frontiers | Frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 18 May 2022
doi: 10.3389/frai.2022.855184

Real-Time Inference With 2D Convolutional Neural Networks on Field Programmable Gate Arrays for High-Rate Particle Imaging Detectors

Yeon-jae Jwa, Giuseppe Di Guglielmo, Lukas Arnold, Luca Carloni and Georgia Karagiorgi*

Columbia University, New York, NY, United States

We present a custom implementation of a 2D Convolutional Neural Network (CNN) as a viable application for real-time data selection in high-resolution and high-rate particle imaging detectors, making use of hardware acceleration in high-end Field Programmable Gate Arrays (FPGAs). To meet FPGA resource constraints, a two-layer CNN is optimized for accuracy and latency with KerasTuner, and network *quantization* is further used to minimize the computing resource utilization of the network. We use "High Level Synthesis for Machine Learning" (*hls4ml*) tools to test CNN deployment on a Xilinx UltraScale+ FPGA, which is an FPGA technology proposed for use in the front-end readout system of the future Deep Underground Neutrino Experiment (DUNE) particle detector. We evaluate network accuracy and estimate latency and hardware resource usage, and comment on the feasibility of applying CNNs for real-time data selection within the currently planned DUNE data acquisition system. This represents the first-ever exploration of employing 2D CNNs on FPGAs for DUNE.

Keywords: data selection, particle imaging, liquid argon time projection chamber, hardware acceleration of deep learning, real-time machine leaning, fast machine vision, data acquisition system, trigger system

OPEN ACCESS

Edited by:
Mia Liu,
Purdue University, United States

Reviewed by:
Gabriel Nathan Perdue,
Fermi National Accelerator Laboratory (DOE), United States
Alexander Radovic,
Borealis AI, Canada
Yongbin Feng,
Fermi National Accelerator Laboratory (DOE), United States

****Correspondence:***
Yeon-jae Jwa
yj2429@columbia.edu

Specialty section:
This article was submitted to Frontiers in Artificial Intelligence, a section of the journal Frontiers in Artificial Intelligence

Received: *14 January 2022*
Accepted: *12 April 2022*
Published: *18 May 2022*

Citation:
Jwa Y-j, Di Guglielmo G, Arnold L, Carloni L and Karagiorgi G (2022) Real-Time Inference With 2D Convolutional Neural Networks on Field Programmable Gate Arrays for High-Rate Particle Imaging Detectors. Front. Artif. Intell. 5:855184. doi: 10.3389/frai.2022.855184

1. INTRODUCTION

Modern-day particle physics experiments produce vast amounts of data that must be processed to down-select interesting (and usually rare) signals for further physics study and scientific discovery. This process of data selection is applied across several stages of the data processing pipeline. In recent years, such pipelines have increasingly made use of deep learning (DL) (Radovic et al., 2018; Karagiorgi et al., 2021). Additionally, as data rates grow, there is increased need to accurately and efficiently execute data selection in real time, i.e., at a rate commensurate with data generation throughput and with low latency, by employing "triggers". These are real-time data-driven decisions, which translate physical measures—quantities calculated based on the incoming data itself and/or other external signals—into instructions on which data to keep or permanently discard.

Driven in part by the need to increase accuracy in selecting high-dimensional and highly-detailed data from modern-day particle detectors, machine learning (ML) algorithms based on both supervised and unsupervised learning have been proposed and shown to be capable of effectively triggering on incoming physics data, proving to be a promising solution for the

upcoming data challenges of future experiments. Implementing ML algorithms into dedicated hardware for triggering, such as GPUs, or FPGAs, can potentially guarantee fast execution of the algorithm while taking advantage of the algorithm's accuracy in selecting data of interest with maximal signal efficiency and signal purity. Additionally, software toolkit development projects such as *hls4ml* (Fahim et al., 2021) are providing suitable and user-friendly frameworks for easily employing ML algorithms into hardware for application-specific usage (see, e.g., Aarrestad et al., 2021; Loncar et al., 2021).

Further motivated by a widely used particle imaging detector technology—liquid argon time projection chambers (LArTPCs)—we explore the applicability of algorithms commonly used in image analysis for LArTPC triggering purposes, following Jwa et al. (2019). LArTPCs work by continuously imaging a large and homogeneous 3D detector volume, wherein electrically charged particles are visible through the trails of ionization they leave along their trajectories. This type of technology is employed in searches of rare events such as interactions of dark matter particles or supernova core-collapse neutrinos with the detector medium. More so than for other particle detector technologies, LArTPC data are well-suited for image analysis given that neutrino or other rare event signals are translationally invariant within a generally sparse 2D or 3D image of the detector volume. In past work (Jwa et al., 2019), we have shown that 2D convolutional neural networks (CNNs) tested on simulated raw data from a LArTPC can yield sufficient accuracy and can be implemented onto parallelized data processing pipelines using GPUs to perform data selection in a straightforward way, while meeting the physics performance and latency requirements of future LArTPC experiments.

The need to improve the long-term operation reliability and power utilization of such data processing pipelines motivates the exploration of alternate implementations of CNN-based data selection, specifically implementations on Field Programmable Gate Arrays (FPGAs). FPGAs are low-power digital microelectronics devices commonly used for signal processing and data acquisition applications. They are commonly used in front-end readout electronics systems for particle physics experiments; their on-device nature (often capable of receiving the full-rate of detector-generated data prior to any data filtering or reduction) and their reliability for long-term operation make them attractive for data processing algorithm implementation, especially if only minor pre-processing is necessary in the data pipeline. In general, algorithm implementation into a front-end device is advantageous as it makes large data movement unnecessary, reduces power consumption and trigger latency, and increases reliability. More recently, there has been a growing interest in using FPGAs as accelerators for deep neural networks (Trimberger, 2015).

A number of ML algorithms have already been explored for particle triggering and suitability for FPGA applications; see, e.g., Heintz et al., 2020; Iiyama et al., 2020; Summers et al., 2020; Aad et al., 2021; Deiana et al., 2021; Diotalevi et al., 2021; Elabd et al., 2021; Govorkova et al., 2021; Mikuni et al., 2021. Explored algorithm implementations range from Artificial Neural Networks, to Boosted Decision Trees, Graph Neural Networks, to Autoencoders, etc. In this article, we investigate, for the first time, the implementation of a relatively small 2D CNN onto an FPGA, targeted for use in the front-end readout electronics of the future Deep Underground Neutrino Experiment (DUNE) (Abi et al., 2020a,b,c,d), motivated by previous exploration and findings in Jwa et al. (2019). While the use of CNNs for image classification applications has been established for well over a decade (Ciresan et al., 2011), their specific use in fast-inference applications in particle physics has been restricted to non-LArTPC applications (Duarte et al., 2018, 2019). On the other hand, in the case of LArTPCs, CNNs have been used successfully for *offline* data analysis and physics measurements (see, e.g., Acciarri et al., 2012, 2017a; Adams et al., 2019; Abi et al., 2020e; Abratenko et al., 2021a,b). Keeping in mind the 2D nature and high resolution of LArTPC raw data, we explore and evaluate techniques to reduce the computational resource usage of CNN inference on FPGAs. We focus on the DUNE case, and show that we can meet the technical specifications of the DUNE readout system, while still satisfying the physics accuracy requirements of the experiment. We add that other DL algorithms have also been studied for offline data analysis of LArTPC data (Koh et al., 2020; Drielsma et al., 2021), and would also be worth exploring for FPGA implementation for LArTPC trigger applications.

In Section 2, we describe the DUNE Far Detector (FD) LArTPC in more detail, including its operating principle, and the technical specifications and requirements of its readout and data selection (trigger) system. In Section 3 we explore different CNN architectures, and explore their accuracy in selecting data containing rare signal events, paying attention to the overall size of the network, in anticipation of minimal computational resource availability in the DUNE FD readout system. Section 3.1 describes how simulated raw data from the DUNE FD are prepared as input to the CNN; Section 3.2 describes some CNN architectures and the classification accuracy performance on simulated input images; in Section 3.3, we further optimize the network architecture and hyperparameters in an automated way, using the KerasTuner package[1] (O'Malley et al., 2019), and compare classification accuracy of the automatically optimized network to the non-optimized ones. Throughout all subsections, we also present network accuracy results using "HLS-simulated" versions of the CNNs, produced using the *hls4ml* package (Fahim et al., 2021). One key feature of *hls4ml* is a reduction in accuracy due to quantization of the network, which we avoid by employing quantization-aware training, following (Coelho et al., 2021; Hawks et al., 2021), as discussed in Section 3.4. Finally, in Section 4, we provide estimates of FPGA resource usage of the optimized networks (with and without quantization-aware training), using an *hls4ml* synthesized design for a targeted FPGA hardware implementation. We demonstrate that the use of 2D CNNs for real-time data selection in the future DUNE is viable, and advantageous, given the currently envisioned front-end readout system design.

[1] *KerasTuner*. Available online at: https://keras.io/keras_tuner/ (accessed December 20, 2021).

2. APPLICATION CASE: REAL-TIME DATA SELECTION FOR THE FUTURE DUNE LARTPC

LArTPCs are a state-of-the-art charged-particle detector technology with broad applications in the field of particle physics, astro-particle physics, nuclear physics, and beyond. This high-rate imaging detector technology has been adopted by multiple particle physics experiments, including the current MicroBooNE experiment (Acciarri et al., 2017b), two additional detectors that are part of the upcoming Short-Baseline Neutrino (SBN) program (Antonello et al., 2015), as well as the next-generation DUNE experiment (Abi et al., 2020a,b,c,d), and it is also proposed for future-generation astro-particle physics experiments such as GRAMS (Aramaki et al., 2020). LArTPCs work by imaging ionization electrons produced along the paths of charged particles, as they travel through a large (multiple cubic meters) volume of liquid argon. Charged particle ionization trails drift uniformly toward sensor arrays with the use of a uniform electric field applied throughout the liquid argon volume, and are subsequently read out in digital format as part of 2D projected views of the 3D argon volume. This is illustrated in **Figure 1**. Densely packed sensor arrays sample the drifted ionization charge at a high rate, typically using a 12-bit, 2 MHz Analog to Digital Converter (ADC) system recording the amount of ionization charge per sensor per time-sample, thus imaging charge deposition across 2D projections of the argon volume with millimeter-scale resolution. Typically, digitized image frames of O(10) megabytes each are streamed out of these detectors in real time and at a rate of up to hundreds of thousands of frames per second, amounting to raw data rates of multiple gigabytes to several terabytes (TB) per second.

