ZuSE-KI-AVF: Application-Specific AI Processor for Intelligent Sensor Signal Processing in Autonomous Driving

Gia Bao Thieu^{*}, Sven Gesper^{*}, Guillermo Payá-Vayá^{*}, Christoph Riggers[†], Oliver Renke[†], Till Fiedler[†], Jakob Marten[†], Tobias Stuckenberg[†], Holger Blume[†], Christian Weis[‡], Lukas Steiner[‡], Chirag Sudarshan[‡], Norbert Wehn[‡], Lennart M. Reimann[§], Rainer Leupers[§], Michael Beyer[¶], Daniel Köhler[¶], Alisa Jauch[¶], Jan Micha Borrmann[¶], Setareh Jaberansari[¶], Tim Berthold^{||}, Meinolf Blawat^{||}, Markus Kock^{||}, Gregor Schewior^{||}, Jens Benndorf^{||}, Frederik Kautz^{**}, Hans-Martin Bluethgen^{**}, and Christian Sauer^{**}

*Chair for Chip Design for Embedded Computing, Technische Universität Braunschweig, Germany [†]Institute of Microelectronic Systems, Leibniz Universität Hannover, Germany

[‡]Microelectronic Systems Design Research Group, Technische Universität Kaiserslautern, Germany [§]Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, Germany [¶]Robert Bosch GmbH, Germany [∥]Dream Chip Technologies GmbH, Germany **Cadence Design Systems, Germany

Abstract-Modern and future AI-based automotive applications, such as autonomous driving, require the efficient realtime processing of huge amounts of data from different sensors, like camera, radar, and LiDAR. In the ZuSE-KI-AVF project, multiple university, and industry partners collaborate to develop a novel massive parallel processor architecture, based on a customized RISC-V host processor, and an efficient high-performance vertical vector coprocessor. In addition, a software development framework is also provided to efficiently program AI-based sensor processing applications. The proposed processor system was verified and evaluated on a state-of-the-art UltraScale+ FPGA board, reaching a processing performance of up to 126.9 FPS, while executing the YOLO-LITE CNN on 224x224 input images. Further optimizations of the FPGA design and the realization of the processor system on a 22nm FDSOI CMOS technology are planned.

Index Terms—RISC-V, vertical vector processor, hardwaresoftware system, AI acceleration, sensor processing, FPGA, ASIC

I. INTRODUCTION

Nowadays, Advanced Driver-Assistance Systems (ADAS) draw on a wide range of data from different sensor systems such as radar, LiDAR, or cameras to implement crucial systems, e.g., emergency braking. In the future, the amount of highresolution, multidimensional measurement data will increase even more. Complex future applications, like autonomous driving, still require the real-time processing of these amounts of data. To extract information from these complex data in a real driving scenario, Deep Learning (DL) methods have achieved great success and are thus essential in state-of-theart and future driver assistance systems. In this context, high data volumes from a large number of sensors and the massive computational power requirements especially for the execution of AI algorithms (e.g., Convolutional Neural Networks - CNNs) pose great challenges to the underlying hardware and software systems. Other automotive constraints such as energy efficiency, robustness, flexibility, functional security, and IP security must also be met. Accordingly, a configurable AI processor can meet these requirements.

In the ZuSE-KI-AVF project, the TU Braunschweig is developing and evaluating a massive parallel vector processor architecture, named V²PRO. It must meet special performance, but also energy efficiency conditions. A hardware ecosystem is build, including a RISC-V host processor, the V²PRO as a co-processor, sensor peripherals, and a memory controller, that is developed and optimized by the TU Kaiserslautern. A Virtual-Prototype-based development environment from Cadence Design Systems speeds up the implementation and optimization of applications for V²PRO. A CNN converter is being designed as a software framework so that various neural networks can be easily mapped to V^2 PRO by its users. RWTH Aachen University develops an optimized RISC-V compiler supporting V²PRO extensions. Finally, an FPGA-based demonstrator shows the potential of the V²PRO architecture. At the end of the project, the V²PRO hardware system will be implemented as an ASIC by the Leibniz University Hannover and TU Braunschweig. Multiple camera-, LiDAR-, and radarbased AI applications are developed by the project partners Leibniz University Hannover, Robert Bosch GmbH, and Dream Chip Technologies GmbH as use cases for the evaluation of the V²PRO architecture. The duration of the project is from October 2020 to September 2023.

