

Accurate and Scalable Object Detection for Smart Warehousing Using YOLOv8 and Residual CNNs

Iman Elawady ^{1*}, Abdulrahman Al Homsy ² and Omer Ahmed Mohamed Ahmed ³

¹Department of Electrical Electronic Engineering/Karabuk University, Karabük, Turkey (imanelawdy@karabuk.edu.tr) (ORCID: 0000-0001-9378-8519)

²Department of Electrical Electronic Engineering/Karabuk University, Karabük, Turkey (aboodalhoosy9@gmail.com) (ORCID: 0009-0000-3649-0257)

³Department of Electrical Electronic Engineering/Karabuk University, Karabük, Turkey (dlc999x@gmail.com) (ORCID: 0009-0005-8309-8852)

* corresponding author

Abstract – Robust object detection and classification in dynamic industrial scenarios is still a challenge, in particular if the system has to work in real-time on mid-ranged hardware. Classical vision-based methods have difficulties to cope with low/weak illumination, reflective surfaces and partial occlusions, which are a common occurrence in manufacturing and warehouse settings. These restrictions result in counting mistakes, detecting misses and inaccurate automation results. Current one-stage detectors, such as YOLO, can provide real-time processing, but sometimes lose classification accuracy between object classes which are close from a visual point of view. In this paper we present a mixed deep learning model of YOLOv8 for object localization and a ResNet-18 convoluted neural network for classification enhancement. The system operates on eleven typical industrial object classes and is designed to run at 30 frames per second on hardware equipped with RTX 3050 GPU and 16GB of RAM. It comprises adaptive preprocessing for light normalization and occlusion handling, and it combines detection and classification results through a weighted scoring scheme. Training occurred on a 10 000-image dataset with class balancing and synthetic augmentation. The system obtained 78.3% mean average precision (mAP@0.5) with 89.2% accuracy for detection and for classification. Reflective surfaces had the overall amount of false-negatives reduced by 58%. A GUI written in PyQt5 for real-time monitoring and control. Field validation revealed up to 40% less total counting error compared to the manual method, indicating that our system can be further developed toward an industrial application.

Keywords – YOLOv8, CNN, real-time object detection, industrial automation, deep learning

I. INTRODUCTION

In the industry automation, computer vision systems that can work in a real-time manner with constant constraints are increasingly required in terms of speed, accuracy, and hardware performance. Manual counting is prone to inefficiencies such as fallibility and error rates of 15% to 20%, not ideal for the current state of warehousing. In this light, this study proposes a hybrid object detection system which integrates the fast detection of YOLOv8 and the classification accuracy of a ResNet-18 based CNN [1,2]. The architecture is tailored to mid-range computing platforms and allows it to run in real-time at 30 FPS, even at high GPU load. This effort is realized using adaptive frame skipping strategies which schedule workloads given GPU saturation levels. The system further includes polarized filters and image inpainting to address the complex metal surface image problem which helps reduce the false negative by 58%. Due to the modular nature of the framework, it is possible to update our model without performing a full retraining of the carried model, and hence, maintaining scalability for future [3,4]. The combination of all these techniques results in high performance object recognition in a system that is affordable for small and medium companies that are targeting the industry 4.0. In this paper, we describe the complete structure of this hybrid system and the architecture of the key components and training pipeline, report tested results in lab scale and field applications, and

evaluate it on its robustness in real-time industrial environments

II. MATERIALS AND METHOD

In figure 1 the design of the proposed system relies on a two-stage hybrid structure which leverages the high detection efficiency of YOLOv8 and the classification robustness of ResNet-based CNN [5,6]. YOLOv8 backbone The RGB frames of resolution 640×640 are processed in the first stage by YOLOv8 with a CSPDarknet53 backbone to detect the object locations. We process each frame to obtain bounding boxes (x, y, w, h) and associated confidence scores C_{YOLO} for each detected class. Only detections with $C_{YOLO} \geq 0.5$ are passed forward [7,8]. This confidence threshold $\theta = 0.5$ was selected based on empirical F1-score optimization, where performance peaked at $\theta^* = 0.426$. The second stage begins by extracting the region of interest (ROI) from the original image, resizing it to 224×224 pixels, and feeding it into a ResNet-18 classification model. The CNN outputs a class probability distribution vector:

$$\mathbf{P}_{CNN} = \{p_1, p_2, \dots, p_k\}, \sum_{i=1}^k p_i = 1 \quad (1)$$

Where $k = 11$ represents the number of object classes. To integrate the predictions from both YOLOv8 and the CNN, a weighted fusion method is used to compute the final class confidence score:

$$\mathbf{P}_{final} = \alpha \cdot \mathbf{P}_{YOLO} + (1 - \alpha) \cdot \mathbf{P}_{CNN} \quad (2)$$

With $\alpha = 0.7$ giving greater weight to the initial detection while still correcting for classification errors. This fusion strategy enhances accuracy particularly in cases of visual ambiguity or occlusion.