The future DUNE experiment represents a special case, with the most stringent data processing requirements among all currently running or planned LArTPC experiments. DUNE consists of a near and a far detector complex, which will be located at Fermi National Accelerator Laboratory (Fermilab) in Illinois and at the Sanford Underground Research Facility

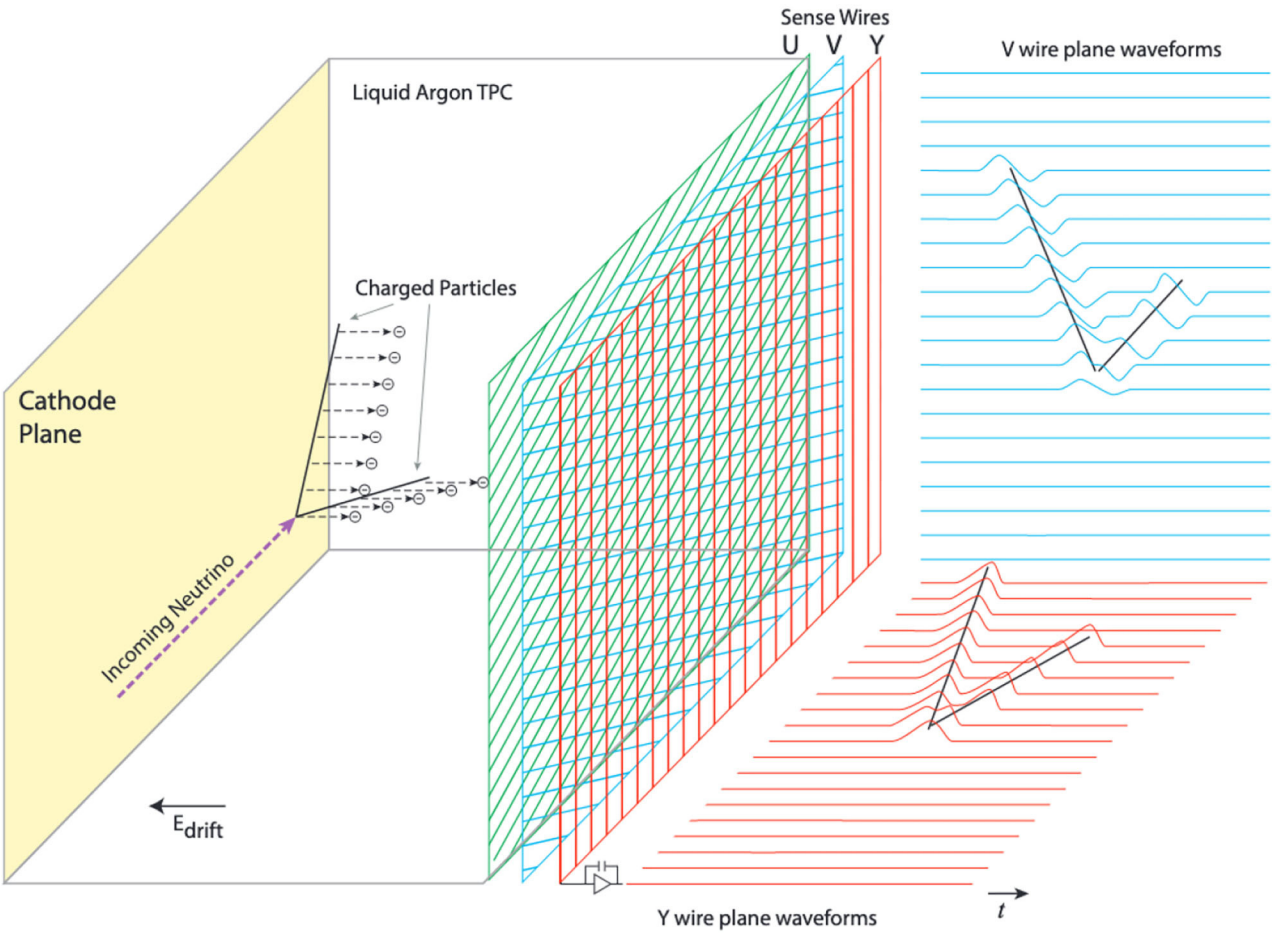

FIGURE 1 | Operating principle of a LArTPC. The ionization electrons are drifted toward sensor arrays, e.g., planes of ionization charge sensor wires. Each wire is connected to an analog amplifier/shaper, followed by an ADC, and its resulting digital waveform is read out continually. Waveforms of adjacent wires appended together form 2D images. Image credit: Acciarri et al. (2017b).

(SURF) in South Dakota, respectively. The far detector (FD) complex will be located 1 mile deep under ground, and will comprise the largest LArTPC ever to be constructed, with an anticipated raw data rate for its first of four LArTPC modules of 1.175 TB/s. This first detector module will be operated continually, and for at least 10 years, with subsequent modules coming online before the end of the current decade. The DUNE FD will therefore be constructed with a readout and data selection system that is required to receive and process an overall raw data rate of 4×1.175 TB/s, achieve a factor of 10^4 data reduction, and maintain $>$ 99% efficiency to particle interactions of interest that are predicted to be as rare as once per century (Abi et al., 2020d).

The scientific goals of DUNE include, but are not limited to, observing neutrinos from rare (once per century) galactic supernova bursts (SNBs) (Abi et al., 2020b, 2021b), searching for rare baryon number violation processes such as argon-bound proton decay and argon-bound neutron-antineutron oscillation, and studying interactions of neutrinos that are produced in cosmic ray air showers in the Earth's atmosphere (Abi et al., 2020b, 2021a). From the data acquisition (DAQ) and data selection (trigger) point of view, these rare physics searches and in particular the requirement to be $>$ 99% efficient to a galactic SNBs with a less than once per month false positive SNB detection rate, cast particularly stringent technical requirements.

More specifically, in order to select these "events", which take place randomly and unpredictably, the DUNE DAQ and trigger system must scan *all* detector data continuously and with zero dead time, and identify rare physics signatures of interest in a "self-triggering" mode—without relying on any external signals prompting data readout. Furthermore, a self-triggering scheme reaching nearly perfect (100%) efficiency for rare physics events is needed in order for DUNE to achieve its full physics program. This further requires temporarily buffering large amounts of data while this processing takes place. In the case of DUNE, buffering constraints translate into a sub-second latency requirement for the trigger decision. Additionally, the trigger decision needs to achieve an overall 10^4 data rate reduction, and with high signal selection efficiency, corresponding to an average of $>$60% efficiency on individual supernova neutrino interactions, and $>$90% efficiency to other rare interactions including atmospheric neutrino interactions and baryon number violating events.

The first DUNE FD module will image charged particle trajectories within 200 independent but contiguous liquid argon volume regions ("cells"). Charged particle trajectories within each cell will be read out by sensor wires arranged in three planes: one charge-collection wire plane, plus two charge-induction wire planes. Each plane's readout corresponds to a particular 2D projected view of the 3D cell volume, and the combination of induction and collection plane information allows for 3D stereoscopic imaging and reconstruction of any given interaction within the 3D cell volume. In total, the first FD module will consist of 384,000 wire sensors, each read out independently; this outnumbers current LArTPC neutrino experiments by more than a factor of 500 (e.g., MicroBooNE makes use of 8,256 wire sensors). For this work, we focus exclusively on charge-collection wire readout. Charge-collection wires give rise to signals which are unipolar in nature (as opposed to charge-induced signals, which are bipolar in nature, and therefore susceptible to cancellation effects). As such, charge-collection readout waveforms preserve sensitivity to charge deposition even for extended charge distributions. Since particle identification (and subsequent data selection decision making) relies on quantifying the amount of charge deposition per unit length of a charged particle track, charge-collection waveform information is anticipated to provide better particle identification performance.

The 200 cells of the first DUNE FD module will be read out in parallel, by 75 "upstream DAQ" readout units. Each unit makes use of a Front-End LInk eXchange (FELIX) PCIe 3.0 card (Borga et al., 2019; Abi et al., 2020d) holding a Xilinx UltraScale+ FPGA to read out digitized waveforms, and pre-process the data. In the nominal DUNE readout unit design, the FPGA processes continuous waveforms in order to perform noise filtering and hit-finding; hit-finding summaries are then sent for additional processing to a FELIX-host CPU system, in order to form trigger candidates (particle interaction candidates); the latter inform a subsequent module-wide trigger decision. An alternate potential solution, and the scope of this work, is to apply more advanced data processing and triggering algorithms within the available FPGA resources on-board the FELIX card, such as CNNs, which can intelligently classify a collection of waveforms representing activity across the entire cell volume in real time. This would eliminate the need of subsequent CPU host (or GPU) processing, potentially increase trigger efficiency and purity (through the use of more intelligent algorithms), and potentially further minimize power consumption needs. It is worth noting that, since most interactions of interest have a spatial extent which is smaller than the cell volume, a per-cell parallelization of triggering algorithms is appropriate, and it is therefore sufficient to focus trigger studies to a per-cell level, ignoring cell volume boundary effects.

3. CNN DESIGN AND OPTIMIZATION FOR REAL-TIME LARTPC DATA SELECTION

In recent years, DL algorithms such as CNNs have been shown to achieve very high signal selection efficiencies when employed in offline physics analyses of LArTPC data. MicroBooNE is leading the development and application of DL techniques, including CNNs, for LArTPC data reconstruction (Acciarri et al., 2017a; Adams et al., 2019; Abratenko et al., 2021a,b), and CNN-based analyses and DL-based reconstruction are actively being developed for SBN and for DUNE (Acciarri et al., 2012; Abi et al., 2020e).