This paper presents the intermediate results of the ZuSE-KI-AVF project and is organized as follows: In Section 2, the hardware system is proposed. Section 3 explains the software development framework and presents the CNN converter. The use cases are described in Section 4. Early results using the FPGA demonstrator and future plans are shown in Section 5. Finally, Section 6 concludes the paper.

II. HARDWARE SYSTEM



Fig. 1. Overview of the project's hardware system

The proposed hardware system is shown in Fig. 1. The system's host processor is realized by a custom RISC-V processor architecture (EIS-V), including instruction and data caches. The high performance data processing is performed by the massive-parallel vector co-processor (V²PRO). The EIS-V processor generates vector (V²PRO-) and memory (DMA-) commands and sends them to the V²PRO co-processor. The memory transfers of the V²PRO's DMA units are optimized by a multi-port direct memory cache (DCMA) system. An AXI infrastructure enables the access to an external DDR4 memory through a memory controller or other peripherals.

A. Massive-Parallel V²PRO Co-Processor

The massive-parallel V²PRO architecture is based on the concept of vertical vector processing. In contrast to the horizontal vector concept, usually known as SIMD (i.e., single instruction multiple data), the vertical concept processes the elements of multidimensional vectors sequentially. Paired with a complex addressing of the vector elements, this enables efficient processing of not only linear vectors, but also more complex addressing schemes (see Fig. 2), as it is used, for example, in convolutional neural networks [1]. Moreover, the vertical co-processor architecture achieves high performance by performing the processing of multiple vectors on different data in parallel units.



Fig. 2. V²PRO complex addressing modes [1]



Fig. 3. V²PRO hierarchy overview

Fig. 3 shows the V²PRO's hardware architecture hierarchy. The architecture consists of multiple clusters that implement multiple vector units. Each vector unit has its own local memory. DMAs (direct memory access; 1 per cluster) transfer data between the local memories and a multi-port cache memory. A vector unit contains a load/store-lane (L/S-lane) and multiple vector lanes. The L/S-lane transfers data between the register files of each vector lane and the local memory of a vector unit. The data processing is done in the vector lanes by its ALUs (arithmetic logical unit). Each vector lane processes one vector.

The V²PRO is controlled by V²PRO and DMA commands. Specific DMA commands are executed in parallel by the DMAs of each cluster. With a 2-dimensional command (2D mode) DMAs can load and store 2D blocks from/to external memory by using a stride mechanism to determine the next line within a block. In 2D mode, DMA commands support parameterizable data padding and broadcasting of the same data to every vector unit's local memory when loading from external to local memory. V²PRO commands are broadcasted to every vector unit in every cluster. Each command has two source operands and a destination operand. The operand's vectors are described by an offset and parameters for the complex addressing of the multidimensional vectors.

B. Direct Cached Memory Access (DCMA)



Fig. 4. Overview DCMA

The single port memory controller allows only one memory access at a time, resulting in a bottleneck when scaling V^2PRO

with multiple clusters, i.e. DMAs. In addition, AI applications, like CNNs, often load the same data or data from the same memory range (e.g., the input image is loaded as overlapping segments to all vector units). A cache between DMAs and external memory can optimize these memory accesses. The structure of the direct cached memory access (DCMA) is shown in Fig. 4. The DCMA implements a multiport cache memory [2], allowing parallel DMA accesses. Thereby, the bottleneck at the external memory controller can be completely removed.

The DCMA consists of a configurable number of RAM (random access memory) modules. The DMAs access the RAMs via the DMA crossbar, which includes an arbiter, allowing only one access per RAM and cycle. DMAs can still access the same cache line because cache lines are distributed onto multiple RAMs. Cache lines are loaded and written back to the external memory via the RAM AXI crossbar and the AXI master module. The DCMA controller manages the cache mechanism, like hit/miss calculation, tag/dirty memory and cache flush.

C. EIS-V Processor

The EIS-V architecture implements the open-source RISC-V 32-bit instruction set, including integer (I), multiplier (M) and compressed instruction (C) extensions. The number of pipeline stages (e.g., IF, ID, MEM, EX, and WB) is configurable. Caches for instruction and data accelerate data accesses. A DMA allows the bypassing of the DCache for immediate write back of data to the external memory.

