sensors

MDPI

Article

A Tensor Space for Multi-View and Multitask Learning Based on Einstein and Hadamard Products: A Case Study on Vehicle Traffic Surveillance Systems

Fernando Hermosillo-Reynoso *,† and **Deni Torres-Roman** †

Center for Research and Advanced Studies of the National Polytechnic Institute, Department of Electrical Engineering and Computer Sciences, Telecommunications Section, Av. del Bosque 1145, El Bajio, Zapopan 45019, Jalisco, Mexico; deni.torres@cinvestav.mx

* Correspondence: fernando.hermosillo@cinvestav.mx; Tel.: +52-331-631-3095

† These authors contributed equally to this work.

Abstract: Since multi-view learning leverages complementary information from multiple feature sets to improve model performance, a tensor-based data fusion layer for neural networks, called Multi-View Data Tensor Fusion (MV-DTF), is used. It fuses M feature spaces $\mathcal{X}_1, \cdots, \mathcal{X}_M$, referred to as views, in a new latent tensor space, $\mathcal{S}$, of order P and dimension $J_1 \times \cdots \times J_P$, defined in the space of affine mappings composed of a multilinear map $\mathcal{T} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \rightarrow \mathcal{S}$—represented as the Einstein product between a $(P+M)$-order tensor $\boldsymbol{\mathcal{A}}$ anda rank-one tensor, $\boldsymbol{\mathcal{X}} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)}$, where $\mathbf{x}^{(m)} \in \mathcal{X}_m$ is the m-th view—and a translation. Unfortunately, as the number of views increases, the number of parameters that determine the MV-DTF layer grows exponentially, and consequently, so does its computational complexity. To address this issue, we enforce low-rank constraints on certain subtensors of tensor $\boldsymbol{\mathcal{A}}$ using canonical polyadic decomposition, from which M other tensors $\boldsymbol{\mathcal{U}}^{(1)}, \cdots, \boldsymbol{\mathcal{U}}^{(M)}$, called here Hadamard factor tensors, are obtained. We found that the Einstein product $\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}$ can be approximated using a sum of R Hadamard products of M Einstein products encoded as $\boldsymbol{\mathcal{U}}^{(m)} \circledast_1 \mathbf{x}^{(m)}$, where R is related to the decomposition rank of subtensors of $\boldsymbol{\mathcal{A}}$. For this relationship, the lower the rank values, the more computationally efficient the approximation. To the best of our knowledge, this relationship has not previously been reported in the literature. As a case study, we present a multitask model of vehicle traffic surveillance for occlusion detection and vehicle-size classification tasks, with a low-rank MV-DTF layer, achieving up to 92.81% and 95.10% in the normalized weighted Matthews correlation coefficient metric in individual tasks, representing a significant 6% and 7% improvement compared to the single-task single-view models.

Keywords: Einstein product; Hadamard product; Hadamard factor tensors; multi-view learning; multitask learning; vehicle traffic surveillance

check for updates

Citation: Hermosillo-Reynoso, F.; Torres-Roman, D. A Tensor Space for Multi-View and Multitask Learning Based on Einstein and Hadamard Products: A Case Study on Vehicle Traffic Surveillance Systems. *Sensors* **2024**, *24*, 7463. https://doi.org/10.3390/s24237463

Academic Editor: George Yannis

Received: 28 September 2024
Revised: 16 November 2024
Accepted: 18 November 2024
Published: 22 November 2024

1. Introduction

Vehicle traffic surveillance (VTS) systems are key components of intelligent transportation systems (ITSs), as they enable the automated video content analysis of traffic scenes to extract valuable traffic data. It includes crucial aspects of vehicle behavior, such as trajectories and speed, as well as traffic parameters, e.g., lane occupancy, traffic volume, and density. These data serve as the cornerstone for a variety of high-level ITS applications, including collision detection [1,2], route planning, and traffic control [3,4]. Currently, there exist several mathematical models for various tasks related to vehicle traffic, each with different conditions and traffic network topologies. For a comprehensive overview of vehicle traffic models, see, e.g., [5].

However, due to the complex nature of vehicle traffic, VTS systems are usually broken down into a set of smaller tasks, including vehicle detection, occlusion handling, and classification [6–14]. Each task is represented as a feature model, which should be related to the underlying task-specific explanatory factors, while it is either developed by human experts (hand-crafted) or automatically learned. These features focus on specific aspects of vehicles, such as texture, color, and shape, which individually provide complementary information to each other. Therefore, finding a highly descriptive feature model is crucial for enhancing the learning process on every VTS task.

Such feature diversity has made data fusion (DF) attractive for leveraging its shared and complementary information. DF allows for the integration of data from different sources to enhance our understanding and analysis of the underlying process [15]. In this context, there are two common DF levels [16]: low-level, where data are combined before analysis, and decision-level, where processed data from each source are integrated at a higher level, such as in ensemble learning [17]. Moreover, the diverse nature of data sources poses challenges, such as heterogeneity across sources, high-dimensional data, missing values, and a lot of redundancy that DF algorithms should address [18,19].

As part of DF, multi-view learning (MVL) is a machine learning (ML) paradigm that exploits the shared and complementary information contained in multiple data sources, called views, obtained from different feature sets [20]. Here, data represented by M views are referred to as M-view data. For instance, an image represented by texture, edges, and color features can be regarded as three-view data. MVL methods can be grouped into three categories: co-training, multiple kernel learning, and subspace learning (SL) [21,22]. Among these, SL-based methods focus on learning a low-dimensional latent subspace that captures the shared information across views [23].

On the other hand, multitask learning (MTL) is another ML paradigm where multiple related tasks are learned simultaneously to leverage their shared knowledge, with the ultimate aim of improving generalization and performance in individual tasks [24–28].

Recently, artificial neural networks (ANNs) have shown superior performance in vision-based VTS systems. ANNs are computational models built from a composition of functions, called layers, which together capture the underlying relationships between the so-called input and output spaces to solve a given task, such as regression or classification [29]. Such layers, including fully connected (FC) and convolutional (Conv), are parameterized by weights and biases structured as tensors, matrices, or vectors, which are learned during training. Notably, the first layers usually act as feature extractors, whereas higher layers capture the relationships between extracted features and the output space.

Furthermore, higher-order tensors [30], or multidimensional arrays, have gained significant attention over the last decade due to their ability to naturally represent multi-modal data, e.g., images and videos, and their interactions. They have been successfully applied in various domains, including signal processing [31], machine learning [32–37], computer vision [38], and wireless communications [39,40]. For instance, tensor methods such as decomposition models have been employed for the low-rank approximation of tensor data, enabling more efficient and effective analysis of such data.

In this work, we propose a computationally efficient tensor-based multi-view data fusion layer for neural networks, here expressed as the Einstein product. Our approach leverages multiple feature spaces to address the limitations inherent to single-view models, such as reduced data representation capacity and model overfitting. It offers improved flexibility and scalability, as it enables the integration of additional views without significantly increasing the computational burden. Finally, we present a case study with a multitask, multi-view VTS model, demonstrating significant performance improvements in vehicle-size classification and occlusion-detection tasks.

1.1. Related Work

Occlusion detection is a challenging problem in vision-based tasks, in which vehicles or some parts of them are hidden by other elements in the traffic scene, making their detection a difficult task. Early works have explored approaches based on empirical models, which infer the presence of occlusion by assuming specific geometric patterns, such as concavity in the shape of occluded vehicles [41–47]. Recently, deep learning (DL) has also been employed for occlusion detection [48–52], where such models are even capable of reconstructing the occluded parts [53,54].

Several algorithms based on ML and DL have been proposed for intra- and inter-class vehicle classification [6,8,9,55–60]. In [8], Hsieh et al. employ the optimal classifier to categorize vehicles as cars, buses, or trucks by leveraging the linearity and size features of vehicles, achieving accuracy of up to 97.0%. Moussa [9] introduces two levels of vehicle classification: the multiclass level, which categorizes vehicles as small, midsize, and large, and the intra-class level, in which midsize vehicles are classified as pickups, SUVs, and vans. In [6], we proposed a one-class support vector machine (OC-SVM) classifier with a radial basis kernel to classify vehicles as small, midsize, and large. By representing vehicles in a 3D feature space (area, width, and aspect-ratio) features, a recall, precision, and f-measure of up to 99.05% were achieved for the midsize class. Other techniques include the gray-level co-occurrence matrix (GLCM) [61], 3D appearance models [62–64], eigenvehicles [65], and non-negative factorization [66–68]. Recently, CNN-based classifiers have been employed, outperforming previous works [55,58–60,69].

Other works based on MLV and MTL have also been developed for VTS systems. For instance, Wang et al. [70] proposed an MVL approach to foreground detection, where three-view heterogeneous data (brightness, chromaticity, and texture variations) are employed to improve detection performance. Then, their conditional probability densities are estimated via kernel density estimation, followed by pixel labeling through a Markov random field. In [71], a multi-view object retrieval approach to surveillance videos integrates semantic structure information from CNNs trained on ImageNet and deep color features, using locality-sensitive hashing (LSH) to encode the features into short binary codes for efficient retrieval. Chu et al. [72] present vehicle detection with multitask CNNs and a region-of-interest (RoI) voting scheme. This framework addresses simultaneously supervision with subcategory, region overlap, bounding-box regression, and category information to enhance detection performance. In [73], a multi-task CNN for traffic scene understanding is proposed. The CNN consists of a shared encoder and specific decoders for road segmentation and object detection, generating complementary representations efficiently. Additionally, the detection stage predicts object orientation, aiding in 3D bounding box estimation. Finally, Liu et al. [74] introduce the Multi-Task Attention Network (MTAN), a shared network with a global feature pooling and task-specific soft-attention modules to learn task-specific features from global features while allowing feature sharing across tasks.

Although, our work is focused on multi-view and multitask VTS systems, some works related to other domains are also overviewed. In [36], a tensor-based, multi-view feature selection method called DUAL-TMFS is proposed for effective disease diagnosis. This approach integrates clinical, imaging, immunologic, serologic, and cognitive data into a joint space using tensor products, and it employs SVM with recursive feature elimination to select relevant features, improving classification performance in neurological disorder datasets. Zadeh et al. [75] introduce a novel model called a tensor fusion network for multimodal sentiment analysis. It leverages the outer product between modalities to model both the intra-modality and inter-modality dynamics. On the other hand, Liu et al. [76] propose an efficient multimodal fusion scheme using low-rank tensors. Experimental validations across multimodal sentiment analysis, speaker trait analysis, and emotion recognition tasks demonstrate competitive performance and robustness across a variety of low-rank settings.

Table 1 offers a comprehensive overview of existing research related to our approach and to VTS systems. It highlights the use of ML and DL approaches, fed either by hand-crafted features or raw data with automatic feature learning, to capture the underlying task patterns. While DL features generally achieve superior performance, they require large, high-quality training sets and high computational complexity models to find suitable representations. Conversely, hand-crafted features can perform competitively for specific tasks, but determining the optimal feature representation is challenging, as no single hand-crafted feature can fully describe the underlying task's relationships.

Furthermore, the emerging trend towards the adoption of ANN models on VTS systems is evident. However, despite their high performance, these models demand substantial memory and computational resources for learning and inference, as their layers are usually overparameterized. To address these challenges, various techniques such as sparsification, quantization, and low-rank approximation have been proposed to compress the parameters of pre-trained layers [77–83]. Among these techniques, low-rank approximation is very often employed. In [79,80], Denil et al. compress FC layers using matrix decomposition models. Conv layers are compressed via tensor decompositions, including canonical polyadic decomposition (CPD) [81,82] and Tucker decomposition [83]. However, compressing pre-trained layers usually results in an accuracy loss, and a fine-tuning procedure is often employed to recover the accuracy drop [82,84–86]. Therefore, some authors have suggested the incorporation of low-rank constraints into the optimization problem [87–89]. Other works have found that compressing raw images before training also contributes to computational complexity reduction, as suggested in [32,90]. Additionally, in [91], tensor contraction layers (TCLs) and tensor regression layers (TRLs) are introduced in CNNs for dimensionality reduction and multilinear regression tasks, respectively. This approach imposes low-rank constraints via Tucker decomposition on the weights of TCLs and TRLs to speed up their computations.

Table 1. Related work summary.

Reference	Input	Method	Contribution
[6,8–10,14]	Single-view	ML	Hand-crafted geometric features represent vehicles for detection and classification using ML-based algorithms
[11,12]	Single-view	DL	CNN models are proposed to perform automatic feature learning for vehicle detection and classification
[65]	Single-view	Eigenvalue decomposition	Eigenvehicles are introduced as an unsupervised feature representation method for vehicle recognition
[66–68]	Single-view	Nonnegative factorization	A part-based model is employed for vehicle recognition via non-negative matrix/tensor factorization
[72–74]	Single-view	DL-based MTL	MTL models based on DL are employed to simultaneously perform multiple tasks, including road segmentation, vehicle detection and classification
[92]	Multi-view	DL	This work employs a YOLO-based model that fuses camera and LiDAR data at multiple levels
[61,93,94]	Single-view	ML	Single-view features, such as HOG, Haar wavelets, or GLCM, represent vehicles for classification in ML models
[95]	Multi-view	Tucker decomposition	A tensor decomposition is employed for feature selection of HOG, LBP, and FDF features
[70,71,96]	Multi-view	MVL	MVL approaches are proposed to enhance vehicle detection, classification, and background modeling by learning richer data representations from color features
[30,97–100]	–	–	These works provide theoretical foundations on tensors and its operations, such as the Einstein and Hadamard products, with applications across multiple domains
[32,77–83,90]	–	DL	Matrix and tensor decompositions are employed for speeding up CNNs by compressing FC and Conv layers and reducing the dimensionality of their input space
[91]	–	DL	Multilinear layers are introduced for dimensionality reduction and regression purposes in CNNs, leveraging tensor decompositions for efficient computation.

1.2. Contributions

The main contributions of this work are the following:

1. We found a novel connection or mathematical relationship between the Einstein and Hadamard products for tensors (for details, see Section 5.2). From this connection, other algorithms for efficient approximations of the Einstein product can be developed.
2. Since multi-view models provide a more comprehensive input space than single-view models, we employ a tensor-based data fusion layer, here called multi-view data tensor fusion (MV-DTF). Unlike other works, our approach maps the multiple feature spaces (views) into a latent tensor space, $\mathcal{S}$, using a multilinear map, here expressed as the Einstein product (see Section 5), followed by a translation.
3. A major drawback of the MV-DTF layer is its high computational complexity, which grows exponentially with the number of views. To address this issue, a low-rank approximation for the MV-DTF layer, here called the low-rank multi-view data tensor fusion (LRMV-DTF) layer, is also proposed. This approach leverages the novel relationship between the Einstein and Hadamard products (see Section 5.2), where the lower the rank values, the more computationally efficient the operation.
4. As a case study, we introduce a high-performance multitask ANN model for VTS systems capable of simultaneously addressing various VTS tasks but which is here limited to occlusion detection and vehicle-size classification. This model incorporates the proposed LRMV-DTF layer as multi-view feature extractor to provide a more comprehensive input space compared to individual spaces.

1.3. How to Read This Article

For a comprehensive understanding of this paper, the following is suggested the following: Section 1 presents the motivation behind our research on VTS systems, as well as a review of their related works, while Section 2 introduces tensor algebra and multilinear maps, which will be essential for understanding the subsequent mathematical definitions; however, if you are already familiar with their theoretical foundations, you can proceed directly to Section 3 to delve into the problem statement and its mathematical formulation, where the main objectives are stated. These objectives are important to understand the major results of the paper. Section 4 provides a comprehensive overview of VTS systems and their associated tasks as an important case study. If you are already familiar with these concepts, proceed to Section 5 for the technical and mathematical details of the MV-DTF layer. Particularly, Section 5 is very important because it presents the novel connection between Einstein and Hadamard products. Section 6 presents the results and their analysis for a deeper understanding of our findings, which are complemented by figures and tables to facilitate data interpretation. Finally, Section 7 provides the conclusions of this work, summarizing the key points and suggesting directions for future research.

2. Mathematical Background

2.1. Notation

In this study, we adopt the conventional notation established in [30], along with other commonly used symbols. Table 2 provides a comprehensive overview of the symbols utilized in this paper. An Nth-order tensor is denoted by $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where the dimension $\mathrm{I_n}$ is usually referred to as the n-mode of $\mathcal{X}$. The ith entry of a vector, $\mathbf{x} \in \mathbb{R}^I$, is denoted as x_{i}; the (i,j)th entry of a matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ by x_{ij}; while the $(i_1, \cdots, i_N)$th entry of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is denoted as $x_{\mathrm{i_1 i_2 \cdots i_N}}$, where $i_n \in [I_n]$ is called the n-mode index. The n-mode fiber of an Nth-order tensor is an I_n-dimensional vector resulted from fixing every index but $\mathrm{i_n}$; i.e., $\mathcal{X}_{\mathrm{i_1 i_2 \cdots i_{n-1} : i_{n+1} \cdots i_N}} \in \mathbb{R}^{I_n}$, where colon mark $:$ denotes all possible values of the n-mode index i_n, i.e., $[I_n]$. The ith n-mode slice of an Nth-order tensor is an $(N-1)$th-order tensor defined by just fixing the $\mathrm{i_n}$ index, i.e., $\mathcal{X}_{::\cdots \mathrm{i_n} : \cdots :}$. Finally, for any two functions, f_1 and f_2, $f_1 \circ f_2$ denotes their function composition. For an understanding on tensor algebra, we refer the interested reader to the comprehensive work by Kolda and Bader [30].

