INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

An Enhanced Barcode Recognition Framework Integrating Yolov5 Detection with Pyzbar-Based Decoding

Vijaya J[1*], G.S. Abhinav[2], Alla Abhiram[3], Nenavathu Pranay[4]

[1]Assistant Professor/ Dept DSAI, IIITNR, Raipur, Chhattisgarh, India

[2, 3, 4,] UG Student/ IIITNR, Raipur, Chhattisgarh, India

***Corresponding author**

DOI: https://doi.org/10.51244/IJRSI.2025.12120048

Received: 17 December 2025; Accepted: 22 December 2025; Published: 04 January 2026

ABSTRACT

Barcode detection is an essential process in contemporary automation, inventory management, and retailing, which enables fast and precise data recovery. The greater dependence on automatic systems has brought about the need for more stable and effective barcode detection systems to deal with complex real-world environments. This paper introduces a hybrid system that combines state-of- the-art deep learning-based object detection with traditional barcode decoding methods in order to increase overall accuracy, reliability, and efficiency. The system implemented under the proposal uses the YOLOv5 model, a commonly used deep learning model renowned for its fast processing speeds and accurate localization. Utilizing YOLOv5, the system provides real-time barcode detection, even with difficult conditions like changing light conditions, occlusion, and cluttered backgrounds. The method improves detection rates greatly while balancing the trade-off between computational complexity and real-time processing constraints. In addition, the barcode scanning process is improved through the implementation of Pyzbar, an efficient library utilized for extracting ordered data from the 1D and 2D formats of barcodes. The system attains high adaptability across different barcode formats and operating environments, while maintaining deployment feasibility on modern edge and desktop platforms through model optimization strategies. In this combined procedure, flexibility can be increased along with minimizing the possibility of obtaining false negatives as well as refining data recovery rates. The outcomes showcase how deep learning-based object detection coupled with conventional decoding methods presents an extensive solution for barcode recognition in changing and complex environments. This research highlights the ability of hybrid models in providing high-performance barcode detection, ultimately leading to innovations in automation and intelligent data management systems.

Keywords—Barcode Detection, YOLOv5, Pyzbar, Object Detection, Deep Learning, Computer Vision

INTRODUCTION

Barcode detection is now an essential field of research because of the extensive use in automation, inventory management, and retail management systems. With enterprises and industries more and more relying on digital measures for efficiency and precision, barcode detection and decoding at high speeds and with guaranteed reliability are extremely important for maximizing operations. From following products within supply chains to facilitating smooth checkout mechanisms within retail outlets, barcode recognition has greatly enhanced operational speed, security, and real-time processing of data within smart environments [1] [11-15]. Barcode detection, however, is problematic in real-world applications such as changing lighting conditions, distortions, different orientations, and occlusions that tend to jeopardize accuracy [2] [18-19]. To overcome these challenges, sophisticated computer vision and machine learning (ML) methods have been incorporated into barcode recognition systems to make them more reliable and flexible. Object detection techniques based on deep learning have transformed barcode localization, with dramatic enhancements in accuracy and resilience [3-4] [9-10].

Conventional barcode scanning methods depended significantly on perfect conditions, like consistent illumination and unobstructed visibility. But with the advent of contemporary deep learning algorithms, barcode detection has become easier, even in less-than-perfect conditions. Of the commonly employed methods, YOLOv5 and Pyzbar have become increasingly popular because of their outstanding barcode detection and decoding capabilities, respectively [5-6].

YOLOv5 (You Only Look Once version 5) is a cutting-edge object detection model with real-time capabilities and high accuracy. It effectively detects barcodes from diverse environments, such as dynamic and cluttered backgrounds. The model is optimized for fast image processing while ensuring accurate localization, which makes it ideal for applications that demand high-speed recognition. It performs well in barcode detection even when the barcodes are partially occluded or distorted [7]. Conversely, Pyzbar is a light, yet effective, library for both 1D and 2D barcode decoding. The main strength of Pyzbar is that it can extract ordered data with small computational overheads. In contrast to most decoding algorithms, Pyzbar also proves to be robust under unsatisfactory situations like blurriness, noisiness, or partial availability of the region of interest corresponding to the barcode. This attribute makes it an effective choice in real-world scenarios where image quality might not necessarily be perfect all the time. [8]

