IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 3, Ser. 4 (May. – June. 2024), PP 22-26
www.iosrjournals.org

Integrating YOLO-v5 and Pyzbar for Accurate and Efficient Barcode Recognition System

Tahani Almabruk[1], Mohammad Gerbade[2], khaled Mohammed[2]
[1]*(Department of Computer Science/ Omar Almukhtar University, Libya)*
[2]*(Department of Computer Science/ Libyan Academy, Libya)*

Abstract:
Accurate and efficient barcode recognition is essential across various logistics, retail, and healthcare industries, as it enables automated tracking, inventory management, and quick transactions. Introducing deep learning techniques has considerably enhanced the accuracy and speed of barcode recognition systems. This paper explores using advanced object detection and decoding algorithms to improve barcode recognition. By employing the YOLO-v5 algorithm, which achieved a mean average precision of 97% at an intersection over the union threshold of 0.5 and 98% over thresholds from 0.5 to 0.95, the system accurately identifies barcode regions within large datasets. The Pyzbar library complements YOLO-v5 by demonstrating a decoding accuracy rate of 91%, effectively converting detected barcodes into readable text. This integrated approach ensures robust and reliable barcode detection while maintaining high decoding efficiency, making it appropriate for use in scenarios necessitating both precision and operational efficiency. The findings highlight the potential of merging YOLO-v5's detection capabilities with Pyzbar's decoding accuracy, establishing a new standard for future research and applications in barcode recognition technology.
Key Word*: Deep Learning; Barcode recognition; YOLO-v5; Pyzbar.*

Date of Submission: 09-06-2024 Date of Acceptance: 21-06-2024

I. Introduction

Barcode recognition systems are essential across numerous industries, assisting with the efficient management of stocks, product tracking, and the checkout counter processes. While traditional barcode recognition technologies have proven effective, they often struggle with speed, accuracy, and adaptability limitations to various environments and barcode conditions. The emergence of advanced deep learning techniques and robust libraries presents an opportunity to improve these systems significantly. This study investigates the integration of YOLO-v5 and Pyzbar to create a highly accurate and efficient barcode recognition system.

YOLO-v5 is recognized for its lightning speed and efficient item recognition. In contrast to traditional methods that typically require multiple processing stages, YOLO-v5 utilizes an individual neural network to immediately forecast bounding boxes and the probability of categories, enhancing its efficiency and making it ideal for real-time applications. By leveraging YOLO-v5's capabilities, we aim to improve the detection and localization of barcodes in diverse and complex scenes [1].

In addition to YOLO-v5, Pyzbar is a specialized library for decoding barcodes from images. It supports a wide range of barcode formats, such as QR codes, EAN, and UPC, making it a versatile solution for barcode recognition. Pyzbar's lightweight and straightforward approach to decoding barcodes allows it to integrate seamlessly with YOLO-v5's object detection capabilities, forming a comprehensive system that can accurately identify and decode barcodes with high precision [2].

Integrating YOLO-v5 with Pyzbar tackles several key challenges in barcode recognition. By merging YOLO-v5's robust object detection with Pyzbar's efficient decoding, the proposed system seeks to attain high accuracy under diverse conditions, such as poor lighting, varied orientations, and partially obscured barcodes. This combined approach capitalizes on the strengths of both technologies, boosting overall system performance and reliability.

This paper seeks to offer an in-depth analysis of the integration process, assessing the system's performance through rigorous testing in real-world scenarios. We will outline the methodologies used, present experimental findings, and compare the proposed system with current barcode recognition solutions. Through this research, we aim to advance the development of more efficient and accurate barcode recognition systems, benefiting numerous applications across various industries.

II. Related Work

Recent progress in deep learning and computer vision have markedly enhanced the accuracy and efficiency of barcode recognition systems. Numerous studies have examined various aspects of barcode detection and recognition, employing innovative algorithms with a focus on improving performance in real-world scenarios.

This section highlights numerous studies that collectively emphasize the significant progress in barcode recognition technology achieved through the use of deep learning algorithms, especially YOLO-v5. By improving accuracy, efficiency, and adaptability to different conditions, these approaches are set to optimize operations in logistics, autonomous driving, and other industrial environments, showcasing the versatility and robustness of YOLO-v5 in practical applications.