In a previous study (Jwa et al., 2019), we have also shown that sufficiently high efficiencies can be reached by processing raw collection plane data from any given DUNE FD cell, prior to removing any detector effects or applying data reconstruction. As such, we proposed a CNN-based triggering scheme using streaming raw 2D image frames, whereby the images are pre-processed, downsized, and run through CNN inference to select ones containing SNB neutrino interactions or other rare interactions of interest on a frame-by-frame basis. The data pre-processing and CNN-based selection method demonstrated

that target signal selection efficiency while reaching the needed 10^4 background rejection could be achieved, given sufficient parallelization in GPUs. As the DUNE FD DAQ and trigger design is subject to stringent power limitations and limited accessibility in the underground detector cavern, a particularly attractive option is to fully implement this pre-processing and CNN-based inference on FPGAs, in particular ones that will be part of the DUNE upstream DAQ readout unit design. We examine the viability of this option in this work.

Specifically, we explore the accuracy of relatively small CNNs in classifying streaming DUNE FD LArTPC cell data, and proceed to employ network optimization in an effort to reduce its computational resource footprint while preserving network accuracy. The following subsections describe the CNN input image preparation (Section 3.1), CNN performance without (Section 3.2) and with (Section 3.3) network optimization, and with quantization-aware training (Section 3.4).

3.1. Input Image Pre-processing

Because of the parallelism in the DUNE FD DAQ and trigger design, we only consider a single cell's worth of data at a time, and focus exclusively on raw collection plane waveforms. Following (Jwa et al., 2019), collection plane waveforms for a single cell in the DUNE FD are simulated in the LArSoft framework[2] (Church, 2013), using the default configuration of the *dunetpc* software, and using an enhanced electronics noise level configuration, to be conservative. Besides electronics noise, the simulation includes radiological impurity background interactions that are intrinsic to the liquid argon volume. The radiological background interactions (predominantly from ^{39}Ar decay) are expected to occur at a rate of 10^7 Hz per FD module, and they are considered as likely backgrounds particularly to supernova neutrino interactions. Signal waveforms from interactions of interest, including low-energy supernova neutrino interactions or other high-energy interactions (proton decay, neutron-antineutron oscillation, atmospheric neutrino interactions, cosmogenic background interactions), are overlaid on top of intrinsic radiological background and electronics noise waveforms.

Given the physical dimension of a cell along the ionization charge drift direction, and the known ionization charge drift velocity, 2.25 ms worth of continuous data from the entire collection plane represents a 2D image exposure of the full cell volume. As such, we define a 2D image in terms of 480 collection plane wire channels spanning the length of the cell volume, times the 2.25 ms drift direction sampled at 2 MHz (4,488 samples) spanning the width of the cell volume. This corresponds to a 2.1 megapixel image, with 12-bit ADC resolution governing the range of pixel values, dictating the amount of ionization charge collected by each wire, and indicating the energy deposit within the 3D volume across the given 2D projection.

For network training purposes, the 2.1 megapixel input images are labeled as containing either electronics noise and radiological background only (NB), or low-energy supernova neutrino interactions (LE), or high-energy interactions (HE), each superimposed with electronics noise and radiological background, according to the simulation truth. **Figure 2** shows example input 2D images before pre-processing steps. We note the sparsity of these images, mostly containing uniformly distributed low-energy activity from noise and radiological backgrounds. While it is possible to train a CNN with 2.1 megapixel images, it is not memory-efficient, and it may furthermore not be an efficient way to propel a CNN to learn the different features between the three event classes (NB, LE, and HE). Following (Jwa et al., 2019), we adopt pre-processing steps that include de-noising (zero-suppression), cropping around the region-of-interest (ROI), and resizing the ROI through down- or up-sampling. The de-noising step uses a configurable threshold for the pixel ADC value and zero-suppresses pixel values below this threshold; a threshold of 520 ADC (absolute scale) was used in these studies, where $\sim$ 500 ADC represents the baseline. ROI cropping was performed by finding a contiguous rectangular region containing pixels with values over 560 ADC. The most extreme image coordinates (smallest and largest channel number, as well as smallest and largest time tick) with pixel values greater than the lower threshold of 560 ADC were used to determine the ROI boundaries. Once an ROI was found, the ROI region was resized (through up-sampling or down-sampling) to occupy exactly 64×64 pixels, as shown in **Figure 3**. Resulting image pixel values were then re-normalized to a range between 0-1 prior to CNN processing.

Resized ROIs were generated for each of the three categories indicated in **Table 1**, with comparable statistics, and used for network training and testing for all studies presented in the subsequent sections. As we are investigating the viability of CNN implementation on FPGAs, computational resource utilization by the CNN is a key concern, and we therefore begin by investigating performance for relatively small-sized CNNs. For this task, we are working with a relatively small data set, split into training (60%), validation (20%) and test set (20%). **Table 1** shows the statistics for only the training and testing data sets.

The overall data processing and data selection scheme proposed and examined in this study is summarized in **Figure 4**.

3.2. Performance of CNN-Based Data Selection

Targeting FPGA implementations, we designed and tested custom CNN architectures with only one or two convolutional layers: **CNN01**, **CNN02**, and a downsized version of the latter, **CNN02-DS**. These networks have far simpler architectures than some of the more popular CNN architectures commonly used in image classification tasks [e.g., VGG (Simonyan and Zisserman, 2014) or ResNet (He et al., 2016) network architectures], by design, as they are targeted for implementation in computational-resource-constrained systems.

The network architecture for **CNN01** is shown in **Figure 5**. **CNN01** has one convolutional layer, with convolutional width kernel dimension (3,3,32), and one max-pooling layer. One fully connected layer follows at the end. In contrast, **CNN02**

[2] *LArSoft*. Available online at: https://larsoft.org/ (accessed December 20, 2021).

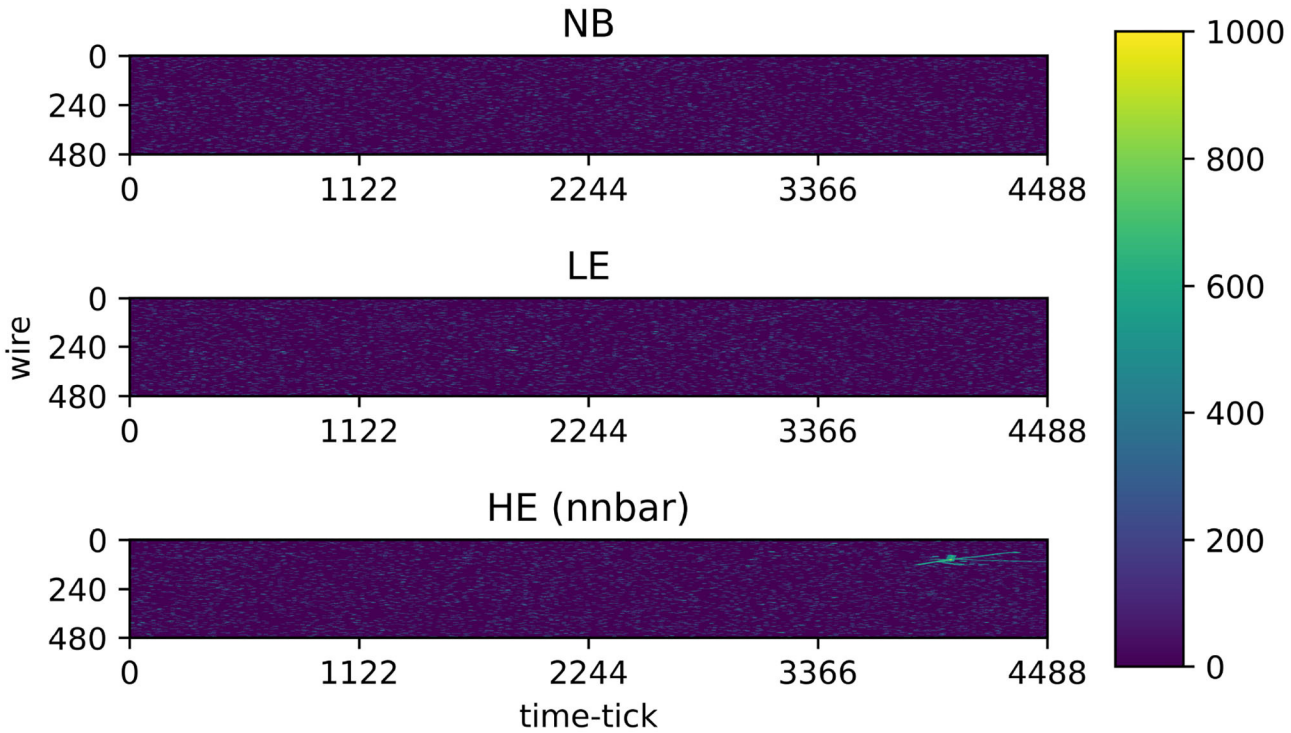

FIGURE 2 | Examples of 2D images formed from one full drift (2.25 ms) of 480 collection plane wires in one DUNE FD cell. **Top**: Image containing electronics noise and radiological background only (NB). **Middle**: Image containing one low-energy supernova neutrino interaction (LE) superimposed with electronics noise and radiological background. **Bottom**: Image containing one high-energy interaction (HE), specifically from neutron-antineutron oscillation (nnbar), superimposed with electronics noise and radiological background. The color *z* axis indicates the wire waveform ADC value (simulated with 12-bit resolution). These images are pre-processed prior to CNN processing.

