Check for updates

ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

Vehicle Surroundings Perception Using Micro-electromechanical Systems Inertial Sensors

Martin Lahr, Johnson Loh, Mike Schwarz, Max Christian Lemme, and Tobias Gemmeke*

Sensors enable vehicles to perceive their surroundings even while parked. Camera-based systems for vehicle surroundings perception are already available as of today. However, camera-based systems result in excessive power consumption and data collection. In this article, a study on always-on surroundings perception of parked vehicles is done using micro-electromechanical systems (MEMS) inertial sensors as an alternative to camera-based systems. The measurements of the MEMS inertial sensors are used to detect and classify 17 different types of events. These events include accidents, vandalism, rain, as well as normal use. A convolutional neural network (CNN) is used to classify the events. The data used to train the CNN is recorded using a test vehicle in a laboratory setting and on a test track. The CNN is quantized for deployment on a CNN hardware accelerator which is packaged with the sensor as a system-in-package (SiP). The novel sensor system achieves a classification accuracy of 88% at a total system power dissipation of 27 µW.

1. Introduction

Connected and autonomous vehicles depend on an array of sensor systems to perceive their surroundings. These encompass a variety of technologies, including camera-based visual systems,[1] inertial measurement units (IMUs), radar,[2] and GPS, among others. These sensor systems primarily operate when the vehicle is in motion. However, vehicles spend a substantial amount of their life cycle parked and stationary. During these idle periods, most of the sensor systems are inactive, which makes the vehicle susceptible to potential risks such as accidents, vandalism, and environmental impacts. Traditional security measures, such as car alarms and anti-theft systems, offer some level of protection but may not provide comprehensive coverage. This necessitates the development of novel sensor-based systems that can continuously monitor parked vehicles, analyze their surroundings, and alert vehicle owners in case of incidents.

In the context of vehicle surroundings perception, one notable technology is Tesla's "Sentry Mode".[3,4] Sentry Mode uses a 360° surround camera system to detect approaching individuals around the vehicle, enabling it to identify potentially suspicious activities and trigger additional vehicle systems, such as alarms. However, camera-based surroundings perception has several drawbacks. Firstly, cameras and object detection processors demand a substantial amount of power, which can deplete the vehicle's battery over time. Secondly, always-on camera systems may potentially violate data privacy regulations, as they continuously record their surroundings. This privacy concern was exemplified in the summer of 2022 when Tesla vehicles were banned in the Chinese district of Beidaihe during a political event, presumably due to the continuous operation of their camera systems.[4] Hence, there exists a need for alternative technologies which do not suffer from these drawbacks.

In this article, we present a novel sensor system for vehicle surroundings perception which is based on micro-electromechanical systems (MEMS) inertial sensors. This approach offers distinct advantages when compared to camera-based systems. Firstly, low-power MEMS sensors allow for continuous operation with minimal power consumption in the microwatt range. As a consequence, they can be operated for long periods of time without depleting the vehicle's battery. Secondly, the inertial sensor signals do not contain any personal information and are, therefore, compliant with data privacy regulations. These properties make MEMS sensors an ideal choice for vehicle surroundings perception.

M. Lahr, M. C. Lemme
Chair of Electronic Devices
RWTH Aachen University
52074 Aachen, Germany
E-mail: martin.lahr@de.bosch.com

M. Lahr
Automotive Electronics
Robert Bosch GmbH
72762 Reutlingen, Germany

J. Loh, T. Gemmeke
Chair of Integrated Digital Systems and Circuit Design
RWTH Aachen University
52074 Aachen, Germany

M. Schwarz
Department of Electrical Engineering and Information Technology
Technische Hochschule Mittelhessen
35390 Gießen, Germany

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202300679.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300679

1.1. Related Work

Camera-based systems for vehicle surroundings perception are already commercially available. To the best of our knowledge, MEMS inertial sensors have not been studied for this application. However, MEMS inertial sensors are frequently used for event detection in both consumer and industrial applications. Examples include predictive maintenance,[5,6] human activity recognition,[7–9] and gait analysis.[10,11]

1.1.1. Camera-Based Perception

The idea of camera-based perception is to detect objects of interest in a camera image. Applications include autonomous driving as well as image classification. For example, camera-based perception may be used for visual inspection of steel and concrete structures.[12,13] Using a CNN, structural damages can be detected with an accuracy of up to 98%. Camera-based perception is also interesting for vehicle surroundings perception as cameras are increasingly integrated into vehicles. From a functional perspective, visual indicators for interesting events, such as car theft attempts, can be detected and trigger a corresponding response.[3] The efficient deployment is key for long-term operation to avoid depleting the car's battery. However, it is rarely analyzed how much power camera-based systems, such as Tesla's Sentry Mode, consume in action.

