IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2023, accepted 7 March 2023, date of publication 16 March 2023, date of current version 24 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3258400

RESEARCH ARTICLE

Accelerating Deep Neural Networks for Efficient Scene Understanding in Multi-Modal Automotive Applications

STAVROS NOUSIAS, (Member, IEEE), ERION-VASILIS PIKOULIS, CHRISTOS MAVROKEFALIDIS, (Member, IEEE), AND ARIS S. LALOS
Industrial Systems Institute, Athena Research Center, 26504 Patras, Greece

Corresponding author: Stavros Nousias (nousias@isi.gr)

This work was supported in part by the H2020 Project "CPSoSaware–Cross-Layer Cognitive Optimization Tools and Methods for the Lifecycle Support of Dependable CPSoS" under Agreement 871738, and in part by the Horizon Europe Project "TRUSTEE–Trust and Privacy Preserving Computing Platform for Cross-Border Federation of Data" under Agreement 101070214. The publication of the article in OA mode was financially supported by HEAL-Link.

ABSTRACT Environment perception constitutes one of the most critical operations performed by semi- and fully- autonomous vehicles. In recent years, Deep Neural Networks (DNNs) have become the standard tool for perception solutions owing to their impressive capabilities in analyzing and modelling complex and dynamic scenes, from (often multi-modal) sensory inputs. However, the well-established performance of DNNs comes at the cost of increased time and storage complexity, which may become problematic in automotive perception systems due to the requirement for a short prediction horizon (as in many cases inference must be performed in real-time) and the limited computational, storage, and energy resources of mobile systems. A common way of addressing this problem is to transform the original large pre-trained networks into new smaller models, by utilizing Model Compression and Acceleration (MCA) techniques, improving both their storage and execution efficiency. Within the MCA framework, in this paper, we investigate the application of two state-of-the-art weight-sharing MCA techniques, namely a Vector Quantization (VQ) and a Dictionary Learning (DL) one, as well as two novel extensions, towards the acceleration and compression of widely used DNNs for 2D and 3D object-detection in automotive applications. Apart from the individual (uni-modal) networks, we also present and evaluate a multi-modal late-fusion algorithm for combining the detection results of the 2D and 3D detectors. Our evaluation studies are carried out on the KITTI Dataset. The obtained results lend themselves to a twofold interpretation. On the one hand, they showcase the significant acceleration and compression gains that can be achieved via the application of weight sharing on the selected DNN detectors, with limited accuracy loss, as well as highlight the performance differences between the two utilized weight-sharing approaches. On the other, they demonstrate the substantial boost in detection performance obtained by combining the outcome of the two unimodal individual detectors, using the proposed late-fusion-based multi-modal approach. Indeed, as our experiments have shown, pairing the high-performance DL-based MCA technique with the loss-mitigating effect of the multi-modal fusion approach, leads to highly accelerated models (up to approximately 2.5× and 6× for the 2D and 3D detectors, respectively) with the performance loss of the fused results ranging in most cases within single-digits figures (as low as around 1% for the class "cars").

INDEX TERMS Model compression and acceleration, multi-modal fusion, object detection, scene analysis, scene understanding, experimental evaluation.

I. INTRODUCTION

Autonomous vehicles (AV) are an integral part of the continuously evolving field of Intelligent Transportation

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.

Systems (ITS) [1] and introduce a variety of technical challenges intertwined with the levels of driving automation, as defined for example by the Society of Automobile Engineers (SAE) [47], ranging from ''no driving automation'' (level 0) to ''full driving automation'' (level 5). Of particular interest are the levels 3 (conditional driving automation in which the system is capable of taking over control for a specific amount of time and/or in specific situations, but the driver must permanently monitor the system and be prepared to take over at any time), and 4 (high driving automation in which the driver need not monitor the system while it is active for specific conditions). Levels 3 and 4 represent the limits of what is possible with today's technology and what is the envisioned next step toward full automation, respectively.

The functionality of an AV system can be represented by three layers that incorporate the tasks of sensing, perception, and decision-making [2]. The first layer, i.e., sensing, includes various sensing devices such as short and long-range radars, visual and/or thermal cameras, and ultrasonic, Light Detection And Ranging (LiDAR), and Global Position System (GPS) sensors [2], which gather relevant data from the environment surrounding the AV. The perception layer utilizes the collected data and extracts information from the scene about, e.g., other traffic objects, obstacles, etc. This information is the basis for reaching decisions in the third and final layer for advanced vehicle control and path planning, to name a few.

As the level of autonomy increases (especially for levels 3 and above), the ability to perceive dynamic and complex scenes from sensory data constitutes one of the most critical functionalities performed by AVs (along with sensing and decision-making) and is a key enabler for the AVs' reliable and safe operation [3], [4], [5], [6]. This is, in turn, translated into an increasing degree of sophistication regarding not only the employed sensors but also their utilization via more advanced processing and fusion techniques. For example, low-cost and low-accuracy ultrasonic sensors are sufficient in low levels of automation, e.g., for parking assistance, and they are used for many years in the automotive industry. At the next level, radars and cameras are increasingly being incorporated in modern cars, e.g., as part of adaptive cruise control systems.

The sophistication required to achieve the desired performance becomes readily apparent if we consider that even a 30 *cm* deviation in the lateral position of a vehicle can lead to dangerous maneuver initiations. Note that in challenging environments, such as urban and dense areas, tunnels, etc., the localization error of modern GPS sensors is orders of magnitude higher than this level [7]. Moreover, an increasing factor to the difficulty of the problem is the fact that the prediction window of active perception systems is very short since the AV should be able to timely adapt to abrupt changes in the surrounding environment (in a fraction of a second [8]), such as the ''random'' motion style of vulnerable users including pedestrians and cyclists. Additionally, the complexity of the surrounding environment requires the use of multiple sensing modalities (including cameras and LiDAR) for increased effectiveness [9]. These arguments simply highlight the fact that, in the context of driving automation, scene understanding solutions (comprising image classification, object detection and tracking, and semantic segmentation, to name a few), must be accurate, fast, and efficient.

With this goal in mind, this paper presents a study concerning the application of MCA on high-performance DNN models used for scene understanding in automotive scenarios. To this end, we focus on state-of-the-art weight-sharing techniques and propose two novel extensions that build upon the concepts of ''global sparsity'' and ''subspace grouping''. These are accompanied by a detailed analysis of the acceleration and compression gains that can be achieved as well as representative simulations. Then, the techniques are applied on modern 2D (i.e., image-based) and 3D (i.e., point-could based) detectors that employ DNNs as well as on a multi-modal detector by describing and adopting a simple late-fusion strategy that combines the outputs of the 2D and 3D detectors. The impact of MCA on the performance of the uni-modal and the multi-modal architectures is evaluated in the well-known KITTI dataset. To the best of our knowledge, this is the first work that demonstrates the positive effects of multi-modal fusion not only on enhancing the performance of deep models for object detection but also on further mitigating the impact on the performance loss incurred by the application of MCA techniques.

In the following sections, we first provide the positioning of the paper through the description of the relevant bibliography and its contribution. Then, the employed model compression and acceleration techniques along with the proposed extensions and analysis, as well as the adopted late multi-modal fusion strategy, are described. Afterwards, a thorough experimental evaluation of the MCA impact on the behaviour of the adopted models for 2D and 3D object detection as well as their late fusion version is presented in automotive scenarios. Finally, we conclude the paper by summarizing and discussing the results of the presented analysis.

Notation: A matrix, a vector, and a scalar are denoted as $\mathbf{X}$, $\mathbf{x}$, and x, respectively. The transpose of a matrix $\mathbf{X}$ is denoted as $\mathbf{X}^{\mathrm{T}}$. The matrix with zero elements is denoted as $\mathbf{0}$ and its size can be inferred by its context. Moreover, $X \in \mathbb{R}^{A\times B}$ denotes the $\mathbf{X}$ matrix of size $A \times B$ with real entries. The operator vec($\mathbf{X}$) stacks the columns of $\mathbf{X}$ into a column vector, while $\cup$ denotes the union operation between two sets. Finally, $\|\cdot\|_2$ and $\|\cdot\|_0$ are the Euclidean norm and the l_0 (pseudo) norm, respectively, while $\otimes$ denotes the Kronecker product.

II. RELEVANT BIBLIOGRAPHY AND CONTRIBUTION

In this section, we present a brief bibliographical survey of the main research areas that this paper builds upon, namely MCA, and object detection based on 2D visual images, 3D point clouds as well as the fusion of the two modalities.

To this end, both state-of-the-art object detection models and the utilization of MCA techniques for their efficient implementation, are presented. Afterwards, the motivation and the main contributions of the paper are outlined.

