applied sciences

MDPI

Article

HPC Platform for Railway Safety-Critical Functionalities Based on Artificial Intelligence

Mikel Labayen [1,2,*,†], Laura Medina [3,†], Fernando Eizaguirre [4,†], José Flich [3] and Naiara Aginako [2]

[1] Autonomous Vehicle Department, CAF Signalling, 20018 Donostia, Spain
[2] Computer Sciences and Artificial Intelligence Department, University of the Basque Country, 20018 Donostia, Spain; naiara.aginako@ehu.eus
[3] Computer Engineering Department, Universitat Politècnica de València, 46022 Valencia, Spain; laumecha@inf.upv.es (L.M.); jflich@disca.upv.es (J.F.)
[4] Embedded Systems Department, Ikerlan Technology Research Centre, 20500 Arrasate/Mondragón, Spain; feizaguirre@ikerlan.es
* Correspondence: mlabayen@cafsignalling.com
† These authors contributed equally to this work.

Abstract: The automation of railroad operations is a rapidly growing industry. In 2023, a new European standard for the automated Grade of Automation (GoA) 2 over European Train Control System (ETCS) driving is anticipated. Meanwhile, railway stakeholders are already planning their research initiatives for driverless and unattended autonomous driving systems. As a result, the industry is particularly active in research regarding perception technologies based on Computer Vision (CV) and Artificial Intelligence (AI), with outstanding results at the application level. However, executing high-performance and safety-critical applications on embedded systems and in real-time is a challenge. There are not many commercially available solutions, since High-Performance Computing (HPC) platforms are typically seen as being beyond the business of safety-critical systems. This work proposes a novel safety-critical and high-performance computing platform for CV- and AI-enhanced technology execution used for automatic accurate stopping and safe passenger transfer railway functionalities. The resulting computing platform is compatible with the majority of widely-used AI inference methodologies, AI model architectures, and AI model formats thanks to its design, which enables process separation, redundant execution, and HW acceleration in a transparent manner. The proposed technology increases the portability of railway applications into embedded systems, isolates crucial operations, and effectively and securely maintains system resources.

Keywords: autonomous and driverless train operation; computer vision and artificial intelligence; high-performance computing; safety-critical; AI hardware accelerator

check for updates

Citation: Labayen, M.; Medina, L.; Eizaguirre, F.; Flich, J.; Aginako, N. HPC Platform for Railway Safety-Critical Functionalities Based on Artificial Intelligence. *Appl. Sci.* **2023**, *13*, 9017. https://doi.org/10.3390/app13159017

Academic Editor: Keun Ho Ryu

Received: 21 June 2023
Revised: 1 August 2023
Accepted: 2 August 2023
Published: 7 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Users of the European rail industry are clamouring for a future Automatic Train Operation (ATO) system since it provides advantages such as lower operating costs, longer product life-cycles for railways and increased safety. Its definition is being worked on by the European Shift2Rail standards group [1]. For fully autonomous train operation, various rolling stock suppliers and stakeholders have already begun researching, developing and testing technologies.

Similarly to other transport sectors, different computational issues are being faced by numerous railway suppliers and stakeholders for CV- and AI-enhanced autonomous train operation. The adoption of computer equipment capable of offering the performance of high-end graphic-processor units while being able to simultaneously meet safety criteria will be necessary for the future of CV and AI advances in the railway sector. These developments will increase the size, speed, and dependability of CV and AI processing calculations.

Through the use of multi-cores, Graphic Processors Units (GPUs), and specialized accelerators, a number of HPC commercial off-the-shelf platforms provide the calculation capabilities required by autonomous systems in fields such as intelligent transportation systems, space, and robotics [2].

However, because of the challenges or barriers that HPC platforms pose to the certification process, such as support for functional and timing isolation and testability, the use of these platforms has historically been viewed as being beyond the reach of the industry of safety-critical systems (i.e., controllability and observability). Therefore, the state-of-the-art (SoA) safety-critical computing platforms cannot currently satisfy these demanding specifications.

The SELENE (Self-monitored Dependable Platform for High-Performance Safety-Critical Systems) project [3] is a European R&D initiative that develops the research provided in this article as a use case demonstration. This work proposes a high-performance platform with safety-related considerations as a main design goal in an effort to bridge this gap. The SELENE platform is an open-source Reduced Instruction Set Computer V (RISC-V) [4] multi-core processor with hardware acceleration for artificial intelligence that supports multiple types of redundancy, real-time performance monitoring, and enforcement mechanisms to ensure that the safety objectives of the applications are satisfied. Additionally, the design of this system-on-open-source chip makes it possible for it to easily adapt to other safety domains. Apart from the use case presented in this article, three other use cases from the automotive and space industries have been used to test this method.

The article is organized as follows: Section 2 gives an overview of some relevant related works and highlights the main differences with our approach. Section 3 describes the railway domain use case in which the approach is being tested. Section 4 presents the use case deployment details analysing the new HW and SW modules included in the platform and the platform architecture, taking into account the safety-related analysis of our use case. Sections 5 and 6 define the test which was carried out and presents the obtained results in order to demonstrate the performance of the solution. Finally, Section 7 presents the conclusions and future work.

2. Related Work

The use of artificial sense (in real-time and via onboard embedded hardware) has been presented via a number of demonstrations in the railway industry.The Siemens autonomous tramway pilot case [5] was one of the first demonstrations that continues to inspire researchers today [6]. With this in mind, vision-based on-board obstacle detection and distance estimation in railroads have become the most pertinent scientific approaches [7]. To test and validate an autonomous obstacle detection system, various experiments and actual pilot cases have been implemented throughout the past few years, [8–13]. Other applications have also been worked on, such as vehicle localization on light trains [14] or railway lateral signalling detection on mainline trains [15,16].

The majority of these demos, similarly to those for self-driving cars, concentrate more on the computing capability of those systems rather than how these platforms may be certified. However, certification of sophisticated computing systems is an active area of study, such as those necessary for fully automated train systems [3]. Major chip suppliers, such as Nvidia and Intel, are also creating particular platforms that support that purpose [17], by including built-in fault-tolerant mechanisms such as lockstep execution or error correction codes in memory structures. Finally, to certify complex systems affordably, a more comprehensive and cutting-edge safety certification technique is required [18].

