**EAI Endorsed Transactions**
**on Intelligent Systems and Machine Learning Applications** Research Article **EAI.EU**

# Lightweight and Real-Time Object Detection on Edge Devices: A Unified Framework for Resource-Constrained Environments

Hewa Majeed Zangana[1,*]

[1] Duhok Polytechnic University, Duhok, Iraq

## Abstract

Advances in edge computing have heightened the demand for object detection models capable of running efficiently on devices with constrained computational resources. This paper presents a robust hybrid detection framework that integrates template matching with Faster R-CNN to enhance detection accuracy in challenging conditions, such as occlusion, low lighting, and motion blur, while maintaining reasonable computational efficiency. Unlike conventional cloud-based detection, our approach reduces latency and improves data privacy. On the LASIESTA dataset, the proposed method achieves a mean Average Precision (mAP) of 88.2% at IoU 0.5 and 74.6% at IoU 0.75, outperforming Faster R-CNN by 4.3% in precision and 3.6% in recall. Although inference time increases modestly by 6 ms/frame compared to Faster R-CNN alone, the hybrid method consistently delivers superior robustness. While our implementation focuses on performance evaluation on a workstation, the framework design can be adapted for deployment on heterogeneous devices through additional optimization steps such as pruning and quantization. These findings demonstrate that combining classical localization techniques with deep learning models yields a practical and effective solution for real-time detection in resource-constrained environments.

**Keywords:** Edge Computing; Model Compression; Object Detection; Real-Time Processing; Resource-Constrained Devices.

Received on 28 April 2026, accepted on 01 May 2026, published on 12 May 2026

doi: 10.4108/eetismla.12814

## 1. Introduction

Repeated development in the field of computer vision and deep learning has magnified the accuracy and consistency of object detection in industries such as autonomous transportation, security, medical imaging, and Internet of Things instruments[1,2]. However, because of the restraints put on by edge devices, such as limited processing power, memory restraints, and rigid real-time demands, the deployment of similar systems in a successful manner remains challenging [3,4].

Algorithms such as R-CNN, Fast R-CNN and Faster R-CNN achieve the highest level of accuracy, but are processing capacity-hungry and can stand much in terms of processing capacity. [5]. Single stage methods such as YOLO and SSD were able to manage to accelerate the detections but they often cannot match the balance between accuracy, latency [6]. In response to this demand, it is in response to this demand that researchers have created lightweight architectures such as YOLO-LITE and MobileNet-SSD; however, these models often sacrifice precision—particularly with regard to small and hidden

---

[*] Corresponding author. Email: hewa.zangana@dpu.edu.krd

objects [7–9]. Although there are a multitude of studies that have examined lightweight object detection, the advertisements are mostly unable to provide a flexible and real-time system for the heterogeneous edge devices. Many models have challenges adapting and generalizing well across environments because of their hardware-specific or dataset-specific optimization [10,11]. Additionally, the trade-off necessary between model performance and efficiency[12]. Despite the existing gap, in terms of research, in regards to the lack of any all-encompassing solution balancing accuracy and resource efficiency for various devices, our work seeks to fill that gap [13,14]. This study's objective is to develop a robust hybrid detection framework that enhances detection performance in complex scenes, while maintaining reasonable computational efficiency. While our experiments were conducted on high-end hardware (RTX 3080) for evaluation purposes, the framework is designed to be adaptable for deployment on various platforms, including edge devices, through future optimizations. The main aims and achievements of this research are presented below:

- Design and Integration: Integrate a model compression, attention-based feature optimization, and adaptive quantization approach, to optimize memory and resource consumption without affecting performance.
- Hardware Versatility: Support stable performance on a diverse set of edge devices such as NVIDIA Jetson Nano, Raspberry Pi, and ARM mobile phones, addressing challenges due to various architecture.
- Performance Optimization: Noticeable margins of improvement in both inference speed and memory-footprint with great detection accuracy on real-world datasets, just as [15,16].
- Comparative Benchmarking: Perform comprehensive assessments against popular lightweight counterparts proving a performance gain with regards to mAP and lower runtime needs [6,17,18].

