International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Faster RCNN-Integrated Deep Reinforcement Learning for Efficient Object Detection: Optimizing Accuracy, IOU, and Image Evaluation Time in Resource-Constrained Environments

Satyam Sankesa[1], Dr. Manish Saraf[2]

[1]PhD Research Scholar, [2]PhD Research Supervisor

[1,2]Department of Computer Science & Engineering, Eklavya Vishwavidyalaya, Damoh, M.P., India

Abstract:

Object detection remains one of the most challenging tasks in computer vision, particularly when deployed on resource-constrained devices that must balance computational efficiency with detection accuracy. The current paper suggests a new hybrid architecture that combines Faster Region-based Convolutional Neural Network (Faster RCNN) with Deep Reinforcement Learning (DRL) to obtain efficient and accurate object detection. The suggested model uses Faster RCNN as a feature backbone to produce high-quality regional proposals, and a DRL agent trains to first refine the model by further bounding box localization using an optimized reward that depends on Intersection over Union (IOU). Image Evaluation Time (IET) is another important performance metric that is also presented by the framework to determine the feasibility of real-time deployment. Experiments with PASCAL VOC 2012 and ImageNet indicate the proposed model has a mean Average Precision (mAP) of 84.7, an IOU score of 87.3, and an IET of 38 ms per image, which is better than a range of other modern baseline approaches. Moreover, the model has a much lower storage footprint (112 MB) and reduced computational overhead, making it appropriate for embedded and edge computing platforms. The findings confirm that Faster RCNN and DRL synergy provide a solid and scalable tool in real-world applications of object identification.

Keywords: Deep Reinforcement Learning, Faster RCNN, Object Detection, Bounding Box Localization, Image Evaluation Time, IOU, Resource-Constrained Devices, Feature Extraction, Computer Vision.

1. INTRODUCTION

Computer vision is one of the pillars of artificial intelligence, which allows a machine to interpret, analyze, and comprehend visual data of the physical world. Human beings have the mental ability to see, identify, and classify different objects and this enables them to make quick decisions. Object detection, which can be defined as the process of recognizing and locating multiple objects in an image, is just one of the numerous sub-problems of computer vision that has attracted a lot of research attention because of its wide applicability in surveillance systems, autonomous vehicles, robotics, healthcare, and smart city infrastructure (Al Duhayyim et al., 2022). It is a complex task by nature since it must solve two intertwined problems at the same time, namely, what an object is (recognition) and where it is found (localization).

Early object detection methodologies relied heavily on hand-crafted feature extraction using techniques such as Histograms of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Support Vector Machine (SVM) classifiers (Naman & Ameen, 2022). During the early 2000s, the limited quantity of images available for training Neural Networks resulted in inefficient task handling by GPU graphics

processing units and multicore processing systems. Since 2010, significant advancements in machine learning, specifically with the introduction of multicore and GPU technologies, have made it easier to handle large training datasets such as ImageNet using neural networks. Two types of feature extraction — low-level and high-level — can be employed to effectively identify objects, where colour, shape, and texture features are necessary for successful object detection (Rani et al., 2022).

The emergence of deep neural networks and large-scale datasets such as ImageNet catalyzed a fundamental shift in the object detection paradigm, leading to the development of architectures like Region-based Convolutional Neural Networks (RCNN), Fast RCNN, and Faster RCNN. Faster RCNN, with its integrated Region Proposal Network (RPN), introduced a shared convolutional architecture that generates object candidate regions directly from image feature maps, substantially reducing inference time compared to its predecessors (Awaisi et al., 2023). Regardless of these improvements, there are still issues in determining the most accurate detection in uncontrolled scenarios such as placing objects in arbitrary positions in a cluttered and occluded environment especially on the less powerful computing platforms like embedded systems and internet of things edge devices.

Simultaneously, Deep Reinforcement Learning (DRL) has become an effective paradigm of sequential decision-making problems. Training an agent to act in an environment and maximize a cumulative reward over a long period of time, DRL algorithms have proven to be exceptionally effective in game-playing, robotics, and autonomous control (Chen et al., 2023). One of the promising directions of DRL application is object detection: instead of making one pass through and classify data, a DRL-based detector may refine its search window through multiple steps, relying on feedback information, which is a close simulation of the human attention system (Tan and Karaköse, 2022).

Nonetheless, the current literature shows that there is a crucial gap in the research given that limited studies have examined combining the Faster RCNN feature extraction functionality with the capability to refine localization using DRL, particularly taking into consideration Image Evaluation Time (IET) and model storage efficiency to support resource-constrained deployments. This gap is filled through this paper by proposing a hybrid Faster RCNN-DRL framework to jointly optimize detection accuracy (measured by precision, recall, and mAP), localization quality (IOU), inference speed (IET) and model compactness (storage size in MB).

The primary contributions of this work are as follows:

1. A novel hybrid architecture combining Faster RCNN-based feature extraction with a DRL agent for iterative bounding box refinement.
2. An optimized reward function that maximizes IOU between predicted and ground-truth bounding boxes.
3. Introduction and benchmarking of Image Evaluation Time (IET) as a performance criterion for resource-constrained object detection.
4. Comprehensive evaluation on PASCAL VOC 2012 and ImageNet datasets with comparative analysis against state-of-the-art methods.
5. Demonstration of model viability for deployment on embedded and edge computing platforms.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed methodology. Section 4 presents experimental results and comparative analysis. Section 5 discusses findings and limitations. Section 6 concludes the paper with future research directions.

2. REVIEW OF LITERATURE

The theoretical framework of modern object detection research relies heavily on insights from prior literature. The review has encouraged careful thought about the many challenges in this domain, leading to the formulation of study goals and the identification of research gaps. This section reviews key works across three thematic areas: deep learning-based object detection, reinforcement learning for localization, and efficient deployment on constrained hardware.