Teerawat Kamnardsiri et al. (2022) examine the effectiveness of various deep convolutional neural network (D-CNN) methods for 1D barcode detection using two newly developed datasets: InventBar and ParcelBar. These datasets simulate real-world conditions, incorporating distortions such as varying lighting, complex backgrounds, rotations, and blurriness. The study evaluates several D-CNN models, including YOLO-v5, YOLOx, EfficientDet, RetinaNet, and Faster R-CNN. The findings indicate that YOLO-v5 is the most robust model, achieving the highest mean average precision in terms of both accuracy and speed across the different datasets [3].

In [4] a novel method for 1D barcode detection that merges deep learning with geometric techniques is introduced. This method combines the rapid localization abilities of deep learning with the precise segmentation provided by geometric methods, enabling it to manage complex backgrounds and high-resolution images effectively. Experimental results demonstrate that this integrated approach surpasses existing sophisticated methods by at least 5% in accuracy and operates more quickly, making it ideal for real-time applications in various industrial settings.

Yi Luo and Jiaxing Chen (2022) designed a QR code recognition system utilizing the YOLO-v5 algorithm. Their study emphasizes the application of LabelImg for dataset annotation and the use of data enhancement techniques to boost model performance. The YOLO-v5s model was chosen due to its optimal balance between accuracy and computational efficiency. The findings from the experiment demonstrate that the system reaches an average precision rate of approximately 90%, highlighting its effectiveness in real-time QR code recognition tasks, particularly in augmented reality applications [5].

Xia Zhu (2021) introduces an efficient algorithm for barcode localization and recognition using the YOLOv-5S network. This research primarily aims to enhance the automation and intelligence of logistics systems. The proposed system consists of two main components: a barcode localization algorithm that uses YOLOv-5S for framing and cutting the barcode, and a barcode recognition algorithm that employs the ZBAR algorithm for decoding. Experimental results show that this approach significantly improves barcode recognition accuracy and efficiency, even under varying lighting conditions and image distortions. This advancement is expected to streamline logistics operations by reducing errors and increasing processing speed [6].

III. The Methodology

This section presents the methodology for developing and assessing the proposed automatic barcode recognition system. In addition, it describes its experimental setup design.

Proposed System

In the proposed system, the collected dataset undergoes preprocessing procedure before initiating the barcode detection stage. In this stage, YOLO-v5 is used to locate barcodes within the dataset images. Following this, the efficient Python library Pyzbar is employed to recognize the characters extracted from the barcodes.

Barcode Detection Using YOLO-v5: YOLO-v5 was released in 2020 by Ultralytics. It marked a significant upgrade to the evolution of the YOLO series (i.e. You Only Look Once) known over the years to perform real-time object detection. The model was built using PyTorch framework and included native and several other folders for speed and accuracy enhancement. One crucial innovation was the introduction of an AutoAnchor algorithm, which optimally modifies the anchor boxes to overlap better with the training dataset. Architecture' modifications on the CSPDarknet53 backbone with the Stem layer make it less memory and computation, besides an SPPF layer to fasten the computation by pooling the features into fixed-size maps. In addition, part of this model's neck is covered by the SPPF and a modified CSP-PAN; furthermore, its head is based on YOLO-v3 for efficient object detection. Other augmentations, including Mosaic, MixUp, and HSV, further enhance YOLO-v5 robustness during training. YOLO-v5 is offered in five scaled versions—from the lightweight YOLO-v5n to the high-performance YOLO-v5x—catering to various application needs and hardware constraints [7].

Barcode Decoding Using Pyzbar: Pyzbar is a library designed for the easy and efficient decoding of both 1D and 2D barcodes. It identifies the type of barcode and extracts the data it contains. This decoded information can then be processed for various purposes, such as extracting textual data, validating the barcode content, or logging the results for further analysis. Pyzbar supports a wide range of barcode types, including QR codes, EAN, UPC, and Code128, making it a versatile tool for numerous applications, including inventory management, document processing, and automated data entry systems [8] [9].