This paper suggests a hybrid barcode recognition system that combines YOLOv5 for barcode localization and Pyzbar for barcode decoding to provide an effective and efficient solution for real-world applications. By leveraging the strengths of deep learning-based object detection and traditional barcode decoding methods, the suggested system is expected to provide better speed, accuracy, and scalability. The deep learning model initially detects the position of the barcode in an image so that Pyzbar can precisely extract and decode the encoded information. Although YOLOv5 is extremely effective in barcode region detection, it is not capable of performing decoding on its own. On the other hand, Pyzbar excels in decoding but needs well-extracted and clean barcode regions to perform at its best. Through combining these two methods, the suggested system is able to attain a perfect equilibrium among accuracy, computational optimization, and real- time processing. This makes it an ideal solution for various sectors, such as retail, logistics, automated stock management, and smart inventory systems. By utilizing this hybrid method, barcode recognition is made more capable of coping with the growing needs of current industries while providing quick and accurate scanning in tough environments. Deep learning-based detection and conventional decoding together provide a viable solution to be implemented in a wide range of real- world applications, thereby enhancing automation and efficiency in systems based on barcodes.

MATERIALS AND METHODS

This study introduces hybrid barcode detection and decoding framework that integrates the YOLOv5 deep learning model with the lightweight Pyzbar library. The system is designed to provide accurate barcode localization in complex visual environments while ensuring efficient decoding with minimal computational load. The complete pipeline includes dataset preparation, preprocessing, YOLOv5 training, model deployment, real-time frame acquisition, barcode detection, barcode decoding, and final result visualization as illustrated in Figure 1.

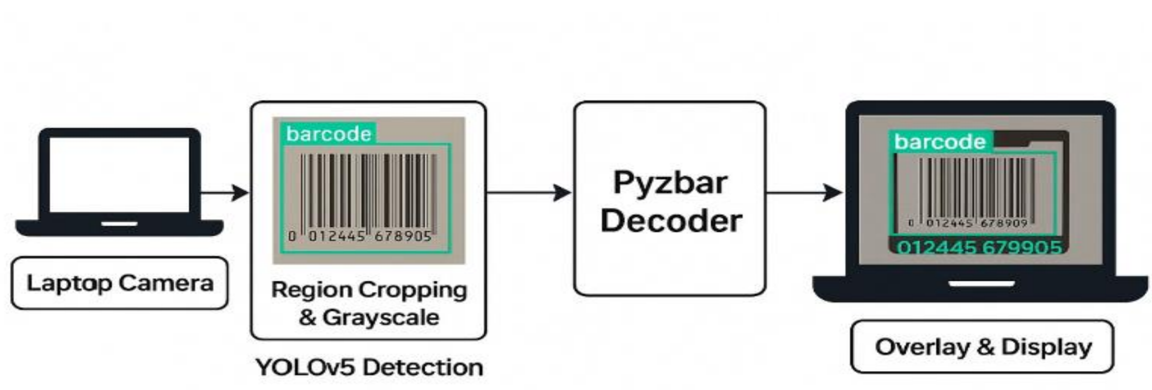

Figure 1: Proposed barcode detection and decoding framework

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Dataset Preparation

The dataset used in this research consists of 4,096 training images and 1,094 testing images collected under diverse environmental conditions. These images include variations in lighting, background clutter, occlusions, and multiple barcode formats such as EAN-13, Code 128, QR, and DataMatrix. All images are manually annotated using LabelImg, and bounding boxes are stored in the YOLO format containing class labels and normalized coordinates. This curated dataset forms a robust foundation for training the YOLOv5 model [20].

Data Preprocessing and Augmentation

Before training, all images are resized to 640 × 640 to ensure compatibility with YOLOv5. To improve model generalization, standard augmentation techniques such as rotation, scaling, and contrast adjustment are applied to a subset of the dataset. These augmentations allow the model to learn the variability found in real-world scenarios. The annotated labels are also verified and normalized to maintain consistency during training.