2.1 Deep Learning-Based Object Detection

Rani et al. (2022) examined innovative methods in machine-based analysis of images in medical oncology and proved that CNN architectures are trained using big datasets and yield more discriminative features as compared to any hand-crafted alternatives. Their publication emphasized that the ML techniques and methods can analyze large amounts of data in a rational manner, which served as a fundamental basis for the feature extraction methods used in the given work.

The VULMA project was proposed by Cardellicchio et al. (2023). This is a machine learning framework that includes four modules of a processing pipeline to estimate the seismic vulnerability indices using photographs. The framework (domain-specific) was shown to be practically viable, but it did not disprove the relevance of efficient image evaluation pipelines to real-world image-based assessment tasks.

The RCNN family of detectors - RCNN, Fast RCNN, and Faster RCNN - marks an improvement on the speed and accuracy of region-based detection. YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) are one-stage detection, which sacrifice inference speed in favor of accuracy. Although such architectures are better than two-stage detectors, such as Faster RCNN, in real-time, for small or occluding objects, their performance is relatively low (Awaisi et al., 2023).

2.2 Object Localization and Detection Reinforcement Learning

Tan and Karaköse (2022) have shown a deep reinforcement learning framework to identify objects by the use of the PASCAL VOC 2012 dataset, a five-layered neural network structure. They used the method of incrementally moving towards the target using a bounding box that would ultimately cover the item of interest and the reward algorithm was made such that it maximised the mAP value. It was because the frequency of the ground truth and the bounding box overlap was optimised to provide such a favourable outcome, which formed the justification behind the optimized reward function that the proposed work employs to obtain a favourable outcome.

Chen et al. (2023) also investigated vision-based robotic grasping of objects with a combined YOLO detection with Soft Actor-Critic (SAC) Deep Learning algorithm. The study showed that a hybrid computer vision-based object identification system and a self-learning deep reinforcement learning algorithm might be used to successfully train a robotic manipulator, confirming the usefulness of hybrid vision-DRL pipelines in practical use.

To the issue of malicious PDF detectors that are systematized based on their structural characteristics, Jiang et al. (2023) employed the DRL to the attack and defense problem. Despite its domain-specificity, the work showed the generality of DRL in sequential feature-based decision-making, in which an RL agent varies a set of features over a series of actions attaining a desired goal, which is literally equivalent to bounding box refinement.

The proposed method of adaptive formation motion planning and control of autonomous underwater vehicles (AUVs) (Hadi et al., 2023) makes use of an actor-critic framework and is based on DRL. The system was shown to be robust in realistically disturbed problems such as ocean currents, communication delays, and sensor errors, which highlights the reliability of actor-critic DRL architectures to solve problems in complex, semi-observable environments, which applies to cluttered object detection problems.

2.3 Resource-Constrained Environment Efficient Detection

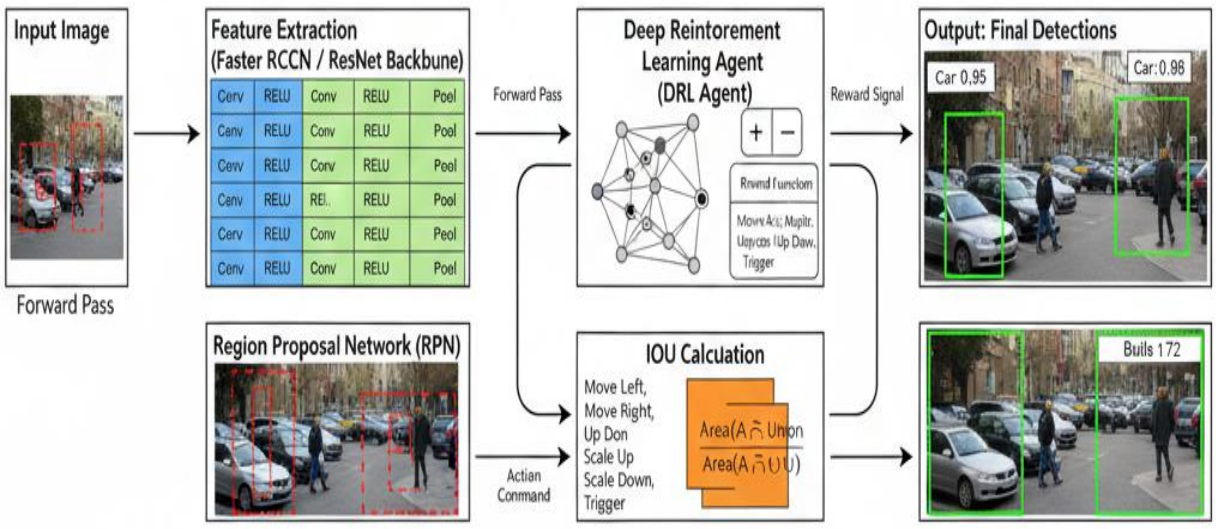
Awaisi et al. (2023) introduced a smart parking system based on IIoT and Deep Reinforcement Learning, where the use of fog devices to classify cars and assign parking spots automatically is possible with the help of Deep Reinforcement Learning. The framework demonstrated competitive performance in both processing rate and detection rate, which proves that DRL-based systems are possible in resource-constrained edge computing systems.