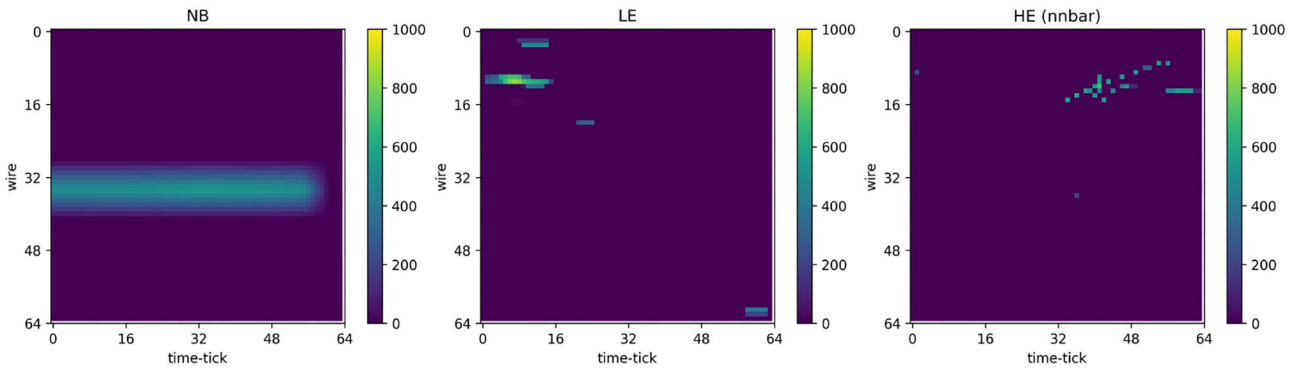

FIGURE 3 | Example ROIs formed after pre-processing. **Left**: Image containing electronics noise and radiological background only (NB). **Middle**: Image containing one low-energy supernova neutrino interaction (LE) superimposed with electronics noise and radiological background. **Right**: Image containing one high-energy interaction (HE), specifically from neutron-antineutron oscillation (nnbar), superimposed with electronics noise and radiological background. The color *z* axis indicates ADC values after down- or up-sampling. These images are renormalized with pixel values ranging from 0 to 1 prior to inputting to a CNN for subsequent processing (data selection).

has two convolutional layers, and one max-pooling layer after each convolution. Also, here, one fully connected layer follows at the end. Finally, **CNN02-DS** is a downsized version of **CNN02**, where the convolution depth is significantly reduced. All three custom network architectures are summarized in **Table 2**.

Table 3 shows the classification performance of the three networks, for a GPU or CPU implementation using Keras[3]. The performance of these three networks is comparable. For

[3]*Keras*. Available online at: https://github.com/keras-team/keras/ (accessed December 20, 2021).

TABLE 1 | Number of ROIs, according to truth label, used for training and testing of CNNs.

	Label: NB	Label: LE	Label: HE
Training set size	12,023	12,050	10,137
Testing set size	4,027	3,970	3,417

Another statistically independent sample of images with similar statistics to the test sample was used for validation purposes during training.
A total of 45,624 ROIs were used in the study.

all three networks, the false positive identification rates (which affect data reduction capability) are comparable, and the (correct) classification accuracy is over 99% for NB labeled ROIs, over 93% for LE labeled ROIs, and over 90% for HE labeled ROIs. Despite the difference in architecture (one vs. two convolution layers) and number of trainable parameters, no clear impact on classification performance is observed.

While accuracy results meet signal efficiency requirements[4], the high false positive rate (in particular for true NB ROIs to be mis-classified as LE events at a rate of 0.5%) suggests a steady-state data reduction factor for a frame-by-frame data selection implementation that is a factor of 50 lower than the required reduction factor of 10^4. This is because the overwhelming majority (>99.9%) of the streaming ROIs in DUNE are expected to be truly NB ROIs, and therefore a 0.5% mis-classification rate would result in approximately one in 200 ROIs being (falsely) selected, as opposed to the targeted one in 10,000. Additional data reduction, however, can be provided by an ROI pre-selection stage, as motivated in Jwa et al. (2019); specifically, approximately only one in 50 2D true NB images are expected to be non-empty after ROI finding (see **Figure 4**) and therefore 98% of the ROIs can be discarded prior to CNN processing.[5] This suggests that an overall factor of 10^4 is achievable.

In this work, the ML models were trained and tested on GPUs with single-precision floating-point arithmetic (standard IEEE 754), and then post-training quantization (PTQ) was performed with the aim of running ML inference on FPGA. It is worth noting that FPGAs support integer, floating-point, and fixed-point arithmetic. An FPGA implementation may require orders of magnitude higher resources, besides higher latency and power costs, when compared with a finely-tuned fixed-point implementation of the same algorithm (Finnerty and Ratigner, 2017). Predictably, PTQ impacts ML classification performance, although the profiling tools in *hls4ml* help the designer decide the appropriate model precision[6]. The resulting accuracy values for PTQ networks targeted for FPGA (with fixed-point precision) are shown in **Table 4**, and contrasted to those with floating-point precision in **Table 5**. We adopted quantization-aware training (QAT) to address this accuracy drop, as discussed in Section 3.4.

[4]In this study, accuracy is defined identically to signal efficiency, i.e., as a true positive classification rate given a set of true labels.

[5]Note that the CNN studies presented in this article are performed exclusively on non-empty ROIs. For images containing LE and HE events, ROI-finding does not cause any additional reduction in efficiency, and the ROI classification accuracy represents the signal efficiency. For images containing only NB, only one in approximately 50 images is kept after ROI-finding.

[6]*Profiling*. Available online at: https://fastmachinelearning.org/hls4ml/api/profiling.html (accessed January 2, 2022).

3.3. Automatized CNN Hyperparameter Optimization Using KerasTuner

In the initial network performance comparison presented in Section 3.2, the classification performance does not appear to be highly sensitive to the network architecture and number of trainable parameters. In general, the choice of network hyperparameters such as the dimensions of hidden layers, and learning parameters, changes the number of trainable variables. Thus, the quality of training can be modulated by tuning the hyperparameters using the training and validation samples. This can be cumbersome to optimize, but further optimization of networks with respect to a large phase-space of hyperparameters can be performed methodically and in an automated way using open-source tools such as KerasTuner (see text footnote[1]) (O'Malley et al., 2019).

We used KerasTuner for hyperparameter optimization for the baseline network architecture **CNN02-DS**. The scanning range and granularity of the hyperparameters explored is shown in **Table 6**. A total of twenty combinations were randomly sampled from the hyperparameter scanning region. As illustrated in **Table 7**, the optimized network **CNN02-DS-OP** with the (marginally) highest classification accuracy, found at 95.22%, corresponds to a network with a first convolution depth of 8, second convolution depth of 16, dense layer size of 12, and learning rate of 2.9×10^{-3}.

3.4. Network Quantization in CNN-Based Data Selection

The cost reduction and performance improvement of fixed-point arithmetic with HLS is highly encouraged when designing ML algorithms for FPGA deployment. Typically, when a trained network within an ML framework (e.g., Keras) on CPU or GPU is translated to HLS, the floating-point precision is reduced to the fixed-point precision of a given configuration. As a consequence, generally, network quantization resulting from fixed-point precision effectively reduces the precision of the calculations for weights, bias, and/or inputs, resulting in lower inference accuracy performance than what would otherwise be possible with floating-point precision. This is evident in **Table 5**.

In principle, one cannot achieve the flexibility and accuracy of a floating-point precision with any fixed-point representation. However, if accuracy can be maintained with an optimized choice of fixed-point precision, one can benefit from the inherent advantage of reduced computing resource utilization. Maintaining of accuracy therefore can be achieved with quantization-aware network training (Gong et al., 2014; Gupta et al., 2015; Han et al., 2016; Coelho et al., 2021; Hawks et al., 2021).

Quantization-aware training (QAT), achieved by committing calculations in ML algorithms with already-reduced fixed-point representation as part of network training, can prevent reduction in inference accuracy. The QKeras package[7] supports quantization-aware training by quantizing any given network using Qlayers. The quantized network derived from a given

[7]*QKeras*. Available online at: https://github.com/google/qkeras (accessed December 20, 2021).

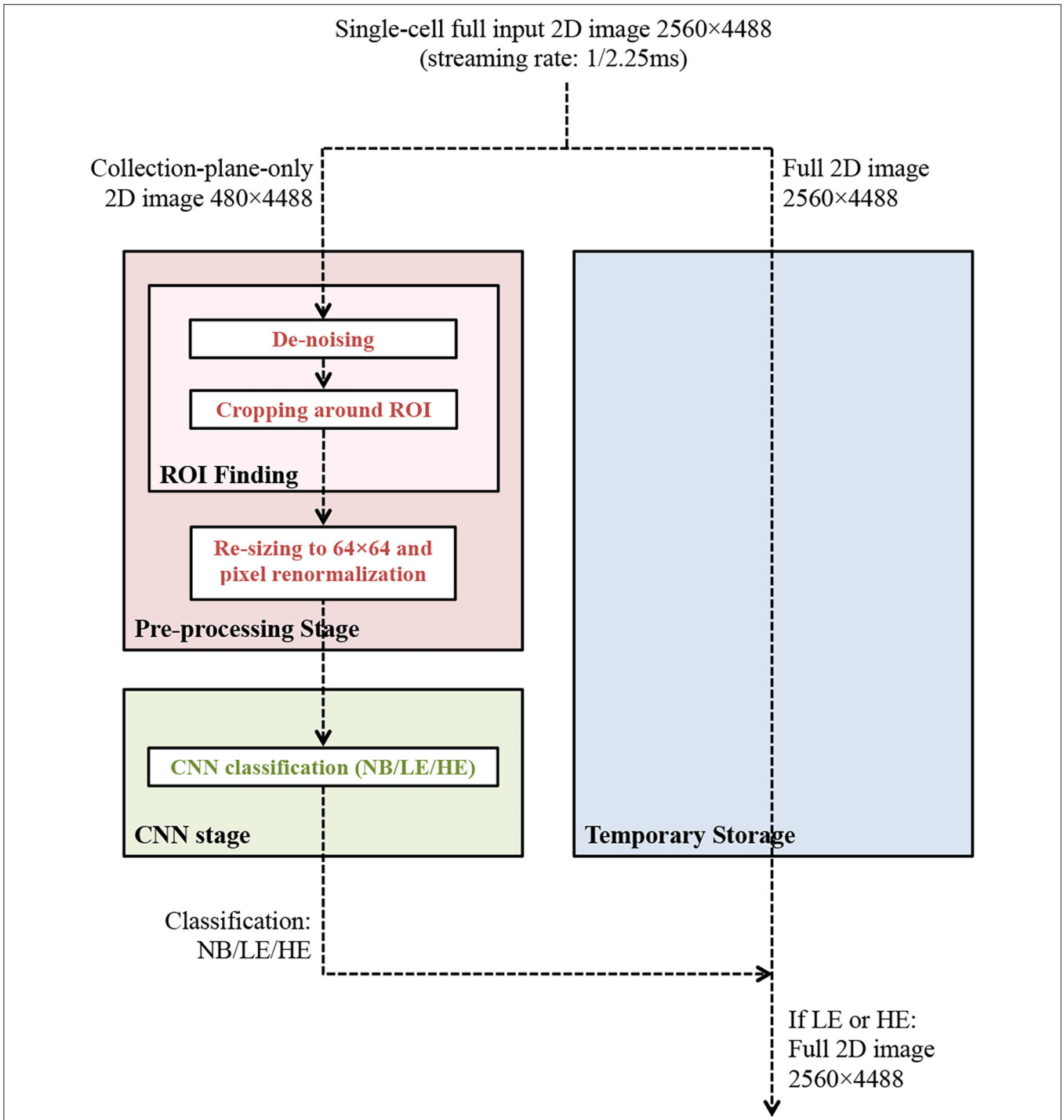

FIGURE 4 | The data processing and data selection scheme under study for potential implementation in the upstream DAQ readout units of the future DUNE FD. The streaming 2D input images contain, > 99.9% of the time, NB data. This overall scheme should select true HE and LE images with > 90% accuracy, and true NB images with > 99.99% accuracy, in order to meet the DUNE FD physics requirements. Additionally, the pre-processing and CNN inference algorithms should meet the computational resources of the DUNE FD upstream DAQ readout units, and the algorithm execution latency should meet the data throughput requirements of the experiment.

network architecture can be constructed by replacing the layers in the initial network to Qlayers. We refer to the quantized version of **CNN02-DS-OP** obtained with QKeras as **Q-CNN02-DS-OP**. The precision configuration of **Q-CNN02-DS-OP** is shown in **Figure 6**. The precision configuration of the reference **CNN02-DS-OP** is shown is shown in **Figure 7**.