YOLOv5 Model Training

The YOLOv5 object detection model is selected for its strong balance between speed and accuracy, making it suitable for real-time applications. Training is carried out on a GPU-enabled workstation using an optimized learning rate of 0.01, a batch size of 16, and 300 epochs. The CIoU loss function is used to optimize bounding-box predictions, while objectness and classification losses ensure accurate detection of barcode regions [16-17].

$$\mathcal{L}_{\text{CIoU}} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v, \quad (1)$$

Where IoU is the intersection-over-union, $\rho^2(\cdot)$ is the Euclidean distance between the predicted box b and ground-truth b^{gt}, c is the diagonal length of the minimum enclosing box, v measures aspect ratio consistency, and α is a trade-off parameter. After training, the best-performing model weights are saved as *best.pt* for further deployment and inference. During evaluation, IoU is important for measuring how good your detection is. TheIoU is given by:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2)$$

Where: Overlap is the intersection area between the predicted and ground truth bounding boxes. And Union is the total area covered by both boxes. YOLO predicts an ”objectness” score that tell whether an object exists inside a bounding box. The objectness score is defined as:

$$\text{Objectness Score} = P(\text{Object}) \times \text{IoU}(\text{predicted box, ground truth box}) \quad (3)$$

Where P (Object) = Probability that an object is present in the bounding box and IoU = Intersection over Union (how much the predicted box overlaps with the ground truth). During training, YOLOv5 optimizes its bounding-box predictions using a specialized loss function that measures how closely the predicted box aligns with the ground-truth annotation. The Bounding Box Loss evaluates the difference between the predicted center coordinates, width, and height and the actual values provided in the dataset.

$$\mathcal{L}_{\text{bbox}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right] \quad (4)$$

Where:

• S = Grid size (image divided into S × S grid).

• B = Number of bounding boxes per grid cell.

• $⊮^{obj}_{ij}$ = Indicator function, which equals 1 if object appears in cell i and bounding box j is responsible, and 0 otherwise.

• (x, y, w, h) = Center coordinates, width, and height of the bounding box.

• ˆx, ˆy, ˆw, ˆh = Predicted values.

YOLO also calculates the classification loss for correctly identifying the object's class (in this case, it is"barcode"). The classification loss is given by:

$$\mathcal{L}_{\text{class}} = \sum_{i=0}^{S^2} ⊮_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (5)$$

Where:

• pi(c) = Ground truth probability for class c in cells i.

• ˆpi(c) = Predicted probability for class c.

The final loss is a weighted sum of the above losses, which is expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{bbox}} + \mathcal{L}_{\text{objectness}} + \mathcal{L}_{\text{class}} \quad (6)$$

Where the objectness loss is given by:

$$\mathcal{L}_{\text{objectness}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \left[⊮_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} ⊮_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \right] \quad (7)$$

Where:

• $⊮^{noobj}_{ij}$ = Indicator function, which equals 1 if no object is present in bounding box j of cell i, and 0 otherwise.

• Ci = Actual confidence, ˆCi = Predicted confidence.

Model Loading and Deployment Setup

During deployment, the trained YOLOv5 model is loaded using the PyTorch Hub API. Although training is performed with GPU acceleration, inference is configured to run on the CPU to ensure compatibility with modern low-resource and edge systems, with performance scalable through additional optimization techniques.. Model weights are initialized once during startup to reduce loading time, and all tensor operations are executed on the CPU, providing real-time responsiveness without requiring specialized hardware.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Real-Time Frame Acquisition

Real-time video frames are captured through OpenCV's *VideoCapture()* interface. Each frame is preprocessed by resizing and normalizing the input before being passed to the YOLOv5 model. To maintain optimal frame rates, the system employs resolution tuning and optional frame skipping, ensuring a smooth balance between detection accuracy and processing speed.

Barcode Detection using YOLOv5

For each incoming frame, YOLOv5 predicts bounding boxes, class labels, and confidence scores.

