2025 28th Euromicro Conference on Digital System Design (DSD) | 979-8-3315-8499-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/DSD67783.2025.00031

AIDA4Edge: Twinning for Excellence in Adaptive Edge Artificial Intelligence

Marko Andjelkovic, Rizwan Tariq Syed, Alessandro Veronesi, Fabian Vargas, Markus Ulbricht, Leticia Bolzani Poehls, Milos Krstic
IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany
{andjelkovic, syed, veronesi, vargas, ulbricht, poehls, krstic}@ihp-microelectronics.com

Davide Bertozzi, Edward G. Jones, Oliver Rhodes
University of Manchester, Manchester, United Kingdom
{davide.bertozzi, edward.jones-3, oliver.rhodes}@manchester.ac.uk

Riccardo Zese, Michele Favalli, Alice Bizzarri, Evelina Lamma, Marco Gavanelli, Elena Bellodi
University of Ferrara, Ferrara, Italy
{riccardo.zese, michele.favalli, alice.bizzarri, evelina.lamma, marco.gavanelli, elena.bellodi}@unife.com

Zoran Peric, Jelena Nikolic, Milan Dincic, Aleksandra Jovanovic, Dejan Ciric, Nikola Vucic, Sofija Peric, Jelena Jovanovic, Milica Stojanovic, Tatjana Nikolic, Goran Nikolic, Jelena Nedeljkovic, Danijel Dankovic, Emilija Zivanovic, Milos Marjanovic, Sandra Veljkovic, Nikola Mitrovic, Bratislav Predic, Tamara Milovanovic
Faculty of Electronic Engineering, University of Nis, Nis, Serbia
{zoran.peric, jelena.nikolic, milan.dincic, aleksandra.jovanovic, dejan.ciric, nikola.vucic, sofija.peric, jelena.jovanovic, milica.stojanovic, tatjana.nikolic, goran.nikolic, jelena.nedeljkovic, danijel.dankovic, emilija.zivanovic, milos.marjanovic, sandra.veljkovic, nikola.mitrovic, bratislav.predic}@elfak.ni.ac.rs, tamaramilovanovic@elfak.rs

***Abstract*— The growing demand for deployment of Artificial Intelligence (AI) on resource-constrained edge devices has motivated extensive research on the design of efficient edge-compatible AI hardware accelerators. One of the most promising solutions are the self-adaptive AI accelerators, capable of optimizing in real time their performance and energy consumption according to application requirements. This work introduces the EU-funded project *Twinning for Excellence in Adaptive Edge Artificial Intelligence* (*AIDA4Edge*), aimed to advance the state-of-the-art in the design of adaptive neural network accelerators for edge applications. The main goal is to develop a novel hybrid self-adaptive neural network architecture combining spiking and artificial neural networks, and supporting runtime adaptation of network functionality, precision and reliability. Furthermore, we aim to enhance the neural network training by incorporating hardware and quantization constraints in an automated tuning engine.**

Keywords—Artificial neural networks, spiking neural networks, hybrid neural networks, adaptive neural networks, neural network accelerators, hyperparameter optimization, quantization, reliability of neural networks

I. Introduction

Artificial Intelligence (AI) algorithms are currently one of the most powerful tools for solving challenging cognitive problems, becoming one of the leading research topics and one of the areas with the largest investment. The integration of AI algorithms into edge devices, referred to as Edge-AI, enables to perform complex data processing directly on local hardware devices, as close as possible to the physical process, bringing the benefits in terms of enhanced performance, reliability and security. With the advancement in AI algorithms, the Edge-AI offers the possibilities for a significant breakthrough in a wide range of applications, e.g., computer vision, audio and speech processing, automated driving, 5G/6G communications, aerospace, industry automation, digital healthcare, etc.

However, along with numerous benefits and potential new applications, implementation of AI algorithms on edge devices is accompanied with many design challenges. Powerful AI algorithms, such as Deep Neural Networks (DNN), impose high processing, memory and power requirements, rendering their implementation on resource-constrained edge devices very challenging. For example, the ResNet-50 requires over 95 MB of memory to store the weights and more than 3.8 billion multiplications to classify a single image [1]. In general, increase in performance of DNNs leads to an increase in complexity of their hardware implementation, resulting in an increase in the power consumption [1, 2]. General-purpose processors are not capable of handling complex AI algorithms with high performance. While graphics processing units (GPU) can provide the required computational capacity, they consume a lot of power. Therefore, edge-oriented AI algorithms are typically implemented on custom-designed hardware platforms known as AI hardware accelerators.

In order to allow efficient execution of complex AI models on resource-constrained edge devices, the design complexity of AI hardware accelerators should be reduced as much as possible. To this end, various techniques such as quantization, pruning, knowledge distillation, and approximate computing have been proposed. However, all these techniques come with the reduction in accuracy, which may be critical in many applications. Moreover, in most cases these techniques are applied

979-8-3315-8499-3/25/$31.00 ©2025 IEEE
DOI 10.1109/DSD67783.2025.00031

Table 1: An overview of related work on adaptive neural networks (NNs)

