HW/SW Co-Design for Integrated AI Systems: Challenges, Use Cases and Steps Ahead

Tanja Harbaum, Iuliia Topko, Alexey Serdyuk, Iris Fürst-Walter, Fabian Kreß, Jürgen Becker

HW/SW Co-Design for Integrated AI Systems: Challenges, Use Cases and Steps Ahead

Tanja Harbaum*, Iuliia Topko* Alexey Serdyuk*, Iris Fürst-Walter*, Fabian Kreß* and Jürgen Becker*
*Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
Email: {harbaum, iuliia.topko, alexey.serdyuk, fuerst, fabian.kress, becker}@kit.edu

***Abstract*—Artificial Intelligence (AI) applications are permeating large parts of the economy, science and society. Enabled by recent advances in algorithms, computer architectures and big data, AI has made significant breakthroughs in a wide range of applications. Not only in the field of computer vision, speech recognition and processing, but also in robotics and many other areas. The trend in many domains is to shift intelligence from the cloud to the edge. However, integrating AI systems in the edge requires the development of powerful solutions under very strict resource constraints. Often, there already exist solutions delivering very good results in the high-performance area, but the use of AI approaches in embedded systems is still a major challenge. Although effective hardware/software co-design methods have been studied for decades, they are still not the standard in the development of complex embedded systems, including integrated AI systems. This paper discusses the open research challenges to realize such a holistic design space exploration for a hardware/software co-design for integrated AI systems. A novel approach for this holistic view of the hardware solutions is introduced, which is intended to evaluate accuracy, latency, as well as energy, and resource requirements. In Addition, this paper provides insights into the use case of an intelligent learning device for automated handwriting, which demonstrates an extremely resource-constrained environment.**

***Index Terms*—Embedded Systems, AI accelerator, hardware/software co-design**

I. Introduction and Motivation

The demands on microarchitectures are constantly increasing and the type of challenges has also changed in recent years. In Addition, most of the breakthrough technologies of recent years were only made possible by increasing the performance of integrated circuits and a further increase is no longer self-evident. New architectures have to be designed in order to meet the increasing demands at this point. With the heterogeneity of these new architectures and the huge amount of data that now needs to be processed, machine learning approaches are increasingly being used. Also, for the development of new Cyber Physical System (CPS) and Internet-of-Things (IoT) products, AI is becoming an increasingly important factor. The performance of embedded systems can be increased enormously by integrating AI algorithms that are adapted to the embedded hardware. Through a hardware/software co-design, a fast and efficient AI execution on embedded systems can be realized. Based on the use case of an intelligent learning device for automatic handwriting, we highlight the challenges of such integrated AI systems in more detail in section III.

With the increasing use of mobile and IoT devices, the integration of AI systems has become necessary in order to meet the ever-increasing demands of these devices [1]. The trend is to move intelligence from the cloud to the edge [2]. The shift to integrated AI systems, on the other hand, requires the development of powerful solutions under very strict resource constraints. On the other hand, there often already exist solutions that deliver very good results in the high-performance area, but the use of AI approaches in embedded systems is still a major challenge [3]. Neural Architecture Search (NAS), a technique for automating the design of artificial neural networks, is becoming increasingly relevant [4]. So far, the focus has mostly been on *accuracy* and the effects on the hardware (properties such as *latency*, *energy consumption* or *chip area*) are rarely considered in more detail or in combination. While effective hardware/software co-design methods have been researched for decades, they are still not the standard in the development of complex embedded systems, including integrated AI systems [2]. One major challenge is that the hardware, software and AI domains cannot simply be replaced or adopted for other scenarios. In the current approaches of NAS, the best solution is found for a fixed data set [4], a change of this data is not foreseen. *Federated Learning (FL)* adopts a different approach, models are trained using various local data sets. There are first approaches to combine these paradigms [5]. But, the influence of the local data sets on the later hardware architecture is not considered yet. Found solutions usually work very well for learned data sets, but as soon as this environment changes, the models quickly lose *accuracy* [6]–[8]. While there are numerous online learning approaches, these cannot usually be implemented in integrated AI systems and there is no impact on the hardware implementation. This leads to a set of research questions that need to be considered: What impact can different data environments have on today's common NAS approaches and the resulting hardware platforms? How can NAS and FL be combined and knowledge distilled from them that can be used to benefit new data-hardware combinations? Can FL be a tool to use transfer learning to be able to derive conclusions about high-performance hardware architectures? Can a multi-criteria approach, considering a trade-off between model quality and hardware, lead to new solutions that have not yet been considered? These questions inevitably lead to further questions: Does the execution of an appropriate assessment also extend to the edge devices or is a purely offline assessment sufficient? How can two data sets be meaningfully compared with each other? How can edge devices be adapted in use? In order to answer these questions, a holistic view of the hardware