- Bounding box coordinates (x, y, w, h)
- Class label: "barcode"
- Confidence score $p \in [0, 1]$

Predictions with confidence scores below 0.5 are discarded to avoid false detections. The validated bounding boxes are then drawn onto the frame using OpenCV's visualization functions. This stage ensures that barcode regions are accurately localized even under complex visual conditions.

Barcode Decoding using Pyzbar

Once the barcode region is localized, it is cropped and converted to grayscale to prepare it for decoding. The Pyzbar library is used to extract information from both one-dimensional (EAN-13, UPC, Code 128) and two-dimensional (QR, DataMatrix) barcode formats. Pyzbar performs pattern recognition and returns the decoded data, which is then overlaid onto the original video frame for real-time display. The decoding workflow is

1. **Scan:** Pyzbar identifies candidate patterns in the grayscale image.
2. **Extract:** Valid barcodes are decoded into text or numeric data.
3. **Overlay:** Decoded content is displayed on the video frame in real time.

Despite the fact that YOLOv5 can detect barcode locations with great accuracy, the bounding boxes formed around the barcode can sometimes be very close to its edges. Extreme cropping can cut off the adjacent area of the barcode known as the quiet zone, which is very critical for some decoding algorithms to get the right interpretation of the symbol. To solve this problem, a small margin is added to the bounding box detected before it is sent to the PyZbar decoder. In particular, a padding of about 10–20% of the bounding box size is added in all directions to make sure the quiet zone is not cut off. This simple post-processing procedure greatly strengthens the decoding process as it eliminates detection–decoding mismatches and hence reduces the cases where a barcode gets detected but fails to decode. The use of this padding setup assures that there will be better compatibility between the deep learning-based detection and the classical decoding stages in the proposed hybrid framework.

Result Display and System Termination

The acknowledgment of the computational load involved in using deep learning–based detection is not only an acknowledgement but also a promise that the proposed framework is flexible enough to fit in with different hardware limits. There are various optimization techniques that can increase the efficiency of the framework and lessen the resource consumption in the course of its adoption. Among these is post-training quantization, which converts model parameters to lower-precision formats (like INT8) to not only reduce memory usage and inference latency but also to have minimal effect on accuracy. Another powerful method is network pruning that works with the removal of weights and filters that do not contribute anything to the model, therefore, a compact model is left behind that keeps the essential feature representations and at the same time reduces the computational complexity.

Moreover, it will be easy to change the framework in a way that it would work with smaller and lighter detector variants like YOLOv5n or the newest nano-scale architectures which are targeted specifically for real-time performance on edge devices. Additionally, the hardware-accelerated inference engines like TensorRT and TensorFlow Lite can be employed to benefit from the platform-specific optimizations that, in turn, will lead to higher throughput and energy efficiency for the supported devices. These optimization methods make it possible for the proposed YOLOv5 + PyZbar framework to cover a wide range of deployment scenarios from high-performance systems through to mobile and embedded platforms with very little resources, all while reliable detection and decoding performance is preserved.

Computational Cost and Deployment Considerations

YOLOv5s is not only the lightest and the most efficient among all object detection models but its still a heavy model of the deep learning family that in comparison with the geometric or rule-based barcode localization techniques require much more computational resources. The latter are normally resorting to simple edge detection, thresholding, or contour analyses, which bring in very little processing overhead and can be done this way. On the other hand, neural network inference goes through many layers of floating-point operations, thus making it inherently - in the case of computing power and energy consumption - more demanding and costly. Hence, the first phase of the proposed YOLOv5s-based detection when it is operating on older smartphones or resource-limited embedded platforms may cause to increase the delay in inference and use of battery more than expected.

Nevertheless, the extra computing resources needed represent a deliberate and unavoidable trade-off that results in the vast detection performance in real-world operating conditions. In these conditions like the ones in the cases of warehouses, retail stores, and logistics centers, non-ideal conditions often characterizing such places can be the very cause of barcodes getting improperly captured like, poor lighting, low contrast between the barcode and the background, motion blur, partial occlusion, surface reflections, and perspective distortions. Under such circumstances, classical geometric methods very often miss the chance to localize the barcode region reliably. By tapping into the strong spatial representation and generalization capability of YOLOv5s, the proposed framework significantly boosts the detection reliability in such diverse and challenging conditions, hence leading to an increased possibility of successful decoding in the successive stages.