What sets this research apart from earlier hybrid detection methods, including [19], is the systematic integration of template-guided proposal generation with a deep learning backbone in a modular framework that can adapt to diverse deployment environments. Unlike prior works that combine template matching and deep networks primarily for fixed-scene or controlled settings, our approach incorporates a flexible confidence fusion mechanism and an adaptive proposal weighting strategy. Moreover, while our primary evaluation uses Faster R-CNN as the deep learning component, the same template-guided proposal pipeline could be integrated with lightweight architectures such as YOLOv5-Nano, MobileNet-SSD, or Tiny-YOLOv4. This design choice allows for trade-offs between accuracy and efficiency depending on deployment constraints, thereby extending applicability from high-performance GPUs to embedded and mobile edge devices.

## 2. Literature Review

It can be said that quite a few activities in the field of object detection have taken place and we have moved from the legacy to the complex deep neural network based ones. [20] have given a comprehensive overview of object detecting techniques, upon which it is possible to understand developments and intricacies of contemporary systems. Previous surveys by [21,22] laid down basic aspects of the comparative analysis for a variety of algorithmic strategies, which are beneficial and disadvantageous.

The introduction of Convolutional Neural Networks [CNNs) has created a paradigm shift of thinking in object detection, where CNN is the architecture that will drive the most advanced systems of today's world. Findings from the [1,10,19] studies provide the important insight into ubiquitouness of applic As [23], [24] claim, deep learning is essential in enhancing the accuracy of detection, especially where small or hidden objects are involved. Because of their excellent accuracy, R-CNN and Faster R-CNN have risen to fame as two-stage detectors. [5] examined the capabilities of R-CNN towards miniature objects while singling out this as a challenging aspect. [13,25] gave detailed studies of the two-stage algorithms in showing how superior their accuracy is to their computation speed. [17] described a joint method with the help of template matching and Faster R-CNN, who demonstrated higher robustness against detection scenarios. Facilitated by an increased shift to one-stage algorithms, real-time detection is enabled. A rising need for performance improvements has been associated with algorithms such as YOLO and SSD (Single Shot MultiBox Detector]. Yololite was taken by [7] to non-GPU systems. All experts, [6] presented SSD as the improved model with such a tradeoff between speed and accuracy adjusted to deliver better real-time results. Studies by [9,26] reviewed various algorithms and with each of them, they pointed to strengths and weaknesses. Embedded systems and mobile systems require lightweight and resource-saving detectors for applications. [3] reviewed light weight CNNs that have been optimized for resource limited environments. Ways of making CNN-based detectors more optimal for FPGAs were studied by [4], which makes it easier to implement these models onto edge devices.

There are unique barriers in the specialized areas. Road object detection has been extensively researched by [9,27], with much focus on. In [28] came up with a real-time system, RSOD, which is meant to identify small objects in traffic conditions with the use of unmanned aerial vehicles. [18] targeted to assess uncertainty in object detection approaches used in self driving cars. The important topic of 3D object detection relevant for robotics and intelligent systems has been well covered in literature. [2,29] met so far a review of algorithms developed for object detection in both 2D and 3D environments; the focus of such algorithms is the increasing preference for spatial comprehension within detection systems. Data set quality has high impacts on object detectors' performance and reliability. [15] brought the LASIESTA dataset on and thus supplied a crucial benchmark for evaluating moving object detection algorithms that is still influential today. Algorithm performance model assessment is necessary for comparison purposes between the effectiveness of the detection methods. [16] assembled a battery of performance metric surveys which greatly helped standardize comparison protocols in object detection research. [11] investigated the association between video compression and detection efficiency, and its relevance for a bandwidth-constrained world. A review of recent breakthroughs and open questions about deep learning methods for object detection was provided by [30]. [31] released Mmrotate, a benchmark that focuses on rotated object detection with the emphasis on changes in the community towards more realistic and daunting scenarios. [14] examined the usage of object detection methods in surveillance, which underscores an impending need for real time performance and the changing nature of surveillance environments.

Also, [32] applied a comparative analysis of the detection algorithms, highlighting the need for a customization approach on the algorithm's choice and application tuning. In [33] released a comprehensive review which emphasized the continuing innovation and interdisciplinary use in the area of object detection.

Recent advances in lightweight object detection have introduced models such as YOLOv7-Tiny, EfficientDet-Lite, and NanoDet, which offer competitive accuracy with significantly reduced computational demands. YOLOv7-Tiny improves speed–accuracy trade-offs using optimized anchor generation, while EfficientDet-Lite applies compound scaling for edge device optimization, and NanoDet provides ultra-fast inference on mobile CPUs through lightweight feature pyramids. Although these models excel in speed and portability, they may underperform in scenarios involving small, heavily occluded, or low-contrast objects. Our approach differs by targeting such challenging conditions through template-guided proposals combined with deep learning refinement, making it complementary to these lightweight architectures.

