PulseDL-II: A System-on-Chip Neural Network Accelerator for Timing and Energy Extraction of Nuclear Detector Signals

Pengcheng Ai, Zhi Deng, Senior Member, IEEE, Yi Wang, Hui Gong, Xinchi Ran, and Zijian Lang

Abstract—Front-end electronics (FEEs) equipped with high-speed digitizers are being used and proposed for future nuclear detectors. Recent literature reveals that deep learning models, especially 1-D convolutional neural networks (NNs), are promising when dealing with digital signals from nuclear detectors. Simulations and experiments demonstrate the satisfactory accuracy and additional benefits of NNs in this area. However, specific hardware accelerating such models for online operations still needs to be studied. In this work, we introduce PulseDL-II, a system-on-chip (SoC) specially designed for applications of event feature (time, energy, and so on) extraction from pulses with deep learning. Based on the previous version, PulseDL-II incorporates an instruction set computer (RISC) CPU into the system structure for better functional flexibility and integrity. The NN accelerator in the SoC adopts a three-level (arithmetic unit (AU), processing element (PE), and NN) hierarchical architecture and facilitates parameter optimization of the digital design. Furthermore, we devise a quantization scheme compatible with deep learning frameworks (e.g., TensorFlow) within a selected subset of layer types. We validate the correct operations of PulseDL-II on field programmable gate arrays (FPGAs) alone and with an experimental setup comprising a direct digital synthesis (DDS) and analog-to-digital converters (ADCs). The proposed system achieved 60-ps time resolution and 0.40% energy resolution at a signal-to-noise ratio (SNR) of 47.4 dB.

Index Terms—Deep learning, feature extraction, field-programmable gate array (FPGA), front-end electronics (FEEs), model quantization, neural network (NN) accelerator, system-on-chip (SoC).

I. INTRODUCTION

Advances in electronic design and manufacturing technology have greatly influenced the methodology and sensitivity of nuclear detectors. High-speed digitizers [1] with associated processing and storage circuitry revolutionized the front-end electronics (FEEs), especially in performance-critical conditions. For system designers, however, the evolution is a double-edged sword: for one thing, it provides an abundance of raw data for experimenting various feature extraction algorithms (both traditional and intelligent); for another, it poses considerable pressure on the readout system, especially hardware and software capacities within the data link. It is an important issue to maximize the benefits of signal sampling systems while keeping the data throughput controllable and the overall complexity acceptable.

Deep learning [2], essentially multilayer neural networks (NNs) with delicate structures, developed dramatically in the 2010s as a frontier in machine learning and artificial intelligence. Big data from nuclear detectors made it possible to apply this novel method to many tasks directly related to detector signals. For example, in high energy physics (HEP), deep NNs have been applied for particle identification [3], particle discrimination [4], and also low-level triggers [5]. In nuclear medicine, convolutional NNs have been used for accurate estimation of time-of-flight (ToF) in positron emission tomography detectors [6], [7]. In plasma physics, deep variational autoencoders have been utilized to integrate diagnostic data of soft X-ray images [8]. Recently, additional merits and understandings about NNs in nuclear electronics have been explored, such as estimation of heterogeneous uncertainty of nuclear detector signals [9] and computation of the Cramér–Rao lower bound of timing to find out limits for NNs [10].

The needs of pushing NNs to front end arise from the trade-off between data transmission and data processing. For HEP experiments, there are some low-level tasks with online processing demand and stringent latency requirements. In [11], an application-specific integrated circuit (ASIC) is developed for accelerating the inference of a compact autoencoder model to ease data compression for the high-granularity endcap calorimeter. In [5], a dense NN on a field programmable gate array (FPGA) is designed for level-1 trigger directly generated upon raw data. Since the data interfaces and targets in these applications are well defined, a brute-force approach is adopted to implement the network model. In this approach, only weights in the NN can be adjusted; the network architecture is controllable and the overall complexity acceptable.

