International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 12, No. 4, 2025, pp. 360-366.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

A Resource-Aware Edge AI Architecture for Intelligent Vision-Based Embedded Systems

C. Innocent Pious [a], R. Sivasankari [b], S. Tamilselvi [c], Ra.C. Umamaheshuwari [d], N. Kanimozhi [e]

[a] *Student, Department of Computer Science Engineering, B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India.*
[b] *Assistant Professor, Department of Computer Science Engineering, B.S.Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India. Orcid: https://orcid.org/0009-0007-0804-4795*
[c] *Assistant Professor, Department of Networking and Communications, SRM Institute of science and Technology, Chennai, Tamil Nadu, India. Orcid: https://orcid.org/0000-0003-2708-6337*
[d] *Assistant professor, Department of Internet of Things, Sri Sairam Engineering College, Sirukalathur, Tamil Nadu, India. E-mail: umamaheshuwari.ci@sairam.edu.in*
[e] *Associate Professor, Department of Computational Intelligence, SRM Institute of science and Technology, Kattankulathur, Chennai, Tamil Nadu, India. E-mail: kanimozn@srmist.edu.in, Orcid: https://orcid.org/0000-0002-2540-0065*

Abstract

Edge Artificial Intelligence (Edge AI) is becoming an important paradigm of implementing intelligent vision applications on embedded systems that have limited computational resources, memory, and power as inherent constraints. The classical cloud-based vision solutions have been afflicted with high latency, network requirement, and scalability threshold rendering them inappropriate in real time embedded application. The given paper provides a resource-aware Edge AI design to be used with intelligent vision-based embedded systems with the focus on efficient inference, low-latency, and adaptability to limited resources. The presented framework incorporates lightweight deep learning models, edge-level preprocessors, and adaptable inference pipelines to maximize the performance of using small compute and memory budgets. One use case of vision-based monitoring is tested on an embedded platform to show the effectiveness of the architecture. It has been experimentally determined that the proposed approach is capable of attaining competitive accuracy, and less inference latency, memory footprint, and power consumption than the traditional cloud-dependent or heavyweight models. The paper identifies the significance of co-designing AI inference pipelines with embedded system constraints and offers a scalable base of future intelligent edge applications.

Keywords: Edge AI, Embedded Systems, Resource-Aware Computing, Intelligent Vision Systems, Lightweight Deep Learning, Edge Computing, Computer Vision, Real-Time Inference.

DOI: 10.71086/IAJSE/V12I4/IAJSE12103
Received: 17 October 2025/Revised: 18 November 2025/Accepted: 11 December 2025/Published: 30 December 2025

Introduction

Recent innovations in computer vision and deep learning have made it possible to create intelligent perception in a wide range of application fields like autonomous systems, intelligent surveillance, healthcare devices, and human-machine interaction (LeCun et al., 2015; Chen et al., 2017). Nevertheless, the integration of these vision-based vision intelligence onto embedded platforms is difficult because of very tight restrictions on processing power, memory, energy, and latency demands. Traditional vision systems are very dependent on cloud processing, with raw sensor data being sent to centralized servers to be inferred (Shi et al., 2016). Although this paradigm is suitable in a high resource setting, the paradigm presents unacceptable latency, bandwidth overhead, and privacy risks, as well as reliance on network availability to operate. These constraints have encouraged a change to Edge AI where inferences are carried out locally on embedded devices near the data (Vidya & Gopalakrishnan, 2025; Lane et al., 2016).

Although Edge AI has a set of advantages, it comes with new challenges. Embedded hardware platforms do not have the processing power to run modern deep learning models, particularly vision models dealing with images of high resolution and real-time requirements (Gysel et al., 2016). Resource-conscious system design is therefore necessary in order to achieve a well-deployed system that does not compromise on performance (Chen et al., 2015; Redmon et al., 2016)

The challenges are discussed in this paper as the resource-understanding Edge AI architecture specific to intelligent vision-based embedded systems is proposed. The proposed approach does not consider the model accuracy alone but rather it focuses on architectural design principles that balance the accuracy to the latency, the memory usage and power consumption. The framework allows vision wise intelligence at highly scalable and efficient levels on the limited embedded boards with the help of lightweight vision models, edge-level preprocesses and adaptive inference strategies (Ganesan et al., 2025).