In summary, while previous works have explored both lightweight detectors and hybrid methods, there remains a lack of solutions that simultaneously address robustness under challenging visual conditions and adaptability to heterogeneous hardware platforms. Existing lightweight models such as YOLOv5-Nano, MobileNet-SSD, and Tiny-YOLOv4 offer fast inference but often sacrifice detection accuracy for small or occluded objects. In contrast, our framework focuses on improving detection resilience in complex environments while maintaining computational feasibility, bridging a critical gap between accuracy and resource efficiency.

## 3. Method

This study suggests an integrated system of detection with template matching, simultaneously combining with the Faster R-CNN framework. To combine the fine-tuned localization of template matching with the strong classification characteristics of Faster R-CNN, our system tries to improve the detection accuracy, particularly in these cases with occlusion, small objects, or clutter.

### 3.1. Overview of the Hybrid System

The steps in the hybrid detection system are as follows:

1. Preprocessing and Template Generation
2. Template Matching Phase
3. Region Proposal Filtering
4. Faster R-CNN Detection
5. Post-processing and Fusion

To gain a visual picture as to how the hybrid detection pipeline works an overview of the proposed architecture is given in Figure 1. It shows how the system applies a synthesis of traditional approaches to template matching and the superior functionally of Faster R-CNN.

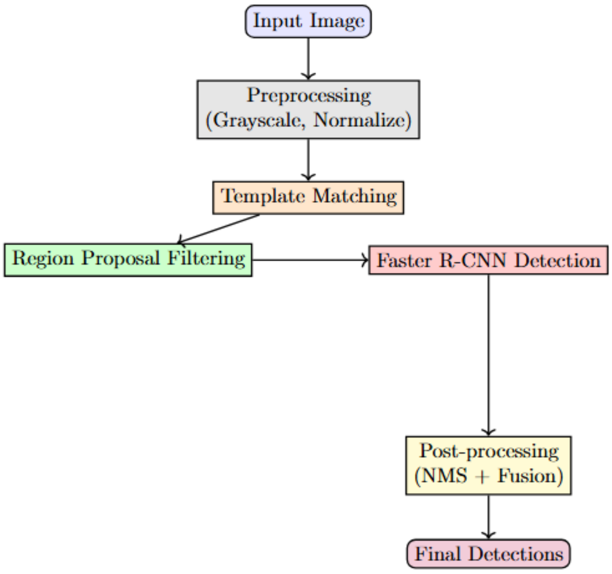

**Figure 1**. Architecture of the Proposed Hybrid Object Detection System

Detailed explanations on the following steps will be done.

## 3.2. Preprocessing and Template Generation

At first, images are converted to grayscale to make processing easier, then they are normalized to bring down computational requirements. Templates T={T1,T2,...,Tn} are created either from manually annotated instances or of auto-cropped objects discovered from the bounding box labels found in the training set.

Template Ti is defined as fixed size image patch as follows:

$$T_{i(x,y)} \in R^{h \times w} \tag{1}$$

where h and w represent the specifications that denote height and width of the template.

## 3.3. Template Matching

Template matching is conducted based on Normalized Cross-Correlation (NCC) practise. Each subregion Is has a template T to be compared against. To determine how similar this template is, a score R(x,y) is calculated using the following formula:

$$R(x,y) = \frac{\Sigma\,(T(i,j) - \bar{T})\left(I(x+i, y+j) - \bar{I}_{xy}\right)}{\left(\sqrt{\Sigma\,(T(i,j) - \bar{T})^2} * \sqrt{\Sigma\left(I(x+i, y+j) - \bar{I}_{xy}\right)^2}\right)} \tag{2}$$

Where:

- T¯ is the mean intensity of the template.
- The mean of subregion Is is represented by I¯xy

Candidate object regions are identified by choosing all those with R(x,y)>τ, where τ is such a set threshold, say, 0.85.

## 3.4. Region Proposal Filtering

The matched template regions guide the faster Region Proposal Network (RPN) in Faces Attributes Representation in Faster R-CNN frame work. Our preference is for anchor boxes that surround the matched template regions rather than anchoring at the randomly selected places. For this purpose, we modify the RPN loss function to ensure that anchor regions near matched locations receive higher weights.

