International Journal of Advanced Smart Convergence Vol.12 No.4 164- 170 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.4.164

IJASC 23-4-20

Comparison of Traditional Workloads and Deep Learning Workloads in Memory Read and Write Operations

Jeongha Lee[1], Hyokyung Bahn[2]

[1]*Graduate Student, Department of Computer Engineering, Ewha University, Korea*
lee-jeongha@ewhain.net
[2]*Professor, Department of Computer Engineering, Ewha University, Korea*
bahn@ewha.ac.kr

Abstract

With the recent advances in AI (artificial intelligence) and HPC (high-performance computing) technologies, deep learning is proliferated in various domains of the 4th industrial revolution. As the workload volume of deep learning increasingly grows, analyzing the memory reference characteristics becomes important. In this article, we analyze the memory reference traces of deep learning workloads in comparison with traditional workloads specially focusing on read and write operations. Based on our analysis, we observe some unique characteristics of deep learning memory references that are quite different from traditional workloads. First, when comparing instruction and data references, instruction reference accounts for a little portion in deep learning workloads. Second, when comparing read and write, write reference accounts for a majority of memory references, which is also different from traditional workloads. Third, although write references are dominant, it exhibits low reference skewness compared to traditional workloads. Specifically, the skew factor of write references is small compared to traditional workloads. We expect that the analysis performed in this article will be helpful in efficiently designing memory management systems for deep learning workloads.

Keywords: *Deep learning, Memory Reference, Memory Operation, Read, Write, Skewness.*

1. Introduction

As artificial intelligence (AI) technology advances dramatically, deep learning is increasingly being adopted in modern intelligent systems. Accordingly, deep learning has become an indispensable part of our daily lives [1, 2, 3]. Various kinds of living services internally perform image processing and/or text analysis with deep learning frameworks such as TensorFlow [4, 5]. Mobile services also utilize deep learning techniques for smart services [6].

As the data size of deep learning grows, analyzing the memory reference characteristics of deep learning workloads becomes important. Although the memory size of the system continues to extend, it is not easy to

Manuscript Received: october. 18, 2023 / Revised: october. 23, 2023 / Accepted: october. 28, 2023
Corresponding Author: bahn@ewha.ac.kr
Tel: +82-2-3277-2368, Fax: +82-2-3277-2306
Author's affiliation: Department of Computer Engineering, Ewha University, Professor, Korea

Copyright© 2023 by The Institute of Internet, Broadcasting and Communication. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0)

accommodate the entire memory footprint of ever growing deep learning dataset due to the scalability limit of DRAM medium and its power consumption [7, 8]. Specifically, the manufacturing of DRAM cannot scale down the density below 5 nanometers, and the power consumption of DRAM increases largely in accordance with the memory capacity used. As DRAM is volatile memory, consistent refresh of each cell is necessary for maintaining data even when the data is not accessed [7, 8]. Note that this refresh operation is responsible for a large portion of power consumption in memory systems [9].

For this reason, analyzing memory references is important to design an efficient memory management system for deep learning workloads. In this article, we analyze the memory reference characteristics of deep learning workloads in comparison with traditional workloads. In particular, we characterize read and write operations separately and observe some important characteristics of deep learning memory references, which are very different from traditional workloads.

First, when comparing instruction and data references, instruction references account for a little portion in deep learning workloads, which is quite different from traditional workloads. Specifically, instruction reference accounts for 1-3.3% in deep learning workloads while 15.3-37.5% in traditional workloads. Second, when comparing read and write references, write reference accounts for dominant portion of 63.7-80.4%, which is also different from traditional workloads. Third, although write reference accounts for a majority of memory references, it exhibits low reference skewness compared to traditional workloads. Specifically, the skew factor of write references is very small compared to traditional workloads.

Based on the analysis conducted in this article, we can summarize the result such that efficient memory management for deep learning workloads is more difficult than traditional workloads because of specific memory reference characteristics. We hope that the result of this article will be helpful in managing future memory management systems for deep learning workloads by considering the characteristics we analyzed.

