SCIENCE CHINA
Information Sciences

• LETTER •

April 2024, Vol. 67, Iss. 4, 149402:1–149402:2
https://doi.org/10.1007/s11432-023-3934-y

An analysis of TinyML@ICCAD for implementing AI on low-power microprocessor

Guoqing LI[1,2,4*†], Jingwei ZHANG[2†], Meng ZHANG[2*], Tuo LI[1,4], Tinghuan CHEN[3] & Jun YANG[2]

[1]*Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd., Jinan* 250013*, China;*
[2]*School of Integrated Circuits, Southeast University, Nanjing* 210096*, China;*
[3]*School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen* 518172*, China;*
[4]*Shandong Inspur Artificial Intelligence Research Institute Co., Ltd., Jinan* 250013*, China*

Received 28 September 2023/Revised 4 January 2024/Accepted 17 January 2024/Published online 25 March 2024

Tiny machine learning (ML) is an important application area of artificial intelligence (AI), which focuses on deploying AI on small hardware platforms with constrained resources [1, 2]. The high-level TinyML contests at high-level conferences promote the development of TinyML areas, such as DAC-SDC [3] and TinyML@ICCAD [4]. In 2022, the first TinyML Design Contest was held in conjunction with the International Conference on Computer-Aided Design (ICCAD). This multi-month competition focused on addressing real-world problems necessitating the implementation of machine learning algorithms on low-end microprocessors. Our SEUer team secured the 2nd place. An integral feature of this challenge was the provision of a standardized low-end microprocessor platform, enabling participants to develop and benchmark state-of-the-art algorithms. This study analyzes and discusses the methods developed by TOP-8 entries as well as representative results.

Contest introduction. The TinyML@ICCAD'22 tasked participants with crafting an open-source AI/ML algorithm for automated ventricular arrhythmias (VAs) detection from intracardiac electrogram (IEGM) recordings, while ensuring compatibility with the designated microprocessors [5].

The stipulated development board for the contest is the NUCLEO-L432KC, equipped with an ARM Cortex-M4 core running at 80 MHz, boasting 258 Kbytes of Flash memory, along with 64 Kbytes of SRAM. It supports STM32 X-Cube-AI, constituting an integral part of the STM32Cube Expansion Package.

The training dataset and validation dataset comprise 24588 and 5625 IEGM recordings, respectively. The hidden test dataset is reserved for the official evaluation conducted exclusively by contest organizers. Each recording spans 5 s, sampled at a rate of 250 Hz. Pre-processing involves applying a band-pass filter with a pass-band frequency of 15 Hz and a stop-band frequency of 55 Hz. There are 8 categories in the datasets and 3 categories are VAs.

The submitted designs are assessed using a comprehensive evaluation metric that considers inference latency score (L_n), memory usage score (M_n), and detection performance score (F_β), which are defined as follows:

$$F_\beta = (1+\beta^2) \times \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precison}) + \text{Recall}}, \tag{1}$$

where $\beta = 2$, this weighting places higher importance on recall, given that accurately detecting life-threatening VAs is paramount.

$$L_n = 1 - \frac{L - \text{Min}_L}{\text{Max}_L - \text{Min}_L}, \tag{2}$$

where L is the actual inference latency, $\text{Min}_L = 1$ ms, and $\text{Max}_L = 200$ ms.

$$M_n = 1 - \frac{M - \text{Min}_M}{\text{Max}_M - \text{Min}_M}, \tag{3}$$

where M is the sum of RW Data, RO Data, and Code, as reported by Keil during the project compilation, $\text{Min}_M = 5$ KiB, and $\text{Max}_M = 256$ KiB. The final score is calculated by

$$\text{Score} = 100F_\beta + 20L_n + 20M_n. \tag{4}$$

It can be found that F_β score occupies a large proportion of the final score. Detection performance is more important for comprehensive performance.

TOP-8 designs. There were a total of 41 out of more than 150 participating teams that successfully implemented their designs on the hardware platform provided. The TOP-8 teams have generously shared their source codes, contributing to the collaborative spirit of the competition.

The Gatech-EIC-Lab team from Georgia Institute of Technology obtains the championship. They propose a convolutional neural network (CNN) consisting of 1 convolutional layer with a very large kernel (size = 85) and 3 fully connected (FC) layers. The convolutional layer implements the channel expansion ($1 \to 3$) and down-sampling ($1250 \to 37$). The output sizes of the 1st and 2nd FC layers are 20 and 10, respectively. The output size of the last FC layer is 2 for binary classification.

The SEUer team from Southeast University in China obtains the 2nd place. They propose a CNN consisting of 3 convolutional layers and 2 FC layers. The mean pooling

* Corresponding author (email: liguoqing.aicer@gmail.com, zmeng@seu.edu.cn)
† Li G Q and Zhang J W have the same contribution to this work.

Table 1 Algorithms, parameters, MAdds, and comprehensive performance of TOP-8 designs[a)]

Rank & team	Algorithm	Layer	Parameter	MAdds	Final score	F_β	Latency (ms)	Flash (KiB)
1 Gatech-EIC-Lab	CNN	1Conv+3FC	2695	11875	**135.43187**	0.972	1.747	26.39
2 SEUer	CNN	3Conv+2FC	892	**5068**	132.98377	0.946	1.712	24.48
3 MIT-HAN-Lab	DB	–	–	–	132.91372	0.934	0.538	**11.18**
4 HuskyCS-Deepical	CNN	5Conv+1FC	1442	56756	132.84182	**0.978**	26.197	35.46
5 UBPercept	DT	–	–	–	132.21299	0.93	**0.221**	16.40
6 MAD-AI	CNN	5Conv+2FC	**442**	26860	131.9082	0.953	17.745	26.81
7 SDUAES	CNN	2Conv+1FC	1944	48924	131.59601	0.955	21.879	27.78
8 VIPS4Lab@UNIVR	CNN	1Conv+2FC	6897	9770	130.3599	0.945	4.843	51.98

a) The best is in bold.