What is more, it should be mentioned that the framework is not primarily confined to one specific

RESULTS

The suggested hybrid framework handles detection and decoding as two separate but successive steps, thus their performances are assessed through different metrics. The detection accuracy of the YOLOv5 model is obtained through common object detection metrics, namely, precision, recall, and mean Average Precision (mAP) at IoU thresholds of 0.5 as well as 0.5:0.95, which represent the extent to which barcode regions are accurately identified in the original input images. On the other hand, the performance of the decoding is determined by taking the percentage of barcodes decoded correctly among those detected and forwarded to the PyZbar module as the measure. Therefore, this decoding accuracy indicates the performance of the classical decoding stage once the reliable localization is performed. In contrast, a PyZbar-only pipeline that attempts direct decoding from the full image is highly sensitive to background clutter, orientation, and noise, whereas the proposed YOLOv5 + PyZbar approach significantly improves decoding success by first ensuring accurate spatial localization.

The dataset was manually annotated with bounding boxes for every barcode instance to ensure accurate supervision during training. After annotation, the dataset was divided into an 80% training set and a 20% validation set, enabling balanced learning and reliable performance evaluation. All barcode formats were assigned equal importance to maintain robustness across different symbologies. Each image included a single class label, “barcode,” since the detection task focused solely on identifying barcode regions. This consistent labeling strategy ensured that the model learned to generalize effectively across diverse barcode types and imaging conditions. The performance of the barcode detection system was assessed using several standard

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

metrics: training and validation losses, precision, recall, and mean Average Precision (mAP). In addition, visual outputs on diverse test cases were evaluated to determine the system's generalization capability.

Training and Validation Losses

The model demonstrated effective learning behavior as seen in the continuous and stable decline of all loss components over the training epochs. These include:

Box Loss (train/box loss & val/box loss): Decreased steadily, indicating improved accuracy in predicted bounding box coordinates.

Objectness Loss (train/obj loss & val/obj loss): Showed consistent decline, reflecting the model's growing confidence in actual barcode presence.

Classification Loss (train/cls loss & val/cls loss): Remained near zero due to the presence of only a single class (barcode), as expected.

Figure 2 shows the training and validation loss curves over epochs, affirming that the model converges effectively without over fitting.

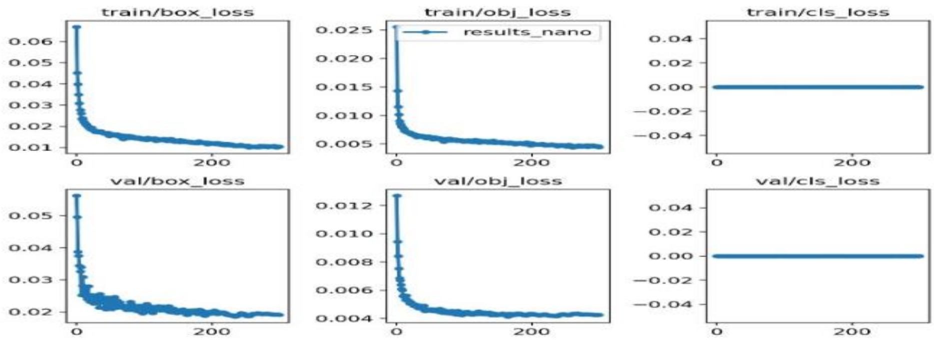

Figure 2: Training and validation loss curves over epochs.

Precision and Recall Analysis

Figure 3 shows the convergence of evaluation metrics across the training cycle, where both precision and recall rise rapidly during the initial epochs and stabilize close to 1.0. This indicates that the model produces very few false positives while successfully detecting nearly all barcode instances. Such consistently high precision and recall values demonstrate the robustness and reliability of the proposed system under diverse testing conditions.