2. Ratio of Instruction and Data Read/Write

Pytorch and TensorFlow are popular deep learning frameworks for creating learning models with LSTM and Convolution layers. In this article, we collect memory reference traces while running TensorFlow with LSTM and Convolution layers. For extracting memory reference traces while running deep learning workloads, we utilize the Callgrind module of Valgrind tool set [10]. We collect memory reference traces of four deep learning workloads: IMDB, Spam detection, Fashion MNIST, and MNIST. IMDB classifies positive or negative ratings from 50,000 movie reviews by making use of 1D Convolution layers. Spam detection decides whether an email is spam or not according to the contents of the email by making use of LSTM layers. FashionMNIST classifies 10 types of clothing images including bags, shoes, and pants by making use of 2D Convolution layers. MNIST identifies text images of numbers 0 to 9 by making use of LSTM layers. For comparison purpose, we make use of memory reference traces of traditional workloads consisting of game, office, PDF, and photo. Game is a traditional card game app called Freecell. Office is a document editing software called Gedit. PDF is a document viewer application called KGhostview. Photo is an image browser software called Geeqie.

Figs. 1 and 2 show the distributions of memory references for the deep learning workloads and traditional workloads, respectively. Memory references can be classified into read instruction, read data, and write data. As shown in Fig. 1, in deep learning workloads, read instruction accounts for a very small portion of memory references. Specifically, read instruction is responsible for 1.0-3.3%. Note that this is not the case for traditional workloads in Fig. 2 where 15.3-37.5% are read instruction. This implies that the size of data to be referenced

in deep learning workloads is larger than traditional workloads for running the same number of instructions.

Also, write data accounts for a large portion of memory references in deep learning workloads. Specifically, write data is responsible for 63.7-80.4% of total memory references irrespective of dataset and workload types as shown in Fig. 1. In traditional workloads, however, read data accounts for a majority of memory references in most cases though write data is dominant in some workload cases such as photo, where write data accounts for 56.6% of memory references. In game, office, and PDF, read data accounts for 64.5%, 54.9%, and 68.7%, respectively.

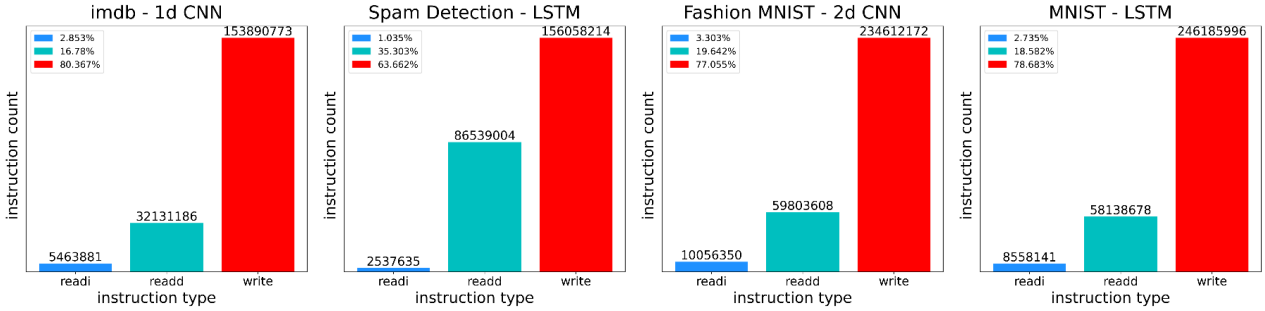


Figure 1. Ratio of memory references in deep learning workloads.