There are currently no commercial solutions that guarantee high-performance equipment with the safety requirements to be met in the railway sector. This work aims to be the first on-board HW prototype to execute AI functions safely (and in real time) in railway operations.

3. Use Case Definition

The use case presented in this paper, which is intended to validate the HPC SELENE platform, has been titled as automatic accurate stopping and safe passenger transfer, and it consists of automatic functionality collection based on CV- and AI-enhanced techniques. Figure 1 shows the graphical representation of it. The use case specifically highlights the following three features:

- Data collection and synchronization: this captures data from stereo vision-capable cameras in real time and it synchronizes and rectifies data of both video stream signals.
- Automatic station detection and accurate stop aligning the vehicle and platform: this detects the station platform and it accomplishes precise localization inside the platform area by detecting, recognizing and tracking visual patterns. The visual landmarks have been chosen to maximize the results of the detection and identification process in any possible lighting conditions. Visual stereo sensors that have been properly calibrated assess the physical distance.
- Safe passenger transfer: this captures data from rear cameras and it manages automatic safe door functionality preventing (a) door opening operation if the train and platform are not precisely aligned and (b) door shutting operations if any passengers are entering or exiting.

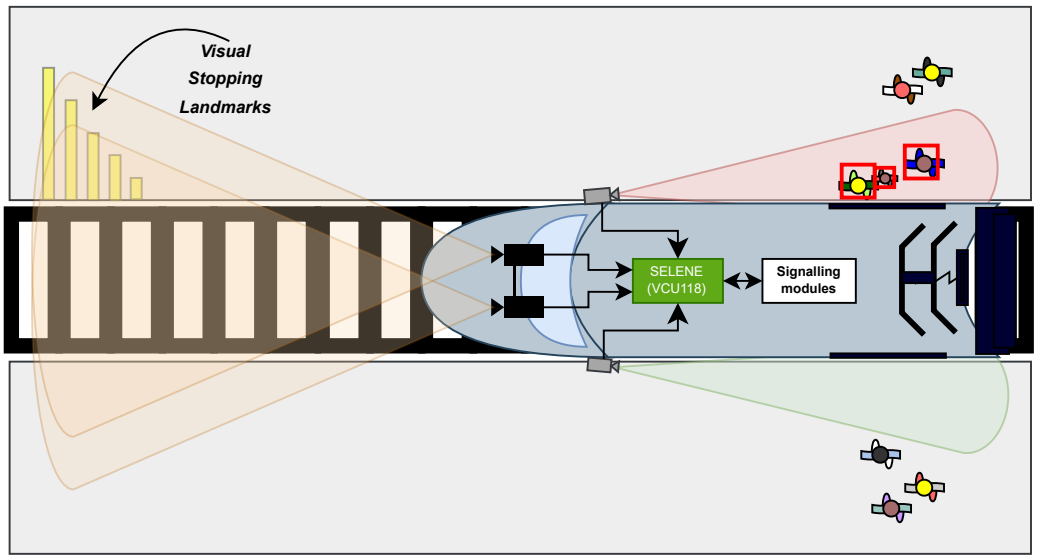

Figure 1. Physical set-up of the solution. Equipment distribution on the train.

Apart from the functional requirements, there are two key requirements to be met in this use case. The first is to keep the processing time as low as possible, since the accurate stop with a moving train (and its inertia) does not allow latencies that could turn the results of the visual analysis into obsolete data, as these would not be useful for an accurate control of the vehicle. On the other hand, the passenger detection functionality has to be secured by combinations of redundant executions over isolated resources.

3.1. System Set-Up

As shown in Figure 1, the set-up consists of two cameras (located in the train cabin) in properly calibrated stereo vision configuration, another two rear cameras (pointing at the passenger doors) and the Xilinx VCU118 board which incorporates the SELENE platform. Each camera sends a real time video stream into the system and all of these streams are analysed using the CV- and AI-enhanced algorithms to extract valuable data and send information to the next signalling equipment (decision making and actuators modules).

3.2. System Workflow

The solution architecture contains three main logical modules. The first one captures data coming from cameras and it synchronizes them in time. If the data comes from stereo cameras, it also rectifies them. The second one performs a real time data analysis using CV and AI techniques. The third collects the results of the analysis, and for those safety functionalities the 2oo2 (two-out-of-two) RootVoter (RV) logic is applied.

The most demanding computer resource functionalities are concentrated in the second logical module which is fully executed in the VCU118 board and SELENE platform:

- Platform landmark detection and identification: this detects the start/end landmarks of the platform by a pre-trained AI model (YOLOv4 [19] architecture) inference process, determining if the train is on the platform and establishing a reference point in the approximation phase for the ultimate accurate stop.
- Distance estimation: this support the precise stop process in the platform area. The distance to station stopping landmarks is calculated updating the predicted remaining distance of ATO. This calculus is based on a dense disparity map calculated by the Semi-Global Block Matching (SGBM) method [20].
- Passenger detection: using the same techniques but a different AI model, it detects passengers when they are boarding or exiting the train, managing the door opening and closing commands.

4. Deployment on SELENE Platform

This section describes the use case deployment details and it focuses specifically on four main contributions of this work: the HW accelerator, acceleration runtime, hypervisor and rootvoter.

The SELENE platform builds upon a combination of a multi-core and accelerators, which are prototyped on a FPGA System On a Chip (SoC), based on the non-proprietary RISC-V instruction set architecture (ISA). Due to their open nature, the use of an open ISA with a Linux OS and Jailhouse hypervisor [21] offers flexibility and an extension at the SW level. All these features are made compliant with the highest safety integrity levels across domains by building adequate safety measures such as monitoring, fault containment, diverse redundancy (RV availability), ease for testability, etc., in the HW and SW layers. The architecture of our railway use case and how it is implemented on top of this platform is described below.

In the use case presented in this work, safety considerations are different for each functionality. Automatic accurate stop is not a safety-related function since if the train stops beyond the platform the doors are not opened. On the contrary, a safe passenger transfer has safety implications since closing doors when passengers are still getting in/out of the train might endanger their physical health.