Table 2. Basic notation used in this work.

$\mathbb{R}, \mathbb{N}, \mathbb{B}$	The field for real, natural, and binary numbers
$\simeq$	It denotes isomorphism between two structures
$[N]$	The subset of natural numbers $\{1, \cdots, N\} \subset \mathbb{N}$
$\boldsymbol{\mathcal{X}}, \mathbf{X}, \mathbf{x}, x$	Tensor, matrix, column vector, and scalar
$\dim(V)$	The dimension of a vector space, V
$\odot$	Hadamard product
$\otimes$	Tensor product
$\times_n$	The n-mode tensor-matrix product
$\circledast_N$	The Einstein product along the last N modes
$X^{(i_1,\cdots,i_M)}$	The $(i_1, \cdots, i_M)$th element of a sequence, $\{X^{(1,\cdots,1)}, \cdots, X^{(I_1,\cdots,I_M)}\}$ indexed by $i_1 \in [I_1], \cdots, i_M \in [I_M]$, where X can be a scalar, vector, or tensor
$\mathcal{X}_m$	The feature space for the m-th view
$\mathcal{Y}_t$	The output space for the t-th task
f_t	The t-th classification task
$g, \hat{g}$	The MV-DTF layer and its low-rank approximation
h_t	The t-th task-specific function
$\mathcal{H}_t$	Hypothesis space of the classifiers for the t-th task
$\mathcal{S}$	The latent tensor space
P	The order of the latent tensor space $\mathcal{S}$
$J_1 \times \cdots \times J_P$	The dimension of the latent tensor space $\mathcal{S}$
$R^{(j_1,\cdots,j_P)}$	For a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J_1\times\cdots\times J_P\times I_1\times\cdots\times I_M}$, it denotes the tensor rank of the $(j_1, \cdots, j_P)$-th subtensor $\boldsymbol{\mathcal{A}}_{j_1\cdots j_P:\cdots:}$

2.2. Multilinear Algebra

This section provides an overview of basic concepts of multilinear algebra, such as tensors and their operations over a set of vector spaces.

Definition 1 (Multilinear map [101])**.** *Let $V_1, \cdots, V_M$ and W be vector spaces over a field, $\mathbb{R}$. And let $\mathcal{T} : V_1 \times \cdots \times V_M \to W$ be a function that maps an ordered M-tuple of vectors, $(\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(M)}) \in V_1 \times \cdots \times V_M$, into an element, $\mathbf{w} \in W$, where $\mathbf{v}^{(m)} \in V_m \forall m \in [M]$. If, for all $a, b \in \mathbb{R}$ and $\mathbf{v}^{(m)}, \mathbf{u}^{(m)} \in V_m \forall m \in [M]$, Equation (1) holds, then $\mathcal{T}$ is said to be a multilinear map (or an M-linear map); i.e., it is linear in each argument.*

$$\begin{aligned} \mathcal{T}(\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(m-1)}, a\,\mathbf{v}^{(m)} + b\mathbf{u}^{(m)}, \mathbf{v}^{(m+1)}, \cdots, \mathbf{v}^{(M)}) = \\ a\mathcal{T}(\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(m-1)}, \mathbf{v}^{(m)}, \mathbf{v}^{(m+1)} \cdots, \mathbf{v}^{(M)}) + b\mathcal{T}(\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(m-1)}, \mathbf{u}^{(m)}, \mathbf{v}^{(m+1)} \cdots, \mathbf{v}^{(M)}) \end{aligned} \tag{1}$$

Definition 2 (Tensor product)**.** *Let $V_1, \cdots, V_M$ and W be real vector spaces, where $\dim(V_m) = I_m \forall m \in [M]$, and $\dim(W) = J$. Then, the tensor product of the set of M vector spaces $V_1, \cdots, V_M$, denoted as $V_1 \otimes \cdots \otimes V_M$, is another vector space of dimension $\dim(V_1 \otimes \cdots \otimes V_M) = \prod_{m=1}^{M} \dim(V_m)$, called tensor space, together with a multilinear map, $\pi : V_1 \times \cdots \times V_M \to V_1 \otimes \cdots \otimes V_M$, that satisfies the following universal mapping property* [101,102]*: for any multilinear map $\mathcal{T} : V_1 \times \cdots \times V_M \to W$, there exists a unique linear map, $\Phi : V_1 \otimes \cdots \otimes V_M \to W$, such that $\mathcal{T} = \Phi \circ \pi$.*

Definition 3 (Tensor)**.** *Let $V_1, \cdots, V_M$ be vector spaces over some field, $\mathbb{F}$, where $\dim(V_m) = I_m \forall m \in [M]$. An M-order tensor, denoted as $\boldsymbol{\mathcal{X}}$, is an element in the tensor product $V_1 \otimes \cdots \otimes V_M$.*

Definition 4 (m-mode matricization [30])**.** *The m-mode matricization is a mapping that rearranges the m-mode fibers of a tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_M}$, into the columns of a matrix, $\mathbf{X}_{(m)} \in \mathbb{R}^{I_m \times J}$, where $J = \prod_{k=1,k\neq m}^{M} I_k$.*

Definition 5 (Rank-one tensor)**.** *Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_M}$ be an Mth-order tensor, and let $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}$ be a set of M vectors, where $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$ for all $m \in [M]$. Then, if $\boldsymbol{\mathcal{X}}$ can be written using the tensor* *product* $\mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)}$*, it is said to be a rank-one tensor, and its* $(i_1, \cdots, i_M)$*-th entry will be determined by* $x_{\mathrm{i}_1\cdots\mathrm{i}_M} = \prod_{m=1}^{M} x_{i_m}^{(m)}$.

Definition 6 (Tensor decomposition rank)**.** *The decomposition rank, R, of a tensor,* $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_M}$*, is the smallest number of rank-one tensors that reconstructs* $\mathcal{X}$ *exactly as their sum. Then,* $\mathcal{X}$ *is called a rank-R tensor.*

Definition 7 (Tensor multilinear rank)**.** *For any Mth-order tensor,* $\mathcal{X}$*, its multilinear rank, denoted as* $\mathrm{mlrank}(\mathcal{X})$*, is the M-tuple* $(r_1, \cdots, r_M)$*, whose mth entry,* r_m*, corresponds to the dimension of the column space of* $\mathbf{X}_{(m)}$*, i.e.,* $r_m = \dim(\mathrm{Col}(\mathbf{X}_{(m)}))$*, formally called m-mode rank.*

Definition 8 (Tensor m-mode product)**.** *Given a tensor,* $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_M}$*, and a matrix,* $\mathbf{U} \in \mathbb{R}^{J\times I_m}$*, their m-mode product, denoted as* $\mathcal{X} \times_m \mathcal{U}$*, produces another tensor,* $\mathcal{Y} \in \mathbb{R}^{I_1\times\cdots\times I_{m-1}\times J\times I_{m+1}\times\cdots\times I_M}$*, whose* $(i_1, \cdots, i_{m-1}, j, i_{m+1}, \cdots, i_M)$*th entry is given by Equation (2). Therefore,* $\mathcal{Y} = \mathcal{X} \times_m \mathbf{U} \Longleftrightarrow \mathbf{Y}_{(m)} = \mathbf{U}\mathbf{X}_{(m)}$.

$$y_{i_1,\cdots i_{m-1} j i_{m+1}\cdots i_M} = \sum_{i_m=1}^{I_m} x_{i_1,\cdots i_m \cdots i_M} \cdot u_{j i_m} \tag{2}$$

2.3. Einstein and Hadamard Products

In this section, the fundamental concepts for the mathematical modeling of the MV-DTF layer are presented, including the Hadamard and Einstein products.

Definition 9 (Inner product)**.** *For any two tensors,* $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_M}$*, their inner product is defined as the sum of the product of each entry, as Equation (3) shows:*

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_M=1}^{I_M} a_{\mathrm{i}_1\mathrm{i}_2\cdots\mathrm{i}_M} b_{\mathrm{i}_1\mathrm{i}_2\cdots\mathrm{i}_M} \tag{3}$$

Definition 10 (Hadamard product)**.** *The Hadamard product of two Nth-order tensors* $\mathcal{A}$*,* $\mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_M}$*, denoted as* $\mathcal{A} \odot \mathcal{B}$*, results in an Mth-order tensor,* $\mathcal{C} \in \mathbb{R}^{I_1\times\cdots\times I_M}$*, such that its* $(i_1, \cdots, i_M)$*th-entry* $c_{\mathrm{i}_1\cdots\mathrm{i}_M}$ *is equal to the element-wise product* $a_{\mathrm{i}_1\cdots\mathrm{i}_M} \cdot b_{\mathrm{i}_1\cdots\mathrm{i}_M}$.

Definition 11 (Einstein product [100,103])**.** *Given two tensors,* $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_M\times K_1\times\cdots\times K_N}$ *and* $\mathcal{B} \in \mathbb{R}^{K_1\times\cdots\times K_N\times J_1\times\cdots\times J_P}$*, of order* $M+N$ *and* $N+P$*, their Einstein product or tensor contraction, denoted as* $\mathcal{A} \circledast_N \mathcal{B}$*, produces an* $M+P$ *tensor,* $\mathcal{C} \in \mathbb{R}^{I_1\times\cdots\times I_M\times J_1\times\cdots\times J_P}$*, whose* $(i_1, \cdots, i_M, j_1, \cdots, j_P)$*th entry is given by the inner product between subtensors* $\mathcal{A}_{i_1\cdots i_M : \cdots :}$ *and* $\mathcal{B}_{: \cdots : j_1\cdots j_P}$*, as Equation (4) shows:*

$$c_{\mathrm{i}_1,\cdots,\mathrm{i}_M,\mathrm{j}_1,\cdots,\mathrm{j}_P} = \sum_{k_1=1}^{K_1} \cdots \sum_{k_l=1}^{K_N} a_{\mathrm{i}_1\cdots\mathrm{i}_M\mathrm{k}_1\cdots\mathrm{k}_N} b_{\mathrm{k}_1\cdots\mathrm{k}_N\mathrm{j}_1\cdots\mathrm{j}_P} = \langle \mathcal{A}_{i_1\cdots i_M : \cdots :}, \mathcal{B}_{: \cdots : j_1\cdots j_P} \rangle \tag{4}$$

The product $\mathcal{A} \circledast_M \mathcal{B}$ can be understood as a linear map, $T : \mathbb{R}^{I_1\times\cdots\times I_M} \to \mathbb{R}^{J_1\times\cdots\times J_P}$; i.e., for any two scalars $\alpha, \beta \in \mathbb{R}$, and tensors $\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{R}^{K_1\times\cdots\times K_N\times J_1\times\cdots\times J_P}$, the following properties hold:

1. Distributive : $\mathcal{A} \circledast_N (\mathcal{B}_1 + \mathcal{B}_2) = \mathcal{A} \circledast_N \mathcal{B}_1 + \mathcal{A} \circledast_N \mathcal{B}_2$.
2. Homogeneity: $\alpha\mathcal{A} \circledast_N \mathcal{B}_1 = \mathcal{A} \circledast_N \alpha\mathcal{B}_1 = \alpha(\mathcal{A} \circledast_N \mathcal{B}_1)$.

2.4. Subspace Learning

Recent advances in sensing and storage technologies have resulted in the generation of massive amounts of complex data, commonly referred to as big data [104,105]. These data are often represented in a high-dimensional space, making their visualization and analysis a challenging task. To address these challenges, subspace learning methods have emerged as a powerful approach to learning a low-dimensional representation of high-dimensional data [106,107], such as the spatial and temporal information encoded in videos. In this section, a brief review of linear and multilinear methods for subspace learning is presented, highlighting their advantages and disadvantages.

2.4.1. Linear Subspace Learning (LSL)

Given a dataset, $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, of N samples, arranged in matrix form as $\mathbf{X} \in \mathbb{R}^{I \times N}$, whose n-th column vector corresponds to the n-th sample $\mathbf{x}^{(n)} \in \mathbb{R}^{I}$, LSL seeks to find a linear subspace of $\mathbb{R}^{I}$ that best explains the data. The resulting subspace can be spanned by a set of $J < I$ linearly independent basis vectors, $\mathbf{u}_1, \cdots, \mathbf{u}_J$, where $\mathbf{u}_j \in \mathbb{R}^{I}$. By leveraging this subspace, high-dimensional data can be projected onto a lower-dimensional space $\mathbb{R}^{J}$, as Equation (5) shows:

$$\mathbf{G} = \mathbf{U}^T\mathbf{X} = \mathbf{X} \times_1 \mathbf{U}^T \tag{5}$$

where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_J] \in \mathbb{R}^{I \times J}$ is called the factor matrix, whose columns correspond to the basis vectors, and $\mathbf{G} \in \mathbb{R}^{J \times N}$ is the projection of the input matrix $\mathbf{X}$ onto $\mathbf{U}$.

A wide variety of techniques have been proposed to address the LSL problem, ranging from unsupervised approaches such as principal component analysis [108], factor analysis (FA) [109], independent component analysis [110], canonical correlation analysis [111], and singular value decomposition [112], as well as supervised approaches like linear discriminant analysis [113]. Subsequently, such techniques aim to estimate $\mathbf{U}$ by solving optimization problems such as maximizing the variance or minimizing the reconstruction error of the projected data.

Although LSL methods have shown great effectiveness in modeling vector-based observations, they face difficulties when addressing multidimensional data. Then, to apply LSL methods on tensor data, it is necessary to vectorize them. Unfortunately, this transformation very often leads to a computationally intractable problem due to the large number of parameters to be estimated, and the model may suffer from overfitting. Furthermore, vectorization also destroys the inherent multidimensional structure and correlations across modes of tensor data [30,106].

2.4.2. Multilinear Subspace Learning (MSL)

Multilinear subspace learning is a mathematical framework for exploring, analyzing, and modeling complex relationships over tensor data, preserving their inherent multidimensional structure. According to Lu [106], the MSL problem can be formulated as follows: Given a dataset $\{\mathcal{X}^{(1)}, \cdots, \mathcal{X}^{(N)}\}$ arranged in tensor form as $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M \times N}$, where subtensor $\mathcal{X}_{:,\ldots:,n}$ corresponds with the n-th data point $\mathcal{X}^{(n)} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, MLS seeks to find a set of M subspaces that best explains data, where the mth subspace resides in $\mathbb{R}^{I_m}$ and is spanned by a set of $J_m < I_m$ linearly independent basis vectors, $\mathbf{u}_1^{(m)}, \cdots, \mathbf{u}_{J_m}^{(m)} \in \mathbb{R}^{I_m}$. The MSL problem can be formally defined using Equation (6):

$$\underset{\mathbf{U}^{(1)},\cdots,\mathbf{U}^{(M)}}{\arg\max} \; \Phi(\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(M)}, \mathcal{X}) \tag{6}$$

where $\mathbf{U}^{(m)} = [\mathbf{u}_1^{(m)}, \ldots, \mathbf{u}_{J_m}^{(m)}] \in \mathbb{R}^{I_m \times J_m}$ is a matrix whose columns correspond to the basis vectors of the m-th subspace, and Φ denotes a function to be maximized.

A classical MSL technique is the Tucker decomposition [30], which aims to approximate a given Mth-order tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, into a core tensor, $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_M}$, multiplied along the m-mode by a matrix, $\mathbf{U}^{(m)}$, for all $m \in [M]$, as Equation (7) shows:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \cdots \times_M \mathbf{U}^{(M)} \tag{7}$$

where $\mathbf{U}^{(m)} \in \mathbb{R}^{R_m \times I_m}$ is the m-th factor matrix associated with the m-mode fiber space of $\mathcal{X}$, $\mathcal{G}$ captures the level of interaction on each factor matrix, and $R_m = \operatorname{rank}(\mathbf{X}_{(m)})$.

Similarly, canonical polyadic decomposition [30] aims to approximate a given Mth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ into as a sum of R rank-one tensors, as Equation (8) shows:

$$\mathcal{X} \approxeq \sum_{r=1}^{R} \lambda_r \mathcal{X}^{(r)} = \sum_{r=1}^{R} \lambda_r \cdot \left(\mathbf{u}^{(r,1)} \otimes \cdots \otimes \mathbf{u}^{(r,M)} \right) \tag{8}$$

where $\lambda_r \in \mathbb{R}$ is the r-th weighting term, and $\mathbf{u}^{(r,m)} \in \mathbb{R}^{I_m}$ is the m-mode factor vector for the r-th rank-one tensor $\mathcal{X}^{(r)}$, while Equation (8) is exact iff R is the decomposition rank.

While MSL effectively mitigates several drawbacks related to LSL methods, it has some disadvantages. First, the intricate mathematical operations required for MSL methods very often involve high computational complexity, impacting both time and storage requirements. Moreover, MSL requires a substantial amount of data to effectively capture the intricate relationships of multilinear subspaces. Therefore, addressing these challenges is crucial to ensuring proper learning.

