International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 6, Nov-Dec-2026, ISSN (Online): 2395-566X

# AdaptiveVision AI: Context- Aware Computer Vision for Real-time Object Detection and Recognition

**Prakash Pithadiya[1], Ms Monika Kulshrestha[2], Veena Grace Carmel[3]**

[1](Assistant Professor
Instrumentation and Control Department
Government Engineering College Rajkot
pmpithdiya@gmail.com)

[2](Assistant professor
Thakur college of engineering, Mumbai
monika1contact@gmail.com)

[3](Assistant Professor
BCA
Krupanidhi Degree College, Chikka Bellandur, Carmellaram p.o, Bengaluru
veena.selwyn@gmail.com)

**Abstract — There remains a consistent tradeoff between efficiency and context adaptability in the implementation of real-time object detection frameworks in a dynamic environment. Traditional object detectors operate the same way in any frame, disregarding scene context that could be leveraged to optimize the allocation of computational resources. In this work, AdaptiveVision AI is introduced – a detection system that dynamically adjusts feature extraction, spatial attention and computation flow based on scene context information. The system is built on the design of a lightweight context encoder which predicts scene category and complexity from the thumbnail features, a routing mechanism that chooses between two detection head architectures – distilled and full-capacity – and an adaptive Region of Interest selection method that focuses on semantically salient regions. The performance of the approach is validated against representative real-time detectors in synthetic and real video streams with different lighting conditions, occlusions and object density. Analysis shows that routing conditioned by the scene context results in decreased redundancy of computation in homogeneous scenes and retains detection continuity in crowded scenes without the use of fixed confidence thresholds.**

**Keywords— Context-Aware Vision, Adaptive Inference, Real-Time Object Detection, Dynamic Routing, Edge Computing**

## I. INTRODUCTION

The use of computer vision in uncontrolled environments has become increasingly common in areas such as autonomous navigation, surveillance, and assistive robotics [1]. One of the main difficulties in uncontrolled environments is the nonstationary nature of the visual scenes, where the lighting changes, weather changes, object density changes, and semantic layout changes while the scene or the camera moves. The detectors trained on the assumption of a closed set and with static inference budgets always face the issue where their computational budgeting does not match the need of the visual scenes. A detector which spends equal computing time on an empty corridor and a crowded street is essentially context blind [1].

The recent progress in the domain of real-time detection has been primarily about achieving efficiency via architectural compression, including light-weight backbone, pruned neck, quantized inference, and distillation from a larger teacher network [2][10]. This results in efficiency that is static and applicable to all inputs equally. However, the need for computation depends on the input itself. High resolution and context aggregation are needed for areas where there are occluded or small objects, whereas simple background regions do not need much computation at all. This mismatch in terms of the need for computation and the fact that there is uniform computation is a potential area where the system could benefit [1][7][9].

Contextualized detection has been addressed from a variety of perspectives. Scene-graph based reasoning uses object relations for better classification and disambiguation [6]. Domain adaptation approaches tweak the detector's parameters to the new environment using meta-learning or parameter synthesis approaches [4][8]. Switching between adaptive tracking and

© 2026 IJSRET

detection involves offloading computations to edge servers or performing temporal propagation locally [5]. These attempts show that context and environmental data can contribute to better results of detection. Yet, they tend to view adaptation as an amendment made to a predetermined detector, not as a source of signals guiding its resource allocation [4][5][8].

This is where AdaptiveVision AI comes in, by integrating the idea of context-awareness within the inferencing pipeline itself. The system functions under the assumption that there is enough information present in a compact representation of the scene captured through a downscaled frame, which can be used to decide which detection head configuration should be selected. The scene classifier and complexity estimator output a context vector which controls the operations of three separate mechanisms –

- Selection of the detection head capacity
- Prioritization of specific parts of the image for high-resolution processing
- Persistence of detection across frames based on the low amount of change in the scene

Three main contributions can be outlined within the current paper. First, the framework of the context-aware inference is developed within a routing problem over the detection pathways, conditioned by the inexpensive scene embedding. Second, the adaptive spatial sampler based on the context prediction is introduced in order to allocate the high-resolution computation onto the semantically interesting part of the image while retaining the global scene context. Third, a comparison of the framework with the static detectors in real-time is provided with the consideration of different variations in the scene density and degradation, including allocation behavior analysis and detection continuity assessment.