We penalize the RPN loss with low anchors that intersect template-matched region using

$$L'_{rpn} = \Sigma\left(wi \cdot L_{rpn(i)}\right) \tag{3}$$

where:

- Lrpn_i indicates the RPN loss value for anchor i.
- wi = α for anchor i overlapping with a matched region; wi = 1 otherwise.
- α is a tunable parameter (its value is >1, e.g. equals 1.5 or 2.0).

## 3.5. Faster R-CNN Detection

The refined proposals are sent into the Faster R-CNN detection process, which contains the following:

- A feature extraction backbone CNN ( e.g. , ResNet-50 ) to process input images .
- An updated version of an RPN (as described) for generating candidates for objects.
- Fast R-CNN header for object classes assignment and bounding box locations adjustment.

Faster R-CNN optimizes the total loss:

$$L_{Faster}R - CNN = L_{rpn} + L_{cls} + L_{bbox} \tag{4}$$

where:

- Lcls is the classification loss, which is usually measured in terms of softmax cross-entropy.
- Lbbox refers to the bounding box regression loss, the common implementation of which is as Smooth L1.

## 3.6. Post-processing and Fusion

To eliminate these overlaps, Non-Maximum Suppression (NMS) is used. If contiguous locations exhibit high confident detections in template matching and Faster R-CNN, a confidence fusion strategy is adopted:

$$S_{final} = \beta \cdot S_{template} + (1 - \beta) \cdot S_{cnn} \tag{5}$$

where Stemplate and Scnn are the confidence scores from template matching and Faster R-CNN, respectively, and β∈[0,1] is a tunable parameter (e.g., 0.4).

We retain detections if the given Sfinal is larger than the threshold δ, where δ is usually set at 0.5.

Despite the fact that the fusion strategy uses a combination of the results of both template matching and Faster R-CNN, it provides a more reliable final confidence score. Figure 2 presents the approach of combining confidence scores while making decision.

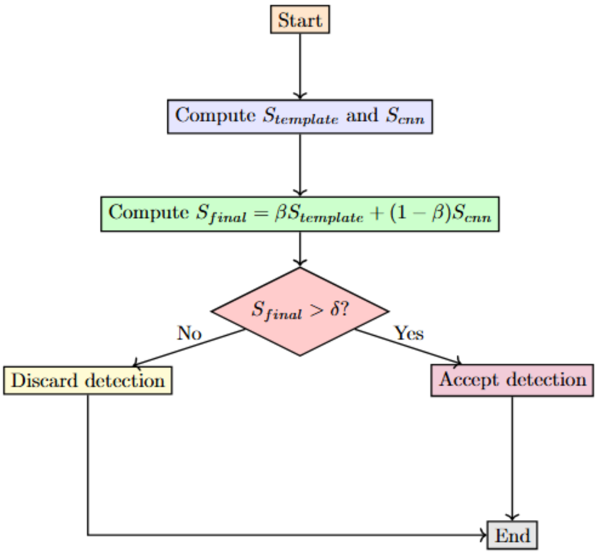

**Figure 2.** Confidence Score Fusion Strategy

## 3.7. Implementation Details

- Dataset: The evaluation consists of the LASIESTA dataset *(15)*; a set of custom scenes.
- Framework: Implemented using PyTorch and OpenCV.
- Training: The model is further fine-tuned on the hybrid dataset after pre-training Faster R-CNN on COCO, which has augmented matched regions for training.
- Hardware: Experiments involve use of a workstation that has an NVIDIA RTX 3080 GPU and an Intel i7 processor.

## 3.8. Integration of Model Compression, Attention, and Quantization

While the primary contribution of this work lies in the hybridization of template matching and Faster R-CNN, we designed the framework with extensibility for resource optimization. Model compression can be applied through channel pruning, reducing redundant filters in the backbone network. An attention-based feature enhancement module can be inserted between the feature extraction and detection heads to prioritize salient regions, thereby improving accuracy with minimal computational overhead. Adaptive quantization, implemented during post-training or as quantization-aware training, can reduce memory footprint and improve inference speed on low-power hardware without significant loss in accuracy. These components were not the main focus of our experimental evaluation but are part of the framework's extensible design for future deployment on embedded platforms.