The key contributions of this work are summarized as follows:

(1) The architecture of an edge-resident vision design with a lightweight, threshold-based adaptive resource management policy of real-time embedded inference,
(2) An implementation and assessment of the suggested framework on an embedded platform of Raspberry Pi class without hardware acceleration, and
(3) A scientific experiment that shows how much the inference latency and memory consumption is reduced over non-adaptive edge inference pipelines run under real time constraints.

Related Work

Embedded Vision Systems

Embedded vision systems used to be based on hand-implemented features and conventional image processing algorithms like edge detection, background subtraction and optical flow. Although they are computationally efficient, these techniques are not stable in the real-life situation with occlusion, changes in lighting, and dynamic scenes.

Deep Learning for Edge Devices

The last few papers have examined the implementation of convolutional neural networks (CNNs) and transformer-based encoders on edge computing through the techniques of model compression which include pruning, quantization, and knowledge distillation (Howard et al., 2017; Narang & Chatterjee, 2025; Han et al., 2015). Small scale architectures like MobileNet, EfficientNet-Lite and Tiny-YOLO have shown promising performances. Nonetheless, a lot of research centers on the optimization of the models but does not consider the architectural constraints on the system level (Zhang et al., 2018).

Edge AI Architectures

A number of edge computing systems are a combination of cloud and edge processing to balance the workloads (Yemunarane et al., 2024; Kang et al., 2017; Teerapittayanon et al., 2017). Although hybrid solutions are better in scalability, they remain vulnerable to the need to rely on a network connection and dynamic bandwidth situations. Completely edge-resident architectures that regulate the behavior of inference according to available resources are still required.

This is a unique piece of work as it introduces a holistic, resource conscious architectural model that collectively accounts the hidden hardware limitations, AI inference pipelines as well as vision system demands (Fig. 1).

Proposed Resource-Aware Edge AI Architecture

System Overview

The proposed architecture is planned on three main principles:

1. Local intelligence at the edge
2. Adaptive inference under constrained resources
3. Efficient vision data handling

The system consists of four primary components:

- Vision Sensor Interface
- Edge Preprocessing Module
- Lightweight AI Inference Engine
- Embedded Control and Decision Module