1.1.2. Predictive Maintenance

Predictive maintenance is an interesting area of application for MEMS inertial sensors.[5,6] For example, vibration signals have been be used to detect faults in rolling element bearings of electric motors.[5] To this end, vibration signals were recorded at various locations of an electric motor with sampling rates of 12 and 48 kHz. The primary objective was to detect the presence and severity of mechanical faults within rolling element bearings. To simulate these faults, mechanical defects were introduced into the bearings and the motor's vibrations were recorded. Using signal processing methods such as envelope analysis, many of the bearing faults could be detected.[5] This was further advanced by employing a CNN for bearing fault classification.[14] The CNN was able to identify different categories of bearing faults with an accuracy of 99%. This included cases which were hard to identify using signal processing techniques.[5] The results demonstrate that CNNs are well-suited for the classification of inertial sensor measurements.

1.1.3. Smart MEMS Sensors

As of today, many machine learning applications involve separate devices for sensing and signal processing. In fact, signal processing is often performed on GPUs which require several orders of magnitude more power than sensing itself. Moreover, delays may arise due to the communication between the sensor and the processor. To address these problems, a tight integration between sensing and signal processing is required. Such an integration is realized by "smart MEMS sensors".[15] Smart MEMS sensors integrate conventional MEMS sensors with low-power microcontrollers on a single chip. However, general-purpose microcontrollers have very limited resources which pose a challenge to deep neural networks.[16,17] To address this challenge, we extend the concept of smart MEMS sensors further by integrating a hardware accelerator for CNNs.

1.2. Our Contributions

In this article, we make several contributions to the field of machine learning: 1) Vehicle Surroundings Perception Dataset: We present a dataset consisting of IMU measurements and event labels which is designed specifically for the use-case of vehicle surroundings perception. The dataset is published online and free for anyone to use (https://github.com/boschresearch/mems-vehicle-surroundings-perception); 2) CNN-Based Event Detection: Based on our dataset, we develop and train a CNN model for event detection; and 3) Smart MEMS Sensors: We present the first MEMS sensor with integrated CNN hardware accelerator. This sensor allows to execute CNN models in real-time without the need for an additional processor.

The paper's structure is organized as follows: in Section 2, we provide an overview of the vehicle surroundings perception use-case and present details about our dataset. Section 3 describes the architecture and training of the CNN model for event detection, as well as the integration of the hardware accelerator. The results of our analyses are presented in Section 4. Finally, we summarize our findings in Section 5.

2. Vehicle Surroundings Perception

In this section, we describe the use-case of vehicle surroundings perception, the measurement protocol used to record the dataset, and the dataset itself.

2.1. Use-Case

The idea of vehicle surroundings perception is to detect and classify events in the vicinity of parked vehicles. We consider a total of 17 different events which are listed in **Table 1**. These events include different types of accidents, vandalism, and other mechanical impacts. If certain events are detected, additional vehicle systems may be activated. For instance, if a parking collision is detected the owner of the vehicle may be notified via smartphone and cameras may be activated to document the incident. Likewise, if heavy rain is detected, the owner may be notified to relocate their vehicle to avoid potential damage. In cases of vandalism, an alarm may be activated.

To achieve these goals, the sensor system must be active even when the vehicle is turned off. This requires a low power design for both sensing and classification so that the system can be operated for long periods of time without depleting the car battery. To this end, we propose a sensor system consisting of a low-power MEMS inertial sensor and a dedicated ultra-low-power hardware accelerator to execute a CNN for event detection.

2.2. Measurement Protocol

To train the CNN model, we collected inertial measurements using a test vehicle. The test vehicle was prepared with three

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 1. Events which are considered for vehicle surroundings perception.

Event type	No. measurements
Collision with car	78
Collision with shopping cart	91
Open car door	58
Close car door	117
Open trunk	15
Close trunk	30
Open fuel cap	20
Close fuel cap	15
Unlock bonnet	15
Lock bonnet	15
Close bonnet	15
Lift car	61
Knocking on window	61
Release wheel nut	63
Scratch door with key	60
Rain	10
Other	99
Total	**823**

sensor modules placed at the roof, rear, and door of the car, as depicted in **Figure 1**. Each of these sensor modules contained a Bosch BMA400 acceleration sensor and a Bosch BMI270 inertial sensor, whose specifications are listed in **Table 2**.

The BMA400 accelerometer measures acceleration in three spatial directions (x, y, and z) while the BMI270 inertial sensor measures both acceleration and angular velocity in three spatial directions. Notably, the BMI270 has a higher sampling rate than the BMA400. Moreover, the BMI270 also has a smaller sensor offset of ± 20 mg. The sensor offset describes the measured acceleration when the actual acceleration is zero and is a crucial performance metric of an accelerometer. Thus, the BMI270 is more accurate than the BMA400. In contrast, the BMI270 also requires significantly more power.

The test vehicle was subjected to different events in a vehicle laboratory and on a test track. The events are listed in Table 1. Each measurement was repeated several times and from different directions. For each measurement, the accelerations and angular velocities of all sensors were recorded. Our setup had 9 measurement channels per location and 27 measurement channels in total.

