

Limited Resources Usage of Deep-Learning-Based IoT Devices in Healthcare

1st Xian Gao

*TSYS School of Computer Science
Columbus State University
Columbus, GA, U.S
gao_xian@students.columbusstate.edu*

2nd Peixiong He

*Department of Computer Science and Software Engineering
Auburn University
Auburn, AL, U.S
pxh0029@auburn.edu*

3rd Yi Zhou*

*TSYS School of Computer Science
Columbus State University
Columbus, GA, U.S
zhou_yi@columbusstate.edu*

4th Xiao Qin

*Department of Computer Science and Software Engineering
Auburn University
Auburn, AL, U.S
xqin@auburn.edu*

Abstract—Deep learning models are widely used in healthcare. Given the large amount of data and the need for real-time processing, running these models on IoT devices is a viable solution. This study examines the impact of deep learning models on the limited resources of IoT devices. We selected three models introduced at the 2023 CVPR conference and executed them on a Raspberry Pi with different dataset sizes to measure CPU utilization, memory utilization, and energy consumption. The results show that for the pruned Yolov5 model, CPU utilization has the strongest positive correlation with energy consumption (5.43×10^{-4} Joules), while memory utilization shows the most significant negative correlation (-4.27×10^{-4} Joules). In the FasterNet-T0 model, dataset size has the strongest positive correlation with energy consumption (2.75×10^{-5} Joules). For the Lite-Mono model, dataset size shows the most significant positive correlation (1.84×10^{-4} Joules), while memory utilization has the strongest negative correlation with energy consumption (-2.52×10^{-4} Joules). These results highlight the key factors correlated with energy consumption across different models, providing valuable insights for deploying deep learning models on IoT devices.

Index Terms—Deep Learning, IoT, Healthcare, Hardware Utilization, Energy Consumption

I. INTRODUCTION

THE rapid expansion of the Internet of Things (IoT) in healthcare has generated an immense volume of data that demands real-time processing to extract meaningful insights. Traditional centralized server methods often lead to significant latency issues, making them unsuitable for real-time analysis. This inefficiency is compounded in underserved and remote areas, where unreliable or nonexistent internet connections further hinder the functionality of deep learning models reliant on cloud infrastructure, which are crucial for effective healthcare delivery. Decentralizing data processing by running deep learning models on IoT devices emerges as a promising solution to this challenge. This approach relocates deep learning models, computing power, and storage closer

to the data source, thereby reducing latency and enabling real-time, efficient processing even in connectivity-limited environments, thus paving the way for more adaptive and responsive deep-learning-driven healthcare solutions [1].

However, in practice, IoT devices still have some disadvantages, including limited resources. The CPU performance of IoT devices is lower than that of powerful servers in data centers, limiting the complexity and scale of computing tasks they can perform. IoT devices also have limited storage capacity relative to centralized cloud data centers, which means they cannot store large amounts of data for extended periods. Many IoT devices are battery-powered, which limits the amount of computing they can perform because intensive computing tasks can quickly drain the battery. Therefore, managing the relationship between deep learning models and the limited resources of IoT devices is a serious challenge [2] [3].

II. RELATED WORK

A. IoT Devices in Healthcare

The rapid growth of smart sensors, actuators, networking technology, and low-power chips has greatly enhanced IoT devices, particularly in healthcare [4]. This advancement has improved patient care by enabling remote monitoring, personalized treatment, and efficient healthcare delivery [5]. Common healthcare IoT devices include fall detectors, calorie estimators, pulse monitors, pacemakers, defibrillators, and neuro-stimulators [6]. These devices are typically categorized as wearable or implantable. Wearable devices are worn on the body, while Implantable Medical Devices (IMDs) are inserted into the body. Both types integrate microprocessors, sensors, power sources, and wireless modules to collect, analyze, and transmit physiological data for further processing or direct intervention [7].

IoT devices use both hardware and software components. Popular hardware platforms include Raspberry Pi, Arduino,

* Y. Zhou is the corresponding author

and ESP8266. Raspberry Pi is a versatile single-board computer with a full OS, suitable for complex tasks. Arduino is an easy-to-use prototyping platform for basic IoT and sensor projects. ESP8266 is a low-cost Wi-Fi microcontroller used in wireless IoT devices for small networking projects [8]. This paper focuses on experiments using the Raspberry Pi.