Beyond applications above, there are also some scenarios where we intend to fulfill the advantage of NNs while keeping more flexibility. NN accelerators based on the array...
of PEs serve this purpose [12], [13]. However, the operation of PE-based accelerators relies on the transactions between accelerators and a host processor. This is easily available in sophisticated computer systems, but not in HEP FEEs. A system-on-chip (SoC) digital design with microcontrollers/microprocessors integrated can efficiently schedule transactions related to the NN accelerator and, thus, make the system autonomous and independent. The SoC NN accelerator is among a future upgrade of FEE for the Nuclotron-based Ion Collider fAcility (NICA)-multipurpose detector (MPD) experiment (Section II).

In the following parts of this article, we will recursively discuss three elements of design perspectives, as shown in Fig. 1. They are the following.

1) **Nuclear Electronics (α):** Readout system and signal features.
2) **NN (β):** Architectural research and network training.
3) **Digital Design (γ):** NN accelerator and SoC.

The three independent elements will produce several intersections, i.e., application training, hardware mapping, system prototype, and joint validation. We will use the same notations (α, β, and γ) to relate each section to associated topics.

### II. ECAL at NICA-MPD (αγ)

The MPD at the NICA collider is designed for protons/heavy ions collisions to study the basic quantum chromodynamics structure of matter [14]. The ECAL in this detector and its FEE are described in [15]. More recent developments and beam tests of ECAL can be found in [16] and [17]. Silicon photomultipliers coupled to the shashlik-type calorimeter were used to transform scintillation light into current pulses, which would later be transmitted through flat cables. A 64-channel 12-bit commercial analog-to-digital converter (ADC) board with 62.5-MHz sampling rate digitized the analog signals and sent waveform samples to back end through optical fibers. The current FEE reached ≈1-ns timing resolution and 6% energy resolution in experimental measurements.

Reaching sub-ns time resolution (i.e., on the order of 150 ps) by ECAL is of great significance for NICA-MPD. With this time resolution, ECAL can work in the ToF mode for auxiliary time measurements as an important supplement to the main ToF detector. However, the current FEE is insufficient to support such precision for timing. Real-time transmission of raw ADC samples brings about high data bandwidth and power consumption (about 250 mW/channel). This in return limits the permissible sampling rate of ADC and prevents the FEE from fully realizing the potential resolution.

In our previous work [10], we have demonstrated that NNs can effectively achieve near-optimal resolution for feature extraction in the nuclear detector dataflow. Deploying NNs at FEE is the key point to ensure performance while substantially reducing bandwidth and power consumption. Although commercial NN accelerators might be a solution to this problem, they do not actually fit into the particular needs of the application scenario. More importantly, they are not efficient in accelerating 1-D convolutional NNs (Section III), which are the target workload for nuclear pulse signals.

Therefore, in future upgrade plans of ECAL, digital logic accelerating NN inferences is proposed to be integrated into FEE for extraction of signal features (time and energy), along with high-speed preamplifiers and ADCs (≈200 MHz). To the best of our knowledge, this is the first implementation of the SoC NN accelerator for low-level edge-intelligence tasks in HEP experiments. The work reported in this article is aimed at, but not restricted to, application of the NN accelerator for ECAL at NICA-MPD.

### III. Signal Feature Extraction With NN (αβγ)

#### A. Building Blocks of Network Structure

Deep learning models are evolving at a tremendous pace. New elements and architectures are frequently coming out. Here, we only focus on 1-D convolutional NNs [10], [18], because they not only fit into the dimensionality of the problem, but also succeed in many machine learning tasks and facilitate parallel computing.

We select four representative building blocks: 1-D convolution layer, 1-D deconvolution (or transpose convolution) layer, fully connected layer, and nonlinear activation (such as ReLU [19]). Ai et al. [10] give a nice visualization of the former three layers. Regarding the nonlinear activation, it is the key for inductive learning (and thus intelligent signal processing) [20]. With nonlinearity, weights in the mapping function are selectively turned off/scaled by the nonlinear function. This subset of NN layers is supported by mainstream deep learning frameworks, e.g., TensorFlow [21], which is used together with Keras [22] in this article.