2.3. Dataset

The dataset contains a total of 823 measurements, which underwent the following preprocessing steps. The signals of the BMA400 accelerometer were upsampled from 100 to 400 Hz and synchronized with the BMI270 signals. From each measurement, a 4-second segment starting at the onset of the event was extracted. Subsequently, these segments were normalized to achieve a zero mean across all channels. This normalization step was implemented to ensure that the event detection algorithm is independent of the slope of the road surface. Then, each segment was labeled with the corresponding event type. Certain event types, such as the passage of another car or a passenger sitting in the vehicle, did not show any noticeable acceleration patterns. These events were collectively labeled as "other". Our final dataset is summarized in **Table 3**.

Figure 2 shows four preprocessed signals captured by the BMA400 acceleration sensor located at the rear. Firstly, Figure 2a shows the measurement of a car door which is closed with regular force. This measurement is characterized by a brief acceleration burst with an amplitude of up to $4\,\mathrm{m\,s^{-2}}$. Secondly, Figure 2b shows the measurement of a parking collision where another vehicle slowly impacts the stationary test vehicle. At the time of impact, the colliding vehicle was driving at walking speed. The parking collision is characterized by a mechanical oscillation of the car chassis with a duration of $\approx$2 s. Thirdly, Figure 2c shows the event "open fuel cap" which has a maximum amplitude of $\approx 0.15\,\mathrm{m\,s^{-2}}$. The amplitude is relatively small and therefore the signal is hard to distinguish from noise. Finally, Figure 2d shows the event "close fuel cap" with a maximum amplitude of $\approx 0.2\,\mathrm{m\,s^{-2}}$. While the amplitude of the event "close fuel cap" is also relatively small, it is slightly larger compared to the event "open fuel cap".

3. Experimental Section

In this section, we describe the methods which are used in this article. We begin by presenting the model architecture of the CNN which is used for event detection. Subsequently, we describe the training and evaluation of the CNN model. Lastly, we present the system architecture of our sensor system.

3.1. CNN Model Architecture

For event detection and classification, we use a CNN with convolutions computed over time. The model architecture of the CNN is shown in **Table 4**. The CNN architecture is inspired by the neural network architecture presented in ref. [18]. For the considered use-case, it required key adjustments in the pre-processing and feature extraction components. In summary, the CNN consists of two downsampling layers for preprocessing, three convolutional layers, a fully-connected layer, and a softmax layer. The downsampling layers downsample the input signals from 400 to 100 Hz using two decimation filter stages. Subsequently, the sensor signals are transformed using convolutions, followed by ReLU activation, and max. pooling with stride 3. The kernel size and pool size parameters were chosen to fit the pre-existing CNN accelerator,[18] while the number of filters were tuned using a parameter sweep. The outputs of the last convolutional layer are flattened and passed through a fully-connected layer and softmax layer for classification. The softmax layer has 17 output neurons which corresponds to the number of events.

3.2. Training & Evaluation

The CNN is trained using the Adam optimizer[19] with fixed learning rate 0.001 and batch size 32 for 100 epochs. The model

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



Figure 1. Our measurement setup. a) Test vehicle including sensor modules which was subjected to different types of mechanical impacts. The insets show the sensor modules which were used to measure the acceleration and angular velocity during the impacts. b) Measurement of another vehicle colliding with our test vehicle while parked.

Table 2. MEMS sensors used for data recording. The power consumption is determined at $V_{DD} = 1.8$ V and suitable performance mode.

Sensor	Measurements	No. channels	Sampling rate [Hz]	Power consumption [W]	Offset [mg]
BMA400	Acceleration	3	100	2.6×10^{-5}	±80
BMI270	Acceleration, angular velocity	6	400	1.2×10^{-3}	±20

Table 3. Summary of our vehicle surroundings perception dataset.

No. signals	No. channels	Signal duration [s]	Sampling rate [Hz]
823	27	4	400

is evaluated using stratified 5-fold cross validation (CV).[20] To this end, the dataset is partitioned into five folds such that each fold contains the same number of samples per class. Subsequently, the model is trained on four folds while the fifth fold is used for evaluation. This process is repeated five times for all possible combinations of training and test folds. The entire 5-fold cross-validation is then repeated $N = 20$ times. This is done to minimize statistical influences of the training process due to random weight initialization and stochastic optimization.

3.3. End-to-End Sensor System Architecture

As ultra-low power consumption is necessary in an always-on application scenario, a dedicated IC, such as an application-specific integrated circuit (ASIC), is suited for the implementation of the digital signal processing. Here, the hardware architecture from ref. [18] is used for evaluation. Note, that the choice of this implementation is mainly reasoned by both the close relation of the CNN model (see Table 4) and the stream processing use-case. Due to the temporal nature of input samples and output features, the data-driven processing scheme is especially efficient for low-rate input streams, since it triggers sub-computations of the CNN model on demand. The power consumption is reported in the order of sub- µW, which comprises only a fraction of the sensor's power