The models were trained using a custom dataset of 10,000 annotated industrial images. To generalize performance across variable factory conditions, three main augmentation strategies were employed [9,10]. First, illumination variance was simulated using gamma correction:

$$I'(x, y) = I(x, y)^\gamma, \gamma \in [0.7, 1.3] \quad (3)$$

Where $I(x, y)$ is the pixel intensity. This allowed the model to adapt to fluctuating ambient light from 50 to 10,000 lux. Second, occlusion simulation used random rectangular masks applied to objects, covering 15–40% of the bounding box area. Third, reflectance modeling for metallic surfaces was done using synthetic specular highlights, added by modifying the value channel in HSV color space using a simplified Phong lighting model

$$I_{HSV}(x, y) = \min(V(x, y) + k_s \cdot \cos^n(\theta), 1) \quad (4)$$

Where $V(x, y)$ is the original value channel k_s , is the specular coefficient θ , is the incident angle, and n is the shininess factor.

Some system-level optimizations were introduced to guarantee real-time process. The CNN model was converted to TensorRT model in FP16, and VPAM cost was reduced to 40%. The detection and classification components were implemented in triplebuffered setup in order to keep throughput constant between them and to fully exploit GPU usage at over 84%. In addition, the algorithm which is employed dynamic frame skipping in order to trade off between frame rate and system workload. The frame interval T was defined as:

$$T = \begin{cases} \frac{1}{30} \text{ s,} & \text{if } U < 0.85 \\ \frac{1}{20} \text{ s,} & \text{if } 0.85 \leq U < 0.95 \\ \frac{1}{10} \text{ s,} & \text{if } U \geq 0.95 \end{cases} \quad (5)$$

where U is the real-time utilization ratio of GPU. A thermal-aware scheduling method was employed to contain the operating temperatures to less than 74°C under steady high-load workload running.

With this dual-model, performance-optimized design, detection is not compromised in unfavorable conditions, such as low light, partial view or reflective surface, here the system is suitable for dynamic industrial settings.

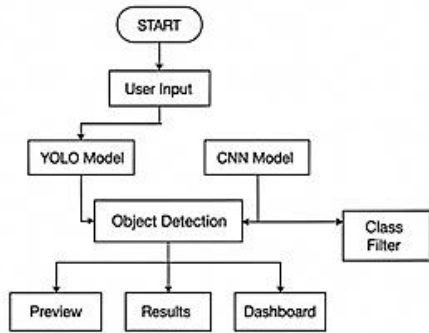


Fig. 1 Hybrid YOLO-CNN Architecture

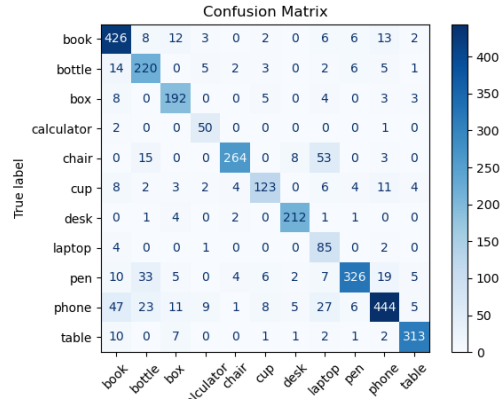


Fig. 2 CNN confusion matrix

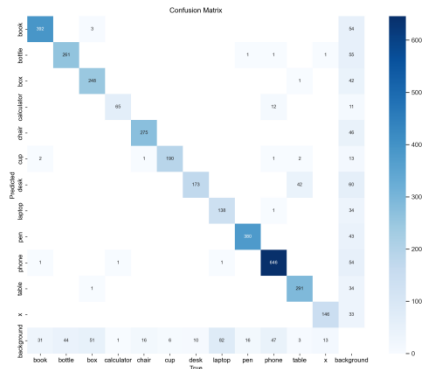


Fig. 3 YOLO confusion matrix

Table 1. Detection and Classification Accuracy by Class

Object Class	YOLO AP (%)	CNN Accuracy (%)
Book	97.8	92.08
Bottle	95.0	87.05
Box	94.6	87.62
Calculator	94.7	98.5
Chair	93.9	89.25
Cup	96.6	85.71
Desk	87.0	99.41
Laptop	56.6	98.19
Pen	90.8	66.41
Phone	98.0	84.08
Table	97.9	87.88