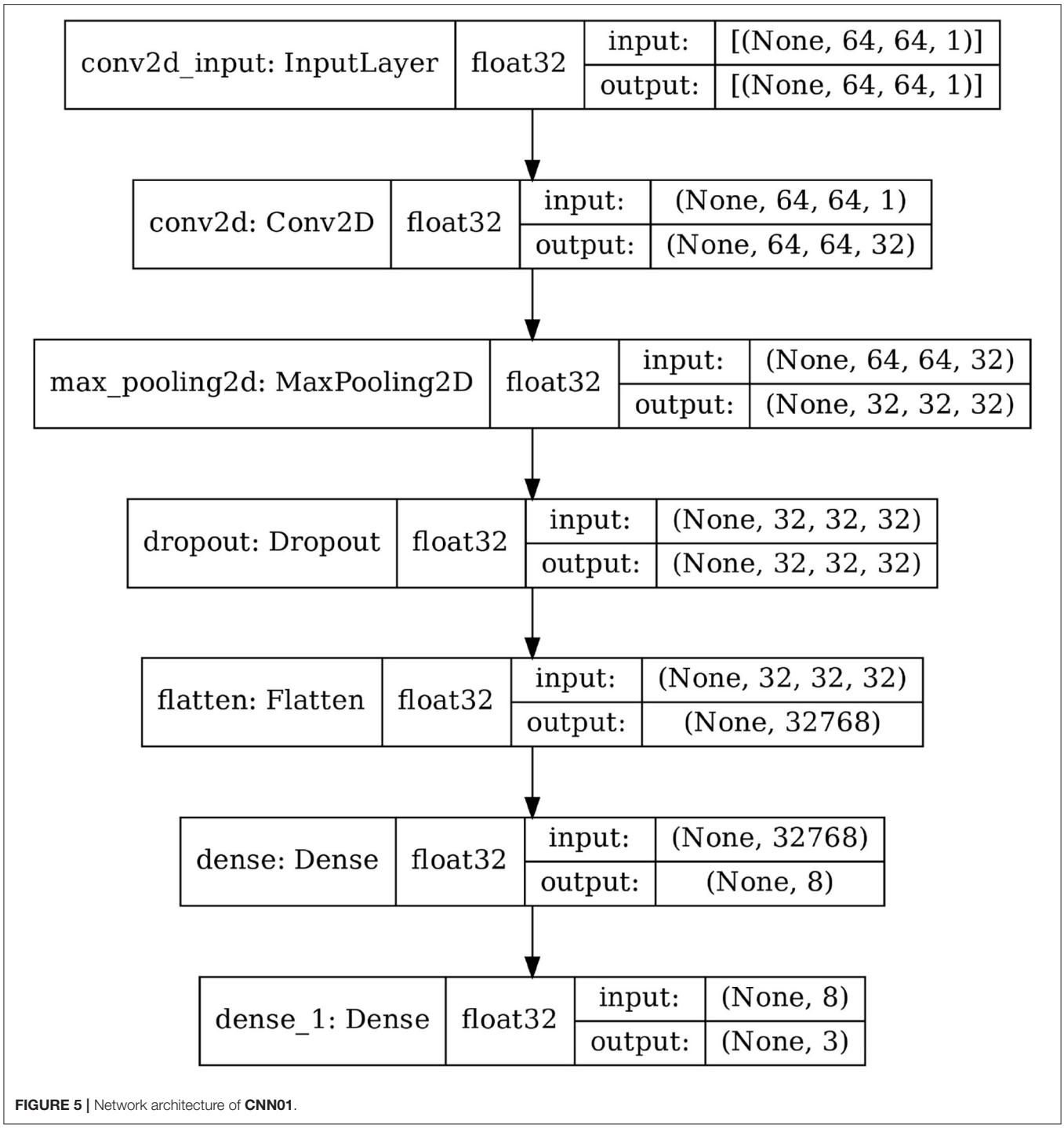

FIGURE 5 | Network architecture of **CNN01**.

The classification results obtained using the reference **CNN02-DS-OP** network with and without PTQ are shown in **Table 8**; the corresponding results obtained with the quantization-aware trained (QAT) **Q-CNN02-DS-OP** are shown in **Table 9**.

For the network trained without QAT, **CNN02-DS-OP**, the overall classification accuracy for the entire testing sample (superset of three truth labels) drops significantly with PTQ, from 95.4 to 72.4%. For the network trained with QAT, **Q-CNN02-DS-OP**, however, the overall classification accuracy is maintained for what would be an equivalent FPGA implementation (with PTQ), at 95.2 and 95.2%. This demonstrates that a relatively small CNN, applied on a frame-by-frame basis, and trained with quantization that is consistent with FPGA fixed-point precision, can achieve the accuracy (signal efficiency and target data reduction factor) required for the DUNE FD.

TABLE 2 | Summary of explored CNN architectures.

	CNN01	CNN02	CNN02-DS
Number of convolution layers	1	2	2
Convolution kernel dimension (first conv.)	3×3×32	3×3×32	3×3×4
Convolution kernel dimension (second conv.)	N/A	3×3×64	3×3×8
Number of max-pooling layers	1	2	2
Max-pooling dimension (first max-pool)	2×2	2×2	4×4
Max-pooling dimension (second max-pool)	N/A	2×2	4×4
Number of trainable parameters	262,499	149,923	1,395

TABLE 3 | Classification accuracy comparison for **CNN01**, **CNN02**, and **CNN02-DS** on GPU or CPU.

CNN01	**NB %**	**LE %**	**HE %**
True NB	99.4	0.55	0
True LE	3.8	94.2	1.9
True HE	3.4	6.1	90.5
CNN02	**NB %**	**LE %**	**HE %**
True NB	99.5	0.50	0
True LE	4.0	93.2	2.8
True HE	3.2	6.6	90.2
CNN02-DS	**NB %**	**LE %**	**HE %**
True NB	99.5	0.52	0
True LE	3.7	94.4	1.9
True HE	3.0	6.5	90.5

TABLE 4 | Classification accuracy comparison for **CNN01**, **CNN02**, and **CNN02-DS**, using post-training quantization (PTQ).

CNN01	**NB %**	**LE %**	**HE %**
True NB	98.1	1.8	0.02
True LE	6.6	89.1	4.3
True HE	19.4	37.7	42.9
CNN02	**NB %**	**LE %**	**HE %**
True NB	98.1	0.25	1.9
True LE	22.8	10.6	66.6
True HE	21.5	3.7	74.7
CNN02-DS	**NB %**	**LE %**	**HE %**
True NB	99.5	0.47	0
True LE	4.9	93.1	1.9
True HE	21.2	40.1	38.7

TABLE 5 | Combined classification accuracy for true NB, LE, and HE ROIs for floating-point vs. PTQ fixed-point implementations of the trained networks.

	CNN01 %	CNN02 %	CNN02-DS %
Floating-point accuracy	94.9	94.5	95.0
Fixed-point accuracy (PTQ)	78.5	60.7	79.1

*The combined classification accuracy is evaluated collectively on all of the testing set ROIs in **Table 1**, combined.*

TABLE 6 | Scanning range and granularity of the hyperparameters explored during automated network optimization using KerasTuner.

Hyperparameter	**Range**	**Default value**
First convolution depth (conv1)	[4, 8, 16]	4
Second convolution depth (conv2)	[8, 16, 32]	8
Dense layer size (fc)	[8, 12, 16, 20, 24]	12
Learning rate (lr), logarithmic sampling	$[2 \times 10^{-4}, 2 \times 10^{-2}]$	2×10^{-3}

TABLE 7 | Classification accuracy for the five top-performing and default (**CNN02-DS**) hyperparameter configurations.

	conv1	conv2	fc	lr	Accuracy (%)
First-best (**CNN02-DS-OP**)	8	16	12	2.9×10^{-3}	95.22
Second-best	16	32	12	4.9×10^{-4}	95.21
Third-best	4	16	20	6.0×10^{-4}	95.21
Fourth-best	16	8	16	7.0×10^{-4}	95.19
Fifth-best	16	8	12	1.9×10^{-3}	95.19
Default	4	8	12	2×10^{-3}	95.09

*Note that the default accuracy obtained during hyperparameter optimization slightly differs from that in **Table 5**, due to differences in (random) initialization of the network weights before training, and randomness during the training.*

4. ESTIMATION OF FPGA RESOURCE USAGE

In this section, we estimate FPGA resource usage and examine whether a Xilinx Virtex-7 UltraScale+ FPGA can accommodate a pre-trained CNN that meets the accuracy as well as resource and latency specifications of the DUNE FD DAQ and trigger system.