A. DNN-BASED OBJECT DETECTION IN AUTOMOTIVE APPLICATIONS

Object detection constitutes a fundamental operation of perception systems in autonomous vehicles. In recent years, DNN models have contributed significantly to the improvement of object detection performance, concerning both 2D (image-based based captured by visual cameras) and 3D (i.e., point-cloud based captured by LiDAR) detectors [10]. Works concerning DNN-based 2D detectors can be broadly categorized into two-stage and one-stage approaches [11]. Detectors following the two-stage approach first generate region proposals on the input image and then assess each region regarding the presence of one or multiple objects and the class each of them belongs to. On the other hand, single-stage object detectors produce directly both the location and the class of each object in the input image. Although two-stage detectors usually perform better [11], single-stage detectors such as the Single Shot MultiBox Detector (SSD) [12], SqueezeDet [13], the You Only Look Once (YOLO)v2 detector [14] and EfficientDet [15], are generally preferred for autonomous driving applications, due to their lower computational and storage requirements. The classes of interest here are typically vehicles, cyclists, and pedestrians.

Similar to the 2D case, DNNs have also been ubiquitously employed for point cloud-based detection with the proposed models following mainly two directions. In the first one, called grid-based, the irregular point clouds are initially transformed into a regular representation that can be processed by ordinary convolutional layers, while, in the second direction, called point-based, the models operate directly on the points of the cloud. In general terms, the performance of grid-based models depends heavily on the resolution of the underlying grid, while point-based detectors are computationally more demanding than their grid-based counterparts. Some of the first DNN-based 3D detectors, such as VoxelNet [16], utilized 3D convolutions, with Sparsely Embedded Convolutional Detection (SECOND) [17] introducing sparse 3D convolutions to reduce complexity. On the other hand, PointPillars [18] introduced the notion of pillars and employed only 2D convolutions, thus, being able to achieve both high precision and fast inference times. Other high-performing DNN models for 3D object detection are PointRCNN [19], its extension Part-A^2 Net [20], and Point-Voxel-RCNN (PV-RCNN) [21], which is one of the first detectors to exploit both grid-based and point-based approaches.

Finally, there is an active direction for object detection that involves the fusion of information originating from different modalities, with the most common one being the fusion of 2D (visual images) and 3D (point clouds) related information [22]. Focusing on the time when fusion takes place, three approaches can be followed; early, late, and middle fusion. In the first case, information from the two modalities is combined at an early stage where the data are actually generated, while, in the second one, fusion is performed at the stage of decisions. The latter case, i.e., middle, refers to fusing information in any intermediate stage of the overall DNN-based multi-modal object detection. Currently, there is no consensus on which approach is the best choice as all have pros and cons associated with their adoption [22]. The "late" fusion approach, which is of particular interest in this paper, on the one hand, is simple and flexible (as any changes in processing a sensing modality, do not require re-training of the whole multi-modal detector) but, on the other hand, has a high computational cost and memory requirements. Thus, such approaches may benefit considerably from the application of MCA techniques.

B. MODEL COMPRESSION AND ACCELERATION IN AUTOMOTIVE APPLICATIONS

DNNs [23] have been employed in numerous application domains in the last several years, including autonomous driving which is the main theme of this paper. However, the high performance of DNNs is typically related to analogously high requirements regarding computational and storage resources. This becomes problematic in automotive applications due to the necessity of very fast inference times on the one hand, and the limited computational, storage, and energy resources of mobile systems, on the other [24].

Regarding DNN-based object detection, research activities have focused mainly on designing and utilizing compact DNN models such as SqueezeDet [13] and Mini-YOLOv3 [25], aiming at their efficient implementation on embedded devices [26]. Towards this goal, the incorporation of MCA techniques for transforming pre-trained, highly-performing (yet resource-demanding) DNN models, into lighter versions while mitigating the impact on the achieved performance, is also gaining popularity. This is especially true for the case of image-based detectors. The authors in [27] employ a combination of pruning criteria for removing up to 90% of parameters of YOLOv3, reporting virtually no loss in performance. In [28], detectors based on binary-weight neural networks (whereby parameters are quantized to just 1 bit) are proposed, utilizing a knowledge-transfer method to aid their training, using a full-precision teacher network. In [29], an efficient version of the YOLOv3 detector is obtained via a comprehensive pruning scheme including layer-level and channel-wise pruning, while light-weight image-based detectors are also proposed in [30], via a combination of knowledge transfer and pruning strategies. Finally, [31] utilized a Dictionary Learning-based vector quantization technique, for the acceleration of SqueezeDet and ResNetDet (both proposed in [13]) by roughly 60% and 70%, respectively, with negligible accuracy loss.

On the other hand, the literature concerning the utilization of MCA techniques in 3D detectors is currently limited. The work in [32] proposes a multi-task 3D detector and resorts on pruning of unimportant parameters for a 2× speed-up of the model inference time. Furthermore, [33] utilizes vector quantization on two state of the art LiDAR-based 3D detectors, achieving acceleration rates of over 5× with a negligible loss for the "car" and "cyclist" classes, and acceptable loss for the "pedestrian" one.

Finally, to the best of our knowledge, there are no works that discuss the application of MCA on multi-modal object detection DNN-based models.

C. MOTIVATION AND CONTRIBUTION

So far, the use of MCA techniques on automotive object detection is limited mainly to the application of simple pruning and/or scalar quantization techniques on single-modality DNN detectors with the majority of works employing visual images (see Table 1 for an overview of the existing literature). This paper aims to move a step forward by introducing more elaborate weight-sharing MCA approaches that have been shown to outperform other rivals in the related literature, on multi-modal object detection in the automotive domain.

To this end, firstly, we focus on the state-of-the-art VQ [34] and DL [31] based techniques that rely on the design of codebooks with a preset structure (in terms of their size, number of utilized codewords, etc.). By observing that such a structure limits their flexibility and adaptability on the problem at hand for achieving better acceleration and/or compression ratios, two novel extensions are proposed adding flexibility regarding the inherent trade-offs between compression (memory footprint) and acceleration (computational power) during the system design phase. Secondly, we study for the first time the impact of the considered MCA techniques on the performance of multi-modal DNN-based object detection by introducing a simple, yet effective, late-fusion method.

In more detail, the contributions of the paper are summarized in the following points:

- Two new concepts are introduced, namely, (a) global sparsity that allows the underlying optimization procedure for MCA to partially determine the structure of the codebooks and (b) subspace grouping that allows sharing not only at the level of codewords but also at the level of codebooks. In both cases, the trade-off between performance and acceleration / compression ratio is better addressed.
- A late-fusion approach based on the non-maximal suppression of the individual modalities of the detectors is presented and evaluated. As it is demonstrated by our experiments, the resulting multi-modal detector offers a substantial performance improvement over the individual uni-modal systems, both in their original and in their accelerated forms.
- A thorough investigation related to the acceleration and compression of 2D (image) and 3D (point-cloud) convolutional object detectors (SqueezeDet [13] and PointPillars [18], respectively), towards their efficient deployment as core parts of the perception systems in vehicular perception systems, is presented.
- Image-based high-performance DL-based MCA technique with the loss-mitigating effect of the multi-modal fusion approach leads to highly accelerated models (up to approximately 2.5× and 6× for the 2D and 3D detectors, respectively) with the performance loss of the fused results ranging in most cases within single-digits figures (as low as around 1% for the class "cars"). The KITTI dataset [35] was used for evaluation purposes in our experiments.

III. WEIGHT SHARING VIA PRODUCT QUANTIZATION

Viewing the convolution operation as a series of dot-products between input and kernel vectors in an N-dimensional space (with N being the number of input/kernel channels), product quantization aims at reducing the number of required operations by splitting the initial space into S, N'-dimensional subspaces (where $N' = N/S$), and limiting the number of allowed representations in each of them. To be more specific, the number of representations in each subspace is reduced via VQ, namely, by approximating the original kernel sub-vectors using a small set of representatives called codewords (and their collection, a codebook). In doing so, product quantization approximates the original convolution using only dot-products between input and codewords, instead of the originals.

Conventionally, VQ is treated as a clustering problem solved via the popular k-means algorithm [34], however, a recently proposed technique treating the problem from a Dictionary Learning perspective, has been shown to achieve up to 2× acceleration gain over conventional approach [31].

Assuming there are M 3D kernel volumes in the convolution layer, with 2D filters of size $p \times p$, the conventional and the DL-based approximation schemes (referred to simply as VQ, and DL, respectively, hereafter) can be expressed as follows:

$$\text{VQ}: \ \mathbf{W} \approx \mathbf{C}\Gamma, \qquad \text{DL}: \ \mathbf{W} \approx \mathbf{D}\Lambda\Gamma, \tag{1}$$

where the columns of $\mathbf{W} \in \mathbb{R}^{N' \times p^2 M}$ and $\Gamma \in \mathbb{R}^{K_{vq} \times p^2 M}$ contain the sub-vectors of all kernel volumes (of a particular subspace) and assignment vectors, respectively. Matrix $\mathbf{C} \in \mathbb{R}^{N' \times K_{vq}}$ denotes the representatives (or cluster centroids) in the VQ approximation whereas $\mathbf{D} \in \mathbb{R}^{N' \times L_{dl}}$ and $\Lambda \in \mathbb{R}^{L_{dl} \times K_{dl}}$ denote the dictionary and the matrix of sparse coefficients, respectively, for the DL approximation.