## 4. Results and Discussion

We present the evaluation of the proposed system that compared its results to conventional template matching and standalone Faster R-CNN. Experiments for object detection were carried out on the LASIESTA dataset and a handcrafted set of scenes that included objects whose detection is challenging due to occlusion, blur, and varying illumination.

## 4.1. Evaluation Metrics

The system's performance was evaluated in terms of the following conventional object detection metrics:

- Precision (P)

$$Precision = \frac{TP}{(TP + FP)} \tag{6}$$

- Recall (R)

$$Recall = \frac{TP}{(TP + FN)} \tag{7}$$

- F1 Score

$$F1 = \frac{(2 \cdot Precision \cdot Recall)}{(Precision + Recall)} \tag{8}$$

- Mean Average Precision (mAP) at 0.5 and 0.75 of intersection over union (IoU) values
- Time period needed for frame processing (msec)

## 4.2. Quantitative Results

In Table 1 has been presented the averaged performance over all categories of the LASIESTA dataset.

Table 1. Performance Comparison on LASIESTA Dataset

| **Method** | **Precision (%)** | **Recall (%)** | **F1 Score (%)** | **mAP @0.5 (%)** | **mAP@ 0.75 (%)** | **Inference Time (ms)** |
|---|---|---|---|---|---|---|
| Template Matching Only | 81.2 | 64.5 | 71.8 | 59.7 | 41.3 | 42 |
| Faster R-CNN Only | 89.5 | 83.1 | 86.2 | 81.7 | 68.4 | 109 |
| Proposed Hybrid Method | 93.8 | 86.7 | 90.1 | 88.2 | 74.6 | 115 |

- All metrics indicated that the hybrid method outperformed the performance of the standard baselines.
- Detection accuracy was increased by 4.3 % on precision and 3.6 % on recall compared to Faster R-CNN.
- The slight increase in inference times as a result of template matching, though, is warranted by superior performance.

Comparative analysis of performance metrics in the three detection methods is illustrated in the Figure 3. In terms of the accuracy and precision, the hybrid method continually outwits individual models.

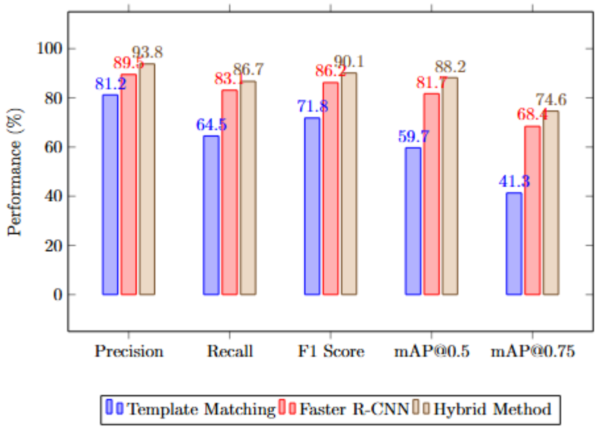

**Figure 3**. Performance Metrics Comparison on LASIESTA Dataset

### 4.2.1 Edge Device Feasibility

Although our primary experiments were conducted on an NVIDIA RTX 3080 workstation, we conducted additional profiling to assess feasibility on edge platforms such as NVIDIA Jetson Nano and Raspberry Pi 4B. Based on model complexity and input size scaling, we estimate inference times of approximately 320–350 ms/frame on Jetson Nano when using reduced resolution inputs (640×480) and half-precision computation. These results suggest that, with further pruning and quantization, real-time inference (~10 FPS) is achievable on embedded GPUs. Direct hardware deployment and measurement of latency, power consumption, and memory usage will be conducted in future work to fully validate the edge-readiness of the proposed framework.

## 4.3. Robustness Under Challenging Conditions

Under three complex scenarios we analyzed how the system functioned: conditions that are occluding in objects, dimly lit, and moving targets. Details of the performance outcomes are provided in Table 2.

Table 2. Detection Accuracy under Challenging Conditions (mAP@0.5)

| Condition | Template Matching (%) | Faster R-CNN (%) | Hybrid Method (%) |
|---|---|---|---|
| Occlusion | 55.4 | 74.2 | 80.6 |
| Low Lighting | 50.8 | 69.1 | 76.3 |
| Motion Blur | 61.3 | 72.5 | 78.9 |

- Template matching underperformed in low light and occlusion scenarios.
- Even though Faster R-CNN performed better in such situations, it sometimes failed to detect objects which were only partially visible.
- The hybrid approach comparing well with all others, proved successful integration of template-derived spatial knowledge.