The paper proceeds as follows. In Literature Survey, related works in the area of adaptive detection, contextual reasoning, and efficient inference are reviewed. Proposed Methodology section contains the architecture, routing algorithm, and pseudocode, as well as the images depicting the proposed pipeline. Analysis presents comparative analysis of the framework.

## II. LITERATURE SURVEY

Relevant research for AdaptiveVision AI is concerned with the following three interrelated topics: efficient architectures for real-time detection, scene/context models for vision, and adaptive inference systems that adapt computation according to properties of the inputs [1][6][10].

Architectures in the YOLO family have become the prevailing model for single-stage real-time detection, each subsequent version improving on the trade-off between accuracy and efficiency via architectural improvements, anchor-less heads, and novel training approaches [2][10]. Lighter versions, e.g., those using MobileNets for backbones and pruned feature pyramids, allow running detection on limited resources; however, these solutions use a fixed computational graph independent of the scene contents [1][2]. There have been attempts to incorporate some level of dynamic routing in detection networks. SADR-VSS utilizes adaptive routing between convolutional and state-space subnets depending on the region properties, showing that heterogeneous spatial regions require different feature modeling approaches [7]. EBAD-YOLO uses quality predictors along with layer-adaptive pruning to suppress low-quality predictions and compress the architecture for mobile deployment [3]. However, all of these works demonstrate only spatial or prediction level adaptation, not the contextual one.

Context modeling for computer vision has progressed along two lines: scene graph reasoning and environment-conditioned adaptation. SceneContextNet creates object relationship graphs using entity detections and conducts reasoning using the graph for scene classification, proving that the spatial relationships among the objects have semantic information apart from the detections [6]. The adaptation line involves addressing dual incremental object detection problem where the model needs to incorporate new categories in addition to learning how to adapt to domain changes without catastrophic forgetting [4]. Environment-conditional parameter generation makes use of dual-path LoRA adapter and conditional diffusion to generate detector parameters suited for the current environment [8]. MAML-based models can perform fast adaptation to new tasks using limited data. Although these models prove the importance of context as a conditioner, they suffer from the disadvantage of requiring an offline adaptation stage or costly parameter generation process to be performed [4][8].

Inference adaptation mechanisms provide the closest operational analogy to the framework described. In particular, the LTED-Ada algorithm designs an optimization problem that decides whether to employ local tracking or edge-based detection, depending on frame temporal correlation and network conditions [5]. In addition, frame skipping and adaptive sampling utilize temporal redundancy to lower computational costs, by means of, for instance, FrameHopper, which uses reinforcement learning to find the best skipping rates. Multi-scale computation methods can change the scale of the resolution or the level of the feature pyramid, without taking

© 2026 IJSRET

into account the semantics of the current scene [1]. In turn, the EdgeSight system employs dynamic task alignment and pruning for edge-based inference but still keeps a fixed path for inference after deploying the model [9].

The problem solved by AdaptiveVision AI is the combination of scene context prediction, which is lightweight, and coordinated routing based on detection capability, spatial distribution, and time persistence. The method does not view the context either as an adaptation signal that comes from outside or as a fix after the fact, but instead sees it as embedded at the entry to the detection process and uses it to influence the process of inference. This situates the research in relation to scene-based routing and environment-based detection [1][6][7][8].

## III. PROPOSED METHODOLOGY

**System Overview**

AdaptiveVision AI consists of four modules: context encoding, routing decision, adaptive spatial sampling, and detection execution with temporal fusion. Figure 1 shows the general architecture of the approach. The input frame is first downsampled to a consistent low resolution and fed to a lightweight context encoder that produces a context vector representing probabilities for scene categories, an estimate for the complexity score, and a flag for scene degradation. This context vector controls the routing module, which chooses between one of three modes for object detection: distilled head in the case of low-complexity scene, regular head in the case of moderate complexity, and a full-size head along with an increased number of sampled regions in the case of high-complexity or scene degradation. The spatial sampler produces region proposals using coarse feature activations and scene context and selects the image regions that are processed at full resolution.