B. IoT-Device-Based Deep Learning Models in Healthcare

Healthcare IoT devices generate vast amounts of data daily, necessitating rapid data analysis on smart mobile devices. Deep learning models are effective for processing this data, identifying underlying features, and improving IoT application performance on smart devices [9]. These models in healthcare fall into four categories: medical diagnosis, home-based and personal healthcare, disease prediction, and human behavior recognition [10].

Medical diagnosis models like HealthFog use ensemble deep learning on edge devices for heart disease analysis, delivering healthcare as a fog service [11]. Home-based models, such as MASK R-CNN in the iHome DentalHealth-IoT System, use Faster R-CNN to predict dental disease with high accuracy [12]. Disease prediction models, like DFCNet, employ wearable sensors and deep learning to detect pulmonary cancer from CT images [13]. Human behavior recognition models, such as deep residual bidirectional LSTM networks, enhance recognition rates by integrating bidirectional cells and stacked residual connections [14].

While these models are widely used, they are not the most recent. This paper will focus on experiments with three new deep learning models introduced at the 2023 CVPR conference for healthcare applications.

C. Current Exploration of Energy Model on Raspberry Pi

Studying the energy consumption model of IoT devices, especially Raspberry Pi, when running deep learning models is crucial. These models can help assess the energy efficiency of devices during the execution of complex tasks, thereby laying the groundwork for designing more energy-efficient systems [15].

For instance, the Energy Measurement Model (EMM) serves as an energy estimation tool designed for the Raspberry Pi 3 Model B+. This software-based model estimates a device's local energy consumption by considering CPU utilization [16]. Another model incorporates CPU and Ethernet usage as variables, effectively translating resource usage data into accurate energy consumption information for single-board computers (SBCs) such as the Raspberry Pi 2 Model B [17]. Yet another model was developed by recording the system utilization and power consumption of devices under test, followed by regression analysis on the collected data [18].

While these models consider several factors influencing energy consumption, not all necessary variables are accounted for. In this paper, we will design and conduct experiments considering CPU utilization, memory utilization, dataset size, and AI algorithm type.

III. METHOD

A. Testing Device

The Raspberry Pi 3 Model B+ (RPI 3B+) is powered by a 64-bit Broadcom BCM2837B0 quad-core Cortex-A53 SoC running at a clock speed of 1.4 GHz, supported by 1 GB of LPDDR2-SDRAM. It features a Gigabit Ethernet interface via USB 2.0, achieving a maximum data throughput of 300 Mbps. The board also supports dual-band Wi-Fi (2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac), Bluetooth 4.2, and BLE. It is equipped with four USB 2.0 ports, forty GPIO pins, a full-size HDMI connector, a MIPI DSI display port, a MIPI CSI camera port, and a microSD card slot. Power can be supplied through the GPIO connector or via Power over Ethernet (PoE) using an optional PoE HAT [19].

B. Testing Models

In this experiment, we selected three papers from the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), introducing cutting-edge deep learning models designed to optimize resource utilization and consumption in IoT devices for healthcare.

The first paper introduces the Dependency Graph (Dep-Graph) model, enabling structural pruning across various architectures such as CNNs, RNNs, GNNs, and Transformers. This method ensures systematic pruning across multiple layers, aiming for consistent removal of non-essential parameters. This approach mitigates structural issues and prevents significant performance degradation post-pruning [20]. Subsequently, we applied pruning techniques to the YOLOv5 model, selecting the pruned YOLOv5 as our experimental model.

The second paper introduces a model named FasterNet, which utilizes a novel technique called partial convolution (PConv) to enhance the extraction of spatial features. This approach reduces redundant computations and memory access, thereby enabling FasterNet to achieve significantly faster processing speeds across different devices without compromising accuracy in vision tasks [21]. Among the various versions of FasterNet, we selected FasterNet-T0, the smallest model in the series, as our experimental model.