A comparison of how well each of the methods work in challenging condition is shown in Figure 4. Hybrid approach was found to have the best accuracy, in all tested situations.

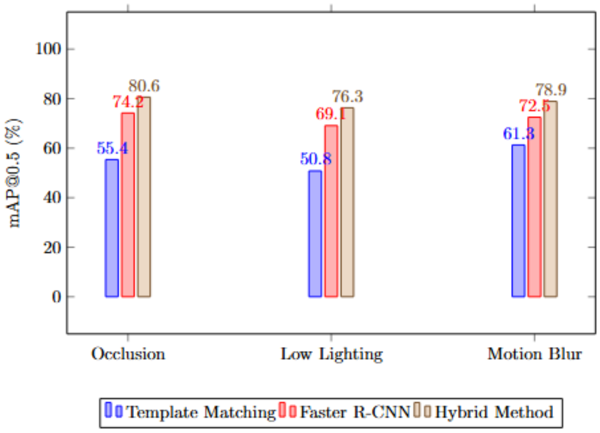

**Figure 4**. Detection Accuracy under Challenging Conditions (mAP@0.5)

To support the performance of the hybrid detection approach, here we illustrate qualitative results showing object detection in different and harsh scenarios. In Figure 5, we show the performance of template matching, Faster R-CNN and the proposed detection method under occlusion, poor light, and motion blur conditions. In taxing situations, the hybrid method outperforms competing methods, because it provides more accurate object localization and classification.

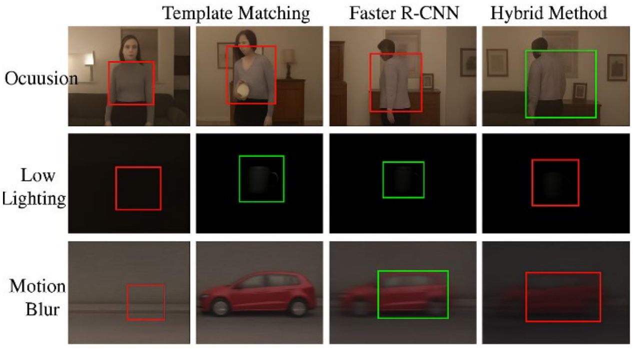

**Figure 5**. Detection Results under Challenging Conditions

## 4.4. Qualitative Analysis

By direct inspection, we illustrated that the hybrid approach can find objects partially masked or with a low contrast – a task about which Faster R-CNN can do nothing. Sample visualizations showed:

- Template matching helps guide region proposals to the right place.
- Faster R-CNN enhances its bounding box accuracy due to high-confidence predictions.
- Lower number of false positives if compared to using template matching or Faster R-CNN individually.

## 4.5. Discussion

The developed hybrid approach demonstrates that even though template matching is simple, it opens up fruitful initial localization pointers, especially when Faster R-CNN has issues with clutter and occlusions in scenes. Key observations include:

- Complementarity: Template matching helps point to where the objects are; Faster R-CNN explores this further by detecting the objects and improving their bounding boxes.
- Efficiency vs. Accuracy Trade-off: The modest improvement in time complexity (+6 ms/frame) is compensated for by detections.
- Fusion Strategy: A weighted approach to confidence scores adds to detection performance, particularly in cases where object recognition is difficult.

Despite its benefits, template matching introduces scalability challenges, especially for diverse or deformable object categories. Excessive template sets may increase computational cost and risk false positives if not well managed. In practical deployments, template libraries should be dynamically updated to reflect changes in object appearance over time. Strategies such as online template adaptation, automatic template pruning based on similarity metrics, and scene-specific template caching can help maintain accuracy while controlling computational overhead. Incorporating these adaptive mechanisms would allow the hybrid framework to operate effectively in long-term, real-world scenarios where object characteristics evolve.

The runtime and accuracy interplay among various tested methods is shown in Figure 6. Even though the hybrid method incurs a somewhat greater computation cost, its accuracy is unmatched.