Due to the need for high-performance (based on parallel executions) and function separation, each task execution should rely on distinct RISC-V cores and isolated cells (except NoSafety functions which can share the same cell but should be isolated from the rest of the safety-related executions). This may be accomplished by utilizing the SELENE platform HW/SW isolation features based on the Jailhouse hypervisor.

Figure 2 shows the SELENE platform architecture in the HW and SW domains incorporating the functionalities that need to be executed in the presented use case. In the HW domain, the separation into different RISC-V CPU cores of the three main tasks can be seen. The passenger detection function, being a safety function, is redundant and its two outputs are managed by the RV of the SELENE platform. Three of the processes (two for passenger detection and one for platform landmark detection) also use hardware acceleration for their inferences. Finally, the accurate stop function requires two separate RISC-V CPU, one for AI-model inference used for landmark detection and another for distance estimation based on stereo matching algorithms. Both of them are very resource-consuming.

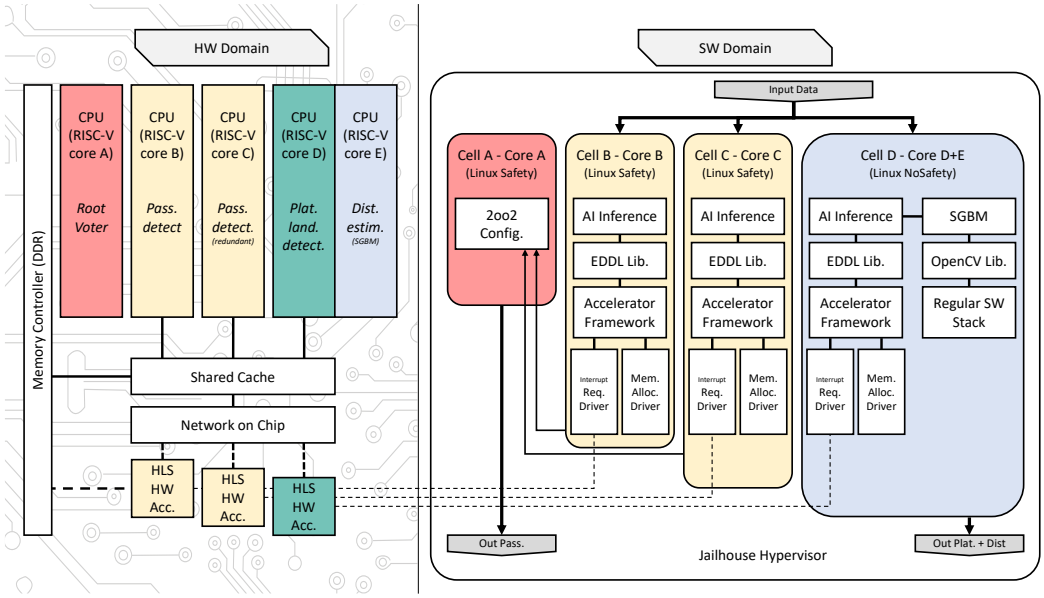

Figure 2. HW and SW architecture of the proposed solution.

In the SW domain we can see the control of the different executions of the functionalities through the creation of cells controlled by the Jailhouse hypervisor. Moreover, we can also appreciate the different SW stack executed in each core depending on the task under execution as well as the interrupts required to make use of HW acceleration.

The safe passenger transfer function, based on AI-enhanced passenger detection, is developed in one core with a replica (in the second core) to build a 2oo2 redundant system, such as those required to achieve high criticality in the railway sector [22]. A comparison of the function results is carried out by the RV HW modules incorporated in the SoC. SELENE hardware monitors make sure that safety properties are preserved.

In order to deploy this entire HW and SW architecture on the XIllinx VCU118 board, some research and development was required beyond the SoA. The exact contributions of this work with respect to the SoA are as follows:

- HLSinf HW Accelerator extension: the creation of new layers to support the YOLOv4 architecture: a Support Tensor Machine (STM) layer which is a grouping of the three different layers (softmax, hyperbolic tangent and element-to-element multiplication), and an ADD layer (element-to-element addition).
- New Acceleration Runtime: enabling HLSinf HW accelerator and Linux OS communications, accelerators control, memory allocation, and interruption manager.
- AI-inference SW library extension: a new compute service, called SELENE, to port and extend the inference library, making the platform compatible with most known AI architectures.
- Hypervisor new extension: a porting solution to RISC-V CPU and enabling process isolation.
- RootVoter new extension: a porting solution to RISC-V CPU and enabling safety-related executions on the SELENE platform based on redundant execution.

The AI hardware accelerator, AI-Inference library, and an acceleration runtime method created in this work comprise the SELENE Accelerator Framework (SAF). It works as follows: first, the European Distributed Deep Learning (EDDL) [23] inference library initializes the HLSInf [24] HW accelerator using the generated JSON configuration file. Next, the inference input data (i.e., the data to be processed) is loaded in the main memory shared with the accelerator. For this purpose, a dedicated input buffer is allocated in the memory using the Memory Allocation Driver. As soon as the input is loaded, the inference library runs the accelerator and blocks the process until the accelerator has finished (or

until the timeout has been reached). The final step for the EDDL is to read the output buffer to retrieve the inference output data (i.e., the data processed by the accelerator).

4.1. SELENE AI HW Accelerator

The HLSinf accelerator is a high-level synthesis open-source FPGA accelerator which creates an efficient hardware IP for ASIC or FPGA targets and is used for inference processes of AI models based on convolutions. The central characteristic of this accelerator is flexibility, as it allows a specific AI hardware accelerator to be designed and implemented to the particular use case.

It is designed using the channel slicing concept, where a set of input channels are processed in parallel, and a set of output channels are produced in parallel. This allows the programmer to select the degree of parallelism at the design time, where a bigger parallelism implies a bigger accelerator size and more FPGA resources. This speed-up flexibility allows the user to define the best well-suited parallelism considering the available FPGA resources and the degree of parallelism desired.