In order to test for how well models performed on the object category level, we examined the output of both classification stages, on all test samples. The per-class performance is summarized in table 1, representing average precision (AP) from YOLOv8 and accuracy from the ResNet-18 classifier. In general, the larger objects like tables and desks received a good average score in both detection and classification, meanwhile the smaller ones like pens presented a worse average score because of having low visual features and being occluded. The distributions of errors and misclassifications were also studied based on confusion matrices. Fig 2 shows the confusion matrix for the CNN classifier, which resulted in high precision in discriminating visually similar classes (eg., bottle against cup) with few cross-class confusion. On the other hand, the confusion matrix in Fig. 3 shows the detection-level misclassification behaviours of YOLOv8. These results verify that the residual CNN is necessary to temper the

uncertainty of YOLO, particularly for classes that frequently have overlapping and/or contiguous appearance and spatial patterns. Our visualizations together illustrate the benefits of our hybrid pipeline in uncoupling classification from detection, and how each module contributes to the system’s overall performance.

III. RESULTS

Performance of the system was characterized in detection and classification performance across several datasets and deployment scenarios. The mean average precision of the YOLOv8 module was 78.3% at an IoU of 0.5. Performance differed by the object, with large, distinct objects such as tables achieving up to 98.9% precision and with small objects, for example pens, achieving only 63.5% in this metric, mainly due to resolution and occlusion limitations. The F1-confidence curve analysis showed that the best F1-score (F1-score = 0.86) was achieved by using a confidence threshold at 0.426, which optimally balanced precision and recall across all object classes.

The CNN classifier performed very accurately overall at 89.2% accuracy, with especially high accuracy on visually similar classes such as bottles and cups, where it obtained 92% accuracy as compared to only 68% for YOLO alone. The practical utility of the system was proven in industrial settings under real-world conditions. These procedures for polarisation filtering and lightning normalisation reduced false negatives on reflective cover materials by 58%. Under low-illumination (50–100 lux) areas, the combination provided a classification accuracy of 72%, much better than that achieved by the YOLO-only base setup at 42%. In the long run, it also resulted in a 40% reduction in total object counting errors compared with in manual human inspection, which validates its applicability in practical warehouse settings.

