2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) | 978-1-6654-5818-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAIIC54071.2022.9722636

TinyML: A Systematic Review and Synthesis of Existing Research

Hui Han
Data Science Department
Fraunhofer Institute for Experimental Software Engineering IESE
Kaiserslautern, Germany
hui.han@alumnos.upm.es

Julien Siebert
Data Science Department
Fraunhofer Institute for Experimental Software Engineering IESE
Kaiserslautern, Germany
julien.siebert@iese.fraunhofer.de

***Abstract*— Tiny Machine Learning (TinyML), a rapidly evolving edge computing concept that links embedded systems (hardware and software) and machine learning, with the purpose of realizing ultra-low-power and low-cost and efficiency and privacy, brings machine learning inference to battery-powered intelligent devices. In this study, we conduct a systematic review of TinyML research by synthesizing 47 papers from academic and grey publication since 2019 (the early TinyML publication starts from 2019). Relevant TinyML literature is analyzed from five aspects: hardware, framework, datasets, use cases, and algorithms/model. This systematic review will serve as a roadmap for understanding the literature within the new emerging field of TinyML.**

Keywords—TinyML, Systematic review, Data synthesis, MCUs, TensorFlow Lite, Neural networks

I. Introduction

TinyML (tiny machine learning) is a relatively new term that encompasses research and development at the intersection of embedded systems and machine learning [1]. The goal of TinyML is to apply machine learning inference on extremely low-power (under a milliwatt), low-cost (55 cents in 2023) microcontrollers (MCUs) [2]. Through the implementation of various battery-powered MCUs and streaming applications, TinyML facilitates real-time, on-site data collection, processing, analysis and interpretation. [3]. As a result, TinyML provides low latency, low power, and high privacy while avoiding the energy cost and data loss associated with wireless communication between edge devices and the cloud [4].

Although TinyML is an important and growing field, the academic research associated with the term remains at a very early stage at this time. As a result, efforts to synthesize TinyML research into an integration of a broad body of knowledge have been relatively limited [5]. To fill this gap, we propose a systematic method and synthesis of current research on TinyML.

A synthesis from various aspects can clarify issues and identify relations in a structured manner [6]. Therefore, to assist scholars to improve the understanding of TinyML, we adopt synthesis with a comprehensive and structured list of elements. Specifically, this article contributes to the TinyML literature by synthesizing current research with five aspects: hardware, framework, datasets, use cases, and algorithms/model.

This review serves as guidance for research exploration on TinyML. It aims to gain a better understanding of the state of the art of TinyML and to offer guidelines to TinyML practice. This study also provides researchers and practitioners with directions for future research in this emerging field.

The reminder of the paper is organized as follows. In Section II, we present a note on review methodology. Section III discuss the search results and Section IV describes data synthesis results. Finally, in Section V, we provide future research needs, and conclude.

II. Methodology

A systematic review approach is used to select studies and identify relevant articles [7]. Additionally, a synthesis methodology is adopted to synthesize the broad scope of selected papers in an integrative way [8]. PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) instruction is applied here to display a flow diagram of the literature search process (shown in Fig. 1).

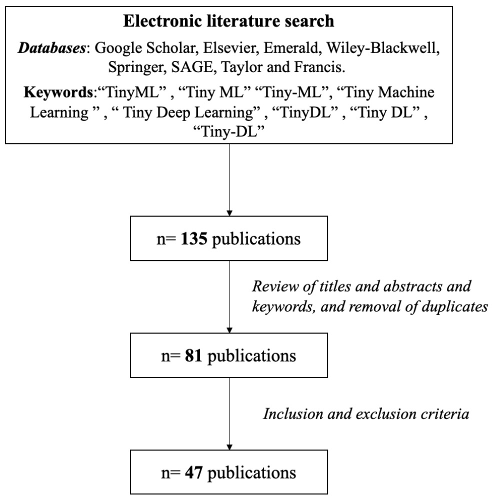
Electronic literature search
Databases: Google Scholar, Elsevier, Emerald, Wiley-Blackwell, Springer, SAGE, Taylor and Francis.
Keywords:"TinyML" , "Tiny ML" "Tiny-ML", "Tiny Machine Learning " , " Tiny Deep Learning" , "TinyDL" , "Tiny DL" , "Tiny-DL"
n= **135** publications
Review of titles and abstracts and keywords, and removal of duplicates
n= **81** publications
Inclusion and exclusion criteria
n= **47** publications

Fig. 1. PRISMA flow diagram.