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

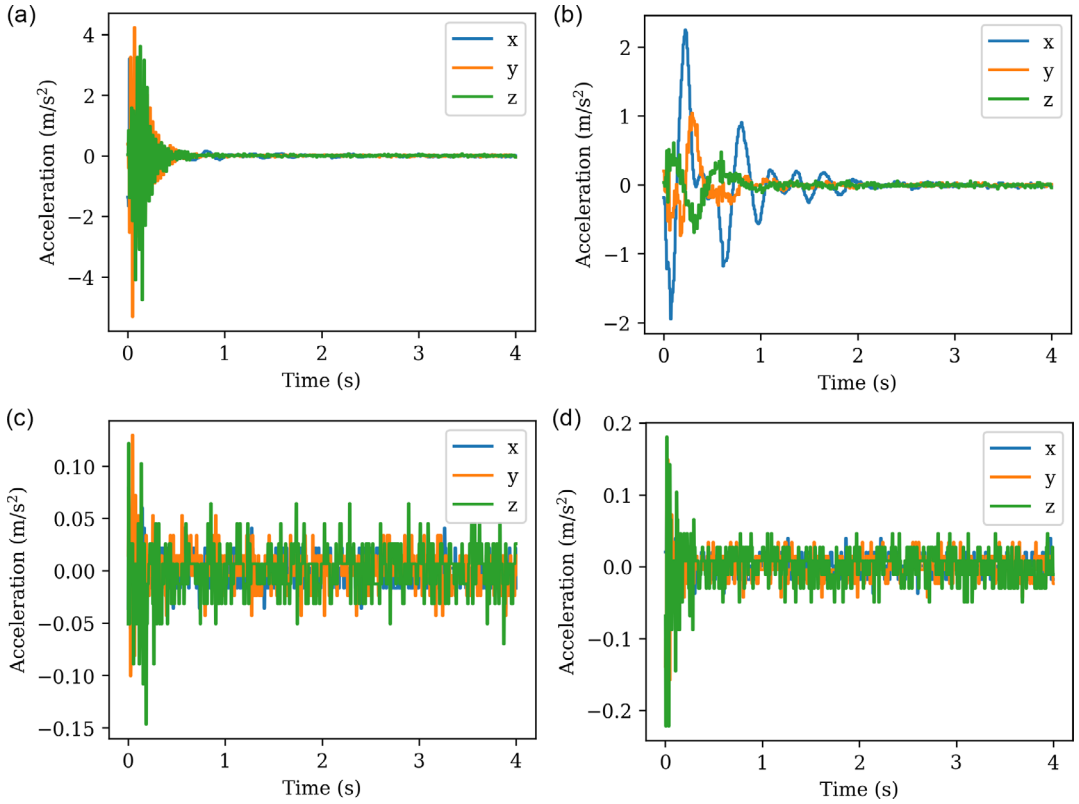


Figure 2. Preprocessed acceleration signals of different events. a) Event "close car door", b) Event "collision with a car", c) Event "open fuel cap", and d) Event "close fuel cap".

Table 4. Model architecture of the CNN used for event detection. The signal dimensions are given as channels × length.

Layer	No. filters	Kernel size	Input dims.	Output dims.
Downsampling ↓ 2	–	5	27 × 1600	27 × 800
Downsampling ↓ 2	–	5	27 × 800	27 × 400
Conv. + ReLU + MaxPooling	12	5	27 × 400	12 × 132
Conv. + ReLU + MaxPooling	24	5	12 × 132	24 × 42
Conv. + ReLU + MaxPooling	48	5	24 × 42	48 × 12
Flatten	–	–	48 × 12	576
Fully connected	–	–	576	17
Softmax	–	–	17	17

(see Table 2). However, number of MAC operations and required precision of the classifier model differ, due to proposed novel application scenario. Therefore, the design space is explored systematically in Section 4.

Figure 3 shows the MEMS sensor integrated with a digital back-end for edge computing. As the integration of MEMS and IC can either be done on the same substrate or by combining multiple chips through packaging,[21] we evaluate our setup using the latter method. Multi-chip solutions enable the integration of chips with different technology nodes in a cost-effective manner.

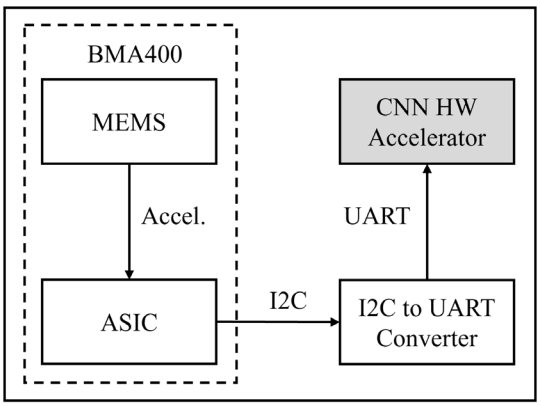


Figure 3. Block diagram of the sensor system architecture designed as system-in-package.

4. Results and Discussion

In this section, we present the results of our event detection algorithm and the overall sensor system used for vehicle surroundings perception.