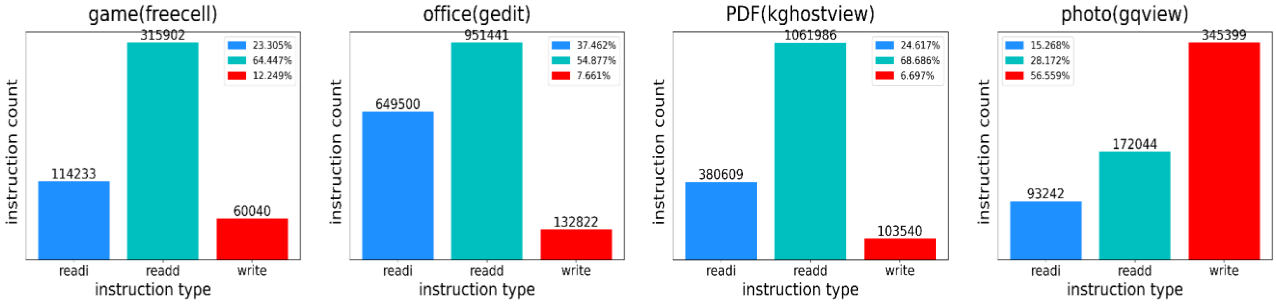


Figure 2. Ratio of memory references in traditional workloads.

3. Memory Reference Trends

Figs. 3 and 4 show the memory reference trend for deep learning and traditional workloads, respectively, based on memory addresses. In the figure, blue and red curves represent read and write operations, respectively. The memory address discussed here is not a physical memory address, but a logical address generated for each workload. As shown in the figure, though there are differences in reference frequency for each workload, a significant number of memory references are concentrated in specific address areas in both deep learning and traditional workloads. The reason that the difference in memory reference addresses by workload is not large is because when each process is created, the memory address areas are formed into code, data, stack, heap, and library areas, and memory references are made based on these areas.

To further examine memory reference trends, Figs. 5 and 6 show memory references only for accessed memory blocks rather than all memory regions for deep learning and traditional workloads, respectively. For this purpose, each memory block is assigned a unique number. As shown in the figure, we can see in deep learning workloads that read and write operations occur symmetrically to some extent. That is, in regions where reading occurs a lot, writing also appears a lot. This is because the steps of training by reading data and writing the result after that training generate memory references repeatedly. Unlike deep learning workloads,

we can see that write operations occur irrelevant to read operations in traditional workloads as shown in Fig. 6.

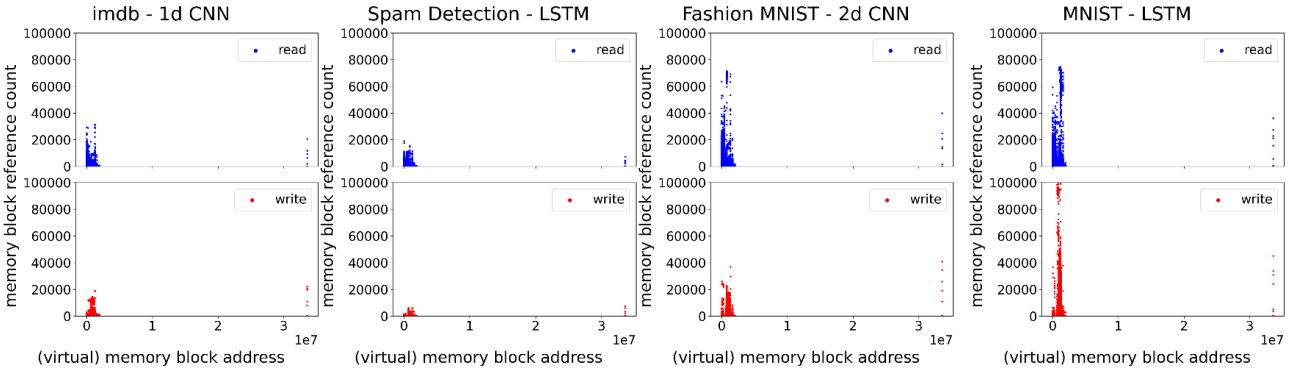


Figure 3. Memory reference trend of deep learning workloads based on memory addresses.