Experimental Setup

This study was conducted to assess the effectiveness of the automatic barcode recognition system, taking advantage of the strengths of both YOLO-v5 and Pyzbar. The setup consisted of several critical components: dataset gathering, preprocessing, model training, model evaluation measures, and experimental technique.

Dataset Collection: We have compiled a diverse dataset comprising of 1,633 images of 1D barcodes taken by mobile cameras. This collection includes images of barcodes rotated at various angles and features a variety of everyday products. The dataset effectively simulates real-world conditions for product barcodes, with images taken under varying conditions to enhance diversity. Among the 1,633 images, some are of high quality, while others exhibit significant degradation. Additionally, the dataset includes images captured under both favorable and unfavorable lighting conditions, further contributing to its robustness and variability.

Dataset Preprocessing: The Roboflow platform was utilized for dataset preprocessing, leveraging its extensive array of augmentation techniques to enhance the dataset through various transformations on the existing images. Data augmentation proved advantageous by mitigating overfitting, improving the model's generalization capabilities, and enhancing the robustness of the machine learning models. The augmentation techniques implemented in this study included the addition of noise (up to 0.3%), blurring (up to 2.5 px), rotation (within a range of -15° to +15°), resizing images to dimensions of 640x640, and adjusting brightness (from -24% to +24%). As a result, the size of the target dataset increased from 1,633 to 3,900 barcode images.The initial step in preparing a model for barcode identification involves annotating images using the labeling tool provided by Roboflow. During this process, markers are placed on the barcode region within each image. This step is vital as it involves instructing the model by labeling the training data, in practice is known as "supervised learning." In supervised learning, the model is trained on pre-labeled examples, enabling it to learn and identify patterns for precise barcode recognition.

Model Training and Optimization: YOLO-v5 is developed using the PyTorch deep learning framework, recognized for its rapidly changing computational graph and ease of use, making it suitable for both research and production environments. PyTorch's flexibility and broad community support have contributed to its popularity in developing and deploying deep learning models. During the training phase, various parameters were optimized, including the learning rate, batch size, and augmentation strategies, to enhance model performance. The training process was conducted on a GPU-accelerated workstation, which significantly expedited the training procedure.

Evaluation Metrics. Mean Average Precision (mAP) is utilized as the evaluation metric for object detection models, providing a comprehensive performance measure by integrating precision and recall across various intersections over union (IoU) thresholds. The calculation of mAP involves several key steps. Firstly, precision, defined as the ratio of true positive detections to the total number of detections, assesses the accuracy of the detected objects. Recall, the ratio of true positive detections to the total ground truth objects, evaluates the model's ability to identify all instances of objects in the dataset. Secondly, a precision-recall curve is plotted for each class by varying the confidence threshold, illustrating the trade-off between precision and recall. Thirdly, each class's Average Precision (AP) is calculated as the area under the precision-recall curve, summarizing the model's capability to balance precision and recall for that specific class. Finally, the mean average precision (mAP) is derived by averaging the AP values across all classes, offering a singular metric that reflects the model's overall performance. The formula for mAP is [10] [11]:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i,$$

where N represents the number of classes and AP_i is the average precision for class i. mAP is often reported at different IoU thresholds, such as mAP @0.5, which calculates the metric at a single IoU threshold of 0.5, and mAP @0.5:0.95, which averages the AP values over multiple IoU thresholds from 0.5 to 0.95 in increments of 0.05. This extended range provides a more thorough evaluation of the model's detection capabilities.

Experimental Procedure: A Nano version of the YOLO-v5 (with training lasting for 100 epochs) was employed in the barcode detection stage. This technique randomly split the annotated dataset into training, validation, and testing sets, with 70%, 20%, and 10% respectively.

IV. Results and Discussion

The YOLO-v5 model was trained and tested in the proposed system to localize barcodes within the specified dataset, consisting of 3,900 annotated images. The achieved outcomes are presented in Table no 1.

Table no 1: Presents the mAP of YOLO-v5 and the accuracy of Pyzbar on the targeted dataset.

