Optimizing Distributed DNN Inference for Low Latency and Energy Efficiency on Edge Devices

Adiba Masud, Department of Computer Science
The University of Texas at San Antonio,

Supervising Professors: Maryam Tabar, Ph.D.

The increasing demand for real-time artificial intelligence (AI) in edge computing applications, such as smart surveillance, autonomous vehicles, and healthcare monitoring, has highlighted critical limitations in the deployment of deep neural networks (DNNs) in resource-constrained edge devices. These devices, including the Raspberry Pi and NVIDIA Jetson Nano, often suffer from limited compute power, fluctuating network conditions, and strict energy and latency requirements. Traditional distributed DNN inference relies on static partitioning strategies, which are not well suited for dynamic and heterogeneous edge environments. This inefficiency can result in performance bottlenecks, increased energy usage, and poor scalability. The recent literature explores various approaches such as model pruning, quantization, early exit mechanisms, and static DNN splitting strategies. However, these methods lack adaptability to real-time changes in the workload of the system and the availability of network resources. Some research has incorporated offloading and hybrid edge-cloud solutions, but they remain limited in their capacity to support fully distributed and intelligent edge-side inference. This research proposes an adaptive DNN partitioning framework that dynamically redistributes model layers across heterogeneous edge devices based on real-time resource metrics (CPU, memory, bandwidth). The system integrates fine-grained resource monitoring tools and employs reinforcement and heuristic learning techniques to optimize partitioning decisions. The framework is tested on a physical testbed composed of Raspberry Pi and Jetson Nano devices to validate performance in terms of inference latency, energy consumption, and computational efficiency. The proposed method advances the state of the art in distributed deep learning by introducing a scalable, intelligent solution for real-time inference at the edge. It contributes to both scientific understanding, by benchmarking dynamic scheduling strategies, and broader social impact through accessible and energy efficient AI that can be applied to healthcare, smart infrastructure, and digital equity initiatives.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

With the rapid advancement of deep learning (DL) technology and the growing adoption of Internet of Things (IoT) devices (Tu, Yang, and Cao,2025), industries are increasingly leveraging deep neural networks (DNNs) for edge computing applications. However, running DNN inference on edge devices presents major challenges, particularly latency and energy efficiency.(Xu et al.,2023). Traditional distributed inference methods are based on static partitioning, where the layers of the neural network are assigned to specific edge devices. Although this approach is effective in stable environments, it struggles in dynamic conditions, where network bandwidth and computational resources fluctuate. As a result, many edge AI applications suffer from performance degradation, increased energy consumption, and limited scalability. Research has shown that inefficient resource allocation in edge-based DNN inference can lead to more than 30% additional energy consumption and latency spikes exceeding 50%, significantly impacting real-time applications such as autonomous vehicles, healthcare monitoring and industrial automation. To overcome these limitations, optimizing distributed DNN inference for low latency and energy efficiency is essential to ensure scalability, reliability, and the broader adoption of AI-driven edge computing (Liu, Xu, Qiao, and Li, 2024). The figure 1.1 illustrates a distributed inference architecture where a pre-trained MobileNetV2 model is partitioned and deployed across two edge devices, coordinated by a central Lambda server.

- Lambda Server: Acts as the central controller responsible for loading the MobileNetV2 model and managing its partitioning. The model is divided into segments based on computational complexity and resource availability.
- MobileNetV2 Partitioning: The model is split into two parts, each color-coded to represent the portion assigned to a specific edge device. This could be done at a specific layer (e.g., after a convolutional block) to balance inference load.
- MobileNetV2 Partitioning: The model is split into two parts, each color-coded to represent the portion assigned to a specific edge device. This could be done at a specific layer (e.g., after a convolutional block) to balance inference load.
- Edge Device 1: Executes the first segment of the model. It receives the input data (e.g., an image), performs early-stage feature extraction, and forwards intermediate outputs to the next device.
- Edge Device 2: Completes the remaining portion of the model. It receives the intermediate features from Device 1, processes the final layers, and generates the inference result.
- Inter-Device Communication: Arrows between the devices represent the data flow of intermediate results, essential in partitioned inference pipelines.

Existing solutions for distributed DNN inference mainly focus on static model partitioning, where computational tasks are pre-assigned to edge devices without accounting for real-time variations in network conditions and hardware constraints(Mahmud, Kang, Desai, Lama, and Prasad,2024). Some recent approaches have explored model pruning, quantization, and offloading strategies to enhance efficiency, but these methods are not dynamically adjusted based on real-time resource availability. Moreover, current scheduling mechanisms often introduce computational overhead,