A. A NEW GLOBAL-SPARSITY CONSTRAINT

In this paper, we explore a novel approach by imposing the sparsity constraint on Λ, adding flexibility to the mechanism followed in [31], whereby sparsity was imposed by restricting the number of non-zero elements in each column of Λ to a pre-selected sparsity level value ρ, thus, every codeword

TABLE 1. Overview of bibliography relevant to model compression and acceleration on automotive scene understanding.

Paper	ultirow210em**Year**	**Sensing modalities**			**MCA method**
		state-of-the-artDAR (3D)	Fusion		
Krittayanawach et al. [27]	2019	✓	–	–	Pruning
Xu et al. [28]	2019	✓	–	–	Scalar quantization, Distillation
Wang et al. [29]	2020	✓	–	–	Pruning
Tsai et al. [30]	2020	✓	–	–	Pruning
Zhou et al. [32]	2021	–	✓	–	Pruning
Nousias et al. [33]	2021	–	✓	–	Vector quantization
Pikoulis et al. [31]	2022	✓	–	–	Vector quantization
This work	2023	✓	✓	✓	Vector quantization, two extensions

contained in codebook $\mathbf{D}\Lambda$ is a linear combination of ρ atoms from $\mathbf{D}$.

Instead, we propose restricting the total number of its non-zero elements, regardless of their locations. Denoting this number as P, in this case, the codewords are constructed as a linear combination of ρ_i atoms, $i = 1, \ldots, K_{dl}$, with $\sum_{i=1}^{K_{dl}} \rho_i = P$, hence the increased flexibility. To avoid confusion, we will refer to this new approach as DL-GS (namely, Global Sparsity), and the original approach presented in [31], as DL-LS (namely, Local Sparsity).

To solve the sparse coding problem (i.e. the optimization concerning Λ) stemming from (1), under the DL-GS approach, we first rewrite the cost function as follows:

$$\mathcal{E} = ||\mathrm{vec}(\mathbf{W}) - (\Gamma^{\mathrm{T}} \otimes \mathbf{D})\mathrm{vec}(\Lambda)||_2, \tag{2}$$

where the identity $\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^{\mathrm{T}} \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$ has been employed [36]. Using (2) as the cost function, the minimization problem for the sparse coding step can be written as

$$\min_{\zeta} ||\mathrm{vec}(\mathbf{W}) - (\Gamma^{\mathrm{T}} \otimes \mathbf{D})\zeta||_2 \qquad \text{s.t. } ||\zeta||_0 = P, \tag{3}$$

and can be solved via the classical Orthogonal Matching Pursuit (OMP) algorithm [37].

The flexibility that is introduced to the problem at hand via the global sparsity constraint leads to measurable improvement of DL the technique regarding the quantization error (resulting in analogous acceleration gains), as our experiments show. However, perhaps of even more importance is the fact that contrary to the local sparsity constraint, the solution attained via global sparsity has the inherent ability to reduce the size of the used codebook by setting entire columns of Λ equal to $\mathbf{0}$. Thus, this variant of the DL technique can support hybrid MCA approaches combining weight sharing with (indirect) pruning. It is also noted that a solution involving group sparsity constraints (with the groups being Λ's columns) might be even more beneficial towards this end, although such a direction was not pursued here.

B. COMPUTATIONAL and STORAGE COMPLEXITY

We denote as $\mathcal{T}_o$, $\mathcal{T}_{vq}$, and $\mathcal{T}_{dl}$, the computational complexities (in terms of Multiply and Accumulate (MAC) operations) of the original and the approximate versions of a convolutional layer. Using the VQ and DL weight-sharing methods, respectively we can show that the following equations apply [31], [34]

$$\mathcal{T}_o = m^2 p^2 MN \tag{4}$$

$$\mathcal{T}_{vq} = m^2 NK_{vq} \tag{5}$$

$$\mathcal{T}_{dl} = m^2 (NL_{dl} + \rho SK_{dl}) \tag{6}$$

Moreover, the acceleration ratio achieved by the two weight-sharing approaches is defined as the ratio of the original over the accelerated complexities,

$$\alpha_{vq} \equiv \mathcal{T}_o / \mathcal{T}_{vq} \tag{7}$$

$$\alpha_{dl} \equiv \mathcal{T}_o / \mathcal{T}_{dl} \tag{8}$$

Finally, it can be easily shown that the VQ and DL-based approximations yield the same acceleration ratio when the employed parameters satisfy the following equality:

$$L_{dl} = K_{vq} \left(1 - \frac{\rho\, c}{N'}\right), \tag{9}$$

where $c > 1$ is a coefficient linking the sizes of the DL-based and the VQ-based codebooks, i.e. $K_{dl} = c\, K_{vq}$ holds.

On the other hand, regarding the storage complexity, in the case of the original layer, there are $p^2 MN$ kernel weights to be stored (omitting the negligible storage requirements of the bias weights for simplicity). Hence the storage complexity of the original layer is obtained simply as $\mathcal{S}_o = p^2 MN \times b_{\mathrm{float}}$, where b_{float} denotes the number of bits used by the system for floating point representation.

Adopting the weight-sharing approach, the original weights are partitioned into S subspaces, with each subspace being represented by a codebook consisting of real numbers and a set of indices pointing to the codewords in the codebook.

More specifically, in the case of VQ, the codebook for each subspace consists of K_{vq} codewords of length N', i.e., $N'K_{vq}$ real numbers in total. Additionally, there are $p^2 M$ indices with each index taking values in $\{1, 2, \ldots, K_{vq}\}$, indicating the codeword used to represent the corresponding original subvector. Thus, the (layer) storage complexity for the VQ technique can be expressed as:

$$\mathcal{S}_{vq} = \underbrace{NK_{vq} \times b_{\mathrm{float}}}_{\mathbf{C}} + \underbrace{p^2 MS \times \lceil \log_2(K_{vq}) \rceil}_{\Gamma} \tag{10}$$

On the other hand, in the DL case, the codebook is further decomposed as the matrix-product $\mathbf{D}\Lambda$, with the dictionary $\mathbf{D}$

consisting of L_{dl} atoms of length N', for a total of $N'L_{dl}$ real numbers, while Λ consists of K_{dl} sparse columns of length L_{dl}, each containing ρ non-zero real coefficients. Thus, the storage complexity in the case of the DL technique is obtained as follows:

$$\mathcal{S}_{dl} = \underbrace{NL_{dl} \times b_{\text{float}}}_{\mathbf{D}} + \underbrace{\rho K_{dl} S \times b_{\Lambda}}_{\Lambda} + \underbrace{p^2 MS \times b_{\Gamma}}_{\Gamma}, \tag{11}$$

where $b_{\Lambda} = \left(\lceil \log_2(L_{dl}) \rceil + b_{\text{float}}\right)$, $b_{\Gamma} = \lceil \log_2(K_{dl}) \rceil$. Finally, similarly to the acceleration ratios, the compression ratios achieved by the VQ and DL techniques, are defined as the ratio of the storage requirement of the original versus the approximate layer, i.e $\tau_{vq} = \mathcal{S}_o/\mathcal{S}_{vq}$, $\tau_{dl} = \mathcal{S}_o/\mathcal{S}_{dl}$, respectively.

1) SUBSPACE GROUPING

A direction that could be pursued to boost the achieved compression ratio is that of subspace grouping, whereby each codebook is designed to represent a group of subspaces instead of a single one. To be more specific, the main idea is to group all the subvectors falling into the selected group of subspaces, and estimate the codebook that best represents them jointly, utilizing the approximations defined in (1), for the VQ and DL approaches, respectively. From a technical standpoint, this simply means that matrix **W** holds the subvectors of a number of subspaces, instead of a single one.

Understandably, using the same codebook to represent more than one subspace, reduces the total number of codebooks that needs to be stored. This is reflected in the contribution of **C** in (10), and that of **D**, Λ in (11), respectively, whose storage complexities take now the form

$$N'N_g K_{vq} \times b_{\text{float}} \tag{12}$$

and

$$N'N_g L_{dl} \times b_{\text{float}} \tag{13}$$

$$\rho K_{dl} N_g \times b_{\Lambda} \tag{14}$$

respectively, with $N_g \in \{1, 2, \ldots, S\}$ denoting the number of used subspace groups. Note that $N_g = S$ corresponds to no grouping (i.e. each group consists of a single subspace), while for $N_g = 1$, all subspaces are represented by a single codebook. Note finally that subspace grouping does not alter the achieved acceleration ratio since the latter depends only on the size and structure of the used codebooks, not their total number.

C. DISCUSSION ON IMPLEMENTATION ISSUES

MCA is treated here from an algorithmic point of view as it is the case with the vast majority of relevant works in the field, many of whom are referenced here (e.g., [10]). Indeed, following the main body of the MCA-related bibliography, the acceleration gains are reported here in percentage/rate of parameters or operations reduced and not in actual execution time speed-up.

The main reason behind this lies in the fact that many of the proposed MCA techniques alter the conventional flow of operations in deep architectures and are intended for specialized implementations in embedded devices with limited resources. This is especially true for elaborate techniques such as the VQ and DL weight-sharing approaches adopted in this work (as opposed, e.g., to pruning strategies that simply reduce the dimensions of the network or scalar quantization that reduces the number of bits in the arithmetic representation of the parameters).