In the project's hardware system, the EIS-V's main purpose is the control flow and the generation of V²PRO and DMA commands for the V²PRO co-processor. The DCache was extended to fetch a complete DMA command struct (32 bytes) and to send it to the co-processor in one cycle. Through custom RISC-V instruction set extensions, V²PRO commands (16 bytes) are efficiently generated and sent to V²PRO in a single cycle.

D. Memory Controller



Fig. 5. Architecture overview of the DDR4 DRAM controller

Fig. 5 shows the architecture of the DDR4 memory controller dedicated for this project. It is designed to satisfy the requirements of the V²PRO system such as low latency, high throughput, and data security. The frequency ratio between the memory controller and the PHY is 1:4, similar to stateof-the-art memory controllers such as [3]. This allows the controller to operate at a lower clock frequency, which is required to meet timing and frequency constraints. In order to compensate for the frequency difference and avoid stalling of the PHY, the controller issues 4 DRAM commands/addresses (i.e., commands/addresses corresponding to the next 4 PHY cycles) to the XILINX PHY.

This memory controller integrates a specialized applicationspecific address mapping unit [4] that minimizes the total number of DRAM page misses, one of the largest latency penalties of DRAMs. The RD/WR access traces of the applications are analyzed offline by simulation to determine an optimized address mapping that reduces the overall latency of the transactions. This address mapping (logical-to-physical) is changeable during run-time. Thus, for each application, a different address mapping is configured to achieve low latency and high throughput. Our memory controller with default configuration shows DCMA performance improvements of up to 4.5% for different V²PRO settings compared to the implementation of the XILINX MIG memory controller. In addition, the optimized application-specific address mapping is expected to improve the performance even further.

E. Safety and Security

Due to the security requirements of the automotive industry, the memory controller shown in Fig. 5 integrates an AES-128 encryption/decryption module. This AES-128 unit is fully transparent to the system and requires no interaction from software-level. As the module is on the critical path, it is optimized for low latency (5 clock cycles). The write latency is fully hidden due to the processing time in the controller. In the future, an automatic online key generation during the initialization phase is planned with the help of a DRAM-based Physical Unclonable Function (PUF).

Furthermore, the threat of an untrustworthy supply chain is considered. Therefore, different logic locking techniques are evaluated in this project. Logic locking encrypts the hardware design using additional logic gates (key gates) to prevent an adversary in untrustworthy design houses or foundries from modifying the hardware maliciously [5]. The evaluation is conducted for multiple key lengths to quantify the influence of the protection scheme on the area and performance of the designed hardware.

Against random errors, V^2 PRO lanes can be optionally protected using dual-core lockstep (DCLS). We verify our approach using error injection, which can be executed not only during design time, but optionally also at runtime. Further safety measures are evaluated at the application level (see Sec IV-B).

III. SOFTWARE DEVELOPMENT FRAMEWORK

A. RISC-V Compiler

Open-source RISC-V compilers offer a fast toolchain for programming newly designed RISC-V applications but lack the consideration of the underlying microarchitecture. Information about the pipeline architecture, register accesses, and, e.g., timing of control operations can enable a compiler to avoid hazards when scheduling instructions. Circumventing hardware stalls



Fig. 6. Overview of the retargetable compiler toolchain

by reordering instructions can lead to significant performance gains. Furthermore, a customized RISC-V compiler allows taking application-specific hardware modifications into account. As the EIS-V handles the control of the V²PRO coprocessor, additional instructions will be added to reduce the processor's control overhead and increase the relative computation time of the vector units. The modifications include, e.g., hardware loops and optimization of the V²PRO's interfacing by introducing V²PRO's instruction generation into the compiler.

For this purpose, we utilize a commercial retargetable compiler from Synopys [6]. The compiler is fitted to the EIS-V processor by modeling it in the architecture description language nML. As illustrated in Fig. 6, the LLVM-frontend uses customized compiler header files to convert the C or C++ code to an intermediate representation (IR). The header files can contain intrinsics, customized data types, and supplemental information that allows the mapping of complex C structures on the processor's ISA. The behavior of the backend is controlled by the nML processor model of the EIS-V. Tasks, such as register allocation, instruction scheduling, and instruction mapping depend on the nML input.

Therefore, the compiler in this project yields machine code that is highly optimized for the target architecture.