#### B. Autoencoder-Based Network Architecture

We propose to use the autoencoder-based network architecture [23] as our reference model, shown in Fig. 2. It is composed of an encoder containing convolution layers, a decoder containing deconvolution layers, an optional bypass between the encoder and the decoder, and a regression network, which can be located at the far end [18] or at the bottleneck [10]. The decoder is optional when the regression network is at the bottleneck. This reference model is the workload targeted by the NN accelerator, which will be discussed in Section IV.
C. Quantization-Aware Training and Validation

Conventional artificial NNs use floating-point numbers to represent weights and intermediate feature maps, which is straightforward for CPUs and GPUs, but not friendly to customized digital ASICs. It is possible to convert the floating-point model into the fixed point using 8- or 16-bit quantization. However, directly converting a well-trained model will result in significant degradation of accuracy.

To tackle the problem, a quantization-aware training scheme [24] natively supported by deep learning frameworks is utilized to gradually transform the original floating-point model into the quantized model with nearly no loss of accuracy. The three stages of the process are shown in Fig. 3. In each stage, waveform samples coming from detector simulation or experimental measurement are provided as network input, and signal features (such as time or energy) are provided as ground-truth labels to optimize parameters of the network and validate its performance.

The official code for quantization-aware training is assumed to be used with CPUs. To make it compatible with the NN accelerator here, we postprocess the layerwise results with rescale and bit shift on hardware (Section IV-C) and rewrite the mechanism for quantized-model export in the hardware–software codesign (Section IV-D).

IV. SoC ACCELERATOR DESIGN (βγ)

A. Brief Review of PulseDL

For application of NNs to process detector signals at FEE, we developed the first version of the NN accelerator customized for pulse processing, named PulseDL [25], [26]. PulseDL worked as a coprocessor and communicated with an reduced instruction set computer (RISC) processor through a proprietary bus. The accelerator architecture of PulseDL was mainly composed of a 4 × 4 PE array, fed by dedicated row buffers and column buffers, and followed by spatial and temporal adder trees (ATs). In operation of each PE, operands from the feature map vector and the kernel matrix are multiplied and accumulated.

The first version of the chip, although a successful practice, has some notable issues: 1) an RISC CPU outside the chip (or accelerator) is needed to schedule transactions, which is not convenient in use; 2) dynamic quantization (i.e., deciding the rounding bit after the feature map is obtained) may bring about additional overheads of time; 3) the AT structure, especially the temporal AT, has yet to be optimized for area and performance; and 4) finally, only manual configuration procedure is devised to map NN models onto the chip, while extension to deep learning frameworks could be a more automatic solution.

The above limitations motivate us to develop PulseDL-II, the new version of the digital design.

B. PulseDL-II: SoC Structure

The primary improvement of PulseDL-II is the system structure. We integrate an RISC CPU, the Arm Cortex-M0 microcontroller, and associated AHB/APB buses into the digital design to form an SoC, as shown in Fig. 4 [which solves issue 1]. The Cortex-M0 core is an intellectual property (IP) distributed freely as trial by the Arm company [27]. It is a microcontroller featuring the small footprint and low power.

The SoC has three major parts: the NN Pulse Processor (PulseDL-II accelerator), the Cortex-M0 SoC, and the Dual-Port AHB RAM. The PulseDL-II NN accelerator is mounted on the processor AHB bus as a peripheral. In this SoC, we have multiple input–output peripheral devices, such as quad/normal SPIs, UARTs (with or without the internal buffer), JTAG, and GPIOs.