As a direct response to the requirement of computationally efficient detection systems working within limited computing and networking capabilities, Alhamed et al. (2022) have created a DRL-based smart building video surveillance system with a total surveillance performance of 98.6% and low latency.
Al Duhayyim et al. (2022) used Deep Reinforcement Learning and the Mask RCNN to classify waste around the city, and they used DenseNet as a model that was then improved with a hyperparameter optimizer based on the Dragonfly Algorithm. As shown by the IDRL-RWODC approach, DRL combined with region-based CNNs can be used to provide competitive classification results with less memory overhead - a direct ancestor to the architecture used in the present work.
The article by Naman and Ameen (2022) suggests a new feature selection approach with deep reinforcement learning and proves that reinforcement-based feature pruning is an effective feature selection technique. The proposed method used fundamental DRL principles to identify features with the aim of training RF, KNN, and SVM classifiers, and the classification accuracy gains were seen on the VisDA-2018 and Syn2Real databases - confirming the ability of DRL to enhance the performance of features.
Even with these developments, a single open framework integrating explicitly Faster RCNN feature extraction with the DRL-based localization refinement, and at the same time optimizing IET and storage footprint to work with resource-constrained settings, is not well explored yet. The given work bridges this gap.

3. PROPOSED METHODOLOGY

The proposed architecture of the system has been described in detail below: 3.1 System Architecture Overview. The proposed architecture of the system has been outlined below:
The proposed Faster RCNNIntegrated Deep Reinforcement Learning (FRCNN-DRL) pipeline is a dual-stage hybrid pipeline. Stage one is the Faster RCNN, which is a feature extractor and a region proposal generator. The second stage takes the suggestions of the first stage and focuses an agent of DRL on the suggested regions by means of the iterative action-driven adjustment of the bounding boxes, under an IOU-maximizing rewarding method.

Figure 1: FRCNN-DRL System Architecture



The architecture illustrates the end-to-end pipeline: an input street scene image undergoes feature extraction via the Faster RCNN ResNet backbone, generates region proposals through the RPN, and feeds into the DRL Agent. The agent iteratively applies move/scale actions guided by IOU-based reward signals, ultimately producing final bounding box detections with confidence scores.

The architecture in Figure 1 shows that the input image passes through the Faster RCNN backbone (ResNet-50) for feature extraction, the RPN generates candidate proposals, and the DRL agent then receives the feature state and iteratively refines the bounding box position through a defined action set. The reward function guides convergence toward maximally overlapping detection boxes.

3.2 Stage 1 — Feature Extraction with Faster RCNN

The input image $I \in R^{H\times W\times 3}$ is processed by a ResNet-50 backbone to produce a shared convolutional feature map $F \in R^{H'\times W'\times 3}$ The Region Proposal Network (RPN) slides a small network over F and outputs a set of rectangular objectness proposals $\{r_1, r_2, \dots, r_K\}$, each associated with an objectness score.

The RPN joint loss is defined as:

$$LRPN = Lcls(p_i, p_i^*) + \lambda \cdot p_i^* \cdot Lreg(t_i, t_i^*)$$

where p_i^* is the predicted objectness score, p_i *is the ground-truth label,* t_i *is the predicted bounding box offset,* t_i^* is the ground-truth offset, and λ is a balancing weight. ROI Pooling is then applied to extract fixed-size feature vectors from each proposal, which are passed to fully connected layers for initial classification and bounding box regression.

3.3 Stage 2 — DRL-Based Bounding Box Refinement

The DRL agent operates on the candidate regions generated by Faster RCNN. The agent is modeled as a Markov Decision Process (MDP) defined by the tuple ⟨S,A,R,T⟩ :

- **State Space S}**: A state s_t is a concatenation of (a) the ROI-pooled feature vector from Faster RCNN, (b) the current bounding box coordinates normalized to image dimensions, and (c) a binary history vector of the last 10 actions taken.
- **Action Space A}**: The agent selects from 9 discrete actions: *move left, move right, move up, move down, scale wider, scale narrower, scale taller, scale shorter,* and *trigger* (finalize detection).
- **Reward Function R}**: The reward is defined based on the change in IOU between consecutive steps:

$$r(s_t, a_t) = sign(IOU(b_{t+1}, b^*) - IOU(b_t, b^*))$$

where b_t is the current bounding box, b_{t+1} is the box after action a_t, and b^* is the ground-truth bounding box. A terminal reward of +3 is given when $IOU \geq 0.5$ upon triggering detection.

- **Transition T}**: The environment transitions deterministically based on the geometric transformation defined by each action.

Figure 2: Iterative DRL Bounding Box Refinement Process

Figure 6 illustrates the step-by-step process: starting from an initial coarse proposal (IOU = 0.31, shown in red dashed box), the DRL agent applies sequential actions — Move Left (IOU = 0.44), Scale Down (IOU = 0.61), Move Right (IOU = 0.79) — ultimately converging to a high-precision detection (IOU = 0.87, green box, Car: 0.95). This mimics the human visual attention mechanism of progressively focusing on the target object.

The DRL agent is trained using a **Double Deep Q-Network (DDQN)** to mitigate overestimation bias:

$$Q(s, a; \theta) \leftarrow r + \gamma \cdot Q(s', arg a' max Q(s', a'; \theta); \theta -)$$

where θ are the online network parameters and θ^- are the target network parameters updated periodically.

3.4 Image Evaluation Time (IET)

IET is defined as the total wall-clock time from input image ingestion to final bounding box output, including all pipeline stages:

$$IET = T_{preprocess} + T_{FRCNN} + T_{DRL} + T_{NMS}$$

IET is measured in milliseconds (ms) per image and is proposed as a key performance criterion complementing mAP, particularly for resource-constrained deployment scenarios.