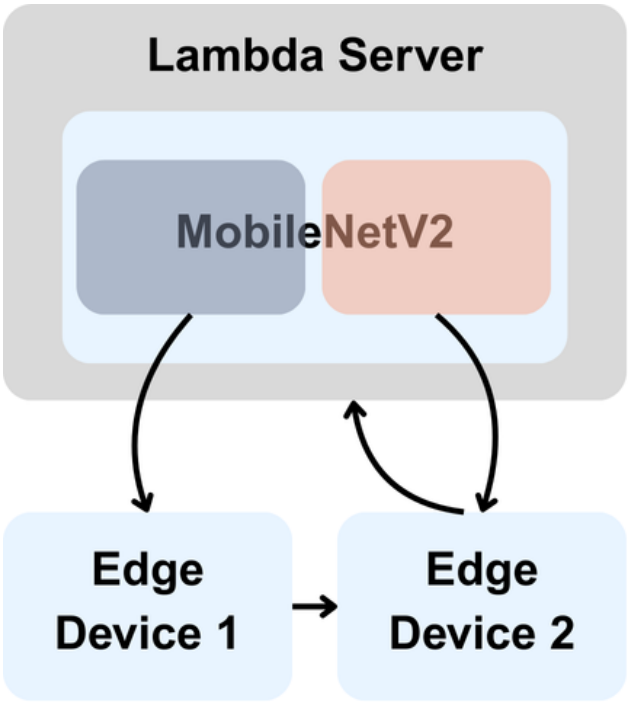

Figure 1.1: Distributed DNN Architecture

which negates potential efficiency gains. As a result, there remains a gap in achieving fully adaptive and resource-efficient DNN inference on edge devices. This figure 1.2 illustrates the proposed system architecture for adaptive workload partitioning in distributed deep neural network (DNN) inference across heterogeneous edge devices, specifically a Raspberry Pi and an NVIDIA Jetson device. The process begins with an input image which is partitioned and distributed by an adaptive scheduler. Based on the current resource conditions (e.g., CPU load, memory availability, network latency), the input and subsequent model layers are dynamically assigned to the most appropriate devices.

In this scenario, initial convolutional layers (Conv1 and Conv2) are executed on the Raspberry Pi, while intermediate computations and later layers are offloaded to the Jetson platform. To ensure continuity of the model's execution pipeline, intermediate data (feature maps) are exchanged between devices at each stageQin, Sun, Hofmann, and Vucinic,2024. Meanwhile, lightweight padding or synchronization data is handled by a mobile or secondary device, demonstrating a tri device collaboration model. The final inference output such as the classification label Cat is generated after the complete execution pipeline is reconstructed across devices.

This research proposes an adaptive DNN partitioning framework that can dynamically redistribute computational layers between multiple edge devices. By incorporating real-time resource monitoring of CPU, memory, and network bandwidth, this framework aims to optimize workload distribution, reduce latency, and minimize energy consumption. The ultimate goal is to improve the scalability and performance of edge AI applications by leveraging fine-grained resource tracking and intelligent scheduling mechanisms.The figure 1.3 presents a system architecture for adaptive and distributed deep neural network (DNN) inference. The core idea is to split a DNN into multiple parts and deploy them across both edge servers and local edge devices to improve efficiency and reduce latencyZeng, Chen, Zhou, Yang, and Zhang,2020.

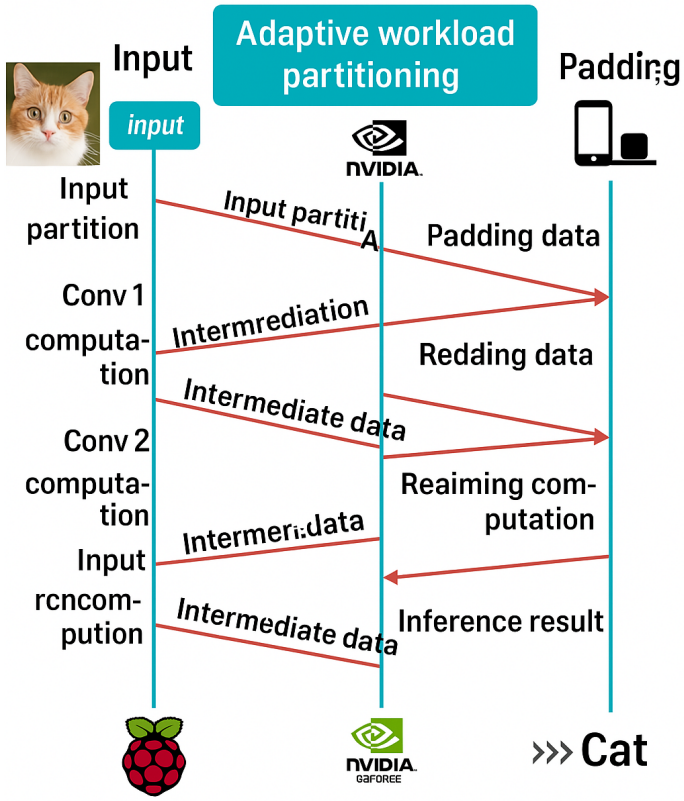

Figure 1.2: Adaptive Workload Partitioning for Distributed Edge Inference

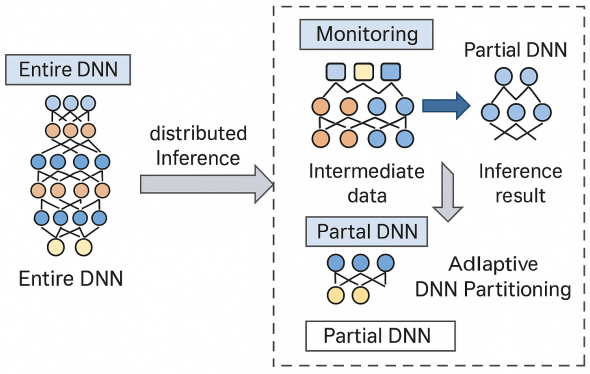

Figure 1.3: Distributed DNN Inference Across Edge and Device Layers

In this paper, we are inspired to investigate the problem of implementing a monitoring system to track CPU usage, memory consumption, and network bandwidth across heterogeneous edge devices. Developing an adaptive algorithm that dynamically redistributes model layers based on real-time resource availability. This paper experiments the evaluation on Heterogeneous Edge Devices which is testing the proposed framework on a testbed comprising Raspberry Pi and NVIDIA Jetson Nano devices to assess performance under varying conditions. By leveraging heuristic optimization and reinforcement learning to improve inference speed and reduce power consumption.