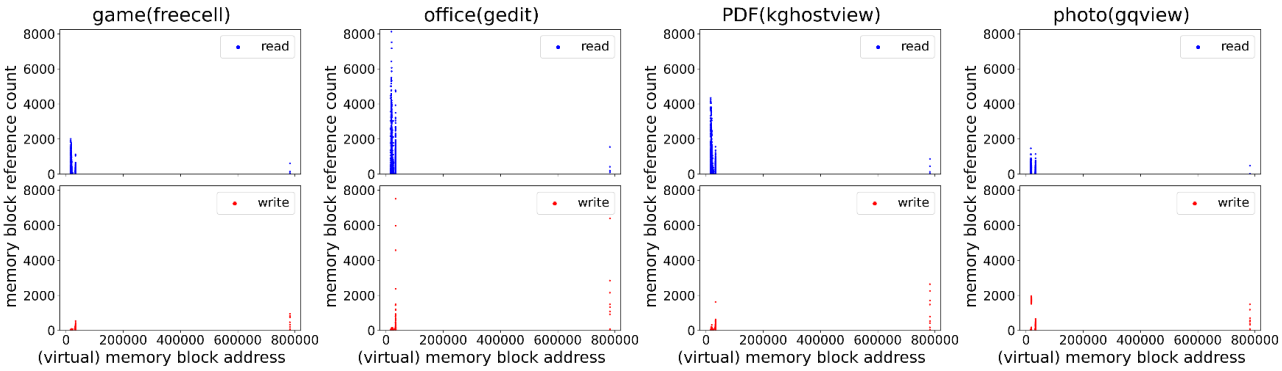


Figure 4. Memory reference trend of traditional workloads based on memory addresses.

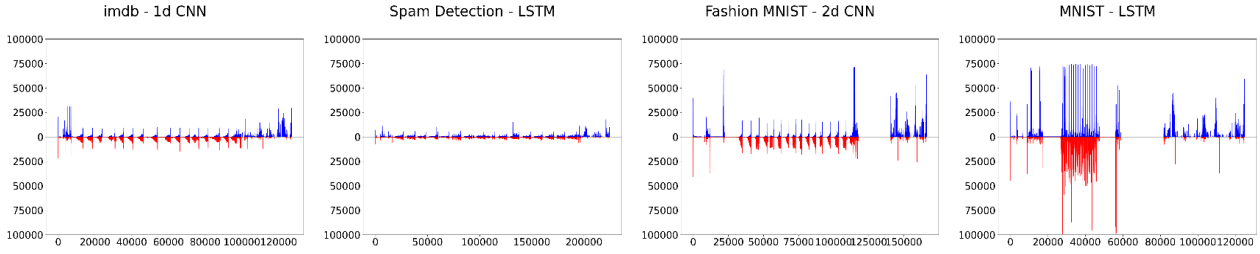


Figure 5. Memory reference trend of deep learning workloads with unique block number.