B. Virtual Prototype

Virtual Prototypes (VPs) are a well-established technique for the development of complex modern microelectronic systems. Being software models of hardware components VPs can be used to simulate entire SoCs consisting of processors, networks-on-chip or interconnects, memories, and peripherals. Our VP is implemented based on the widely used SystemC/TLM2.0 C++ class library, which provides a discrete event-simulation kernel and language constructs and infrastructure to describe hardware and system components and their connections. By increasing the level of abstraction, VPs enable simplified and fast design space exploration in a methodology like in [7] or a substantially increased simulation performance allowing for simulation of complete systems comprising multiple processor cores close to real time. The simulation performance of very complex systems can be further improved by distributed parallel SystemC simulation [8]. This performance increase enables early development and verification of system software including operating systems according to the shift-left methodology.

The Virtual Prototype developed in this project integrates the instruction set simulators of a RISC-V core (Imperas) and the V^2PRO co-processor, which are both connected to the DRAM memory controller by a special interconnect which translates

between the different levels of abstraction representing the architecture shown in Figure 1.

Further, the VP in this project is extended to apply nonintrusive software verification techniques for detecting issues like buffer or stack overflows or manipulation like returnoriented programming attacks (ROP-attack). While currently available solutions (e.g., Valgrind or the Google Sanitizers) influence the software footprint regarding memory consumption and performance (2x-40x slowdown), the solution developed in this project aims to find the root causes from the outside without modifying the executable. This is especially important for embedded systems in which changes in timing or memory footprint have a major impact.

C. CNN Converter

With the CNN Converter framework (Fig. 7), software developers can quickly map different neural network architectures on the V²PRO. The CNN Converter takes the neural network description and the trained weights as the input. By analyzing the individual layers with hardware information (V²PRO configuration like number of cluster, units, data precision) the network weights are automatically converted from floating point representation to optimized fixed point values.



Fig. 7. Overview of the CNN Converter framework

In a preprocessing step, the vector (V²PRO) and memory (DMA) commands for the V²PRO co-processor are generated separately and optimized depending on the hardware configuration. Then, the C++ application, which will run on the EIS-V processor, is built with a CNN library, including references to the previously generated vector/DMA V²PRO commands. From that application, the RISC-V compiler generates optimized machine code (including memory initialization) for execution on the EIS-V processor.

IV. USE CASES

A. Camera-based Applications



Fig. 8. Overview of the camera based side-mirror application

Modern vehicles are equipped with several cameras, which provide a vast amount of real-time information about the environment as well as of the interior of the vehicle. The information in the captured images can be extracted by stateof-the-art deep neural networks with high accuracy. Since these algorithms require a huge amount of processing power and energy, sophisticated hardware is needed to run them in realtime. Within the project, we focus on developing a camerabased system, which captures images from the side-mirror perspective with integrated 3D object detection and tracking run on the V²PRO. Such a system improves safety in different situations and enables for ADAS applications, like a turning assistant, which alerts the driver when pedestrians or cyclists are about to cross the vehicles path, a door opener warning in case a cyclist is approaching from the back, or a lane change assistant.

The system will be implemented on an FPGA platform as a demonstrator in a first step and is summarized in Fig. 8. The raw images, which are provided by an image sensor and lens, are processed by an image signal processor (ISP) developed by Dream Chip Technologies GmbH forming a dedicated camera module usable for ADAS applications. As a real-time ISP, it supports up to 4k resolution at 60 FPS. Among typical modules like demosaicing, bad pixel detection and correction, denoising, black level compensation, lens shade correction, and color correction, the ISP also supports high dynamic range (HDR) image fusion, global and local tone mapping, as well as multiple output paths for human and machine vision.

From each RGB image 3D boxes, described by their positions, dimensions, and headings, are extracted for four classes of objects (pedestrians, bicycles, vehicles, motorcycles). The object detector is based on the CenterNet [9] architecture combined with an EfficientNetV2 [10] as the backbone, providing a good balance between efficiency and accuracy. The detected objects are tracked over time by applying a matching and several temporal filters. This enables extracting more detailed information on the objects, like trajectory extraction, object velocity, and path prediction. Finally, the captured images are visualized on a screen overlayed by the tracked objects.

B. Radar-based Applications



Fig. 9. Generic radar signal processing chain for object detection and classification

High-resolution automotive imaging radars are a key sensor technology to enable autonomous driving and advanced driver assistance systems. This is due to their high robustness against harsh weather and lighting conditions as well as the comparatively low cost. Challenging demands are placed on the resolution and target separability in all dimensions. To meet the requirements on the direction-of-arrival estimation, superresolution estimation algorithms are utilized, which demand high computational power from the underlying hardware. One class of algorithms especially suitable for this application is constituted by Sparse Bayesian Learning based methods. Those exploit the signal sparsity in the angular domain and provide super-resolution capability based on a single measurement. By decoupling the calculations of separate angular directions, a high level of parallelization is enabled. The massive parallel architecture of the V²PRO allows to take full advantage of this property. This permits a significant speed-up of calculation time and facilitates the real-time multi-target direction-of-arrival estimation.