Given the fact that a specialized implementation (e.g., in hardware) of the networks is out of the scope of the paper and keeping in mind that existing tools (such as PyTorch [38], TensorFlow [39], etc.) do not support vector quantization natively, in our experiments, we emulated the effect of weight-sharing with the goal of assessing the performance of the “quantized” network (regarding detection accuracy) and demonstrating the potential of the employed MCA technique. The methodology (for the emulation) which entails substituting parameter sub-vectors with the corresponding codewords (thus, leaving the number of parameters and the overall architecture of the “quantized” networks unaltered), is commonly adopted in current MCA literature concerning weight sharing techniques.

IV. LATE MULTI-MODAL FUSION STRATEGY

A simple late fusion strategy is proposed for processing the detection outcomes of the two considered modalities, namely, 2D visual images and 3D point clouds. An illustration of the strategy is presented in Figure 1. The concept behind the fusion approach is to select the outcome of the detector with the highest confidence score, i.e., either the 2D detection based on SqueezeDet (lower branch in Figure 1) or the 3D detection based on PointPillars (upper branch in Figure 1). To this end, the well known Non-Maximal Suppression (NMS) algorithm [40] is employed to process the detection outcomes of the two branches. Note that for the 3D detection branch, initially, the 3D bounding boxes are projected on the 2D image and these projections are assigned the confidence score of the 3D detector. Thus, NMS receives a 2D image that contains bounding boxes from both modalities before fusing them.

Let us now focus briefly on the used projection mechanism. To this end, let **P** and **R** denote the camera intrinsic and transformation matrices, respectively. Let us, also, denote a 3D point and its projection to the 2D plane as $\mathbf{x}_{\text{3D}} = [X, Y, Z, 1]^T$ and $\mathbf{x}_{\text{2D}} = [x, y, 1]^T$, respectively. Then, $\mathbf{x}_{\text{2D}}$ is obtained from $\mathbf{x}_{\text{3D}}$ as follows

$$\mathbf{x}_{\text{2D}} = \mathbf{P}\mathbf{R}\mathbf{x}_{\text{3D}}. \tag{15}$$

Considering a 3D bounding box $\mathcal{B}_{\text{3D}}$ as a set of 8 points $\mathbf{x}^i_{\text{3D}}$, $i = 1, 2, \ldots, 8$, then, its projection to the 2D plane, namely, $\mathcal{B}_{\text{3D}_{proj}}$, is obtained by, first, projecting all $\mathbf{x}^i_{\text{3D}}$'s using (15) and, then, computing the corresponding axis-aligned bounding box. Moreover, let $\mathcal{P}_{\text{2D}}$ and $\mathcal{P}_{\text{3D}_{proj}}$ denote the sets of predicted bounded boxes from the 2D and 3D

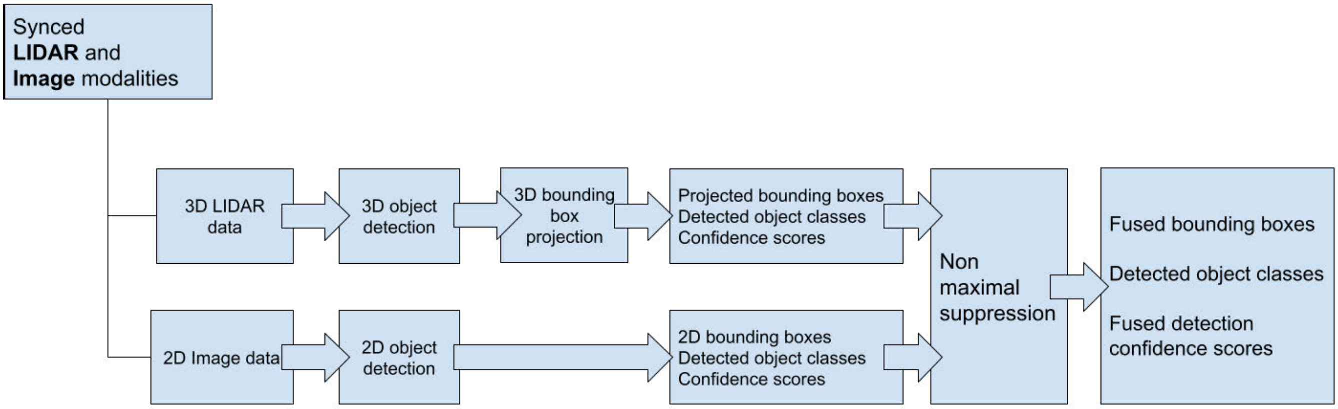


FIGURE 1. Architecture of late multi-modal fusion approach.

(after projection) object detectors, respectively, and let $\mathcal{S}_{2\mathrm{D}}$ and $\mathcal{S}_{3\mathrm{D}_{proj}}$ denote the sets of the corresponding confidence scores. Then, the outcome of the late fusion mechanism is obtained as

$$\mathcal{P}' = \mathrm{NMS}\left(\mathcal{P}, \mathcal{S}, \lambda_{\mathrm{NMS}}\right), \tag{16}$$

where $\mathcal{P} = \mathcal{P}_{2\mathrm{D}} \cup \mathcal{P}_{3\mathrm{D}_{proj}}$, $\mathcal{S} = \mathcal{S}_{2\mathrm{D}} \cup \mathcal{S}_{3\mathrm{D}_{proj}}$, while λ_{NMS} is the Intersection Over Union (IOU) threshold that defines the selection of bounding boxes [40]. For the sake of completeness, the NMS algorithm is presented in Algorithm 1.

Algorithm 1 Non-Maximal Suppression Algorithm

```
procedure NMS($\mathcal{B}$, $c$, $\lambda_{NMS}$)    ▷ Bounding boxes $\mathcal{B}$, scores $c$ and threshold $\lambda_{NMS}$
    $\mathcal{B}_{NMS} \leftarrow \emptyset$
    for $\mathbf{b}_i \in \mathcal{B}$ do    ▷ Iterate boxes
        $d \leftarrow False$
        for $\mathbf{b}_j \in \mathcal{B}$ do
            if $same(\mathbf{b}_i, \mathbf{b}_j) > \lambda_{NMS}$ then
                if $score(c, \mathbf{b}_j) > score(c, \mathbf{b}_i)$ then
                    $d \leftarrow True$
                end if
            end if
        end for
        if not $d$ then
            $\mathcal{B}_{NMS} \leftarrow \mathcal{B}_{NMS} \cup \{\mathbf{b}_i\}$
        end if
    end for
    return $\mathcal{B}_{NMS}$    ▷ Return NMS bounding boxes
end procedure
```

V. EXPERIMENTAL EVALUATION

A performance evaluation of the employed acceleration/compression techniques, using state-of-the-art convolutional DNNs, is presented in this section, along with the effect of multi-modal fusion on the DNNs before and after the application of MCA. More specifically, in Experiment I, we evaluate the representation power of the various VQ and DL approximation schemes presented in Section III, by measuring the quantization error incurred by the techniques, namely the residual between the original subvectors $\mathbf{W}$ defined in (1), and their approximations. This experiment helps us gain insight into the employed techniques and set optimal values for the required parameters. In Experiment II, the focus is on the application of multi-modal fusion of 2D-based and 3D-based data for object detection in an automotive setting. It is shown that multi-modal fusion has a positive impact on the performance of object detection not only before but also after the application of MCA techniques.

A. EXPERIMENT I: MEASURING THE QUANTIZATION ERROR

Here, we focus our attention on the comparative performance of the employed weight-sharing techniques, the relation between the achieved acceleration and compression ratios, as well as the role of subspace grouping. Experiment I is based on individual, pre-trained layers from the widely used image classification CNNs ResNet50 [41] and SqueezeNet [42]. Note that the latter constitutes also the backbone network for the 2D object detector studied in Experiment II. After experimentation, the parameter values that yielded the best results were as follows: subspace dimension $N' = 8$, $c = 3$ (i.e., the DL codebook was 3 times larger than the VQ one), sparsity level (for DL) $\rho = 2$. Finally, to enable a direct comparison between the VQ and DL results, the involved parameters satisfied equality (9), meaning that the two rivals yielded the same acceleration ratio.