The preset workflow is described as follows. The waveform samples coming from ADC and self-trigger logic are transmitted into the double-port buffer, preferably through the quad SPI interface. The Cortex-M0 core periodically accesses the double-port buffer and relays the input data to the NN accelerator. When the accelerator finishes computing, it raises a signal to the Cortex-M0 core and pushes feature maps into the system RAM under the coordination of the core. The data transmission between the system RAM and the accelerator is repeated several times, until the final feature outputs are ready. Finally, the output data are sent out through a buffered UART and collected by subsequent electronics.

It should be mentioned that transactions on the processor AHB bus generate unavoidable overheads in the total time budget (see Section IV-E). At the current stage, we have kept the CPU-centric design to maximize universality, but will probably incorporate direct memory access (DMA) with data processing abilities when we gather more information about the performance of different NN architectures.

C. PulseDL-II: Accelerator Architecture

Another important improvement is the accelerator architecture itself. The updated digital design of the PulseDL-II accelerator is shown in Fig. 5. A new hierarchical level, the AU, is added into the topology. There are three hierarchical levels in total: AU, PE, and NN, as shown in the bottom middle of this figure. In essence, PE plays a role, which is more self-contained and can be regarded as a miniature PulseDL accelerator. Accordingly, AU takes the responsibility of MAC at a more fine-grained level than the original PE in PulseDL; instead of performing MACs with a macro-PE array and postprocessing the results by macro-AT structures with lumped control logic, PulseDL-II distributes MACs into AUs in each PE and uses micro-AT structures and dedicated control logic. By these functional adjustments, both efficiency and flexibility have been improved, and the software mapping scheme fits better into the hardware, which, in turn, reduces power and area. A template-based methodology is adopted to design logic elements and to make elements in each level
adjustable. Compared with other methods, such as high-level synthesis, this methodology allows designers to perform cycle-accurate optimization and fine control of data interfaces, at the expense of more expertise and manpower. In the example design referred in Sections IV-F and V, we use four AUs in each PE and 15 PEs in each NN. The transactions are assumed to be directed to a single NN device.

For quantization compatible with TensorFlow or other deep learning frameworks, a functional block of rescale and bit shift [24] is integrated after temporary results are generated by bias and activation [which solves issue 2]). Rescale and bit shift is a procedure to adjust the scale of the output feature map, so that the full quantization bits can be effectively utilized. It serves the purpose in a way friendly to integer-only digital logic. The actual quantization bits used in the inference will determine the loss of accuracy by quantization.

In the PulseDL-II accelerator, other enhancements include broadcasting/multicasting input feature maps and kernels to multiple AUs, optimizing the (temporal) AT with the partial sum accumulator [which solves issue 3]), adding function blocks for bias addition and activation, and a whole new implementation of the mapping mode coordinator for different layers.

D. Hardware–Software Codesign

To ensure the proper functioning of the SoC, a framework for hardware–software codesign is implemented along with PulseDL-II, as shown in Fig. 6 [which solves issue 4]).
E. Embedded Software With Weight-Stationary Mapping

Within the SoC structure, different mapping rules are allowed depending on the requirements of application. For NNs with small/medium size, a weight-stationary mapping scheme can be adopted to reduce transactions when weights are used repeatedly. In this scheme, the operation is divided into two phases. In the preparation phase, weights are stored into PEs before waveform samples come in. In the inference phase, only input data, output data, and intermediate feature maps are transferred in and out of PEs. Besides, the weight-stationary embedded software enables the following features.

Layerwise Inference Pipelining: Weights for different layers are mapped to different groups of PEs; in inference,
due to the innovation in the accelerator architecture and connected layer #1, the reduction is very significant, mainly intensive layers, such as convolution layer #3 and fully reduces running time and energy consumption. For compute-

Fig. 9(a) and (b), it can be seen that PulseDL-II integrates parameters, shown in Fig. 8.

Fig. 7. Illustrated basic dataflow (top) and dataflow with pipelining and parallelism enabled (bottom).