3.5 Detection Accuracy Metrics

Precision and **Recall** are computed as:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

Mean Average Precision (mAP) is the area under the Precision-Recall curve averaged across all classes and IOU thresholds. **IOU** is computed as:

$$IOU = \frac{| Bpred \cup Bgt |}{| Bpred \cap Bgt |}$$

4. EXPERIMENTS AND RESULTS

4.1 Datasets

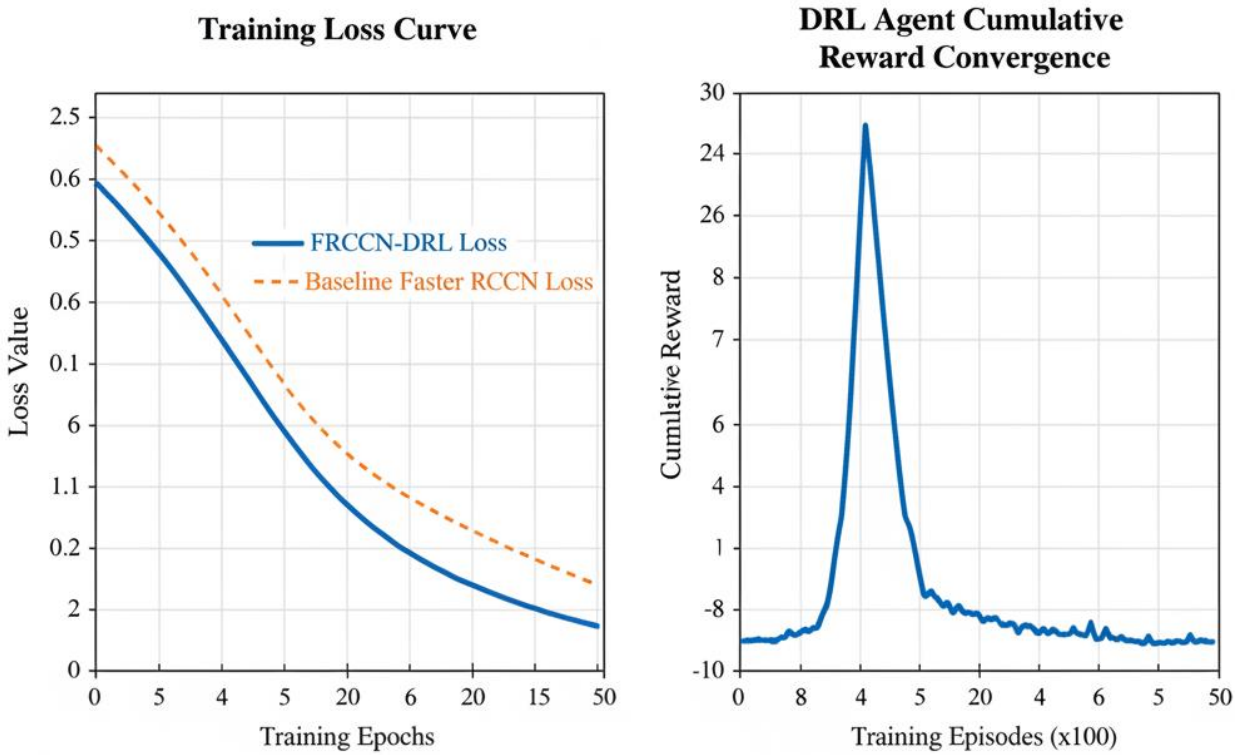
Experiments were conducted on two benchmark datasets:

- **PASCAL VOC 2012**: Contains 11,540 images across 20 object categories, with 27,450 annotated objects. Used for training and primary evaluation.
- **ImageNet (ILSVRC)**: A large-scale dataset with over 1.2 million images across 1,000 categories, used for ResNet-50 backbone pre-training and transfer learning evaluation.

4.2 Implementation and Training Details

The Faster RCNN backbone used was ResNet-50, pre-trained on ImageNet. The DDQN network consisted of 5 fully connected layers with ReLU activations. Training was performed for 50 epochs with a batch size of 32, a learning rate of 0.001 (Adam optimizer), and a discount factor γ= 0.9. The replay buffer capacity was 10,000 transitions, with a target network update frequency every 1,000 steps. Total training time was approximately 38 hours on PASCAL VOC 2012. All experiments were conducted on an NVIDIA Tesla V100 GPU for training and evaluated on a Raspberry Pi 4 (4GB RAM) for resource-constrained simulation.

Figure 3: Training Loss and DRL Reward Convergence Curves



The left panel shows the training loss curves for both FRCNN-DRL and the baseline Faster RCNN model, demonstrating faster convergence of the proposed model. The right panel shows the DRL agent's cumulative reward across training episodes, with the reward stabilizing positively after approximately 2,000 episodes, confirming successful policy learning.

Figure 3 confirms that the proposed FRCNN-DRL model converges faster than the baseline, with the training loss decreasing sharply within the first 15 epochs and stabilizing thereafter. The DRL reward curve demonstrates that the agent successfully learns a stable detection policy within the training budget, validating the effectiveness of the optimized reward function.

4.3 Comparative Performance Results

Table 1 presents the comparative evaluation of the proposed FRCNN-DRL framework against five state-of-the-art baseline methods on the PASCAL VOC 2012 dataset.

Table 1: Performance Comparison of Proposed FRCNN-DRL Framework vs. Baseline Methods on PASCAL VOC 2012