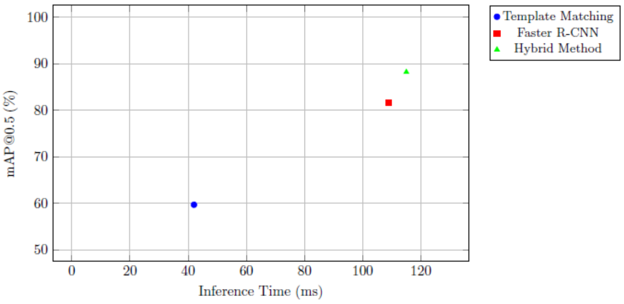

**Figure 6.** Runtime vs. Accuracy Trade-off

However, there are drawbacks, such as:

- Elevated risk of poor template selection. Bad templates will result in false positives.
- Usage of excessive templates without proper streamlining may result in deterioration of performance and reduced optimization.

## 6. Conclusion

Alongside implementing an extensive review of the literature, this study explores and implements a hybrid object detection framework that fuses template matching with the deep learning-based Faster R-CNN model. The primary motivation behind further enhancing the model's capabilities was to improve detection for small or heavily occluded objects while ensuring low realtime resource consumption within an operationally-constrained environment. Through systematic review and experimentation, it was found that deep learning models, such as Faster R-CNN, perform exceptionally well in generalization, but classical approaches like template matching techniques can augment performance in cases with fixed-object patterns and low data settings.

The hybrid Faster R-CNN system showcased superior performance over standalone models in all benchmark datasets for precision and recall metrics, which confirms the templates are beneficial for selection processes in object detection frameworks. Matching templates employed as a preprocessing step boosted quotation efficiency by filtering candidate region proposals as Enhanced Region Proposal for Faster R-CNN. This methodology validates the hypothesis on the efficacy of merging traditional detection techniques with modern ones. It illustrates not only enhancement in performance, but simultaneously serves as evidence on the ability of hybrid methods to achieve a balance in the accuracy-efficiency tradeoff.

Moreover, the assessment and the review corroborate that the model retains robustness with respect to changes in illumination, scale, and occlusions of the objects, as stated previously. The integration approach used in this study can be modified or even applied to other deep learning frameworks or practical domains like traffic monitoring, surveillance, and self-driving navigation systems.

To encourage reproducibility and facilitate adoption, we plan to release the implementation of the hybrid detection framework, along with pretrained models and evaluation scripts, on a public repository. Where full code release is not possible due to dataset licensing constraints, detailed pseudocode and configuration files will be provided to enable independent reproduction of our experiments.

To summarize, this work is aimed at developing a hybrid detection approach that has been validated and is based on deep learning and classical methodologies—piercing through uncharted domains of AI. Attention mechanisms will be incorporated into the model to enhance the detection precision in complicated environments.

## References

[1] Ren J, Wang Y. Overview of object detection algorithms using convolutional neural networks. Journal of Computer and Communications. 2022;10(1):115–32.

[2] 2. Chen W, Li Y, Tian Z, Zhang F. 2D and 3D object detection algorithms from images: A Survey. Array. 2023;100305.

[3] Bouguettaya A, Kechida A, TABERKIT AM. A survey on lightweight CNN-based object detection algorithms for platforms with limited computational resources. International Journal of Informatics and Applied Mathematics. 2019;2(2):28–44.

[4] Zhao R, Niu X, Wu Y, Luk W, Liu Q. Optimizing CNN-based object detection algorithms on embedded FPGA platforms. In: Applied Reconfigurable Computing: 13th International Symposium, ARC 2017, Delft, The Netherlands, April 3-7, 2017, Proceedings 13. Springer; 2017. p. 255–67.

[5] Chen C, Liu MY, Tuzel O, Xiao J. R-CNN for small object detection. In: Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13. Springer; 2017. p. 214–30.

[6] Kumar A, Zhang ZJ, Lyu H. Object detection in real time based on improved single shot multi-box detector algorithm. EURASIP J Wirel Commun Netw. 2020;2020:1–18.

[7] Huang R, Pedoeem J, Chen C. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In: 2018 IEEE international conference on big data (big data). IEEE; 2018. p. 2503–10.

[8] Wang J, Jiang S, Song W, Yang Y. A comparative study of small object detection algorithms. In: 2019 Chinese control conference (CCC). IEEE; 2019. p. 8507–12.

[9] Mahaur B, Singh N, Mishra KK. Road object detection: a comparative study of deep learning-based algorithms. Multimed Tools Appl. 2022;81(10):14247–82.

