MLAE2: Metareasoning for Latency-Aware Energy-Efficient Autonomous Nano-Drones

Mozhgan Navardi, Tinoosh Mohsenin
Department of Computer Science & Electrical Engineering, University of Maryland Baltimore County, USA

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181715

***Abstract*—Safety, low-cost, small size, and Artificial Intelligence (AI) capabilities of drones have led to the proliferation of autonomous tiny Unmanned Aerial Vehicles (UAVs) in many applications which are dangerous, unknown, or time-consuming for humans. Deep Neural Networks (DNNs) have enabled autonomous navigation while using captured data by drone sensors as input to the model. Due to the extreme complexity of DNNs, cloud-based approaches have been highly addressed in which a drone is connected to the cloud and sends the data to the cloud, and takes the result. On the other hand, emerging tiny machine learning models and edge computing brings significant improvement in energy efficiency and latency with respect to cloud-based approaches. However, there is a trade-off in these two implementations for model accuracy, latency, and energy efficiency. For instance, applying tiny machine learning models leads to lower latency but it sacrifices model accuracy in comparison to cloud-based computing. To address these challenges, we consider multiple models and introduce a new approach named MLAE2 which applies Metareasoning approach for Latency-Aware Energy-Efficient autonomous drones. Metareasoning monitors parameters such as latency and energy consumption for different algorithms and chooses the appropriate algorithm due to the environmental situation changes. To Evaluate our approach we extract the power consumption and latency for both cloud-based computing and edge computing while deploying multiple models on a tiny drone named Crazyflie. The experimental results show that MLAE2 successfully meets the latency constraint while maximizing model accuracy and improving energy efficiency.**

***Index Terms*—Metareasoning, Tiny Machine Learning, Autonomous Systems, Obstacle Avoidance, Drone Navigation.**

I. Introduction and Related Work

Nowadays, Internet of Things (IoT) devices such as tiny Unmanned Aerial Vehicles (UAVs) have attracted significant attraction which has led proliferation of autonomous systems [1], [2]. Autonomous systems have enabled numerous indoor and outdoor applications such as search and rescue, and source seeking which are unsafe or impassible in some cases for humans [3]. Machine Learning (ML) algorithms have shown significant performance in such systems for autonomous drone navigation and object detection [4].

Tiny drones have equipped with various sensors such as LIDAR and cameras and the collected data by these sensors can be fed to ML Neural Networks (NNs). Vision-based Deep Neural Networks (DNNs) [3], [5], [4] or Reinforcement Learning (RL) [6], [7], [8], [9], [10] approaches can be deployed on such tiny drones to enable them to perform complex tasks. Due to the intensive computational requirements of DNNs models, cloud-based approaches which provide unlimited computational capacity have been highly addressed in this area. However, cloud-based implementation requires drone and cloud communication to transfer and process raw data to the server and send back the result to the drone. Therefore, bandwidth limitation leads to latency communication in cloud-based implementation which is challenging in real-time applications such as autonomous drone navigation. Moreover, security concerns and power consumption due to communication are other challenges of such approaches. Edge computing is a promising solution to solve these problems.

Resource-constrained devices like tiny UAVs have limited sources of power and computation capacity. Therefore, for edge computing, DNNs need to be optimized with regard to the number of parameters and computations. These requirements have led to the tiny machine learning system's emergence which brings DNNs on low-power and resource-constrained devices. Tiny machine learning is able to significantly reduce energy and latency in comparison with the cloud-based approaches as all the process has been done on the edge. However, model optimization techniques like pruning [11] and quantization [12] reduce the model size and lead to the accuracy drop. On the other hand, since the application space is continuously changing for edge devices it is vital to prevent misprediction [13]. Therefore, both cloud-based and edge deployment can be challenging in different situations.