This accelerator has been integrated into the SELENE SoC and interconnected with memory and the RISC-V cores using an AXI interconnect. This HLSinf accelerator on the SELENE platform can be customized to support specific data formats and Neuronal Network (NN) layers and currently supports several well-known AI models such as YOLOv3/4, Tiny-YOLO, or VGG16. The accelerator and the CPU cores share the same memory, which minimizes the cost of data movement and allows fine-grain HW/SW co-designs of the AI algorithms between the RISC-V cores and the AI accelerators to be performed. In addition, HLSinf has been designed to run in the EDDL library, providing the support needed to run offloaded AI model layers on the FPGA. It can configure and compile a given subset of network layers for use in an inference process running with EDDL. HLSinf and EDDL allow a perfectly coupled HW/SW co-design approach where some parts of the model run in the FPGA, whereas the rest run in the CPU or GPU when available.

Figure 3 shows the design of the accelerator in the SELENE Platform. This has been defined around the dataflow model using modules interconnected by data streams. This dataflow model accelerates the overall throughput of the design as it enables task-level pipelining, permitting several operations to start before the previous functions have completed all of their operations.

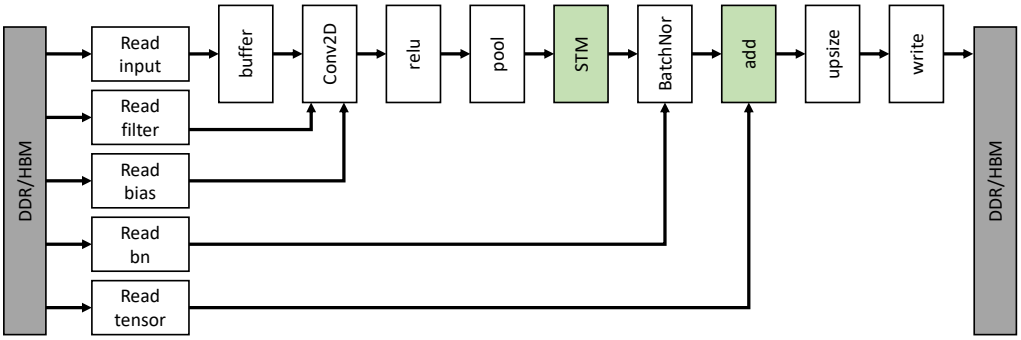

Figure 3. Design of the HLSinf accelerator (new contributions in green color).

4.2. Acceleration Runtime

This work also presents a new low-level runtime that allows the HW accelerators included in the SELENE platform to communicate with the Linux operating system. The SELENE Acceleration Runtime (SAR) is the lowest software level and it controls the accelerators, ensures memory allocation, and manages interruptions. The runtime also interacts with the EDDL inference library using an Open-CL-like Application Program Interface (API). The EDDL and the low-level runtime are included in the SELENE Linux Image deployed on the NOEL-V-based [25] platform. Thus, the final application running on

the NOEL-V infers the AI algorithms and makes the deployment of the HW accelerators transparent for the use case. The SAR can handle multiple kernels and is designed to easily configure the control of all the kernels with a parametric register from a JSON file.

The accelerators require a contiguous physical memory block for data input and output. As we are using the Linux OS and we cannot directly write to the main memory (it would end in an OS crash), we use a kernel driver, called the Memory Allocation Driver, to ensure the contiguous memory allocations. An API has been designed for interfacing the SAR with the upper software level. This API contains a light OpenCL C++ compatibility layer for easier operations.

4.3. AI-Inference SW Library

The EDDL library is a general-purpose, open-source, deep-learning library used for the training and inference processes of NN models. One of the key features of EDDL is its ability to work with a wide variety of hardware, including CPUs, GPUs or FPGA. This allows users to take advantage of the best hardware for their specific use case and makes it easy to switch between different hardware platforms. Interoperability is provided with the EDDL Open Neural Network Exchange (ONNX) format support, as it allows pre-trained models in ONNX format to be loaded and it ensures compatibility with other frameworks.

The SELENE platform relies on Linux as the default operating system. In particular, it uses a Debian-based RISC-V Linux adaptation to NOEL-V. The AI software toolchain integration is built on top of this Linux distribution. To deploy artificial intelligence models on the SELENE platform, the EDDL library has been extended by including the SELENE platform as a new computing target. The EDDL library is then deployed on top of the Linux OS running on the NOEL-V processor. This allows for the inference process to be executed entirely within the NOEL-V multi-core system. Additionally, SAF is used to offload heavy computations to the SELENE AI hardware accelerators to speed up the inference process.

4.4. Jailhouse Hypervisor

Jailhouse is a partitioning hypervisor based on Linux. It configures CPU and device virtualisation features of the hardware platform in such a way that none of the resulting domains, called cells, can interfere with each other in an unexpected way. Jailhouse currently officially supports the x86-64, ARMv7 32-bit and ARMv8 64-bit architectures. For the SELENE SoC, which uses the NOEL-V processor, the code has been ported to the open RISC-V ISA. The main implementation challenge has been the decoding of transformed/pseudo instructions stored in dedicated system registers on processor exceptions (e.g., memory access violations).

As shown in Figure 2, different cells have been created for different processes (even for redundant ones) to isolate some functionality from the rest, avoiding the sharing of resources and interruptions between them.

4.5. RootVoter

The current version of SELENE SoC includes four RV cells, each has a maximum of 16 datasets to vote. The voting scheme MooN and the timeout interval are configured during cell initialization. The RV driver for Linux enables (a) resetting and initializing each RV cell, (b) loading the datasets to the dedicated RV registers, (c) polling the voting results from each RV cell, (d) parsing voting results and diagnosing the errors (if any).

The voting logic assumes that the datasets are loaded during T clock cycles (configured by software) after the configuration command. The voting starts when all N datasets are loaded to the set registers, or when at least M datasets are loaded by the end of time interval T. If at the end of this time interval less than M datasets are available, then the RV cell reports a timeout.

Once the voting is completed the, RV reports an agreement status that indicates whether at least M datasets (out of N) match among themselves and validity flags that

indicate whether each particular dataset matches with the rest. The use case of this work uses only one cell for RV and the chosen voting scheme is 2oo2.