After signal processing, the resulting information-rich point clouds can be exploited for reliable radar-based environmental perception. In the scope of the project, neural network architectures for object detection on radar point clouds [11], i.e., the simultaneous localization and classification of objects, and their implementation on the V²PRO are investigated. A crucial step, both regarding the detection performance and the efficiency of the implementation, in these architectures is the mapping from point clouds into regular image-like data structures, that are well-suited for the computation on the V²PRO. The training and evaluation of these networks is performed on data captured by a prototypic high-resolution multiple-input and multiple-output (MIMO) automotive radar with 64 virtual channels operating at 77 GHz.

In addition to the safety mechanisms mentioned in Sec. II-E, measures on application-level are evaluated to enable low cost protection. Soft errors (e.g., bit-flips) in the radar data affect subsequent processing steps and therefore pose a serious threat to the functional safety of the system. Traditional protection mechanisms (e.g., triple modular redundancy, TMR) are not ideal for high dimensional radar data due to the high overhead they entail. Therefore, we systematically define small observation windows in the range-Doppler spectrum to detect peaks caused by soft errors. Our method enables low cost protection for 2D FFTs and can reliably detect and mitigate errors even at high error rates [12].

C. LiDAR-based Applications



Fig. 10. A semantically segmented point cloud. The different colors present different classes.

LiDAR scanners acquire range information and are capable of creating 3D point clouds from their surrounding. In this application, the solid-state LiDAR scanner (ibeoNEXT) by Ibeo Automotive Systems GmbH is used for data acquisition. In addition to range data, this scanner provides point-wise information like pulse width, blooming, or existence probability. Several scanners with varying fields of view (11.2°, 60°) are mounted on the chassis of a demonstration vehicle to capture its surroundings. The acquired data provides the base for semantic segmentation, in which the individual points are classified for the use in applications of, e.g., autonomous driving (Fig. 10). With regard to the expected high area and energy efficiency of convolution operations on the V²PRO, the CNN SalsaNext [13] was chosen for this task. Presented in three modules, SalsaNext first reduces the input's dimension from 3D to 2D, then performs common CNN operations for segmentation and classification and in a final step projects the results onto the input point cloud.

The additional information provided by the ibeoNEXT scanners are investigated with regards to their effect on the detection performance by directly using their point clouds as CNN input. However, a dataset for this sensor is not available. Therefore a specific dataset will be captured with our demonstration vehicle. The demonstration vehicle also serves for experiments of online processing with the V²PRO. Reducing the input data's dimensions involves the computation of square root, division, arcussinus and arcustanges. For these functions, efficient approximations are to be implemented on the V²PRO. The ML-model mainly utilizes common CNN operations already implemented in the CNN Converter, see Sec. III-C. Its extension is required for more unconventional operations, e.g. layout transformation or concatenate operations. Mislabeling caused by the CNN is tackled by locating the k-nearest neighbors of an input point and finding the most frequent label. Additionally, the CNN model needs to be optimized by applying compression methods and eliminating operators. The impact of these modifications on the detection performance are to be determined and evaluated. Ultimately, our goal is to perform live semantic segmentation on the V²PRO system using our demonstration vehicle.

V. INTERMEDIATE RESULTS AND FUTURE PLANS

During the first part of the project, Version 1 of the hardware system was implemented and synthesized on an ALDEC TySOM-3A-ZU19EG Embedded Prototyping Board, including a state-of-the-art Xilinx UltraScale+ FPGA. The YOLO-LITE [14] was used as an exemplary CNN application on 224x224 input images and is running on the FPGA board. Table I shows the results compared to an NVIDIA 3600 mobile GPU and an NVIDIA Xavier GPU. V²PRO was configured with 2 clusters and 2 vector units per cluster (2C2U) and 8 cluster and 8 units per cluster (8C8U). The second 8C8U configuration was synthesized with the current maximum frequency of 400 MHz. V²PRO achieves less total FPS, but reaches a much higher efficiency by a better usage of the available resources (performance utilization = $\frac{real \ perf}{peak \ perf}$).