The third paper introduces a model named Lite-Mono, which effectively integrates CNNs and Transformers. It features a Consecutive Dilated Convolutions (CDC) module for extracting rich multi-scale local features and a Local-Global Features Interaction (LGFI) module that utilizes the self-attention mechanism to encode extensive global information into the features. This integration enhances accuracy while reducing the number of trainable parameters [22]. Since Lite-Mono has several versions, we opted to use the Lite-Mono version as our experimental model.

C. Testing Dataset

The testing dataset is sourced from Kaggle, specifically the 'Brain Tumor Dataset' contributed by Preet Viradiya. This dataset contains scanned brain images of patients diagnosed with brain tumors. It includes separate files for training and testing data, with distinct features and labels. The images are

provided in JPG format [23]. Originally sized at 128.4 MB, for this experiment, we selected a subset of the dataset as our testing dataset. To illustrate hardware utilization and energy consumption trends across varying dataset sizes, we utilized a 100 MB subset from the dataset. This subset was divided into ten distinct sub-datasets, each increasing by 10 MB compared to the previous one.

D. Experiment Design

This research will initially assess the impact of deep learning models on the limited resources of IoT devices. To achieve this, we will measure CPU utilization, Memory utilization, and energy consumption of the Raspberry Pi running the aforementioned models under varying dataset sizes ranging from 10 MB to 100 MB. Subsequently, this study aims to develop energy consumption models where CPU utilization, Memory utilization, dataset size, and the type of deep learning model serve as determinant factors. Regression methods will be employed to determine the weights of these factors in the models. Specifically, (a) a model will be proposed where total energy consumption is a function of CPU utilization, Memory utilization, and dataset size; (b) another model will focus on CPU utilization as a function of dataset size; and (c) an additional model will address Memory utilization as a function of dataset size.

E. Energy Consumption Measurement

Regarding the measurement of energy consumption, we used an electricity usage monitor plug. While executing the deep learning models, we connected the power supply of the test device to the plug. We recorded the time taken by the models to run on different datasets and recorded the real-time power displayed by plug every ten seconds. After the execution was complete, we calculated the average power over the entire runtime. We then used the energy consumption formula $E = P \cdot (t/1000)$, where E = energy measured in Joules or kilowatt-hours (kWh), P = power units in watts, and t = time over which the power or energy was consumed, to calculate the energy consumption for each run. This measurement experiment was repeated five times, and the average energy consumption was calculated.

F. A Comparison with Other Energy Consumption Prediction Method

In practice, besides the methods mentioned in this paper for predicting the power consumption of IoT devices running deep learning models, there is also the approach of using a machine learning or deep learning model specifically designed to predict power consumption. Compared to the methods discussed in this paper, this approach has broader applicability, and therefore, offers better efficiency, flexibility, and cost-effectiveness in the long term. However, the methods described in this paper provide higher accuracy for predicting the power consumption of IoT devices when running specific deep learning models, require less data quantity and quality, and have lower complexity for the power consumption prediction model.

IV. EXPERIMENT

A. The Pruned YOLOv5 Model

After experimenting according to the aforementioned design, we obtained the following measurement results for the pruned YOLOv5 model.

The Pruned YOLOv5 Model			
Dataset Size(MiB)	CPU Utilization(%)	Memory Utilization(%)	Energy Consumption(kWh)
10	94.67	51.26	1.872e-4
20	94.23	50.47	3.796e-4
30	94.35	52.38	1.248e-3
40	94.41	52.14	2.288e-3
50	94.14	53.52	3.172e-3
60	94.72	53.83	3.848e-3
70	94.28	53.65	4.732e-3
80	94.46	54.21	5.72e-3
90	94.39	55.19	5.252e-3
100	94.54	57.73	5.304e-3

Fig. 1. The measurement result of the pruned YOLOv5 model

1) *Hardware Utilization Model*: The data indicates that when running the pruned YOLOv5 deep learning model on a Raspberry Pi, CPU utilization remains relatively stable at around 94% as the dataset size increases, showing minimal variation. However, memory utilization gradually increases from 51.26% at 10 MiB to 57.73% at 100 MiB. This suggests that while the pruned YOLOv5 model's demand for CPU resources remains relatively consistent, its demand for memory resources significantly rises with larger dataset sizes.

