Lightweight CNN hardware accelerator using the ternary quantization method for fault diagnosis of CNC machinery

Ching-Che Chung, *Senior Member*, *IEEE*, Yu-Pei Liang, *Member*, *IEEE*, and Jo-Chen Huang
Department of Computer Science and Information Engineering
National Chung Cheng University
No. 168, University Rd., Min-Hsiung, Chia-Yi, Taiwan
Email: wildwolf@cs.ccu.edu.tw, ypliang@cs.ccu.edu.tw

***Abstract*—In the context of computer numerical control (CNC) machinery, fault diagnosis traditionally involves complex formula conversions to extract characteristics and categorize faults. However, such a method is unsuitable for hardware implementation due to high resource usage. This paper proposes a convolution neural network (CNN) approach for fault classification and hardware acceleration using ternary quantization and batch normalization techniques to reduce data access for weights and improve accuracy. The proposed CNN hardware accelerator is implemented on FPGA (VC707) and reduces memory usage by 83.8% compared to floating-point operations. Furthermore, the proposed method achieves 97.6% accuracy in CNC machinery fault classification.**

Keywords—Ternary quantization, Convolution neural network (CNN), Hardware accelerator, Fault diagnosis

I. Introduction

The popularity of computer numerical control (CNC) machines in product processing are due to Industry 4.0 and advances in science and technology. While CNC machines are suitable for processing precise parts based on operator input, a long-term operation may damage bearing parts. To detect CNC machine health and bearing faults, installing accelerometers to collect vibrational signals for analysis is a viable solution. Deep learning neural networks (DNNs) have been applied to this research topic, including autoencoders [1], restricted Boltzmann machines (RBMs), deep belief networks (DBNs) [2], convolutional neural networks (CNNs) [3,4], and multilayer perceptron (MLP) neural networks [5]. However, previous research used complex preprocessing methods such as RMS, variance, and Fourier transform, which require additional computing time and are challenging to implement as hardware accelerators.

The methods mentioned above are limited to software solutions only. Previous work required accelerometer data for detecting bearing faults of CNC machinery to be collected and analyzed in an offline software program, which could increase detection time and sacrifice real-time capabilities. Therefore, integrating the solution into a hardware accelerator and installing it directly on the CNC machinery is better for achieving real-time fault diagnosis. No previous research has been dedicated to optimizing CNN hardware for CNC machinery fault detection. Convolution operations can be optimized in various ways, including memory space, computation latency, and workload distribution in processing elements (PEs).

This paper proposes a lightweight CNN network for detecting faults in CNC machinery, which is implemented on FPGA to achieve real-time fault detection. The proposed method uses a simple data preprocessing approach. According to evaluations, the method can diagnose at least once in under 1 second, significantly reducing labor costs and achieving real-time bearing fault diagnosis. The rest of this paper is organized as follows: Section II presents the proposed design. Then the trade-off in hardware implementation will be discussed. Finally, the hardware implementation detail and experimental results are shown in Section III, followed by Section IV's conclusion.

II. Proposed Design

This paper trains the model using bearing data from CWRU [12] and TensorFlow. During experimentation, the accuracy of training was found to be influenced by the sampling rate and sampling window of input data. The bearing fault types include one normal case and nine bearing faults with different damage diameters for ball, inner, and outer race faults.

TABLE I. THE ARCHITECTURE OF THE PROPOSED TERNARY CNN MODEL. (CNN MODULE: FILTER PARAMETER (HEIGHT × WEIGHT × OUT)

Layer name	CNN Modules	Output data size
Layer 1	Conv(3×3×32)	32×32×32
Layer 2	Maxpool(2×2)	16×16×32
Layer 3	Conv(3×3×32)	16×16×32
Layer 4	Maxpool(2×2)	8×8×32
Layer 5	Conv(3×3×32)	8×8×32
Layer 6	Maxpool(2×2)	4×4×32
Layer 7	Conv(3×3×16)	4×4×16
Layer 8	Maxpool(2×2)	2×2×16
Layer 9	Fc(64,16)	64
	Fc.ba(16,16)	16
Layer 10	Fc(16,10)	10

Table I shows the proposed ternary CNN network model for fault diagnosis that was tested multiple times. Input data size impacts subsequent operations and accuracy, with higher input data size resulting in more operations and lower input data size leading to lower accuracy. An increase in trained parameters leads to more storage and calculations, and the number of output channels generally increases with the number of layers. However, hardware implementation should consider keeping

This work was supported in part by the National Science and Technology Council of Taiwan under Grant MOST-111-2221-E-194-049-.

2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan) | 979-8-3503-2417-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCE-Taiwan58799.2023.10227058

channels low for better hardware controller design. The fourth convolutional layer intentionally decreases the number of channels, and reducing the input data from Layer 8 to Layer 9 further helps reduce the computational complexity in Layer 9.

Quantization reduces the bit width of parameters to improve performance and reduce hardware overhead. Three common quantization methods include Binary Weight Network (BWN) [6], DoReFa-Net [7], and Ternary Weight Network (TWN) [8]. Previous works [8,9] show that TWN has lower error rates than DoReFa-Net and BWN; therefore, this work uses TWN as the quantization method. Batch normalization [10] is also included in the proposed design to further optimize the network, with one batch normalization layer (Fc.ba) added after Layer 9.

The decimal bits of input sensor data to be retained during hardware design are defined first. It is acceptable to reserve four digits in the decimal bits of input data, and the accuracy is still accepted. Next, the number of decimal bits for the lookup table for weights is set to 9 after considering the influence on accuracy. For each convolution layer, changing the number of decimal bits of the output value from 5 to 6 significantly impacts accuracy. Therefore, it was decided to reserve 6 bits for the fractional part of the output value of each convolution layer. Finally, the final selection of decimal bits for hardware implementation is based on multiple tests.