Barcode Dataset size	Barcode Detection using YOLO-v5		Barcode Decoding using Pyzbar
3900	mAP @0.5	mAP @0.5:0.95	91%
	97%	98%	

The research assesses the system's efficiency through two critical phases: barcode detection and decoding. As outlined in Table 1, the YOLO-v5 algorithm excels in the barcode detection phase, reaching a mAP of 97% at an IoU threshold of 0.5 and a mAP of 98% over a range of IoU thresholds from 0.5 to 0.95. These elevated mAP scores highlight the model's ability to accurately pinpoint barcode areas within a dataset of 3900 images, ensuring dependable detection capabilities. Following this, in the barcode decoding phase, the Pyzbar library achieves an impressive accuracy rate of 91%, demonstrating its proficiency in accurately translating detected barcodes into readable formats. The combination of YOLO-v5's precise detection prowess and Pyzbar's high decoding accuracy culminates in a highly reliable and efficient barcode recognition system, perfectly suited for applications demanding accuracy and efficiency.

The dataset size could significantly challenge the study's validity, given that it originally comprised just 1633 images, later expanded to 3900 through annotation. The authors are addressing this limitation by considering data augmentation methods to create additional data instances. Another potential solution might be to explore transfer learning techniques, allowing the model to be trained on multiple datasets.

Figure 1 Samples of the utilized datasets where the YOLO-v5 successfully identified the barcode.

V. Conclusion

The present study highlights the effectiveness of integrating advanced object detection and decoding algorithms to achieve superior barcode recognition. The YOLO-v5 algorithm, demonstrating an impressive mAP of 97% at an IoU threshold of 0.5 and 98% across a range of IoU thresholds from 0.5 to 0.95, excels in accurately identifying barcode regions within a large dataset. Additionally, the Pyzbar library achieves a remarkable decoding accuracy rate of 91%, efficiently converting detected barcodes into readable text. This combined system not only ensures robust and reliable barcode detection but also maintains high efficiency in decoding, making it well-suited for various practical applications requiring both accuracy and operational efficiency. The synergy between YOLO-v5's detection precision and Pyzbar's decoding accuracy underscores the potential for developing reliable and efficient barcode recognition systems, setting a high standard for future research and applications in this domain.

References

[1] Ultralytics. "YOLOv5 Documentation." https://docs.ultralytics.com/yolov5/ (accessed June 13, 2024).

[2] P. Maske, K. Pardhi, S. Hatzade, R. Sharma, and S. Wazalwar, "Image Security Barrier (ISB): Hide valuable information in image using machine learning," in *2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22)*, 2022: IEEE, pp. 1-6.

[3] T. Kamnardsiri, P. Charoenkwan, C. Malang, and R. Wudhikarn, "1D Barcode Detection: Novel Benchmark Datasets and Comprehensive Comparison of Deep Convolutional Neural Network Approaches," *Sensors,* vol. 22, no. 22, p. 8788, 2022.

[4] Y. Xiao and Z. Ming, "1D Barcode Detection via Integrated Deep-Learning and Geometric Approach," *Applied Sciences,* vol. 9, no. 16, p. 3268, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/16/3268.

[5] Y. Luo and J. Chen, "Two-Dimensional codes recognition algorithm based on Yolov5," *Academic Journal of Computing & Information Science,* vol. 5, no. 7, pp. 68-72, 2022.

[6] X. Zhu, "Design of barcode recognition system based on YOLOV5," in *Journal of Physics: Conference Series*, 2021, vol. 1995, no. 1: IOP Publishing, p. 012052.

[7] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction,* vol. 5, no. 4, pp. 1680-1716, 2023.

[8] Pyzbar. "Pyzbar: A Python library for decoding barcodes." https://pypi.org/project/pyzbar/ (accessed 10, June, 2024).

[9] OpenCV. "OpenCV: Open Source Computer Vision Library." https://opencv.org/ (accessed 10, June, 2024).

[10] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*, 2020: IEEE, pp. 237-242.

[11] M. Zhang and L. Yin, "Solar cell surface defect detection based on improved YOLO v5," *IEEE Access,* vol. 10, pp. 80804-80815, 2022.