(1250 → 625) is used to reduce parameters and MAdds before the 1st convolutional layer. The convolutional layers implement the channel expansion (1 → 2 → 4 → 8) and down-sampling (625 → 103 → 20 → 4) by middle stride size (6, 5, 4). The output size of the 1st FC layer is 16, and the output size of the 2nd FC layer is 2 for classification.

The MIT-HAN-Lab team from Massachusetts Institute of Technology gets the 3rd place. Different from Gatech-EIC-Lab and SEUer, they use a traditional ML method. They suppose that the dataset is linearly separable, and the dataset cloud be classified by decision boundary (DB). The key is to extract the number of peaks from the data points, and then classify the VAs and non-VAs by this feature. They utilize the Bayesian search feature offered by Weights & Biases Sweeps to improve the accuracy of the proposed DB.

The HuskyCS-Deepical team from University of Connecticut & The University of Texas gets the 4th place. They propose a CNN consisting of 5 convolutional layers and 1 FC layer. The UBPercept team from University at Buffalo gets the 5th place. Like MIT-HAN-Lab, they also use a traditional ML method decision tree (DT) to classify the VAs and non-VAs. The MAD-AI team from Politecnico di Torino in Italy gets the 6th place. They propose a CNN consisting of 5 convolutional layers and 2 FC layers. The SDUAES team from Shandong University in China gets the 7th place. They propose a CNN consisting of 2 convolutional layers and 1 FC layer. VIPS4Lab@UNIVR from University of Verona gets the 8th place. They propose a CNN consisting of 1 dilated convolutional layer for a large local receptive field and 2 FC layers.

Results and analysis. In the TOP-8 entries, 6 teams selected the CNN of deep learning (DL) method and only 2 teams selected the traditional ML method. It is shown that DL is becoming more and more popular for VA detection. As shown in Table 1, the optimal F_β, latency, and Flash occupation are 0.978, 0.221 ms, and 11.18 KiB, respectively. Note that all the TOP-3 entries get high F_β, low latency, and low flash occupation. Although F_β plays the most important role in the final score, only high F_β is not enough to achieve the championship. For example, HuskyCS-Deepical achieves the highest $F_\beta = 0.978$ but a large latency (larger than 20 ms), which only ranks the 4th. Furthermore, achieving either low latency or low Flash occupation alone does not guarantee a high final score or a favorable ranking. For instance, the entry Xtreme-XNOR obtains a rather low latency (0.242 ms) and low Flash occupation (16.29 KiB) but a low $F_\beta = 0.492$, which only ranks the 39th. Actually, most entries get F_β higher than 0.94, latency lower than 20 ms, and Flash occupation lower than 30 KiB in the TOP-8 teams. Moreover, their final scores are very close, which indicates fierce competition among the top entries. In addition, it can be found traditional ML methods obtain lower latency and lower Flash occupation, but cannot obtain high F_β (only about 0.93). The DL methods obtain a higher F_β score than traditional ML methods, but they have more parameters and MAdds.

The AI/ML algorithms designed by entries are deployed on the NUCLEO-L432KC after compiling by X-Cube-AI and MDK-ARM. According to our experimental results, good optimization options can reduce more than 60% Flash occupation. For example, the SEUer's design occupies 82.77 KiB Flash before optimization and only occupies 24.48 KiB Flash after optimization. More memory access leads to lower inference speed. Good compiler and compilation options are important for the low latency and low Flash occupation.

Lessons. Both deep learning and traditional machine learning can achieve good performance for VA detection. DL methods tend to offer higher accuracy and robustness, while traditional machine learning methods exhibit lower latency and a smaller memory footprint. The advanced hardware and compilation optimization will affect deployment performance. It is worthwhile to try various optimization strategies for low latency and low memory occupation.

Conclusion. In this work, the first ACM/IEEE TinyML Contest at ICCAD for VA detection on low-end microprocessors is reviewed and analyzed. The contest task is introduced. The open-source TOP-8 designs and their comprehensive performance are introduced and analyzed. This contest and its findings provide valuable knowledge and inspiration for researchers and engineers working in the field of TinyML, especially in the context of VA detection and healthcare applications.

Acknowledgements This work was partly supported by Key R&D Program of Guangdong Province (Grant No. 2021B11-01270006), National Key R&D Program of China (Grant No. 2023YFB4402900), Shandong Provincial Natural Science Foundation (Grant Nos. ZR2023QF056, ZR2023QF050), Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX22_0051), and National Natural Science Foundation of China (Grant No. 62304197).

References

1 Lin J, Zhu L, Chen W M, et al. Tiny machine learning: progress and futures [feature]. IEEE Circ Syst Mag, 2023, 23: 8–34
2 Chen S Y, Cheng T H, Fang J M, et al. TinyDet: accurately detecting small objects within 1 GFLOPs. Sci China Inf Sci, 2023, 66: 119102
3 Jia Z, Xu X, Hu J, et al. Low-power object-detection challenge on unmanned aerial vehicles. Nat Mach Intell, 2022, 4: 1265–1266
4 Jia Z, Li D, Xu X, et al. Life-threatening ventricular arrhythmia detection challenge in implantable cardioverter-defibrillators. Nat Mach Intell, 2023, 5: 554–555
5 Jia Z, Li D, Ping L, et al. 2022 ACM/IEEE TinyML Design Contest at ICCAD. https://tinymlcontest.github.io/TinyML-Design-Contest