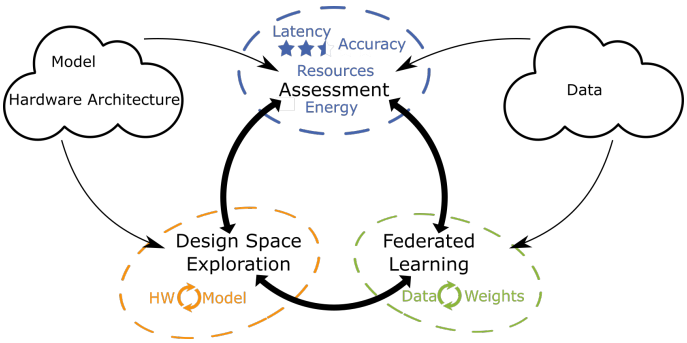

Fig. 1. Holistic approach for a data-centered HW/SW co-design for integrated AI systems.

solutions is necessary, which includes accuracy, latency and energy and resource requirements. In section II, we present a holistic approach that extends Design Space Exploration (DSE) with FL approaches to integrate the data sets into the exploration. In section III, we present the initial implementation steps of the proposed holistic approach into low-power systems. The paper is concluded by the section IV, which also provides a brief preview of future work.

II. Data-centric HW/SW Co-Design for integrated AI systems

We propose a holistic HW/SW Co-Design for integrated AI systems that extends DSE with FL approaches to integrate the data sets into the exploration. An overview is given in Figure 1. A design space is created that is composed of the three dimensions of data, model and hardware architecture. We also want to develop a multi-criteria evaluation that makes it possible to compare different solutions with each other in a suitable way and, if necessary, to identify suitable HW model combinations for a specific environment. The aim is to create a holistic framework, which is described in more detail in subsection II-B.

A. Fundamentals

1) Design Space Exploration: The design space exploration for integrated AI systems is at the intersection of the classical design space exploration for embedded systems and the methods of AutoML, which are used for the search of optimal neural network architectures. Therefore, the design space of integrated AI systems includes the design spaces of both domains. Embedded systems are usually designed specifically for an application and consist of an optimized hardware-software solution. Important criteria for such a solution include memory requirements, energy efficiency and latency. Several solutions are often considered as candidates and then a selection is made based on the pareto optimum [9]. An essential factor for a successful exploration of possible solutions is the choice of strategy with which the design space is searched [9]. There are different approaches, in principle, a compromise is always made between the required accuracy of the evaluation and affordable effort. As described, the design space of integrated AI systems combines several design spaces. Searching in this space is therefore a particular challenge, as the dimensionality of this space increases while the space of potential solutions shrinks. In addition, integrated AI systems are used in dynamic environments where the data can change. A change in the data affects the design space and solutions that previously seemed valid may become invalid. There are several NAS frameworks for integrated AI systems that combine NAS approaches using reinforcement learning [10], [11] or Bayesian optimization [12], [13] to manage the search in this complex design space. In [12], the verification of possible architectures is performed using Hardware-in-the-Loop (HiL) approaches, which only measure the latency of the inference and the memory utilization, but not the energy consumption. In [14], an approach is used to implement a data-aware NAS in which the search space for the Hyperparameter Optimization (HPO) is linked to certain data set parameters. However, the correlation between different data sets is not evaluated and the approach only works for predefined data sets.