Fig. 8. NN workload used to compare two versions of the design.

PEs can operate simultaneously as independent computing devices.

Event-Level Parallelism: Each event (In this article, we use the terminology event to mean a series of waveform samples generated by the detector response to a single initial particle.) is assigned a unique token and recorded by the Cortex-M0 core; in inference, the token will be traced by the core and passed in company with feature maps along the pipeline.

Fig. 7 shows the improvement of throughput (at the risk of prolonging latency) when above two features are enabled. The improvement is relevant to the proportion of time in data transmission and the proportion of time in NN computation.

F. Evaluation

We compare the performance of PulseDL and PulseDL-II on an FPGA (Xilinx ZCU104 evaluation board) platform. The working frequency is set to 100 MHz for both versions. For fair comparison with PulseDL, the PulseDL-II NN accelerator is isolated when measuring power and area.

We use an NN workload for both versions of the hardware, comprising three convolution layers and two fully connected layers and containing \( \approx 33.4k \) MACs and 18.8k trainable parameters, shown in Fig. 8.

The evaluation results are shown in Fig. 9. In Fig. 9(a) and (b), it can be seen that PulseDL-II greatly reduces running time and energy consumption. For compute-intensive layers, such as convolution layer #3 and fully connected layer #1, the reduction is very significant, mainly due to the innovation in the accelerator architecture and mapping scheme. These two layers contribute largely to the overall improvement in performance (1.83× less) and power (1.81× less).

For resource utilization in Fig. 9(c), PulseDL-II integrates more multipliers (64 versus 360), but the average utilization of FPGA hardware resources is comparable or less. The lookup tables (LUT) divided by 8-bit multipliers are decreased from 739 to 529 (1.40× less), and the flip-flops (FF) divided by 8-bit multipliers are increased from 920 to 933 (1.01× more).

The preliminary evaluation shows the advancement of PulseDL-II for common workloads used in feature extraction of nuclear detector signals. For one thing, the new accelerator architecture, supported by the SoC structure, boosts the performance of inference; for another, power reduction is observed, and resource utilization is acceptable, which makes the ASIC implementation much more promising.

V. SYSTEM VALIDATION (\( \alpha\beta\gamma \))

A. Experimental Setup

To validate the whole system, we establish an experimental environment without actually connecting the electronics to the detector, as shown in Fig. 10. A direct digital synthesis (DDS) signal generator (AD9106—ARDZ-EBZ) working at 156.25 MHz produces four-channel analog signals, which can be programed by an Arm microcontroller board (SDP-K1) beforehand. At the other side, we use an FPGA development board (MZU07A—EV) equipped with two ADC cards (ADS4225) sampling at 125 MHz to receive and digitize the analog signals. The data acquisition (DAQ) front end and the SoC with the NN accelerator are implemented as digital logic on the FPGA development board.

Fig. 11 shows the digital logic for DAQ, a prerequisite for feature extraction by the accelerator SoC. We use self-trigger to get a snapshot of data samples in the ring buffer and the timestamp. The triggered data are fed to event DAQ for both monitoring the current waveform and preprocessing the waveform for the accelerator SoC. Multiple integrated logic analyzers (ILAs) and the virtual input–output (VIO) are inserted into the dataflow to probe internal signals and manipulate the data path.

B. Experimental Results

In the experiment, we prestore the CRRC waveform (step signal filtered by the bandpass capacitance–resistance circuit) into the DDS signal generator. The waveform function is shown in the following equation:

\[
s(t) = K \left( \frac{t - t_0}{\tau} \right) \exp\left(\frac{-(t - t_0)^2}{2\sigma^2}\right) u(t - t_0)\tag{1}\]

where \( u(t) \) is the step function. We set \( \tau \) to 40 ns, and set \( K = K_1 K_2 \), where \( K_1 \) is a constant keeping the signal-to-noise ratio (SNR) relative to the baseline noise (\( \sigma_{\text{base}} \)) to 47.4 dB

\[
\text{SNR} = 20 \log_{10} \left( \frac{K_1}{\sigma_{\text{base}}} \right) = 47.4 \text{ dB}\tag{2}\]
and $K_2$ is uniformly sampled in the range between 0.5 and 2.0. For time analysis, we produce dual-channel synchronous waveform and compare the timing results of two channels. For energy analysis, we first vary the waveform amplitude in a certain range and then use a standard waveform to assess energy predictions.