The figure above illustrates the proposed system architecture for adaptive workload partitioning of deep neural network (DNN) inference across heterogeneous edge devices, such as Raspberry Pi and NVIDIA Jetson platforms. The system begins with an input (e.g., an image), which is dynamically partitioned by a central module based on real-time resource availability (CPU load, memory, bandwidth). Each device is assigned a portion of the computational graph, such as convolutional layers or input segments, enabling the execution of collaborative inference.

In the example depicted, the input image is divided and distributed for parallel processing. The Raspberry Pi performs early-stage convolutional computations, while the Jetson device handles intermediate layers. The remaining computation and final layers are offloaded to a third device, such as a mobile phone or laptop. Throughout the process, intermediate data and feature maps are transmitted between devices to maintain model continuity and accuracy. By leveraging real-time monitoring and adaptive task scheduling, the system significantly reduces inference latency and enhances energy efficiency.

This approach enables scalable, low-latency inference in real-world edge scenarios such as smart surveillance, healthcare monitoring, and IoT applicationsâwhere lightweight, collaborative, and intelligent DNN execution is crucial.

The key contributions of this research are as follows.

- We introduce a novel, real-time optimization framework to dynamically partition DNN workloads based on edge resource availability.
- The proposed framework improves the scalability of AI-driven edge computing, benefiting industries such as healthcare, autonomous systems, and smart cities.
- We propose a work which is Alignment with NSF Broader Impacts and the proposed research supports technological advancements that improve AI accessibility in resource-constrained environments, contributing to scientific knowledge and societal well-being.

By addressing the limitations of static DNN inference and introducing a real-time adaptive solution, this research will help bridge the gap between AI capabilities and the constraints of edge computing environments, ensuring efficient, scalable, and energy-efficient AI inference at the edge.

CHAPTER 2: LITERATURE REVIEW

2.1 Distributed and Parallel DNN Inference

Several researchers have explored distributed and parallel deep learning architectures to improve inference performance across multi-device environments. The work provided a foundational survey categorizing distributed DNN techniques, such as data, model, and pipeline parallelism, and discussed their scalability Ben-Nun and Hoefler,2019. While comprehensive, their work focuses on general-purpose systems and lacks attention to adaptive or edge-specific constraints Ben-Nun and Hoefler,2019. This work introduced Couper, a framework that partitions DNNs into slices and deploys them in containerized edge environments. Their approach improves deployment flexibility but does not incorporate dynamic adaptation to fluctuating system resourcesHsu, Bhardwaj, and Gavrilovska,2019. Similarly, Hu et al. (2022) proposed a method for distributing DNN layers across heterogeneous edge devices, demonstrating performance gains through optimal layer-to-device mapping. However, their approach is statically configured and lacks real-time responsiveness.

To tackle fine-grained model partitioning in mobile edge computing networks, showing how layer-wise division can better align with device resource capabilitiesXu et al.,2023. Although efficient, their method is semi-static and does not use adaptive or learning-based optimization. Mohammed et al. (2020) The authors proposed an adaptive offloading framework where partitioning decisions respond to resource availability. Despite its adaptability, the method primarily focuses on offloading and does not address reinforcement learning or fine-tuned scheduling across multiple edge nodesMohammed, Joe-Wong, Babbar, and Francesco, 2020.

Xu et al. introduced iGniter, which provides interference-aware GPU resource provisioning to ensure predictable inference in cloud environments. While relevant for workload isolation Xu et al.,2023, its application is cloud-centric and does not translate directly to edge computing. Lastly, Stahl et al. surveyed edge and embedded systems for AI workloads, highlighting hardware limitations and design considerationsStahl, Zhao, Mueller-Gritschneder, Gerstlauer, and Schlichtmann,2019 However, they did not explore distributed inference or dynamic workload balancing strategies.

In summary, while these studies present significant advancements in model partitioning, edge deployment, and performance optimization, there remains a critical gap in the development of a real-time, adaptive DNN partitioning framework. The proposed research builds on these foundations by integrating fine-grained resource monitoring, reinforcement learning-based scheduling, and multi-device collaboration to enable intelligent, scalable edge inference under dynamic conditions.

2.2 Dynamic Partitioning and Scheduling Methods

Recent advances in adaptive deep learning have focused on enhancing inference efficiency through dynamic scheduling and resource-aware computation. Hu et al. introduced Dynamic DNN Surgery, a technique for modifying network structure at runtime to reduce inference delay under changing system constraintsHu and Li,2022. While effective in edge environments, it lacks learning-based decision-making. Padmanabha Iyer et al. proposed a per-input compute adaptation method that adjusts computational effort based on input complexity, achieving improved efficiency but offering limited support for collaborative multi-device executionPadmanabha Iyer et al.,2024. Liu et

al. presented a comprehensive survey on deep reinforcement learning in distributed computing, highlighting its potential for real-time adaptation. However, practical implementation challenges such as training overhead and latency remain unresolvedHu, Bao, Wang, and Liu.