None of the previous work can adequately estimate the HW resources and the associated parameters such as energy requirements, chip area and latency during the design space exploration. In addition, the generated solutions are fixed to one data set and there are no approaches to distill knowledge from known solutions in order to use this for new solutions with other data sets.

2) Federated Learning: Machine Learning (ML) approaches are applied in various different fields and deployed on versatile devices, such as IoT sensors, AI accelerators, CPUs, GPUs, computer clusters. In most of these cases, a standard ML deployment process consists of the following steps: collecting and preparing data on a centralized server, choosing and training a model on the server, evaluating and deploying the model. Deploying a model on embedded devices is a challenging task and demands more additional steps. After getting a ML model into the real world, accuracy might drop down [6]–[8] and the model might behave abnormally. That means the target dataset of the current environment is significantly different from the source dataset. One of the possible solutions to address this issue is to use Domain Adaptation techniques. In the standard ML approach new data should be gathered on the central server again and the model retrained in order to improve the model prediction. However, with an increasing number of devices deployed in the environment, loading all new data to the server becomes infeasible. Moreover, data might be personal or proprietary and meant not to be shared nor stored in one place. A new form of ML training to address this issue is called FL. FL is a distributed ML paradigm used for decentralized training on a large number of endpoints, where each end-device stores data locally and collaboratively learns a shared predictive model. Continuous training of the shared model has not stopped, as long as all participating devices extract common knowledge to achieve higher accuracy of the global model. Nevertheless, smartphones, IoT and embedded devices are still heavily constrained in the training capabilities. Open-source framework FedML [15] offers on-device training on smartphones and cross-cloud GPU servers, but for the rest

constrained devices integration of the FL paradigm remains unsolved.

Most NAS techniques start searching for a solution by designing a search space that incorporates all possible neural network architectures, i.e. start the search from scratch without using the previously explored solution. *Meta-learning* or *learning to learn* [16] aims at learning new algorithms by leveraging information from previous experience [17]. Similar technique can be applied to improve the performance of a global model on a target domain by using the knowledge learned by the model from another related domain. Transfer Learning (TL) and Domain Adaptation (DA), in particular, aim at extracting knowledge from one domain and transferring to another one. In general, ML models consist of layers, where different layers learn different features. Initial layers compile basic features of a dataset, while the later layers focus more on explicit tasks. Then they are connected to the last layer to generate the output. In this case, the DA can be implemented as follows: copy the entire trained model from the source domain, freeze the first few layers, since they learn basic features that are general mostly to all types of data. Then retrain or fine-tune the rest top layers to adapt these specialized features for the new dataset.

Integrated AI systems have so far mostly been reduced to their dedicated use without being able to react significantly to changes. By extending integrated AI systems with the FL paradigm and TL approaches, it is possible to implement update mechanisms for ML models.

3) Assessment for AI models and integrated AI systems: Up to now, benchmarking for AI models has primarily focused on model quality, i.e. how high the prediction quality (*Model Quality*) of a model is. This is a measure of how close the predictions of the ML model come to the actual results (*Ground Truth*). The model quality is compared uniformly for different models on the same data set. However, the hardware performance is usually not specified, so that the suitability for use in integrated AI systems cannot be estimated. The benchmarks ImageNet [18], [19], NuScenes [20], ApolloScape [21] and Cityscapes [22] were evaluated for this purpose. There are not only benchmarks for AI models, but also for accelerators. Here, hardware performance is largely limited to latency only, and energy efficiency is rarely considered. However, this is particularly important for integrated AI systems in order to be able to assess usability and autonomy. MLPerf [23] provides AI models and clear rules for comparing similar systems: There are several subcategories, e.g. *Train* and *HPC* for training high-performance computers to achieve a given model quality and *Edge*, *Mobile* and *Tiny* for comparing the inference of accelerators. However, the focus is one-sided either on model quality (e.g. classification: Accuracy, detection: mAP, recommendation: AUC) or on hardware performance (e.g. inference latency, throughput and energy consumption). A multi-criteria evaluation of AI models is required for integrated AI systems: Both model quality and hardware performance are of great importance. However, as these are often linked in opposite directions due to model complexity, there is a trade-off between accuracy and resources that can be evaluated using a multi-criteria approach.