4.1. Event Detection

We evaluate the classification accuracy of the event detection algorithm for different types of events. For this analysis, the

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

CNN is trained with $N = 20$ repetitions as described in Section 3.2. The results of this analysis are shown in the confusion matrix in **Figure 4**. The confusion matrix shows the number of predicted events versus actual events in percent. We find that the events are detected correctly with an average accuracy of 89%. Notably, the events "close engine bonnet", "close car door", and "open trunk" are detected with the highest accuracy of 100%. These events can be detected very reliably using MEMS inertial sensors. This may be attributed to the distinct acceleration pattern seen in Figure 2a and the relatively high acceleration amplitude. Furthermore, the events "collision with a shopping cart" and "collision with a car" are detected with an accuracy of 96% and 90%, respectively. Thus, parking collisions can also be detected relatively reliably.

While these events can be detected well using MEMS inertial sensors, others cannot be detected very well. In particular, the events "close fuel cap" and "open fuel cap" have the lowest classification accuracies of 46% and 67%, respectively. The reason for the low classification accuracies is that these events are often confused with other events. Specifically, the event "close fuel cap" is confused with the event "open fuel cap" 48% of the time while the event "open fuel cap" is confused with the events "other" and "close fuel cap" 18% and 13% of the time, respectively. It is worth noting, that the events "open fuel cap" and "close fuel cap" have relatively small acceleration amplitudes, see Figure 2. Furthermore, the events occur at the same location which makes it hard to distinguish them. As a consequence, these events cannot be detected very well in our setup. However, we note that these events represent different motions which potentially could be distinguished if the sensors were placed closer to the fuel cap.

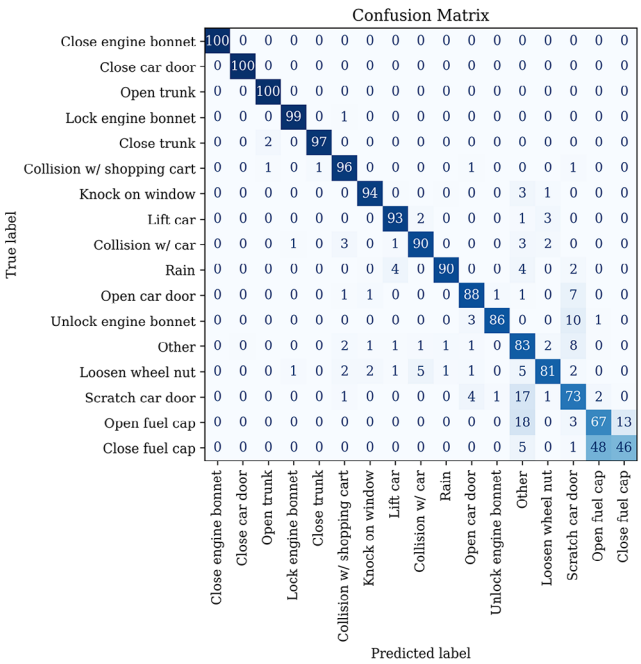


Figure 4. Confusion matrix of the event detection algorithm. The confusion matrix shows predicted events versus actual events in percent.

In summary, we find that many events can be detected reliably using MEMS inertial sensors. These events are characterized by large acceleration amplitudes and distinct signal patterns. In contrast, some events cannot be detected very well. These events are characterized by relatively small acceleration amplitudes.

4.2. Influence of the Sensor Location

The location of the sensor relative to the vehicle is a crucial factor for our application. Therefore, we investigate the influence of the sensor location on the classification accuracy. To this end, the CNN is trained multiple times using only a subset of the available sensors. Specifically, the CNN is trained using the sensors located at the door, rear, and roof separately. For this analysis, the input layer of the CNN described in Table 4 is adapted to the input dimensions, while the other layers remain unchanged.

The results are shown in the boxplot in **Figure 5** for $N = 20$ repetitions. The boxplot shows the classification accuracies for each individual sensor location as well as for all sensors combined. When only one sensor location is used, the rear provides the highest accuracy of 87%, followed by the roof with 86% and the door with 85%. Our results indicate that the MEMS sensors located at the rear provide the most important information for the event detection algorithm. The reason for this could be the shorter distance between the event source and the sensor as well as the symmetrical position with respect to the vehicle's axis. However, it should be noted that the accuracies vary only within 2% across the different sensor locations. Therefore, we note that while the sensor location has an influence on the event detection, this influence is only modest. When all sensors are combined, the classification accuracy increases to 89% which is 2% higher compared to using only a single location. This suggests, that by placing multiple MEMS sensors at different locations across the vehicle, the classification accuracy can be improved.

4.3. Influence of the Sensor Type

The type of the sensor is another important factor for our application. Therefore, we investigate the influence of the sensor type

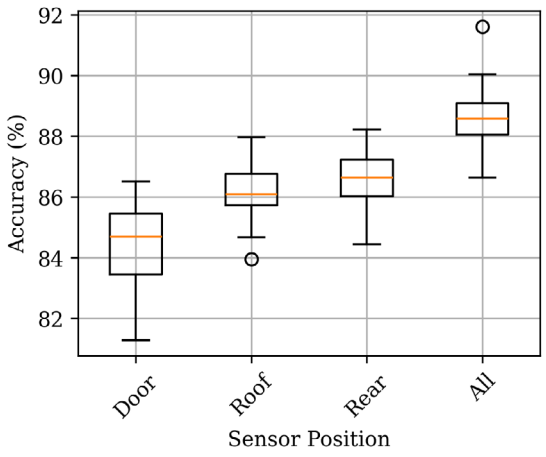


Figure 5. Classification accuracy of the CNN for different sensor locations.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

on the event detection algorithm. To this end, the CNN is trained multiple times using either the BMA400 or BMI270 measurements. The BMI270 measurements are further divided into acceleration measurements, gyroscope measurements, and both. This separation is done to assess the individual contributions of the accelerometer and gyroscope.