Reference(s)	Description of adaptation mechanisms
[9, 10]	*Adaption of operating settings*: Dynamic voltage and frequency scaling can achieve over 50 % reduction in power consumption of neural networks. However, operation at reduced supply voltage increases the likelihood of hardware faults.
[11-13]	*Adaptation of data precision*: Reprogramming of neural network weights at runtime may be applied to adjust the precision of data representation according to application requirements. Reducing precision leads to a reduction in power consumption. Moreover, reduced precision may contribute to better fault tolerance.
[14, 15]	*Adaptation of neural network architecture to enhance performance*: Reconfiguration of hardware architecture in order to change the NN model or functionality of NN accelerator. Typical approaches include layer skipping and early exit mechanisms.
[16, 17]	*Adaptation of neural network architecture to enhance reliability*: Runtime activation of redundant hardware resources is a common approach to enhance fault tolerance in dynamic environments. Redundancy could be applied at different abstraction levels.
[18, 19]	*Multi-domain adaptation:* Performing reconfiguration across multiple parameters may provide a fine-grained control of neural network operation. For example, varying quantization, supply voltage and clock frequency may be applied to control precision and performance [18]. Alternatively, reconfigurable multi-core accelerator could be used to maintain a tradeoff between performance, reliability and power consumption [19].

in a static manner, without considering that in real application scenarios AI accelerators may be subjected to dynamic application requirements, as well as varying operating conditions. In such cases, traditional hardware accelerators with static architecture (fixed functionality) would not be efficient in terms of energy consumption and performance.

For safety-critical domains, such as healthcare, automotive, banking, industrial automation, and aerospace, an additional requirement for AI accelerators is reliability. Special design measures should be applied to enhance the robustness to both transient and permanent faults, which may cause unrecoverable data loss, malfunction, system failure, or even loss of lives. For example, in autonomous vehicles, as a result of soft errors a truck may be misclassified as a bird [3], which could result in an accident if the driver does not intervene. Faults in AI accelerators may also reduce the classification accuracy [4, 5]. While traditional redundancy-based fault tolerance techniques can improve reliability, they come at the cost of increased power consumption. Furthermore, static fault tolerance strategies are often suboptimal in dynamic real-world applications, where reliability requirements may vary over time.

A promising solution for AI applications with varying operating conditions and application requirements are the self-adaptive AI accelerators. Self-adaptive accelerators are capable of dynamically adjusting their functionality in response to real-time workload demands and environmental changes [6, 7]. This flexibility enables to maintain in real time a tradeoff between performance (latency), computational accuracy, energy consumption and reliability.

Design of self-adaptive AI accelerators is an open research topic. Despite numerous published works, there is no a comprehensive design methodology for self-adaptive neural network accelerators. In this work we present the EU-funded Horizon Europe Twinning project AIDA4Edge [8], aimed to solve some of the key challenges in the design of self-adaptive AI neural network hardware accelerators. The project is currently in its first year, and this work summarizes the main concept and plans for research work.

The rest of the paper is organized into four sections. In Section II, the state-of-the-art in self-adaptive neural network accelerators is summarized. Section III introduces briefly the AIDA4Edge project. In Section IV, the joint research project is briefly introduced. The paper is concluded in Section V.

II. Overview of the State-of-the-Art in Adaptive Neural Networks

Adaptive neural network accelerators (also referred to as self-adaptive neural networks, dynamic neural networks, self-aware neural networks, or reconfigurable neural networks) have emerged as a promising solution to address the dynamic computational demands of modern AI systems, especially in power- and resource-constrained edge applications. By dynamically adjusting the neural network architecture, data representation format and operating settings (supply voltage and operating frequency), it could be possible to achieve a fine-grained runtime control of performance (latency), accuracy, power consumption, and reliability. A summary of adaptation methodologies [9-19] for neural networks is given in Table 1.

A general concept of an adaptive neural network accelerator is illustrated in Fig. 1. To support self-adaptivity, the target system (neural network accelerator) should be equipped with additional resources for monitoring the system state and decision making. Various monitoring mechanisms can be implemented on the chip to track the changes in operating parameters (supply voltage, temperature), monitor the task execution, and detect hardware faults. Data is collected from monitors and processed to evaluate the system state. Based on the system's state and the application requirements, the system is reconfigured in real time to meet the new application demands.

While adaptive neural network accelerators offer benefits in terms of improved efficiency and flexibility of AI hardware, several technical challenges remain. One of the main limita-