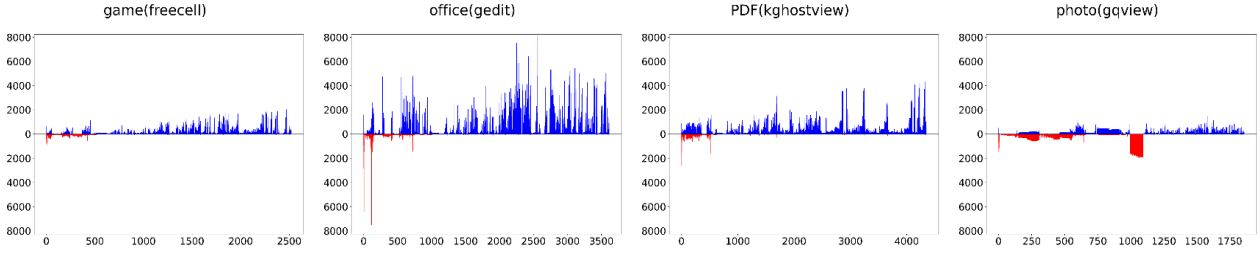
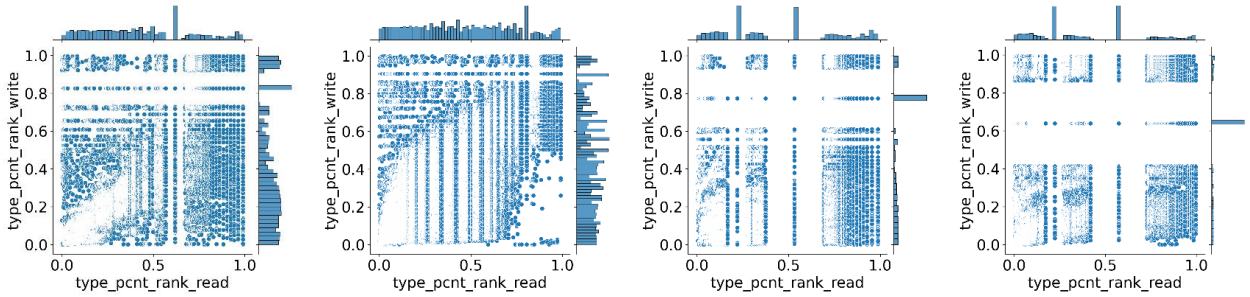


Figure 6. Memory reference trend of traditional workloads with unique block number.

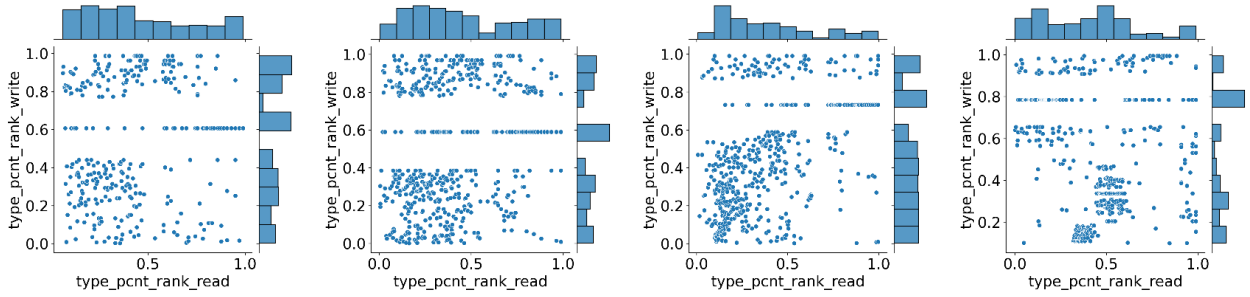
4. Correlation between Reads and Writes

In this section, we analyze the correlation of read and write references. That is, we investigate whether memory areas with high read rankings also have high write rankings. Figs. 7 and 8 plot the correlation of read and write rankings for the same block for deep learning and traditional workloads, respectively. As shown in the figure, there is a certain level of correlation between the rankings of read references and write references. In particular, the correlation of read and write is high in some deep learning workloads like IMDB and Spam Detection. In contrast, as shown in Fig. 8, the correlation between read and write references is relatively lower in traditional workloads. To exactly quantify this, we calculate the Pearson Correlation Coefficient based on read and write rankings. The result shows that the values of IMDB and Spam Detection are 0.72 and 0.76, which are very large compared to traditional workloads, where the values range -0.31 to 0.47.



(a) IMDB (b) Spam Detection (c) Fashin MNIST (d) MNIST

Figure 7. Correlation of read and write rankings for deep learning workloads.



(a) Game (b) Office (c) PDF (d) Photo

Figure 8. Correlation of read and write rankings for traditional workloads.