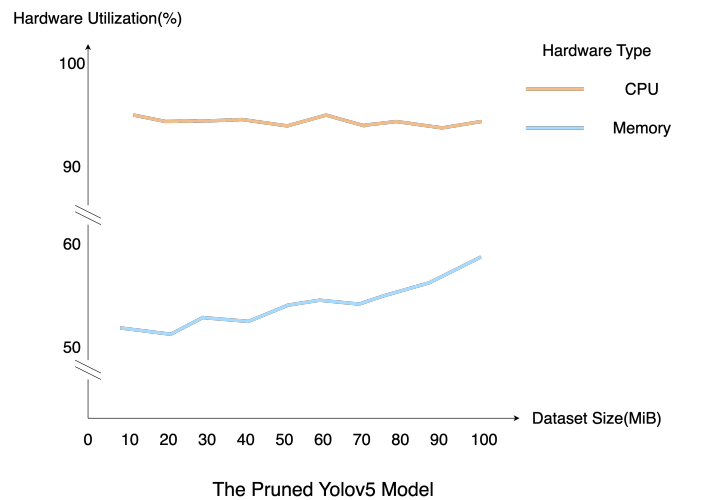


Fig. 2. The hardware utilization of the pruned YOLOv5 model

2) *Energy Consumption Model*: Based on the data in Figure 1, we can plot a line chart to illustrate the trend of power consumption for the pruned Yolov5 model across different dataset sizes. The data indicates that when executing the pruned Yolov5 deep learning model on a Raspberry Pi, power consumption generally increases with dataset size, albeit not in a strictly linear fashion.

Specifically, from 10 MiB to 50 MiB, power consumption steadily rises, showing an almost linear growth trend. As the dataset size further increases from 50 MiB to 80 MiB, power consumption continues to increase, but at a slower rate. Subsequently, from 80 MiB to 90 MiB, there is a slight decrease in power consumption. Finally, as the dataset size increases from 90 MiB to 100 MiB, power consumption shows a slight upward trend.

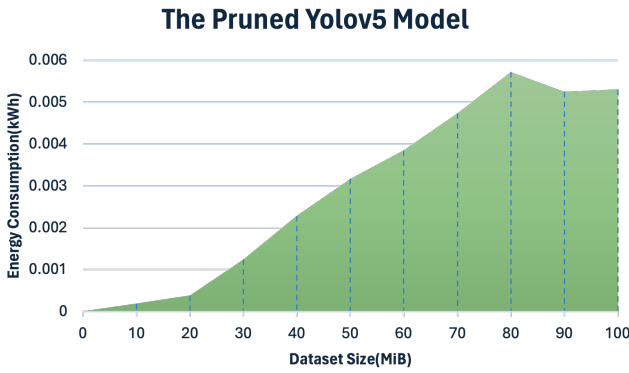


Fig. 3. The energy consumption of the pruned Yolov5 model

We propose an energy model where CPU utilization, Memory utilization, and dataset size act as determinant factors, illustrating their influence on energy consumption.

$$\begin{aligned} \text{Energy Consumption} = & -0.0304 \\ & + (5.43 \times 10^{-4} \times \text{CPU utilization}) \\ & + (9.4 \times 10^{-5} \times \text{Dataset Size}) \\ & - (4.27 \times 10^{-4} \times \text{Memory utilization}) \end{aligned}$$

B. The FasterNet-T0 Model

After experimenting according to the above design, we obtained the following measurement results for the FasterNet-T0 model.

1) *Hardware Utilization Model*: The data above indicates that when executing the FasterNet-T0 deep learning model on a Raspberry Pi, CPU and memory utilization remain relatively stable as the dataset size increases. Specifically, CPU utilization stabilizes around 29%, while memory utilization remains consistently around 34%. This suggests that the FasterNet-T0 model maintains a consistent demand for computational resources, with minimal impact from variations in dataset size on its CPU and memory usage rates.