In the first part of Experiment I, we present a performance evaluation concerning the two variants of the sparse coding step of the DL technique (as described in Section III), in comparison to the performance obtained by VQ, for a range of acceleration ratios. For this evaluation, we measure the Mean Squared Error (MSE) between the original and approximate kernels of individual convolutional layers from

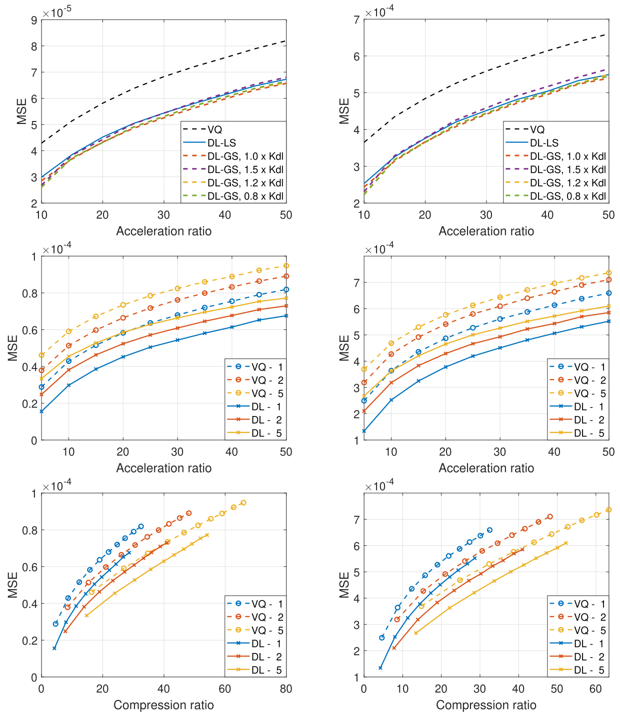


FIGURE 2. MSE as a function of acceleration and compression of DL vs VQ techniques for layers res4f-branch2b of ResNet50 (left), and fire8-expand3 × 3 of SquezeNet (right), using subspace group sizes of 1 (all rows), 2, and 5 (bottom two rows). Where not mentioned, DL refers to the GS variant.

the employed CNNs. A representative instance of this experiment, involving layers (a) res4f-branch2b of ResNet50 (256 kernels of size 3 × 3 × 256), and (b) fire8-expand3 × 3 of SquezeNet (256 kernels of size 3 × 3 × 64), is shown in the top row of Figure 2.

As it is apparent in this figure, the DL-based (both variants) techniques outperform their VQ rival leading to significantly lower MSE for the same acceleration, or, equivalently to a significantly higher acceleration ratio, for the same level of incurred error. Regarding the two variants of DL, it can be observed that the added flexibility enabled by the global sparsity constraint leads to (relative) acceleration gains of DL-GS vs DL-LS of up to approximately 10% in the shown examples, depending on the specific configuration and target acceleration.

It is noted here that for the global sparsity variant, the achieved acceleration/compression is decoupled from the number of employed representatives (i.e. the number of representatives can be altered to meet e.g. specific memory needs without affecting the achieved acceleration), which can be exploited during system design. To enable a direct comparison between the two DL variants in this particular experiment, we set the (global) sparsity P in DL-GS equal to ρK_{dl} where ρ, K_{dl} are the local sparsity and number of representatives, respectively, set for DL-LS (the computational & storage complexity of the DL-based codebook depends on the number of nonzero coefficients in Λ, but not on their locations). The number of representatives for DL-GS was set to various multiples of K_{dl}, as shown in the top row of Figure 2.

We mention finally that since DL-GS generally outperformed DL-LS in all our comparative experiments, in the following, we focus only on the DL-GS variant (indicated hereafter simply as DL).

In the second row of Figure 2, the performance of VQ vs DL, in terms of MSE, is depicted for different values of subspace grouping. As expected, increasing the number of subspaces per group has an impact on performance without changing the relative comparison between VQ and DL. On the other hand, in the third row of Figure 2, the MSE achieved by VQ and DL is depicted versus the achieved compression gain. Again, the advantage of DL vs VQ becomes readily apparent, namely, for the same level of incurred error, the employment of the DL technique results in a considerably higher acceleration and compression ratio.

Finally, we should notice that, as it is apparent from Figure 2, subspace grouping can be used to better control the achieved compression as a function of the acceleration ratio and the incurred quantization error, thus offering additional flexibility at the system design phase. Specifically, one can achieve higher compression ratios by increasing the group size, sacrificing either the achieved acceleration (to keep the system performance constant) or the incurred MSE (to keep the acceleration constant).

For illustration purposes, let us focus on an example drawn from the MSE values on the bottom right plot of Figure 2. Specifically, as it can be observed there, one can incur roughly the same quantization error of $\approx 5 \times 10^{-4}$ (i.e. comparable accuracy loss), by using the following combinations (α: acceleration ratio, τ: compression ratio):

VQ:
a) No grouping: $\alpha \approx 20$, $\tau \approx 15$
b) Groups of 2: $\alpha \approx 15$, $\tau \approx 21$

DL:
a) No grouping: $\alpha \approx 40$, $\tau \approx 25$
b) Groups of 2: $\alpha \approx 30$, $\tau \approx 30$
c) Groups of 5: $\alpha \approx 25$, $\tau \approx 38$

B. EXPERIMENT II: APPLICATION ON MULTI-MODAL FUSION DRIVEN OBJECT DETECTION

In this experiment, we evaluate the performance of the presented weight-sharing MCA techniques when paired with the proposed multi-modal fusion scheme, combining two automotive detection modalities. Specifically, the SqueezeDet [13] and PointPillars [18] models have been used for image-based and point-cloud-based object detection, respectively. Note that SqueezeDet is a representative of a family of lightweight models used for 2D automotive object detection (along with other models such as Mini-YOLOv3 [25]). Being a lightweight model, it can be considered as a worst-case scenario, thus, helping us assess the performance of the employed MCA techniques under non-favorable conditions, namely, when less redundancy in the parameters is present (as opposed to larger models such as PointPillars). The late

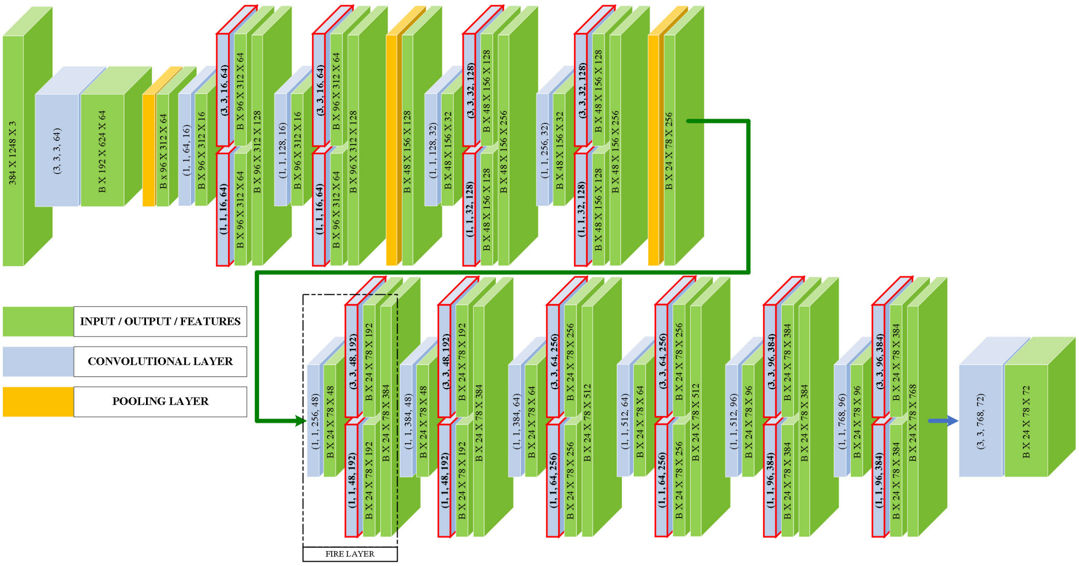


FIGURE 3. Architecture of SqueezeDet (2D detector). The convolutional layers highlighted by the red frames constitute the target layers in our acceleration experiments. *B* is the batch size, *H* is the height and *W* is the width of a volume kernel. C_{L-1} is the number of channels of the previous layer.

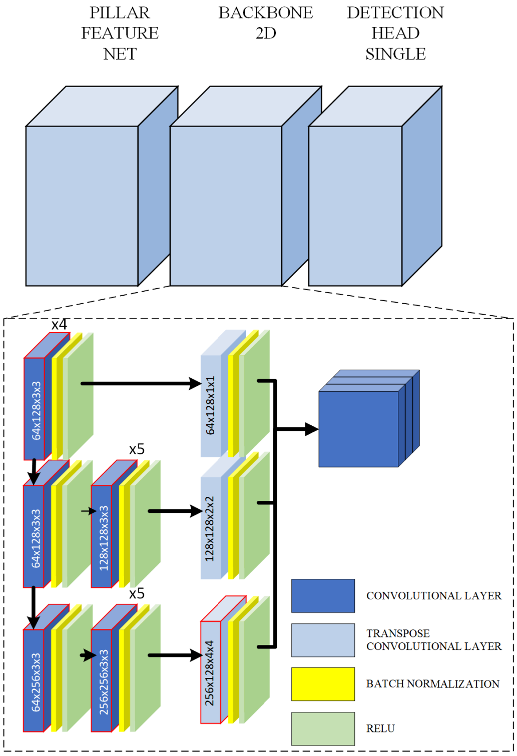


FIGURE 4. Architecture of PointPillars (3D detector). The convolutional layers highlighted by the red frames constitute the target layers in our acceleration experiments.

fusion method (described in Sec. IV) has been implemented via modifying the well-known OpenPCDet suite.[1]

1) DESCRIPTION OF THE DEEP MODELS

SqueezeDet is a fully convolutional detection network presented by Wu et al. [13], consisting of a feature-extraction part that extracts high dimensional feature maps for the input image, and ConvDet, a convolutional layer to locate objects and predict their class. For the derivation of the final detection, the output is filtered based on a confidence index also extracted by the ConvDet layer. Figure 3 presents the overall architecture of the deep networks, the convolutional volume kernel shapes, and the feature tensor shapes.