5. Analysis of Reference Skewness

In this section, we analyze the skewed popularity of memory references of deep learning workloads in comparison with traditional workloads. This is important for determining the hot data of deep learning workloads that reside in memory and setting an appropriate size of memory for the system running the workload. Skewed popularity distributions are usually modeled by the Zipf-like distribution. So, our analysis focuses on the modeling of memory references as a Zipf-like distribution. Figs. 9 and 10 show the number of times that a block has been referenced for the ranking of the block, where ranking 1 is the most frequently referenced block. Note that both axes in the figure are in log-scale. The curve in the figure shows that references

are excessively biased to some hot blocks. The left part of the curves can be well modeled by a straight line (denoted as blue, red, and green lines for read, write, and total references), which implies that the reference frequency of the ranking i is proportional to $1/i^b$, where b is the slope of the line. This type of distribution is called a Zipf-like distribution. When b approaches 1, the popularity of blocks is heavily skewed. The skew factor in our analysis is in the range of 0.3 to 0.5 regardless of operation types in deep learning workloads. This is different from traditional workloads shown in Fig. 10, where the skew factor of write operations ranges between 0.4 and 0.8. Although write accounts for a large portion of memory references in deep learning workloads, the skew factor is relatively smaller compared to traditional workloads. In other words, there are many memory writes in deep learning workloads but they are not excessively concentrated to some hot blocks.

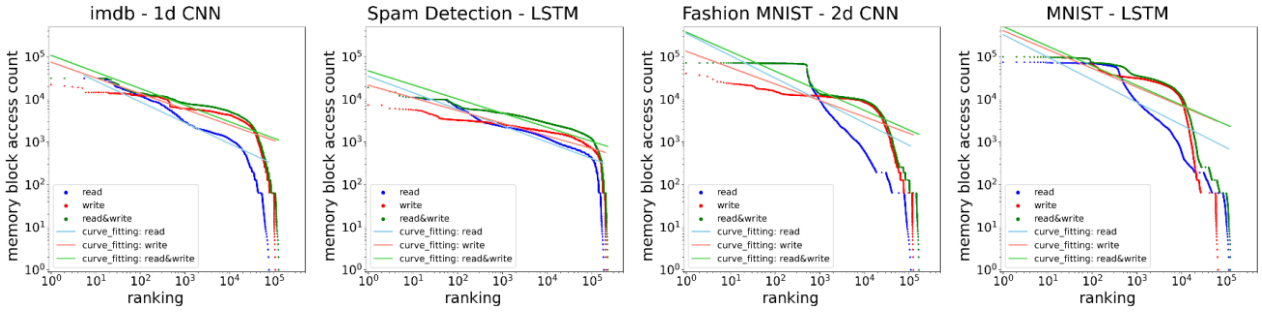


Figure 9. Memory reference trend as block ranking increases (deep learning workload).

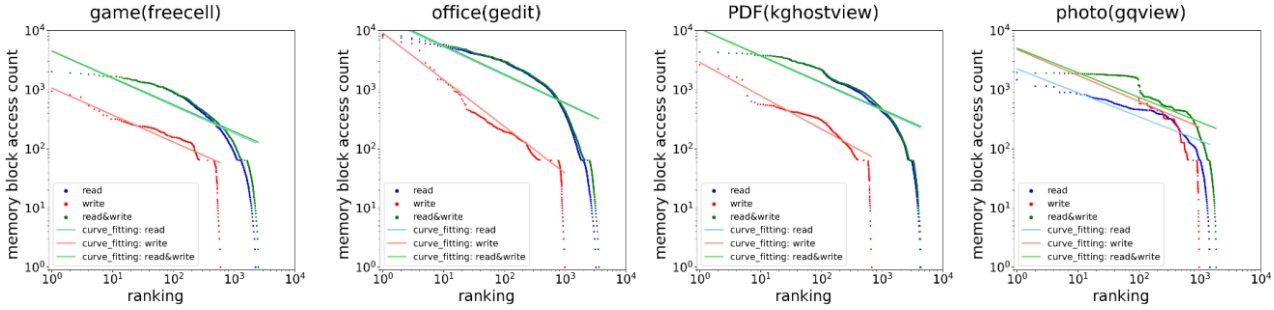


Figure 10. Memory reference trend as block ranking increases (traditional workload).