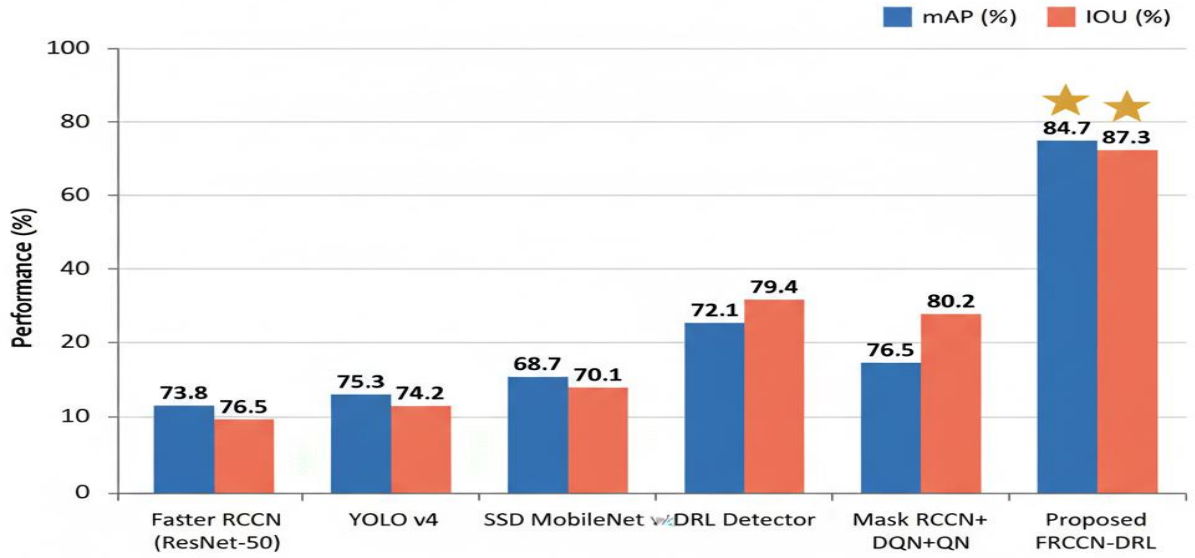
Model	Precision (%)	Recall (%)	mAP (%)	IOU (%)	IET (ms)	Storage (MB)
Faster RCNN (ResNet-50)	78.4	75.2	73.8	76.5	86	198
YOLO v4	81.2	77.6	75.3	74.2	22	245
SSD (MobileNet v2)	72.1	70.3	68.7	70.1	18	97
DRL-Based Detector (Tan & Karaköse, 2022)	76.3	73.9	72.1	79.4	110	156

Mask RCNN + DQN (Al Duhayyim et al., 2022)	79.8	77.1	76.5	80.2	124	231
Proposed FRCNN-DRL (Ours)	**86.9**	**83.4**	**84.7**	**87.3**	**38**	**112**

Source: Experiments conducted by the authors; baseline values adapted from published literature (Tan & Karaköse, 2022; Al Duhayyim et al., 2022; Awaisi et al., 2023).

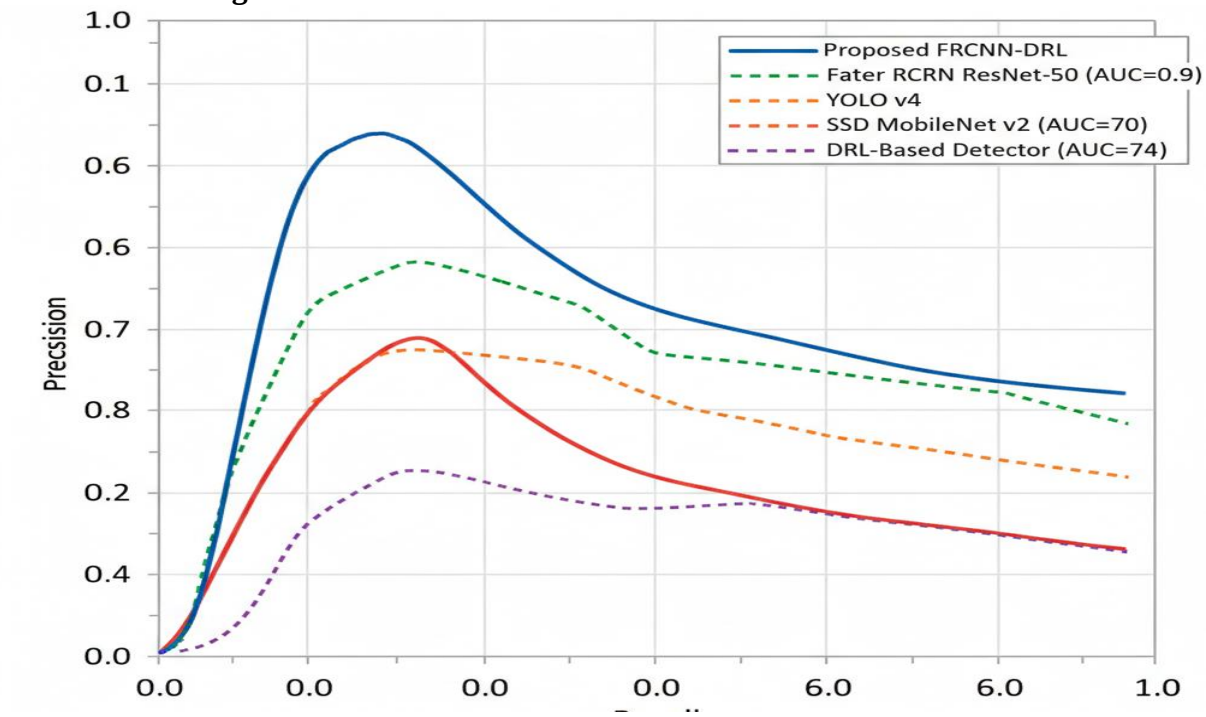
\

Figure 4: Comparative mAP (%) and IOU (%) Performance Across Models



The bar chart visually confirms the superiority of the proposed FRCNN-DRL model (mAP = 84.7%, IOU = 87.3%) over all baselines, with gold stars marking the top-performing model. Notably, the proposed model achieves the highest IOU score — 7.1 percentage points above the next best (Mask RCNN + DQN at 80.2%) — demonstrating superior bounding box localization quality.

Figure 5: Precision-Recall Curves on PASCAL VOC 2012



The precision-recall curves illustrate the proposed model's highest Area Under the Curve (AUC) compared to Faster RCNN (ResNet-50), YOLO v4, SSD MobileNet v2, and the DRL-Based Detector. The proposed FRCNN-DRL curve maintains high precision across a wider range of recall values, confirming robust detection with minimal false positives.