2) *Energy Consumption Model*: Based on the data in Figure 4, we can generate a line chart to illustrate the trend of power

The FasterNet-T0 Model			
Dataset Size(MiB)	CPU Utilization(%)	Memory Utilization(%)	Energy Consumption(kWh)
10	29.36	34.73	2.628e-4
20	29.14	34.62	5.4e-4
30	29.22	34.45	7.92e-4
40	29.51	34.29	1.08e-3
50	29.47	34.28	1.368e-3
60	29.25	34.35	1.62e-3
70	29.39	34.30	1.908e-3
80	29.28	34.36	2.16e-3
90	29.41	34.30	2.484e-3
100	29.63	34.43	2.736e-3

Fig. 4. The measurement result of the FasterNet-T0 model

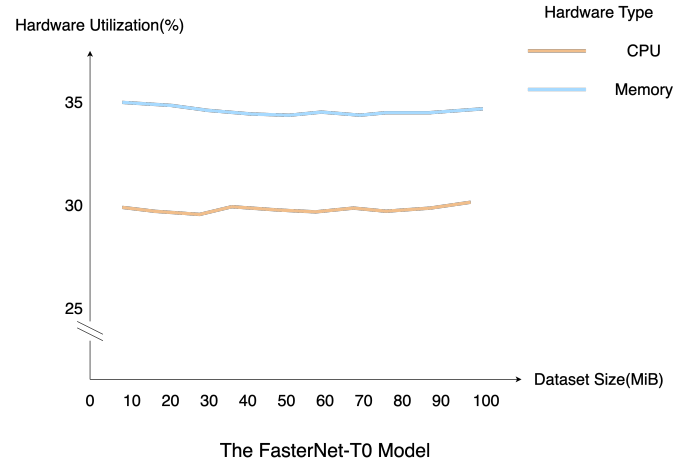


Fig. 5. The hardware utilization of the FasterNet-T0 model

consumption for the FasterNet-T0 model with different dataset sizes. The data indicate that when executing the FasterNet-T0 deep learning model on a Raspberry Pi, power consumption shows a stable linear growth trend as the dataset size increases. This suggests that power consumption is directly proportional to the dataset size, reflecting the model's consistent demand for computational resources when processing datasets of varying sizes.

We could propose an energy model where CPU utilization, memory utilization, and dataset size serve as determinant factors for these models, and then demonstrate how these

The FasterNet-T0 Model

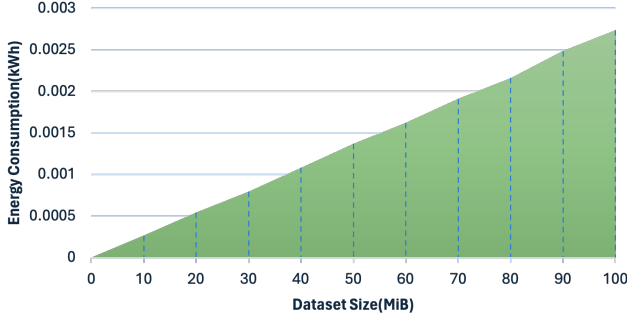


Fig. 6. The energy consumption of the FasterNet-T0 model

factors influence energy consumption.

$$\begin{aligned} \text{Energy Consumption} = & -0.0018 \\ & + (4.16 \times 10^{-5} \times \text{CPU utilization}) \\ & + (2.75 \times 10^{-5} \times \text{Dataset Size}) \\ & + (1.72 \times 10^{-5} \times \text{Memory utilization}) \end{aligned}$$

C. The Lite-Mono Model

After experimenting according to the above design, we obtained the following measurement results for the Lite-Mono model.

The Lite-Mono Model			
Dataset Size(MiB)	CPU Utilization(%)	Memory Utilization(%)	Energy Consumption(kWh)
10	88.75	33.31	2.808e-4
20	87.62	33.48	5.72e-4
30	88.46	33.63	2.704e-3
40	88.83	33.42	5.72e-3
50	87.81	32.35	7.28e-3
60	86.69	34.23	9.36e-3
70	87.37	32.87	1.144e-2
80	87.74	34.16	1.248e-2
90	86.95	33.48	1.404e-2
100	87.56	33.64	1.56e-2

Fig. 7. The measurement result of the Lite-Mono model

1) *Hardware Utilization Model*: The data from running the Lite-Mono Model on a Raspberry Pi indicate that CPU and memory utilization exhibit slight fluctuations across different dataset sizes. CPU utilization ranges from 86.69% to 88.83%, demonstrating relative stability with minimal variation. Similarly, memory utilization fluctuates between 32.35% and

34.23%, showing modest changes. Overall, while the Lite-Mono model adjusts its resource usage slightly in response to dataset size, its demands on CPU and memory remain relatively consistent and do not significantly escalate with larger datasets.