A. Searching

We utilize web-based resources (Google Scholar) as the database. In addition, we manually search for articles from key authors in the TinyML field. The search strategy (each keyword for each time separately) is detailed below:

The research of Dr. Hui Han is funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan Fellowship Programme and the Fraunhofer Institute for Experimental Software Engineering IESE.

- The keyword “TinyML” is used for the first-round search.
- The keyword “Tiny ML” is used for the second-round search.
- The keyword “Tiny-ML” is used for the third-round search.
- The keyword “Tiny Machine Learning” is used for the fourth round search.
- The keyword “ Tiny Deep Learning” is used for the fifth-round search.
- The keyword “Tiny-DL” is used for the sixth-round search.
- The keyword “TinyDL” is used for the seventh-round search.
- The keyword “Tiny DL” is used for the eighth-round search.

In order to full search, we also use the six main databases as complementary sources:

- Elsevier (https://www.sciencedirect.com/);
- Emerald (https://www.emeraldinsight.com/);
- Wiley-Blackwell (https://onlinelibrary.wiley.com/);
- Springer (https://www.springer.com/fr);
- SAGE (https://us.sagepub.com/en-us/nam/home);
- Taylor and Francis (https://www.tandfonline.com/).

B. *Selection*

In order to give broader insights, selected studies not only contain academic literature but also grey publishing such as reports and working paper and the online newspaper. The publication is chosen in two steps: first by review of title and abstract and keywords, then by full-text review. The selection is not restricted by year of publication but papers have to be written in English because of limited translation resources (such as lack of language experts for translation work). All web-based resources are accessible in printed or downloadable form.

The initial literature search yielded 135 results. After reviewing the titles, abstracts and keywords, as well as removing duplicates, we obtained 81 unique records for further evaluation. After applying the inclusion and exclusion criteria when reading the full text, we excluded 34 articles for a final total of 47 articles (36 primary studies and 11 reviews) used for data synthesis.

C. *Data Extraction and Data Synthesis*

For data extraction and synthesis, we employ the content analysis method, which identifies the appearance of specific words, topics, or concepts within a text [9].

The purpose of the data synthesis approach is to integrate diverse range of studies into a conceptual map describing five TinyML elements: hardware, framework, datasets, use cases, and algorithms/model. Full details of the data extraction and synthesis are available from the first author upon request.

III. Search Results

Of the 47 publications in the review, 26 publications are conference proceedings, 15 journal articles including 13 research articles and 2 review articles, 2 book sections and 1 book, 1 report, 1 working paper and 1 newspaper article. We list the reference in Table 1.

TABLE I. Publication Types

Publication Types	**References**
Conference Proceedings	[10][11][2][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34]
Research Article	[3][35][36][37][38][39][40][41] [42][43][44][45][46]
Review Articles	[1][4]
Book Section	[47]
Book	[48][49]
Report	[5]
Working Paper	[50]
Newspaper Article	[51]

Figure 2 lists the main sources (by year) that have issued two or more publishing items. In total, there are 24 different types of sources involved in this review. Among the publication, most papers are published in different conferences, and 6 articles are published in 4 types of journals respectively: Sensors (3 papers), IEEE Circuits and Systems Magazine (1 paper), IEEE Transactions on Circuits and Systems II: Express Briefs (1 paper) and Journal of Sensor and Actuator Networks (1 paper). TinyML Research Symposium (10 papers) is very popular among these conferences referring to TinyML. In addition, 11 papers were pre-print from arXiv and one book was published in 2019 by O’Reilly Media and one newspaper article is from IEEE IoT Newsletter.

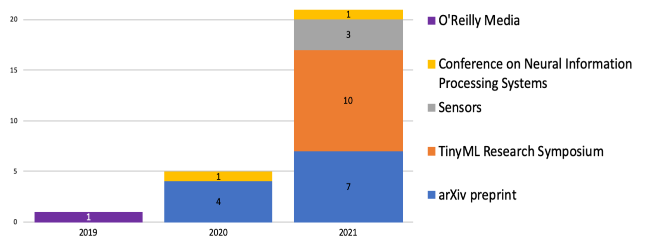

Fig. 2. Publication source since 2019.