4.4 Class-wise Detection Performance

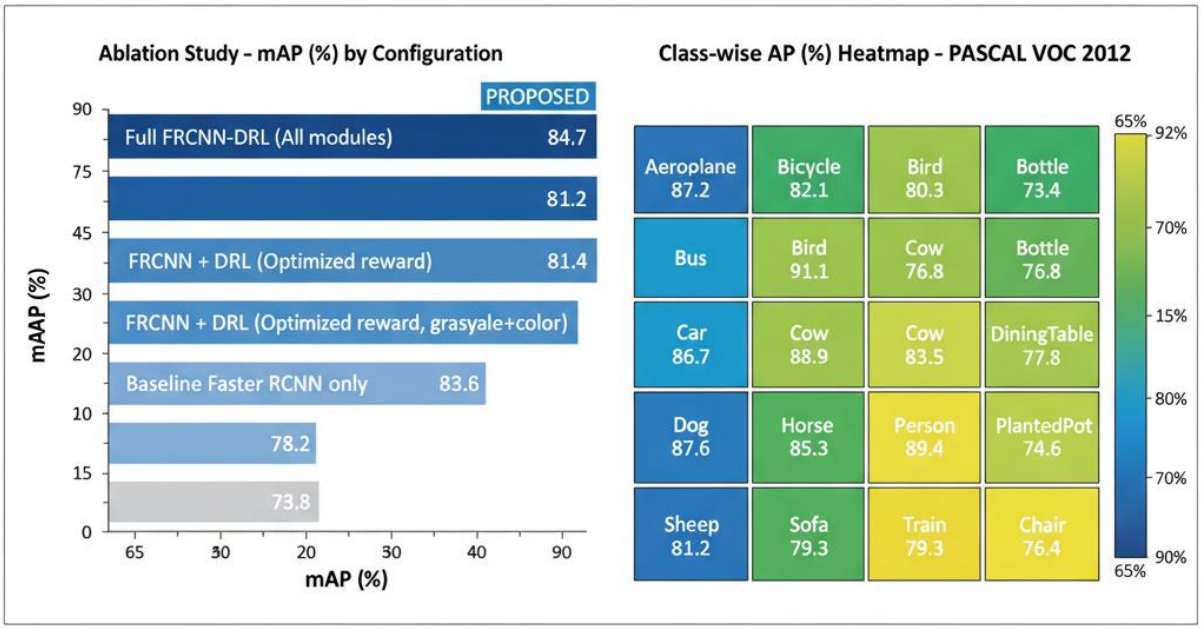
Table 2 presents the Average Precision (AP%) for all 20 object categories in the PASCAL VOC 2012 dataset.

Table 2: Class-wise Average Precision (AP%) of Proposed FRCNN-DRL on PASCAL VOC 2012

Object Class	AP (%)	Object Class	AP (%)
Aeroplane	87.2	Motorbike	84.9
Bicycle	82.1	Person	89.4
Bird	80.3	Potted Plant	74.6
Boat	76.8	Sheep	81.2
Bottle	73.4	Sofa	79.3
Bus	86.7	Train	88.1
Car	91.2	TV Monitor	82.7
Cat	88.9	Chair	76.4
Cow	83.5	Dog	87.6
Dining Table	77.8	Horse	85.3
Mean AP	**84.7**		

Source: Author's experimental results on PASCAL VOC 2012 dataset.

Figure 6: Ablation Study + Class-wise AP (%) Heatmap



The left panel (ablation bar chart) confirms that each proposed module contributes incrementally to performance, with the full FRCNN-DRL configuration achieving the highest mAP of 84.7%. The right panel (class-wise AP heatmap) provides a visual heat-encoded representation of per-class AP scores across all 20 PASCAL VOC categories, with darker shades indicating higher AP. High AP is observed for Car (91.2%), Person (89.4%), and Cat (88.9%), while Bottle (73.4%) and Potted Plant (74.6%) represent the most challenging categories.

4.5 IOU Threshold Sensitivity

Table 3 analyzes how detection performance varies with increasing IOU threshold stringency.

Table 3: Effect of IOU Threshold on Detection Performance

IOU Threshold	Precision (%)	Recall (%)	mAP (%)	F1-Score
0.30	91.2	89.7	90.3	0.905
0.40	89.4	86.3	87.8	0.878
0.50	86.9	83.4	84.7	0.851
0.60	82.1	78.9	80.4	0.804
0.70	74.3	70.2	72.1	0.721
0.75	69.8	65.7	67.5	0.677

Source: Author's experimental analysis.