5. Test Description

Several tests have been defined to validate the platform. The tests are centered on critical requirements of the performance, process isolation and redundant execution, and also on the integration of third party libraries such as OpenCV [26]. The tests have also been found suitable to evaluate SELENE outputs in comparison to available market solutions that have higher TRL in inference execution but lack safety in-built mechanisms such as Alveo from Xillinx [27] and Jetson AGX Xavier from Nvidia [28].

In order to run the tests, part of a private *CAF* dataset [29] containing stereo images of an urban railway environment has been used. The dataset contains 19 sequences on the railway track. A sequence defines a record that starts at one station and spans the next station or two until the train stops. The frames are rectified RGB colour images coming from a stereo camera stored with lossless compression using 8-bit PNG files. The size of the images is 1280 × 720 (HD). Only a sub-part of this database has been used, those frames where landmarks and passengers are present. In total, 200 stereo image pairs from ten different sequences were used.

- System workflow validation test: the entire workflow of the system has been validated using dataset images. Apart from the correct functioning of the main functionalities, special attention has been paid to the following two points:
  - Back support libraries for the distance calculus: distance calculus requires an available implementation for the SGBM algorithm [26]. This implementation is ready in the OpenCV library but must be validated to ensure that the libraries that compute stereo SGBM matching can be cross compiled and executed in the SELENE platform with RISC-V architecture.
  - Model parsing compatibility: the models used for the use case are trained using the Darknet framework [30]. The Darknet output is not compatible with other frameworks and, for that reason, ONNX has been chosen as the sharing format. As ONNX establishes a standard format, but there are no standard parsers or exporters, the compatibility of exported models with the EDDL ONNX parser must be validated. Inference tests were used to validate this compatibility.
- AI models (passenger and landmark detectors) inference performance test: this test focuses on the performance of the machine learning algorithm in the platform. In the test, the Tiny-YOLOv4 inferences for landmark and passenger detection are executed with different computing precision. The models are 608 × 608 RGB image input models that were trained using transfer learning with a database labelled with railway traffic signals, platform landmarks and people/passengers. In addition, the goal has also been to compare the performance of the accelerator against SoA existing hardware such as Xilinx Alveo and Nvidia AGX Xavier after normalising inference time with respect to frequency.
  - VCU118: this test aims to compare the performance executing the Tiny-YOLOv4 use-case model in the VCU118 SELENE platform using different accelerator configurations (different NN layer distribution on CPU and HW accelerator and different bit number precision). It also compares the performance of the accelerator against the CPU on inference tasks to calculate the impact of implementing the accelerator over the whole platform performance.
  - Xilinx Alveo: this test is based on an inference benchmark (a technology-agnostic evaluation) evaluating HLSinf in a Xilinx Alveo Board in order to validate and evaluate the accelerator in an existing environment to isolate the results from the custom SW stack that is required for VCU118 board. Tiny-YOLOv4 for railway signalling detection was evaluated on a Xilinx Alveo with external Intel CPUs facilitating the evaluation of the accelerator isolated from CPU performance.

  - Nvidia Jetson AGX Xavier: the same image inference test is executed in the GPU of the SoA edge computing platforms.

- Distance calculus performance test: an evaluation on SGBM performance is also a target for the test. The performance of the SGBM algorithm also allows a CPU speed evaluation.
- Process isolation test: unfortunately, the process of porting the Jailhouse hypervisor to the SELENE platform could not be completed in time before the end of the project and is still ongoing. However, within this work, the correct functioning of hypervisor has been tested over RISC-V architecture using a QEMU [31] machine emulator and virtualizer. This consists of concurrently executing multiple applications of the use case on a single RICS-V SoC allowing it to evaluate non-interference properties. This also allows any impact on application precision to be evaluated as well as the performance impact of shared/contended resources. First of all, each process has been executed separately to obtain the performance data without interference from other processes. Then, in a second cell, a workload is introduced incrementally based on micro-benchmarks in interference analysis [32]. All combinations to two of the four cells have been tested. The result of these tests has been compared to the initial evaluation performed in isolation to check that performance degradation is bounded and functional behavior remains unaffected. With this configuration, the impact of several types of interference (shared memory, shared cache, shared buses) on each selected algorithm has been studied.
- Redundancy and RV test: two different tests have been carried out for RV evaluation. The first one at use-case level where *PassengerDetector* functionality is executed redundantly on the SELENE platform. The RV is configured for a 2oo2 scheme. The PC is used for interaction with the processes on the SELENE platform. Instead of the real door-closing command system, a stub is running on the PC to receive the command from the *PassengerDetector*.
  Each *PassengerDetector* process sends a vote containing the command value to the RV. In order to simulate the failure, a script has been developed enabling it to be injected in order to vote failure, send a wrong vote and test the system. The RV checks whether both of the two processes send the same vote. The master process checks the result of the RV. If the check was successful, the master process sends a command with the door-closing signal. If the check is not successful, it will send an order to keep the doors opened.
  The second one is related to low-level platform validation, where the RV subsystem has been validated by means of FPGA-based Fault Injection (FFI). This application performs a staggered redundant execution of a matrix multiplication kernel with two replicated processes. At the end of the kernel execution, each redundant process calculates the digest (CRC32) for the output results. These digests (from each process) are loaded to the dedicated dataset registers of the RV cell. For the sake of simplicity, this application uses only one RV cell. The voting scheme configured for the RV cell is 2oo2, and the configured timeout (maximum time to wait for the datasets) is 1 ms.
  FFI experiments have been carried out using a customized version of DAVOS [33] fault injection tool. Faults have been injected into the CPU cores: Cell C (which executes one of the kernel replicas), and Cell B (which executes the monitoring process). The considered faultload comprises single bit-flips in those cells of FPGA configuration memory that configure targeted SoC components (CPU cores). A total of ten thousand faults have been injected during FFI experiments (5000 faults per each targeted CPU core).
  The outcome of each individual injection run (fault effect) is described in terms of failure modes. The fault is masked when it produces no effect on the system. The fault leads to Replica fail when the RV raises the validity flag for one of the replicas. The fault leads to replica timeout when the RV raises the timeout flag for one of the replicas. Finally, the fault effect is double the modular redundancy fail when the RV is unable to establish an agreement, and the kernel result does not match the fault-free run.