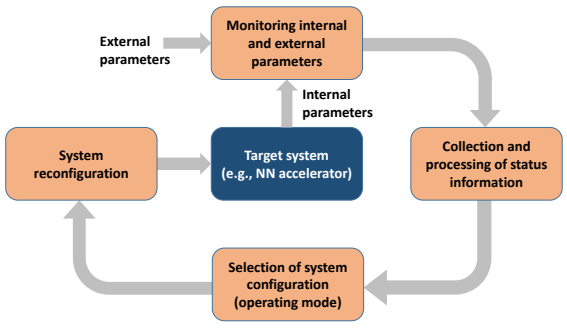

Fig. 1. A general concept of a self-adaptive neural network accelerator

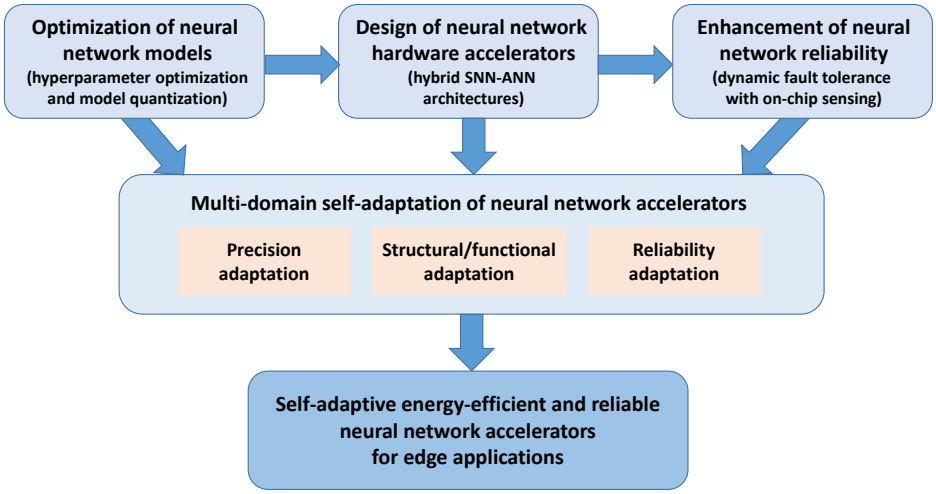

Fig. 2. AIDA4Edge scientific concept for self-adaptive energy-efficient and reliable neural network accelerators for edge applications

tions of existing solutions is that usually a single adaptation scenario is considered. As can be seen in Table 1, most works have analyzed the adaptivity in a single domain, such as adaptation of operating settings, precision adaptation, adaptation of accelerator architecture to optimize performance or adaptation of accelerator architecture to enhance reliability. Only few recent works [18, 19] have considered adaptation across multiple details, which could bring the benefits in better utilization of resources.

Another important design aspect is the complexity of adaptation mechanisms. Namely, additional hardware resources for self-monitoring and reconfiguration may impose high overhead. Dynamically modifying system functionality requires complex control logic, which can introduce significant latency, area, and power penalties. In time-sensitive applications, such as real-time image processing or autonomous navigation, this overhead can outweigh the benefits of adaptation. To the best of our knowledge, there is no much work on the design of scalable adaptation mechanisms that will produce minimum impact on the operation of target accelerator.

Driven by the aforementioned challenges, our ultimate goal is to explore novel design solutions, in particular through the multi-domain adaptation, in order to improve the overall performance of adaptive neural network accelerators.

III. AIDA4Edge Overall Concept

AIDA4Edge is a 3-year Horizon Europe Twinning project, running from October 2024 to September 2027. The main goal of Twinning projects is to support an institution from a developing European country (EU member state or associated state) in enhancing its scientific competences, where that institution also serves as the project coordinator. At least two partners from two different developed EU countries, referred to as "advanced partners", are involved in the project to provide know-how transfer to the project coordinator. In AIDA4Edge, the project coordinator is the Faculty of Electronic Engineering, University of Nis (Serbia), while the advanced partners are IHP - Leibniz Institute for High Performance Microelectronics (Germany), University of Manchester (United Kingdom) and University of Ferrara (Italy).

The main topic of AIDA4Edge project is related to *adaptive neural network accelerators for edge applications*. The scientific methodology, illustrated in Fig. 2, is focused on exploring design solutions for energy-efficient and reliable neural network accelerators through runtime adaptivity. Combining the complementary expertise of all four partners, the AIDA4Edge project addresses three main topics:

- **Design and optimization of neural network models**. The primary focus is on design of efficient neural network models through hyperparameter optimization and model quantization.
- **Design of neural network accelerators.** The focus is on design of neural network hardware accelerators, such as Spiking Neural Networks (SNN) and classical Artificial Neural Networks (ANN), and their deployment on commercial FPGA platforms.
- **Fault-tolerant solutions for neural network accelerators.** The aim is to analyze the impact of design constrains on the reliability of neural networks, and explore various low-cost fault-tolerant solutions.

One of the main directions in each of the three aforementioned topics is related to investigating approaches for runtime adaptivity of neural network accelerators, thus enabling to maintain in real-time a balance between power consumption, performance, accuracy and reliability. In AIDA4Edge project, we aim to investigate multi-domain adaptation mechanisms, such as adaptation of data precision (by online reconfiguration of neural network weights), adaptation of neural network functionality (by online modification of neural network architecture) and reliability adaptation (by activating fault-tolerant mechanisms on demand).

The core activities are focused on knowledge exchange and joint research with respect to the aforementioned scientific topics. In addition, the project also addresses the knowledge transfer in relation to research management and administration. Fig. 3 illustrated the overall AIDA4Edge methodology based on 5 main components implemented in the form of 5 distinct work packages (WPs):