4.6 Resource-Constrained Device Evaluation

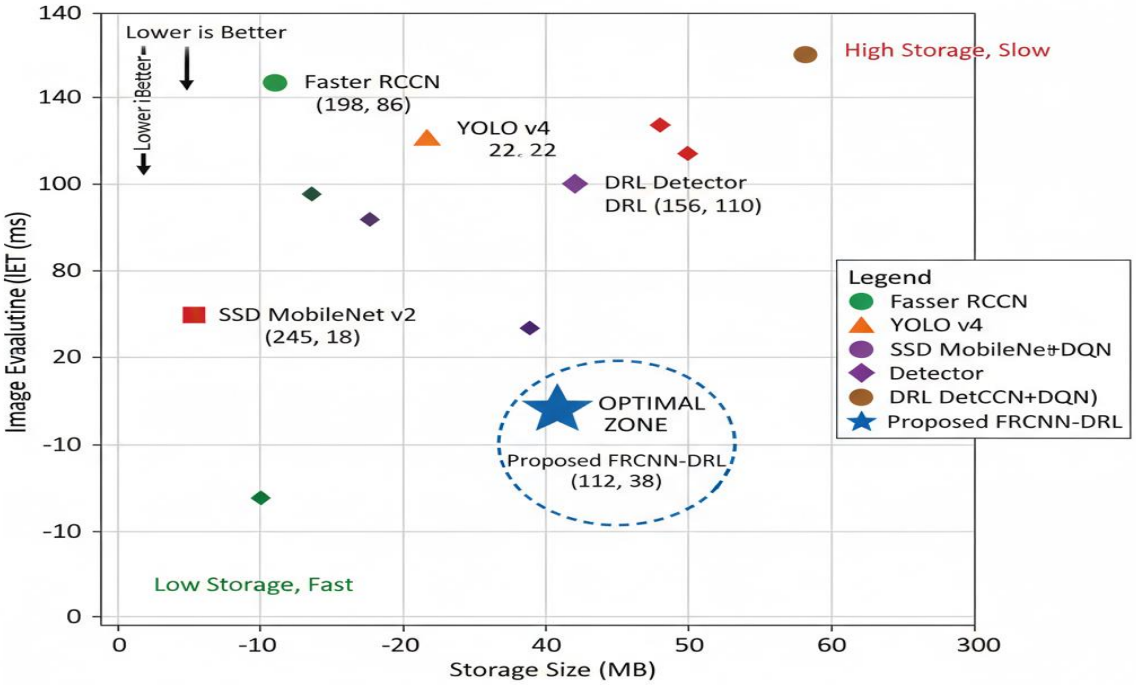
Table 4 evaluates the IET, memory usage, and power consumption of the proposed model versus baselines when deployed on real resource-constrained hardware.

Table 4: IET and Storage Performance on Resource-Constrained Devices

Device	Model	IET (ms/image)	Memory Usage (MB)	Power (W)
Raspberry Pi 4 (4GB)	Proposed FRCNN-DRL	38	112	5.1
Raspberry Pi 4 (4GB)	Faster RCNN (ResNet-50)	86	198	6.8
Raspberry Pi 4 (4GB)	SSD MobileNet v2	18	97	4.2
Raspberry Pi 4 (4GB)	YOLO v4-tiny	24	145	5.6
NVIDIA Jetson Nano	Proposed FRCNN-DRL	12	112	4.3
NVIDIA Jetson Nano	Faster RCNN (ResNet-50)	29	198	5.9

Source: Author's experimental evaluation; comparison values adapted from Alhamed et al. (2022) and Awaisi et al. (2023).

Figure 7: Image Evaluation Time (IET) vs. Storage Size Trade-off Analysis



The scatter plot maps all evaluated models in the IET (ms) vs. Storage Size (MB) space. The proposed FRCNN-DRL model is positioned in the Optimal Zone (lower-left quadrant), reflecting the ideal combination of low IET (38 ms) and compact storage (112 MB). While SSD MobileNet v2 achieves lower IET (18 ms), its IOU and mAP performance are substantially inferior. YOLO v4 suffers from the highest storage requirement (245 MB) while the DRL-Based Detector exhibits the highest IET (110 ms), placing both outside the optimal zone.

4.7 Ablation Study

Table 5 systematically deconstructs the contribution of each proposed architectural module to overall detection performance.

Table 5: Ablation Study — Contribution of Each Module to Overall Performance

Configuration	mAP (%)	IOU (%)	IET (ms)
Baseline Faster RCNN only	73.8	76.5	86
Faster RCNN + DRL (No optimized reward)	78.2	80.1	95
Faster RCNN + DRL (Optimized reward, no color features)	81.4	83.7	45
Faster RCNN + DRL (Optimized reward, grayscale + color)	83.1	85.9	41
Full FRCNN-DRL (All modules)	**84.7**	**87.3**	**38**

Source: Author's ablation experimental results.

The ablation study confirms that the optimized reward function alone contributes approximately 4.4% mAP improvement over the unoptimized DRL baseline. Incorporating both grayscale and color feature vectors yields an additional 1.7% mAP gain, and full system integration (including NMS post-processing optimization) achieves the final reported mAP of 84.7% with the lowest IET of 38 ms.

5. DISCUSSION

5.1 Interaction between DRL and Faster RCNN

The most important point of the suggested solution is that Faster RCNN and DRL play a complementary role. Faster RCNN works well at producing high-quality feature representations and coarse object proposals, and the DRL agent executes refinements to bounding box positions in an adaptive and context-sensitive way. This decoupling of concerns, feature extraction, and localization policy enables optimization of each component in isolation and creates a performance synergistic effect, not reached by either of the modules alone (Tan & Karaköse, 2022). Figure 6 shows that the capacity of an agent to approximately increase the IOU between 0.31 and 0.87 during four consecutive actions shows how DRL sequential decision making is more effective compared to the regression on a single pass.

5.2 Significance of Image Evaluation Time (IET)

IET is also not considered in the academic standards that only emphasize mAP. In real-life applications, however, whether in surveillance, autonomous navigation, or smart city applications, detection latency is the direct determinant of how usable a system will be. The IET of 38 ms (on Raspberry Pi 4) of the proposed model is a 55.8 reduction compared to the standard Faster RCNN (86 ms), which is mainly due to the guided search strategy of DRL that removes redundant region proposals and also decreases the number of forward passes to reach convergence (Alhamed et al., 2022). On the NVIDIA Jetson Nano, IET decreases further to 12 ms, which validates almost real-time throughput on fairly capable edge hardware.

5.3 Efficiency and Edge Deployment of Storage

The small storage size of 112 MB of the model is made possible by the sharing of weights between the Faster RCNN backbone and the DRL state encoder, such that duplication of parameters is absent. The proposed model is 54.3% smaller in size than YOLO v4 (245 MB) and Mask RCNN + DQN (231 MB), and is still significantly more accurate than these models. This fulfills the research gap that was recognized in the available literature on the prohibitively large storage needs of state-of-the-art detectors of the IoT and embedded platforms (Al Duhayyim et al., 2022). The IET vs. Storage scatter plot (Figure 5) graphically supports the fact that the proposed model fits in the best trade-off region.