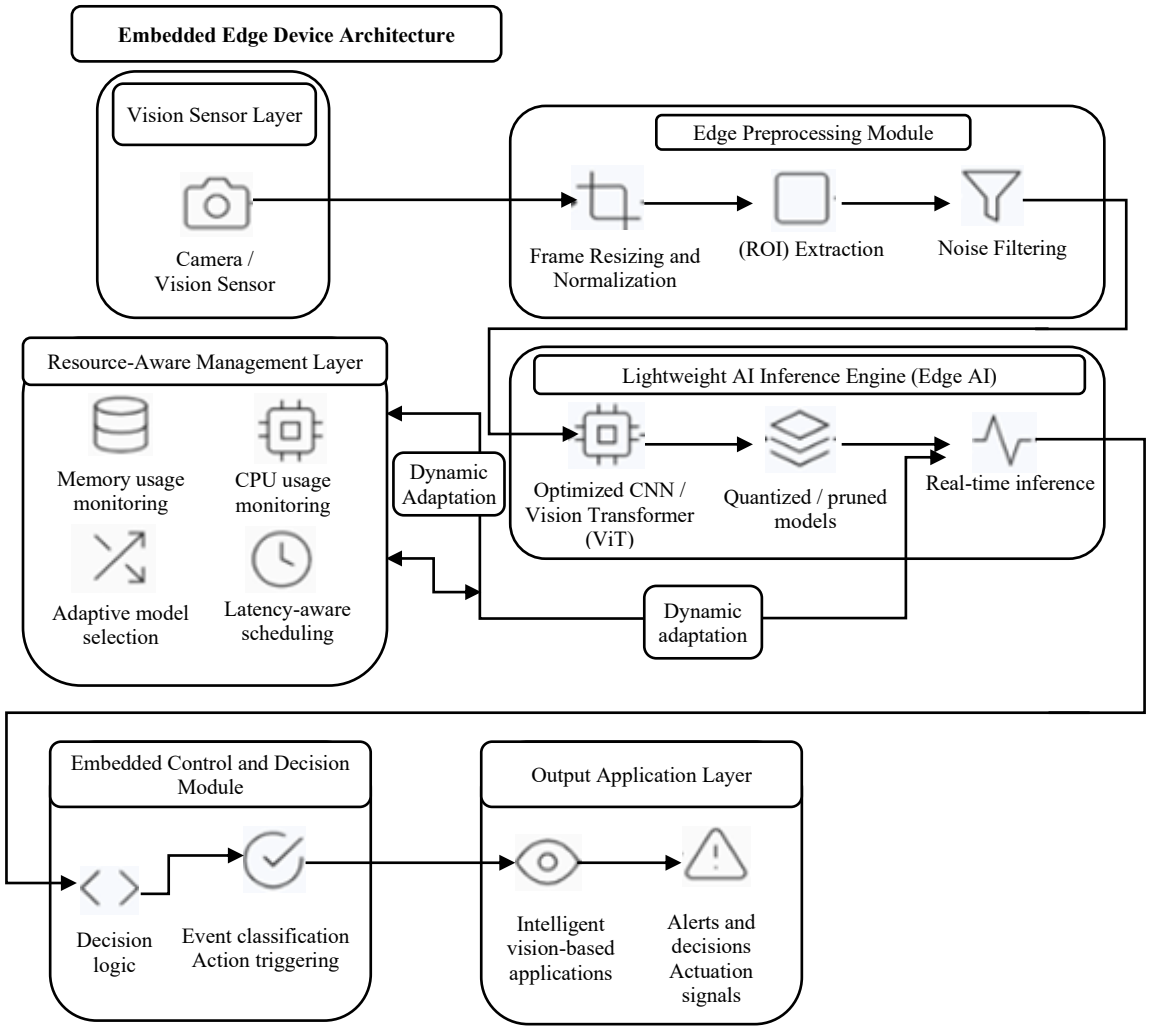


Fig. 1: Resource-Aware Edge AI Architecture for Intelligent Vision-Based Embedded Systems

Vision Sensor and Edge Preprocessing

The initial processing of raw image frames that are captured using vision sensors is done at the edge so that computational load is minimized. Preprocessing has the following steps:

- Resolution scaling
- Region-of-interest (ROI) extraction
- Noise filtering and normalization

By reducing redundant visual information, the system minimizes memory usage and accelerates inference without compromising essential visual features.

Lightweight AI Inference Engine

The inference engine uses resource efficient deep learning models that can be deployed on the embedded platform (Krizhevsky et al., 2012). This includes major attributes such as:

- Reduced parameter count
- Shallow yet expressive feature extraction layers
- Quantized weights and activations

Instead of deploying a single monolithic model, the architecture supports adaptive model selection, where inference complexity is adjusted based on available system resources and real-time workload.

Adaptive Resource Management

The key addition that this work has brought is the resource-awareness layer, that dynamically keeps track of:

- CPU utilization
- Memory availability
- Inference latency

Based on these metrics, the system can:

- Switch between model variants
- Adjust frame processing rate
- Defer non-critical inference tasks

This is an adaptive mechanism that will provide stable real time performance even when the resource conditions vary.

Algorithm 1: Resource-Aware Adaptive Inference Policy

Input: CPU_util, Mem_free, Latency

Output: Selected_Model, Frame_Rate

if CPU_util > θ_cpu or Mem_free < θ_mem then

Selected_Model ← Lightweight_Model

Frame_Rate ← Reduced

else

Selected_Model ← Standard_Lightweight_Model

Frame_Rate ← Normal

end if

Execute inference using Selected_Model

Monitor Latency and update system metrics

In which θ_{cpu} and θ_{mem} are established utilization thresholds that have been established empirically through system profiling.