The results of this analysis are shown in **Figure 6**. The BMI270 acceleration signals yield the highest classification accuracy of 89%, followed by the BMA400 acceleration signals with 88%. While this improvement is relatively small, it could be attributed to the higher sensor accuracy and sampling rate as detailed in Table 2. In contrast, the gyroscope measurements of the BMI270 yield the lowest classification accuracy of 84%. When combining both gyroscope and acceleration measurements of the BMI270, the accuracy improves from 84% to 87%. However, this is still lower compared to using only the acceleration measurements of the BMI270. While this observation is counter-intuitive, it could be explained by the model's inability to identify the most relevant signals. It is expected that this observation vanishes when more training data is available. Finally, we note that when all sensors are combined, the events can be detected with an accuracy of 89% which is the same as using only the BMI270 acceleration signals.

In summary, we find that acceleration measurements of the BMI270 yield the highest classification accuracy. When comparing accelerometers and gyroscopes, we find that accelerometers generally yield higher classification accuracies. This suggests that accelerometers are better suited for vehicle surroundings perception than gyroscopes.

4.4. CNN Quantization

Based on the results of Sections 4.2 and 4.3, we choose a single-sensor setup for further investigation. For this setup, we choose the BMA400 accelerometer located at the rear. The rear location was chosen because it yielded the highest classification accuracy out of the different sensor locations. Furthermore, the BMA400 yielded a relatively high classification accuracy while consuming significantly less power than the BMI270, which is crucial for our application.

The BMA400 acceleration signals are fed into the CNN hardware accelerator as shown in Figure 3. Although the hardware accelerator in ref. [18] generally fits proposed CNN model, an appropriate degree of quantization needs to be found for fixed-point arithmetic, while keeping classification accuracy high. In a first step, the softmax layer is removed from the full-precision baseline, since it does not change the prediction result and omits logic for the non-linear function. Then, the activation distribution is investigated to fit the required number range to fixed-point representations. The number ranges of coefficients and activations from preliminary downsampling and CNN are defined separately, such that the maximum value $x_{\max}$ is within $2^{n-2} \leq x_{\max} < 2^{n-1}$ with n being the number of required integer bits. Since an equal word length is assumed for all coefficients and intermediate results, additional bits are used to increase the resolution with fractional bits.

Figure 7 shows the sweep across a range of word lengths. A steep decrease of accuracy on the test set is observed starting at $n = 16$ bits. To find a sufficient trade-off between quantization resolution and achieved accuracy, we chose the design at the knee point with $n = 16$ (test set accuracy: 88%) for further investigation.

4.5. System Performance Evaluation

The measurement results of the fabricated chip in ref. [18] are used to calibrate the CNN accelerator power consumption of the proposed use-case. In this case, the word length and the number of MAC operations per prediction are used for scaling dynamic power, while static power is assumed to remain constant. First, we use post-synthesis results of MAC processing elements in ref. [22] to normalize from $n = 12$ to $n = 16$ bits. In a second step, the classifier size is adjusted according to their computational complexity. Here, we assume a linear dependency of dynamic power and the total number of MAC operations.

The relative difference between reference and our use-case results is summarized in Equation (1).

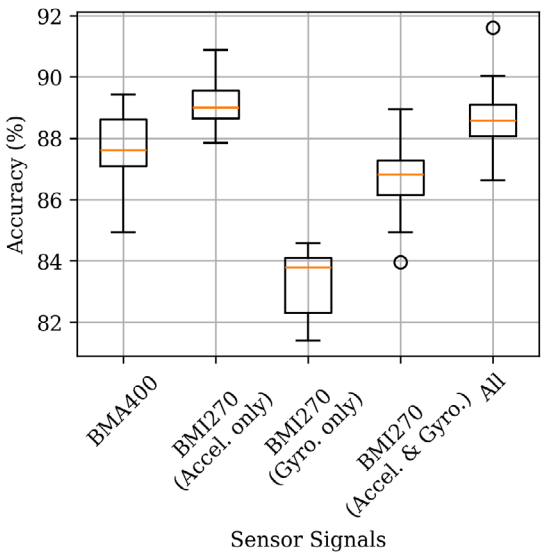


Figure 6. Classification accuracy of the CNN for different types of input signals.