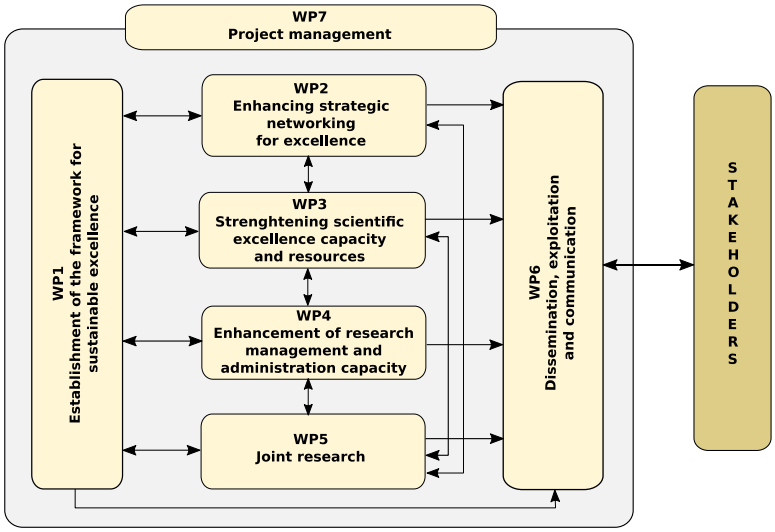

Fig. 3. AIDA4Edge work packages and their interrelations

- **WP1: Establishing the framework for sustainable excellence**. This WP focuses on establishing conditions for long-term collaboration between the project partners and the interested stakeholders.
- **WP2: Enhancing strategic networking for excellence.** This WP aims to enhance the networking between the project partners and interested stakeholders through organization of networking events.
- **WP3: Strengthening scientific excellence capacity and resources.** This WP deals with knowledge sharing between the project partners, implemented through staff exchanges.
- **WP4: Enhancing research management and administration capacities.** The aim of this WP is to enhance the competences in acquisition and management of collaborative research projects.
- **WP5: Joint research.** This WP implements a join research project, combing the competences of all partners, with the aim to advance the state-of-the-art in the design of adaptive neural network accelerators.

IV. Joint Research Project: A Hybrid Adaptive Neural Network Accelerator for Vision Applications

The research part of AIDA4Edge project, executed through WP5, aims to solve some of the key issues related to the design of energy-efficient and reliable self-adaptive neural network accelerators for resource-constrained edge platforms. In particular, our goal is to establish a comprehensive methodology for the design of adaptive neural network accelerators, considering different flavors of runtime adaptivity, thus enabling to achieve fine-grained optimization in terms of performance, power consumption, accuracy, and reliability.

As a case study, we aim to validate the developed solutions for video monitoring applications, i.e., for object recognition and motion detection. The solutions will be implemented on a commercial FPGA platform from Xilinx, and the validation will be performed in a laboratory environment.

A. Background

Video monitoring has become vital across various domains, including security, healthcare, transportation, and industrial automation. The integration of AI in video monitoring applications significantly enhances their effectiveness by enabling real-time analysis, anomaly detection, and intelligent decision-making without constant human oversight. AI-powered systems can automatically identify suspicious behavior, track individuals or objects, recognize faces, and detect safety violations, making surveillance more efficient. The combination of video monitoring with AI algorithms has the potential to transform traditional surveillance into smart, adaptive systems capable of learning and evolving over time.

Conventional Edge AI systems for video monitoring are based on an artificial neural network (ANN) fed by data from classic frame-based cameras that sample each pixel of the entire image with a certain sampling frequency (frame rate). In practice, only a small part of the image is dynamically active and changes between two frames, while the majority of pixels are dynamically inactive. Sampling all pixels with the same rate leads to over-sampling of dynamically inactive pixels generating a huge amount of redundant data, but also to undersampling of dynamically active pixels causing a motion blur. An ANN must be very complex to be able to process this huge amount of data (from each pixel in each frame), which is a fundamental reason why it consumes a large amount of energy and is very demanding for implementation on edge platforms.

A potential solution to the aforementioned problem is to replace classical frame-based cameras with event-based cameras. Event-based cameras provide a low-power alternative to frame-based cameras for capturing dynamic scenes. They are biologically inspired to solve the problem of undersampling and oversampling. Event-based cameras encode local pixel-wise brightness changes in asynchronous streams of events rather than image frames and yield sparse, energy-efficient encodings of scenes, in addition to low latency, high dynamic range, and

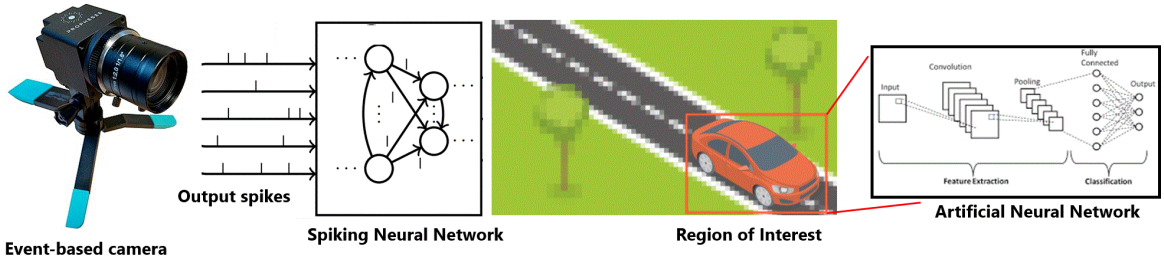

Fig. 4: Proposed concept of combined SNN-ANN accelerator for vision applications employing event-based camera

lack of motion blur, representing an excellent solution for video monitoring applications.

However, connecting event-based cameras to ANNs is challenging as it requires the conversion of a stream of events into pixel values, which is a very complex task for a large number of pixels. The solution to this problem comes from the emerging brain-inspired SNNs that offer compelling features. SNNs are ideal for processing the output of event-based cameras because they share a fundamental event-driven nature, making them highly compatible with the sparse, temporal, and asynchronous data produced by such cameras. Moreover, due to their event-driven nature and the use of spikes for data transfer, SNNs are promising more energy-efficient computation and communication. At the same time, with current technology ANNs are better suited for object recognition due to their ability to learn complex hierarchical features, their efficient training algorithms (e.g., backpropagation), and their scalability with large datasets.