As it can be observed from Figure 3, the feature-extraction (convolutional) part of SqueezeDet is based on SqueezeNet [42], which is a fully convolutional neural network that employs a special architecture that drastically reduces its size while remaining within the state-of-the-art performance territory. Its building block is the "fire" module that consists of a "squeeze" 1 × 1 convolutional layer to reduce the number of input channels, followed by 1 × 1 and 3 × 3 "expand" convolutional layers that are connected in parallel to the "squeezed" output. SqueezeNet consists of 8 such modules connected in series.

On the other hand, PointPillars [18] is designed for 3D object detection using LIDAR point clouds. Its architecture consists of three main stages. More specifically, the first stage transforms the point cloud into a pseudo-image by grouping the points of the cloud into vertical columns, called pillars, that are positioned based on a partition of the $x - y$ plane. The second stage consists of feature extraction backbone network providing high-level feature-rich representations of the input. Finally, object detection takes place in the third stage, which is responsible for producing 3D bounding boxes and confidence scores for the classes of interest.

[1] https://github.com/snousias/multimodal-fusion-driven-automotive-scene-analysis

2) DESCRIPTION OF THE DATASET

Our experiments were based on the well-known KITTI benchmark dataset [35], [43]. The KITTI object detection dataset is used for training and post-quantization retraining of the detectors and the tracking dataset is used for testing and evaluation. For training, the dataset consists of 7481 training images and 7518 test images, as well as the corresponding point clouds comprising a total of 80256 labeled objects. For evaluation, the KITTI tracking dataset contains annotations or eight different classes with 21 training sequences and 29 test sequences. Three classes are considered for evaluation, namely cars, cyclists, and pedestrians. Table 3 presents the number of visible objects per class and for each track in the evaluation dataset, comprising in total of 27300 cars, 11470 pedestrians, and 1938 cyclists.

3) THE ADOPTED TRAINING PROCEDURE

Both networks were trained with the KITTI object detection dataset [35]. For the deployment and retraining of PointPillars, the OpenPCDet framework was employed [44]. For the initial evaluation, pre-trained instances were used, while for retraining, the Adam optimizer was employed with learning rate $l_r = 0.003$, weight decay rate $D_W = 10^{-2}$ and a batch size $B = 4$. Training took place in an NVIDIA Geforce RTX 2080 with 16GB VRAM and compute capability 7.5.

For the training of the SqueezeDet architecture, Stochastic Gradient Descent (SGD) was used. The following values for the hyperparameters where selected for training: batch size $B = 8$, learning rate $LR = 10^{-4}$, with a weight decay rate $D_W = 10^{-4}$, a learning rate decay rate of $D_{LR} = 2 * LR/N_e$, number of steps $N_s = 3 \times N_{tr}$ and a dropout rate of 50%, over a total of $N_e = 300$ epochs. Training and testing took place in an NVIDIA GeForce GTX 1080 graphics card with 8GB VRof AM and compute capability 6.1 in a Intel(R) Core(TM) i7-4790 CPU @ 3.60Hz based system with 32GB of RAM. A data augmentation scheme was adopted, according to which the bounding boxes drift by $k_x * 150$ and $k_y * 150$ pixels across the x-axis and the y-axis, respectively, where $k_x, k_y \sim U(0, 1)$. A 50% probability is also assumed to flip an object.

4) ACCELERATING 2D AND 3D OBJECT DETECTORS

In this experiment, we follow the acceleration strategy proposed in [34], whereby isolated parts of the network (e.g., individual layers) are quantized progressively, in stages, beginning at the original network. After each stage, the remaining original layers are retrained (or, fine-tuned). The reported acceleration ratios are defined in III-B.

Concerning SqueezeDet, the focus is on its feature-extraction part, namely consists of 8 ''fire'' modules connected in series. SqueezeNet is responsible for roughly 83% of the total 5.3×10^9 MAC operations and 76% of the approximately 16 MB storage space required by SqueezeDet. Since it constitutes an already efficient network, we only targeted SqueezeNet's ''expand'' layers in our experiments. Acceleration was performed in 8 acceleration stages (one ''expand'' module per stage), followed by fine-tuning.

Concerning PointPillars, its feature-extraction (backbone) stage is responsible for 97.7% of the total MAC operations required. In total, the Pointpillars network encompasses 4.835×10^6 parameters and requires 63.835×10^9 MACs. For a good balance between acceleration and accuracy loss, we only targeted the convolutional layers of the backbone network comprising the second stage of PointPillars. Specifically, the targeted 2D- and 4 × 4 transposed 2D- convolutional layers, are responsible for approximately 47% and 44.4% of the total required MACs, respectively. Acceleration was performed on 16 acceleration stages with each stage involving the quantization of a particular layer, followed by fine-tuning. Using the acceleration ratios α = 10, 20, 30, and 40 on the targeted layers leads to a reduction of the total required MACs by 82%, 86%, 88%, and 89%, or equivalently, to total model acceleration of PointPillars by 5.6×, 7.6×, 8.6×, and 9.2×, respectively.

TABLE 2. Acceleration and compression gains for the SqueezeDet and PointPillars networks under study, concerning both the Feature Extraction (FE) part and the Total model.

Model	**SqueezeDet**	
Gain	MAC reduction (%)	
Model Part	FE	Total
α = **10**	74	60
α = **20**	78	65

Model	**Pointpillars**	
Gain	MAC reduction (%)	
Model Part	FE	Total
α = **10**	84	82
α = **20**	88	86
α = **30**	91	88
α = **40**	92	89

TABLE 3. Information for each route in KITTI tracking dataset.

Track	# Cars	# Pedestrians	# Cyclist
0000	243	22	154
0001	2681	112	0
0002	1032	180	75
0003	363	0	0
0004	818	65	60
0005	1275	0	139
0006	550	0	0
0007	2258	67	0
0008	1046	0	0
0009	2859	29	0
0010	603	30	14
0011	3405	201	0
0012	144	64	41
0013	55	929	237
0014	455	122	0
0015	899	752	537
0016	836	2027	272
0017	0	782	101
0018	1354	0	0
0019	927	6088	308
0020	5497	0	0
Total	27300	11470	1938

TABLE 4. Average precision for detection network and their respective fusion. Acceleration approaches $VQ_{a=10}$ and $DL_{a=10}$ demonstrate the robustness of multi-modal fusion as an approach to combine the benefits of weak detectors.

		Original			**VQ**$_{a=10}$						**DL**$_{a=10}$					
Class	*Difficulty*	*Fusion*	*Image*	*LIDAR*	*Fusion*		*Image*		*LIDAR*		*Fusion*		*Image*		*LIDAR*	
Car	Easy	83.8	63.7	78.0	83.3	**-0.5**	56.8	-6.9	77.8	-0.2	82.6	-1.2	58.9	-4.8	77.8	-0.2
	Moderate	85.4	56.4	79.0	83.7	-1.7	46.8	-9.6	78.9	-0.1	84.0	**-1.4**	53.4	-3.0	78.8	-0.2
	Hard	84.7	55.5	79.0	77.7	-7.0	45.1	-10.4	78.8	-0.1	77.7	-7.0	46.8	-8.6	78.7	-0.3
Pedestrian	Easy	86.2	76.4	78.3	80.1	-6.2	57.8	-18.6	75.5	-2.8	81.4	**-4.9**	62.8	-13.6	76.3	-2.0
	Moderate	86.1	76.4	68.1	64.4	-21.8	42.7	-33.7	64.4	-3.7	70.2	**-15.9**	50.9	-25.4	65.9	-2.2
	Hard	85.0	69.3	60.8	63.4	-21.6	41.8	-27.6	58.4	-2.5	69.3	**-15.8**	50.2	-19.1	59.4	-1.5
Cyclist	Easy	71.2	68.4	56.8	59.2	-12.0	49.2	-19.2	46.0	-10.9	65.3	**-5.9**	53.7	-14.7	47.8	-9.0
	Moderate	70.3	63.2	57.3	59.1	-11.2	44.8	-18.4	46.5	-10.9	61.4	**-8.9**	52.5	-10.8	48.3	-9.0
	Hard	68.3	61.4	50.2	58.1	-10.2	43.3	-18.0	46.7	-3.5	59.9	**-8.4**	46.6	-14.8	48.2	-2.0

5) METRICS

The official KITTI evaluation detection metrics include Bird's Eye View (BEV), 3D, 2D, and Average Orientation Similarity (AOS). The 2D detection is done in the image plane and average orientation similarity assesses the average orientation (measured in BEV) similarity for 2D detections [43]. The KITTI dataset is categorized into easy, moderate, and hard difficulties, and the official KITTI leaderboard is ranked by the performance of "moderate". Each 3D ground truth detection box is assigned to one out of three difficulty classes easy, moderate, hard), and a 40-point Interpolated Average Precision metric is separately computed on each difficulty class, according to [45].