The next steps of this project contain demonstrators running AI-based, complex applications (see Sec. IV), further FPGA design optimizations, and the ASIC realization of the processor system on a 22nm FDSOI CMOS technology. V²PRO version 2 will have SIMD features and further application-specific DCMA optimizations. At the end of the project, the behavioral V²PRO architecture design will be published as open source.

VI. CONCLUSION

This paper presented the hardware-software system concept, use cases, and intermediate results of the ZuSE-KI-AVF project.

TABLE I RESULTS OF YOLO-LITE RUNNING ON DIFFERENT GPUS AND V^2PRO

	GPU		V ² PRO		
	3060M	Xavier	2C2U	8C8U	8C8U
#CUDA Cores or #Vec. Lanes	3840	512	8	128	128
Clock [GHz]	1.282	1.377	0.25	0.25	0.4
FPS	510	218	7.1	88.6	126.9
Efficiency [FPS/Core/GHz]	0.1	0.3	3.6	2.8	2.5
Theor. Peak Perf. [GOPS]	9845.0	1400.0	4.0	64.0	102.4
Real Perf. [GOPS]	245.8	105.1	3.4	42.7	61.2
Perf. Utilization [%]	2.5	7.5	85.9	66.7	59.7
Memory Bandwidth [GB/s]	336	127	6.4	6.4	6.4

A first FPGA-based demonstrator with a CNN example application shows significantly better efficiency compared to mobile GPUs. In the next steps of the project, new FPGA- and ASICbased demonstrators for different use cases will further evaluate the proposed system.

ACKNOWLEDGMENT

The work is supported in part by the German Federal Ministry of Education and Research (BMBF) within the project ZuSE-KI-AVF under contract no. 16ME0379.

REFERENCES

- S. Nolting, F. Giesemann, J. Hartig, A. Schmider, and G. Paya-Vaya, "Application-specific soft-core vector processor for advanced driver assistance systems," in 2017 27th International Conference on Field Programmable Logic and Applications (FPL), 2017
- [2] S.-S. Ang, G. Constantinides, P. Cheung, and W. Luk, "A Flexible Multiport Caching Scheme for Reconfigurable Platforms," in Reconfigurable Computing: Architectures and Applications, Berlin, Heidelberg, 2006
- [3] Synopsys, Inc., "Synopsys DDR IP Solutions," https://www.synopsys.com/designware-ip/interface-ip/ddr.html, 2022, Last Access: 17.11.2022.
- [4] M. Jung et al., "ConGen: An Application Specific DRAM Memory Controller Generator," in Proceedings of the Second International Symposium on Memory Systems, 2016
- [5] D. Sisejkovic, L. M. Reimann, E. Moussavi, F. Merchant and R. Leupers, "Logic Locking at the Frontiers of Machine Learning: A Survey on Developments and Opportunities," 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC), 2021
- [6] M. Hohenauer et al., "A methodology and tool suite for C compiler generation from ADL processor models," Proceedings Design, Automation and Test in Europe Conference and Exhibition, 2004
- [7] Kautz, F., Blume, H. & Sauer, C. Methodology for an Early Exploration of Embedded Systems using Portable Test and Stimulus Standard. 2022 35th SBC/SBMicro/IEEE/ACM Symposium On Integrated Circuits And Systems Design (SBCCI), 2022
- [8] Sauer, C., Bluethgen, H. & Loeb, H. Distributed, loosely-synchronized systemC/TLM simulations of many-processor platforms. *Proceedings Of The 2014 Forum On Specification And Design Languages (FDL)*, 2014
- [9] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points", arXiv preprint arXiv:1904.07850, 2019.
- [10] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training", In: International Conference on Machine Learning. PMLR, 2021
- [11] M. Ulrich et al., "Improved Orientation Estimation and Detection with Hybrid Object Detection Networks for Automotive Radar," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 2022
- [12] M. Beyer, A. Guntoro and H. Blume, "Fault-tolerant Radar Signal Processing using Selective Observation Windows and Peak Detection," 30th European Signal Processing Conference (EUSIPCO), 2022
- [13] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertaintyaware semantic segmentation of LiDAR point clouds," in Advances in Visual Computing, pp. 207–222, Springer International Publishing, 2020
- [14] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," in 2018 IEEE International Conference on Big Data (Big Data), 2018