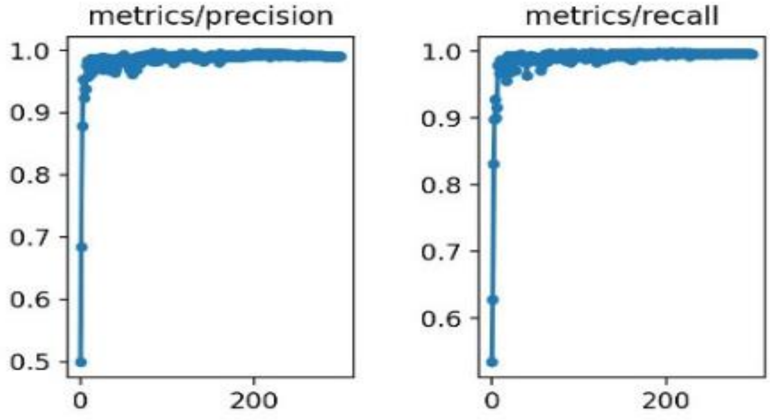

Figure 3: Precision and recall convergence over training epochs

Mean Average Precision (mAP)

Figure 4 illustrates the progression of mAP metrics throughout training. The mAP@0.5 value approached 1.0, indicating highly accurate alignment between predicted and ground-truth bounding boxes. Similarly, mAP@0.5:0.95 showed steady improvement and stabilized above 0.85, demonstrating strong performance under stricter IoU requirements. These results highlight the model's strong generalization capability and its effectiveness in achieving precise localization across diverse barcode scenarios.

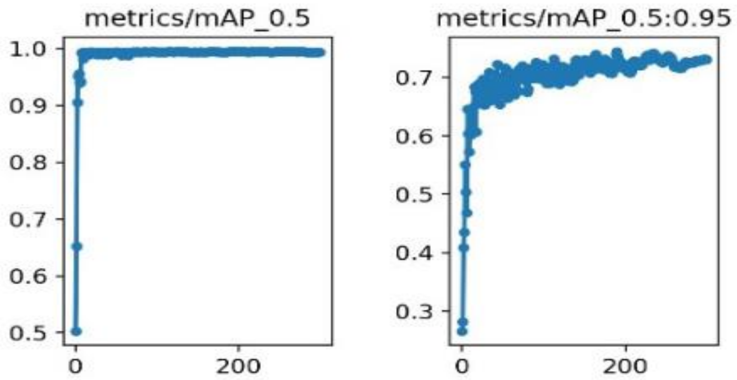

Figure 4: Evolution of mean average precision (mAP) metrics over epochs.

The proposed hybrid system, which integrates YOLOv5 for barcode localization with Pyzbar for decoding, demonstrated efficient convergence without signs of overfitting, while maintaining high detection accuracy and very few false positives. It also provided reliable decoding performance even under challenging imaging conditions. This balanced combination of robust detection and lightweight decoding makes the approach well-suited for real-time applications in retail, warehouse automation, and logistics.

Visual Results

To provide a clear understanding of the performance of our barcode detection pipeline, we present a series of visual results showcasing various detection scenarios. Figure 5 illustrate the output of our custom-trained YOLOv5 model when applied to a range of test samples containing different barcode types and conditions. Each image demonstrates the model's ability to accurately locate and extract bar-code regions, even under challenging conditions such as rotation, partial occlusion, or low resolution. The detected barcodes are highlighted with bounding boxes, and the successfully decoded values (when applicable) are displayed alongside. These results validate the robustness and generalization capability of our model in real-world settings, supporting its potential use in practical applications like inventory tracking, retail checkout systems, and logistics automation. In each case, the YOLOv5 model successfully localized the barcode regions, and the Pyzbar library decoded the contents (when readable) with high reliability. Detected barcodes were annotated with bounding boxes, and decoded information was overlaid near each box for easy interpretation.

Figure 5: Visual results of barcode detection on test images.