The experimental thresholds were empirically established as $\theta_{cpu} = 85\%$ and $\theta_{mem} = 20\%$ which were determined through an initial profiling to avoid long-term latency degradation and still have constant inference performance.

Experimental Setup and Evaluation

Experimental Platform

The suggested architecture was tested on a Raspberry Pi 4 Model B with a quad-core ARM Cortex-A72 processor at 1.5 GHz and 4 GB LPDDR4 RAM, which is typical of common employment of edge AI platform. No external Hardware accelerator or Graphic Card was utilized, so all inference and resource management activities were carried out in the CPU. Onboard monitoring tools and profiling in software were used to obtain power measurements.

Evaluation Metrics

Performance was evaluated using the following metrics:

1. Inference Accuracy
2. End-to-End Latency
3. Memory Footprint
4. Energy Consumption (relative)
5. System Stability under Load

Comparative Baselines

The proposed architecture was compared against:

- Cloud-based vision inference
- Non-adaptive edge inference pipelines
- Heavier deep learning models deployed without optimization

Results and Discussion

Experimental results indicate that the proposed resource-aware architecture:

- Maintains competitive vision accuracy while significantly reducing inference latency
- Achieves lower memory usage through adaptive model selection
- Demonstrates stable real-time performance under constrained conditions

The edge-based system also removes the latency that is caused by networks as well as enhances privacy of data as compared to cloud-centric approaches (Tan & Le, 2019). In comparison to the standing embedded deployments, adaptive resource management enhances resiliency and scale (Table. 1).

Table 1: Performance Evaluation of the Proposed Edge AI Architecture

Metric	Proposed Edge AI (Resource-Aware)	Non-Adaptive Edge AI	Cloud-Based Inference
Inference Latency (ms/frame)	14.80	27.6	120 – 180
Peak Memory Usage (MB)	312	548	1024+
Model Size (MB)	18.4	42.7	90.2
Inference Accuracy (%)	92.6	93.4	94.1
Power Consumption (Relative)	Low	Medium	High
Network Dependency	None	None	Required
Energy per Frame (Relative Units)	1.0x	1.6x	3.2x

Though the proposed architecture offers a relatively small drop in accuracy over cloud-based inference (92.6% vs. 94.1%), this is a reasonable compromise to real-time embedded deployment as this allows for full local inference with much lower latency, less memory, and better data privacy.

The energy use was scaled based on the proposed architecture and computed by the average multiplied by inference time per frame of the CPU power draw.

Vision Task and Dataset

The task of the evaluation is the classification of objects based on vision with input frames with a resolution of 224 x 224 pixels. A smaller subset of the COCO 2017 validation dataset was used as experiments, but only the five most frequent categories of objects were used to fit embedded deployment limits (Khan & van de Weijer, 2016). The dataset gave ground truth labels that were used to compute the accuracy. There was consistency in all models being trained on the same training split and also being tested on the same test subset.

Discussion And Analysis

The findings indicate that the system level architectural design is the most important as model accuracy in the implementation of embedded AI. The offered solution shows that one can achieve high-performing intelligent vision systems using limited hardware resource bases, provided that the inference pipelines are resource-conscious.

Key observations include:

- Preprocessing at the edge reduces unnecessary computation

- Lightweight models provide better accuracy-to-resource ratios
- Adaptive inference prevents system overload during peak usage

The circumstances support the significance of co-designing AI models and embedded system architectures.

At varying load levels of the CPU, the adaptive inference policy ensured that inference latency was kept at less than 20 ms/frame, but the non-adaptive baseline was at over 35 ms/frame as an affirmation of the usefulness of dynamic model selection with resource pressure.