3. Problem Statement and Mathematical Definition

In this section, the problem to be addressed is formulated in natural language, outlining specific tasks related to VTS systems. Subsequently, the inherent challenges are mathematically formulated.

3.1. Problem Statement

Given a traffic surveillance video of τ seconds, recorded with a static camera, where multiple moving vehicles are observed, we aim to comprehensively model vehicle traffic using a multitask, multi-view learning approach. This model simultaneously addresses various tasks, such as vehicle detection, classification, and occlusion detection, each of them represented by specific views that partially describe the underlying problem. By projecting multi-view data into a unified, low-dimensional latent tensor space, which builds a new input space for the tasks, our approach should improve the model performance and provide a more comprehensive representation of different study cases, e.g., the traffic scene, compared to single-task, single-view learning models.

3.2. Mathematical Definition

3.2.1. Multitask, Multi-View Dataset: The Input and Output Spaces

Consider a collection of T supervised classification tasks related to VTS systems, such as vehicle detection, classification, and occlusion detection, where, to the t-th task, corresponds a dataset, $\mathbb{D}^{(t)}$, composed of K_t M-view labeled instances, e.g., moving vehicles, as Equation (9) shows:

$$\mathbb{D}^{(t)} = \left\{ \left(\mathbf{x}^{(1,1,t)} \cdots, \mathbf{x}^{(1,M,t)}, \mathbf{y}^{(1,t)} \right), \cdots, \left(\mathbf{x}^{(K_t,1,t)} \cdots, \mathbf{x}^{(K_t,M,t)}, \mathbf{y}^{(K_t,t)} \right) \right\} \tag{9}$$

where $\mathbf{x}^{(k,m,t)}$ is the feature vector of the k-th instance over the m-th view and t-th task, belonging to the feature space $\mathcal{X}_m \subset \mathbb{R}^{I_m}$, i.e., $\mathbf{x}^{(k,m,t)} \in \mathcal{X}_m$, the M-tuple $\left(\mathbf{x}^{(k,1,t)}, \cdots, \mathbf{x}^{(k,M,t)} \right)$ is an element of the input data space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_M$, and $\mathbf{y}^{(k,t)}$ its corresponding true label in an output space, $\mathcal{Y}_t \subset \mathbb{R}^{O_t}$.

3.2.2. Task Functions

For the t-th task, we aim to learn a multi-view classification function, $f_t : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \rightarrow \mathcal{Y}_t$, that predicts, with high probability, the true label $\hat{\mathbf{y}}^{(k,t)}$ of the k-th instance, as Equation (10) shows, where f_t belongs to some hypothesis space, $\mathcal{H}_t$.

$$\hat{\mathbf{y}}^{(k,t)} = f_t \left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)} \right) \tag{10}$$

Consequently, the dimension O_t of the output space $\mathcal{Y}_t$ represents the number of classes in the t-th learning task.

3.2.3. The Parametric Model

Considering the high-dimension of the input data space, it seems reasonable to project multi-view data onto a low-dimensional latent space, $\mathcal{S}$, by learning some mapping $\boldsymbol{g} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, as Equation (11) shows:

$$\begin{aligned} &\mathfrak{z}^{(k)} = \boldsymbol{g}\left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)}\right) \\ \text{s.t.}\quad &\dim(\mathfrak{z}^{(k)}) \leq \dim(\mathcal{X}_1 \times \cdots \times \mathcal{X}_M) \end{aligned} \tag{11}$$

where $\mathfrak{z}^{(k)} \in \mathcal{S}$ is the projection of the k-th instance, and $\dim(\mathfrak{z}^{(k)})$ can be either unidimensional (e.g., J), or multidimensional (e.g., $J_1 \times \cdots \times J_P$). If we need a more efficient mapping, $\boldsymbol{g}$, a low-rank approximation function, $\hat{\boldsymbol{g}}$, is required instead of $\boldsymbol{g}$.

Let $h_t : \mathcal{S} \to \mathcal{Y}_t$ be the t-th task-specific mapping that predicts the label $\hat{\mathbf{y}}^{(k,t)}$ from the k-th instance $\mathfrak{z}^{(k)}$ embedded in the latent space $\mathcal{S}$, as shown in Equation (12), where h_t can be represented by, e.g., ANN, SVM, or random forest (RF) algorithms. In consequence, the function composition $h_t \circ \boldsymbol{g} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{Y}_t$ can determine the t-th task function f_t.

$$\hat{\mathbf{y}}^{(k,t)} = h_t(\mathfrak{z}^{(k)}) \tag{12}$$

3.2.4. The Optimization Problem

For a given multitask, multi-view dataset, $\{\mathbb{D}^{(1)}, \cdots, \mathbb{D}^{(T)}\}$, our problem can be reduced to learn simultaneously the set of functions $\{f_1, \cdots, f_T\}$ that minimizes the multi-objective empirical risk of Equation (13) [114]:

$$\begin{aligned} \min_{h_1,\cdots,h_T,g} \quad & \sum_{t=1}^{T} \frac{\lambda_t}{K_t} \sum_{k=1}^{K_t} \mathcal{L}_t\left(\left(\mathfrak{z}^{(k)}, \mathbf{y}^{(k,t)}\right), h_t\right) \\ \text{s.t.} \quad & \mathfrak{z}^{(k)} = \boldsymbol{g}\left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)}\right) \\ & \dim(\mathfrak{z}^{(k)}) \leq \dim(\mathcal{X}_1 \times \cdots \times \mathcal{X}_M) \end{aligned} \tag{13}$$

where $f_t = h_t \circ \boldsymbol{g}$ belongs to some hypothesis space $\mathcal{H}_t$, $\mathcal{L}_t : \mathcal{S} \times \mathcal{Y}_t \times \mathcal{H}_t \to \mathbb{R}_+$ is the loss function related to the t-th task that measures the discrepancy between the true label and the predicted one, and $\lambda_t \in \mathbb{R}_+$ is a weighting parameter, determined either statically or dynamically, which controls the relative importance of the t-th task.

3.2.5. Objectives

The main objectives are as follows:

1. For a multi-view input space of M views, to learn a mapping $\boldsymbol{g} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, where $\mathcal{S}$ is a low-dimensional latent tensor space with $\dim(\mathcal{S}) = J_1 \times \cdots \times J_M$ or J (see Section 5.1, particularly Equation (20)).
2. To reduce the computational complexity of $\boldsymbol{g}$, a low-rank approximation, $\hat{\boldsymbol{g}}$, needs to be learned.
3. For a set of T tasks, e.g., VTS tasks, the set of task-specific functions $h_1, \cdots, h_T$ must be learned, where $h_t : \mathcal{S} \to \mathcal{Y}_t$, and $\mathcal{Y}_t$ is the output space of the t-th task.
4. To evaluate the performance of our approach, a multitask, multi-view model for the case study of VTS systems (see Section 6.2) is employed.

4. Vehicle Traffic Surveillance System: Multitask, Multi-View Input Space Formation

In this section, we provide a general description of several tasks associated with a typical vision-based VTS system, including background and foreground segmentation, occlusion handling, and vehicle-size classification. Together, these tasks enable the estimation of traffic parameters, such as traffic density, vehicle count, and lane occupancy, inferred from the video. Specifically, these parameters are essential for high-level ITS applications.

4.1. Background and Foreground Segmentation

Let $\mathcal{V} \in \mathbb{Q}^{H \times W \times B \times N}$ be a fourth-order tensor representing a traffic surveillance video, recorded at a FPS frame rate with a duration of τ seconds. Here, $\mathbb{Q} = \{0, \cdots, 255\}$, W, and H represent the image spatial dimensions, corresponding to width and height, respectively, and B is the dimensionality of the image spectral coordinate system, i.e., the color space in which each pixel lives, or the number of spectral bands in hyper-spectral imaging (HSI). For example, $B = 1$ corresponds to grayscale, while $B = 3$ corresponds to RGB color space. Finally, $N = \tau \cdot FPS$ denotes the number of frames in the video.

From the aforementioned tensor $\mathcal{V}$, it is important to note the following:

1. The nth frontal slice $\mathcal{V}_{:::\mathrm{n}} \in \mathbb{Q}^{H \times W \times B}$ represents the nth frame of the video at time $t_n \forall n \in [N]$.
2. The third-mode fiber $\mathcal{V}_{\mathrm{ji:n}} \in \mathbb{Q}^{B}$ denotes the (i, j)th pixel value at frame n, where $(i, j) \in \mathbb{I}$ is the pixel location belonging to the image spatial domain $\mathbb{I} = [W] \times [H]$.
3. Each pixel value is quantized using D bits per spectral band. For simplicity, here, we assume the 8-bit grayscale color space $\mathbb{Q} = \{0, \cdots, 255\}$, i.e., $B = 1$, but it can be extended to other color spaces. Consequently, $\dim(\mathcal{V})$ reduces to $H \times W \times N$.
4. Every (i, j)th pixel value can be modeled as a discrete random variable, X_{ij}, with a probability mass function (pmf), denoted as $\mathbb{P}(X_{ij} = x)$, where $x \in \mathbb{Q}$.
5. For any observation time, $\tau_o < \tau$, the pmf of any pixel can be estimated, denoted as $\hat{\mathbb{P}}(X_{ij} = x)$.

Then, tensor $\mathcal{V}$ can be decomposed as Equation (14) shows and Figure 1 illustrates:

$$\mathcal{V} = \mathcal{B} \odot \bar{\mathcal{M}} + \mathcal{F} \tag{14}$$

where $\mathcal{B} \in \mathbb{Q}^{H \times W \times N}$ is called the background tensor, $\mathcal{F} \in \mathbb{Q}^{H \times W \times N}$ is the foreground tensor, and $\mathcal{M} \in \mathbb{B}^{H \times W \times N}$ is the binary mask of the foreground tensor, whose (j, i, n)-th entry $m_{\mathrm{jin}} = 1$ if the (i, j)th pixel value v_{jin} of $\mathcal{V}$ at frame n is part of the foreground tensor $\mathcal{F}$; otherwise $m_{\mathrm{jin}} = 0$, $\bar{\mathcal{M}} \in \mathbb{B}^{H \times W \times N}$ the complement of $\mathcal{M}$, and $\mathcal{F}$ can be obtained from the Hadamard product $\mathcal{V} \odot \mathcal{M}$.

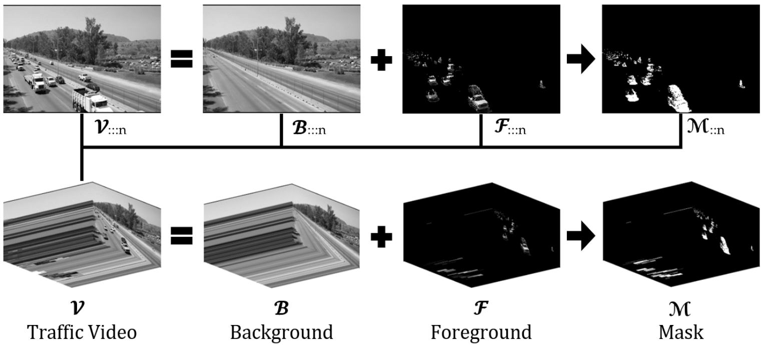

$\mathcal{V}_{:::\mathrm{n}}$
$\mathcal{B}_{:::\mathrm{n}}$
$\mathcal{F}_{:::\mathrm{n}}$
$\mathcal{M}_{::\mathrm{n}}$
$\mathcal{V}$
Traffic Video
$\mathcal{B}$
Background
$\mathcal{F}$
Foreground
$\mathcal{M}$
Mask

Figure 1. Illustration of the traffic surveillance video decomposition model.

4.2. Blob Formation

After decomposing $\mathcal{V}$ into the background and foreground tensors, various moving objects, including vehicles, pedestrians, and cyclists, can be extracted by analyzing $\mathcal{F}$ or its mask, $\mathcal{M}$. One such technique is called connected components analysis (CCA) [115,116]. CCA recursively searches at every nth frontal slice $\mathcal{M}_{::n}$ for connected pixel regions (see Definition 12), referred to in the literature as binary large objects (blobs), which can contain pixels associated with moving objects.

Definition 12 (Blob). *A blob, denoted as S, is a set of pixel locations connected by a specified connectivity criterion (e.g., four-connectivity or eight-connectivity* [117]*). Specifically, a pixel located at $(i, j) \in \mathbb{I}$ belongs to blob S if there exists another pixel location, $(i', j') \in S$, such that the connectivity criterion is met, as Equation (15) shows:*

$$S = \{(i, j) \in \mathbb{I} | \exists (i', j') \in S : (i', j') \neq (i, j) \wedge d_{IP}\big((i, j), (i', j')\big) = \delta\} \tag{15}$$

where $d_{IP} : \mathbb{I} \times \mathbb{I} \to \mathbb{R}$ is an inter-pixel distance that establishes the connectivity criterion given some threshold value, $\delta \in \mathbb{R}$, and $S \subseteq \mathbb{I}$.

For every blob $S^{(n)}$ detected at frame n, a blob mask, $\mathbf{S}^{(n)} \in \mathbb{B}^{H \times W}$, can be formed whose entries are given by Equation (16). Note that the pixel values of blob $S^{(n)}$ can be obtained from the product $(\boldsymbol{\mathfrak{F}}_{::n} \odot \mathbf{S}^{(n)}) \in \mathbb{Q}^{H \times W}$.

$$s_{ji}^{(n)} = \begin{cases} 1, & (i, j) \in S^{(n)} \\ 0, & otherwise \end{cases} \tag{16}$$

4.3. Vehicle Feature Extraction and Selection

Feature extraction can be considered a mapping, $\zeta : \mathbb{Q}^{H \times W} \to \mathcal{X}$, that transforms a given blob, S, into a low-dimensional point, $\mathbf{x} \in \mathcal{X}$, called the feature vector, as shown in Equation (17):

$$\mathbf{x} = \zeta(\boldsymbol{\mathfrak{F}}_{::n} \odot \mathbf{S}) \tag{17}$$

where $\mathcal{X}$, called the feature space, captures specific aspects of blobs $S^{(n)}$, e.g., color, shape, or texture.

The image moments (IMs) are a classical hand-crafted feature extractor that provides information about the spatial distribution, shape, and intensity of a blob image. Typical features extracted via the IM include centroid, area, orientation, and eccentricity. Formally, the (p, q)-th raw IM for blob S is given by the bilinear map of Equation (18):

$$x_{pq} = \sum_{j=1}^{W} \sum_{i=1}^{H} j^p i^q b_{ij} = \mathbf{B} \times_1 \boldsymbol{\zeta} \times_2 \boldsymbol{\eta} \tag{18}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^H$, $\boldsymbol{\eta} \in \mathbb{R}^W$ are vectors whose i-th and j-th entries are $\zeta_i = i^p$ and $\eta_j = j^q$, respectively.

4.4. Vehicle Occlusion Task

Assuming there are V_n vehicles on the road at the n-th frame, each associated with a specific blob $S^{(n,v)}$, let $\mathcal{B}^{(n)}$ denote the set of these blobs, and let $\tilde{\mathcal{B}}^{(n)} = \left\{\tilde{S}^{(n,u)}\right\}_{u=1}^{U_n}$ be the set of blobs detected via CCA in the n-th frame, where $U_n \leq V_n$. The v-th vehicle, with blob $S^{(n,v)} \in \mathcal{B}^{(n)}$, is occluded by the u-th detected blob $\tilde{S}^{(n,v)} \in \tilde{\mathcal{B}}^{(n)}$ if any of the conditions in Equation (19) are met.

$$\begin{cases} S^{(n,v_1)} \cap \tilde{S}^{(n,v_2)} \neq \emptyset & S^{(n,v)} \text{ and } \tilde{S}^{(n,u)} \text{ are occluded} \\ S^{(n,v)} \cap \tilde{S}^{(n,u)} = \tilde{S}^{(n,u)} & S^{(n,u)} \text{ is totally occluded by } \tilde{S}^{(n,v)} \\ (S^{(n,v)} \cap \tilde{S}^{(n,u)} \neq \emptyset) \wedge (S^{(n,v)} \cap \tilde{S}^{(n,u)} \neq S^{(n,v)}) & S^{(n,v)} \text{ is partially occluded} \end{cases} \tag{19}$$

Given the set of detected blobs $\tilde{\mathcal{B}}^{(n)}$ in the n-th frame, the vehicle occlusion detection aims to predict, with high probability, a set of $W_n \leq U_n$ blobs $\hat{\mathcal{B}}^{(n)} = \{\hat{S}^{(n,1)}, \cdots, \hat{S}^{(n,W_n)}\}$, each containing more than one vehicle. To achieve this, an occlusion feature space, $\mathcal{X}_1 \subset \mathbb{R}^{I_1}$, is constructed using a feature extraction mapping $\zeta_1 : \mathbb{Q}^{W \times H} \to \mathcal{X}_1$ to capture the vehicle occlusion patterns. In this space, every detected blob, $\tilde{S}^{(n,u)}$, is represented by an I_1-dimensional feature vector $\mathbf{x}^{(n,u)} \in \mathcal{X}_1$. Assuming occlusions are only composed of partially observed vehicles, a classification function, $f_1 : \mathcal{X}_1 \to \{0, 1\}$, can be built to predict whether a detected blob, $\tilde{S}^{(n,u)}$, has more than one vehicle.