Metareasoning as an anytime algorithm improves the agent's decision-making process based on the current situation [14], [15], [16], [17], [18], [19], [20]. As a result, the drone which is the agent can dynamically switch between cloud-based and edge computing implementation while considering the power consumption, latency constraints, and model accuracy metric. Metareasoning as a higher-level unit monitors the environment and provides the best algorithm in the current situation. In this paper, we proposed MLAE2; a Latency-Aware Energy-Efficient autonomous drone navigation that has applied a Metareasoning approach to improving latency. In summary, the main contributions of this paper are as follows:

- Edge implementation of tiny machine learning models on a resource-constrained tiny drone.
- Considering cloud-based and edge computing advantages for energy-efficient autonomous drone navigation.
- Applying metareasoning for latency-aware decision-making drone navigation based on environment situation.
- Meeting both latency and power consumption constraints in low-power real-time applications.

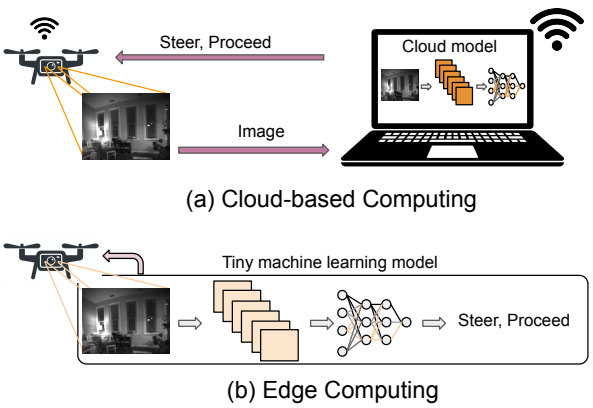

Fig. 1: A high-level diagram of cloud-based computing and edge computing [21]. (a) In cloud-based computing, a DNN model is implemented on the server and the drone constantly communicates with the server. In this approach, the drone will be failed if the communication is interrupted or disconnected. (b) A tiny model is implemented on a drone. The model has lower accuracy due to the model size optimization.

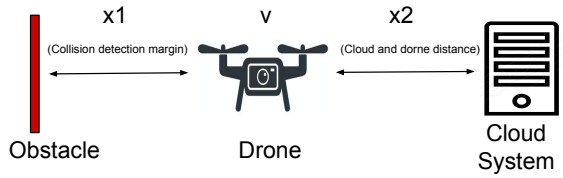

Fig. 2: Environment parameters are constantly changing during drone navigation and can affect drone performance. For instance, drone and obstacle distance ($x1$), drone and server distance ($x2$), and drone velocity (v).

II. Proposed MLAE2 Approach

In this section, we discussed the motivation behind the proposed approach. Moreover, we illustrated the system overview and explained the DNN model architecture. Then, we presented the MLAE2 approach for autonomous drone navigation.

A. Motivation

In this section, we give a motivational example to show the effect of different vision-based DNNs approaches on autonomous navigation. Figure 1 illustrates two potential approaches; cloud-based and edge computing. Figure 1 (a) shows how drone and server communicate together through WiFi in cloud-based implementations. In this approach, the communication latency will be increased when the drone and server distance increases. On the other hand, in the edge computing an optimized tiny machine learning model with lower accuracy processes data shown in Figure 1 (b). Since drone navigation is a real-time task, the inference latency for processing data has to be less than the constraint latency. However in cloud-based computing even though we have higher accuracy, there is no guarantee of meeting the latency constraint due to environmental changes. Figure 2 shows some parameters that can cause the cloud-based implementation to be failed. While the drone is flying in the environment, distance from the obstacle ($x1$) or server ($x2$) can change the communication latency and constraint latency, respectively. Therefore, it is crucial to present an approach that guarantees the latency requirements while maximizing accuracy.

