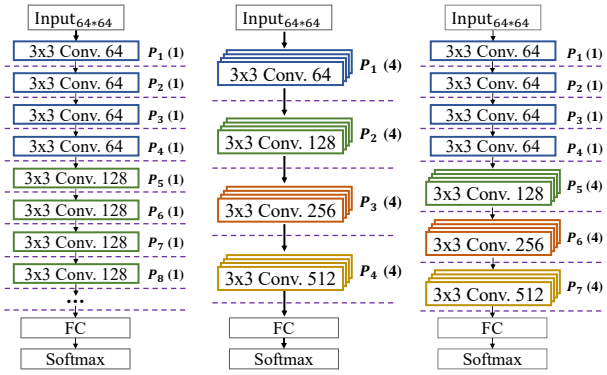
Poster: Energy-Aware Partitioning for Edge AI

Dewant Katare[1], Mengying Zhou[1,2], Yang Chen[2], Marijn Janssen[1], Aaron Yi Ding[1]
[1] Delft University of Technology, [2] Fudan University

***Abstract*—Model partitioning is a promising solution to reduce the high computation load and transmission of high-volume data. Within the scope of Edge AI, the fundamentals of model partitioning involve splitting the model for local computing at the edge and offloading heavy computation tasks to the cloud or server. This approach benefits scenarios with limited computing and battery capacity with low latency requirements, such as connected autonomous vehicles. However, while model partitioning offers advantages in reducing the onboard computation, memory requirements and inference time, it also introduces challenges such as increased energy consumption for partitioned computations and overhead for transferring partitioned data/model. In this work, we explore hybrid model partitioning to optimize computational and communication energy consumption. Our results provide an initial analysis of the trade-off between energy and accuracy, focusing on the energy-aware model partitioning for future Edge AI applications.**

***Index Terms*—Model Partitioning, Energy efficiency, Edge AI**



(a) Meta (b) Ensembled (c) Hybrid

Fig. 1: Architecture comparisons of three approaches

I. Introduction

Model partitioning has been explored as a promising technique to reduce model performance challenges, such as computation cost, latency and energy consumption, on IoT devices. It enables the deployment of AI models to be facilitated with local devices, edge servers, and cloud servers [1]–[3]. Model partitioning divides the AI model into two parts for processing. The low-computational-demand operations of the model are executed locally on an edge or multiple edge devices, while the high-computational-demand operations are offloaded to high-performance edge servers. This partition offloading mechanism improves the feasibility and scalability of running AI on IoT devices from latency and energy. Compared to fully offloading, running a small part of the model locally allows the AI model to transform the raw data input into a communication-acceptable data size [2]. Additionally, by utilizing more powerful edge servers, model inference time is reduced compared to resource-constrained local devices [4]. These two advantages of model partitioning make AI model execution more energy-efficient and faster, particularly when the local devices are powered by limited-capacity batteries.

However, model partitioning introduces potential costs. Partitioning itself is a computational task that can only be performed on-board, resulting in additional energy consumption and computational delays. This cost can make model partitioning a negative optimization when the AI model itself is not very complex, as the cost of partitioning operations is higher than centralized computing. Another cost is the increase in communication overhead. Improper partitioning points can lead to suboptimal communication costs, which in turn affect overall latency and energy consumption. Moreover, the computation and communication costs are closely intertwined in different partitioning decisions. Finding an optimal trade-off based on different circumstances is not a trivial task.

Considering the above-mentioned issues, we explore a hybrid model partitioning approach on CNN models to jointly optimize computational and communication energy. The channel partitioning method on CNN allows it to be divided into multiple offloading components for parallel operations. Instead of using traditional fixed-size partitions, we use hybrid partitioning methods to profile model partitions with the device resources. The hybrid partitioning approach with varying sizes enables more reasonable partitioning and transmission based on the model's profile and computing device capability. Our preliminary results provide insights into the trade-off between energy and latency, offering clues for energy-aware model partitioning in future Edge AI applications.

II. Energy-aware Model Partition Strategies

Our model partitioning approaches are divided into fixed and hybrid strategies.

1. Fixed Model Partition: AI models can be partitioned by parallelizing operations [5]. In Convolutional Neural Networks (CNNs), this involves independently computing tensors for a given input while preserving the architecture's input and output dimensions. The classic channel partitioning method

This work is supported by European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 956090 and grant agreement No. 101021808, and National Natural Science Foundation of China (No. 62072115). Dewant Katare is the Corresponding Author (email: d.katare@tudelft.nl).

2024 IEEE/ACM Symposium on Edge Computing (SEC) | 979-8-3503-7828-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SEC62691.2024.00067

2837-4827/24/$31.00 ©2024 IEEE
DOI 10.1109/SEC62691.2024.00067

for CNN classification [6] conveys the principles of parallel layer execution. In convolutional layers, each filter generates a feature map that can be divided into channels for partitioning. Output feature maps are split along the channel dimension, allowing each edge device to compute a subset of these maps. Filters are distributed across devices, containing the same model dimensions to prevent adversarial effects. Similarly, fully connected layers are parallelized using spatial partitioning (height and width dimensions), where each device independently computes a subset of output feature maps. Unlike convolutional layers, fully connected layers require transferring only a subset of inputs, reducing overall communication costs. Building on channel partitioning principles, the following section introduces two specialized approaches.