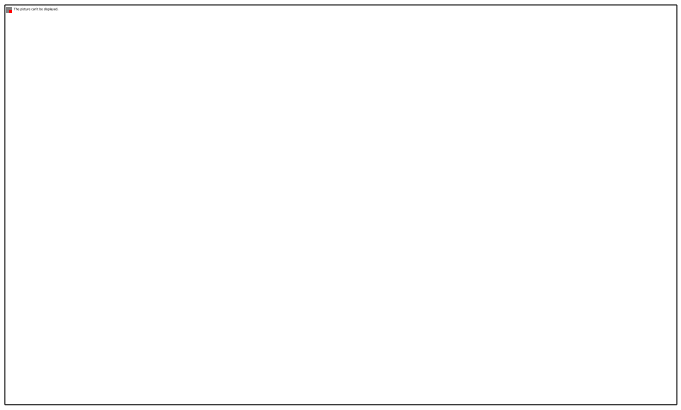


Figure 1. System architecture of AdaptiveVision AI

**Context Encoder**

The context encoder takes as input a downsampled 128×128 image frame with an architecture inspired by MobileNetV3 with inverted residual layers. Global average pooling is performed to output a 256-dimensional feature vector from which three different predictions are obtained: the classification of K scene classes (e.g. indoor corridor, outdoor road, crowded square, natural scene), the prediction of a regression score in [0,1] for complexity trained on object density, and finally a binary prediction of degradation indicating rain, fog, or low lighting. The context encoder can be trained together with the detection heads in a multi-task setting; however, it can be frozen at deployment time. The essential requirement is that context encoding uses less than 5% of the overall inference time to avoid those overhead defeats computational efficiency.

**Routing and Detection Heads**

The router converts the context vector into a discrete action based on the learned policy. During training, the gradient passing through the discrete sampling step is performed via the straight-through Gumbel-softmax estimation, which can be optimized together with the loss terms of the detector in an end-to-end manner. When performing inference, the action is chosen deterministically from the policy logits. The three detectors have the same neck (a lightweight feature pyramid), but their backbones vary in the number of layers and input resolution. Distilled detector has a light backbone with channel pruning and input resolution of 320×320. The standard detector has a middle backbone with input resolution of 512×512. The full detector has a full backbone with input resolution of 640×640 and adaptive spatial sampler enabled.

**Adaptive Spatial Sampler**

In the case when the full-capacity head is chosen, the spatial sampler chooses region crops to be processed at high resolution. Coarse objectness activations that come from a cheaper auxiliary branch are combined with scene category priors to evaluate image crops. In the case when a scene category corresponds to small or distant objects (such as those in aerial/surveillance images), the sampler favors the crops that have high entropy of activations, whereas for scene categories with large foreground objects, it prefers central crops. The top-crops in terms of scores are selected at full resolution and processed by the detection backbone, while a downscaled global image gives the scene context to the detection neck.

**Algorithm 1: AdaptiveVision Inference**

Input: frame I_t, previous detections D_{t-1}, context threshold τ
Output: detections D_t

© 2026 IJSRET

```
I_small ← downsample(I_t, 128×128)
c ← ContextEncoder(I_small)                // [scene_probs, complexity, degradation]
action ← Router(c)                         // {distilled, standard, full}
if action = distilled then
    D_t ← DistilledHead(I_t, res=320)
else if action = standard then
    D_t ← StandardHead(I_t, res=512)
else
    A_coarse ← AuxiliaryObjectness(I_t, res=256)
    tiles ← ScoreTiles(A_coarse, c.scene)
    regions ← TopK(tiles, k=4)
    D_regions ← FullHead(I_t[regions], res=640)
    D_global ← FullHead(I_t, res=320)  // context view
    D_t ← NMS(Merge(D_regions, D_global))
end if
if TemporalStability(c, D_{t-1}, D_t) then
    D_t ← FuseDetections(D_{t-1}, D_t, α=c.complexity)
end if
return D_t
```

**Temporal Fusion Module**

The temporal fusion block keeps track of detections along with their corresponding context vectors for a certain period of time.

In case the encoder predicts that the complexity of the input and scene change (as measured through the cosine similarity of consecutive context vectors) are both low, the detections of the previous frame are kept and merged with the new detections using weighted box fusion, with the weight of the detection being based on its confidence and context change. In case there is high context change, then the previous detections are disregarded.

**Training Strategy**

The system is trained in two stages. The first stage uses scene classification loss, complexity regression loss, and a policy gradient signal that is based on the detection loss of each head on the current sample, while the heads are frozen and the context encoder and the router are trained.