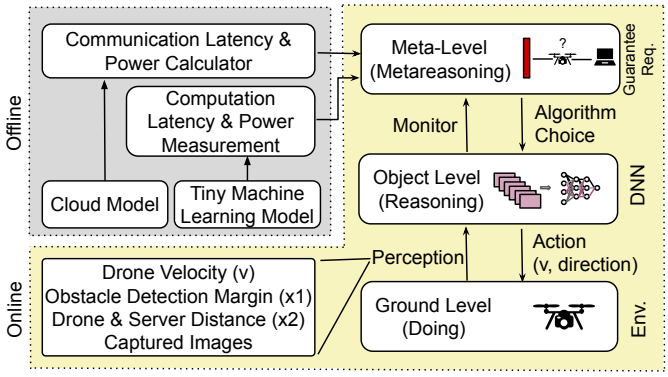

Fig. 3: A system overview of the proposed approach includes online and offline phases. In the offline phase, power consumption and latency are measured and calculated for both edge and cloud implementations. These measurements along with environmental factors (like drone and server distance) go to the metareasoning level for parameter adjustment (like velocity and obstacle detection margin) and algorithm choice in the online phase.

B. System Overview

To deal with the environmental changes challenges and meet the power and latency constraints, we applied metareasoning. Metareasoning by monitoring the changes provides better decision-making for the drone while exploring the environment. Figure 3 depicts an overview of the proposed MLAE2 approach which includes two offline and online phases.

Offline Phase. In the offline phase, power consumption and latency are measured. We measured computation latency and power measurement when deploying the model on the tiny drone. On the other hand, since most portion of the latency and power consumption is related to the communication in the cloud-based implementation, we considered communication power and latency. To measure the latency, we extracted the communication latency every five meters and store it in an array. Then, we calculated the continuous latency with the help of regression. We explained in more detail about the latency communication in Section III-B.

Online Phase. The online phase includes three levels: ground level, object level, and meta-level. In the ground level unit, the drone flies in the environment and sends perception to the object level. In the object level unit, we have two algorithms cloud-based and edge computing. This unit monitors by the meta-level and the metareasoning approach chooses one of the algorithms. The meta-level unit considers latency constraint which is calculated based on the current drone velocity (v) and collision detection margin (x1) and compared it with communication and computation latency. Then, it chooses the best algorithm that meets the latency constraints. The algorithm will be executed in the object level unit and the result will be sent to the ground level as an action to be performed by the drone.

C. Vision-based Deep Neural Networks (DNNs)

The main application that we considered in this paper is autonomous drone navigation. We used two energy-efficient pre-trained models presented in [5]: A cloud-based model and an optimized model for edge computing.

Algorithm 1 Metareasoning for Latency-Aware Navigation (MLA)

1: $t_{edge} \leftarrow measure\_edge\_latency()$
2: $t_{cloud\_based}, x2 \leftarrow measure\_cloud\_based\_latency()$
3: $t_{current} \leftarrow 0$
4: $t_{constraint} \leftarrow x1/v, s \leftarrow x2/v$
5: **while** $(t_{current} < t_{operation})$ **do**
6: $\quad i = t_{current}/s$
7: $\quad$ **if** $t_{cloud\_based}[i] < t_{constraint}$ **then**
8: $\quad\quad t_{current} \leftarrow t_{current} + t_{cloud\_based}[i]$
9: $\quad$ **else if** $t_{edge} < t_{constraint}$ **then**
10: $\quad\quad t_{current} \leftarrow t_{current} + t_{edge}[i]$
11: $\quad$ **else**
12: $\quad\quad t_{constraint} \leftarrow min(t_{edge}, t_{cloud\_based}[i])$
13: $\quad\quad v = x1/t_{constraint}, s = x2/v$
14: $\quad$ **end if**
15: **end while**

Cloud-based Model. Cloud-based model architecture is based on mobilenetV1 [22] architecture with a width multiplier of 0.5 and 97% accuracy. The input image size is 324x244, the number of parameters is about 830K, and the performance when deploying this model on an M1 mac pro is 14.8 GOPS.

Tiny Machine Learning Model. This model is a one-layer residual block resnet [23] with 92% accuracy. This model is optimized and quantized with 8-bit post-train quantization. The image input size is 200x200 and it has about 100K parameters.

D. Metareasoning for Latency-Aware Energy-Efficient (MLAE2) Drone Navigation

In this section, we presented the detailed implementation of the proposed MLAE2 approach in this paper. Metareasoning monitored the navigation process and guarantee that meet the latency constraints while using energy-efficient DNN models.