A combined evaluation of model quality and hardware performance for integrated AI systems with application-independent absolute scores does not yet exist.

B. Holistic HW/SW Co-Design approach for integrated AI systems

A holistic concept of a data-centered HW/SW co-design for integrated AI systems is to be designed and implemented. An overview of a potential toolbox is outlined in Figure 2. First, the search space of the hardware architectures and parameters should be limited. The hardware parameters and given requirements for the integrated AI system are used by the DSE to adapt the design space. This design space should be variable and change depending on the life cycle of the integrated AI system: In the first iteration, the design space is the largest, as the hardware is still freely definable. Later in operation, the design space is restricted to the HPO and NAS design spaces. An evaluation strategy is to be designed that can evaluate relevant metrics of the architecture search using mixed-fidelity evaluation approaches. To accomplish this, different approaches such as HiL, hardware simulation or analytical models have to be supported. State-of-the-art approaches such as Bayesian optimization, genetic algorithms and one-shot approaches have to be investigated in order to implement a suitable search strategy. In combination with the FL approach, a deployment strategy has to be developed to update trained models on already deployed devices.

As mentioned earlier in subsection II-A, FL focuses on improving the global model by extracting common general knowledge from all participating devices, besides models that deployed on devices in domains should capture and take into account local features for the correct prediction. The proposed approach to solve the contradiction is to apply DA in the FL system. That entails devices being able to fine-tune the global model based on domain data to ensure higher accuracy, which is not the case for most resource-constrained devices. The strategy is based on the FL architecture for resource-constrained devices and their corresponding domains.

A multi-criteria evaluation of integrated AI systems is to be developed by creating an absolute score from a combined evaluation of model quality (e.g. accuracy) and hardware performance (e.g. latency, energy efficiency). At the beginning, a linear combination based on [24] is designed, taking into account pareto optimal states. One focus should be on the appropriate weighting of the accuracy-resource trade-off in different applications or performance areas. MLPerf can serve as a data basis here. In addition, a methodology is to be developed that compares different domains with each other and can quantify the difference between them. A consideration of the learning curve or the training latency can represent initial approaches here. This approach should then be extended to data sets, initially with a low-fidelity and a subset of a data set. This should ultimately lead to a similarity measurement for TL, which can be used by FL to evaluate the transfer of knowledge into a new domain. A combined metric is to be provided for the DSE so that a multi-criteria evaluation of a