The NN model we use is similar to Fig. 8. It is comprised of two convolution layers and three fully connected layers and contains 9.8k MACs and 5.8k trainable parameters. Waveform samples from the monitoring branch in the DAQ are used to train the NNs and to export network weights for online processing. Identical waveform samples are used for traditional feature extraction methods.

In Fig. 12, we show experimental results of time/energy prediction by different methods, including traditional methods (interpolated constant fraction discrimination [28] for time and waveform integration for energy), offline floating-point NNs, and online quantized (8-bit fixed point) NNs. It can be seen that the NNs work better than traditional methods in the same conditions. Since quantization effects will influence the final online resolution, we observe a slight degradation of resolution when comparing the bottom with the middle in each subfigure. It will be eliminated if more quantization bits are used (such as 16-bit fixed point).

Here, we report resource utilization, power, and performance of the experimental system. Measured on the Xilinx Zynq UltraScale+ FPGA, the DAQ uses 2825 LUTs, 517 FFs, eight block RAMs (36 kb each), and eight UltraRAMs (288 kb each) and consumes 0.371-W dynamic power. The accelerator SoC uses 89,540 LUTs, 75,028 FFs, and 48 block RAMs and consumes 0.541-W dynamic power. The device static power is 0.594 W. Both the dynamic power of the DAQ and the static power are expected to improve significantly with the ASIC implementation. The time for on-chip NN inference is 113.8 $\mu$s at 100-MHz working frequency. With time consumption from data input–output, the total latency of a single event is expected to be 165 $\mu$s. When working in the pipelined mode, the throughput is 8.3k events/s.
VI. CONCLUSION

The ability and potential of NNs in feature extraction of nuclear detector signals are investigated. Based on the advantage of NNs, we prototype an NN accelerator-centric FEE for ECAL at NICA-MPD. Application-specific NN architectures are proposed, and quantization-aware training is extended for use by customized computing devices.

The major part of our work is to develop an SoC digital system with the NN accelerator. The SoC approach is flexible not only in changing weights of a target NN, but also in accommodating a variety of network workloads within the selected subset. We elaborate on the design of PulseDL-II from both hardware and software aspects. A comparison between PulseDL-II and its previous version demonstrates the advancement of digital design. Finally, system validation on an FPGA platform is done.

In the future, we will evaluate the whole system in real-world nuclear detector dataflow. We also plan to tape out with 28/65-nm process after the ASIC layout has been finished.