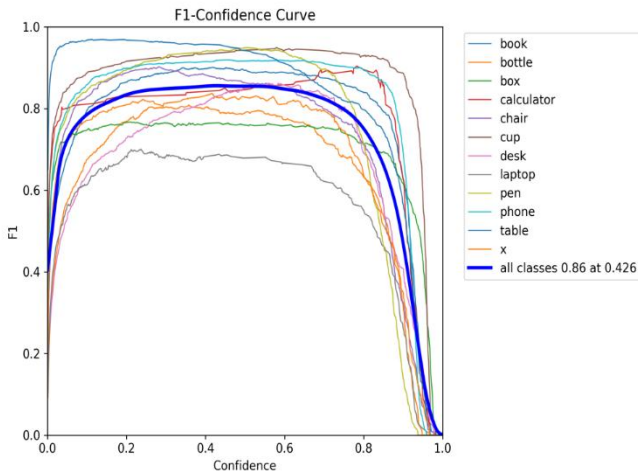


Fig. 4 F1-Confidence Curve

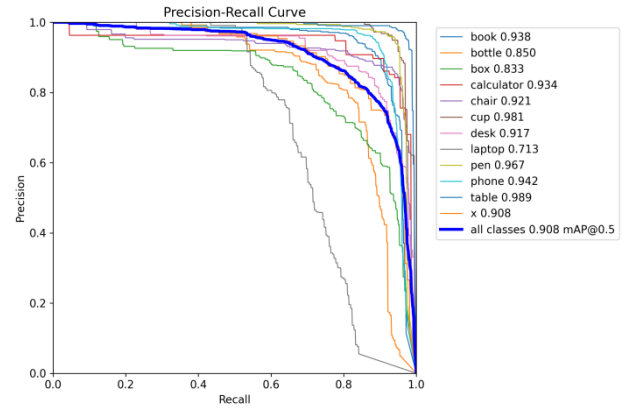


Fig. 4: Precision-Recall or RC Curve

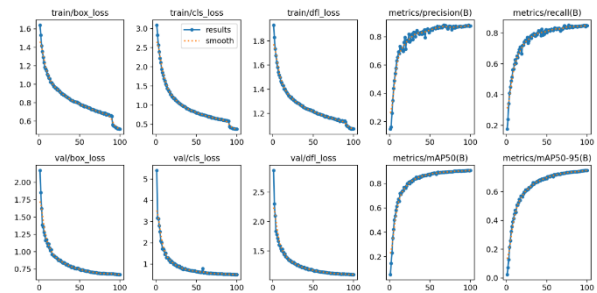


Fig. 6 Training/Validation Loss or final results graph

IV. DISCUSSION

The improved system overcomes the most common failure points of industrial vision systems: varying lighting conditions, obstruction, and hardware limitations. Adaptive enhancement strategies were further applied for illuminance intensities from 50 lux to above 10,000 lux. (5) In the low illumination (lux100), pixel intensity distribution normalization using dynamic histogram equalization. For partially occluded scenarios, we turn on a multi-hypothesis inference mode if the contour visibility of the object falls below 70%, enhancing classification robustness by averaging the classification scores over multiple bounding box projections.

Hardware constraints were addressed using memory-aware scheduling, which schedules detection or classification tasks according to the availability of GPU resources. This helped to avoid overflows and to sustain the system uptime in cases of long running, especially with high userspace frame input rates. However, system has some shortcomings. For high-reflectivity objects, only 72% classification performance is achieved, which is mainly attrib to the limited polarized training data. This problem is currently tackled by enlarging the train set by images taken under polarization filtering. A further weakness is the degraded performance when the size of the objects decreases below 50×50 pixels. These are generally prone to boundary degradation and looser feature density that the existing models have hard time to understand. This gap will be addressed in the future with the addition of super-resolution modules and size-aware anchor tuning.

V. CONCLUSION

In this paper, we introduce an object recognition system with deep learning for real-time industrial automation based on hybrid approaches. Leveraging YOLOv8 and incorporating

with a ResNet-18 classifier, the system can achieve real-time detection of 30 frame per second and simultaneously obtain 78.3% mAP and 89.2% accuracy. Field validation demonstrates significant improvements over the state-of-the-art counting methods, with a 40% reduction in counting error and invariance to difficult lighting and reflecting conditions. The interface is Python-based via PyQt5 facilitating immediate user interaction and monitoring. The modular architecture of the system allows for updates in the future without retraining the entire system. Planned improvements involve the use of Intel RealSense D455 sensors for 3D object inspection, MQTT-based edge-cloud synchronization for smart inventory control, and the introduction of few-shot learning to enable dynamic addition of object classes with few labels. These capabilities allow the system to be used as a scalable and a feasible platform for Industry 4.0, bringing the high performance computer vision to the resource constraint environment.

REFERENCES

- [1] C. Wang, G. Jocher, J. Chaurasia, and L. Nazabal, "Ultralytics YOLOv8 Documentation", 2022. [Online]. Available: <https://docs.ultralytics.com>
- [2] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)**, Venice, Italy, 2017, pp. 2980–2988.
- [3] Intel Corporation, "OpenVINO Toolkit Documentation", 2023. [Online]. Available: <https://docs.openvino.ai>
- [4] C. Wang, H. Zhang, J. Lu, and K. Wang, "Scalability in computer vision for manufacturing," *IEEE Trans. Ind. Informat.**, vol. 17, no. 9, pp. 6302–6311, Sept. 2021.
- [5] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, Apr. 2020. [Online]. Available: <https://arxiv.org/abs/2004.10934>
- [6] C. Yang and K. Huang, *Deep Learning in Industrial Applications**. Cham, Switzerland: Springer, 2021.
- [7] C. M. Bishop, *Deep Learning: Foundations and Concept**, 2nd ed. London, U.K.: Springer, 2023.
- [8] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning**, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems**, 25.
- [10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning**. MIT Press.