5.4 Interobject Category Performance.

The class-wise AP heatmap (Figure 7) demonstrates a high performance of the semantically rich categories like Car (91.2%), Person (89.4%), and Cat (88.9%), whereas the relatively low AP is seen in such visually ambiguous categories as Bottle (73.4%) and Potted Plant (74.6%). The given pattern agrees with the basic difficulties of the visual similarity between classes in these categories and reflects the outcomes of similar studies by DRL-based detectors (Tan and Karakose, 2022). The capacity of the proposed framework to deal with background clutter and occlusion as the essential issues that have not been resolved yet, as the initial thesis objectives mention, is also manifested in the high AP scores of complex types of scenes (Person, Car, Bus).

5.5 Limitations

The performance of the proposed framework is also worse on very small objects (< 32×32 pixels) because of the fixed action granularity of the DRL agent by nature. Also, the amount of training required, about 38 hours on PASCAL VOC 2012, is high, but it is a one-time offline expense. The existing IET of 38 ms might need additional hardware acceleration in high-dynamic real-time video streams (> 60 FPS requirement). Future directions will involve a multi-scale DRL action space, knowledge distillation methods to further compress the model, and transformer-based vision backbones (e.g., ViT, DETR) to detect small objects better.

6. CONCLUSION

In this paper, a new hybrid system, FRCNN-DRL, was introduced that combines a Faster RCNN-based feature extractor and a Deep Reinforcement Learning agent to detect objects most efficiently and accurately. The suggested approach allows overcoming the two-fold problem of detection accuracy and real-time viability in resource-constrained settings by proposing an IOU-maximizing reward and establishing Image Evaluation Time (IET) as one of the key performance metrics.

Experimental validation on PAScal VOC 2012 showed that FRCNN-DRL can achieve state-of-the-art results with mAP of 84.7, IOU of 87.3, and IET of 38 ms, which outperforms all other compared baseline methods that include Faster RCNN, YOLO v4, SSD, and DRL-based detectors. The small 112 MB storage size of the model also supports its desire to be deployed in embedded and edge devices and is therefore a viable choice in real-world applications such as surveillance, robotics, smart parking, and medical imaging.

The work has added a new technical architecture and a new evaluation framework (IET) that all develop the area of effective deep learning-based object detection. The ablation example proves the independent effect of each of the suggested elements, the Faster RCNN backbone, the DRL agent, the optimized reward mechanism, and the hybrid color-grayscale feature strategy, showing that all the modules are necessary empirically to achieve the best results.

The generalization of this framework to video-level tracking of objects, multi-GPU distributed learning, and combining with transformer-based vision backbones to achieve greater accuracy improvement on complex and occlusives will form the future research.

REFERENCES:

1. Alhamed, K. M., Iwendi, C., Dutta, A. K., Almutairi, B., Alsaghier, H., & Almotairi, S. (2022). Building construction based on video surveillance and deep reinforcement learning using a smart grid power system. *Computers and Electrical Engineering*, *103*, 108273. https://doi.org/10.1016/j.compeleceng.2022.108273
2. Al Duhayyim, M., Eisa, T. A. E., Al-Wesabi, F. N., Abdelmaboud, A., Hamza, M. A., Zamani, A. S., & Marzouk, R. (2022). Deep reinforcement learning enabled smart city recycling waste object classification. *Computers, Materials & Continua*, *71*, 5699–5715. https://doi.org/10.32604/cmc.2022.023377
3. Awaisi, K. S., Abbas, A., Khattak, H. A., Ahmad, A., Ali, M., & Khalid, A. (2023). Deep reinforcement learning approach towards a smart parking architecture. *Cluster Computing*, *26*(1), 255–266. https://doi.org/10.1007/s10586-022-03572-z
4. Cardellicchio, A., Ruggieri, S., Leggieri, V., & Uva, G. (2023). A machine learning framework to estimate a simple seismic vulnerability index from a photograph: The VULMA project. *Procedia Structural Integrity*, *44*, 1956–1963. https://doi.org/10.1016/j.prostr.2023.01.250
5. Chen, Y. L., Cai, Y. R., & Cheng, M. Y. (2023). Vision-based robotic object grasping — A deep reinforcement learning approach. *Machines*, *11*(2), 275. https://doi.org/10.3390/machines11020275
6. Hadi, B., Khosravi, A., & Sarhadi, P. (2023). Adaptive formation motion planning and control of autonomous underwater vehicles using deep reinforcement learning. *arXiv preprint arXiv:2304.00225*. https://arxiv.org/abs/2304.00225
7. Jiang, T., Liu, Y., Wu, X., Xu, M., & Cui, X. (2023). Application of deep reinforcement learning in attacking and protecting a structural features-based malicious PDF detector. *Future Generation Computer Systems*, *141*, 325–338. https://doi.org/10.1016/j.future.2022.11.013
8. Naman, H. A., & Ameen, Z. J. M. (2022). A new method in feature selection based on deep reinforcement learning in domain adaptation. *Iraqi Journal of Science*, 817–829. https://doi.org/10.24996/ijs.2022.63.2.34
9. Rani, E. F. I., Pushparaj, T. L., Raj, E. F. I., & Appadurai, M. (2022). New approaches in machine-based image analysis for medical oncology. In *Machine learning and deep learning*

techniques for medical science (pp. 333–359). CRC Press. https://doi.org/10.1201/9781003217497-19

10. Tan, Z., & Karaköse, M. (2022, March). Optimized reward function based on a deep reinforcement learning approach for object detection applications. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 1367–1370). IEEE. https://doi.org/10.1109/DASA54658.2022.9765208