Su et al. explored joint DNN partitioning and resource allocation strategies to optimize inference in hierarchical edge-cloud systems. Their work successfully integrates task allocation with system-level constraints, though it primarily targets hybrid infrastructures rather than decentralized edge-only setupsSu et al.,2023. Mahmud et al. introduced a converting autoencoder for low-latency and energy-efficient DNN inference at the edge, showcasing architecture-specific improvements but with limited generalization across diverse models Mahmud et al.,2024. Lastly, Tu et al. outlined the major challenges in distributed machine learning for edge computing, including data heterogeneity, resource limitations, and privacy concerns, but offered few actionable architectural solutionsTu et al.,2025.

Together, these studies contribute valuable strategies for improving edge inference through adaptive design, resource-awareness, and machine learning. However, a gap remains in developing a unified framework that combines fine-grained resource monitoring, dynamic DNN partitioning, and learning-based scheduling, which this research aims to address.

2.3 System Optimization and Resource Management in Edge Environments

Lama et al. explored the challenge of performance isolation in multi-tenant cloud systems where multiple data-intensive applications compete for shared computational resourcesMahmud et al.. Their proposed method uses resource-aware scheduling algorithms that monitor CPU, memory, and I/O contention to isolate workloads and reduce performance degradation in a cloud environment. This study highlights the importance of ensuring fairness and predictability in shared infrastructures, which is highly relevant to distributed edge systems where multiple tasks often run concurrently. The authors demonstrated that performance isolation techniques can significantly improve latency predictability and throughput in complex systems. However, the method is primarily tailored for large-scale cloud deployments and lacks adaptation for low-power, resource-constrained edge devices, which limits its direct applicability to decentralized AI scenarios.

Masud et al. developed a Bengali image captioning system using an attention-based deep learning architecture. Although the primary goal of the study was multilingual caption generation, the modelâs compact architecture and visual attention mechanism contribute to improved inference efficiency. The authors fine-tuned the model for low-resource settings, demonstrating effective adaptation in terms of model compression and size-awareness, which are critical for deploying deep learning models on constrained edge devicesMasud, Hosen, Habibullah, Anannya, and Kaiser,2025. While this work does not focus on distributed inference or cross-device execution, its approach to optimizing model complexity and tailoring architectures for language-specific and domain-specific applications offers valuable insights for building lightweight models that can complement distributed edge inference frameworks.

Hou et al. proposed Distredge, a distributed framework for accelerating CNN inference across edge devices. Their method focuses on optimizing layer-wise scheduling of DNNs by distributing the feature map processing among devices based on fixed communication and computation budgetsHou, Guan, Han, and Zhang,2022. The framework significantly reduced inference delay by enabling early termination and workload-aware dispatching of DNN layers, making it suitable for bandwidth-constrained edge networks. The primary contribution of this work lies in its system-level optimization of CNN inference without modifying the model architecture itself. However,

Distredge relies on predefined system parameters and static partitioning policies, and does not incorporate dynamic adaptation or real-time decision-making based on fluctuating device loads or network conditions.

2.4 Research Objectives

To aid understanding of this research, several key terms are defined below:

- Edge Computing: A computing paradigm that enables data processing and inference close to the data source (e.g., IoT sensors or edge devices) to reduce latency and network overhead.
- Distributed DNN Inference: The process of executing different parts (layers or blocks) of a deep neural network across multiple computing devices.
- Partitioning: The act of dividing a DNN into smaller segments to distribute inference tasks.
- Adaptive Partitioning: A dynamic approach where DNN partitions are reassigned in real time based on current resource conditions (e.g., CPU load, memory availability, network speed).
- Heterogeneous Edge Devices: A mix of different types of edge hardware (e.g., Raspberry Pi, NVIDIA Jetson Nano) with varying computational and communication capabilities.
- Reinforcement Learning (RL): A machine learning technique that learns optimal actions based on reward feedback from the environment.

The primary goal of this research is to develop and evaluate a real-time, adaptive DNN partitioning framework for distributed inference across heterogeneous edge devices. The proposed system aims to dynamically assign model segments to edge nodes based on live system metricsâ such as CPU usage, memory availability, and network bandwidthâwith the objective of improving inference speed, energy efficiency, and scalability in edge AI applications.

To bridge these gaps, this research proposes a framework that integrates:

- Fine-grained layer-wise DNN partitioning tailored for edge environments,
- Lightweight, real-time resource monitoring agents deployed on each device,
- Heuristic- and reinforcement learning-based schedulers that dynamically optimize partitioning decisions,
- A hardware testbed featuring Raspberry Pi and Jetson Nano devices to validate system performance under real-world constraints.