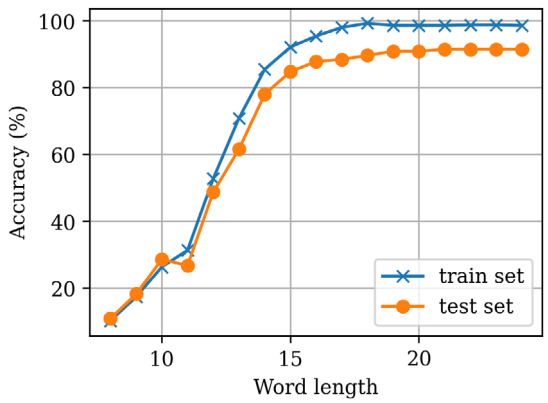


Figure 7. Quantization of proposed classifier to fixed-point arithmetic with word length n.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 5. Performance metrics of different sensor systems.

	Ours	[25]	[26]	[27]	[28]
Platform	22 nm CMOS	FPGA	FPGA	65 nm CMOS	65 nm CMOS
Includes sensor	✓	✗	✗	✗	✗
DNN model	Our CNN	TinyYOLOv3	TinyYOLOv3	ResNet-50	ResNet-50
Task	Vehicle Surroundings perception	Object Detection	Object Detection	Image Classification	Image Classification
Dataset	Our Dataset	MS-COCO	MS-COCO	ImageNet	ImageNet
Input Dimensions	27×1600	$640 \times 480 \times 3$	$640 \times 480 \times 3$	$256 \times 256 \times 3$	$256 \times 256 \times 3$
MAC Ops.	$\mathbf{4.9 \times 10^5}$	5.56×10^{9} [a)]	5.56×10^{9}	7×10^{9} [b)]	7×10^{9}
Performance	**88%** (Accuracy)	30.9 (mAP)	34.01 (mAP)	77.15% [c)] (Top1-Accuracy)	77.15 % (Top1-Accuracy)
Total power [W]	$\mathbf{2.7 \times 10^{-5}}$	3.36	5.52	2.48×10^{-1}	2.47×10^{-1}

[a)]Number of MAC operations for TinyYOLOv3 extracted from ref. [29]. [b)]Number of MAC operations for ResNet-50 extracted from ref. [27]. [c)]Top-1 accuracy from ResNet-50 extracted from ref. [30].

$$P_{\mathrm{dyn,VSP}}(n=16) = \frac{N_{\mathrm{VSP}}}{N_{\mathrm{ref}}} \cdot \frac{E_{\mathrm{MAC},n=16}}{E_{\mathrm{MAC},n=12}} \cdot P_{\mathrm{dyn,ref}}(n=12) \tag{1}$$

In the end, the resulting CNN accelerator consumes 0.622 μW at 0.5 V. In the context of a multi-chip solution, as motivated in Section 3.3, the CNN accelerator and the MEMS sensor are integrated independently as shown in Figure 3. Adding the sensor power from Table 2, the resulting system power consumption is ≈27 μW. When comparing the results with Table 2 it is evident, that the CNN hardware accelerator consumes only a fraction of the total power.

4.6. Performance Comparison

The key performance metrics of our sensor system are summarized in **Table 5**. Our sensor system can be integrated into vehicles as a standalone system for surroundings perception of parked vehicles. The barrier for integration is very low as the system has small dimensions and a power consumption of only 27 μW. To put this into perspective, our sensor system could be operated using a standard coin cell battery such as CR2032 for months (without considering self-discharge[23]). An alternative to MEMS inertial sensors is camera-based systems. However, to the best of our knowledge, there is no comprehensive evaluation of state-of-the-art systems such as Tesla Sentry Mode[3] available in the scientific literature. Therefore, we focus our comparison on individual components such as object detection models and dedicated processors.

Camera-based vehicle surroundings perception can be considered as an object detection task or image classification task. For object detection, YOLO may be used as a state-of-the-art model capable of achieving competitive results in the COCO benchmark.[24] Using dedicated hardware accelerators based on field-programmable gate arrays (FPGA), a performance of up to 34.01 mAP can be achieved.[25,26] This requires 5.56×10^9 MAC operations per image at a power consumption of up to 5.52 W, see Table 5. Note that the values include only the object detection processor without camera sensors, which require additional power.

For image classification on the ImageNet dataset, a ResNet-50 model architecture is considered. With a dedicated hardware accelerator based on 65 nm CMOS technology, this model achieves a top-1 classification accuracy of 77.15% which requires 7×10^9 MAC operations per image at a power consumption of 0.25 W (without considering camera sensors). We note that the classification accuracy is evaluated on a different dataset and therefore cannot be compared. However, the required resources can be compared. It is evident, that camera-based vehicle surroundings perception requires several order of magnitude more resources compared to MEMS inertial sensors.

In summary, we find that camera-based systems could be used for vehicle surroundings perception. Since, to our knowledge, such applications have not been published in the literature, it is not clear how accurate camera-based solutions would be. However, the power consumption of camera-based solutions is significantly larger compared to our proposed sensor system.

5. Conclusion

In this article, we investigated surroundings perception of parked vehicles using MEMS inertial sensors. For event detection, we used a CNN. We found that events can be detected with an accuracy of 88%. This demonstrates that vehicle surroundings perception is generally possible using MEMS inertial sensors. However, the accuracy varied significantly across different events. While some events could be detected with very high accuracy, others could not be detected very well. Regarding the sensor location, we found that a sensor placement at the rear of the vehicle is ideal. Furthermore, we found that accelerometers generally perform better at surroundings perception than gyroscopes.