Thus, the combined SNN-ANN approach utilizes the SNN as a natural match for event-based sensory data pre-processing, while the ANN can take advantage of more stable training dynamics and faster convergence, resulting in greater task performance. The goal is to achieve ANN-like performance at a drastically-reduced power budget.

B. Proposed Solution – A Hybrid Adaptive Neural Network Accelerator Combining SNN and ANN

We propose a hybrid neural network model that combines SNNs and ANNs in a hybrid AI system, taking advantage of the best of both concepts, and leveraging their synergy to develop an ultra-low-power edge-compatible processing pipeline for video monitoring applications. The proposed concept is based on the expertise of the University of Manchester in SNNs and the University of Ferrara in ANNs, and it is depicted in Fig. 4.

The hybrid SNN-ANN approach is a unique new advance over the state-of-the-art, where design and optimization of the whole event-based acquisition and processing pipeline is still in the early stage, and synergies between SNNs and ANNs for the sake of ultra-low power operation have not been extensively investigated. To the best of our knowledge, several prior works have analyzed the hybrid SNN-ANN architectures for vision applications [20-22], but they have not addressed the multi-domain adaptation scenarios.

This dual-stage pipeline uses the SNN for rapid region-of-interest identification and activation rate control of ANN, while the ANN handles deeper analysis. Therefore, the input temporal information can be effectively extracted by leveraging the asynchronous computing capabilities of SNN, while the ANN enables trouble-free training and implementation on standard machine learning hardware.

The basic idea of the proposed approach is to use the event-based camera to generate asynchronously a series of events and feed them to the SNN. The SNN analyzes the events and determines the image portion that is dynamically active at any given point in time. Then, only the events corresponding to pixels in that small "region of interest" of the image are converted into pixel values (such conversion is not an issue since the number of pixels is small). The pixel values are then fed to the ANN for inference. Thus, the ANN processes only the pixels determined by the SNN, enabling to capture the dynamics of the input. Also, the SNN may decide at which frequency the ANN should be activated, for instance at high frequency in high-dynamic-range scenes.

By pursuing the efficient integration of ANN and SNN and mitigating their individual shortcomings, the proposed processing pipeline offers the following key advantages:

a) As ANN is intended to process only a small part of the image, it can be of low-complexity, consuming a small amount of energy.
b) As SNN is intended to control the ANN instead of independently solving the problem of video monitoring, it can also be of very low complexity, overcoming the historical concern associated with their training at large scale.

As a result, the hybrid processing pipeline supports a two-fold adaptivity dimension:

- ***Adaptivity in space***, since the SNN points to the most interesting part of an image to be further processed by the ANN.
- ***Adaptivity in time***, since the SNN can understand when the scene is changing quickly and tune the inference rate accordingly through direct control over ANN activation. Otherwise, the inference rate can be kept low.

In addition to the innovative hybrid SNN-ANN concept with spatial and temporal adaptivity in terms of processing of input data, we also plan to investigate two additional adaptivity domains for ANN:

- ***Adaptivity in terms of data precision*** by runtime reprogramming of ANN weights.
- ***Adaptivity in terms of reliability*** through ANN self-awareness and runtime reconfiguration.

To support the adaptivity of ANN in terms of data precision, we will investigate various low-bit quantization models for ANN weights. In terms of reliability, we will investigate the on-

chip sensing and runtime reconfiguration for a multi-core ANN accelerator. Additionally, to enhance the neural network training process, we will investigate innovative approaches for automated tuning of neural network hyperparameters, taking into consideration the impact of hardware constraints and low-bit quantization.

As a proof of concept, we aim to implement the SNN in software that will be executed on a hard processor embedded in FPGA, while the ANN will be developed as a separate digital block and implemented on the same FPGA.

The following provides an overview of the planned research with respect to tuning of hyperparameters and architecture, low-bit quantization and reliability of neural networks.

C. *Tuning of Hyperparameters and Architecture of ANN*

Hyperparameters in neural networks are the settings defined before training that control the learning process and model architecture. Some of the key hyperparameters include learning rate, number of epochs, number of layers and neurons per layer, and activation function. Proper tuning of hyperparameters is essential for achieving high performance of neural network model and avoiding issues like underfitting or overfitting.

One of the main research objectives is to improve the performance of automated hyperparameter tuning. The basis for this research is the SymbolicDNN-Tuner developed by the University of Ferrara [23, 24]. SymbolicDNN-Tuner is an advanced framework for automated and efficient exploration of hyperparameters and neural network architecture. Unlike conventional manual tuning approaches, which require substantial computational resources and expertise, SymbolicDNN-Tuner leverages Probabilistic Logic Programming (PLP) rules to drive the tuning process systematically. The tuner integrates Bayesian Optimization (BO) with a novel rule-based approach, where Symbolic Tuning Rules (STRs) are used to dynamically modify the hyperparameter search space, introduce new hyperparameters, and adjust network structure in response to specific training problems detected during the previous optimization iterations. Each STR is associated with a Tuning Action (TA) which has the purpose of updating the search space values or neural network architecture, guiding the search to a better model. The probability that each TA will be applied is updated constantly over tuning iterations, as the system gathers more training evidence, through Learning from Interpretation (LFI), a specialized form of Inductive Logic Programming (ILP). This adaptive approach enables SymbolicDNN-Tuner to refine its optimization strategies over time, ultimately leading to a more efficient and resource-effective ANN tuning process. The workflow of the SymbolicDNN-Tuner is illustrated in Fig. 5 and a more detailed description can be found in [23, 24].