Algorithm 1. Metareasoning for Latency-Aware Navigation (MLA) algorithm gets two energy-efficient models as the input. Lines 1 and 2 measure the edge latency for computations and cloud-based latency array for communication latency. Lines 3 and 4 initialize current time $t_{current}$ and constraint latency $t_{constraint}$ based on collision detection margin factor x1. Moreover, it initializes s as a factor for increasing communication latency in the cloud-based implementation. In the online phase of the algorithm, lines 5-15, the meta-level unit keeps monitoring latency and chooses cloud-based or edge implementation that meets the constraint. Lines 12 and 13 show how meta-level adjusts drone velocity v and communication latency factor s if the latency requirements can not be met on the current setting. For instance, if both communication and computation latency does not meet constraint latency, the meta-level decreases the drone velocity. Therefore, the drone has more time to process the data before the collision.

III. Experimental Results

A. Experimental Setup

To evaluate the proposed MLAE2 approach a tiny drone named Crazyflie [4] is used in this paper. Figure 4 (a) depicts that Crazyflie is equipped with a gray-scale camera and an AI-deck which includes a GAP8 with eight RISC-V processors. For the cloud-based implementation, we used a MacBook Pro

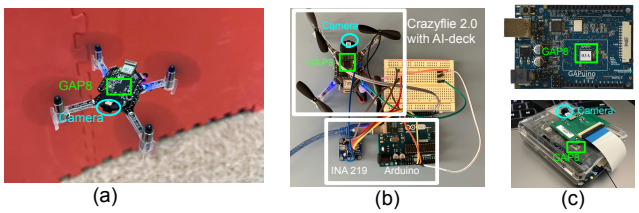

Fig. 4: (a) Crazyflie with AI-Deck, (b) drone power measurement setup [5], [6], and (c) power measurement setup includes GAPuino with camera [24].

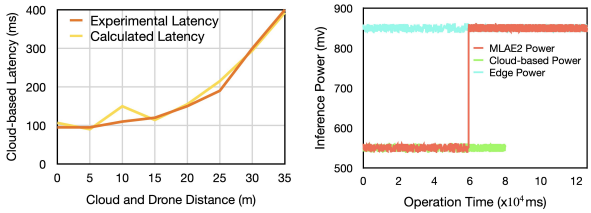

Fig. 5: (a) Cloud-based experimental and calculated latency for the inference phase while the cloud and drone distance is increasing. (b) Inference phase power trace for three approaches: cloud-based, edge computing, and MLAE2. Drone velocity (v) is 0.5 m/s and the collision margin ($x1$) is 15 cm.

with an Apple M1 Pro chip. We used two pre-trained models in [5] as tiny machine learning and cloud models. Figure 4 (b) and (c) depict two different setups for the power measurement including a drone [5], [6] used in this paper and GAPuino [24], [25], [26] which has the GAP8 processor [27]. After extracting offline power consumption and latency, we used them in the simulation while implementing the proposed algorithm in this paper. We considered 20mv and 4ms oscillation for the offline power and latency in the simulation.

B. Experimental Results

In this section, we explained inference latency measurement in the offline phase of the proposed approach. Then, power consumption and latency trace is presented with the MLAE2 approach implementation.

Offline Inference Latency. Figure 5 (a) illustrates latency measurement for the cloud-based implementation when we deployed the cloud model on a laptop and the drone communicated with the laptop through WiFi while flying. We did the experiments every five meters which are shown as experimental latency in Figure 5. Then, with the help of regression, we plot a continuous latency to have the latency for all of the distances. The results depict that with an increase in distance between the drone and the server, the latency is also increasing and after 35m the server will be disconnected. We also measure the computation latency when applying edge computing and the results show latency is equal to 40 ms.