Fig. 8. The hardware utilization of the Lite-Mono model

2) *Energy Consumption Model*: Based on the data in Figure 7, we can generate a line chart to illustrate the trend of power consumption for the Lite-Mono model with varying dataset sizes. The data show that when executing the Lite-Mono deep learning model on a Raspberry Pi, power consumption generally follows a rapid growth trend as the dataset size increases, albeit non-linearly. Particularly notable is the sharp increase in power consumption between dataset sizes of 20MiB and 30MiB, indicating significant non-linear growth. This suggests that larger datasets markedly elevate the demand for computational resources, resulting in higher power consumption. Overall, dataset size exerts a substantial, non-linear impact on the power consumption of the Lite-Mono model. We propose an energy model where CPU utilization,

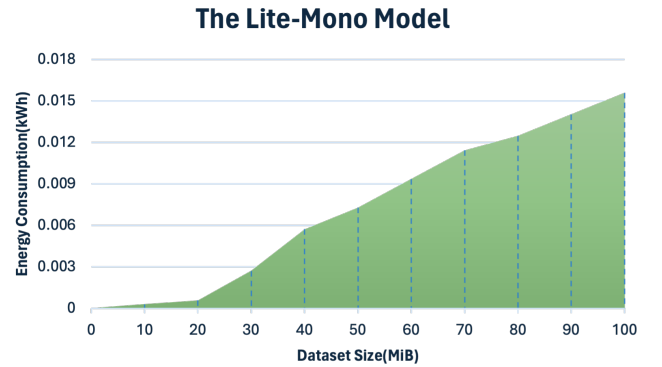


Fig. 9. The energy consumption of the Lite-Mono model

memory utilization, and dataset size act as determinant factors

for these models, demonstrating how these factors influence energy consumption.

$$\begin{aligned} \text{Energy Consumption} &= 0.0023 \\ &+ (4.59 \times 10^{-5} \times \text{CPU utilization}) \\ &+ (1.84 \times 10^{-4} \times \text{Dataset Size}) \\ &- (2.52 \times 10^{-4} \times \text{Memory utilization}) \end{aligned}$$

V. CONCLUSION AND FUTURE SCOPE

This study investigates the performance of three deep learning models—pruned Yolov5, FasterNet-T0, and Lite-Mono—on a Raspberry Pi with different dataset sizes. We focus on how these models affect the limited resources of IoT devices, including CPU utilization, memory utilization, and energy consumption.

Specifically, the Lite-Mono model's CPU utilization remains stable between 87% and 89%, while its memory utilization fluctuates slightly between 32% and 34%. The pruned Yolov5 model demonstrates stable CPU utilization close to 94% across different dataset sizes, with memory utilization gradually increasing as the dataset size grows. In contrast, the FasterNet-T0 model maintains around 29% CPU utilization and 34% memory utilization across various dataset sizes. The experimental results clearly indicate that the choice of deep learning model significantly impacts the resource utilization of IoT devices.

The Lite-Mono model exhibits the highest energy consumption, with power usage increasing nearly linearly with dataset size. The pruned Yolov5 model also shows a nearly linear increase in energy consumption from 10 to 80 MiB, but there is a decrease in power usage between 80 and 100 MiB. Finally, the FasterNet-T0 model's energy consumption also increases nearly linearly with dataset size.

In terms of future scope, we plan to select a lightweight deep learning model widely used in healthcare practice. First, we will measure the performance of this model on IoT devices with different dataset sizes, including CPU utilization, memory utilization, and energy consumption. Next, we will test methods from existing research aimed at improving the performance of lightweight deep learning models on IoT devices. Finally, through experimentation, we aim to discover new methods to enhance performance from a novel perspective.