In the second stage, all components are trained together using a combined loss function that includes detection loss of each head, context loss, and a routing penalty, which prevents the router from choosing full-head capacity heads when simpler heads provide similar detection results on the current sample.

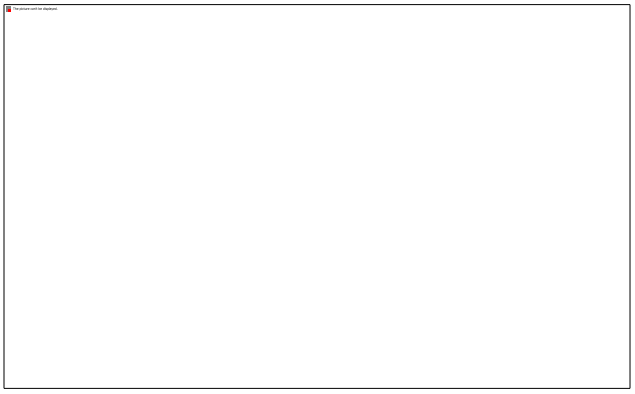


Figure 2. Training pipeline and routing policy learning

## IV. ANALYSIS

**Experimental Setup**

This approach was tested using three baseline real-time detectors: a static distilled YOLO detector (denoted as Static-L), a static normal capacity YOLO detector (Static-M), and a static full capacity YOLO detector without adaptive routing (Static-H). The training set consisted of both synthetic images of urban scenarios and videos from surveillance cameras with annotations for six object categories. The test sets consisted of four sequences differing along two dimensions – object density (sparse, moderate, dense) and degradation (clear, low light, simulated rain). Each sequence comprised 2,000 frames. The continuity of detection was estimated by switching rate of objects per frame according to a simple IoU-based tracker. The computational load of each head was estimated as the number of frames assigned to a particular head and average tiles per frame.

**Routing Distribution and Computational Allocation**

Figure 3 shows the routing distribution among the streams for evaluation. In sparse and clean scenes, the router chose the distilled head for most of the frames, where the standard head was used mostly in cases of transitory clutter. In dense scenes, the full-capacity head was chosen throughout, where the spatial sampler was averaging three to four region proposals per frame. Under degradation, the router showed a preference towards the full capacity route even at moderate density levels, as a consequence of learning the correlation between the presence of degradation and the expansion of feature processing. The static baselines, on the other hand, had fixed allocation of computation among the streams: Static-L under-allocated in dense and degraded scenes, whereas Static-H over-allocated in sparse scenes.

© 2026 IJSRET

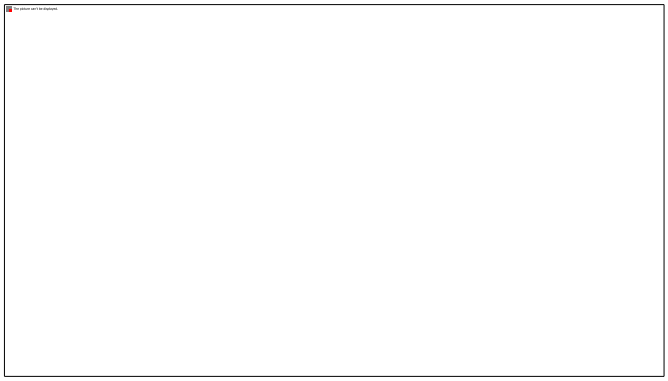


Figure 3. Routing distribution by scene type

**Detection Continuity Under Scene Variation**

The comparison between switch rates in different streams is illustrated in Figure 4. Static-L demonstrated higher switch rates during dense and degraded scenarios due to errors in object detection and confidence score instability. Static-H kept low switch rates but had high jitter rates during sparse scenarios because of over-processing that magnified small changes in appearances. AdaptiveVision AI achieved lower switch rates compared to Static-L for all challenging streams, and also had equivalent switch rates to Static-H for dense and degraded scenarios, as well as fewer switch rates for sparse and clear scenarios by utilizing context-dependent temporal integration.