The CNN model was quantized for deployment on a hardware accelerator. Our smart MEMS sensor consisting of accelerometer and hardware accelerator has a total system power consumption of 27 μW. Therefore, it can be operated for long periods of time without depleting the car battery.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

Acknowledgements

The first author would like to thank Dr. Roman Föll (Robert Bosch GmbH) for helpful discussions.

Open Access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in Github - Bosch Research at https://github.com/boschresearch/mems-vehicle-surroundings-perception.

Keywords

ASIC, CNN, inertial sensor, machine learning, MEMS, surroundings perception

Received: October 20, 2023
Revised: March 14, 2024
Published online: April 10, 2024

[1] E. Marti, M. A. de Miguel, F. Garcia, J. Perez, *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 94.
[2] S. Sun, A. P. Petropulu, H. V. Poor, *IEEE Signal Process Mag.* **2020**, *37*, 98.
[3] Tesla, https://www.tesla.com/ownersmanual/model3/en_us/GUID-56703182-8191-4DAE-AF07-2FDC0EB64663.html (accessed: October 2023).
[4] M. Harris, *IEEE Spectr.* **2022**, *59*, 40.
[5] W. A. Smith, R. B. Randall, *Mech. Syst. Signal Process.* **2015**, *64*, 100.
[6] D. Neupane, J. Seok, *IEEE Access* **2020**, *8*, 93155.
[7] A. Reiss, D. Stricker, in *2012 16th Int. Symp. on Wearable Computers*, IEEE, Newcastle, UK, June **2012**, pp. 108–109.
[8] A. Ignatov, *Appl. Soft Comput.* **2018**, *62*, 915.
[9] J. R. Kwapisz, G. M. Weiss, S. A. Moore, *ACM SIGKDD Explor. Newsl.* **2011**, *12*, 74.
[10] W. Teufl, M. Lorenz, M. Miezal, B. Taetz, M. Fröhlich, G. Bleser, *Sensors* **2018**, *19*, 38.
[11] A. S. Alharthi, S. U. Yunas, K. B. Ozanyan, *IEEE Sens. J.* **2019**, *19*, 9575.
[12] Y.-J. Cha, W. Choi, O. Büyüköztürk, *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361.
[13] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731.
[14] M. Xia, T. Li, L. Xu, L. Liu, C. W. De Silva, *IEEE/ASME Trans. Mechatron.* **2017**, *23*, 101.
[15] D. Droste, H. Symanzik, T. Gießelmann, M. Ulm, I. Galdi, R. Campagna, in *IEEE 48th European Solid State Circuits Conf. (ESSCIRC)*, Milan, Italy, September **2022**, pp. 12–16.
[16] S. S. Saha, S. S. Sandha, M. Srivastava, *IEEE Sens. J.* **2022**, *22*, 21362.
[17] R. Immonen, T. Hämäläinen, *J. Sens.* **2022**, *2022*, 7437023.
[18] J. Loh, T. Gemmeke, in *2022 IEEE Nordic Circuits and Systems Conf. (NorCAS)*, Oslo, Norway, October **2022**.
[19] D. P. Kingma, J. Ba (Preprint), *arXiv:1412.6980*, v1, submitted: Dec. **2014**.
[20] D. Berrar, *Encyclopedia of Bioinformatics and Computational Biology*, Academic Press, Oxford **2019**, pp. 542–545, ISBN 978-0-12-811432-2.
[21] A. C. Fischer, F. Forsberg, M. Lapisa, S. J. Bleiker, G. Stemme, N. Roxhed, F. Niklaus, *Microsyst. Nanoeng.* **2015**, *1*.
[22] T. Stadtmann, C. Latotzke, T. Gemmeke, in *2020 19th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, IEEE, Miami, FL, December **2020**.
[23] D. Griffith, *IEEE Solid-State Circuits Mag.* **2022**, *14*, 51.
[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, in *Computer Vision – ECCV 2014*, Springer, Cham **2014**, pp. 740–755.
[25] Z. Yu, C.-S. Bouganis, *Applied Reconfigurable Computing. Architectures, Tools, and Applications*, Springer, Cham **2020**, pp. 330–344, ISBN 9783030445348.
[26] S. Ki, J. Park, H. Kim, *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 3882.
[27] A. Ardakani, C. Condo, W. J. Gross, *IEEE Trans. Comput.* **2020**, *69*, 138.
[28] M. Ahmadi, S. Vakili, J. M. P. Langlois, *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 3184.
[29] R. Huang, J. Pedoeem, C. Chen, in *2018 IEEE Int. Conf. on Big Data (Big Data)*, IEEE, Seattle, WA, December **2018**.
[30] K. He, X. Zhang, S. Ren, J. Sun, in *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, June **2016**, pp. 770–778.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH

26404567, 2024, 5, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300679 by Rwth Aachen Hochschulbibliothek, Wiley Online Library on [17/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License