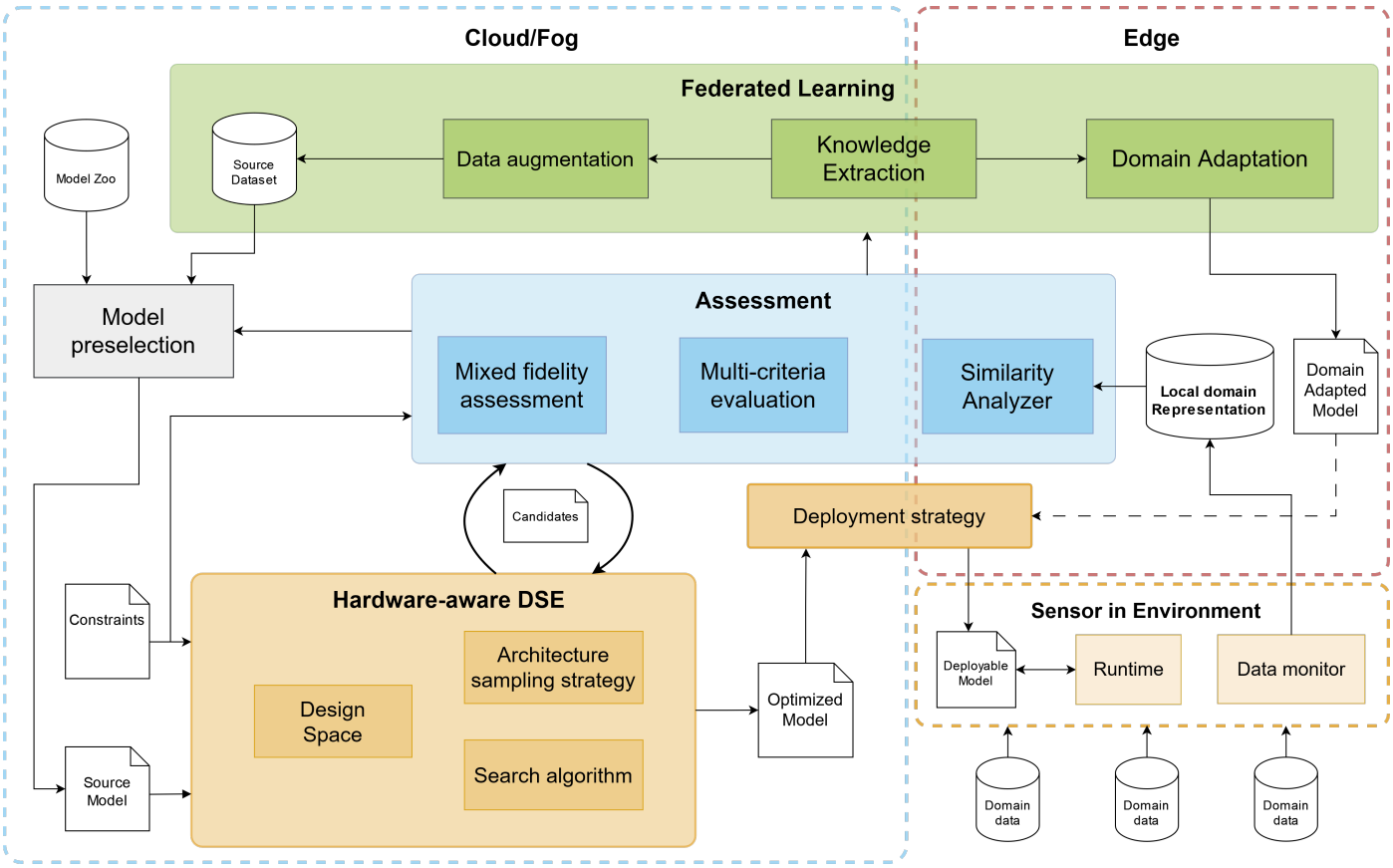

Fig. 2. Toolflow for a data-centric HW/SW Co-Design for integrated AI systems.

found architecture provides suitable comparison values to other solution candidates.

III. Use Case: Intelligent Learning Device for Automated Handwriting

Based on the first results we obtained from our research introduced in II, we present the initial DSE steps as a part of the holistic approach for a state-of-the-art use case of an intelligent pen. Together with partners from industry and academia, we are pursuing the goal of helping children to learn handwriting in the bi-nationally funded research project Kaligo-based Intelligent Handwriting Teacher (KIHT). The aim of this joint project is to develop an intelligent learning device for automated handwriting, composed of existing components, which can be made available to as many students as possible [25]. The first challenge here is to recognize the trajectory of the pen on a sheet of paper without an external reference model. Therefore, the focus is not just on word or letter recognition, but on the challenging task of using inertial sensors to retrace the trajectory of a pen without relying on external reference systems. The nearly unlimited freedom to let the pen glide over the paper has not yet provided a satisfactory solution to this challenge in the state-of-the-art methods, even with sophisticated algorithms and AI approaches [25]. The second challenge is to develop and deploy the AI algorithms in extremely resource-constrained devices (III-C).