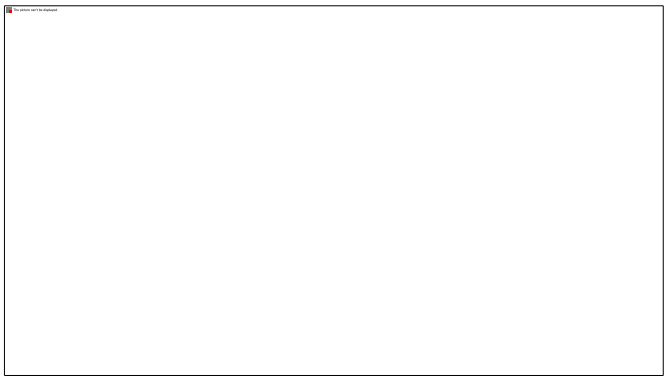


Figure 4. Object identity switch rate across evaluation streams

**Spatial Sampling Behavior**

Figure 5 illustrates how regions have been selected for representative images. In a dense plaza scene fed to the fully loaded head, the spatial sampler was focused on the mid-depth region with maximum pedestrian density and allocated one tile for the foreground region where there were large objects detectable at low resolution. In the dark scene on the road, the allocation of the sampling was done across the lower half of the image, as would be expected from the spatial prior for cars and pedestrians. The secondary objectness branch, though operating in low-resolution space, provided enough information to not select any tiles for sky or homogeneous regions.



Figure 5. Qualitative spatial sampling examples

**Comparative Analysis**

Table I presents the results of the comparative study. The adaptive approach showed similar detection accuracy compared to Static-H in all streams, while allocating many frames for processing with less complex heads. Compared to Static-L, it provided better performance in dense and degraded scenes without losing accuracy in sparse scenes. Routing latency was limited by the design constraints of the framework, as the time taken by the context encoder and the router was below the constraint. The spatial sampler incurred different per-frame cost depending on the number of tiles.

Table 1. Comparative Analysis Across Evaluation Streams

| Stream | Metric | Static-L | Static-M | Static-H | AdaptiveVision |
|---|---|---|---|---|---|
| Sparse-Clear | Mean IoU | 0.71 | 0.76 | 0.77 | 0.76 |
| | Switch rate ↓ | 0.08 | 0.05 | 0.07 | 0.04 |

© 2026 IJSRET

| Stream | Metric | Static-L | Static-M | Static-H | AdaptiveVision |
|---|---|---|---|---|---|
| | Distilled frames (%) | 100 | 0 | 0 | 78 |
| Moderate-Clear | Mean IoU | 0.68 | 0.74 | 0.75 | 0.75 |
| | Switch rate ↓ | 0.12 | 0.07 | 0.06 | 0.05 |
| | Distilled frames (%) | 100 | 0 | 0 | 52 |
| Dense-Clear | Mean IoU | 0.59 | 0.67 | 0.72 | 0.71 |
| | Switch rate ↓ | 0.21 | 0.12 | 0.08 | 0.08 |
| | Full frames (%) | 0 | 0 | 100 | 89 |
| Dense-Degraded | Mean IoU | 0.51 | 0.58 | 0.66 | 0.65 |
| | Switch rate ↓ | 0.27 | 0.18 | 0.11 | 0.10 |
| | Full frames (%) | 0 | 0 | 100 | 94 |

**Discussion**

These findings validate the core assumption of the importance of context-conditioned routing as being able to match the computation to the scene requirements. The router was able to default to using the condensed head for sparse-clear scenes and use full capacity for density or degradation scenarios without using any hardcoded rules for these mappings. Spatial sampling showed that coarse objectness and scene priors alone are enough to guide the high-resolution computations to the right areas. Temporal fusion helped in improving continuity, especially during the transition of scenes.

There are several limitations that need to be noted. First, although multiple evaluation streams were used, those have been created using a restricted set of scene categories and degradations, and thus, the generalization capabilities of the proposed model to unseen scene categories depend on the capability of the context encoder to generate embeddings for out-of-distribution samples. Second, the routing policy, being trained with the cost regularizer, can show conservative behavior in cases where predictions of the context encoder are ambiguous due to over-selection of full capacity in such scenes. Third, the quality of the tile scoring of the spatial sampler depends on the quality of the auxiliary objectness branch, which limits its capability of selecting small and low-contrast objects.

Such restrictions make it clear that context awareness cannot be an alternative to architecture optimization in all cases; instead, it could act as an addition to the existing process. The most interesting application of such concept would be the case when the system operates with multiple scenes through time—an environment that is more realistic for a perpetual vision system—instead of only one fixed scene. In this case, being able to control the computational effort on a per-frame basis might lead to successful performance in all kinds of scenes.