IV. Data Synthesis Results

Synthesis can spot the location of every issue on an integrative map of TinyML. This step aims to paint one abstraction frame to precisely record the information gained from selected studies. We use Mendeley and Microsoft Excel spreadsheets to synthesis the 47 papers into five elements:

hardware, framework, datasets, use cases, and algorithms/model [1], [2], [4], [41].

A. Hardware

TinyML is operated on low-power microcontrollers boards with extensive hardware extraction [50]. In this study, we list the main hardware (Table 2) applied by two or more publishing items. 8 papers generally mentioned Microcontroller units (MCUs) without hardware details. On the other hand, 5 papers clearly specified STM32 MCU and 2 papers on Apollo3 MCU. ARM Cortex-M processors (mentioned by 4 papers) are commonly integrated by these MCUs when applied TinyML.

TABLE II. HARDWARE

Hardware	Count
MCUs	8
STM32	5
ARM Cortex-M (M0 and M7) series	4
Ambiq Apollo 3	2
Arduino Nano 33 BLE Sense	2
FPGA	2

In contrast to cell phone and cloud platforms, MCUs are generally small (around 1cm^3), low-cost (around \$1) and energy-saving (around 1mW). Therefore, they are the ideal hardware platforms for TinyML. An MCU consists of a CPU, embedded flash (eFlash) memory for code and Static Random Access Memory (SRAM) for data bit, as well as input/output peripherals [18]. Specifically, microcontrollers from the STM32 (32-bit) family, based on ARM Cortex-M processors, support both small projects and end-to-end platforms. In detail, the ARM Cortex-M processors perform a single operation at a time, which are optimized for low-cost, low latency and low power [52]. Additionally, the Apollo3 MCU is designed for ultra-low power and portable, smart devices with an integrated ARM Cortex-M processor [53]. Arduino Nano 33 BLE Sense contains a series of embedded sensors, a Cortex-M4 microcontroller and BLE [50]. Field Programmable Gate Arrays (FPGAs) are semiconductor-integrated circuits that execute all operations in a parallel way [33].

B. Framework

TinyML frameworks are typically used to enable ML models into various MCU-based edge devices [1].

The variety of embedded systems including hardware and software needs to be addressed for TinyML to obtain board understanding [5]. Therefore, we analyzed the framework (software) concerning hardware shown in Fig. 3. From Figure 3, we can find the relation between hardware and the framework (software). For instance, TensorFlow Lite (10 papers) is the most well-known framework (software) adopted in hardware. It is frequently an alternative for the term "TinyML". TensorFlow Lite is Google's open-source machine learning framework that deploys a special format model on mobile and embedded devices [5], such as STM32, Apollo3 and ARM Cortex-M7. The Framework TinyOL is adopted by Arduino Nano 33 BLE Sense. Tiny RespNet TensorFlow and PYNQ are used by FPGA.

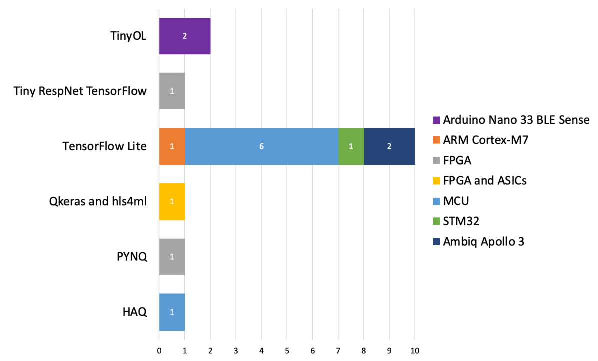

Fig. 3. TinyML frameworks by hardware.

C. Use Cases

Although TinyML is in its infancy, there are a large amount of well-established use cases that apply TinyML in solving real-life issues [4]. In this study, the use cases are keyword spotting, image classification, visual wake words, object detection, anomaly detection, semantic segmentation, motor control, gesture recognition, forecasting, face recognition and activity detection. In addition, we noticed 16 papers that proposed a new tool or improve the current technology without belonging to any use cases type. Therefore, we added the technology improvement/new tools as another new type of use case. Fig. 4 shows an overview of the use cases by categorizing 47 papers (we named the new use case type as "technology improvement/new tools", therefore it is not shown in the Fig. 4).