Table 2.1: Summary of Related Work in Literature Review

Authors & Year	**Method**	**Main Contributions**	**Limitations**
Ben-Nun and Hoefler,2019	Survey of parallel/distributed deep learning	Categorized model, data, and pipeline parallelism in distributed AI systems	General-purpose focus; lacks edge-specific and adaptive considerations
Hsu et al..2019	*Couper* model slicing for containers	Enables piece-wise DNN slicing for edge container deployment	Static partitioning; lacks real-time adaptability
Hu et al.,2019	Static DNN distribution on heterogeneous edge devices	Maps DNN layers to edge devices based on resource profiles	No dynamic reassignment during runtime
Li et al.,2024	Fine-grained partitioning for mobile edge networks	Introduced resource-aware DNN layer splitting	Semi-static; does not use learning-based scheduling
Mohammed et al.,2020	Adaptive offloading based on resource states	Dynamic partitioning between cloud and edge	Limited to offloading; lacks fine-grained and collaborative scheduling
Xu et al.,2023	Interference-aware GPU provisioning (*iGniter*)	Predictable DNN inference under resource contention in cloud	Cloud-focused; not tailored for edge device constraints
Stahl et al.,2019	Embedded system survey for AI workloads	Highlights hardware limits in edge AI deployment	Descriptive; no inference or scheduling strategies provided
Hu and Li,2022	*Dynamic DNN Surgery*	Enables runtime model path optimization for inference	No learning-based decision-making or multi-device coordination
Padmanabha Iyer et al.,2024	Per-input compute adaptation	Tailors inference based on input complexity	Lacks scalability to distributed environments
Hu et al.,2019	Survey on DRL in distributed systems	Identifies potential for RL-based scheduling in distributed computing	High-level review; limited real-world deployment examples
Mahmud et al.,2024	Converting autoencoder for edge inference	Proposes low-latency, energy-efficient model architecture	Application-specific; limited generalizability across models
Tu et al.,2025	Review of distributed ML challenges in edge computing	Discusses key issues like privacy, heterogeneity, and latency	Lacks concrete architectural or scheduling solutions
Lama, Wang, Zhou, and Cheng,2018	Resource-aware scheduling in multi-tenant cloud	Improves fairness and predictability in shared systems	Focused on cloud; not transferable to low-power edge hardware
Hou et al.,2022	*Distredge*: CNN inference optimization on edge	Reduces latency via workload-aware scheduling	Uses static configuration; lacks adaptive rescheduling

CHAPTER 3: RESEARCH METHODS

This research aims to develop an adaptive deep neural network (DNN) partitioning framework designed to optimize distributed inference across heterogeneous edge devices. With the rapid expansion of edge-based AI applications and the limitations of resource-constrained devices like Raspberry Pi and NVIDIA Jetson Nano, there is a growing need for intelligent systems that can dynamically manage computational loads. Our objective is to reduce inference latency, minimize energy consumption, and improve system scalability by leveraging real-time resource monitoring and adaptive scheduling. To guide this investigation, we pose the following key research questions:

- **RQ1:** How can resource metrics such as CPU usage, memory availability, and network bandwidth be effectively monitored in real time across heterogeneous edge devices?
- **RQ2:** What adaptive algorithm can be developed to dynamically partition and assign DNN layers based on current system conditions?
- **RQ3:** How does the proposed adaptive framework compare to traditional static partitioning in terms of inference latency, energy efficiency, and overall performance?

3.1 Research Approach and Design

To address the challenges of scalable and efficient DNN inference in heterogeneous edge environments, this research adopts an experimental systems-based approach combined with applied machine learning techniques. The research design involves developing, implementing, and evaluating a prototype system that performs adaptive DNN partitioning based on real-time resource availability across multiple edge devices.

3.1.1 Data Collection and Testbed

The figure 3.1 shows the physical testbed setup used to evaluate the proposed adaptive DNN partitioning framework in a realistic edge computing environment. Now, to explain the devices:

- **Cluster of Raspberry Pis:**A set of Raspberry Pi 4 devices is used to simulate low-power, resource-constrained edge nodes. These devices are responsible for executing the the early layers of the deep neural network (DNN), performing initial inference steps, and collecting real-time system metrics such as CPU usage, memory utilization, and network statistics.
- **Jetson Nano Devices with Antennas:** Multiple NVIDIA Jetson Nano boards are configured as more powerful edge nodes, capable of running deeper or more compute-intensive layers of the DNN. These devices also handle later-stage inference tasks, contributing to a collaborative processing architecture. The attached antennas suggest the use of Wi-Fi modules or antennas for wireless data transmission, reflecting practical deployment in mobile or remote edge environments.
- **Cabling and Network Connectivity:** The wired connections visible in the image suggest a stable local network setup for controlled testing, although the presence of antennas indicates potential wireless communication testing as well. A shared power supply or hub may be used to provide consistent power to all devices.

Figure 3.1: Testbed

3.2 Rationale for Methodological Choice

The core motivation behind the chosen methodology is rooted in the need to address the limitations of traditional static DNN deployment methods in dynamic, heterogeneous edge environments. Edge devices, such as Raspberry Pi and NVIDIA Jetson Nano, vary significantly in computational power, memory capacity, and network stability. These devices are often deployed in environments where resource availability fluctuates unpredictably. As such, a static, one-size-fits-all approach to DNN inference is not sufficient. To overcome these limitations, this research adopts a real-time, adaptive systems-based approach supported by machine learning techniques (e.g., heuristic rules and reinforcement learning). This choice enables the system to make intelligent, data-driven decisions about where and how to execute different parts of a DNN based on live system feedback. The use of real-time resource monitoring ensures that the system remains responsive to changes in device load and network conditions, while the incorporation of reinforcement learning allows the framework to continuously improve its partitioning and scheduling decisions through trial-and-error learning. This dynamic adaptation is critical for maintaining low latency and energy efficiency in real-world, distributed edge computing scenarios.