REFERENCES

- [1] Chellammal Surianarayanan, John Jeyasekaran Lawrence, Pethuru Raj Chelliah, Edmond Prakash, and Chaminda Hewage. A survey on optimization techniques for edge artificial intelligence (ai). *Sensors*, 23(3):1279, 2023.
- [2] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y. Zomaya. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020.
- [3] Alberto Marchisio, Muhammad Abdullah Hanif, Faiq Khalid, George Plastiras, Christos Kyrkou, Theodoris Theodorides, and Muhammad Shafique. Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges. In *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 553–559, 2019.
- [4] Partha Pratim Ray, Dinesh Dash, and Debashis De. Edge computing for internet of things: A survey, e-healthcare case study and future direction. *Journal of Network and Computer Applications*, 140:1–22, 2019.
- [5] Chunyan Li, Jiayi Wang, Shuihua Wang, and Yudong Zhang. A review of iot applications in healthcare. *Neurocomputing*, page 127017, 2023.
- [6] Trayush Trayush, Ruchika Bathla, Sonia Saini, and Vinod Kumar Shukla. Iot in healthcare: Challenges, benefits, applications, and opportunities. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 107–111, 2021.
- [7] Farha Nausheen and Sayyada Hajera Begum. Healthcare iot: Benefits, vulnerabilities and solutions. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 517–522, 2018.
- [8] Md. Milon Islam, Sheikh Nooruddin, Fakhri Karray, and Ghulam Muhammad. Internet of things: Device capabilities, architectures, protocols, and smart applications in healthcare domain. *IEEE Internet of Things Journal*, 10(4):3611–3641, 2023.
- [9] Xiaoqiang Ma, Tai Yao, Menglan Hu, Yan Dong, Wei Liu, Fangxin Wang, and Jiangchuan Liu. A survey on deep learning empowered iot applications. *IEEE Access*, 7:181721–181732, 2019.
- [10] Hamidreza Bolhasani, Maryam Mohseni, and Amir Masoud Rahmani. Deep learning applications for iot in health care: A systematic review. *Informatics in Medicine Unlocked*, 23:100550, 2021.
- [11] Shreshth Tuli, Nipam Basumatary, Sukhpal Singh Gill, Mohsen Kahani, Rajesh Chand Arya, Gurpreet Singh Wander, and Rajkumar Buyya. Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments. *Future Generation Computer Systems*, 104:187–200, 2020.
- [12] Lizheng Liu, Jiawei Xu, Yuxiang Huan, Zhuo Zou, Shih-Ching Yeh, and Li-Rong Zheng. A smart dental health-iot platform based on intelligent hardware, deep learning, and mobile terminal. *IEEE journal of biomedical and health informatics*, 24(3):898–906, 2019.
- [13] Anum Masood, Bin Sheng, Ping Li, Xuhong Hou, Xiaor Wei, Jing Qin, and Dagan Feng. Computer-assisted decision support system in pulmonary cancer detection and stage classification on ct images. *Journal of biomedical informatics*, 79:117–128, 2018.
- [14] Yu Zhao, Renrong Yang, Guillaume Chevalier, Ximeng Xu, and Zhenxing Zhang. Deep residual bidir-ilstm for human activity recognition using wearable sensors. *Mathematical Problems in Engineering*, 2018:1–13, 2018.
- [15] Sang Hyeon Lee, Tacklim Lee, Seunghwan Kim, and Sehyun Park. Energy consumption prediction system based on deep learning with edge computing. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pages 473–477, 2019.
- [16] Kamar Kesrouani, Houssam Kanso, and Adel Noureddine. A preliminary study of the energy impact of software in raspberry pi devices. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 231–234, 2020.
- [17] Luca Ardito and Marco Torchiano. Creating and evaluating a software power model for linux single board computers. In *Proceedings of the 6th International Workshop on Green and Sustainable Software*, pages 1–8, 2018.
- [18] Fabian Kaup, Stefan Hacker, Eike Mentzendorff, Christian Meurisch, and David Hausheer. The progress of the energy-efficiency of single-board computers. *Tech. Rep. NetSys-TR-2018-01*, 2018.
- [19] "raspberrypi 3 model b+." <https://www.raspberrypi.com/products/raspberrypi-3-model-b-plus/>.
- [20] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16091–16101, 2023.
- [21] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S.-H. Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12021–12031, 2023.
- [22] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, 2023.
- [23] "brain tumor dataset." <https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset/>.