Meta Partition: This method employs channel partitioning to independently separate each layer, as illustrated in Fig. 1(a). The model is divided by computing each layer's tensors as subgroups, then merging their outputs to form the tensors for subsequent layers. Independent layer processing necessitates combining output tensors to aggregate weights and generate final outputs. While this approach incurs high communication costs in dynamic wireless networks, in our fully connected vehicular ecosystem, the benefits of parallel layer execution offset the communication overhead from vehicles and vision sensors transmitting large volumes of data.

Ensembled Partition: The Meta approach incurs high communication costs due to frequent data transfers between layers. To address this, the Ensembled approach combines layers with similar dimensions, reducing communication overhead and optimizing end-to-end energy consumption. This method, also utilized for hardware acceleration in machine learning to lower memory usage, merges operations from multiple convolutional layers into a single block (see Fig. 1(b)) instead of handling each layer individually as in the Meta approach. Partitioning maintains consistent input and output dimensions, allowing the number of fused layers to vary based on the available computational resources of participating devices. While increasing the number of fused blocks can raise communication costs, this approach is ideal for edge-cloud deployments of AI models with repetitive layer structures, such as ResNet.

2. Hybrid Model Partition: In a vehicle-edge ecosystem with heterogeneous devices, model partitioning must be dynamic and adaptable. The host device monitors the computational resources of participating devices and adjusts partitioning based on each device's task requirements and available resources, as shown in Fig 1(c). The partitioning algorithm optimizes partition scheme based on energy-related parameters, including available CPU/GPU computing resources, estimated memory, and latency. While this approach may increase the computational load and power usage on the host device, a balanced tradeoff and further optimizations can achieve overall end-to-end energy savings across all devices.

Takeaway: The three approaches provide different partitioning size and combination strategies, which further result in diverse effects on the model's accuracy and power consumption.

TABLE I: Model Performance Metrics

Model	Approach	Accuracy (%)	Power (Wh)
MobileNet (3.7 MB)	Meta	70.90	1.93
	Ensembled	67.40	1.69
	Hybrid	65.10	1.81
SqueezeNet (11.0 MB)	Meta	78.59	1.88
	Ensembled	79.38	2.58
	Hybrid	74.29	2.34
ResNet (45.9 MB)	Meta	88.43	5.71
	Ensembled	89.61	5.49
	Hybrid	84.31	5.22

III. Results

We evaluate the above-mentioned partitioning approaches using model accuracy and power consumption on NVIDIA Jetson Nano. **Models:** Pre-trained MobileNet, ResNet, and SqueezeNet are used for partitioning and deployment due to their computationally intensive convolutional layers. **Accuracy:** Under the nuScenes object classification dataset, partitioned models shows reduced accuracy compared with benchmarks, as shown in Table I. **Power Consumption:** Power is measured with tegrastats interface and an HV power monitor. Smaller models like MobileNet and SqueezeNet achieve energy savings, though with decreased accuracy. This tradeoff reflects the balance between energy efficiency and model performance.

IV. Conclusion and Future Work

Model partitioning demonstrates promise for deploying complex AI models on resource-constrained edge devices. To fully realize benefits in energy savings, latency reduction, and lower communication loads, future work will focus on metrics directly related to these performance gains. We plan to explore combining model partitioning with approximation strategies, particularly bit-wise operations, for advanced models such as vision transformers (ViTs) and vision-language models. Optimizing inference operations and edge deployment of these complex models with balanced approximation techniques could enhance energy efficiency and provide sustainable Edge AI applications.

References

[1] F. McLean, L. Xue, C. X. Lu, and M. Marina, "Towards edge-assisted real-time 3d segmentation of large scale lidar point clouds," in *Proceedings of the EMDL 2022*, 2022, pp. 1–6.

[2] M. Wu, F. R. Yu, and P. X. Liu, "Intelligence networking for autonomous driving in beyond 5g networks with multi-access edge computing," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 5853–5866, 2022.

[3] Y. Wu, C. Cai, X. Bi, J. Xia, C. Gao, Y. Tang, and S. Lai, "Intelligent resource allocation scheme for cloud-edge-end framework aided multi-source data stream," *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 1, p. 56, 2023.

[4] Z. Yang, M. Chen, Z. Zhang, and C. Huang, "Energy Efficient Semantic Communication over Wireless Networks with Rate Splitting," *arXiv preprint arXiv:2301.01987*, 2023.

[5] H. Wu, W. J. Knottenbelt, and K. Wolter, "An efficient application partitioning algorithm in mobile environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 7, pp. 1464–1480, 2019.

[6] W. Zhang, N. Wang, L. Li, and T. Wei, "Joint compressing and partitioning of CNNs for fast edge-cloud collaborative intelligence for IoT," *Journal of Systems Architecture*, vol. 125, p. 102461, 2022.