17495–17507, Dec. 2021, doi: 10.1109/JIOT.2021.3081555.

[17] J. Li, D. Zhang, M. Zhou, and Z. Cao, "A motion blur QR code identification algorithm based on feature extracting and improved adaptive thresholding," *Neurocomputing*, vol. 493, pp. 351–361, 2022, doi: 10.1016/j.neucom.2022.04.041.