To measure the accuracy of our approach we project the 3D bounding boxes on the 2D image and evaluate the outcome with the 2D KITTI evaluation suite deriving the average precision via the Precision/Recall curve. The Precision/Recall curve is defined as

$$\mathrm{AP}|_R = {}^1\!/_{|R|} \sum_{r \in R} \rho_{interp}(r) \tag{17}$$

averaging the precision values provided by $\rho_{interp}(r)$, according to [45]. In our setting, we employ forty equally spaced recall levels,

$$R_{40} = \{1/40, 2/40, 3/40, \ldots, 1\} \tag{18}$$

The interpolation function is defined as $\rho_{interp}(r) = \max_{r':r' \geq r} \rho(r')$, where $\rho(r)$ gives the precision at recall r, meaning that instead of averaging over the observed precision values per point r, the maximum precision at recall value greater or equal than r is taken.

For each detection, the IOU score is computed as the ratio of the area of intersection to the area of union between the predicted and ground-truth bounding boxes. A true positive occurs when IOU $> \lambda$ and the predicted class is the same as the ground-truth class, for some predefined threshold λ. A false positive occurs when IOU $< \lambda$ or a different class is detected, meaning that unmatched bounding boxes are taken as false positives for a given class. Precision, recall and mean average precision (mAP) are subsequently calculated according to [46]. It is important to highlight that the performance of the image detector, the LIDAR detector, and their fusion is measured using the 2D benchmark via projecting the bounding boxes to the 2D modality space.

6) RESULTS

For this experiment, the initial network architectures are compared with the accelerated ones via the vector quantization and dictionary learning approaches for an acceleration ratio $a = 10$. Table 4 presents the results for 2D, 3D and fusion-based object detection using the Average Precision (AP) per class and per difficulty, for all objects within the dataset.

As the table reveals, in all cases, the fusion of modalities generates better results than each detector's ones, showcasing the acceptable performance of even a simplistic late fusion approach. The compression of the models in all cases, deteriorates the detection outcome of the individual detectors as the highlighted columns indicate. However, it is interesting to note that the late fusion approach improves the performance of the overall model even when the MCA techniques are applied, resulting in accelerations of about 2.5× and 6× for the 2D and 3D detectors, respectively, while the performance loss of the fused results ranging in most cases within single-digits figures (as low as around 1% for the class "cars").

Comparing the performance of the utilized uni-modal detectors, it becomes readily apparent, that the 3D LIDAR based detector is much more resilient with respect to the incurred accuracy loss due to the application of acceleration/compression. This comes as a direct consequence of the fact that SqueezeNet (i.e. the back-bone network of SqueezeDet) constitutes an already optimized lightweight network, as opposed to PointPillars, whose architecture is much more "redundant" in the number of filters/parameters. Additionally, it can be observed that the performance of the DL-based weight-sharing MCA technique, is universally better than the one obtained via the VQ-based approach This indicates as expected that the gains in terms of weight approximation (i.e. quantization) error presented in Experiment I, are translated to analogous gains concerning the performance loss of the accelerated networks.

Finally, let us provide some indicative examples of object detection using the 2D, 3D and fusion-based approaches. Figures 5 and 6 present qualitative outcomes of detector fusion. In the figures, green boxes represent the 3D outcomes, red boxes the 2D detector outcomes and the blue boxes

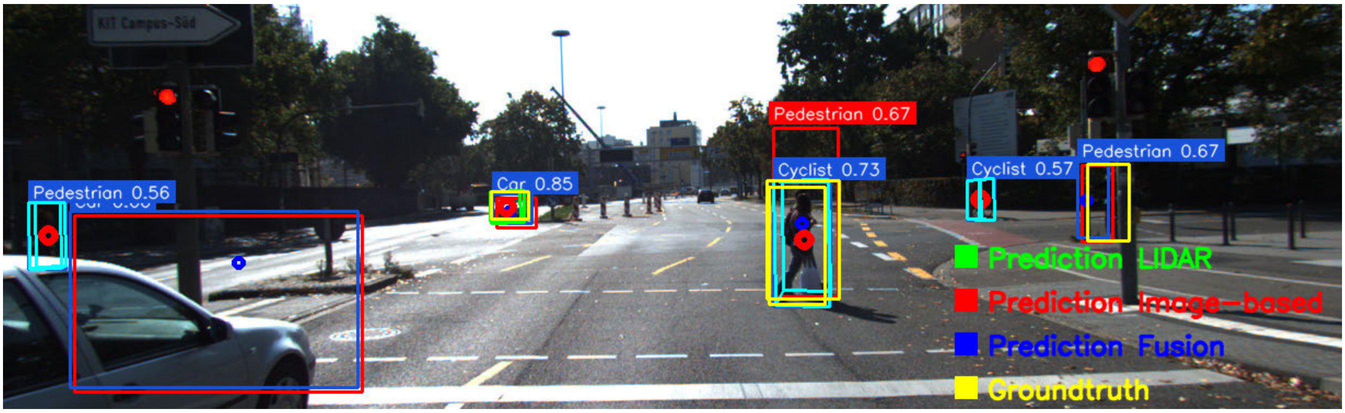


FIGURE 5. Qualitative fusion evaluation outcome. Tracking route #0008.

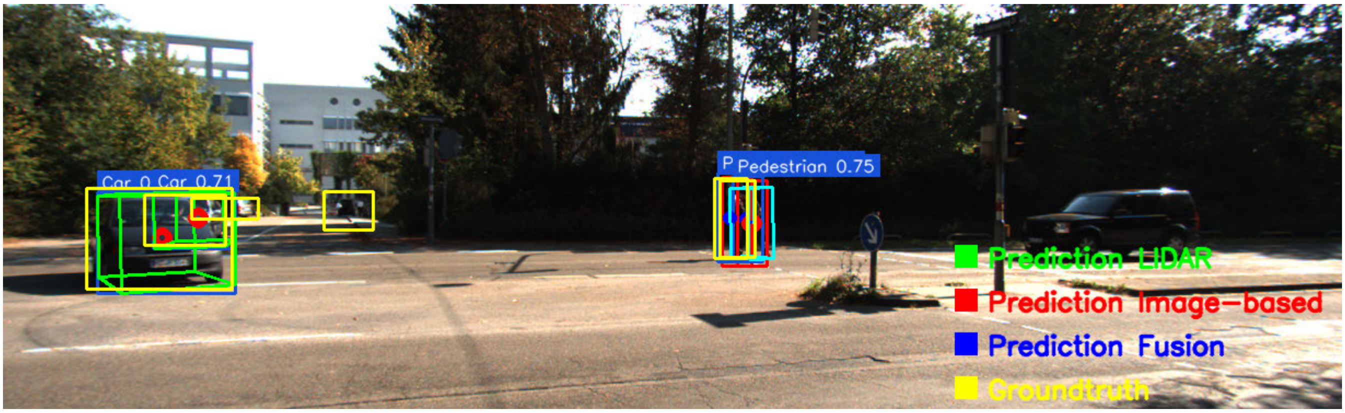


FIGURE 6. Qualitative fusion evaluation outcome. Tracking route #0014.

represent their fusion., We can identify that at least two cars are captured by only one of the two detectors which subsequently contributes to fusion outcome.

VI. CONCLUSION

This work investigates the application of weight-sharing methods in deep learning-based scene analysis for automotive scenarios. The impact of transforming (i.e., accelerating and compressing) two well-known DNN models is evaluated on 2D image-based, 3D LiDAR-based and fusion-based detection approaches. The KITTI dataset is used for the evaluation of the presented approaches. Two state-of-the-art weight sharing techniques are considered and two novel extensions are proposed and their efficacy is presented via Experiment I. Comparing the uni-modal vs multi-modal detection approaches, it is demonstrated that the multi-modal fusion not only improves the performance of the individual detectors, but also considerably improves the performance of the networks when they are accelerated / compressed by the considered weight sharing techniques.

REFERENCES

[1] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1038–1051, Feb. 2019.

[2] A. Kloukiniotis, A. Papandreou, A. Lalos, P. Kapsalas, D.-V. Nguyen, and K. Moustakas, "Countering adversarial attacks on autonomous vehicles using denoising techniques: A review," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 61–80, 2022.

[3] W. Chang, S. Burton, C. W. Lin, Q. Zhu, L. Gauerhof, and J. McDermid, "Intelligent and Connected Cyber-Physical Systems: A Perspective from Connected Autonomous Vehicles," in *Intelligent Internet of Things*, F. Firouzi, K. Chakrabarty, and S. Nassif, Eds. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-30367-9_7.

[4] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.

[5] G. Xie, Y. Li, Y. Han, Y. Xie, G. Zeng, and R. Li, "Recent advances and future trends for automotive functional safety design methodologies," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5629–5642, Sep. 2020.

[6] S. Nousias, N. Piperigkos, G. Arvanitis, A. Fournaris, A. S. Lalos, and K. Moustakas, "Empowering cyberphysical systems of systems with intelligence," 2021, *arXiv:2107.02264*.

[7] M. Elazab, A. Noureldin, and H. S. Hassanein, "Integrated cooperative localization for vehicular networks with partial GPS access in urban canyons," *Veh. Commun.*, vol. 9, pp. 242–253, Jul. 2017.