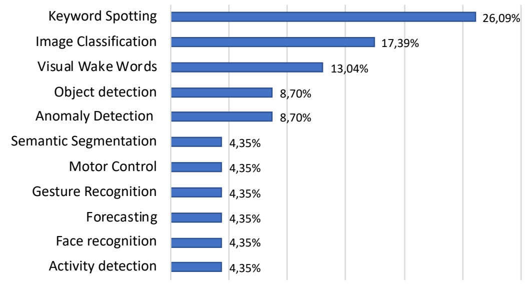

Fig. 4. TinyML use cases.

The dominant use case is keyword spotting with 6 articles which cover 26.09% of TinyML studies. The second use case is image classification, where 17.39% of the studies classified, followed by visual wake words with 13.04%. Object detection and anomaly detection are the same ordered with 8.07% (2 papers). The remaining six use cases are semantic segmentation, motor control, gesture recognition, forecasting, face recognition and activity detection (4.35%).

The most popular and largely deployed case of TinyML is keyword spotting [37]. Keyword spotting is the sensible detection of certain words and short sentences. For example, the initiation of virtual assistants like Siri (Apple), Cortana (Microsoft), and Alexa (Amazon). TinyML can be widely used for keyword spotting because a specific word or phrase identification needs accordingly low power consumption [2].

Anomaly detection is generally deployed on MCUs that divide normal samples from abnormal samples [2]. It has numerous applications such as checking for anomalous audio, temperature or IMU (inertial measurement unit) data to issue early warnings of potential breakdowns [50].

Machine learning inference of various senor data from low-power image sensors, photoplethysmogram (PPG) optical sensors, gyroscope sensors, microphones, accelerometers, and other embedded devices enable market and industrial applications such as image classification, face recognition, object detection, gesture recognition, semantic segmentation, and forecasting [37]. Some use cases have been proven feasible, but have yet to contact consumers as they are too new, like visual wake words [4].

D. Datasets

Many free and public datasets are relevant to TinyML use cases [4]. Fig. 5 shows the relationship between the datasets and TinyML use cases. Speech commands are audio datasets released by Google for training and evaluating keyword spotting systems, which sorts short audio clips into a distinct set of classes [23]. COCO dataset is applied to train, validate and test for visual wake words models [2]. When referring to the application of two datasets or more, we find the use case of image classification commonly employs datasets ImageNet [16] and MNIST [19], [25]. ImageNet is also used in the use case of visual wake words [18]. From Figure 5, we could find the suitable datasets for each use case model evaluation.

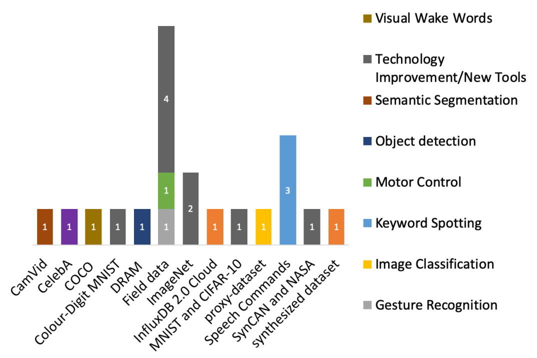

Fig. 5. Existing datasets by TinyML use cases.

Notwithstanding the accessibility of these open-source datasets, most of deployed TinyML models are trained and evaluated on massive datasets. These proprietary datasets that are huge in size are not specific for developing TinyML applications. The lack of suitable TinyML datasets poses a substantial obstacle to the progress of academic research [4].

E. Algorithms/Model

Fig. 6 lists the typical algorithm/model for solving TinyML use cases problems. As we know, neural networks (NN) are the main force for both traditional machine learning and TinyML [15]. In particular, due to low CPU and memory usage, some TinyML use cases can also use non-NN algorithms like random forest [24].

From Fig. 6, we find convolutional neural networks (CNN) are employed in image recognition and computer vision such as TinyML use cases image classification, face recognition and activity detection. Deep neural networks (DNNs) have been used by many TinyML use cases such as visual wake words, keyword spotting and image classification. However, one of the main challenges to use DNNs for solving TinyML use cases is the steadily growing number of parameters (from millions to billions to 1 trillion parameters within the next decade) [13]. In addition, some papers only mentioned NN in general without pointing out exact type of NN (CNN, DNN, RNN) used. Therefore, Fig. 6 displays CNN and DNN and NN at the same time.