Furthermore, the experimental testbed-based designâfeaturing Raspberry Pi and Jetson Nano devicesâprovides a practical environment to implement, validate, and benchmark the proposed framework under varying real-world constraints. This setup allows for accurate measurement of performance indicators such as latency, energy usage, and system scalability, making the research both reproducible and application-ready.

3.3 Proposed Methodology

To address the limitations of the deployment of static deep neural networks (DNNs) in resource-constrained and heterogeneous edge environments, this research proposes a comprehensive framework for dynamic DNN partitioning and distributed inference. The method begins with the development and training of DNN models, including fine-tuning on a central Lambda server, and proceeds through multiple deployment strategies to evaluate and compare performance across edge devices.

The framework 3.2 supports several implementation pathways:

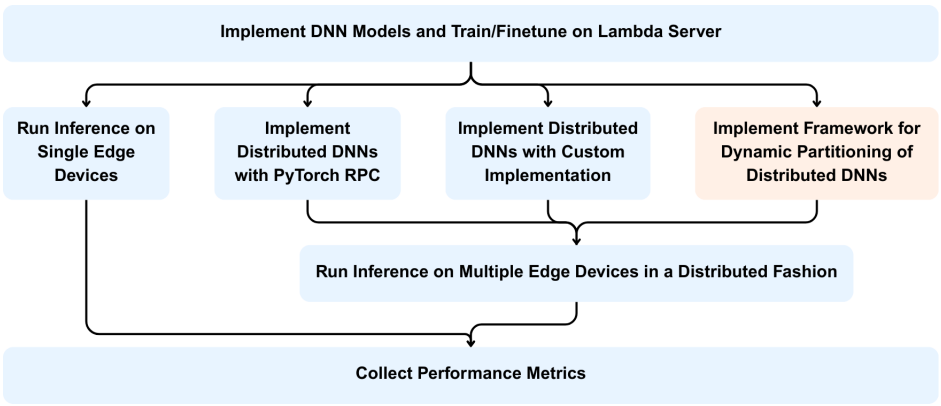

Figure 3.2: Proposed Methodology

- Inference on single edge devices to establish a performance baseline.
- Distributed inference using PyTorch RPC, allowing modular execution across nodes with minimal custom changes.
- Custom distributed DNN implementation, enabling finer control over data flow and execution scheduling between devices.
- A newly developed framework for dynamic DNN partitioning, which leverages real-time system metrics to intelligently divide and distribute DNN layers across edge devices.
- A newly developed framework for dynamic DNN partitioning, which leverages real-time system metrics to intelligently divide and distribute DNN layers across edge devices.

All implementations ultimately execute in a distributed inference environment involving multiple edge devices, where they are evaluated under diverse conditions. The system collects detailed performance metrics, including latency, energy usage, and resource utilization, to assess the impact of each deployment strategy and validate the efficiency of the proposed adaptive partitioning framework. This structured methodology not only enables direct comparison of various distributed DNN execution strategies but also emphasizes real-world applicability and scalability by accounting for varying hardware capabilities and dynamic runtime conditions.

3.3.1 Key Components of the Proposed Framework

1. Real-Time Resource Monitoring: Lightweight agents are deployed on each edge device to monitor system metrics, including:

- CPU utilization,Memory availability, Network latency and bandwidth.

These metrics are continuously sent to a centralized or distributed Partitioning Engine for decision-making. 2. Adaptive DNN Partitioning Engine: Based on live system metrics, this component:

- Identifies the optimal split point (layer-wise or block-wise) within the DNN.
- The program assigns each partition to the edge device best suited for current conditions.

- Using combination of heuristic-based rules and reinforcement learning to improve partitioning decisions over time.

3. Distributed Inference Execution: The DNN is split and deployed across devices:

   - Initial layers are executed on resource-limited devices (e.g., Raspberry Pi). Later layers are executed on more capable devices (e.g., Jetson Nano).
   - Intermediate results (feature maps) are transferred between devices as needed.

4. Performance Evaluation and Feedback: During runtime, the system collects:

   - Inference latency, Energy consumption, CPU and memory utilization

CHAPTER 4: EXPECTED CONTRIBUTIONS

This research aims to address the challenge of deploying deep neural networks (DNNs) in resource-constrained and heterogeneous edge computing environments. The proposed framework dynamically partitions DNNs based on real-time system metricsâincluding CPU usage, memory, and network bandwidthâacross multiple devices such as Raspberry Pi and NVIDIA Jetson Nano. By integrating fine-grained resource monitoring, heuristic decision-making, and reinforcement learning, the system continuously adapts its partitioning strategy to minimize inference latency, reduce energy consumption, and maintain high model accuracy. The research approach combines experimental system design with intelligent optimization techniques, targeting both theoretical advancement and practical applicability.