A. *Handwriting and Pen*

Handwriting is still a very effective tool. There are many studies that show that handwritten notes are much more effective than those written on a keyboard [26], [27]. It is therefore sensible and necessary to continue using handwriting, to teach it to children and to support the learning process. Learning to write has two aspects: One is to learn and be able to reproduce letter shapes, the other is the development of muscle memory to be able to write fast and efficiently. The first aspect is taught by a multitude of apps on mobile devices, most apps only allow to trace letter shapes with the finger, whereas the more serious apps use the special electronic pens which come with more expensive tablet computers. In addition, a normal pen should be used, as writing on glass is entirely different from writing on a sheet of paper. In order to lower the barriers to computerized handwriting instruction and to also cover the second aspect of handwriting learning, KIHT propose to use an instrumented pen which writes on paper and transmits movement information to a wirelessly connected computer. Therefore, STABILO has developed an instrumented pen that captures handwriting movements in detail and precisely describes their structure in terms of their acceleration, maximum speed and braking phases. By adding an inertial measurement unit (IMU), a force sensor, a radio module and a battery to a pen, STABILO has created an electronic pen (see Figure 3) which is well equipped to

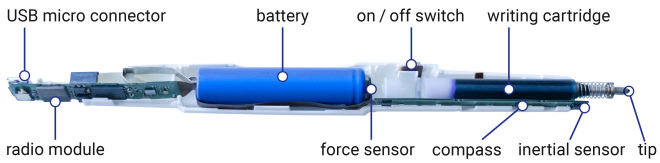

Fig. 3. Current model of the electronic pen with one half of the body removed.

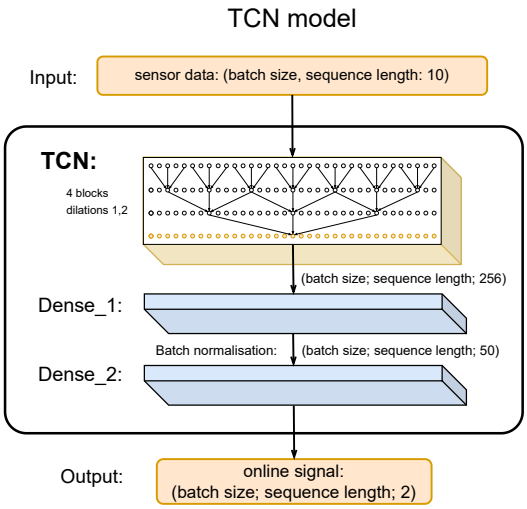

Fig. 4. The dedicated TCN model [29].

measure writing movements in much greater detail than could be observed before.

B. AI Algorithms Based on Deep Learning

Simple integration of the inertial data fails due to excessive drift, so an alternative method of tracking the pen tip had to be found. The recent development of Deep Neural Networks (DNNs) offers an opportunity, and early experiments [28] showed promising results. The main focus is on the challenging task of trajectory reconstruction from IMU sensors in the pen, by using DNNs. A novel and complete pipeline has been designed for this purpose and includes preprocessing, a neural network architecture inspired by Temporal Convolutional neural Network (TCN), and an evaluation protocol based on the Fréchet distance [29]. The data was recorded by placing a sheet of paper on the tablet and using a hybrid pen to provide ground truth data via the tablet as well as IMU signals. The signals are divided into sections corresponding to single written samples and timestamps are added as an additional channel. As a next step, not relevant sections of the input signals, for example the pen-down and end pen-up movements, are removed. Dynamic Time Warping (DTW) alignment is employed to find an alignment path between the timestamps of the pen and the tablet. Then the data is splitted into strokes during the training phase [25]. A TCN is selected because the past signals play an essential role in the reconstruction, since from a graphomotor point of view, a movement is conditioned by its past trajectory. Such convolutional layers allow to increase the receptive field without increasing the number of weights trained in the network. Our network architecture is based on four blocks of a noncausal TCN followed by two dense layers. The use of a TCN has the advantage of being faster to train and less prone to vanishing gradient than LSTM networks, especially in the case of long sequence [30]. The choice of parameters results in a receptive field of forty-nine. The network remains light as it contains less than half a million of parameters so it can be trained with a limited amount of data. During training, the model is trained to minimize the Mean Square Error (MSE) between sensor value predictions and DTW-aligned tablet trajectory. For evaluation purposes, the Fréchet distance is used as a metric. It is well defined to evaluate the model performance as it measures the distance between two curves by taking into account the location and ordering of the points along the curves.