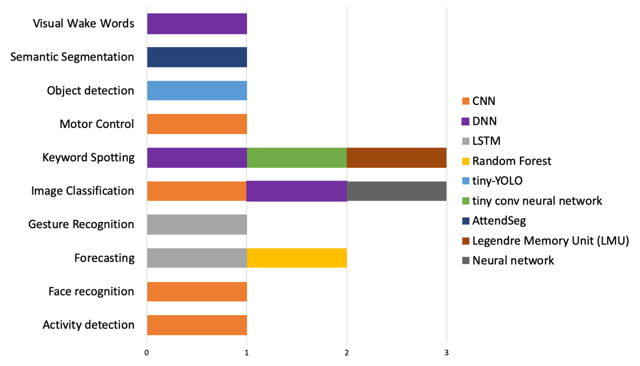

Fig. 6. TinyML algorithm/model by use cases.

Long short-term memory (LSTM) is used for natural language processing such as speech recognition or music generation [54]. Using LSTM particularly with CNN for human gesture recognition and motor control is well documented to obtain a high level of accuracy [14].

V. Conclusion

TinyML is an essential and fast-developing field that requires trade-offs among diverse integral components (hardware, software, machine learning algorithms) [5]. In this paper, we contribute a systematic literature review in reference to the data synthesis results of 47 publications on TinyML since 2019.

We focus on five elements: hardware, framework, datasets, use cases, and algorithms/models. Future studies could add more elements like the TinyML application area (Industry 4.0, vehicular services, smart spaces, smart agriculture and farming, eHealth, etc.).

TinyML has the potential to exploit an entirely new domain of smart applications throughout manufacturing and business and personal life areas [27]. TinyML proposes innovative solutions in different fields and provides novel study directions, which would be a promising area for scholars to explore [5].

Acknowledgment

The research of Dr. Hui Han is funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan Fellowship Programme and the Fraunhofer Institute for Experimental Software Engineering IESE.

References

[1] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices – optimizations and applications," *Sensors*, vol. 21, no. 13, pp. 1–44, 2021.

[2] C. Banbury *et al.*, "MLPerf tiny benchmark," *Conference on Neural Information Processing Systems*, 2021. [Online]. Available: http://arxiv.org/abs/2106.07597.

[3] G. Signoretti, M. Silva, P. Andrade, I. Silva, E. Sisinni, and P. Ferrari, "An evolving tinyml compression algorithm for IOT environments based on data eccentricity," *Sensors*, vol. 21, no. 12, pp. 1–25, 2021.

[4] C. R. Banbury *et al.*, "Benchmarking tinyML systems: challenges and direction," *arXiv preprint arXiv:2003.04821*, 2020. [Online]. Available: http://arxiv.org/abs/2003.04821.

[5] S. Soro, "TinyML for ubiquitous edge AI," 2021.

[6] H. Han, H. Xu, and H. Chen, "Social commerce: A systematic review and data synthesis," *Electron. Commer. Res. Appl.*, vol. 30, pp. 38–50, 2018.

[7] B. Kitchenham, "Procedures for performing systematic reviews," *Keele Univ.*, vol. 33, pp. 1–26, 2004.

[8] C. L. Downey, W. Tahir, R. Randell, J. M. Brown, and D. G. Jayne, "Strengths and limitations of early warning scores: A systematic review and narrative synthesis," *Int. J. Nurs. Stud.*, vol. 76, pp. 106–119, 2017.

[9] K. Krippendorff, *Content analysis: an introduction to its methodology*. SAGE Publications Inc, 2004.

[10] F. Alongi, N. Ghielmetti, D. Pau, F. Terraneo, and W. Fornaciari, "Tiny neural networks for environmental predictions: an integrated approach with Miosix," in *IEEE International Conference on Smart Computing*, 2020, pp. 350–355.

[11] M. Lootus, K. Thakore, S. Leroux, G. Trooskens, A. Sharma, and H. Ly, "A VM / containerized approach for scaling tinyML applications," in *TinyML Research Symposium*, 2021, pp. 1–6.

[12] P. Blouw, G. Malik, B. Morcos, A. R. Voelker, and C. Eliasmith, "Hardware aware training for efficient keyword spotting on general purpose and specialized hardware," in *TinyML Research Symposium*, 2021, pp. 1–5.

[13] S. Ghamari *et al.*, "Quantization-guided training for compact tinyML models," in *TinyML Research Symposium*, 2021.