At the end of the experiment, DAVOS calculates the percentage of each failure mode as the ratio between the number of registered failure modes of each type and the total number of injected faults.

6. Test Results

This section shows the results for the SELENE platform and compares the results with SoA platforms.

6.1. AI Model Inference Performance Results

The results for the evaluation can be seen in Table 1, together with the evaluation results printed on the input image in Figure 4. In the figure, we can see several columns of execution times of the inference of one image using the Tiny-YOLOv4 model at different platforms.

Table 1. Tiny-YOLOv4 inference times on SELENE's VCU118 board and the comparison with (a) the inference when SELENE's HW accelerator is executed at Xilinx Alveo (also using EDDL) (b) the inference when the use case is executed on the GPU of AGX Xavier and (c) the inference at CPU frequency downscaled AGX Xavier (in order to be able to set a same level comparison). "Transform" and "Others" layers are executed in CPUs. All measurements are given in milliseconds (ms).

	Tiny YOLOv4 Layers	**VCU118 (100 MHz) (SELENE)**	**Alveo (250 MHz) (HW-Acc of SELENE)**	**AGX Xavier GPU (1.377 MHz)**	**AGX Xavier GPU (250 MHz)**
	All executed on CPU	45,685,922	-	-	-
FP32	HLSinf Transform Others	1799 479 4	364 7 0	17	94
	Total	2282	371		
FP16	HLSinf Transform Others	1548 726 4	119 6 2	13	72
	Total	2278	127		
INT8	HLSinf Transform Others	796 1135 4	66 15 1	N/A	N/A
	Total	1935	82		

(**a**) (**b**)

Figure 4. Platform landmark (**a**) and Passenger (**b**) detectionsfor automatic accurate stopping and safe passenger use case. Note: (**b**) image corresponds to a platform shot. It is an example that emulates the images captured from the rear-view mirrors of the train, as the images captured for these tests are not publishable due to GDPR issues.

The VCU118 (SELENE) corresponds to the SELENE platform, using the HLSinf accelerator as the AI hardware accelerator of the platform, the EDDL library as an inference library and the SAF as the interface between the HLSinf accelerator (100 MHz) and the EDDL library. The Alveo corresponds to the inference time of the inference outside the

SELENE platform, using an Intel i7-7800-X (3.45 GHz) for the non-supported HLSinf layers and the HLSinf accelerator (200–250 MHz) deployed on the Alveo U200 board for the supported HLSinf layers. The AGX Xavier GPU corresponds to inference on the GPU of the Jetson family AGX board. Finally, the last column corresponds to the GPU downscaled because the results needed to be adjusted for the frequency of the GPU (1377 MHz) to equal the frequency of the Xilinx Alveo (200–250 MHz) in order to compare the performance of the accelerator isolated from the underlying physical technology, which limits the operation frequency.

The SELENE VCU118 platform results in Table 1 include different layer execution configuration and bit precision levels. Note that the time for one forward operation in the CPU is 4,568,592 ms, while in FP32 using the HLSinf accelerator the time lowers to 2282 ms. This result means an acceleration factor of $\times$2002 that rises to $\times$2361 when running on INT8 precision.

Comparing the GPU downscaled and the Alveo EDDL columns, the GPU behaves better while using FP32 precision. On the FP16, the HLSinf accelerator achieves 119 ms inference time per image. Taking into account that not all the layers are embedded in the accelerator, it produces slightly more inference time than downscaled. When the precision falls back to INT8, the inference time for HLSinf is 66 ms per image, together with the CPU preprocessing time required, the time to execute a forward pass on one image is 82 ms. Unfortunately, the comparison at INT8 precision is not possible as the GPU available drivers do not handle fewer than 16 bits per parameter.

6.2. Distance Calculus Performance Results

The results represented in Table 2 show that the actual inference time in the SELENE platform (two cores RISC-V CPU) is much higher than in the AGX Xavier (eight cores ARM CPU) but a direct comparison is not representative. Cumulative processing time over all cores must be calculated to obtain the computing time for all processes. In the full process time, the results show that the RISC-V CPU performance is 6.66% of the ARM performance, however operation frequency is not the same in both platforms, so frequency normalization shall be applied to obtain the actual performance for the CPU. SELENE platform CPUs run at 100 MHz and the ARM CPU runs at 2.2 GHz. The results normalizing the frequency show that the RISC-V CPU performance is actually greater than the ARM CPU.

Table 2. Depth map calculus execution times (OpenCV SGBM function in CPU). SELENE's VCU118 and AGX Xavier. The last two columns show the comparison between both platform: Raw core number normalization and Frequency Agnostic (F.A.) (100 MHz vs. 2.2 GHz) downscaled. All measurements are given in seconds (s).

					VCU118 w.r.t AGX Xavier	
	VCU118 (2 Core) SELENE		**AGX Xavier (ARM 8 Core)**		**Raw Comp.**	**F.A. Comp.**
OpenCV SGBM	Total time	Time Using Single Core	Total time	Time Using Single Core	%	%
Matching Time	48	96	0.8	6.4	6.66	146.67
Filtering Time	20	40	0.3	2.4	6	132

This test has also been used to check OpenCV compatibility and performance of an algorithm to estimate the distance from the train cabin to a stop signal on the platform. After compiling and installing OpenCV for RISC-V 64-bit architecture, a performance test consisting of the execution of the Semi-Global Block Matching (SGBM) function of OpenCV was carried out. As shown in Figure 5, the OpenCV SGBM function takes two stereo images (taken with a stereo camera, producing a left and right image) and tries to match the images creating, as a result, a disparity map which represents the distance between the detected landmarks or people to the cameras on the train.

(**a**) (**b**) (**c**)

Figure 5. Distance calculusto platform landmark using stereo vision camera (left (**a**) and right (**b**) images) and extracted depth map (**c**). Green frames represent detected landmark bounding box and the red frame the landmark's corresponding area in depth map. This are is used for distance calculus.