C. Hardware/Software Co-Design

Due to the constraints applied to DSE, such as a network architecture, number of model parameters and low-power devices, several different approaches have been investigated to deploy DNNs onto the pen. One is to divide DNNs into several partitions [31]. This allows to run a part of the network on the pen and another part on the tablet. However, the goal remains to find a solution, to integrate the entire trajectory recognition into the pen. The pen should also be able to be used without other devices to avoid the distraction of the tablet for the children. For the model quantization both Post Training Quantization (PTQ) and Quantization-aware Training (QAT) are considered. Further network optimizations are achieved by using approximate computing techniques that replace the full precision operations with less precise counterparts. It allows to reduce the usage of the Floating-Point Unit (FPU) of the microcontroller or even to perform the computations on a platforms without such a unit. In [31] and [32] we show, that quantization of the Long Short-Term Memory (LSTM)-based Neural Networks (NNs) for time-series classification task and using approximate computing leads only to a slight accuracy loss.

D. First results

For training of the TCN, the KIHT dataset is used, which is composed of 106 different writers and $18,854$ samples from the following categories: characters, words, equations, geometric shapes, and sentences. The test set is composed of 344 samples from 10 different users not seen during the training phase. The results are divided into 3 categories: mono-strokes, multi-strokes and global to allow comparison of reconstruction with and without hovering. For mono-stroke samples the Fréchet distance is in average 0.11, which points to a very accurate reconstruction. The Fréchet distance is lower for multi-stroke samples, which leads to an average Fréchet distance of 0.18 for all samples. We show that it is possible to perform the inference of the TCN-based regression model using only 8-bit fixed-point quantization without significant reconstruction precision loss and that the accuracy degradation of the approximate multiplication can be partially compensated with QAT. For

example, the (3,8) fixed point quantized TCN-49 achieves the average Fréchet distance of 0.21 on the test data set, requiring only the fourth of the original memory space for the weights (460 KB of ROM) [33]. Future work will focus on improving the reconstruction of complex hovering parts, which are covered by the multi-stroke samples. The trajectory reconstruction from IMU signals is still the subject of research and is not sophisticated enough to adequately assess shape fidelity. Another focus will be the trade-off between model accuracy and required resources, which we will examine in more detail in order to be able to perform a multi-criteria evaluation as mentioned in section II-B.

IV. Conclusion and Future Work

In this paper, we have presented the challenges of integrated AI systems and how we intend to tackle them. We have given insights into our project KIHT, which aims to develop an intelligent learning device for automated handwriting. In the future, we will work on further refining and implementing the holistic approach presented for the hardware/software Co-Design for integrated AI systems.

References

[1] Z. Zhou, X. Chen, E. Li, L. Zeng *et al.*, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[2] O. Bringmann, W. Ecker, I. Feldner, A. Frischknecht *et al.*, "Automated hw/sw co-design for edge ai: State, challenges and steps ahead," in *Proceedings of the 2021 International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES/ISSS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 11–20.

[3] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[4] C. White, M. Safari, R. Sukthanker, B. Ru *et al.*, "Neural Architecture Search: Insights from 1000 Papers," Jan. 2023, arXiv:2301.08727 [cs, stat].

[5] J. Yuan, M. Xu, Y. Zhao, K. Bian *et al.*, "Resource-aware federated neural architecture search over heterogeneous mobile devices," *IEEE Transactions on Big Data*, pp. 1–11, 2022.

[6] H. Ren, D. Anicic, and T. A. Runkler, "Tinyol: Tinyml with online-learning on microcontrollers," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[7] C. Cioflan, L. Cavigelli, M. Rusci, M. De Prado, and L. Benini, "Towards on-device domain adaptation for noise-robust keyword spotting," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 82–85.