[14] B. Coffen and M. S. Mahmud, "TinyDL: edge computing and deep learning based real-time hand gesture recognition using wearable sensor," in *IEEE International Conference on E-Health Networking, Application and Services*, 2020, pp. 1–6.

[15] G. Crocioni, G. Gruosso, D. Pau, D. Denaro, L. Zambrano, and G. di Giore, *Characterization of neural networks automatically mapped on automotive-grade microcontrollers*. Association for Computing Machinery, 2021.

[16] S. Disabato and M. Roveri, "Incremental on-Device tiny machine learning," in *International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, 2020, pp. 7–13.

[17] H. Doyu, R. Morabito, and M. Brachmann, "A tinyMLaaS ecosystem for machine learning in IoT: overview and research challenges," in *International Symposium on VLSI Design, Automation and Test*, 2021, pp. 1–6.

[18] C. Banbury *et al.*, "Micronets: neural network architectures for deploying tinyml applications on commodity microcontrollers," in *Proceedings of Machine Learning and Systems*, 2021, pp. 1–16.

[19] F. Fahim *et al.*, "hls4ml: an open-source codesign workflow to empower scientific low-power machine learning devices," in *TinyML Research Symposium*, 2021, pp. 1–10.

[20] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.

[21] J. Kwon and D. Park, "Toward data-adaptable tinyML using model partial replacement for resource frugal edge device," in *International Conference on High Performance Computing in Asia-Pacific Region*, 2021, pp. 133–135.

[22] H. F. Langroudi, V. Karia, T. Pandit, and D. Kudithipudi, "TENT: efficient quantization of neural networks on the tiny edge with Tapered FixEd PoiNT," in *TinyML Research Symposium*, 2021, pp. 1–8.

[23] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, "MCUNet: tiny deep learning on IoT devices," in *Conference on Neural Information Processing Systems*, 2020, pp. 1–15.

[24] A. Navaas Roshan, B. Gokulapriyan, C. Siddarth, and P. Kokil, "Adaptive traffic control with tinyML," in *International Conference on Wireless Communications, Signal Processing and Networking*, 2021, pp. 451–455.

[25] A. J. Paul, P. Mohan, and S. Sehgal, "Rethinking generalization in american sign language prediction for edge devices with extremely low memory footprint," in *IEEE Recent Advances in Intelligent Computational Systems*, 2020, pp. 147–152.

[26] H. Ren, D. Anicic, and T. A. Runkler, "The synergy of complex event processing and tiny machine learning in industrial IoT," in *ACM International Conference on Distributed and Event-based Systems*, 2021, pp. 126–135.

[27] B. Sudharsan *et al.*, "TinyML benchmark: executing fully connected neural networks on commodity microcontrollers," in *IEEE World Forum on Internet of Things*, pp. 19–21.

[28] F. Svoboda *et al.*, "Resource efficient deep reinforcement learning for acutely constrained tinyML devices," in *TinyML Research Symposium*, 2021, pp. 1–8.

[29] C. Toma, M. Popa, and M. Doinea, "A.I. neural networks inference into the IoT embedded devices using tinyml for pattern detection within a security system," in *International Conference on Informatics in Economy Education, Research and Business Technologies*, 2020, pp. 14–22.

[30] C. Vuppalapati, A. Ilapakurti, K. Chillara, S. Kedari, and V. Mamidi,

"Automating tiny ML intelligent sensors DevOPS using microsoft Azure," in *IEEE International Conference on Big Data*, 2020, pp. 2375–2384.

[31] C. Vuppalapati, A. Ilapakurti, S. Kedari, J. Vuppalapati, S. Kedari, and R. Vuppalapati, "Democratization of AI, albeit constrained IoT devices & Tiny ML, for creating a sustainable food future," in *International Conference on Information and Computer Technologies*, 2020, pp. 525–530.

[32] S. Siddiqui, C. Kyrkou, and T. Theocharides, "Mini-NAS: a neural architecture search framework for small scale image classification applications," in *TinyML Research Symposium*, 2021, pp. 1–8.

[33] B. Jiao *et al.*, "A 0 . 57-GOPS / DSP object detection PIM accelerator on FPGA," in *Asia and South Pacific Design Automation Conference*, 2021, pp. 13–14.