4.5. Vehicle Classification Task

Given a set of vehicle-size labels (e.g., small, midsize, large) represented in a vector space, $\mathcal{Y}_2 \subset \mathbb{R}^{O_2}$, called the output space, the vehicle classification task aims to predict, with high probability, the true label $\mathbf{y}^{(n,u)} \in \mathcal{Y}_2$ for an unseen vehicle blob instance, $\tilde{S}^{(n,u)} \in \tilde{\mathcal{B}}^{(n)}$, at frame n. First, each blob, $\tilde{S}^{(n,u)}$, is mapped into some feature space, $\mathcal{X}_2 \subset \mathbb{R}^{I_2}$, using a feature extraction mapping, $\zeta_2 : \mathbb{Q}^{W \times H} \to \mathcal{X}_2$, constructed to explain the vehicle-size patterns. From this space, a feature vector, $\mathbf{x}^{(n,u)}$, associated with $\tilde{S}^{(n,u)}$, is derived. Then, a classification function, $f_2 : \mathcal{X}_2 \to \mathcal{Y}_2$, can be built to predict the label of a vehicle blob instance, $\tilde{S}^{(n,w)}$.

5. A Multi-View Data Tensor Fusion Layer and the Connection Between the Einstein and Hadamard Products

In this section, the concept of a multi-view data tensor fusion (MV-DTF) layer and its connection with Einstein and Hadamard products are introduced. Basically, MV-DTF is a form of an FC layer for multi-view data; i.e., it is an affine function, but instead of using a linear map, our layer employs a multilinear map to encode the interactions across views. Additionally, a low-rank approximation for the MV-DTF layer is also proposed to reduce its computational complexity.

5.1. Multi-View Tensor Data Fusion Layer: The Mapping g as an Einstein Product

Inspired by previous works [36,75,76], we restrict the function space of the MV-DTF layer to the affine functions characterized by a multilinear map, $\mathcal{T} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, followed by a translation and, possibly, a non-linear map, σ, as Equation (20) shows:

$$\mathcal{Z}^{(k)} = g\left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)}\right) = \sigma\left(\mathcal{T}\left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)}\right) + \mathcal{B}\right) \tag{20}$$

where g is the MV-DTF layer, the P-order tensor $\mathcal{Z}^{(k)} \in \mathcal{S}$ is the projection of the k-th instance $(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)})$ onto the latent tensor space $\mathcal{S}$, called the fused tensor, with dimension $J_1 \times \cdots \times J_P$, $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_P}$ is the translational term, formally called bias, and the mapping $\sigma : \mathcal{S} \to \mathbb{R}^{J_1 \times \cdots \times J_P}$.

Definition 13 specifies how a multilinear map can be represented using coordinate systems, and from this representation, a tensor can be induced for every multilinear map.

Definition 13 (Coordinate representation of a multilinear map [101])**.** *Let $V_1, \ldots, V_M$, and W be real vector spaces, where $\dim(V_m) = I_m$ for all $m \in [M]$, and $\dim(W) = J$. Let $\{\mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(J)}\}$ be the standard basis for W. And let $\mathcal{T} : V_1 \times \cdots \times V_M \to W$ be a multilinear map. Given an ordered M-tuple $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}) \in V_1 \times \cdots \times V_M$, where $\mathbf{x}^{(m)} \in V_m$, the map $\mathcal{T}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)})$ is completely determined by a linear combination of basis vectors $\mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(J)}$ and scalars $\{\alpha_{ji_1 \cdots i_M} \in \mathbb{R} | i_1 \in [I_1], \cdots, i_M \in [I_M], j \in [J]\}$, as Equation (21) shows.*

$$\mathcal{T}\left(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}\right) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} \sum_{j=1}^{J} x^{(1)}_{i_1} \cdots x^{(M)}_{i_M} \alpha_{ji_1 \cdots i_M} \mathbf{e}^{(j)} \tag{21}$$

The collection of scalars can then be arranged into an $(M+1)$th-order tensor, denoted as $\mathcal{A} \in \mathbb{R}^{J \times I_1 \times \cdots \times I_M}$, which determines $\mathcal{T}$, and whose $(j, i_1, \cdots, i_M)$-th entry $a_{ji_1 \cdots i_M}$ corresponds with $\alpha_{ji_1 \cdots i_M}$.

Next, Definition 14 establishes a connection between the Einstein product and multilinear maps via the universal property of multilinear maps (see Definition 2).

Definition 14. *Let $\mathbf{x}^{(1)},\cdots,\mathbf{x}^{(M)}$ be a set of vectors, where $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$ for all $m \in [M]$. And let $\mathcal{T}: \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{J}$ be the multilinear map induced via the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J\times I_1\times\cdots\times I_M}$, and $\pi : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_M}$ is the multilinear map associated with the tensor product $\mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_M}$. For $\boldsymbol{\mathcal{X}} = \pi(\mathbf{x}^{(1)},\cdots,\mathbf{x}^{(M)}) \in \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_M}$, the Einstein product $\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}$ can be understood as a linear map, $\boldsymbol{\Phi} : \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_M} \to \mathbb{R}^{J}$. Then, $\mathcal{T}$ and $\boldsymbol{\Phi}$ are related by the universal property of multilinear maps, as Equation (22) shows.*

$$\mathcal{T}(\mathbf{x}^{(1)},\cdots,\mathbf{x}^{(M)}) = (\boldsymbol{\Phi} \circ \pi)(\mathbf{x}^{(1)},\cdots,\mathbf{x}^{(M)}) = \boldsymbol{\Phi}(\boldsymbol{\mathcal{X}}) = \boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}} \tag{22}$$

For the multilinear map $\mathcal{T} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$ in Equation (20), Definition 13 ensures the existence of a tensor, $\mathcal{A} \in \mathbb{R}^{J_1\times\cdots\times J_P\times I_1\times\cdots\times I_M}$, that determines $\mathcal{T}$, and Definition 14 provides the associated linear map $\boldsymbol{\Phi} : \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_M} \to \mathcal{S}$ of $\mathcal{T}$. From the above definitions, Equation (20) can be rewritten in tensor form as Equation (23) shows, where $\boldsymbol{\mathcal{X}}^{(k)} = \mathbf{x}^{(k,1)} \otimes \cdots \otimes \mathbf{x}^{(k,M)} \in \mathbb{R}^{I_1\times\cdots\times I_M}$ is a rank-one tensor resulted from the tensor product of the M view vectors associated with the k-th instance.

$$\boldsymbol{\mathcal{Z}}^{(k)} = g(\mathbf{x}^{(k,1)},\cdots,\mathbf{x}^{(k,M)}) = \sigma\Big(\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}^{(k)} + \boldsymbol{\mathcal{B}}\Big) \tag{23}$$

Note that Equation (23) represents a differentiable expression with respect to tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$. Consequently, their values can be learned using optimization algorithms such as stochastic gradient descent (SGD), where the number of parameters to learn, denoted as L, corresponds with the number of entries of tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, as Equation (24) shows. Note that L scales exponentially with the number of views, M, and the order P of $\mathcal{S}$. Specifically, for $I_m = J_p = I\, \forall m \in [M], p \in [P]$, L is reduced to $L = I^P(1 + I^M) \simeq I^{M+P}$. This exponential growth can lead to computational challenges while increasing the risk of overfitting due to the induced curse of dimensionality [118–120]; i.e., the number of samples needed to train a model grows exponentially with its dimension.

$$L = \left(\prod_{m=1}^{M} I_m\right) \cdot \left(\prod_{p=1}^{P} J_p\right) + \prod_{p=1}^{P} J_p = \prod_{p=1}^{P} J_p \left(\prod_{m=1}^{M} I_m + 1\right) \tag{24}$$

5.2. Hadamard Products of Einstein Products and Low-Rank Approximation Mapping $\hat{g}$

Low-rank approximation is a well-known technique that not only allows for reducing model parameter storage requirements but also helps in alleviating the computational burden of neural network models [81,82,85–89,121]. Based on these facts, in this work, we explore a CPD-based low-rank structure, illustrated in Figure 2, to overcome the curse of dimensionality induced via the MV-DTF layer. This structure helps reduce the number of parameters required for the MV-DTF layer, and it is computationally more efficient (see Proposition 1). But before presenting this structure, the concept of Hadamard factor tensors is first introduced in Definition 15.

Definition 15 (Hadamard factor tensors). *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J_1\times\cdots\times J_P\times I_1\times\cdots\times I_M}$ be a $(P+M)$-order tensor, whose $(j_1,\cdots,j_P)$-th subtensor results from fixing every index but the last M modes; i.e., $\boldsymbol{\mathcal{A}}_{j_1\cdots j_P:\cdots:} \in \mathbb{R}^{I_1\times\cdots\times I_M}$ for all $j_p \in [J_p]$ and $p \in [P]$ can be approximated as a rank-$R^{(j_1,\cdots,j_P)}$ tensor using the CPD, as Equation (25) shows:*

$$\boldsymbol{\mathcal{A}}_{j_1\cdots j_P:\cdots:} \approxeq \sum_{r=1}^{R^{(j_1,\cdots,j_P)}} \mathbf{v}^{(j_1,\cdots,j_P,r,1)} \otimes \cdots \otimes \mathbf{v}^{(j_1,\cdots,j_P,r,M)} \tag{25}$$

where the number of subtensors in $\boldsymbol{\mathcal{A}}$ corresponds to the dimension of the latent space $\mathcal{S}$; i.e., $J_1 \times \cdots \times J_P$, each $(j_1,\cdots,j_P)$-th subtensor $\boldsymbol{\mathcal{A}}j_1 \cdots j_P : \cdots :$ has a specific rank, $R^{(j_1,\cdots,j_P)} \in \mathbb{N}$, which can be different across subtensors, and for $\mathbf{v}^{(j_1,\cdots,j_P,r,m)} \in \mathbb{R}^{I_m}$, known as the m-mode factor vector, the superscripts $j_1,\cdots,j_P$ identify the $(j_1,\cdots,j_P)$-th subtensor to which it corresponds, $r \in [R^{(j_1,\cdots,j_P)}]$ identifies its associated r-th rank-one tensor in the CPD, and $m \in [M]$ its mode.

Then, the set of factor vectors along the m-mode can be rearranged into a $(P+2)$-order tensor, $\mathfrak{U}^{(m)} \in \mathbb{R}^{J_1\times\cdots\times J_P\times R\times I_m}$, here called the m-mode Hadamard factor tensor, whose $(P+2)$-mode fibers $\mathfrak{U}^{(m)}_{j_1\cdots j_P r:} \in \mathbb{R}^{I_m}$ are given by Equation (26):

$$\mathfrak{U}^{(m)}_{j_1\cdots j_P r:} = \begin{cases} \mathbf{v}^{(j_1,\cdots,j_P,r,m)}, & r \le R^{(j_1,\cdots,j_P)} \\ \mathbf{0}, & r > R^{(j_1,\cdots,j_P)} \end{cases} \tag{26}$$

where $\mathbf{0} \in \mathbb{R}^{I_m}$ is the zero vector, and $R = \max R^{(j_1,\cdots,j_P)}$ is the maximum rank across subtensors, employed to avoid inconsistencies due to different rank values between subtensors.

Figure 2 illustrates the concept of Hadamard-factor tensors for the multilinear map $\mathcal{T}: \mathbb{R}^5 \times \mathbb{R}^3 \to \mathbb{R}^3$ with associated tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{3\times5\times3}$. Here, there is a two-view data $(M = 2)$ with dimensions $I_1 = 5$, and $I_2 = 3$, respectively; the order and dimension of the latent tensor space are $P = 1$ and $J_1 = J = 3$, and hence, there are three subtensors, $\boldsymbol{\mathcal{A}}_{1::}, \boldsymbol{\mathcal{A}}_{2::}, \boldsymbol{\mathcal{A}}_{3::} \in \mathbb{R}^{5\times3}$, associated with the tensor $\boldsymbol{\mathcal{A}}$. For subtensor $\boldsymbol{\mathcal{A}}_{1::}$, its rank is $R^{(1)} = 3$; hence, $\boldsymbol{\mathcal{A}}_{1::} = \mathbf{v}^{(1,1,1)} \otimes \mathbf{v}^{(1,1,2)} + \mathbf{v}^{(1,2,1)} \otimes \mathbf{v}^{(1,2,2)} + \mathbf{v}^{(1,3,1)} \otimes \mathbf{v}^{(1,3,2)}$ for subtensor $\boldsymbol{\mathcal{A}}_{2::}$, $R^{(2)} = 1$, i.e., $\boldsymbol{\mathcal{A}}_{2::} = \mathbf{v}^{(2,1,1)} \otimes \mathbf{v}^{(2,1,2)}$, while for subtensor $\boldsymbol{\mathcal{A}}_{3::}$, $R^{(3)} = 2$, and $\boldsymbol{\mathcal{A}}_{3::} = \mathbf{v}^{(3,1,1)} \otimes \mathbf{v}^{(3,1,2)} + \mathbf{v}^{(3,2,1)} \otimes \mathbf{v}^{(3,2,2)}$. From these vectors, two Hadamard factor tensors, $\mathfrak{U}^{(1)} \in \mathbb{R}^{3\times3\times5}$ and $\mathfrak{U}^{(2)} \in \mathbb{R}^{3\times3\times3}$, can be constructed, corresponding to the first and second views, respectively. The second-mode dimension of these tensors corresponds to the greatest subtensor rank, i.e., $R = \max R^{(j)} = R^{(1)} = 3$, to avoid heterogeneous rank values across subtensors. Hence, the second and third subtensors incorporate two and one additional zero vectors, respectively, as Figure 2 shows.

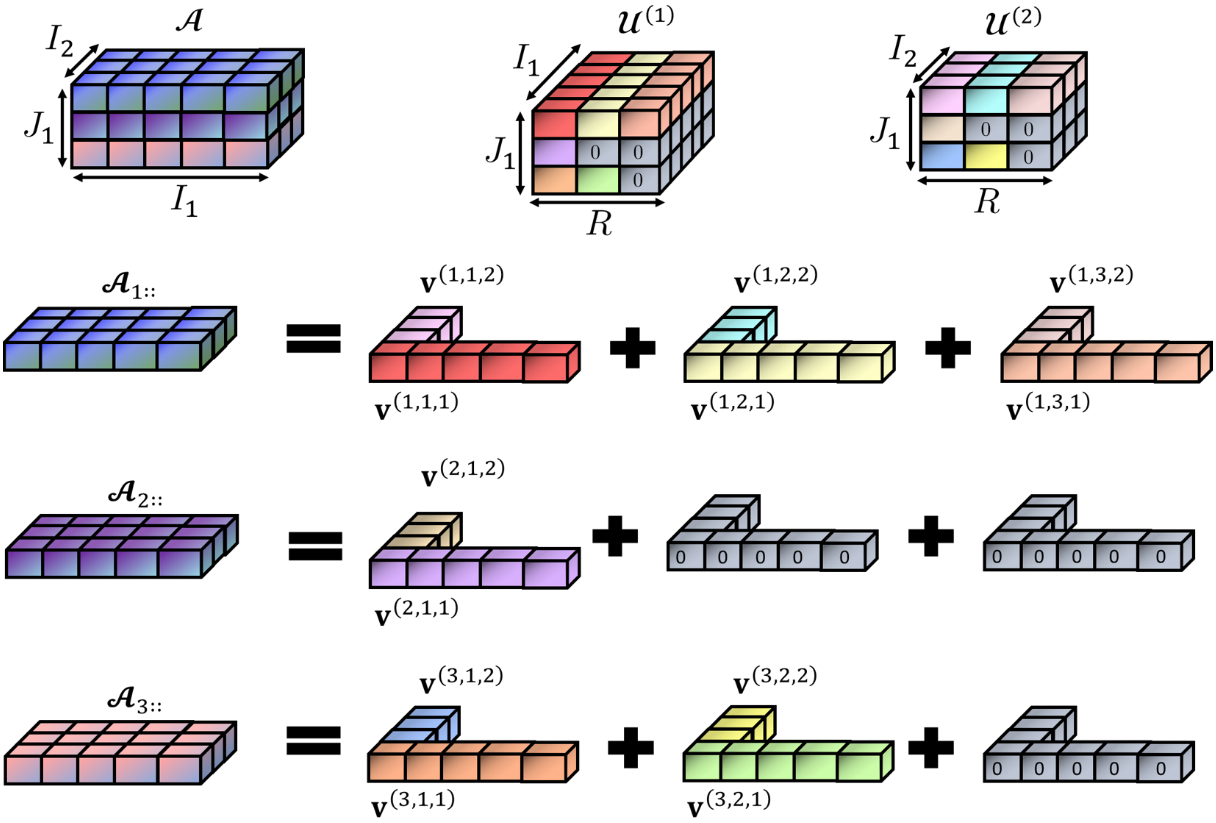


Figure 2. Illustration of subtensors of $\boldsymbol{\mathcal{A}}$ and the Hadamard factor tensors $\mathfrak{U}^{(1)} \in \mathbb{R}^{J\times R\times I_1}$, and $\mathfrak{U}^{(2)} \in \mathbb{R}^{J\times R\times I_2}$ for the multilinear map $\mathcal{T}: \mathbb{R}^{I_1} \times \mathbb{R}^{I_2} \to \mathbb{R}^{J}$, where $I_1 = 5$, $I_2 = 3$, $J = 3$, and $R = 3$.