REFERENCES

[1] F. Ameli et al., “A low cost, high speed, multichannel analog to digital converter board,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 936, pp. 286–287, 2019.
[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, Feb. 2015.
[3] S. Choudhury, “Tau identification with deep neural networks at the CMS experiment,” IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 2194–2200, Aug. 2021.
[4] M. Astrain et al., “Real-time implementation of the neutron/gamma discrimination in an FPGA-based DAQ MTCA platform using a convolutional neural network,” IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 2173–2178, Aug. 2021.
[5] B. Clerbaux, M. C. Molla, P.-A. Petitjean, Y. Xu, and Y. Yang, “Study of using machine learning for level 1 trigger decision in Juno experiment,” IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 2187–2193, Aug. 2021.
[6] E. Berg and S. R. Cherry, “Using convolutional neural networks to estimate time-of-flight from PET detector waveforms,” Phys. Med. Biol., vol. 63, no. 2, Jan. 2018, Art. no. 02LT01.
[7] S. I. Kwon et al., “Ultrafast timing enables reconstruction-free positron emission imaging,” Nature Photon., vol. 15, no. 12, pp. 914–918, Dec. 2021.
[8] A. R. Garola et al., “Diagnostic data integration using deep neural networks for real-time plasma analysis,” IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 2165–2172, Aug. 2021.
[9] P. Ai, Z. Deng, Y. Wang, and C. Shen, “Universal uncertainty estimation for nuclear detector signals with neural networks and ensemble learning,” J. Instrum., vol. 17, no. 2, Feb. 2022, Art. no. P02032.
[10] P. Ai, Z. Deng, Y. Wang, and L. Li, “Neural network-featured timing systems for radiation detectors: Performance evaluation based on bound analysis,” J. Instrum., vol. 16, no. 9, Sep. 2021, Art. no. P09019.
[11] G. D. Guglielmo et al., “A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC,” IEEE Trans. Nucl. Sci., vol. 68, no. 8, pp. 2179–2186, Aug. 2021.
[12] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” IEEE J. Solid-State Circuits, vol. 52, no. 1, pp. 127–138, May 2017.
[13] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, “Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices,” IEEE J. Emerging Sel. Topics Circuits Syst., vol. 9, no. 2, pp. 292–308, Jun. 2019.
[14] (Dec. 13, 2022). The MultiProcessor Detector MPD (Conceptual Design Report). Accessed: Oct. 4, 2022. [Online]. Available: http://mpd-jirn.ru/wp-content/uploads/2016/04/MPD_CDR_en.pdf
[15] (Dec. 13, 2022). MPD NICA Technical Design Report of the Electromagnetic Calorimeter (ECAL). Accessed: May 24, 2021. [Online]. Available: http://mpd.jirn.ru/wp-content/uploads/2019/01/TDR_ECAL_v3.6_2019.pdf
[16] C. Shen et al., “Development of shashlik electromagnetic calorimeter for the NICA/MPD,” J. Instrum., vol. 14, no. 6, Jun. 2019, Art. no. T06005.
[17] Y. Li et al., “Beam test results of two shashlyk ECAL modules for NICA-MPD,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 958, Apr. 2020, Art. no. 162833.
[18] P. Ai, D. Wang, G. Huang, N. Fang, D. Xu, and F. Zhang, “Timing and characterization of shaped pulses with MHz ADCs in a detector system: A comparative study and deep learning approach,” J. Instrum., vol. 14, no. 3, Mar. 2019, Art. no. P03002.
[19] B. Jacob et al., “Quantization and training of neural networks for real-time plasma analysis,” J. Instrum., vol. 17, no. 2, Feb. 2022, Art. no. P02032.
[20] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” 2016, arXiv:1603.04467.
[21] F. Chollet. (Dec. 13, 2022). Keras. Accessed: Nov. 1, 2022. [Online]. Available: https://github.com/keras-team/keras
[22] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. MICCAI, Cham, Switzerland: Springer, 2015, pp. 234–241.
[23] B. Jacob et al., “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2704–2713.
[24] P. Ai et al., “PulseDL: A reconfigurable deep learning array processor dedicated to pulse characterization for high energy physics detectors,” Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip., vol. 978, Oct. 2020, Art. no. 164420.
[25] J.-L. Chen et al., “FPGA implementation of neural network accelerator for pulse information extraction in high energy physics detectors,” Nucl. Sci. Techn., vol. 31, no. 5, p. 46, Apr. 2020.
[26] (Dec. 13, 2022). Arm Limited Website. Accessed: Nov. 1, 2022. [Online]. Available: https://www.arm.com/
[27] Y. Fan et al., “Research and verification on real-time interpolated timing discrimination in an FPGA-based DAQ MTCA platform using a convolutional neural network,” IEEE Trans. Nucl. Sci., vol. 67, no. 10, pp. 2246–2254, Oct. 2020.