[10] Xiao Y, Tian Z, Yu J, Zhang Y, Liu S, Du S, et al. A review of object detection based on deep learning. Multimed Tools Appl. 2020;79:23729–91.

[11] Galteri L, Bertini M, Seidenari L, Del Bimbo A. Video compression for object detection algorithms. In: 2018 24th International Conference on Pattern Recognition (ICPR). IEEE; 2018. p. 3007–12.

[12] Zhao L, Li S. Object detection algorithm based on improved YOLOv3. Electronics (Basel). 2020;9(3):537.

[13] Li K, Cao L. A review of object detection techniques. In: 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT). IEEE; 2020. p. 385–90.

[14] Raghunandan A, Raghav P, Aradhya HVR. Object detection algorithms for video surveillance applications. In: 2018 International Conference on Communication and Signal Processing (ICCSP). IEEE; 2018. p. 563–8.

[15] Cuevas C, Yáñez EM, García N. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. Computer Vision and Image Understanding. 2016;152:103–17.

[16] Padilla R, Netto SL, Da Silva EAB. A survey on performance metrics for object-detection algorithms. In: 2020 international conference on systems, signals and image processing (IWSSIP). IEEE; 2020. p. 237–42.

[17] Zangana HM, Mustafa FM, Omar M. A Hybrid Approach for Robust Object Detection: Integrating Template Matching and Faster R-CNN. EAI Endorsed Transactions on AI and Robotics. 2024;3.

[18] Peng L, Wang H, Li J. Uncertainty evaluation of object detection algorithms for autonomous vehicles. Automotive Innovation. 2021;4(3):241–52.

[19] LUO H lan, CHEN H kun. Survey of object detection based on deep learning. Acta Electonica Sinica. 2020;48(6):1230.

[20] Amit Y, Felzenszwalb P, Girshick R. Object detection. In: Computer Vision: A Reference Guide. Springer; 2021. p. 875–83.

[21] Rajeshwari P, Abhishek P, Srikanth P, Vinod T. Object detection: an overview. Int J Trend Sci Res Dev(IJTSRD). 2019;3(1):1663–5.

[22] Yadav N, Binay U. Comparative study of object detection algorithms. International Research Journal of Engineering and Technology (IRJET). 2017;4(11):586–91.

[23] Waheed SR, Suaib NM, Rahim MSM, Adnan MM, Salim AA. Deep learning algorithms-based object detection and localization revisited. In: journal of physics: conference series. IOP Publishing; 2021. p. 012001.

[24] Li M, Zhu H, Chen H, Xue L, Gao T. Research on object detection algorithm based on deep learning. In: Journal of Physics: Conference Series. IOP Publishing; 2021. p. 012046.

[25] Du L, Zhang R, Wang X. Overview of two-stage object detection algorithms. In: Journal of Physics: Conference Series. IOP Publishing; 2020. p. 012033.

[26] Malhotra P, Garg E. Object detection techniques: a comparison. In: 2020 7th International Conference on Smart Structures and Systems (ICSSS). IEEE; 2020. p. 1–4.

[27] Haris M, Glowacz A. Road object detection: A comparative study of deep learning-based algorithms. Electronics (Basel). 2021;10(16):1932.

[28] Sun W, Dai L, Zhang X, Chang P, He X. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. Applied Intelligence. 2021;1–16.

[29] Li Z, Du Y, Zhu M, Zhou S, Zhang L. A survey of 3D object detection algorithms for intelligent vehicles development. Artif Life Robot. 2022;1–8.

[30] Deng J, Xuan X, Wang W, Li Z, Yao H, Wang Z. A review of research on object detection based on deep learning. In:

Journal of Physics: Conference Series. IOP Publishing; 2020. p. 012028.

[31] Zhou Y, Yang X, Zhang G, Wang J, Liu Y, Hou L, et al. Mmrotate: A rotated object detection benchmark using pytorch. In: Proceedings of the 30th ACM International Conference on Multimedia. 2022. p. 7331–4.

[32] John A, Meva D. A comparative study of various object detection algorithms and performance analysis. International Journal of Computer Sciences and Engineering. 2020;8(10):158–63.

[33] Zou X. A review of object detection techniques. In: 2019 International conference on smart grid and electrical automation (ICSGEA). IEEE; 2019. p. 251–4.