Proposition 1 presents the primary result of this work, i.e., the mathematical relationship between Einstein and Hadamard products. To the best of our knowledge, this relationship is not known.

Proposition 1. *Let $\mathcal{X} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ be a rank-one tensor, where $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$ for all $m \in [M]$. And let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J_1 \times \cdots \times J_P \times I_1 \times \cdots \times I_M}$ be a $(P+M)$-order tensor induced via the multilinear map $\mathcal{T} : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{J_1 \times \cdots \times J_P}$, which can be decomposed into a set of M factor tensors $\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(M)}$ for a given rank, $R \leq \max \operatorname{rank}(\boldsymbol{\mathcal{A}}_{j_1 \cdots j_P : \cdots :})$, where $\mathcal{U}^{(m)} \in \mathbb{R}^{J_1 \times \cdots \times J_P \times R \times I_m}$ for all $m \in [M]$. Then, $\mathcal{T}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)})$ can be approximated by a sum of R Hadamard products of Einstein products, as Equation (27) shows:*

$$\mathcal{T}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}) = \boldsymbol{\mathcal{A}} \circledast_M \mathcal{X} \approxeq \sum_{r=1}^{R} \bigodot_{m=1}^{M} \left(\mathcal{U}^{(m)}_{: \cdots : r :} \circledast_1 \mathbf{x}^{(m)} \right) = \left[\bigodot_{m=1}^{M} \left(\mathcal{U}^{(m)} \circledast_1 \mathbf{x}^{(m)} \right) \right] \circledast_1 \mathbf{1}^R \tag{27}$$

where $\mathcal{U}^{(m)} \in \mathbb{R}^{J_1 \times \cdots \times J_P \times R \times I_m}$ is the m-mode Hadamard factor tensor. A vector of all ones is denoted as $\mathbf{1}^R \in \mathbb{R}^R$. And $\bigodot_{m=1}^{M} \left(\mathcal{U}^{(m)} \circledast_1 \mathbf{x}^{(m)} \right) = \left(\mathcal{U}^{(1)} \circledast_1 \mathbf{x}^{(1)} \right) \odot \cdots \odot \left(\mathcal{U}^{(M)} \circledast_1 \mathbf{x}^{(M)} \right)$.

Proof. In Appendix A. □

By leveraging Proposition 1 for tensor $\boldsymbol{\mathcal{A}}$ in Equation (23), the MV-DTF layer g can be approximated through a more efficient low-rank mapping $\hat{g} : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{J_1 \times \cdots \times J_P}$, called the low-rank multi-view data tensor fusion (LRMV-DTF) layer, defined in Equation (28), where the m-mode factor tensor $\mathcal{U}^{(m)} \in \mathbb{R}^{J_1 \times \cdots \times J_M \times I_m \times R}$, associated with the m-th view, contributes to building every k-th fused tensor $\mathcal{Z}^{(k)}$.

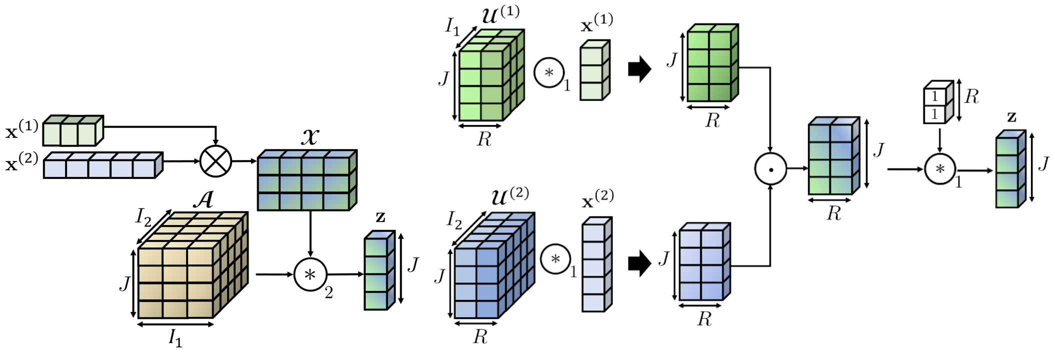
$$\mathcal{Z}^{(k)} \approxeq \hat{g}\left(\mathbf{x}^{(k,1)}, \cdots, \mathbf{x}^{(k,M)} \right) = \sigma\left(\left[\bigodot_{m=1}^{M} \left(\mathcal{U}^{(m)} \circledast_1 \mathbf{x}^{(k,m)} \right) \right] \circledast_1 \mathbf{1} + \mathcal{B} \right) \tag{28}$$

From this approximation, the number of parameters required for the LRMV-DTF layer, denoted as $\hat{L}$, is provided in Equation (29). Note that the product of the I_m-dimensions related to the views in L (Equation (24)) has been replaced with a summation, which yields fewer parameters to learn compared to those in the MV-DTF layer, reducing the risk of overfitting.

$$\hat{L} = \prod_{p=1}^{P} J_p \cdot \left(R \sum_{m=1}^{M} I_m + 1 \right) \tag{29}$$

An illustration of our layers is shown in Figure 3a (MV-DTF), and Figure 3b (LRMV-DTF). Here, the number of views $M = 2$, and their dimensions $I_1 = 3$, and $I_2 = 5$ respectively. The order of the latent space is $P = 1$, and its dimension $\dim(S) = J = 4$. Consequently, the multilinear map is $\mathcal{T} : \mathbb{R}^3 \times \mathbb{R}^5 \to \mathbb{R}^4$, with associated tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{4 \times 3 \times 5}$, and bias $\mathbf{b} \in \mathbb{R}^4$. However, vector $\mathbf{b}$ is fixed to zero $\mathbf{0} \in \mathbb{R}^4$ for simplicity. For low-rank approximation, the rank of the (j)-th subtensor is $R^{(j)} = R = 2\, \forall j \in [4]$. Hence, according to Definition 15, tensor $\boldsymbol{\mathcal{A}}$ can be decomposed into two factor tensors: $\mathcal{U}^{(1)} \in \mathbb{R}^{4 \times 2 \times 3}$, and $\mathcal{U}^{(2)} \in \mathbb{R}^{4 \times 2 \times 5}$, associated with the first and second views, respectively.

This relationship between the Einstein and Hadamard product enables a rank-R CPD for every subtensor $(j_1, \cdots, j_P)$ of tensor $\boldsymbol{\mathcal{A}}$ and, consequently, a low-rank approximation.



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

Figure 3. Illustration of the MV-DTF and LRMV-DTF layers. (**a**) MV-DTF layer $\mathbf{z} = \boldsymbol{\mathcal{A}} \circledast_1 \boldsymbol{\mathfrak{X}}$, where $\boldsymbol{\mathfrak{X}} = \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}$. (**b**) LRMV-DTF layer $\mathbf{z} = ((\boldsymbol{\mathfrak{U}}^{(1)} \circledast_1 \mathbf{x}^{(1)}) \odot (\boldsymbol{\mathfrak{U}}^{(2)} \circledast_1 \mathbf{x}^{(2)})) \circledast_1 \mathbf{1}^R$.

5.3. Dimension J or $J_1 \times \cdots \times J_P$, Order P of the Latent Space $\mathcal{S}$, and the Rank R: The Hyperparameters of the MV-DTF and LRMV-DTF Layers

The proposed layers introduce three hyperparameters to tune: the order P and the dimension $J_1 \times \cdots \times J_P$ of $\mathcal{S}$, and the rank value R:

1. **Latent space dimension**: It determines the expressiveness of the latent space to capture complex patterns across views. High-dimensional spaces enhance expressiveness but also increase the risk of overfitting, while low-dimensional spaces reduce expressiveness but mitigate the risk of overfitting.
2. **Latent space order**: It is determined by the architecture of the ANN. For instance, in multi-layer perceptron (MLP) architectures, the input space dimension is unidimensional, e.g., $\dim(\mathcal{S}) = J$; hence, $P = 1$. In contrast, the input space of CNNs is multidimensional; thereby, $P \geq 2$.
3. **Rank**: It determines the computational complexity of the MV-DTF layer. For low rank values on subtensors $\boldsymbol{\mathcal{A}}_{j_1 \cdots j_P : \cdots :}$ of $\boldsymbol{\mathcal{A}}$, the number of parameters to learn can be reduced, but it may not capture complex interactions across views effectively, limiting the model performance. Conversely, high rank values increase the capacity to learn complex patterns in data, but they may lead to overfitting.

5.4. MV-DTF and LRMV-DTF on Neural Network Architectures: The Mapping Set $\{h_1, \cdots, h_T\}$

According to the desired level of fusion [16], two primary configurations can be employed where our data fusion layer can be incorporated in an ANN architecture:

1. **Feature extraction**: The MV-DTF layer g can be integrated into an ANN to map the multi-view input space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_M$ into some latent space, $\mathcal{S}$, for multi-view feature extraction; see Figure 4a,c. Here, both the order P and dimension $J_1 \times \cdots \times J_P$ of $\mathcal{S}$ must correspond with the order and dimension of the input layer in the architecture of the ANN.
2. **Multilinear regression**: The MV-DTF layer performs multilinear regression to capture the multilinear relationships between the multi-view latent space $\mathcal{U}_1 \times \cdots \times \mathcal{U}_M$ and the output space $\mathcal{Y}$ for single-task learning (see Figure 4b). Here, $\mathcal{U}_m$ is the m-th single-view latent space obtained from the mapping $\zeta_m : \mathcal{X}_m \to \mathcal{U}_m$, where $\mathcal{X}_m$ is the m-th single-view input space. Consequently, the dimension and order of the latent space must correspond with those of the output space.

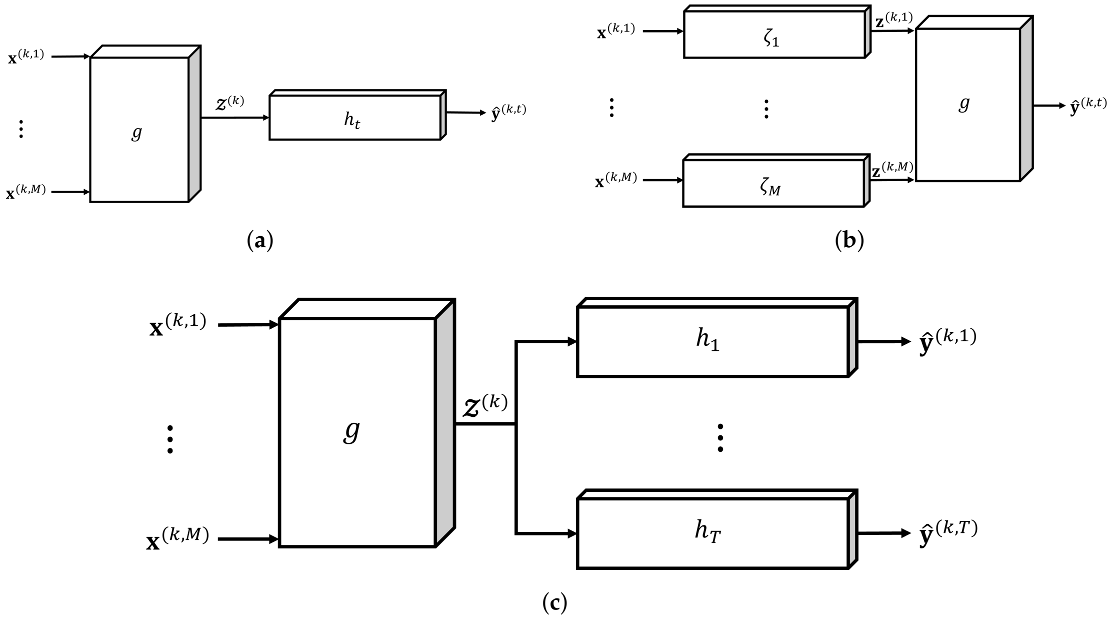


Figure 4. Primary configurations for incorporating the MV-DTF layer in neural network architectures. (**a**) MV-DTF layer for multi-view feature extraction on single-task learning, where $g : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{J_1 \times \cdots \times J_P}$, and $h : \mathbb{R}^{J_1 \times \cdots \times J_P} \to \mathbb{R}^{O_t}$. (**b**) MV-DTF layer for multilinear regression on single-task learning, where $\zeta_m : \mathbb{R}^{I_m} \to \mathbb{R}^{H_m}$, and $g : \mathbb{R}^{H_1} \times \cdots \times \mathbb{R}^{H_M} \to \mathbb{R}^{O_t}$. (**c**) MV-DTF for multi-view feature extraction on multitask learning, where $g : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^{J_1 \times \cdots \times J_P}$, and $h_t : \mathbb{R}^{J_1 \times \cdots \times J_P} \to \mathbb{R}^{O_t}$.

6. Results and Discussion

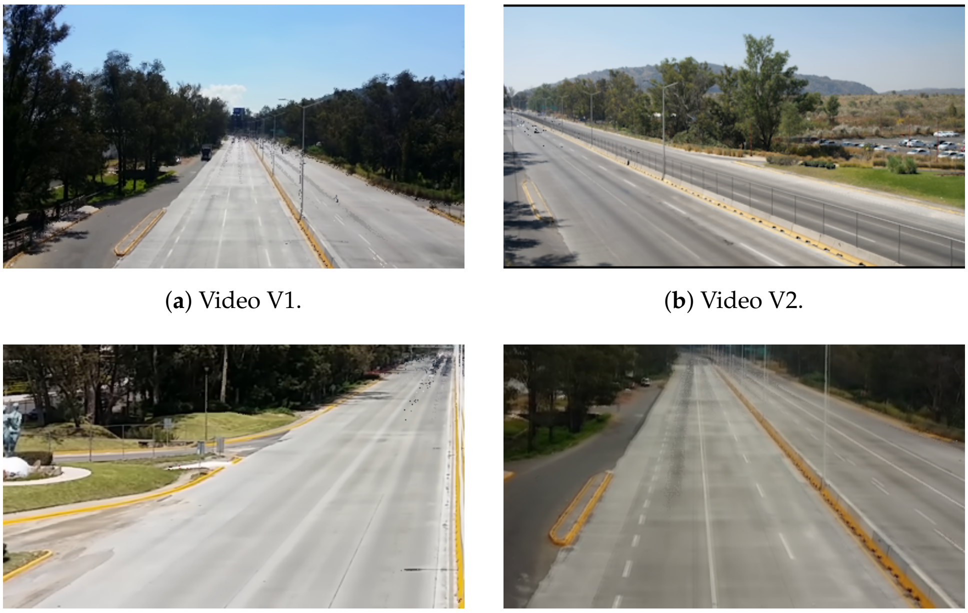
6.1. Dataset Description

To test the effectiveness of the proposed MV-DTF layer, we conduct experiments on four real-world traffic surveillance videos, encompassing more than 50,000 frames of footage with a resolution of 420 × 240 pixels and recorded at a frame rate of 25 FPS (accessible via [122]). Sample images from each test video can be observed in Figure 5, while technical details are provided in Table 3.

A collection of over 92,000 images of vehicles was then extracted from the test videos using the background and foreground method. Each image has been manually labeled for two tasks ($T = 2$): (1) occlusion detection, where vehicles are categorized as occluded or non-occluded (labeled to as 1 and 0, respectively), and (2) vehicle-size classification, where non-occluded vehicles are categorized as small (S), midsize (M), large (L), or very large (XL), with labels 1 to 4. Next, one-hot encoding was used to represent the class labels of each task. Consequently, the output spaces for the classification and occlusion detection tasks become $\mathcal{Y}_1 \subset \mathbb{B}^4$, and $\mathcal{Y}_2 \subset \mathbb{B}^2$, respectively, i.e., $O_1 = 4$ and $O_2 = 2$.

In addition, three subsets of image moment-based features were extracted and normalized for each vehicle image: (1) a 4D feature space, $\mathcal{X}_1 \subset \mathbb{R}^4$ (i.e., $I_1 = 4$), consisting of the vehicle blob solidity, orientation, eccentricity, and compactness features; (2) a 3D feature space, $\mathcal{X}_2 \subset \mathbb{R}^3$ (i.e., $I_2 = 3$), encompassing the vehicle's width, area, and aspect ratio; and (3) a 2D feature space, $\mathcal{X}_3 \subset \mathbb{R}^2$, representing the vehicle centroid coordinates. Together, the three feature spaces form a three-view input space $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$ of dimension $4 \times 3 \times 2$; i.e., the number of views is $M = 3$, and $I_1 = 4$, $I_2 = 3$, and $I_3 = 2$ are the dimensions of each feature space.

As a result, two datasets, denoted as $\mathbb{D}^{(1)}$ and $\mathbb{D}^{(2)}$, were created from the test videos, where $\mathbb{D}^{(1)}$ corresponds to the occlusion detection task and $\mathbb{D}^{(2)}$ to the vehicle-size classification task. Both datasets encompass vehicle instances represented in a three-view feature space, available in [123]. Tables 4 and 5 provide a summary of our datasets, detailing the distribution of images across the occlusion and vehicle-size classification tasks.



(**a**) Video V1.

(**b**) Video V2.