III. Experimental Results

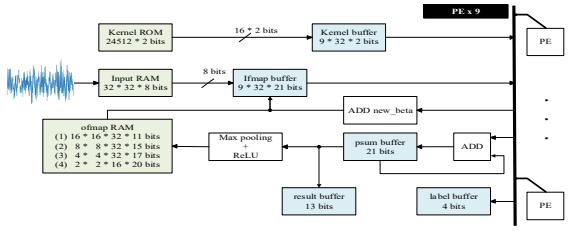

Fig. 1. Overall block diagram of the hardware architecture.

Fig. 1 illustrates the overall block diagram of the proposed hardware accelerator. The two main memories are for the kernel and the input feature (IF) map. The block memory includes kernel, input data, and output feature (OF) map. Among them, the ofmap RAM will be further divided into four memory pieces, namely ofmap_ram1, ofmap_ram2, ofmap_ram3, and ofmap_ram4. In the convolution calculation process, at least two memory pieces are required to store input and output data. In this architecture, using only two memory pieces for sharing is not cost-effective. Therefore, it was decided to use four ofmap RAMs in the final design.

TABLE II. THE TOTAL USED MEMORY ON THE FPGA.

On-chip memory	Float32 Memory(KB)	Fixed point Memory(KB)	Reduction ratio
Input RAM	32	8	75%
Kernel ROM	766	47.875	93.75%
Ofmap RAM	338	127.75	62.2%
Total size	1136	183.625	83.8%

Table II demonstrates that the proposed CNN hardware accelerator occupies only 183.625 Kb memory when using fixed-point arithmetic. Compared to conventional neural networks that use floating-point operations, the proposed method can reduce total memory usage by 83.8%. The input RAM is reduced because the input data is expressed in an 8-bit fixed-point format, resulting in a 75% reduction of memory space compared to floating-point numbers. Kernel ROM uses a lookup table method for TWN quantization, enabling weights to be stored in memory using 2 bits. This method is 16 times better than the method without a lookup table. The reduction of Ofmap RAM is due to the fixed-point operation of the proposed CNN hardware accelerator, and the output bit numbers in each layer output are 11, 15, 17, and 20 bits, respectively.

TABLE III. COMPARISON TABLE

	This work		[3]	[11]	[8]		
Software platform	Tensorflow		Tensorflow	N/A	Caffe		
Dataset	CWRU	MNIST	CWRU	CWRU	MNIST		
Data pre-processing	Only discard decimal point bit	No	Signal-to image conversion	Transfer learning	No		
Architecture	CNN		CNN	CNN	LeNet-5		
Parameters (k)	24.5		100,648	N/A	1,256		
Quantization method	Ternary Network		No	No	Ternary Network	Binary Connect	Binary Neural Network
Number of identifiable categories	10		5	6	10		
Accuracy	97.6%	98.84%	99.79%	91.8%	99.35%	98.82%	88.6%

Table III demonstrates that the proposed design achieves the lowest parameter number while maintaining acceptable accuracy. The proposed CNN network requires fixed-point operations and does not need complex data preprocessing for the CWRU dataset. The MNIST dataset is used for comparison, and the proposed CNN network can also handle MNIST dataset with 98.84% accuracy. This paper successfully applies the quantization method to the CWRU dataset, balancing accuracy and hardware cost.

IV. Conclusion

This paper presents a lightweight CNN hardware accelerator for diagnosing bearing faults in CNC machinery. The proposed accelerator can process vibrational sensor data in real time, providing an efficient solution for detecting bearing faults and can achieve 97.6% accuracy.

References

[1] Rui Zhao, et al., "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, Jan. 2019.

[2] Siyu Shao, et al., "Learning features from vibration signals for induction motor fault diagnosis," *in Proc. ISFA*, pp. 71-76, Aug. 2016.

[3] Long Wen, et al., "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Industrial Electronics*, vol. 65, no. 7, pp. 5990-5998, Jul. 2018.

[4] Shaobo Li, et al., "An ensemble deep convolutional neural network model with improved DS evidence fusion for bearing fault diagnosis," *Sensors*, vol. 17, no. 8, 1729, Jul. 2017.

[5] Hua Su and Kil To Chong, "Induction machine condition monitoring using neural network modeling," *IEEE Trans. Industrial Electronics*, vol. 54, no. 1, pp. 241-249, Feb. 2017.

[6] Matthieu Courbariaux, et al., "Binarized Neural Networks: Training deep neural networks with weights and activations constrained to +1 or -1, arXiv:1602.02830v3 [cs.LG]," *arXiv.org*, Mar. 2016.

[7] Shuchang Zhou, et al., "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients, arXiv:1606.06160v3 [cs.NE]," *arXiv.org*, Feb. 2018.

[8] Fengfu Li, et al., "Ternary weight networks, arXiv:1605.04711v2 [cs.CV]," *arXiv.org*, Nov. 2016.

[9] Chenzhuo Zhu, et al., "Trained Ternary Quantization, arXiv:1612.01064v3 [cs.LG]," *arXiv.org*, Feb. 2017.

[10] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167v3 [cs.LG]," *arXiv.org*, Mar. 2015.

[11] Ran Zhang, et al., "Transfer learning with neural networks for bearing fault diagnosis in changing working conditions," *IEEE Access*, vol. 5, pp. 14347-14357, Jun. 2017.

[12] Case Western Reserve University Bearing Data Center Website (http://csegroups.case.edu/bearingdatacenter/download-data-file).