The estimated hardware usage for the quantized inference block of each of the optimized CNNs (**Q-CNN02-DS-OP** and **CNN02-DS-OP**) from Vivado HLS is shown in **Table 10**. The hardware usage of the discussed inference shows that the target FPGA, a high-end device, is well fit for implementing either the **Q-CNN02-DS-OP** or the **CNN02-DS-OP** network. As expected, the **Q-CNN02-DS-OP** network uses significantly lower FPGA resources. It is worth noting that, in addition to using more resources, **CNN02-DS-OP** (PTQ) has a lower accuracy than **Q-CNN02-DS-OP** (QAT), at 72.4% vs. 95.2%, illustrating the advantages of QAT.

Assuming a clock cycle of 5.00 ns, we find that the design is expected to meet timing requirements, with an inference latency of 4680 clock-cycles, corresponding to 23.4 μs. This is well below the exposure time corresponding to a single input image of 2.25 ms; thus, assuming sufficient parallelization (i.e., at least two input 2D images processed in parallel by each FELIX unit), frame-by-frame real-time data selection based on collection plane-only image analysis with CNNs is a viable solution for the DUNE FD. Note that this does not consider additional resource utilization or latency associated with image pre-processing (ROI finding and down-sizing).

We note that, in the current stage, the ML-based FPGA design has been synthesized, but it has not been implemented yet into the hardware; this is the focus of continuing development efforts.

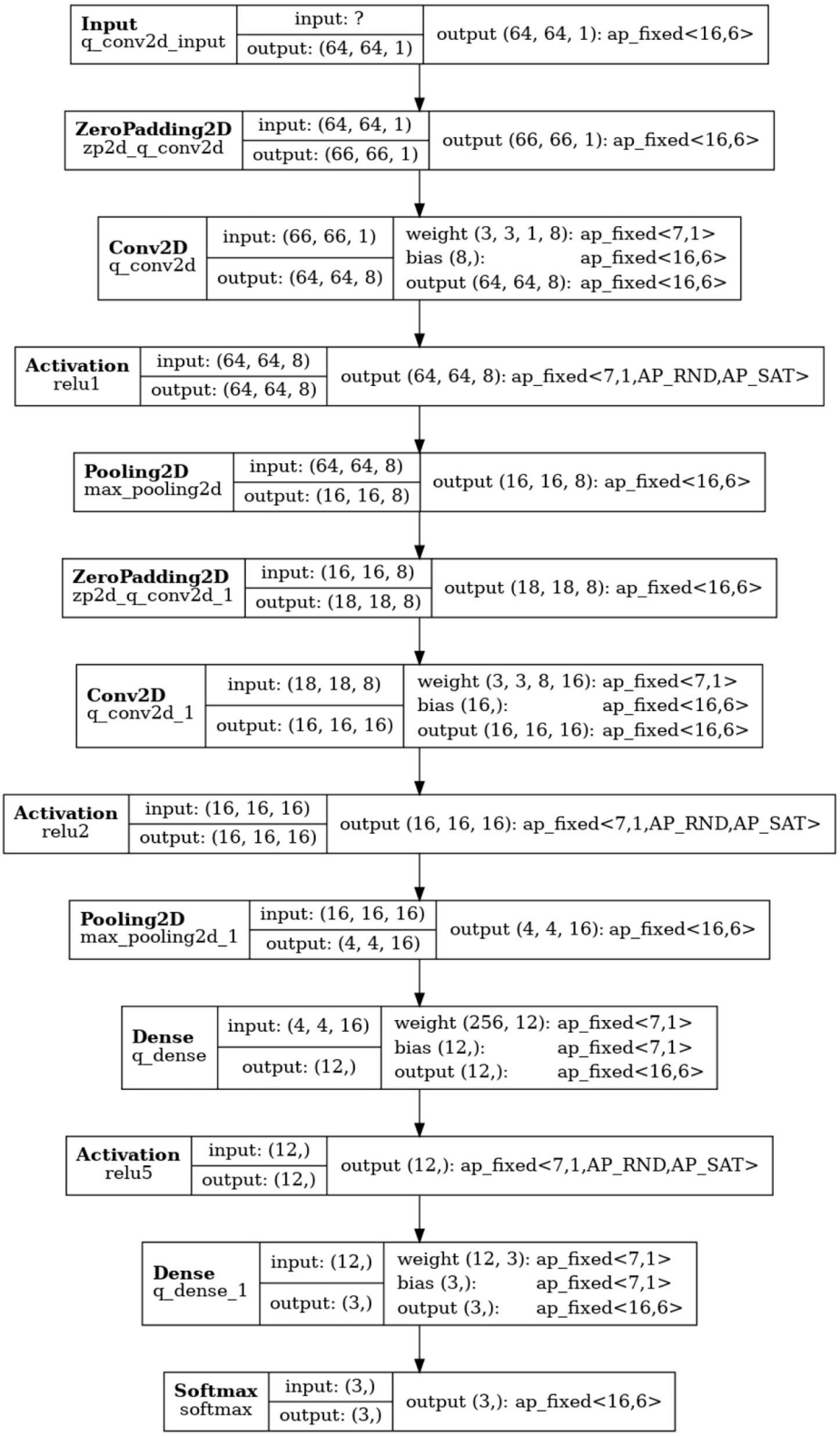

FIGURE 6 | Precision configuration of layers in **Q-CNN02-DS-OP**. The precision configuration of the reference **CNN02-DS-OP** can be found in **Figure 7**. Note that FPGA resource utilization is generally reduced with smaller ap_fixed values.

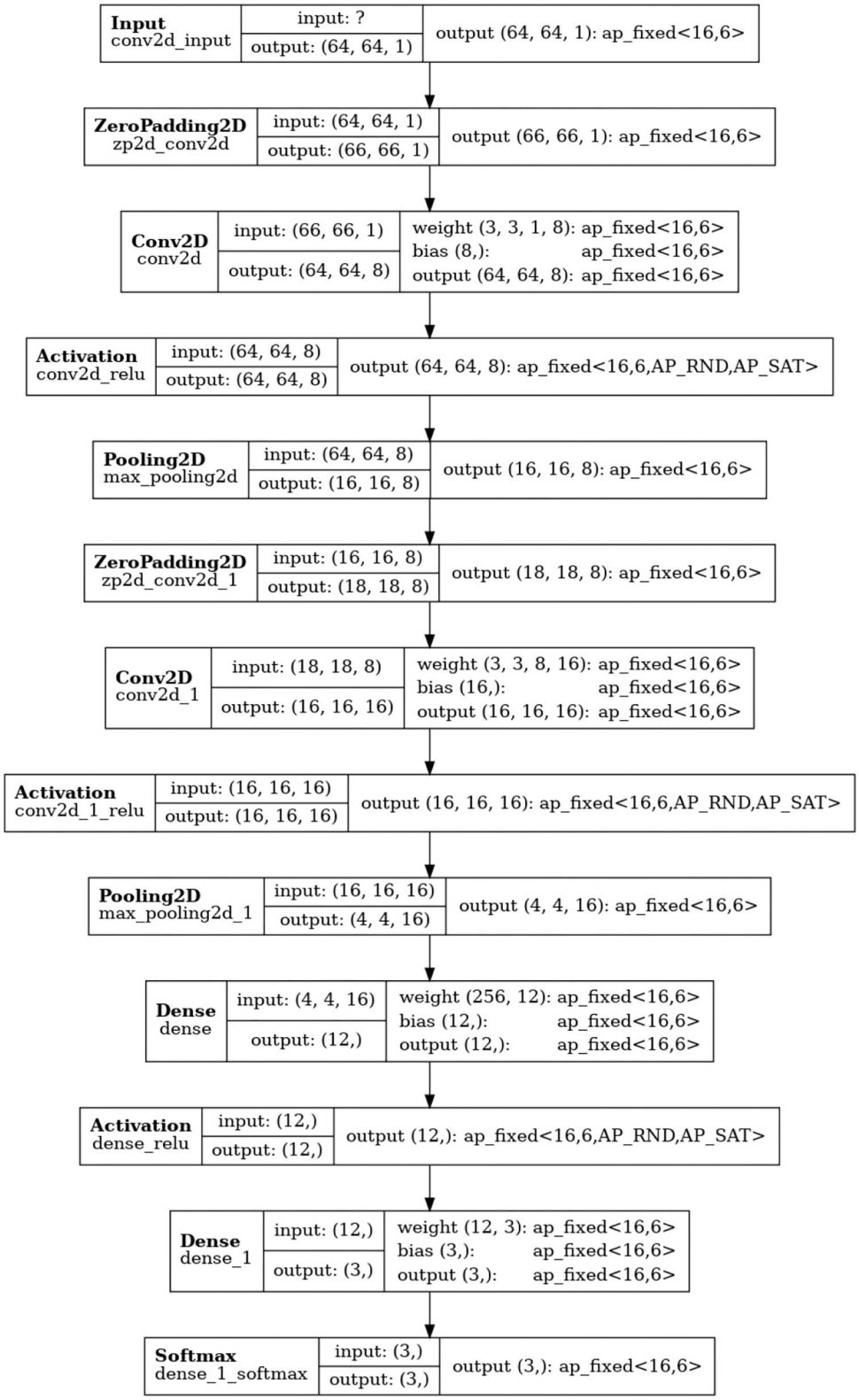

FIGURE 7 | Precision configuration of layers in the reference **CNN02-DS-OP**.

TABLE 8 | Optimized performance for **CNN02-DS-OP**, without quantization-aware network training.

Floating-point	**NB %**	**LE %**	**HE %**
True NB	99.5	0.50	0
True LE	3.5	95.1	1.4
True HE	2.9	6.1	91.0
Total accuracy		95.4	
Fixed-point (PTQ)	**NB %**	**LE %**	**HE %**
True NB	99.8	0.17	0
True LE	6.3	88.9	4.8
True HE	26.9	52.2	20.9
Total accuracy		72.4%	

*The difference in total accuracy for the floating-point case compared to that reported in **Table 7** is due to retraining.*

TABLE 9 | Optimized performance for **Q-CNN02-DS-OP**, with quantization-aware network training.

Floating-point (QAT)	**NB %**	**LE %**	**HE %**
True NB	99.6	0.40	0
True LE	3.8	94.0	2.2
True HE	3.2	5.4	91.4
Total accuracy		95.2	
Fixed-point (QAT)	**NB %**	**LE %**	**HE %**
True NB	99.7	0.32	0
True LE	3.9	94.7	1.4
True HE	3.2	6.4	90.4
Total accuracy		95.2%	

TABLE 10 | Estimated resource utilization from Vivado HLS for CNN inference on a Xilinx UltraScale+ (XCKU115) FPGA.