Online Power Consumption Trace. Figure 5 (b) shows the simulation results for online power trace by considering drone velocity 0.5ms and collision detection margin 0.15m. In this experiment, real power consumption is extracted while deploying the cloud and edge model. The results show that the MLAE2 approach switches to edge computing at 60ms operational time which the reason is explained in the following.

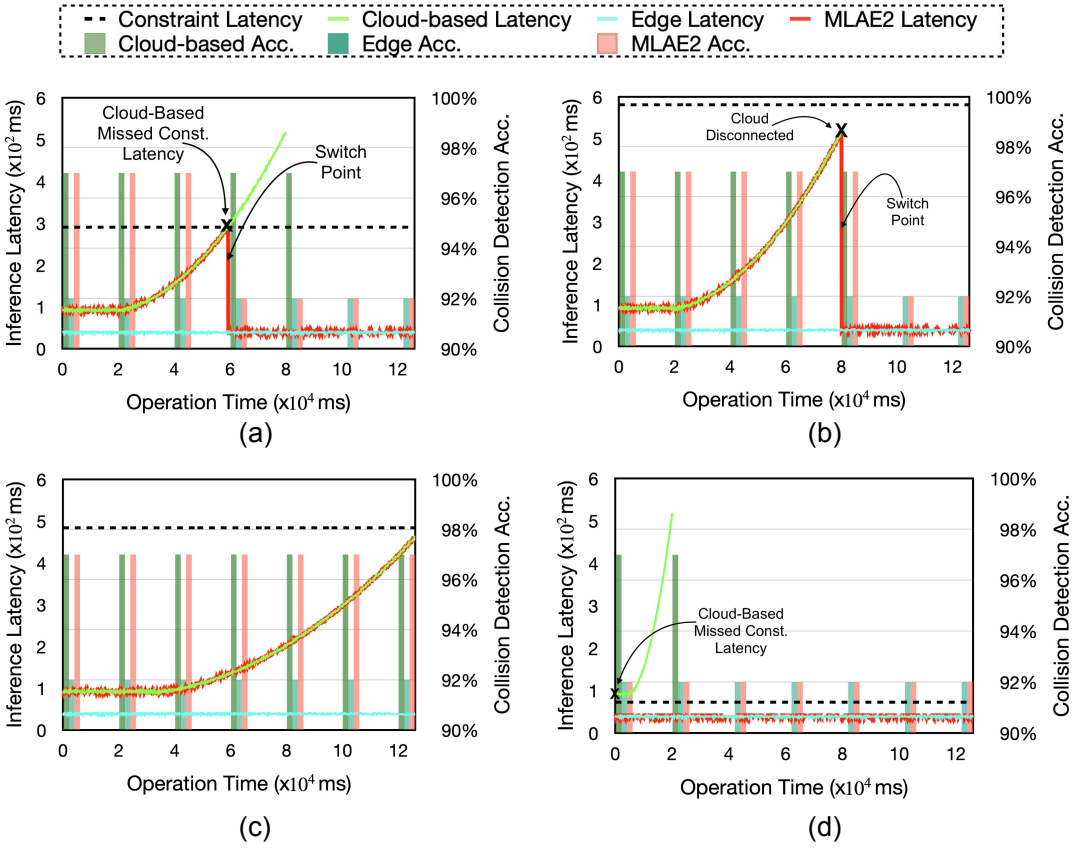

Fig. 6: Online inference latency with different configurations regards to the velocity (v) and collision margin ($x1$) in the simulation: (a) $v = 0.5m/s, x1 = 15cm$, (b) $v = 0.5m/s, x1 = 30cm$, (c) $v = 0.3m/s, x1 = 15cm$, and (d) $v = 2m/s, x1 = 15cm$.

Online Inference Latency Trace. Figure 6 illustrates latency trace and model accuracy of cloud-based, edge, and MLAE2 approaches for 120ms operational time. In Figure 6 (a) cloud-based approach missed the constraint latency by increasing the distance between the server and drone at 60ms while in the MLAE2 approach it switched to edge computing. Figure 6 (b) shows that even by decreasing the drone velocity to increase the latency constraint, the cloud-based approach failed at 80ms due to the communication link disconnection. However, the MLEA2 approach switches to edge computing when the communication link is disconnected. Figure 6 (c) and (d) depict the MLAE2 approach choosing cloud or edge only, respectively. In conclusion, MLAE2 used the cloud-based approach which has the higher accuracy as much as possible and also met the constraint latency.