DISCUSSION

The proposed YOLOv5 + Pyzbar model is compared with FWL-YOLO [21], Barcode Decoder (YOLOv8) [22] and Smart Inference CNN [23]. Recent studies by Y. Qu *et al.* introduce FWL-YOLO, a lightweight barcode detection model optimized for express delivery waybills. Their work demonstrates that compact architectures can achieve high accuracy even on low-quality, cluttered logistics documents. This research highlights the growing trend toward real-time, resource-efficient barcode recognition systems for large-scale delivery operations [21]. D. Max (2025) proposed a YOLOv8-based barcode detection system that leverages modern object detection capabilities for fast and accurate localization. His implementation demonstrates the effectiveness of integrating state-of-the-art detectors with practical decoding workflows in real-time applications. The project highlights how open-source solutions can significantly simplify barcode recognition pipelines for developers and researchers [22]. T. Do *et al.* (2020) proposed a multi-digit CNN approach for efficient barcode decoding, emphasizing smart inference strategies to improve speed and accuracy. Their method effectively handles diverse barcode formats and partial occlusions in real-world images. This work demonstrates the potential of deep learning models for precise and scalable barcode recognition in practical applications [23]. Table 1 shows the Comparison of different object detection and barcode decoding models, highlighting their performance and efficiency. The comparative analysis in Table 1 indicates that the proposed YOLOv5 + Pyzbar framework outperforms other models in both detection and decoding metrics. It achieves the highest precision (98.1%) and recall (97.8%), demonstrating its ability to accurately localize barcode regions while minimizing false positives. The mAP values at both 0.5 (99.2%) and 0.5:0.95 (91.3%) thresholds further confirm its robust detection performance. Additionally, the decoding accuracy of 95.4% highlights the effectiveness of combining YOLOv5 with Pyzbar for reliable real-time barcode recognition, surpassing other lightweight and CNN-based approaches.

Table 1: Comparison of Object Detection and Decoding Models

Model	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**	**Decoding Accuracy**
Proposed YOLOv5 + Pyzbar	**98.1**	**97.8**	**99.2**	**91.3**	**95.4**
FWL-YOLO (Qu *et al.*, 2025)	95.2	94.5	96.0	88.5	92.0
YOLOv8 Decoder (Max, 2025)	94.8	93.9	95.5	87.0	90.5
Multi-digit CNN (Do *et al.*, 2020)	93.5	92.7	94.0	85.2	89.0

CONCLUSION

In this project, a robust and efficient barcode recognition system was developed specifically for waybills used in logistics. By combining the YOLOv5 deep learning model for accurate barcode detection with the pyzbar library for decoding, the system successfully automates the extraction of critical shipment information from static images. The results demonstrate high detection accuracy and reliable decoding performance, even under challenging conditions like poor lighting, skewed angles, or partial occlusions. This hybrid approach not only enhances automation in logistics workflows but also reduces the need for manual scanning and intervention, making it extremely useful for practical use cases like shipment tracking and warehouse management, especially on modern computing platforms. A key contribution of this work is the separation of barcode recognition into two stages: locating the barcode using a deep learning model and decoding its content using a traditional algorithm. YOLOv5 is used to reliably detect barcode regions even in difficult conditions such as poor lighting, cluttered backgrounds, or partial occlusion, while PyZbar is applied to decode the information from the detected region. Although this hybrid approach requires slightly more computational resources than classical methods alone, it provides better robustness and higher detection and decoding accuracy. In future work, the system can be further optimized for mobile and embedded devices by using lighter detection models, model quantization, and hardware-accelerated inference.

Declarations

Author contribution Information

All authors are equally contributed.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The authors does not receive any Fund

Data Availability statement

Throughout the research used the public available data set which is cited in 20.