[8] M. Ucińska and M. Pełka, "The effectiveness of the AEB system in the context of the safety of vulnerable road users," *Open Eng.*, vol. 11, no. 1, pp. 977–993, Oct. 2021.

[9] L. Yang, Y. Yang, G. Wu, X. Zhao, S. Fang, X. Liao, R. Wang, and M. Zhang, "A systematic review of autonomous emergency braking system: Impact factor, technology, and performance evaluation," *J. Adv. Transp.*, vol. 2022, pp. 1–13, Apr. 2022.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.

[13] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 129–137.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[15] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[16] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[17] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[19] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[20] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2020.

[21] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10529–10538.

[22] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Feb. 2020.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[24] J. Borrego-Carazo, D. Castells-Rufas, E. Biempica, and J. Carrabina, "Resource-constrained machine learning for ADAS: A systematic review," *IEEE Access*, vol. 8, pp. 40573–40598, 2020.

[25] Q.-C. Mao, H.-M. Sun, Y. B. Liu, and R.-S. Jia, "Mini-YOLOv3: Real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133529–133538, 2019.

[26] H.-H. Nguyen, D. N.-N. Tran, and J. W. Jeon, "Towards real-time vehicle detection on edge devices with NVIDIA Jetson TX2," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, Nov. 2020, pp. 1–4.

[27] N. Krittayanawach and P. Vateekul, "Robust compression technique for YOLOv3 on real-time vehicle detection," in *Proc. 11th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2019, pp. 1–6.

[28] J. Xu, Y. Nie, P. Wang, and A. M. Lopez, "Training a binary weight object detector by knowledge transfer for autonomous driving," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2379–2384.

[29] Z. Wang, J. Zhang, Z. Zhao, and F. Su, "Efficient YOLO: A lightweight model for embedded deep learning object detection," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.

[30] C.-C. Tsai, Y.-H. Yang, H.-W. Lin, B.-X. Wu, E. C. Chang, H. Yu Liu, J.-S. Lai, P. Y. Chen, J.-J. Lin, J. S. Chang, L.-J. Wang, T. T. Kuo, J.-N. Hwang, and J.-I. Guo, "The 2020 embedded deep learning object detection model compression competition for traffic in Asian countries," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.

[31] E. V. Pikoulis, C. Mavrokefalidis, S. Nousias, and A. S. Lalos, "A new clustering-based technique for the acceleration of deep convolutional networks," in *Deep Learning Applications* (Advances in Intelligent Systems and Computing), vol. 1395 and 3, M. A. Wani, B. Raj, F. Luo, and D. Dou, Eds. Singapore: Springer, 2022, doi: 10.1007/978-981-16-3357-7_5.

[32] S. Zhou, M. Xie, Y. Jin, F. Miao, and C. Ding, "An end-to-end multi-task object detection using embedded GPU in autonomous driving," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2021, pp. 122–128.

[33] S. Nousias, E.-V. Pikoulis, C. Mavrokefalidis, A. S. Lalos, and K. Moustakas, "Accelerating 3D scene analysis for autonomous driving on embedded AI computing platforms," in *Proc. IFIP/IEEE 29th Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Oct. 2021, pp. 1–6.

[34] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4730–4743, Oct. 2018.

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[36] H. Lütkepohl, "Handbook of matrices," *Comput. Statist. Data Anal.*, vol. 2, no. 25, p. 243, 1997.

[37] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*. Berlin, Germany: Springer, 2018, pp. 1–43.

[38] A. Paszke, S. Gross, F. Massa, and A. Lerer, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[39] M. Abadi. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. (Feb. 5, 2015). *The Kitti Vision Benchmark Suite*. [Online]. Available: http://www.cvlibs.net/datasets/kitti

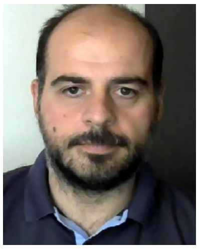
[44] O. Team. (2020). *Openpcdet: An Open-Source Toolbox for 3D Object Detection From Point Clouds*. [Online]. Available: https://github.com/open-mmlab/OpenPCDet

[45] A. Simonelli, S. R. Bulo, L. Porzi, M. Lopez-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1991–1999.

[46] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

[47] C. Morrison and E. S. Sitnikova & Shoval, "A review of the relationship between cyber-physical systems, autonomous vehicles and their trustworthiness," in *Proc. Int. Conf. Cyber Warfare Secur.*, 2018, p. 611.

STAVROS NOUSIAS (Member, IEEE) received the Diploma degree in electrical and computer engineering, the M.Sc. degree in electronics and information processing, and the Ph.D. degree from the University of Patras, in 2011, 2016, and 2022, respectively. From 2015 to 2020, he was a Research Assistant (initially) and a Research Associate (later) with the Visualization and Virtual Reality Group, Department of Electrical and Computer Engineering, University of Patras. From 2019 to 2022, he was a Research Associate with the Industrial Systems Institute, Athena Research and Innovation Center. He has participated in several Horizon 2020 and national research projects as a Research Associate, a Research Engineer, and a Systems Developer. He has authored or coauthored over 35 papers in refereed journals and international conferences. His research interests include computational modeling and simulation, geometric deep learning, point cloud processing, geometry processing, and low-level signal processing.



ERION-VASILIS PIKOULIS received the Diploma, M.A.Sc., and Ph.D. degrees from the Computer Engineering and Informatics Department (CEID), School of Engineering (SE), University of Patras (UoP), Rio Patras, Greece. He is currently a Postdoctoral Researcher with the Signal Processing and Communications (SPC) Laboratory, Department of Computer Engineering and Informatics, University of Patras; and the Industrial Systems Institute (ISI), Research Center Athena, Patras, Greece. His general research interests include automatic seismic signal detection, stochastic signal modeling, parameter estimation, deep learning, pattern recognition, clustering techniques, array processing, beamforming techniques, and inverse problem theory.

CHRISTOS MAVROKEFALIDIS (Member, IEEE) received the Diploma degree in computer engineering and informatics, the master's degree in signal processing systems, and the Ph.D. degree in signal processing for wireless communications from the University of Patras, Greece, in 2004, 2006, and 2011, respectively. Since 2006, he has been a Research Associate with the Signal Processing and Communications Laboratory, Computer Engineering and Informatics Department, University of Patras. Since 2006, he has been affiliated with the Signal Processing and Communication Research Unit, Computer Technology Institute and Press "Diophantus," Patras, Greece. Since 2019, he has been with the Multimedia Information Processing Systems Group, Industrial Systems Institute, Athena Research Center, Patras. He has been involved in numerous national, European, and bilateral research projects in application areas, such as wireless communications and sensor networks, smart grids, and computer vision. In the past, he was also involved in designing integrated circuits for microprocessors that support multimedia operations. His research interests include statistical signal processing and learning with a focus on estimation theory, adaptive/distributed signal processing, and sparse representations. He is a member of the Technical Chamber of Greece. He is a regular reviewer of various journals and conferences in the general area of signal processing.

ARIS S. LALOS received the Diploma, M.A.Sc., and Ph.D. degrees from the Computer Engineering and Informatics Department, University of Patras (UoP), Rio-Patras, Greece, in 2003, 2005, and 2010, respectively. He is a Principal Researcher with the Industrial Systems Institute, Athena Research Centre. He has been a Research Fellow with the Signal Processing and Communications Laboratory, CEID, SE, from 2005 to 2010; and the Signal Theory and Communications (TSC) Department, Technical University of Catalonia (UPC), Barcelona, Spain, from October 2012 to December 2014. From October 2011 to October 2012, he was a Telecommunication Research Engineer with Analogies S. A., an early-stage starts up. In May 2018, he was elected as a Principal Researcher (Associate Research Professor Level with tenure) with the Industrial Systems Institute, Athena Research Centre. He is also a Collaborative Researcher with the Visualization and Virtual Reality Group. He is the author of 113 research papers in international journals (35), conferences (73), and book chapters (5). Many of these publications proposed highly novel frameworks and systems in the areas of digital communication, media processing, 2D-3D signal processing, and learning that go significantly beyond the state-of-the-art. He has participated in several European projects related to the ICT and e-health domain (e.g., COOPCOM, ALPHA, WSN4QoL, KinOptim, MOMIRAS, and MyAirCoach). His general research interests include digital communications, adaptive filtering algorithms, geometry processing, wireless body area networks, and biomedical signal processing.

He received the Best Demo Award in IEEE CAMAD 2014, the Best Paper Award in IEEE ISSPIT 2015, and the World's FIRST 10K Best Paper Award in IEEE ICME 2017. He acts as a regular reviewer of several technical journals. In January 2015, he was nominated as an Exemplary Reviewer of the IEEE Communications Letters. He serves as a Deputy Coordinator for the CPSoSAware H2020 EU Project. He has served as a Technical Coordinator for the GamECAR H2020 EU Project. Furthermore, he has assumed the role of Work Package Leader of five H2020 Projects. After 2018, he formed his own group with ISI that now counts ten active members (post-docs, Ph.D. students, and programming engineers) in research and development projects.