IV. Conclusion

In this paper, we proposed an approach named MLAE2 which applied Metareasoning for Latency-Aware Energy-Efficient autonomous navigation. Two different DNN models are considered in this paper; (1) one model with a high number of computations and a high level of accuracy for cloud-based implementation and, (2) a tiny machine learning model with an optimized model size for edge computing. We calculated latency constraints based on the distance between the drone and the obstacles. In addition, we measured communication latency when the drone is flying and the distance between the server and the drone is increasing. The MLAE2 approach maximized the accuracy of the obstacle detection model for autonomous drone navigation while meeting power and latency constraints. To evaluate the proposed approach, we used a drone named Crazyflie with a low-power GAP8 RISC-V processor to measure the latency and power consumption. The results showed that the MLAE2 approach successfully switches to edge computing when the communication latency is higher than the constraint latency. Moreover, by applying metareasoning, a higher level unit to monitor the drone and changes in environmental situations, the drone could successfully navigate even if the communication link is disconnected due to the high distance between the drone and the server.

V. Acknowledgment

We thank Griffin Bonner, Haoran Ren, Aidin Shiri, Edward Humes, and Tejaswini Manjunath for the initial discussions and experiments. This project was sponsored by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF2120076.

REFERENCES

[1] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet of Things Journal*, 2021.

[2] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and ai for uavs: Opportunities and challenges," *IEEE Internet of Things Journal*, 2022.

[3] B. P. Duisterhof, S. Krishnan, J. J. Cruz, C. R. Banbury, W. Fu, A. Faust, G. C. de Croon, and V. J. Reddi, "Tiny robot learning (tinyrl) for source seeking on a nano quadcopter," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7242–7248.

[4] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64-mw dnn-based visual navigation engine for autonomous nano-drones," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8357–8371, 2019.

[5] M. Navardi, A. Shiri, E. Humes, N. R. Waytowich, and T. Mohsenin, "An optimization framework for efficient vision-based autonomous drone navigation," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022, pp. 304–307.

[6] A. Shiri, M. Navardi, T. Manjunath, N. R. Waytowich, and T. Mohsenin, "Efficient language-guided reinforcement learning for resource constrained autonomous systems," *IEEE Micro*, 2022.

[7] M. Navardi, P. Dixit, T. Manjunath, N. R. Waytowich, T. Mohsenin, and T. Oates, "Toward real-world implementation of deep reinforcement learning for vision-based autonomous drone navigation with mission," *3rd Workshop on Closing the Reality Gap in Sim2Real Transfer for Robotics on Robotics: Science and Systems (RSS)*, 2022.

[8] B. Prakash, N. Waytowich, T. Oates, and T. Mohsenin, "Towards an interpretable hierarchical agent framework using semantic goals," *AAAI Fall Symposium 202 AI-HRI*, 2022.

[9] B. Prakash, N. Waytowich, T. Mohsenin, and T. Oates, "Automatic goal generation using dynamical distance learning," *arXiv preprint arXiv:2111.04120*, 2021.

[10] B. Prakash, N. Waytowich, T. Oates, and T. Mohsenin, "Interactive hierarchical guidance using language," *AAAI Fall Symposium 2021 AI-HRI*, 2021.

[11] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *arXiv preprint arXiv:1710.01878*, 2017.

[12] M. Hosseini and T. Mohsenin, "Qs-nas: Optimally quantized scaled architecture search to enable efficient on-device micro-ai," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2021.

[13] P. Shukla, S. Nasrin, N. Darabi, W. Gomes, and A. R. Trivedi, "Mc-cim: Compute-in-memory with monte-carlo dropouts for bayesian edge intelligence," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2022.