The first research direction towards improvement of hyperparameter tuning is related to the integration of hardware-aware Neural Architecture Search (NAS) into SymbolicDNN-Tuner. This approach aims to address the real-world constraints such as inference latency, power efficiency, and hardware compatibility. Traditional NAS methods optimize architectures primarily for accuracy, but in practical deployments, factors such as computational efficiency and hardware adaptability play a pivotal role. To this end, the proposed framework will integrate new modules to evaluate model performance under specific hardware constraints. A hardware profiler, such as NVDLA, will be used to compute statistics such as inference latency or

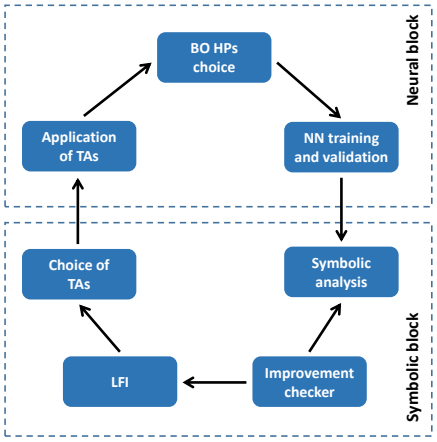

Fig. 5: Workflow for neural network model optimization using SymbolicDNN-Tuner

energy consumption for a given model configuration, providing insights into how the ANN performs on different hardware setups. Additionally, the study will incorporate a cost function tied to the physical area of the device. This metric serves as an additional optimization constraint, ensuring that the final model is not only computationally efficient but also cost-effective for deployment on hardware-constrained platforms.

Once the optimized architecture and hyperparameters are defined, the ANN undergoes training using publicly available image datasets relevant for applications in object recognition and motion detection, offering real-world scenarios to validate the effectiveness of the proposed tuning framework. Following training, we will study the application of a low-bit quantization schema aimed at reducing computational complexity without significantly impairing inference accuracy. These steps are crucial for deploying the trained model on edge devices and resource-constrained environments, where memory and computational efficiency are critical considerations.

By adding low-bit quantization for efficient inference and hardware-aware NAS to the existing SymbolicDNN-Tuner's automated optimization, this research phase aims to develop a comprehensive approach for deep learning model optimization. The proposed methodology should ensure that neural networks are not only high-performing but also computationally feasible and adaptable to real-world deployment scenarios, making them suitable for a broad range of Edge-AI applications.

D. *Low-bit Quantization of ANN and SNN*

To reduce computational complexity and memory footprint of neural network accelerators, and thus reduce also the power consumption, low-bit quantization of neural network weights is applied. Typically, the quantization converts the standard 32-bit floating-point representation into fixed-point or integer representation with fewer bits (e.g., 16, 8 or fewer bits). However, quantization degrades the performance of neural networks in terms of lowered Signal-to-Quantization-Noise-Ratio (SQNR), as well as a narrowed range of data variance, eventually resulting in reduced inference accuracy. Moreover, quantization may significantly affect the hardware implementation of the neural network. Therefore, careful selection of low-bit digital formats is crucial, requiring parameter optimization and thorough per-

formance evaluation. Leveraging the expertise in quantization of the Faculty of Electronic Engineering, University of Nis, various aspects of low-bit quantization, and the impact of low-bit quantization techniques on SNN and ANN performance, will be investigated.

One of the key challenges is establishing the relationship between SQNR and ANN accuracy. Based on prior research by the team from the University of Nis, it has been observed that accuracy can tolerate a reduction in SQNR up to a certain threshold without significant loss. However, when SQNR falls below this threshold, accuracy experiences a sharp decline. It is evident that the dependence between SQNR and accuracy varies depending on the specific ANN architecture and dataset. To date, this issue has been explored for a relatively small number of ANN models. Our objective is to expand this research to a broader range of ANN models and establish more general conclusions.

The research on quantization effects will address both post-training quantization and quantization-aware training, allowing for comparing both approaches on the same benchmarks. As target models, both SNN and ANN will be considered. In terms of quantization-aware training, the possibility of integrating the quantization techniques in the SymbolicDNN-Tuner presented in previous section will be investigated. Additionally, the project will explore the combined application of quantization-aware training and post-training quantization, where quantized para-meters are used during training and further quantization is applied during inference to simplify deployment on edge.

Recent studies have demonstrated a significant potential of quantization to enhance the reliability of ANN accelerators [25, 26]. As quantization reduces the amount of data to be stored in memory, the number of possible fault locations is also reduced, thus improving the overall fault-tolerance. However, several studies have shown that the probability of critical faults is likely to increase with quantization [27]. Therefore, one of the core research directions of the project is to analyze the impact of various quantization techniques on ANN reliability. Moreover, the joint influence of quantization on both accuracy and reliability will be investigated.

Additionally, we will explore the hardware implementation of adaptive quantization techniques capable of dynamically adapting to real-time constraints, such as system complexity, accuracy, and reliability, by selecting the most suitable quantization techniques on-the-fly. To this end, we will investigate approaches for efficient storage of neural network weights and switching between different quantization techniques.