	Block RAM	**DSP units**	**Flip flops**	**Look-up tables**
Available	4,320	5,520	1326720	663360
CNN02-DS-OP (PQT)	331 (7%)	4,309 (78%)	226982 (17%)	163460 (24%)
Q-CNN02-DS-OP (QAT)	187 (4%)	2,106 (38%)	142128 (10%)	138715 (20%)

Block RAM refers to these types of memory elements, digital signal processors (DSPs) are elements dedicated to fast operations in signal processing (such as floating-point multiplication), Flip Flops and Look-up tables are standard.

5. SUMMARY

In recent years, ML algorithms such as CNNs have shown tremendous growth of their use in high energy physics, including physics analysis with LArTPCs (Radovic et al., 2018; Karagiorgi et al., 2021). In particular, CNNs have been shown to achieve very high signal selection efficiencies especially when employed in offline physics analyses of LArTPC data. MicroBooNE is leading the development and application of ML techniques, including CNNs, for event reconstruction and physics analysis as an operating LArTPC (Acciarri et al., 2017a; Adams et al., 2019; Abratenko et al., 2021a,b), and CNN-based analyses and ML-based reconstruction are actively being developed for SBN and for DUNE (Acciarri et al., 2012; Abi et al., 2020e).

Motivated by a previous study (Jwa et al., 2019), showing that CNN-based data selection for LArTPC detectors can yield excellent accuracy even when applied solely at raw collection plane data, we have proposed a real-time, 2D CNN-based, frame-by-frame data selection scheme that is found to be a viable solution for the DUNE FD DAQ and trigger system. Leveraging the extensive parallelization and FPGA resources available within the DUNE FD upstream DAQ readout design, in this proposed scheme, 2D image frames streamed at a total rate of 1.175 TB/s are pre-processed and run through hardware-accelerated CNN inference to classify and select interactions of interest on a frame-by-frame basis. The proposed pre-processing and CNN-based selection method yield target signal selection efficiencies that meet the DUNE FD physics requirements, while also providing the needed 10^4 factor of overall data rate reduction.

The FPGA resource utilization for the CNN inference has been optimized with automatized network optimization and with quantization-aware training so as to avoid accuracy loss due to a fixed-point precision implementation in FPGA. The resulting optimized and quantized CNN (**Q-CNN02-DS-OP**) has been shown to fit within available DUNE FD upstream DAQ readout FPGA resources, and to be executable with sufficiently low latency such that the need for significant buffering resources in the DUNE FD upstream DAQ system can also be relaxed. We note, however, that the pre-processing resource requirements and latency have not been explicitly evaluated, and this will be the subject of future work, as they need to be considered in tandem with the proposed CNN algorithm and implementation.

The findings further motivate future LArTPC readout designs that preserve the physical mapping of readout channels to a contiguous interaction volume as much as possible, in order to minimize pre-processing needs, and preserve spatial correlations that exist within 2D projected views of the interaction volume. Additionally, they motivate the consideration of other image analysis algorithms in the designs of DAQ and trigger systems of future LArTPCs.

DATA AVAILABILITY STATEMENT

The datasets presented in this article are not publicly available; re-use of image inputs used for CNN training and testing requires approval by the Deep Underground Neutrino Experiment (DUNE) Collaboration. Requests to access the datasets should be directed to GK, georgia@nevis.columbia.edu.

AUTHOR CONTRIBUTIONS

YJ was responsible for the design, training, testing, and optimization of CNNs, contributed to HLS simulations, and led the manuscript preparation. GDG was responsible for support and guidance on the HLS setup and simulations, and contributed to manuscript preparation. LA provided

help with HLS setup and simulations, and contributed to manuscript preparation. LC was responsible for research guidance on the HLS implementation. GK was responsible for the conception of the research project, including application use case(s), initiation of collaborative efforts, overall leadership and direction of the research activities, including funding support, and manuscript preparation. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by the National Science Foundation under Grant No. NSF-1914065.

REFERENCES

Aad, G., Berthold, A.-S., Calvet, T., Chiedde, N., Fortin, E. M., Fritzsche, N., et al. (2021). Artificial neural networks on FPGAs for real-time energy reconstruction of the ATLAS LAr calorimeters. *Comput. Softw. Big Sci.* 5, 19. doi: 10.1007/s41781-021-00066-y

Aarrestad, T., Loncar, V., Ghielmetti, N., Pierini, M., Summers, S., Ngadiuba, J., et al. (2021). Fast convolutional neural networks on FPGAs with hls4ml. *Mach. Learn. Sci. Tech.* 2, 045015. doi: 10.1088/2632-2153/ac0ea1

Abi, B., et al. (2021a). Prospects for beyond the standard model physics searches at the deep underground neutrino experiment. *Eur. Phys. J. C* 81, 322. doi: 10.1140/epjc/s10052-021-09007-w

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2020a). Deep underground neutrino experiment (DUNE), far detector technical design report, Volume I Introduction to DUNE. *arXiv preprint arXiv:2002.02967*. doi: 10.1088/1748-0221/15/08/T08008

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2020b). Deep underground neutrino experiment (DUNE), far detector technical design report, Volume II: DUNE Physics. *arXiv preprint arXiv:2002.03005*. doi: 10.48550/arXiv.2002.03005

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2020c). Deep underground neutrino experiment (DUNE), far detector technical design report, Volume III: DUNE Far Detector Technical Coordination. *arXiv preprint arXiv:2002.03008*. doi: 10.1088/1748-0221/15/08/T08009

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2020d). Deep underground neutrino experiment (DUNE), far detector technical design report, Volume IV: Far Detector Single-phase Technology. *arXiv preprint arXiv:2002.03010*. doi: 10.1088/1748-0221/15/08/T08010

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2020e). Neutrino interaction classification with a convolutional neural network in the DUNE far detector. *Phys. Rev. D* 102, 092003. doi: 10.1103/PhysRevD.102.092003

Abi, B., Acciarri, R., Acero, M. A., Adamov, G., Adams, D., Adinolfi, M., et al. (2021b). Supernova neutrino burst detection with the deep underground neutrino experiment. *Eur. Phys. J. C* 81, 423. doi: 10.1140/epjc/s10052-021-09166-w

Abratenko, P., Alrashed, M., An, R., Anthony, J., Asaadi, J., Ashkenazi, A., et al. (2021a). Convolutional neural network for multiple particle identification in the MicroBooNE liquid argon time projection chamber. *Phys. Rev. D* 103, 092003. doi: 10.1103/PhysRevD.103.092003

Abratenko, P., Alrashed, M., An, R., Anthony, J., Asaadi, J., Ashkenazi, A., et al. (2021b). Semantic segmentation with a sparse convolutional neural network for event reconstruction in MicroBooNE. *Phys. Rev. D* 103, 052012. doi: 10.1103/PhysRevD.103.052012

Acciarri, R., Adams, C., An, R., Asaadi, J., Auger, M., Bagby, L., et al. (2017a). Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber. *arXiv preprint arXiv:1611.05531*. doi: 10.1088/1748-0221/12/03/P03011

Acciarri, R., Adams, C., An, R., Asaadi, J., Auger, M., Bagby, L., et al. (2017b). Design and construction of the MicroBooNE detector. *arXiv preprint arXiv:1612.05824*. doi: 10.1088/1748-0221/12/02/P02017

Acciarri, R., Adams, C., Andreopoulos, C., Asaadi, J., Babicz, M., Backhouse, C., et al. (2012). Cosmic ray background removal with deep neural networks in SBND. *Front. Artif. Intell.* 4, 649917. doi: 10.3389/frai.2021.649917

Adams, C., Alrashed, M., An, R., Anthony, J., Asaadi, J., Ashkenazi, A., et al. (2019). Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection chamber. *Phys. Rev. D* 99, 092001. doi: 10.1103/PhysRevD.99.092001

Antonello, M., Baibussinov, B., Bellini, V., Benetti, P., Bertolucci, S., Bilokon, H., et al. (2015). A proposal for a three detector short-baseline neutrino oscillation program in the fermilab booster neutrino beam. *arXiv [Preprint]*. arXiv: 1503.01520. Available online at: https://arxiv.org/pdf/1503.01520.pdf

Aramaki, T., Hansson Adrian, P., Karagiorgi, G., and Odaka, H. (2020). Dual MeV gamma-ray and dark matter observatory - GRAMS project. *Astropart. Phys.* 114, 107–114. doi: 10.1016/j.astropartphys.2019.07.002

Borga, A., Church, E., Filthaut, F., Gamberini, E., de Jong, P., Miotto, G. L., et al. (2019). FELIX based readout of the single-phase protoDUNE detector. *IEEE Trans. Nuclear Sci.* 66, 993–997. doi: 10.1109/TNS.2019.2904660

Church, E. D. (2013). LArSoft: A software package for liquid argon time projection drift chambers. *arXiv [Preprint]*. arXiv:1311.6774v2. Available online at: https://arxiv.org/pdf/1311.6774.pdf

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., Schmidhuber, J. (2011). "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Vol. 2* (Barcelona), 1237–1242. doi: 10.5555/2283516.2283603

Coelho, C. N., Kuusela, A., Li, S., Zhuang, H., Ngadiuba, J., Aarrestad, T. K., et al. (2021). Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. *Nat. Mach. Intell.* 3, 675–686. doi: 10.1038/s42256-021-00356-5

Deina, A. M., Tran, N., Agar, J., Blott, M., Guglielmo, G. D., Duarte, J., et al. (2021). Applications and techniques for fast machine learning in science. *arXiv[Preprint]*. arXiv:2110.13041. Available online at: https://arxiv.org/pdf/2110.13041.pdf

Diotalevi, T., Lorusso, M., Travaglini, R., Battilana, C., and Bonacorsi, D. (2021). Deep learning fast inference on FPGA for CMS muon level-1 trigger studies. *PoS. ISGC2021:005*. doi: 10.22323/1.378.0005

Drielsma, F., Lin, Q., de Soux, P. C., Dominé, L., Itay, R., Koh, D. H., et al. (2021). Clustering of electromagnetic showers and particle interactions with graph neural networks in liquid argon time projection chambers. *Phys. Rev. D* 104, 072004. doi: 10.1103/PhysRevD.104.072004

Duarte, J., Han, S., Harris, P., Jindariani, S., Kreinar, E., Kreis, B., et al. (2018). Fast inference of deep neural networks in FPGAs for particle physics. *J. Inst* 13, P07027. doi: 10.1088/1748-0221/13/07/P07027

Duarte, J., Harris, P., Hauck, S., Holzman, B., Hsu, S.-C., Jindariani, S., et al. (2019). FPGA-accelerated machine learning inference as a service for particle physics computing. *Comput. Softw. Big Sci.* 3, 13. doi: 10.2172/1592124

Elabd, A., Razavimaleki, V., Huang, S. -Y., Duarte, J., Atkinson, M., DeZoort, G., et al. (2021). Graph neural networks for charged particle tracking on FPGAs. *arXiv[Preprint]*. arXiv: 2112.02048. Available online at: https://arxiv.org/pdf/2112.02048.pdf

Fahim, F., Hawks, B., Herwig, C., Hirschauer, J., Jindariani, S., Tran, N., et al. (2021). "HLS4ML: an open-source codesign workflow to empower scientific low-power machine learning devices," in *tinyML Research Symposium 2021*.