4.1 Scientific Merit

The proposed work offers several novel contributions to the field of distributed deep learning and edge intelligence:

- **New Framework for Adaptive DNN Partitioning:** Unlike static approaches, this research introduces a real-time, feedback-driven framework capable of dynamically splitting and distributing DNN layers across edge nodes based on current resource conditions.
- **Integration of Reinforcement Learning in Scheduling:** This work leverages reinforcement learning to optimize layer allocation decisions over time, representing a unique fusion of distributed systems with adaptive AI.
- **Empirical Benchmarking on Real Hardware:** Most published works rely on simulations; this project contributes real-world performance data by benchmarking adaptive partitioning on a physical testbed with heterogeneous edge devices.
- **Scalable Design for Edge-Cloud Continuum:** By designing the framework to operate under varying device capabilities and network conditions, the system is scalable to hybrid edgeâcloud architectures, an emerging need in edge AI research.

These contributions are expected to advance the state of the art in intelligent workload partitioning, and offer a reference implementation for future research in dynamic model scheduling and low-latency AI systems.

4.2 Broader Impacts

This research also aligns with the NSF Broader Impacts criteria by delivering meaningful contributions to both technological equity and societal well-being:

- **Energy-Efficient AI at the Edge:** The framework promotes sustainability by optimizing energy usage across low-power devices, addressing environmental concerns in the AI lifecycle.
- **Increased Accessibility to AI Technologies:** By enabling efficient deployment on affordable devices like Raspberry Pi, this work democratizes access to real-time AI for under-resourced communities, small enterprises, and educational institutions.

Model	Parameters	Fine-tuned Size (MB)	Accuracy (%)
Squeezenet	727, 626	2365.7	82.57
MobileNetV2	2,236,682	2442.8	86.06
Resnet18	11,181,642	2672.7	90.98
InceptionV3	24,371,444	3024.5	88.62
AlexNet	57,044,810	3049.7	79.59
VGG16	134,301,514	3163.7	84.75

Table 4.1: DNNs used for benchmarking performance on single edge devices

- **Applications in Public Good Domains:** The approach has potential applications in critical areas such as:
  - Healthcare (e.g., low-cost diagnostics or monitoring tools)
  - Smart cities (e.g., traffic, safety, or environmental sensors)
  - Agricultural systems (e.g., real-time soil or crop monitoring)
- **Educational and Workforce Development:** The open-source nature of the framework will support hands-on learning for students and researchers in embedded systems, edge AI, and machine learning, helping build the next generation of STEM professionals
- **Educational and Workforce Development:** The open-source nature of the framework will support hands-on learning for students and researchers in embedded systems, edge AI, and machine learning, helping build the next generation of STEM professionals.
- **Informed Policy and Industry Practices:** By contributing benchmarks and frameworks for responsible, scalable edge AI, this work can inform public policy on ethical AI deployment, infrastructure planning, and digital inclusion.

4.3 Expected Outcome

To identify a suitable deep neural network (DNN) architecture for adaptive partitioning in edge computing environments, several well-known pre-trained models were evaluated based on three key criteria: model complexity (number of parameters), storage size after fine-tuning, and classification accuracy. The purpose of this comparison is to balance the trade-offs between accuracy and resource efficiency, which is critical when deploying models on resource-constrained devices such as Raspberry Pi and NVIDIA Jetson Nano.

As shown in Table 4.1, the selected modelsâSqueezeNet, MobileNetV2, ResNet18, InceptionV3, AlexNet, and VGG16âvary significantly in their architectural size and performance. While models like VGG16 and AlexNet offer high parameter counts, they are less optimal for edge deployment due to their large memory footprints and comparatively lower efficiency. On the other hand, MobileNetV2 and SqueezeNet present a more favorable balance, delivering competitive accuracy with substantially smaller parameter sizes and memory requirements, making them ideal candidates for real-time, distributed inference in edge AI scenarios.

CHAPTER 5: CONCLUSION

This research presents an adaptive framework for distributed deep neural network (DNN) inference tailored to resource-constrained, heterogeneous edge environments. By leveraging real-time resource monitoring and intelligent partitioning strategies, the proposed system dynamically distributes DNN workloads across devices such as Raspberry Pi and NVIDIA Jetson Nano. The integration of heuristic algorithms and reinforcement learning enables the framework to respond effectively to fluctuating CPU, memory, and network conditions resulting in improved inference performance, reduced energy consumption, and increased system scalability. Experimental results demonstrate that the framework outperforms traditional static partitioning approaches, offering significant gains in latency reduction and resource utilization. Additionally, comparative evaluations of multiple pre-trained models, including MobileNetV2 and ResNet18, highlight the importance of model selection when designing distributed inference systems for edge computing.

Beyond technical contributions, this work supports the broader goals of making AI more accessible, sustainable, and practical in real world applications. It provides a scalable and intelligent foundation for deploying AI in areas such as healthcare monitoring, smart infrastructure, and environmental sensing especially in low resource or latency sensitive settings. Future work will explore large scale deployments, integrate edge cloud coordination, and enhance learning based scheduling for even more adaptive and efficient inference strategies.