E. Reliability of ANN

To address the requirements of safety-critical applications, an important research goal is to investigate design solutions for reliable neural networks. Given that our use case scenario, i.e., video monitoring, is essential in many safety-critical applications, it is important to investigate techniques for enhancing the reliability of ANN, while also taking into account the performance and energy constraints.

Similarly as for any digital system, reliability of a neural network accelerator can be enhanced by the insertion of redundancy in the design phase. Typical examples are Dual Modular Redundancy (DMR), Triple Modular Redundancy (TMR), and Error Detection and Correction (EDAC) codes. However, these solutions incur significant overhead in terms of area and power consumption, which is in contrast to the main requirements of edge AI solutions. Therefore, adaptive fault-tolerant solutions are essential for edge-oriented neural networks.

A key aspect of research on reliable ANN will be related to the design of a self-aware and self-adaptive multi-core ANN accelerator. Multi-core neural network accelerators are gaining more attention in the context of edge applications because they offer a compelling balance between performance and energy efficiency, enabling complex AI workloads to run on the edge [28-30]. Moreover, multi-core architectures allow for parallel processing of data from multiple inputs (sensors), which is a common requirement in edge systems. Additionally, due to their inherent redundancy, i.e., existence of multiple processing cores, the multi-core platforms are very convenient for implementing low-cost fault-tolerance measures.

Fig. 6 illustrates the conceptual architecture of a self-aware and self-adaptive multi-core neural network accelerator. The system may consist of multiple ANN accelerators with multiple memory blocks to store ANN parameters and data, as well as multiple processing cores to control the ANN and run the application program. To enable self-awareness, each hardware unit should be equipped with monitors (sensors) for detecting faults and tracking the changes in system state. Information from all monitors is collected regularly and stored in a separate memory. A dedicated controller is employed to process the data from all monitors and initiate the required reconfiguration.

The proposed concept is based on the initial design of a multi-core neural network accelerator introduced in previous work of IHP [19]. This solution employs three neural network accelerators, which can be configured at runtime in three main operating modes:

- *High-performance*: All accelerators operate in parallel, each executing its own task.
- *Fault-tolerance*: Multiple accelerators are arranged in N-modular redundancy settings, such as DMR or TMR, providing enhanced fault tolerance.
- *Power consumption and aging reduction*: One or more accelerators are switched off to reduce power consumption and aging, while the operating accelerators can be set into high-performance or fault-tolerance mode.

In general, the sensors for monitoring the parameters that effect reliability (e.g., supply voltage, temperature and work-

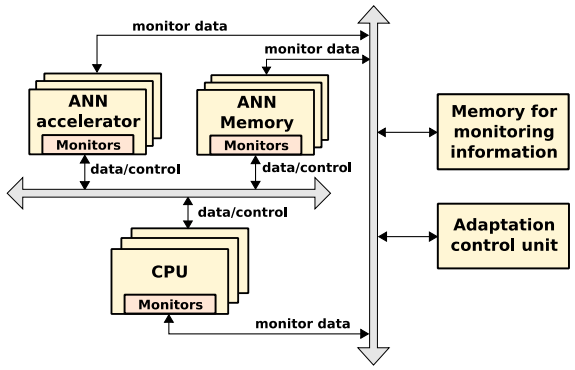

Fig. 6: Design concept for a multi-core neural network accelerator with self-awareness and self-adaptivity for maintaining a trade-off between performance, reliability and power consumption

load) and sensors for detecting transient and permanent faults are needed. The sensing logic should occupy minimal area and introduce minimal power overhead. The sensors should also be fault-tolerant. To this end, we will investigate low-power, low-complexity and fault-aware on-chip sensing techniques. We will also investigate strategies for efficient placement of on-chip sensors, in order to achieve maximal fault coverage with minimal number of on-chip sensors.

Furthermore, we will investigate the techniques for efficient storage and processing of sensing data, and subsequent system reconfiguration. The goal is to ensure that the storage and processing units are fault-tolerant and induce minimal overhead. In addition, the communication between on-chip sensors and corresponding processing logic should not influence the performance of the neural network accelerator. For processing of sensing data and decision on system reconfiguration, we will study the applicability of lightweight machine learning models implemented on custom hardware accelerators.

V. Conclusion

In this work, the EU-funded Twinning project AIDA4Edge is presented. As the project is in its first year, we introduce the scientific concept and general plan for joint research. The project addresses the design solutions for edge-compatible adaptive neural network accelerators, with focus on hybrid accelerator architectures, automated exploration of hyperparameters and architectures, low-bit quantization, and design for reliability. The goal is to advance the state-of-the-art in the design of self-adaptive, energy-efficient and reliable neural network accelerators through a multi-domain adaptation scheme. As a case study, we aim to develop a hybrid SNN-ANN accelerator for vision applications employing the event-based camera.

Acknowledgment

This work has received funding from the Horizon Europe project AIDA4Edge (Grant Agreement No. 101160293). The project is also supported by the United Kingdom Research and Innovation organization.

References

[1] A. Marchisio et al., "Deep Learning for Edge Computing: Current Trends, Cross-Layer Optimizations, and Open Research Challenges," in Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019.

[2] R. Machupalli et al., "Review of ASIC Accelerators for Deep Neural Networks," Microprocessors and Microsystems, vol. 89, 2022.