[8] D. Hussein, T. Belkhouja, G. Bhat, and J. R. Doppa, "Reliable Machine Learning for Wearable Activity Monitoring: Novel Algorithms and Theoretical Guarantees," in *2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, Oct. 2022, pp. 1–9.

[9] A. D. Pimentel, "Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration," *IEEE Design & Test*, vol. 34, no. 1, pp. 77–90, Feb. 2017.

[10] M. Tan, B. Chen, R. Pang, V. Vasudevan *et al.*, "MnasNet: Platform-Aware Neural Architecture Search for Mobile," May 2019.

[11] J. Lin, W.-M. Chen, Y. Lin, J. Cohn *et al.*, "MCUNet: Tiny Deep Learning on IoT Devices," Nov. 2020.

[12] S. S. Saha, S. S. Sandha, L. A. Garcia, and M. Srivastava, "TinyOdom: Hardware-Aware Efficient Neural Inertial Navigation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 71:1–71:32, Jul. 2022.

[13] M. Deutel, G. Kontes, C. Mutschler, and J. Teich, "Augmented Random Search for Multi-Objective Bayesian Optimization of Neural Networks," May 2023.

[14] E. Njor, J. Madsen, and X. Fafoutis, "Data Aware Neural Architecture Search," Apr. 2023.

[15] C. He, S. Li, J. So, X. Zeng *et al.*, "Fedml: A research library and benchmark for federated machine learning," 2020.

[16] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *Artificial Neural Networks — ICANN 2001*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 87–94.

[17] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. Somani, "Neural architecture search benchmarks: Insights and survey," *IEEE Access*, vol. PP, pp. 1–1, 01 2023.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause *et al.*, "Imagenet large scale visual recognition challenge," 2014.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[21] X. Huang, P. Wang, X. Cheng, D. Zhou *et al.*, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2702–2719, 2020.

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson *et al.*, "Mlperf inference benchmark," 2019.

[24] I. Fürst-Walter, A. Nappi, T. Harbaum, and J. Becker, "Design space exploration on efficient and accurate human pose estimation from sparse imu-sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[25] T. Harbaum, A. Serdyuk, F. Kreß, T. Hamann *et al.*, "KIHT: Kaligo-based intelligent handwriting teacher," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024, in press.

[26] S. Oviatt, A. Cohen, A. Miller, K. Hodge, and A. Mann, "The impact of interface affordances on human ideation, problem solving, and inferential reasoning," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 19, 2012.

[27] P. A. Mueller and D. M. Oppenheimer, "The pen is mightier than the keyboard: Advantages of longhand over laptop note taking," *Psychological Science*, 2014.

[28] M. Wehbi, D. Luge, T. Hamann, J. Barth *et al.*, "Surface-free multi-stroke trajectory reconstruction and word recognition using an imu-enhanced digital pen," *Sensors*, vol. 22, 2022.

[29] W. Swaileh, F. Imbert, Y. Soullard, R. Tavenard, and É. Anquetil, "Online handwriting trajectory reconstruction from kinematic sensors using temporal convolutional network," *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–14, 2023.

[30] M. Nan, M. Trăscău, A. M. Florea *et al.*, "Comparison between recurrent networks and temporal convolutional networks approaches for skeleton-based action recognition," *Sensors*, vol. 21, no. 6, p. 2051, 2021.

[31] F. Kreß, A. Serdyuk, T. Hotfilter, J. Hoefer *et al.*, "Hardware-aware workload distribution for ai-based online handwriting recognition in a sensor pen," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022.

[32] F. Kreß, A. Serdyuk, M. Hiegle, D. Waldmann *et al.*, "Atlas: An approximate time-series lstm accelerator for low-power iot applications," in *2023 26th Euromicro Conference on Digital System Design (DSD)*, 2023.

[33] A. Serdyuk, F. Kreß, M. Hiegle, T. Harbaum *et al.*, "Towards the on-device handwriting trajectory reconstruction of the sensor enhanced pen," in *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, 2023.