6. Conclusion

As the dataset of deep learning workloads increasingly grows, it is difficult to accommodate the entire dataset in memory, leading to significant performance degradations. To handle this situation, this article conducted characterization studies for deep learning memory references. In particular, we extracted memory reference traces of deep learning workloads, and analyzed them with respect to reference types, operations, and reference skewness. From our analysis, we observed some important characteristics of deep learning memory references differentiated from traditional workloads. First, instruction references accounts for only 1-3.3% of memory references in deep learning workloads, which is quite different from traditional workloads of 15.3-37.5%. Second, write references are dominant in deep learning workloads accounting for 63.7-80.4% of memory references. This is also different from read-intensive traditional workloads. Third, the skew factor of write references is small compared to traditional workloads. We anticipate that the analysis performed this article will be helpful in managing memory management systems of deep learning workloads efficiently.

Acknowledgement

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1009275) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] S. Dargan, M. Kumar, M.R. Ayyagari and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," Archives of Computational Methods in Engineering, vol. 27, pp. 1071–1092, 2020. DOI: https://doi.org/10.1007/s11831-019-09344-w

[2] J. Li, N. Mirza, B. Rahat and D. Xiong, "Machine learning and credit ratings prediction in the age of fourth industrial revolution," Technological Forecasting and Social Change, vol. 161, pp. 1–13, 2020. DOI: https://doi.org/j.techfore.2020.120309

[3] S. Idowu, D. Strüber and T. Berger, "Asset management in machine learning: state-of-research and state-of-practice," ACM Computing Surveys, vol. 55, no. 7, pp 1–35, 2022. DOI: https://doi.org/10.1145/3543847

[4] H. Fujiyoshi, T. Hirakawa and T. Yamashita, "Deep learning-based image recognition for autonomous driving," IATSS Research, vol. 43, no. 4, pp. 244–252, 2019. DOI: https://doi.org/10.1016/j.iatssr.2019.11.008

[5] J. Xiong, D. Yu, S. Liu, L. Shu, X. Wang et al., "A review of plant phenotypic image recognition technology based on deep learning," Electronics, vol. 10, no. 1, pp. 1–19, 2021. DOI: https://doi.org/10.3390/ electronics10010081

[6] I. H. Sarker, M. M. Hoque, M. K. Uddin and T. Alsanoosy, "Mobile data science and intelligent apps: concepts, AI-based modeling and research directions," Mobile Networks and Applications, vol. 26, pp. 285–303, 2021. DOI: https://doi.org/10.1007/s11036-020-01650-z

[7] E. Lee, H. Kang, H. Bahn and K. G. Shin, "Eliminating periodic flush overhead of file I/O with non-volatile buffer cache," IEEE Transactions on Computers, vol. 65, no. 4, pp. 1145–1157, 2016. DOI: https://doi.org/10.1109/TC.2014.2349525

[8] D. T. Nguyen, H. Kim, H. J. Lee and I. J. Chang, "An approximate memory architecture for a reduction of refresh power consumption in deep learning applications," in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, pp. 1–5, 2018. DOI: https://doi.org/10.1109/ISCAS.2018.8351021

[9] S. Yoo, Y. Jo and H. Bahn, "Integrated scheduling of real-time and interactive tasks for configurable industrial systems," IEEE Transactions on Industrial Informatics, vol. 18, no. 1, pp. 631–641, 2022. DOI: https://doi.org/10.1109/TII.2021.3067714

[10] N. Nethercote and J. Seward, "Valgrind: a framework for heavyweight dynamic binary instrumentation," ACM SIGPLAN Notices, vol. 42, no. 6, pp. 89–100, 2007. DOI: https://doi.org/10.1145/1273442.1250746