Because OpenCV acceleration was not implemented in SELENE platform (it is planned as future work), the SGBM algorithm is just executed in CPU. Table 2 shows the execution times of the SGBM function (divided in Matching Time and Filtering Time) and the comparison with AGX Xavier board execution in its ARM CPU cores.

As expected, the Nvidia AGX Xavier with its eight cores at 2.2 GHz is much faster than the two core SELENE VCU118 at 100 MHz, however, as previously mentioned in this work, this direct comparison is not valid as the SELENE platform is an evaluation HW FPGA board with a frequency much lower than an ASIC implementation. Therefore, the comparison must be normalized to be agnostic of the frequency and the number of cores. After normalizing the cores, the SELENE platform reaches just 6.66% of the performance of the Nvidia AGX Xavier. However, after normalizing the frequencies, the RSIC-V CPU in the SELENE outperforms the ARM, demonstrating that FPGAs can be a valid option from the performance point of view.

6.3. Process Isolation Results

This Jailhouse hypervisor version was successfully executed on the QEMU, with execution of different Linux root (Safety and NoSafety) cells. The use of resources has been monitored validating the isolation capabilities of SELENE platform (shared memory, shared cache, shared buses).

Additionally, we created a simple inmate trying to escape its cell by accessing outside of its allocated memory. This attempt was correctly caught by the hypervisor that sanctioned the faulty access by a page fault exception.

6.4. Redundancy and RootVoter Results

At the use case level, all tests regarding the RV were successful. If the two scripts sent the same vote, the door enable signal is activated. If the two scripts sent a different vote, this failure is successfully detected by the RV and the doors remains closed and blocked.

At the platform level, the results of FFI show that the system has tolerated all faults injected into the kernel replica (Cell C), i.e., 0.00% of 2002 failures. The replicas themselves are quite sensitive to the injected faults: in 0.88% of cases the RV has reported a replica fail, and in 0.16% other cases the RV has reported a replica timeout. The faults injected into the monitoring process (Cell B) have not affected the behaviour of the kernel replicas, and only one 2oo2 failure per 5000 faults has been detected (0.02%).

In such a way, the described experiment has shown that the RV meets its specified functionality, i.e., it detects the errors and timeouts of replicated processes, and it establishes an agreement following the configured voting scheme. Usage of RV in the redundant applications efficiently protects the system against the faults of the replicated processes.

7. Discussion and Future Work

In this work, a new safety-critical and high-performance computing application for real-time AI-enhanced railway use has been introduced. Its design allows process isolation,

redundant execution, HW acceleration and abstraction making the platform compatible for most widely used AI inference techniques and AI model architecture and formats (including open standards such as ONNX).

It is worth highlighting the implementation of specific HW and SW modules for the SELENE platform. A HW accelerator module, which can be customized to support specific data formats and neural network layers, has been deployed. HLSinf accelerator shows great performance on frequency agnostic evaluation. Using quantization and other AI model optimisation methods, the performance improves SoA. Its implementation presents a very high acceleration factor with respect to CPU execution for the Tiny-YOLOv4 algorithm. This work also presents the performance and accuracy evaluation of the use case functionalities over the SELENE platform, comparing it with executions carried out in the most widely used commercial HPC platforms, such as Nvidia's Jetson family boards or Xillinx's FPGAs.

In addition, a custom AI runtime and adapted inference SW, which abstracts the user application layer from platform specific HW configuration, has been carried out.

Finally, the rootover and Jailhouse hypervisor implementations for RISC-V based system compatibility have also been successfully validated, making it possible to execute safety-related functionalities on the platform. This solution guarantees isolating executions of the different functionalities and allows the evaluation of redundant executions with voting system when needed.

Regarding the use case, this work has demonstrated to be a valid HW platform for equipping autonomous trains that require real-time execution of safety (precision stop functionality) and non-safety (precision stop) functions based on CV and AI. With higher maturity, ASIC implementation and railway certification, the SELENE platform could suit railway industry requirements for both non-safety and safety level applications.

The next steps of the investigation will focus on improving real time execution performance (reaching lower inference times) while keeping/increasing the detection accuracy. New NN architectures will be taken into account as candidates to port them into the SELENE board. OpenCV acceleration by HW should be implemented in order to speed up basic computer vision algorithms such as SGBM. On the other hand, they will also focus on more in-depth testing and validating the platform's possibilities for redundant execution (followed by different voting systems such as 2oo3) in order to increase the safety level.

Author Contributions: Conceptualization, M.L. and J.F.; Methodology, M.L., J.F. and N.A.; Software, M.L., L.M. and F.E.; Validation, M.L., L.M. and F.E.; Formal analysis, M.L., L.M. and F.E.; Investigation, M.L., L.M., F.E., J.F. and N.A.; Writing—original draft, M.L.; Writing—review & editing, L.M., F.E. and N.A.; Supervision, J.F. and N.A.; Project administration, M.L. and J.F. All authors have read and agreed to the published version of this manuscript.

Funding: The novel approach presented in this work is being developed as a specific railway use case for autonomous train operation into SELENE European research project. This project has received funding from RIA—Research and Innovation action under grant agreement No. 871467.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: There is no available data.

Acknowledgments: The author like to thank the whole PERception (PER) development team of CAF Signalling for their help and support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shift2Rail —Home. Available online: https://shift2rail.org/ (accessed on 5 May 2023).
2. Reddi, V.J.e.a. MLPerf Inference Benchmark. *arXiv* **2019**, arXiv:1911.02549. [CrossRef]
3. CORDIS—SELENE. Available online: https://cordis.europa.eu/project/id/871467/en (accessed on 5 May 2023).
4. Waterman, A.; Lee, Y.; Patterson, D.A.; Asanović, K. *The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.0.*; Technical Report UCB/EECS-2014-54; EECS Department, University of California: Berkeley, CA, USA, 2014.