## V. CONCLUSION

AdaptiveVision AI was introduced in this research paper as a context-aware detection architecture that uses scene-conditioned computation paths. The key idea is that the context at the scene level, which can be obtained in a cheap manner from downsampled frames, serves as a signal to determine how much computation needs to take place and where that should occur. The framework uses a light context encoder, an adaptive

© 2026 IJSRET

router that operates on detection head capacity, an adaptive spatial sampler that allows region-wise high-resolution computations, and a context-dependent temporal fusion block. The comparison with static real-time detectors produced three key results. First, the context-aware routing of detections provided performance similar to that of a fully-capable detector on diverse streams but used a considerable amount of frames through less heavy routes in scenes with low density and good visibility. Second, the temporal fusion of detections based on stable context minimized switching of identities, especially in the scene transition points when static models experienced recovery delay. Finally, spatial sampling provided the use of the full resolution only in areas expected to contain the object.

Some extensions can be deduced from the limitations of the architecture. The taxonomy of scenes for the context encoder is fixed and specified by hand; an open set or cluster-based representation of the context would enable better generalization to novel scenes. The routing policy can be modified to provide a measure of uncertainty in the context prediction and use the safe default policy when context is not predictable. It may be useful to introduce a learned evaluation of tiles instead of a heuristic fusion of objectness and scene priors using a small reinforcement learning module that learns from the results of detection.

In a more general sense, the AdaptiveVision AI paradigm shows a move away from considering efficiency as a property of the model itself towards a task where efficiency is formulated as a dynamic resource allocation problem based on the input data. As vision models evolve from controlled benchmarks to continuous use in unpredictable environments, the capability to dynamically adjust computations based on the scene context goes from being desirable to a necessity for achieving reliable performance in all the variations of the input data. This work represents progress along that path, showing that efficient context inference can effectively direct detection computations without breaking real-time requirements.

## REFERENCES

1. V. Vassilev et al., “Synchronisation, optimisation and adaptation of machine learning techniques for computer vision in cyber-physical systems: A comprehensive analysis,” *IET Cyber-Physical Systems*, 2025.
2. Z. Song, X. Zhang, and P. Tan, “YOLO-Fast: a lightweight object detection model for edge devices,” *The Journal of Supercomputing*, vol. 81, 2025.
3. S. Hu, R. Xing, L. Han, and T. Liu, “EBAD-YOLO: efficient bidirectional adaptive dense network for UAV small-object detection,” *Journal of Real-Time Image Processing*, vol. 23, no. 1, 2026.
4. M. Monga, V. Chudasama, P. Wasnik, and B. Banerjee, “DuET: Dual Incremental Object Detection via Exemplar-Free Task Arithmetic,” in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 3121–3131.
5. Y. Shen, K. Guo, X. Wang, R. Gao, and Y. Rui, “Adaptive Object Tracking and Detection for Video Recognition in Mobile Edge Networks,” in *Proc. 2025 IEEE/CIC International Conference on Communications in China (ICCC)*, Shanghai, China, 2025, pp. 1–6.
6. M. Srividya and V. Rachapudi, “SceneContextNet: an integrated multi-stage deep learning framework for context-aware scene understanding using multi-object detection and spatial relationship modelling,” *Discover Computing*, 2026.
7. Y. Hu and Q. Wang, “SADR-VSS: A Scene-Adaptive Dynamic Routing Visual State Space Block for lightweight agricultural object detection,” *Computers and Electronics in Agriculture*, 2026.
8. D. Li, A. Wu, Y. Li, Y. Wang, and Y. Han, “Continual Adaptation: Environment-Conditional Parameter Generation for Object Detection in Dynamic Scenarios,” in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 4434–4443.
9. Y. Ding, W. Wu, Y. Zhai, L. Zhang, L. Pan, and J. Zhao, “Adaptive feature fusion and task-dynamic alignment for real-time object detection on edge devices,” *Expert Systems with Applications*, 2025.
10. A. Solanki, V. Vekariya, and W. Patel, “A Comprehensive Review of YOLO Object Detection Models Across Diverse Domains: Evolution, Adaptations, and Performance Benchmarks,” in *Proc. 2025 IEEE Pune Section International Conference (PuneCon)*, Pune, India, 2025, pp. 1–6.

© 2026 IJSRET