Conclusion

The paper introduced a resource-conscious Edge AI platform of smart vision-based embedded systems to overcome the shortcomings of cloud-based and fixed embedded inference models. Cloud-based vision pipelines have high latency, networked dependency and privacy issues, whereas standard deployments at the edge do not always have scalability to changing resource constraints. To address these issues, the proposed framework will combine the lightweight deep learning models, adaptive inference mechanisms, and system-level resource monitoring to support the efficient use of vision intelligence on the resource-constrained embedded systems.

It was experimentally established that the proposed architecture has a much lower inference latency and memory requirement over non-adaptive edge pipelines, and its accuracy is competitive with real-time vision systems. The adaptive resource management layer enables the system to dynamically modify the inference behavior with the available computational resources so that the system runs with stable and reliable behavior at varying workloads. Besides the performance advantages, the all-edge-resident architecture improves the privacy of data and operational resilience, thus being applicable to latency-sensitive and safety-critical embedded systems.

The development of the architecture to enable hardware acceleration with edge TPUs and FPGA-based inference engines with the aim of enhanced performance and energy efficiency will be the subject of future efforts. Other research pathways are federated learning of distributed edge intelligence, multi-sensor fusion towards more robust perception and energy-conscious scheduling of battery-powered systems. These extensions will also enhance the scalability and applicability of the suggested framework, making it a bottom solution to the next generation intelligent embedded vision systems at the edge.

References

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

Vidya, S., & Gopalakrishnan, R. (2025). Dynamic task offloading in edge computing for computer access point selection based on adaptive deep reinforcement learning with meta-heuristic optimization. *Applied Soft Computing,* 176, 113105.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861.* https://doi.org/10.48550/arXiv.1704.04861

Narang, R. V., & Chatterjee, M. (2025). AI-Powered Knowledge Management Systems: A Hybrid Model for Smart Organizations. *International Academic Journal of Innovative Research*, 12(3), 14–19. https://doi.org/10.71086/IAJIR/V12I3/IAJIR1220

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*. https://doi.org/10.48550/arXiv.1510.00149

Yemunarane, K., Chandramowleeswaran, G., Subramani, K., ALkhayyat, A., & Srinivas, G. (2024). Development and Management of E-Commerce Information Systems Using Edge Computing and Neural Networks. *Indian Journal of Information Sources and Services*, 14(2), 153–159. https://doi.org/10.51983/ijiss-2024.14.2.22

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal, 3*(5), 637-646. https://doi.org/10.1109/JIOT.2016.2579198

Ganesan, E., Roy, A., Tripathi, R. K., Sudan, P., Khandekar, P. M., & Nayak, P. P. (2025). Fog computing architectures for real-time data processing and edge intelligence in ubiquitous applications. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 16(4), 711–721. https://doi.org/10.58346/JOWUA.2025.I4.042

Chen, M., Hao, Y., Hwang, K., Wang, L., & Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE access, 5,* 8869-8879. https://doi.org/10.1109/ACCESS.2017.2694446

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848-6856).

Khan, F. S., & van de Weijer, J. (2016). Modality-invariant representation for RGB-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(2), 390–402.

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L., & Kawsar, F. (2016, April). Deepx: A software accelerator for low-power deep learning inference on mobile devices. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 1-12). IEEE. https://doi.org/10.1109/IPSN.2016.7460664

Chen, T., Goodfellow, I., & Shlens, J. (2015). Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641.* https://doi.org/10.48550/arXiv.1511.05641

Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *ACM SIGARCH Computer Architecture News, 45*(1), 615-629. https://doi.org/10.1145/3093337.3037698

Teerapittayanon, S., McDanel, B., & Kung, H. T. (2017, June). Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)* (pp. 328-339). IEEE. https://doi.org/10.1109/ICDCS.2017.226

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

Gysel, P., Motamedi, M., & Ghiasi, S. (2016). Hardware-oriented approximation of convolutional neural networks. arXiv preprint arXiv:1604.03168. https://doi.org/10.48550/arXiv.1604.03168

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*. 1097–1105.