REFERENCES

1. Mashat, A. (2024). A QR code-enabled framework for fast biomedical image processing in medical diagnosis using deep learning. BMC Medical Imaging, 24(1), 198.
2. Liu, Y., Zhang, S., Jin, L., Xie, L., Wu, Y., & Wang, Z. (2019). Omnidirectional scene text detection with sequential-free box discretization. arXiv preprint arXiv:1906.02371.
3. Pappula, K. K., & Anasuri, S. (2024). Deep Learning for Industrial Barcode Recognition at High Throughput. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(1), 79-91.
4. Farhaoui, Y., & El Allaoui, A. (2024, April). Bar Code Detection for AI-Driven EDA and Big Data Analytics. In The International Workshop on Big Data and Business Intelligence (pp. 476-488). Cham: Springer Nature Switzerland.
5. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
6. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
7. Jocher, G., Chaurasia, A., & Qiu, J. Y. (2020). Ultralytics. YOLOv5: PyTorch Implementation of Object Detection Models. GitHub Repository.
8. Pyzbar Developers, 'Pyzbar: Python Barcode Reader Library," GitHub Repository, 2020. Retrieved from https://github.com/NaturalHistoryMuseum/pyzbar.
9. Aggarwal, M., Sahoo, P., Saha, S., & Das, P. (2023). Machine learning-mediated ultrasensitive detection of citrinin and associated mycotoxins in real food samples discerned from a photoluminescent carbon dot barcode array. Journal of Agricultural and Food Chemistry, 71(34), 12849-12858.
10. Tiwari, S., Moharana, S., Mohanty, D. P., Vala, V., Thakur, B. N., & Jagdishbhai, J. A. (2024, July). Synthetic Data Generation and Detection of Barcode QRcode for Mobile Screenshots. In 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) (pp. 1-6). IEEE.
11. Bharath, G. V., & Kalaiarasi, K. (2025, July). Improving accuracy in barcode detection using mobile net algorithm compared to you only look once algorithm. In AIP Conference Proceedings (Vol. 3300, No. 1, p. 020254). AIP Publishing LLC.
12. Vezzali, E., Bolelli, F., Santi, S., & Grana, C. (2024, December). BarBeR-barcode benchmark repository: implementation and reproducibility notes. In International Workshop on Reproducible Research in Pattern Recognition (pp. 85-95). Cham: Springer Nature Switzerland.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

13. Qiao, L. C., Wang, J. L., Gao, B. H., Yang, X. G., Feng, W. T., Zhang, Y. X., ... & Liu, W. (2021, December). Efficient 1D barcode localization method for imagery shipping label using deep learning models. In 2021 12th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP) (pp. 119-124). IEEE.
14. Kim, D., & Kang, J. (2022, January). Barcode Image Identification Based on Maximum a Posterior Probability. In 2022 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-2). IEEE.
15. Vijayaraj, J., Rithikravivarma, V., & Siddharth, S. (2025, February). Smart Barcode Scanning using X-ray Technology with Computer Vision and Machine Learning. In 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL) (pp. 1495-1498). IEEE.
16. Almabruk, T., Adalhamid, S., & Maziq, A. (2025, February). Optimizing Barcode Detection and Recognition with YOLOv5 and Dynamic Barcode Reader Techniques. In 2025 IEEE 22nd International Multi-Conference on Systems, Signals & Devices (SSD) (pp. 732-737). IEEE.
17. Kamnardsiri, T., Charoenkwan, P., Malang, C., & Wudhikarn, R. (2022). 1D barcode detection: Novel benchmark datasets and comprehensive comparison of deep convolutional neural network approaches. Sensors, 22(22), 8788.
18. Linan, W., Jiahang, L., Ryzhova, V. A., Yaryshev, S. N., & Djamiykov, T. S. (2022, September). Development of a Network Camera with a Built-in Mask and QR Code Recognition Function. In 2022 XXXI International Scientific Conference Electronics (ET) (pp. 1-4). IEEE.
19. Ilchuk, M. S. (2024). A ROBUST BARCODE DETECTION SYSTEM WITH MACHINE LEARNING. ВЧЕНІ ЗАПИСКИ, 6202463.https://zenodo.org/records/7465864
20. Qu, Y., & Zhang, F. (2025). FWL-YOLO: a lightweight model for barcode detection and recognition in express delivery waybills. Signal, Image and Video Processing, 19(3), 261.
21. D. Max, "Barcode Decoder: A YOLOv8-Based System for Barcode Detection," GitHub Repository, 2025
22. T. Do, Y. Tolcha, T. J. Jun, and D. Kim, "Smart Inference for Multi digit Convolutional Neural Network based Barcode Decoding," arXiv preprint arXiv:2004.06297, 2020