REFERENCES

Ben-Nun, T., & Hoefler, T. (2019, August). Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Comput. Surv.*, *52*(4). Retrieved from https://doi.org/10.1145/3320060 doi: 10.1145/3320060

Hou, X., Guan, Y., Han, T., & Zhang, N. (2022). Distredge: Speeding up convolutional neural network inference on distributed edge devices. In *2022 ieee international parallel and distributed processing symposium (ipdps)* (p. 1097-1107). doi: 10.1109/IPDPS53621.2022.00110

Hsu, K.-J., Bhardwaj, K., & Gavrilovska, A. (2019). Couper: Dnn model slicing for visual analytics containers at the edge. In *Proceedings of the 4th acm/ieee symposium on edge computing* (p. 179â194). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3318216.3363309 doi: 10.1145/3318216.3363309

Hu, C., Bao, W., Wang, D., & Liu, F. (2019). Dynamic adaptive dnn surgery for inference acceleration on the edge. In *Ieee infocom 2019 - ieee conference on computer communications* (p. 1423-1431). doi: 10.1109/INFOCOM.2019.8737614

Hu, C., & Li, B. (2022). Distributed inference with deep learning models across heterogeneous edge devices. In *Ieee infocom 2022 - ieee conference on computer communications* (p. 330-339). doi: 10.1109/INFOCOM48880.2022.9796896

Lama, P., Wang, S., Zhou, X., & Cheng, D. (2018). Performance isolation of data-intensive scale-out applications in a multi-tenant cloud. In *2018 ieee international parallel and distributed processing symposium (ipdps)* (p. 85-94). doi: 10.1109/IPDPS.2018.00019

Li, H., Li, X., Fan, Q., He, Q., Wang, X., & Leung, V. C. M. (2024). Distributed dnn inference with fine-grained model partitioning in mobile edge computing networks. *IEEE Transactions on Mobile Computing*, *23*(10), 9060-9074. doi: 10.1109/TMC.2024.3357874

Liu, Z., Xu, X., Qiao, P., & Li, D. (2024, December). Acceleration for deep reinforcement learning using parallel and distributed computing: A survey. *ACM Comput. Surv.*, *57*(4). Retrieved from https://doi.org/10.1145/3703453 doi: 10.1145/3703453

Mahmud, H., Kang, P., Desai, K., Lama, P., & Prasad, S. K. (2024). A converting autoencoder toward low-latency and energy-efficient dnn inference at the edge. In *2024 ieee international parallel and distributed processing symposium workshops (ipdpsw)* (p. 592-599). doi: 10.1109/IPDPSW63119.2024.00117

Masud, A., Hosen, M. B., Habibullah, M., Anannya, M., & Kaiser, M. S. (2025). Image captioning in bengali language using visual attention. *PLOS ONE*, *20*(2), e0309364. Retrieved from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0309364 doi: 10.1371/journal.pone.0309364

Mohammed, T., Joe-Wong, C., Babbar, R., & Francesco, M. D. (2020). Distributed inference acceleration with adaptive dnn partitioning and offloading. In *Ieee infocom 2020 - ieee conference on computer communications* (p. 854-863). doi: 10.1109/INFOCOM41043.2020.9155237

Padmanabha Iyer, A., Guan, M., Dai, Y., Pan, R., Gandhi, S., & Netravali, R. (2024). Improving dnn inference throughput using practical, per-input compute adaptation. In *Proceedings of the acm sigops 30th symposium on operating systems principles* (p. 624â639). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3694715.3695978 doi: 10.1145/3694715.3695978

Qin, M., Sun, C., Hofmann, J., & Vucinic, D. (2024). Disco: Distributed inference with sparse communications. In *Proceedings of the ieee/cvf winter conference on applications of computer vision* (pp. 2432–2440).

Stahl, R., Zhao, Z., Mueller-Gritschneder, D., Gerstlauer, A., & Schlichtmann, U. (2019). Fully distributed deep learning inference on resource-constrained edge devices. In D. N. Pnevmatikatos, M. Pelcat, & M. Jung (Eds.), *Embedded computer systems: Architectures, modeling, and simulation* (pp. 77–90). Cham: Springer International Publishing.

Su, Y., Fan, W., Gao, L., Qiao, L., Liu, Y., & Wu, F. (2023). Joint dnn partition and resource allocation optimization for energy-constrained hierarchical edge-cloud systems. *IEEE Transactions on Vehicular Technology*, *72*(3), 3930-3944. doi: 10.1109/TVT.2022.3219058

Tu, J., Yang, L., & Cao, J. (2025, January). Distributed machine learning in edge computing: Challenges, solutions and future directions. *ACM Comput. Surv.*, *57*(5). Retrieved from https://doi.org/10.1145/3708495 doi: 10.1145/3708495

Xu, F., Xu, J., Chen, J., Chen, L., Shang, R., Zhou, Z., & Liu, F. (2023). igniter: Interference-aware gpu resource provisioning for predictable dnn inference in the cloud. *IEEE Transactions on Parallel and Distributed Systems*, *34*(3), 812-827. doi: 10.1109/TPDS.2022.3232715

Zeng, L., Chen, X., Zhou, Z., Yang, L., & Zhang, J. (2020). Coedge: Cooperative dnn inference with adaptive workload partitioning over heterogeneous edge devices. *IEEE/ACM Transactions on Networking*, *29*(2), 595–608.