5. Palmer, A.W.; Sema, A.; Martens, W.; Rudolph, P.; Waizenegger, W. The Autonomous Siemens Tram. In Proceedings of the IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; IEEE: Piscataway, NJ, USA, 2020. [CrossRef]
6. Guerrieri, M.; Parla, G. Smart Tramway Systems for Smart Cities: A Deep Learning Application in ADAS Systems. *Int. J. Intell. Transp. Syst. Res.* **2022**, *20*, 745–758. [CrossRef]
7. Ristić-Durrant, D.; Franke, M.; Michels, K. A Review of Vision-Based On-Board Obstacle Detection and Distance Estimation in Railways. *Sensors* **2021**, *21*, 3452. [CrossRef] [PubMed]
8. Alstom Demonstrates Fully Autonomous Driving of a Shunting Locomotive in the Netherlands. Available online: https://www.alstom.com/press-releases-news/2022/11/alstom-demonstrates-fully-autonomous-driving-shunting-locomotive-netherlands (accessed on 5 May 2023).
9. Train Autonome Service Voyageurs: Essais Réussis. Available online: https://www.youtube.com/watch?v=vlEy7GYe684&ab_channel=GroupeSNCF (accessed on 5 May 2023).
10. Autonomous Train Tests Were Carried Out Succesfully in Finland. Available online: https://www.proxion.fi/en/autonomous-train-tests-were-carried-out-succesfully-in-finland/ (accessed on 5 May 2023).
11. Railtech—DB Cargo Automates Shunting to Boost Single Wagon Load Traffic. Available online: https://www.railfreight.com/railfreight/2021/10/27/db-cargo-automates-shunting-to-boost-single-wagon-load-traffic/?gdpr=accept (accessed on 5 May 2023).
12. Cognitive Pilot—Tram Automation Software Contract Awarded in Shanghai. Available online: https://en.cognitivepilot.com/breaking-news/fitsco-tram-english/ (accessed on 5 May 2023).
13. RailTech—Remote-Controlled Shunting. Available online: https://www.railtech.com/digitalisation/2020/09/25/remote-controlled-shunting-on-tests-in-switzerland/ (accessed on 5 May 2023).
14. Digitale Schiene—Fourteen Eyes on the Road Ahead: Second Sensors4Rail Test Project Successful. Available online: https://digitale-schiene-deutschland.de/en/Sensors4Rail-test-project (accessed on 5 May 2023).
15. Youtube—Train Autonome: Automatisation de la Lecture de la Signalisation Latérale. Available online: https://www.youtube.com/watch?v=WiYavvqh7Bk&ab_channel=GroupeSNCF (accessed on 5 May 2023).
16. La Reconnaissance Faciale des Signaux, le Projet ARTE D'alstom. Available online: https://mediarail.wordpress.com/2022/10/23/alstom-projet-arte-basse-saxe/ (accessed on 5 May 2023).
17. Perez-Cerrolaza, J.; Obermaisser, R.; Abella, J.; Cazorla, F.; Grüttner, K.; Agirre, I.; Ahmadian, H.; Allende, I. Multi-Core Devices for Safety-Critical Systems: A Survey. *ACM Comput. Surv.* **2020**, *53*, 1–38. [CrossRef]
18. Mc Guire, N.; Allende, I. Approaching certification of complex systems. In Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Valencia, Spain, 29 June–2 July 2020; pp. 70–71. [CrossRef]
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef].
20. Hirschmuller, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 328–341. [CrossRef] [PubMed]
21. Siemens. JAILHOUSE. Available online: https://github.com/siemens/jailhouse (accessed on 5 May 2023).
22. Gerstinger, A.; Kantz, H.; Scherrer, C. TAS Control Platform: A Platform for Safety-Critical Railway Applications. *ERCIM News* **2008**, *2008*.
23. Cancilla, M.; Canalini, L.; Bolelli, F.; Allegretti, S.; Carrión, S.; Paredes, R.; Gómez, J.A.; Leo, S.; Piras, M.E.; Pireddu, L.; et al. The DeepHealth Toolkit: A Unified Framework to Boost Biomedical Applications. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 9881–9888. [CrossRef]
24. Flich, J.; Medina, L.; Catalán, I.; Hernández, C.; Bragagnolo, A.; Auzanneau, F.; Briand, D. Efficient Inference Of Image-Based Neural Network Models In Reconfigurable Systems With Pruning And Quantization. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 2491–2495. [CrossRef]
25. NOEL-V. Available online: https://www.gaisler.com/index.php/products/processors/noel-v (accessed on 5 May 2023).
26. OpenCV. Available online: https://opencv.org/ (accessed on 5 May 2023).
27. Accelerating DNNs with Xilinx Alveo Accelerator Cards. Available online: https://docs.xilinx.com/v/u/en-US/wp504-accel-dnns (accessed on 5 May 2023).
28. Jetson AGX Xavier and the New Era of Autonomous Machines. Available online: https://info.nvidia.com/rs/156-OFN-742/images/Jetson_AGX_Xavier_New_Era_Autonomous_Machines.pdf (accessed on 5 May 2023).
29. Etxeberria-Garcia, M.; Zamalloa, M.; Arana-Arexolaleiba, N.; Labayen, M. Visual Odometry in Challenging Environments: An Urban Underground Railway Scenario Case. *IEEE Access* **2022**, *10*, 69200–69215. [CrossRef]
30. Biddle, P.; England, P.; Peinado, M.; Willman, B. The Darknet and the Future of Content Protection. In Proceedings of the ACM Workshop on Digital Rights Management, Washington, DC, USA, 18 November 2002; Volume 2696, pp. 155–176. [CrossRef]
31. QEMU—A Generic and Open Source Machine Emulator and Virtualizer. Available online: https://www.qemu.org/ (accessed on 5 May 2023).

32. Ensuring Software Timing Behavior in Critical Multicore-Based Embedded Systems. Available online: https://www.embedded.com/ensuring-software-timing-behavior-in-critical-multicore-based-embedded-systems/ (accessed on 5 May 2023).
33. DAVOS—A Fault Injection Toolkit for Dependability Assessment, Verification, Optimization and Selection of Hardware Desings. Available online: https://github.com/IlyaTuzov/DAVOS (accessed on 5 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.