Finnerty, A., and Ratigner, H. (2017). Reduce Power and Cost by Converting from Floating Point to Fixed Point, *Xilinx White Paper, WP491 (v1.0)*. Available online at: https://docs.xilinx.com/v/u/en-US/wp491-floating-to-fixed-point

Gong, Y., Liu, L., Yang, M., and Bourdev, L. D. (2014). Compressing deep convolutional networks using vector quantization.

arXiv preprint arXiv:1412.6115. doi: 10.48550/arXiv.1412.6115

Govorkova, E., Puljak, E., Aarrestd, T., James, T., Loncar, V., Pierini, M., et al. (2021). Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider. *Nat. Mach. Intell*. 4, 154–161. doi: 10.1038/s42256-022-00441-3

Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. (2015). Deep learning with limited numerical precision. *arXiv preprint arXiv:1502.02551*. doi: 10.48550/arXiv.1502.02551

Han, S., Mao, H., and Dally, W. J. (2016). "Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016*, eds Y. Bengio and Y. LeCun (San Juan; Puerto Rico).

Hawks, B., Duarte, J., Fraser, N. J., Pappalardo, A., Nhan, T., and Yaman, U. (2021). Ps and Qs: Quantization-aware pruning for efficient low latency neural network inference. *Front. Artif. Intell*. 4, 676564. doi: 10.3389/frai.2021.676564

He, K., Zhang, X., Ren, S., and Sn, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Heintz, A., et al. (2020). "Accelerated charged particle tracking with graph neural networks on FPGAs," in *34th Conference on Neural Information Processing Systems*.

Iiyama, Y., et al. (2020). Distance-weighted graph neural networks on FPGAs for real-time particle reconstruction in high energy physics. *Front. Big Data* 3, 598927. doi: 10.3389/fdata.2020.598927

Jwa, Y.-J., Guglielmo, G. D., Carloni, L. P., and Karagiorgi, G. (2019). "Accelerating deep neural networks for real-time data selection for high-resolution imaging particle detectors," in *2019 New York Scientific Data Summit: Data-Driven Discovery in Science and Industry* (New York, NY). doi: 10.1109/NYSDS.2019.8909784

Karagiorgi, G., Kasieczka, G., Kravitz, S., Nachman, B., and Shih, D. (2021). Machine learning in the search for new fundamental physics. *arXiv preprint arXiv:2112.03769*. doi: 10.48550/arXiv.2112.03769

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/arXiv.1412.6980

Koh, D. H., Soux, P. C., Domine, L., Drielsma, F., Itay, R., Li, Q., et al. (2020). Scalable, proposal-free instance segmentation network for 3D pixel clustering and particle trajectory reconstruction in liquid argon time projection chambers. *arXiv [Preprint]*. arXiv:2007.03083. Available online at: https://arxiv.org/pdf/2007.03083.pdf

Loncar, V., et al. (2021). Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML. *Mach. Learn. Sci. Tech*. 2, 015001. doi: 10.1088/2632-2153/aba042

Mikuni, V., Nachman, B., and Shih, D. (2021). Online-compatible unsupervised non-resonant anomaly detection. doi: 10.1103/PhysRevD.105.055006

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). *Keras Tuner*. Available online at: https://github.com/keras-team/keras-tuner

Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D., Himmel, A., et al. (2018). Machine learning at the energy and intensity frontiers of particle physics. *Nature* 560, 41–48. doi: 10.1038/s41586-018-0361-2

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. a*rXiv[Preprint]*. arXiv:1409.1556. doi: 10.48550/arXiv.1409.1556

Summers, S., Di Guglielmo, G., Duarte, J., Harris, P., Hoang, D., Jindariani, S., et al. (2020). Fast inference of Boosted Decision Trees in FPGAs for particle physics. *J. Inst*. 15, P05026. doi: 10.1088/1748-0221/15/05/P05026

Trimberger, S. M. (2015). Three ages of FPGAs: a retrospective on the first thirty years of FPGA technology. *Proc. IEEE* 103, 318–331. doi: 10.1109/JPROC.2015.2392104

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX

A. TRAINING DETAILS

The Adam optimizer (Kingma and Ba, 2014) was used with learning rate 0.0029, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and ϵ=1e-8. During training, CNN models were "kept" if the validation accuracy was higher than already-kept highest-accuracy models. The optimized (best) model was found when the validation accuracy stopped improving. The accuracy values quoted in the main text were obtained with the test sample, and they are quoted to sufficiently high precision (as allowed by the statistics used) to assess whether the networks meet DUNE's accuracy requirements.

The training curve showing the training and validation accuracy for the **Q-CNN02-DS-OP**, as an example, is shown in **Figure A1**. The loss curve for the same network is shown in **Figure A2**.

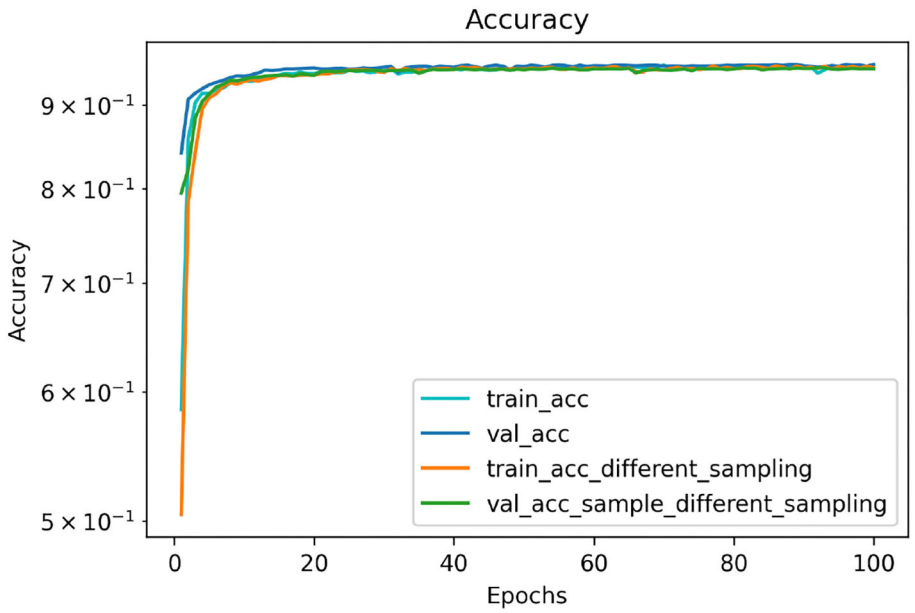

FIGURE A1 | The training and validation accuracy curves of **Q-CNN02-DS-OP**, in cyan and blue, respectively. The best model was found at epoch 88. The training and validation accuracy curves obtained using bootstrapping are overlaid in orange and green, respectively, with the best model found at epoch 92. This comparison further demonstrates that the uncertainty on the accuracy of the network is relatively low.

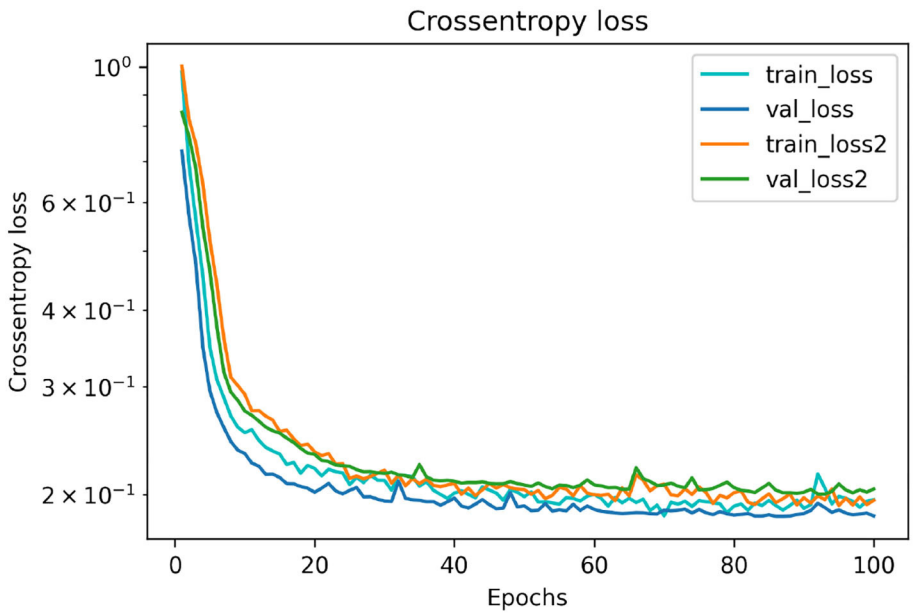

FIGURE A2 | Cross entropy loss of **Q-CNN02-DS-OP** for the training and validation samples, in cyan and blue, respectively. The best model was found at epoch 88. The cross entropy loss curves obtained using bootstrapping are overlaid in orange and green, respectively, with the best model found at epoch 92. This comparison further demonstrates that the uncertainty on the accuracy of the network is relatively low.