(**c**) Video V3.

(**d**) Video V4.

Figure 5. Test videos employed for vehicle-size classification and occlusion detection tasks. They were recorded at different dates and views (click on each Figure to link to the video).

Table 3. Technical details of the test videos.

Video	Duration (s)	Tracked Vehicles	Temporal Samples of Tracked Vehicles
V1	146	137	6132
V2	326	333	19,194
V3	216	239	14,457
V4	677	720	91,870

Table 4. Description of the vehicle instance dataset for the occlusion detection task.

Video	Occluded	Unoccluded
V1	4671	1461
V2	12,684	6510
V3	10,384	4073
V4	41,002	11,084

Table 5. Description of the vehicle instance dataset for the vehicle-size classification task.

Video	Small (Class 1)	Midsize (Class 2)	Large (Class 3)	Very Large (Class 4)
V1	45	4018	390	218
V2	169	11,687	676	152
V3	206	8426	1179	573
V4	777	35,843	3101	1282

6.2. The Multitask, Multi-View Model Architecture and Training

6.2.1. The Multitask, Multi-View Model Architecture

To learn the two tasks, a multitask, multi-view ANN model based on the MLP architecture was employed. The proposed model is structured in four main stages (see Figure 6): (1) hand-crafted feature extractors (shown in red), (2) an MV-DTF/LRMV-DTF layer (in green), (3) the neck (in yellow), and (4) the task-specific heads (in blue). Stages 1 and 2 form the backbone of the model, serving as a feature extractor to capture both low-level and high-level features from the raw data. Stage 3 refines the features extracted from the backbone. Finally, stage 4 performs prediction or inference. In addition, dropout is applied at the end of each stage to reduce the risk of overfitting and enhance the model's generalization.

The MV-DTF/LRMV-DTF layer provides the mapping $T : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to \mathcal{S}$, where $\dim(\mathcal{X}_1) = 4$, $\dim(\mathcal{X}_2) = 3$, and $\dim(\mathcal{X}_3) = 2$. The order P of the latent space $\mathcal{S}$ is fixed to one, i.e., $P = 1$ without loss of generality, which simplifies its dimension $J_1 \times \cdots \times J_P$ to J, a hyperparameter to tune. Therefore, the parameters of the MV-DTF/LRMV-DTF layer are either the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J\times 4\times 3\times 2}$, or the associated Hadamard factor tensors $\mathfrak{U}^{(1)} \in \mathbb{R}^{J\times R\times 4}, \mathfrak{U}^{(2)} \in \mathbb{R}^{J\times R\times 3}$, and $\mathfrak{U}^{(3)} \in \mathbb{R}^{J\times R\times 2}$, where the rank R is a hyperparameter to tune, along with the bias tensor $\mathfrak{B} \in \mathbb{R}^{J}$. Consequently, Equation (28) is reduced to Equation (30) (Equation (30) holds when $R^{(j)}$ is the tensor decomposition rank for all $j \in [J]$):

$$\mathbf{z}^{(k)} = \sigma(\boldsymbol{\mathcal{A}} \circledast_3 \mathfrak{X}^{(k)} + \mathbf{b}) \approxeq \sigma\left(\bigodot_{m=1}^{3} \left(\mathfrak{U}^{(m)} \circledast_1 \mathbf{x}^{(k,m)} \right) \circledast_1 \mathbf{1} + \mathbf{b} \right) \tag{30}$$

where the fused tensor $\mathfrak{Z}^{(k)}$ and bias $\mathfrak{B}$ are transformed to the vectors $\mathbf{z}^{(k)} \in \mathbb{R}^{J}$ and $\mathbf{b} \in \mathbb{R}^{J}$, respectively, and $\mathfrak{X}^{(k)} = \mathbf{x}^{(k,1)} \otimes \mathbf{x}^{(k,2)} \otimes \mathbf{x}^{(k,3)} \in \mathbb{R}^{4\times 3\times 2}$.

To solve this problem (see Section 3), the multi-objective optimization defined in Equation (13) is employed with $T = 2$ and $M = 3$, where h_1 and h_2 are the task-specific occlusion detection and vehicle classification functions, $g : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to \mathcal{S}$ can be either the MV-DTF or LRMV-DTF layer, $\mathcal{L}_1$ and $\mathcal{L}_2$ are the binary cross-entropy and multiclass cross-entropy loss (see Definitions 16 and 17, respectively) for the above tasks, and the task importance weighting hyperparameters $\lambda_1 = 0.4$ and $\lambda_2 = 0.6$ were selected from a finite set of values through cross-validation, a technique often employed by other authors [69].

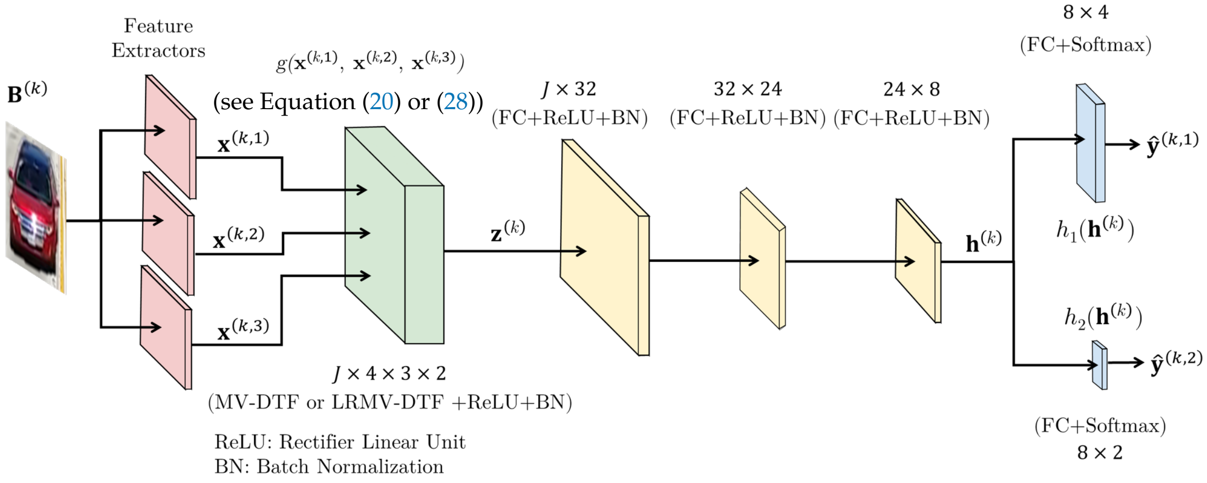


Figure 6. The proposed multitask, multi-view ANN architecture.

Definition 16 (Binary cross-entropy (BCE))**.** *Let $y \in \mathbb{B}$ be the true label of an instance, and let $\hat{y} \in [0,1]$ be the predicted probability for the positive class. The BCE between* $\mathbf{y}$ *and* $\hat{\mathbf{y}}$ *is given by the following:*

$$\mathcal{L}_1(\hat{\mathbf{y}}, \mathbf{y}) = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})] \tag{31}$$

Definition 17 (Multiclass cross entropy (MCE))**.** *Let $\mathbf{y} \in \mathbb{B}^{C}$ be the true label of an instance, related to some multi-classification problem with C classes, encoded in one-hot format. And let $\hat{\mathbf{y}} \in \mathbb{R}^{C}$ be the predicted probability vector, where $\hat{y}_c$ is the probability that the instance belongs to the c-th class. The MCE between* $\mathbf{y}$ *and* $\hat{\mathbf{y}}$ *is given by the following:*

$$\mathcal{L}_2(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{32}$$

6.2.2. Training and Validation

From the total number of tracked vehicles in Table 3, 45% of them were selected from the four test videos via stratified random sampling for training and validation purposes. Including all temporal instances of a particular vehicle can cause data leakage; i.e., the model may learn specific patterns from highly correlated temporal samples, resulting in reduced generalization to unseen vehicles. To prevent this, uncorrelated temporal instances were only considered for each selected vehicle.

Vehicles from the 45% subset, along with their uncorrelated temporal instances, were partitioned into two sets: (1) the training set $\mathbb{D}tr^{(t)}$, containing the 30% of vehicles and their temporal instances; and (2) the validation set $\mathbb{D}va^{(t)}$, with 15% of the vehicles and their instances, where superscript t indexes the task-specific dataset (i.e., $t = 1$ for vehicle-size classification and $t = 2$ for occlusion detection). The remaining 55% of vehicles and their instances comprise the testing set, denoted as $\mathbb{D}_{te}^{(t)}$.

Adaptive moment estimation [124] was employed to optimize the parameters of our model. Training was performed for a maximum of 200 epochs, with an early stopping scheme to avoid overfitting by halting training when performance on $\mathbb{D}_{va}^{(t)}$ no longer improved. The training strategy for our multitask, multi-view model is shown in Algorithm 1, where $\mathbb{F}_{tr}^{(t,b)} \subset \mathbb{D}_{tr}^{(t)}$ is a mutually exclusive batch of the t-th task, i.e., $\mathbb{F}_{tr}^{(t,b)} \cap \mathbb{F}_{tr}^{(t,q)} = \emptyset$ for $b \neq q$, with $b, q \in [K]$, and K is the number of batches.

Algorithm 1 Training scheme.

```
Initialize all weights and biases of the network randomly.
for $i = 1$ to $K$ do
    Optimize the model over batch $\mathbb{F}_{tr}^{(1,i)}$ to minimize the loss for task 1.
    Optimize the model over batch $\mathbb{F}_{tr}^{(2,i)}$ to minimize the loss for task 2.
end for
```

All experiments were conducted and implemented in Python 3.10 and the PyTorch framework on a computer equipped with an Intel Core i7 processor running at 2.2 GHz. To accelerate the processing time, an NVIDIA GTX 1050 TI GPU was employed.

6.3. Performance Evaluation Metrics

In this work, we evaluate the performance of the proposed multitask, multi-view model using six main metrics: accuracy (ACC), F1-measure (F1), geometric mean (GM), normalized Matthews correlation coefficient (MCCn), and normalized Bookmaker informedness (BMn), as detailed in Table 6 (see details of these metrics in [125]). For binary classification, where vehicle instances are categorized into two classes—positive and negative—the performance metrics were directly derived from the entries of a 2 × 2 confusion matrix (CM), characterized by true positives (TPs), false negatives (FNs), false positives (FPs), and true negatives (TNs). In multiclass classification with $C > 2$ classes, the notions of TP, FN, FP, and TN are less straightforward than in binary classification, as the confusion matrix becomes a $C \times C$ matrix whose (i,j)-th entry represents the number of samples that truly belong to the i-th class but were classified as the j-th class. In order to derive the performance metrics, a one-vs.-rest approach is typically employed to reduce the multiclass CM into C binary CMs, where the c-th matrix is formed by treating the c-th class as positive and the rest as the negative class [125,126]. Figure 7 illustrates the CM notion for binary classification (Figure 7a) and multiclass classification with $C = 4$ classes (Figure 7b), which were obtained from the mean values of runs.

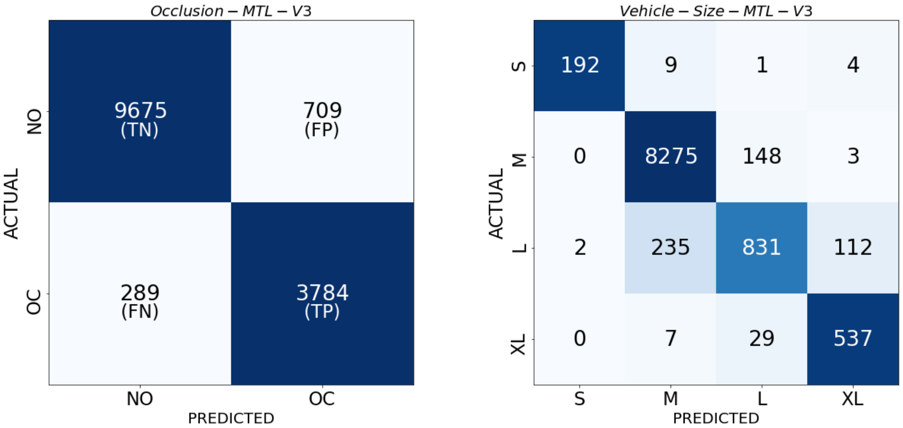
In Table 6, C denotes the number of classes of interest, M is the number of classified instances, $\mathbf{CM} \in \mathbb{R}^{C \times C}$ is a multiclass CM, metrics with subscript c refer to those computed from the c-th binary CM, obtained by reducing the multiclass CM fixing the c-th class. And metrics with subscript w denote weighted metrics, which consider the individual contributions of each class by weighting the metric value of the c-th class by the number of samples, M_c, of class c. This approach provides a "fair" evaluation by considering the impact of imbalanced class distributions on the overall performance.

Furthermore, in order to quantify how much compression is achieved via the LRMV-DTF layer, a compression ratio, Γ, between the number of parameters in the MV-DTF layer and those in the LRMV-DTF layer, i.e., L and $\hat{L}$, is defined in Equation (33). Note that Γ is independent of the latent space dimension, and it depends only on the view dimensions. It ensures that the compression ratio is consistent, regardless of the latent tensor space dimension.

$$\Gamma = \frac{L}{\hat{L}} = \frac{J \times \left(\prod_{m=1}^{M} I_m + 1\right)}{J \times \left(R\sum_{m=1}^{M} I_m + 1\right)} = \frac{\prod_{m=1}^{M} I_m + 1}{R\sum_{m=1}^{M} I_m + 1} \tag{33}$$

Table 6. Mathematical definition of classification performance metrics used in this work (metrics marked with * are biased by class imbalance [127]).

Metric	Equation	Weighted Metric
ACC *	$ACC_c = \frac{TP_c+TN_c}{TP_c+FN_c+TN_c+FP_c}$	$ACC_w = \frac{1}{M}\sum_{c=1}^{C} M_c ACC_c$
F1 *	$F1_c = 2\cdot\frac{PRC_c\cdot SNS_c}{PRC_c+SNS_c}$	$F1_w = \frac{1}{M}\sum_{c=1}^{C} M_c F1_c$
MCC *	$MCC_c = \frac{TP_c\cdot TN_c - FP_c\cdot FN_c}{\sqrt{(TP_c+FP_c)(TP_c+FN_c)(TN_c+FP_c)(TN_c+FN_c)}}$	$MCC_w = \frac{1}{M}\sum_{c=1}^{C} M_c MCC_c$
GM	$GM_c = \sqrt{SNS_c \cdot SPC_c}$	$GM_w = \frac{1}{M}\sum_{c=1}^{C} M_c GM_c$
BM	$BM_c = SNS_c + SPC_c - 1$	$BM_w = \frac{1}{M}\sum_{c=1}^{C} M_c BM_c$
SNS	$SNS_c = \frac{TP_c}{TP_c+FN_c}$	$SNS_w = \frac{1}{M}\sum_{c=1}^{C} M_c SNS_c$
SPC	$SPC_c = \frac{TN_c}{TN_c+FP_c}$	$SPC_w = \frac{1}{M}\sum_{c=1}^{C} M_c SPC_c$
PRC *	$PRC_c = \frac{TP_c}{TP_c+FP_c}$	$PRC_w = \frac{1}{M}\sum_{c=1}^{C} M_c PRC_c$
Global GM	$GGM = \sqrt{SNS_w \cdot SPC_w}$	-
Global BM	$GBM = SNS_w + SPC_w - 1$	-
multiclass MCC *	$mMCC = \frac{M\times\sum_{c=1}^{C} TP_c - \sum_{c=1}^{C} t_c p_c}{\sqrt{M^2-\sum_{c=1}^{C} p_c^2}\times\sqrt{M^2-\sum_{c=1}^{C} t_c^2}}$ $t_c = \sum_{c=1}^{C} \mathbf{CM}_{c:}$ $p_c = \sum_{c=1}^{C} \mathbf{CM}_{:c}$	-



(**a**) Occlusion detection. (**b**) Vehicle-size classification.

Figure 7. Confusion matrices for vehicle-size classification and occlusion detection on video V3 ($J = 2$ and $R = 2$), whose entries correspond to the mean values of runs.

6.4. Hyperparameter Tuning: The Latent Space Dimension J and the Rank R Values

To determine suitable hyperparameters for the low-rank MV-DTF layer, cross-validation via a grid search was employed [128]. Let $\mathcal{R}, \mathcal{J} \subset \mathbb{N}$ be two finite sets containing candidate values for the rank R and latent space dimensionality J, respectively. A grid search trains the multitask, multi-view model, built with the pair $(J, R) \in \mathcal{J} \times \mathcal{R}$, on the training set $\mathbb{D}_{tr}^{(t)}$, and it subsequently evaluates its performance on the validation set $\mathbb{D}_{va}^{(t)}$ using some metric, $\mathcal{M}$. The most suitable pair of values (J^*, R^*) is that which achieves the highest performance metric $\mathcal{M}$ over the validation set $\mathbb{D}_{va}^{(t)}$.