[34] H.-A. Rashid, H. Ren, A. N. Mazumder, and T. Mohsenin, "Tiny RespNet: a scalable multimodal tinyCNN processor for automatic detection of respiratory symptoms," in *TinyML Research Symposium*, 2021, pp. 1–8.

[35] X. Wen, M. Famouri, A. Hryniowski, and A. Wong, "AttendSeg: a tiny attention condenser neural network for semantic segmentation on the edge," *arXiv preprint arXiv:2104.14623*, 2021. [Online]. Available: http://arxiv.org/abs/2104.14623.

[36] A. Capotondi, M. Rusci, M. Fariselli, and L. Benini, "CMix-NN: mixed low-precision CNN Library for Memory-Constrained Edge Devices," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 5, pp. 871–875, 2020.

[37] R. David *et al.*, "TensorFlow Lite Micro: embedded machine learning on tinyml systems," *arXiv preprint arXiv:2010.08678*, 2020. [Online]. Available: http://arxiv.org/abs/2010.08678.

[38] M. de Prado *et al.*, "Robustifying the deployment of tinyML models for autonomous mini-vehicles," *Sensors*, vol. 21, no. 1339, pp. 1–16, 2021.

[39] L. Heim, A. Biri, Z. Qu, and L. Thiele, "Measuring what really matters: optimizing neural networks for tinyML," *arXiv preprint arXiv:2104.10645*, 2021. [Online]. Available: http://arxiv.org/abs/2104.10645.

[40] H. Ren, D. Anicic, and T. Runkler, "TinyOL: tinyML with online-learning on microcontrollers," *arXiv preprint arXiv:2103.08295*, 2021. [Online]. Available: http://arxiv.org/abs/2103.08295.

[41] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-enabled frugal smart objects: challenges and opportunities," *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 4–18, 2020.

[42] A. Wong, M. Famouri, and M. J. Shafiee, "AttendNets: tiny deep image recognition neural networks for the edge via visual attention condensers," *arXiv preprint arXiv:2009.14385*, 2020. [Online]. Available: http://arxiv.org/abs/2009.14385.

[43] M. Z. H. Zim and B. Dhaka, "TinyML: analysis of Xtensa LX6 microprocessor for neural network applications by ESP32 SoC," *arXiv preprint arXiv:2106.10652*, 2021. [Online]. Available: http://arxiv.org/abs/2106.10652%0Ahttp://dx.doi.org/10.13140/RG.2.2.28602.11204.

[44] C. Campolo, G. Genovese, A. Iera, and A. Molinaro, "Virtualizing AI at the distributed edge towards intelligent IOT applications," *J. Sens. Actuator Networks*, vol. 10, no. 1, pp. 1–14, 2021.

[45] H. Miao and F. X. Lin, "Enabling large NNs on tiny MCUs with swapping," *preprint arXiv:2101.08744*, 2021. [Online]. Available: http://arxiv.org/abs/2101.08744.

[46] A. Wong, M. Famouri, M. Pavlova, and S. Surana, "TinySpeech: attention condensers for deep speech recognition neural networks on edge devices," *arXiv preprint arXiv:2008.04245*, 2020. [Online]. Available: http://arxiv.org/abs/2008.04245.

[47] P. Mohan, A. J. Paul, and A. Chirania, "A tiny cnn architecture for medical face mask detection for resource-constrained endpoints," in *Innovations in Electrical and Electronic Engineering*, vol. 756, 2020, pp. 657–670.

[48] P. Warden and D. Situnayake, *TinyML: machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers*. 2019.

[49] M. Rusci, M. Fariselli, A. Capotondi, and L. Benini, "Leveraging automated mixed-low-precision quantization for tiny edge microcontrollers," in *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*, 2020, pp. 296–308.

[50] V. J. Reddi *et al.*, "Widening access to applied machine learning with tinyML," 2021.

[51] H. Doyu, E. Research, R. Morabito, and J. Höller, "Bringing machine learning to the deepest IoT edge with tinyML as-a-service," *IEEE IoT Newsletter*, pp. 1–4, 2020.

[52] Cortex-M, "Arm Cortex-M series processors," *ARM Developer*, 2021. [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-m.

[53] Ambiq, "Apollo3 Blue MCU," *Ambiq*, 2021. [Online]. Available: https://www.ambiq.top/en/apollo3-blue-mcu-eval-board.

[54] R. Jain, V. B. Semwal, and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Syst.*, pp. 1–17, 2021.