[3] G. Li et al., "Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications," In Proc. SC, the International Conference for High-Performance Computing, Networking, Storage and Analysis (SC17), 2017.

[4] L.-H- Hoang et al., "FT-ClipAct: Resilience Analysis of Deep Neural Networks and Improving their Fault Tolerance using Clipped Activation," in Proc. Design Automation and Test in Europe Conference (DATE), 2020.

[5] J. J. Zhang et al., "Analyzing and Mitigating the Impact of Permanent Faults on a Systolic Array Based Neural Network Accelerator," in Proc. VLSI Test Symposium (VTS), 2018.

[6] Z. Du et al., "Self-Aware Neural Network Systems: A Survey and New Perspective," Proceedings of the IEEE, 2020.

[7] Y. Han et al., "Dynamic Neural Networks: A Survey," IEEE Transactions on Pattent Analysis and Machine Intelligence, vol. 44, 2022.

[8] https://aida4edge.elfak.rs/

[9] B. Moons, R. Uytterhoeven, W. Dehaene and M. Verhelst, "14.5 Envision: A 0.26-to-10TOPS/W Subword-Parallel Dynamic-Voltage-Accuracy-Frequency-Scalable Convolutional Neural Network Processor in 28nm FDSOI," in Proc. IEEE Int. Solid-State Circuits Conference (ISSCC), 2017.

[10] S. Liu, A. Karanth, "Dynamic Voltage and Frequency Scaling to Improve Energy-Efficiency of Hardware Accelerators," in Proc. Int. Conference on High Performance Computing, Data and Analytics (HiPC), 2021.

[11] Q. Jin, L. Yang, Z. Liao, "AdaBits: Neural Network Quantization with Adaptive Bitwidths," in Proc. Int. Conference on Computer Vision and Pattern Recongnition (CVPR), 2020.

[12] Z. Song et al., "DRQ: Dynamic Region-based Quantization for Deep Neural Network Acceleration," in Proc. Annual Int. Symposium on Computer Architecture (ISCA), 2020.

[13] G. D. Guglielmo et al., "A Reconfigurable Neural Network ASIC for Detector Front-End Data Compression at the HL-LHC," IEEE Transactions On Nuclear Science, 2021.

[14] J. Yu, T. Huang, "Universally Slimmable Networks and Improved Training Techniques," in Proc. Int. Conference on Computer Vision (ICCV), 2019.

[15] S. Laskaridis et al., "Adaptive Inference throguh Early Exit Networks: Design, Challenges and Directions," in Proc. Annual Int. Workshop on Embedded and Mobile Deep Learning (EMDL), 2021.

[16] K. Khalil et al., "Self-Healing Approach for Hardware Neural Network Architecture," in Proc. International Midwest Symposium on Circuits and Systems (MWCAS), 2019.

[17] N. Cherezova et al., "FORTALESA: Fault-Tolerant Reconfigurable Systolic Array for DNN Inference," in arXiv, 2025.

[18] F. Tu et al., "Evolver: A Deep Learning Processor with On-Device Quantization-Voltage-Frequency Tuning," IEEE Journal of Solid State Circuits, vol. 56, 2020.

[19] R. S. Tariq et al., "FPGA Implementation of a Fault-Tolerant Fused and Branched CNN Accelerator with Recofigurable Capabilities," IEEE Access, vol. 12, 2024.

[20] A. Kugele et al., "Hybrid SNN-ANN: Energy Efficient Classification and Object Detection for Event-Based Vision," Pattern Recognition. DAGM GCPR 2021.

[21] A. Aydin et al., "A Hybrid ANN-SNN Architecture for Low-Power and Low-Latency Visual Perception," in Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2024.

[22] A. Kosta et al., "ANN vs SNN vs Hybrid Architectures for Event-based Real-time Gesture Recognition and Optical Flow Estimation," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2023.

[23] M. Fraccaroli et al., "Symbolic DNN-Tuner: A Python and ProbLog-based system for Optimizing Deep Neural Networks Hyperparameters," SoftwareX, vol. 17, 2022.

[24] M. Fraccaroli et al., "Symbolic DNN-Tuner," Machine Learning, 2022.

[25] B. F. Goldstein et al., "Reliability Evaluation of Compressed Deep Learning Models," in Proc. Latin American Symposium on Circuits and Systems (LASCAS), 2020.

[26] R. T. Syed et al., "Fault Resilience Analysis of Quantized Deep Neural Networks," in Proc. International Conference on Microelectronics (MIEL), 2021.

[27] F. Libano et al., "Understanding the Impact of Quantization, Accuracy, and Radiation on the Reliability of Convolutional Neural Networks on FPGAs," IEEE Transactions on Nuclear Science, vol. 67, no.7, 2020.

[28] A. Symons et al., "Towards Heterogeneous Multi-core Accelerators Exploiting Fine-grained Scheduling of Layer-Fused Deep Neural Networks," in arXiv, 2022.

[29] K. Balaskas et al., "Heterogeneous Accelerator Design for Multi-DNN via Heuristic Optimization," IEEE Embedded Systems Letters, vol. 16, 2024.

[30] P. Houshmand et al., "DIANA: An End-to-End Hybrid DIgital and ANAlog Neural Network SoC for the Edge," IEEE Journal of Solid State Circuits, vol. 58, 2023.