For our case study with a tensor, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J\times4\times3\times2}$, we fixed $\mathcal{J} = \{2, 4, 8, 16, 32, 64, 128, 256\}$ to study the impact of the J value on the classification metrics across tasks, whereas $\mathcal{R} = \{1, 2\}$ was selected based on the rank values that reduce the number of parameters in the LRMV-DTF layer according to the compression ratio (see Table 7), and $R = 3$ and $R = 4$ for performance analysis only. Since our datasets exhibit class imbalance, the MCC as the evaluation metric was used, given its robustness on imbalanced classes, as explained by Luque et al. in [127]. Through this empirical process, we found that $J = 16$ and $R = 2$ achieve the best trade-off between model performance and the compression ratio Γ in the set $\mathcal{J} \times \mathcal{R}$. The sets $\mathcal{J}$ and $\mathcal{R}$, and the most suitable pair of values, $(J^*, R^*) \in \mathcal{J} \times \mathcal{R}$, must be determined for each multitask, multi-view dataset.

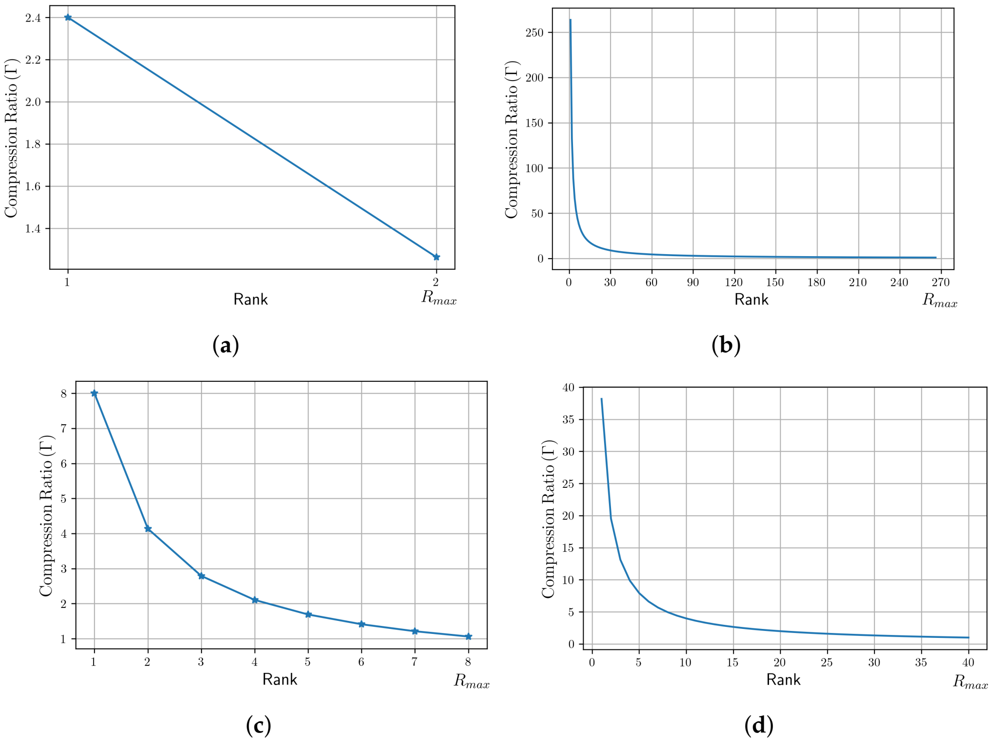
6.5. Performance Evaluation

In this section, the performance of our multitask, multi-view case study in occlusion detection and vehicle classification tasks is evaluated. Our experiments focused on evaluating the impact of the rank, R, and dimension, J, of the latent tensor space $\mathcal{S}$ on computational complexity and model performance. To ensure the consistency of our results, each experiment was repeated 30 times. We first provide the results for the space saving achieved using different R and J values in MV-DFT and its low-rank approximation, LRMV-DTF, followed by an analysis of their effects on the learning phases and model performance.

Table 7 provides a comparison of the Γ compression achieved across different pairs of (J, R) values and two multi-view input space dimensions. It is noteworthy that compression is only achieved for $\Gamma > 1$, and the larger the Γ, the higher the compression. Specifically, for the multi-view space $\mathbb{R}^4 \times \mathbb{R}^3 \times \mathbb{R}^2$, compression is achieved only for $R \leq 2$ (see Figure 8a), while for the multi-view space $\mathbb{R}^{40} \times \mathbb{R}^{30} \times \mathbb{R}^{20}$, a compression can be achieved for higher rank values (see Figure 8b). In consequence, $\Gamma = 1$ provides an upper rank bound, denoted as R_{max}, beyond which compression is no longer achieved. For tensors with greater dimensions or a greater order, the upper rank bound would be greater (see Figure 8).

Table 7. Compression ratio Γ for different pairs of (J, R) values and two multi-view spaces.

(J, R)	**Input Space Dimension** $\dim(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3)$					
	4 × 3 × 2 (Our Case Study)			**40 × 30 × 20**		
	L	$\hat{L}$	Γ	L	$\hat{L}$	Γ
(2, 1)	48	20	2.4	48,000	182	263.7
(2, 2)	48	38	1.26	48,000	362	132.6
(2, 3)	48	56	0.86	48,000	542	88.56
(2, 4)	48	74	0.65	48,000	722	66.48
(8, 1)	192	80	2.4	192,000	728	263.7
(8, 2)	192	152	1.26	192,000	1448	132.6
(8, 3)	192	224	0.86	192,000	2168	88.56
(8, 4)	192	296	0.65	192,000	2888	66.48
(32, 1)	768	320	2.4	768,000	2912	263.7
(32, 2)	768	608	1.26	768,000	5792	132.6
(32, 3)	768	896	0.86	768,000	8766	88.56
(32, 4)	768	1184	0.65	768,000	11,552	66.48



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

(**c**) (**d**)

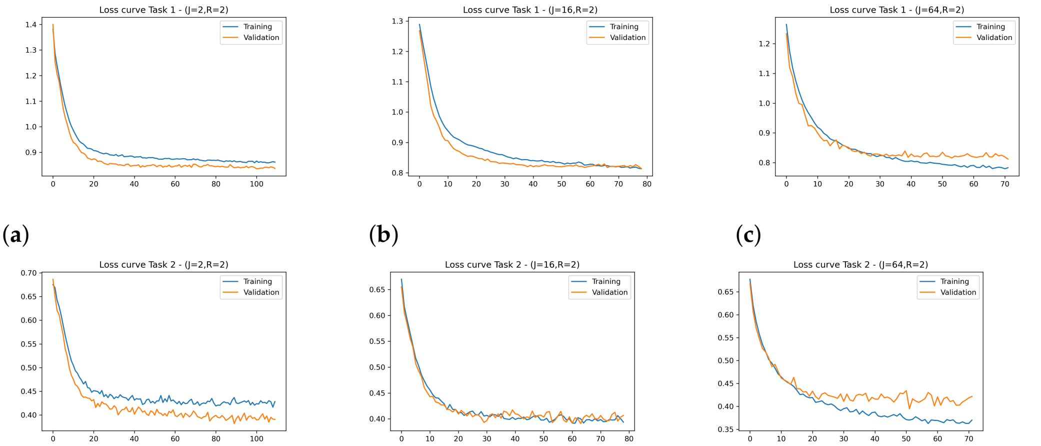
Figure 8. Compression ratio Γ for the set of rank R values that enable compression ($\Gamma > 1$) and multi-view input space dimensions. (**a**) $\dim(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3) = 4 \times 3 \times 2$ our case study. (**b**) $\dim(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3) = 40 \times 30 \times 20$. (**c**) $\dim(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4) = 4 \times 3 \times 2 \times 5$. (**d**) $\dim(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4 \times \mathcal{X}_5) = 4 \times 3 \times 2 \times 5 \times 7$.

Figure 8 illustrates the relationship between the compression ratio Γ and the rank R for various multi-view spaces with different order and dimensionalities. For each space, we observe that, when the rank R increases, the compression ratio Γ decreases. Figure 8a,b show the compression Γ for multi-view spaces with the same order but different dimensionality, while Figure 8c,d show the compression Γ for higher-order multi-view spaces.

Figure 9 shows the training and validation loss curves over epochs for the model using either the MV-DTF or LRMV-DTF layer. From this figure, distinct behaviors in the loss curves can be observed on the training and validation phases:

1. For low-dimensional latent tensor space (see Figure 9a,d), although stable, a slower convergence and higher loss values for both training and validation are observed. This indicates that the model may be underfitting.
2. For high-dimensional latent tensor space (Figure 9c,f), a lower training loss is achieved. However, it exhibits fluctuations in the validation loss, especially for Task 2 (Figure 9f). This suggests that the model begins to overfit as J increases, leading to probable instability in validation performance. The marginal gains in training loss do not justify the increased risk of overfitting.
3. For $J = 16$ (Figure 9b,e), the most balanced performance across both tasks is achieved, showing faster convergence and smoother validation loss curves compared to $J = 2$ and $J = 64$. It achieves lower training loss while maintaining minimal validation loss variability, indicating good generalization.

In the subsequent subsections, the performance evaluation for each task on the tested videos is presented, highlighting the impact of the selected hyperparameters in the model's generalization.



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

(**d**) (**e**) (**f**)

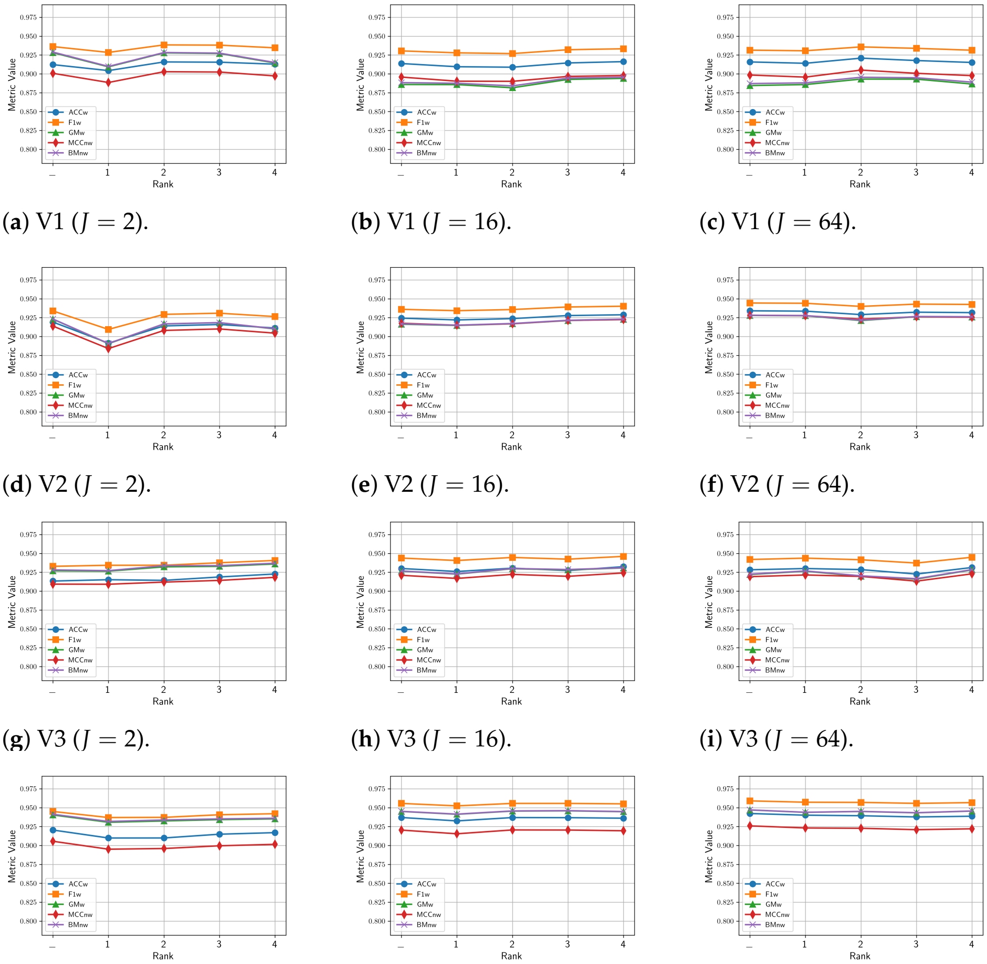
Figure 9. Loss curves obtained during the training and validation stages of the multitask, multi-view model across different latent space dimensions (J values), with fixed rank R = 2, for occlusion detection (first row) and vehicle-size classification (second row). (**a**) Loss curves for task 1 ($J = 2$). (**b**) Loss curves for task 1 ($J = 16$). (**c**) Loss curves for task 1 ($J = 64$). (**d**) Loss curves for task 2 ($J = 2$). (**e**) Loss curves for task 2 ($J = 16$). (**f**) Loss curves for task 1 ($J = 64$).

6.5.1. Vehicle Occlusion Detection Results

This section presents the comparison results of the proposed multitask, multi-view model, with either the MV-DTF or LRMV-DTF layer and different pair of (J, R) values, on the occlusion detection task. Figure 10 shows the mean values of performance metrics obtained from our model for 30 different training runs, evaluated across test videos. Each row corresponds to a specific test video, while each column reflects a particular latent tensor-space dimension J value. As illustrated in Figure 10, a performance drop for different rank R values is very low, especially in high-dimensional spaces (e.g., $J = 16$ and $J = 64$, on the second and third columns of Figure 10). However, in low-dimensional spaces (see the first column of Figure 10 for $J = 2$), the rank choice has a slightly greater impact on the performance, and a fine-tuning rank R value is necessary, as the dimension J decreases.

Additionally, Figure 10 is complemented by Table A1, which presents the mean and standard deviation of performance metrics across multiple runs, with the best and worst values highlighted in blue and red, respectively. From this table, the pair $(16, 2)$ shows the lowest standard deviations across most metrics, providing a good balance between computational complexity (see Table 7) and competitive performance with the MV-DTF layer. Although high-dimensional spaces (e.g., $J = 64$) yield high performance, they also tend to exhibit large standard deviations, potentially increasing the risk of overfitting despite their higher mean values.

Finally, Figure 11 presents a performance comparison between our multi-view multi-task model, using the LRMV-DTF with $(16, 2)$, and single-task learning (STL) single-view learning (SVL) models of SVM and RF, tested across test videos. Figure 11 highlights that the proposed model exhibits higher and more consistent performance than STL-SVL models on all metrics and videos, particularly in V2, V3, and V4. In contrast, the SVM and RF models show a noticeable performance drop in these videos. Overall, the proposed model improves the performance in the MCCnw metric of up to 92.81%, which represents a significant 6% improvement over the SVM and RF models.



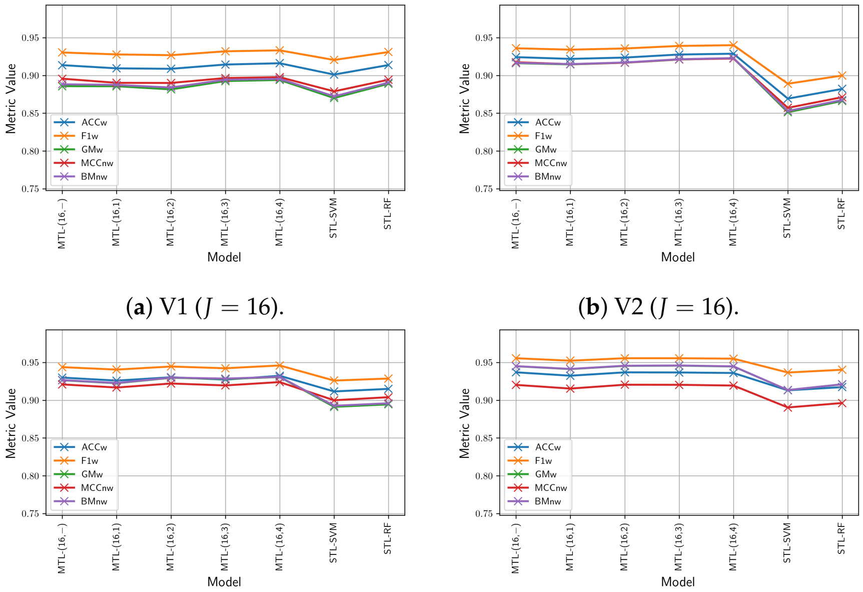
(**a**) V1 ($J = 2$). (**b**) V1 ($J = 16$). (**c**) V1 ($J = 64$).

(**d**) V2 ($J = 2$). (**e**) V2 ($J = 16$). (**f**) V2 ($J = 64$).

(**g**) V3 ($J = 2$). (**h**) V3 ($J = 16$). (**i**) V3 ($J = 64$).

(**j**) V4 ($J = 2$). (**k**) V4 ($J = 16$). (**l**) V4 ($J = 64$).

Figure 10. Mean values for 30 runs of performance metrics achieved on the occlusion detection task in test videos with the multi-view multitask model. The value $R = -$ denotes the results when the MV-DTF layer is employed, whereas the other values correspond to the LRMV-DTF layer.



(**a**) V1 ($J = 16$). (**b**) V2 ($J = 16$).

(**c**) V3 ($J = 16$). (**d**) V4 ($J = 16$).