[14] E. Horvitz, *Metareasoning: Thinking about thinking*. MIT Press, 2011.

[15] R. Ackerman and V. A. Thompson, "Meta-reasoning: Monitoring and control of thinking and reasoning," *Trends in cognitive sciences*, vol. 21, no. 8, pp. 607–617, 2017.

[16] S. T. Langlois, O. Akoroda, E. Carrillo, J. W. Herrmann, S. Azarm, H. Xu, and M. Otte, "Metareasoning structures, problems, and modes for multiagent systems: A survey," *IEEE Access*, vol. 8, pp. 183 080–183 089, 2020.

[17] M. K. Dawson, "Metareasoning approaches to thermal management during image processing," Ph.D. dissertation, University of Maryland, College Park, 2022.

[18] J. Svegliato, C. Basich, S. Saisubramanian, and S. Zilberstein, "Using metareasoning to maintain and restore safety for reliably autonomy," in *Submission to the IJCAI Workshop on Robust and Reliable Autonomy in the Wild (R2AW)*, 2021.

[19] J. Caylor, J. W. Herrmann, C. Hung, A. Raglin, and J. Richardson, "Metareasoning for multi-criteria decision making using complex information sources," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, vol. 12113. SPIE, 2022, pp. 305–313.

[20] E. Carrillo, S. Yeotikar, S. Nayak, M. K. M. Jaffar, S. Azarm, J. W. Herrmann, M. Otte, and H. Xu, "Communication-aware multi-agent metareasoning for decentralized task allocation," *IEEE Access*, vol. 9, pp. 98 712–98 730, 2021.

[21] M. Navardi, E. Humes, and T. Mohsenin, "E2edgeai: Energy-efficient edge computing for deployment of vision-based dnns on autonomous tiny drones," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*. IEEE, 2022, pp. 504–509.

[22] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[23] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] GreenWavesTechnologies, "Greenwaves technologies, gapuino development board. [online]," *Available: https://greenwaves-technologies.com/product/gapuino/*.

[25] M. Scherer, F. Sidler, M. Rogenmoser, M. Magno, and L. Benini, "Widevision: A low-power, multi-protocol wireless vision platform for distributed surveillance," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2022, pp. 394–399.

[26] A. Bahr, M. Schneider, M. A. Francis, H. M. Lehmann, I. Barg, A.-S. Buschhoff, P. Wulff, T. Strunskus, and F. Faupel, "Epileptic seizure detection on an ultra-low-power embedded risc-v processor using a convolutional neural network," *Biosensors*, vol. 11, no. 7, p. 203, 2021.

[27] GreenWavesTechnologies, "Gap8 processor architecture. [online]," *Available: https://greenwaves-technologies.com/manuals/BUILD/HOME/html/index.html*.

[28] S. S. Sahoo, B. Ranjbar, and A. Kumar, "Reliability-aware resource management in multi-/many-core systems: A perspective paper," *Journal of Low Power Electronics and Applications*, vol. 11, no. 1, p. 7, 2021.

[29] N. Rohbani and S.-G. Miremadi, "A low-overhead integrated aging and seu sensor," *IEEE Transactions on Device and Materials Reliability*, vol. 18, no. 2, pp. 205–213, 2018.

[30] M.Hosseini *et al.*, "Neural networks for pulmonary disease diagnosis using auditory and demographic information," in *epiDAMIK 2020: 3rd epiDAMIK ACM SIGKDD International Workshop on Epidemiology meets Data Mining and Knowledge Discovery*. ACM, 2020, pp. 1–5, in press.

[31] D. V. Christensen *et al.*, "2021 roadmap on neuromorphic computing and engineering," *arXiv preprint arXiv:2105.05956*, 2021.

[32] B. Prakash *et al.*, "Guiding safe reinforcement learning policies using structured language constraints," in *SafeAI workshop Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI, 2020.

[33] H.-A. Rashid *et al.*, "A low-power lstm processor for multi-channel brain eeg artifact detection," in *2020 21th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2020.