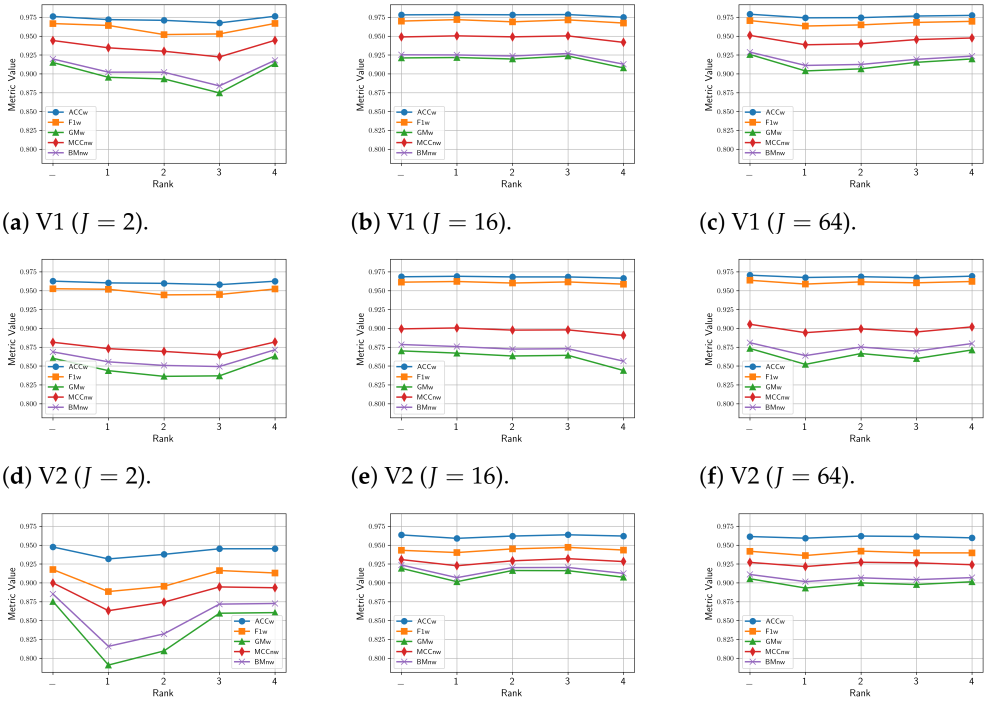
Figure 11. Comparison results between MTL and STL models on the occlusion detection task.

6.5.2. Vehicle-Size Classification Results

This section presents the comparison results of the proposed multitask, multi-view model, incorporating either the MV-DTF or LRMV-DTF layer with different pairs of (J, R) values, on the vehicle-size classification task. Figure 12 shows the mean values of the performance metrics for 30 different training runs on the test videos, where each row and column are related to a specific test video, and latent tensor space dimension, respectively. From this figure, we observe that the lower the J value, the worse the performance. Similarly, high R values generally contribute to improved performance. For $J = 2$, there is a noticeable drop in performance, especially on the GMw, BMnw, and MCCnw metrics, suggesting that low-dimensional spaces fail to capture the complexity of the task. However, as long as J increases to 16 and 64, the metrics stabilize, and the performance drops across ranks becomes negligible, particularly for the ACCw and F1w metrics.

Additionally, Table A2 shows the mean and standard deviation of performance metrics across runs, with the best and worst values highlighted in blue and red, respectively. From this table, we found that high-dimensional spaces tend to yield not only higher mean performance but also lower standard deviation, indicating more stable and consistent outcomes across different test videos. In contrast, low-dimensional spaces (e.g., $J = 2$) are more sensitive to the rank R hyperparameter, particularly for GMw, BMnw, and MCCnw. Consequently, a computationally efficient LRMV-DTF layer can be achieved in high-dimensional spaces by selecting low rank values without a significant performance drop. In contrast, for low-dimensional latent spaces ($J = 2$), the performance is more sensitive to the choice of R, especially for GMw, BMnw, and MCCnw. Therefore, selecting an appropriate rank becomes crucial for low J values to avoid significant drops in performance.

Finally, in Figure 13, a comparison between our multi-view, multitask model and STL-SVL models (SVM and RF) across test videos is presented. This figure highlights the superiority of our multitask model, particularly in videos V3 and V4, where the SVM and RF models again exhibit a significant performance drop. Overall, the proposed model improves the performance in the MCCnw metric by up to 95.10%, which represents a significant 7% improvement over the SVM and RF models.

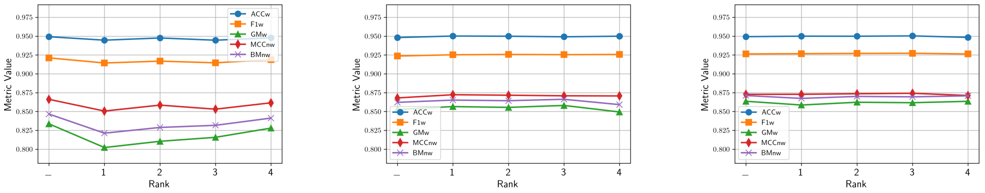


(**a**) V1 ($J = 2$). (**b**) V1 ($J = 16$). (**c**) V1 ($J = 64$).

(**d**) V2 ($J = 2$). (**e**) V2 ($J = 16$). (**f**) V2 ($J = 64$).

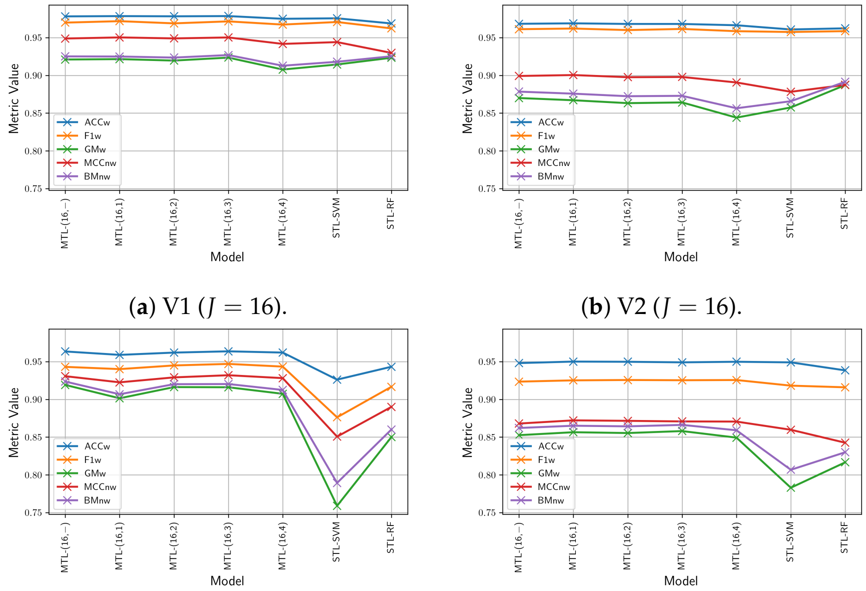
(**g**) V3 ($J = 2$). (**h**) V3 ($J = 16$). (**i**) V3 ($J = 64$).

Figure 12. *Cont.*



(**j**) V4 ($J = 2$). (**k**) V4 ($J = 16$). (**l**) V4 ($J = 64$).

Figure 12. Mean values for 30 runs of performance metrics achieved on the vehicle-size classification task in test videos with the multi-view multitask model. The value $R = -$ denotes the results when the MV-DTF layer is employed, whereas the other values correspond to the LRMV-DTF layer.



(**a**) V1 ($J = 16$). (**b**) V2 ($J = 16$).

(**c**) V3 ($J = 16$). (**d**) V4 ($J = 16$).

Figure 13. Comparison results between MTL and STL models on the vehicle-size classification task.

6.5.3. Comparison with a Multitask Single-View Model

We also provide a comparison between the proposed multitask, multi-view model with its corresponding single-view model in Table A3. The latter model is basically the proposed model but without incorporating the MV-DTF layer, and the input space can only be either $\mathcal{X}_1$, $\mathcal{X}_2$, or $\mathcal{X}_3$. However, in this work, we fix the input space to $\mathcal{X}_2$. Finally, for a fair comparison, this model incorporates a layer that maps the feature space $\mathcal{X}_2$ onto a latent space of dimension J.

The results provided in Table A3 show the overall mean value of weighted metrics across all videos, where it can be observed that incorporating the MV-DTF layer into this single-view model an improvement of up to 1.73% and 1.1% on the BMnw and MCCnw metrics, is achieved. These results are also consistent across all latent space dimensions.

Unlike the F1 metric, the experimental results show that the performance of single-view models does not exhibit a negative impact when the fusion layer is incorporated. Furthermore, even though the model parameters increase, incorporating the MV-DTF layer offers several advantages, including that the layer approximation through Hadamard products allows selecting ranks that, unlike the classical CPD, higher compression rates can be achieved.

Finally, Figures A1 and A2 show the results for the occlusion detection and vehicle-size classification tasks, respectively. In contrast to the performance shown in Figures 10 and 12, Figures A1 and A2 show each metric independently for more detail.

6.6. Discussion

The promising results of the MV-DTF layer and its low-rank approximation LRMV-DTF comprise the following:

1. The performance and consistency of the multitask, multi-view model are significantly influenced by the dimensionality of the latent tensor space (see Figures 10 and 12). For a specific dimension, J^*, the model exhibits two distinct behaviors, given another J value: for $J \leq J^*$, the model tends to underfit, whereas for $J > J^*$, it is prone to overfitting to the training data.
2. A negligible performance drop was observed in our case study as the compression ratio Γ approaches 1, i.e., $\hat{L} \to L$, when the LRMV-DTF layer is employed. This result provides empirical evidence of the underlying low-rank structure in the subtensors $\boldsymbol{\mathcal{A}}_{j:\,\cdot\cdot:}$ of tensor $\boldsymbol{\mathcal{A}}$ in the MV-DTF layer.
3. The maximum allowable rank value R_{max} (upper rank bound) that achieves parameters' compression increases as the dimensions of the multi-view space grow and/or as the number of dimensions (tensor order) increases.

The major limitations of the MV-DTF layer are as follows:

1. Selecting suitable hyperparameters, i.e., the dimensionality $J_1 \times \cdots \times J_P$ or J of the latent tensor space $\mathcal{S}$, and the rank R for the LRMV-DTF layer, is a challenging task.
2. A high-dimensional latent space increases the risk of overfitting, while very low-dimensional spaces may not fully capture the underlying relationships across views, resulting in underfitting.
3. Reducing the rank of subtensors tends to decrease performance and increase the risk of underfitting classification models for low-dimensional latent spaces. Although higher rank values may improve model performance, they also increase the risk of overfitting.
4. The choice of rank R involves a trade-off: higher values increase computational complexity but can capture more complex patterns, while lower values reduce the computational burden but may limit expressiveness of the model, resulting in performance decreasing.

7. Conclusions

In this work, we found a novel connection between the Einstein and Hadamard products for tensors. It is a mathematical relationship involving the Einstein product of the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J \times I_1 \times \cdots \times I_M}$ associated with a multilinear map $\mathcal{T} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, and a rank-one tensor $\boldsymbol{\mathcal{X}} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)}$, where $\dim(\mathcal{S}) = J$, $\dim(\mathcal{X}_m) = I_m$ for all $m \in [M]$, and $\mathbf{x}^{(m)} \in \mathcal{X}_m$. By enforcing low-rank constraints on the subtensors of $\boldsymbol{\mathcal{A}}$, which result by fixing every index but the last M, each j-th subtensor $\boldsymbol{\mathcal{A}}_{j:\,\cdots:}$ is approximated as a rank-$R^{(j)}$ tensor through the CPD. By exploiting this structure, a set of M third-order tensors $\boldsymbol{\mathcal{U}}^{(1)}, \cdots, \boldsymbol{\mathcal{U}}^{(M)}$, here called the Hadamard factor tensors, are obtained. We found that the Einstein product $\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}$ can then be approximated by a sum of R Hadamard products of M Einstein products $\boldsymbol{\mathcal{U}}^{(m)}_{:\,r:} \circledast_1 \mathbf{x}^{(m)}$, where R corresponds to the maximum decomposition rank across subtensors, and $\boldsymbol{\mathcal{U}}^{(m)} \in \mathbb{R}^{J \times R \times I_m}$ for all $m \in [M]$.

Since multi-view learning leverages complementary information from multiple feature sets to enhance model performance, a tensor-based data fusion layer for neural networks, called Multi-View Data Tensor Fusion, is here employed. This layer projects M feature spaces $\mathcal{X}_1, \cdots, \mathcal{X}_M$, referred to as views, into a unified latent tensor space $\mathcal{S}$ through a mapping $g : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, where $\dim(\mathcal{S}) = J$, and $\dim(\mathcal{X}_m) = I_m$ for all $m \in [M]$. Here, we constrain g to the space of affine mappings composed of a multilinear map, $\mathcal{T} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \to \mathcal{S}$, followed by a translation. The multilinear map is here represented by the Einstein product $\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}$, where $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J \times I_1 \times \cdots \times I_M}$ is the induced tensor of $\mathcal{T}$, and $\boldsymbol{\mathcal{X}} \in \mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_M$. Unfortunately, as the number of views increases, the number of parameters that determine g grow exponentially, and consequently, its computational complexity also grows.

To mitigate the curse of dimensionality in the MV-DTF layer, we exploit the mathematical relationship between the Einstein product and Hadamard product, which is the low-rank approximation of the Einstein product, useful when the compression ratio $\Gamma > 1$.

The use of the LRMV-DTF layer based on the Hadamard product does not imply necessarily an improvement of the model performance compared to the MV-DTF layer based on the Einstein product. In fact, the dimension of the latent space J and the rank of subtensors R must be tuned via cross-validation (see Section 6.4). When the decomposition rank of subtensors is less than the upper rank bound R_{max} ($\Gamma > 1$), an efficient low-rank approximation of the MV-DTF layer based on the Einstein product is obtained.

From our experiments, we show that the intoduction of the MV-DTF and LRMV-DTF layers in a case study multitask VTS model for vehicle-size classification and occlusion detection tasks improves its performance compared to single-task and single-view models. For our case study, i.e., a particular case using the LRMV-DTF layer with $J = 16$ and $R = 2$, our model achieved an MCCnw of up to 95.10% for vehicle-size classification and 92.81% for occlusion detection, representing significant improvements of 7% and 6%, respectively, over single-task single-view models while reducing the number of parameters by a factor of 1.3.

Finally, for every case study, the dimension of the latent tensor space, J, and the decomposition rank, R, must be tuned. Additionally, the employment of an MV-DTF layer or a LRMV-DTF layer must be determined while the tradeoff between the model performance and computational complexity is taken into account.

Open Issues

1. A computational complexity analysis must be conducted to evaluate the LRMV-DTF layer efficiency.
2. For VTS systems, to integrate other high-dimensional feature spaces in order to improve the expressiveness of the latent tensor space and its computational efficiency.
3. To explore other tensor decomposition models, such as the tensor-train model, for more efficient algorithms in high-dimensional data.
4. To extend our work to more complex network architectures.
5. To address other VTS tasks within the MTL framework for a more comprehensive vehicle traffic model.

Author Contributions: F.H.-R. and D.T.-R. wrote the article. D.T.-R. contributed the initial concept of applying multitask learning and tensors to VTS systems. F.H.-R. conceptualized the multi-view learning for VTS systems, developed and implemented the associated algorithms, and identified the mathematical relationship between the Einstein and Hadamard products. The theoretical discussion of Einstein and Hadamard products was written by F.H.-R. and D.T.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: No applicable.

Informed Consent Statement: No applicable.

Data Availability Statement: The datasets used to support the findings of this study, particularly for the multi-view multitask model, are publicly available at the following GitHub repository: https://github.com/fhermosillo/VTSMultiviewDatasets (accessed on 20 October 2024). Unfortunately, the code employed in this work is currently private, but it can be made available upon request to reviewers for evaluation purposes. Please refer to the repository or contact the corresponding author for further inquiries or additional data requests.

Acknowledgments: The authors acknowledge the Consejo Nacional de Ciencia y Tecnología (CONACYT) for Ph.D. student grant no. CB-253955. The first author also expresses gratitude to the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV IPN) and its academic staff, especially to Torres, D. Special thanks are extended to the reviewers for their valuable feedback and constructive comments, which greatly contributed to improving the quality of this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

ITS	Intelligent Transportation Systems
VTS	Vehicle traffic surveillance
MVL	Multi-view learning
MTL	Multitask learning
MV-DTF	Multi-View Data Tensor Fusion
LRMV-DTF	Low-Rank Multi-view Data Tensor Fusion

Appendix A. Mathematical Proofs

Appendix A.1. Proof of Proposition

Let $\mathcal{T} : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \rightarrow \mathbb{R}^{J_1 \times \cdots \times J_P}$ be a multilinear map, and let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J_1 \times \cdots \times J_P \times I_1 \times \cdots \times I_M}$ be its associated tensor. Also, let $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}$ be M vectors where the m-th vector $\mathbf{x}^{(m)} \in \mathbb{R}^{I_m}$. And let $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{J_1 \times \cdots \times J_P}$ be the image of the tuple $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)})$ under $\mathcal{T}$, i.e., $\boldsymbol{\mathcal{Z}} = \mathcal{T}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)})$. According to Definition 14, for $\boldsymbol{\mathcal{X}} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, $\mathcal{T}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)})$ can be expressed as $\boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathcal{X}}$, whose $(j_1, \cdots, j_P)$-th entry is given as follows:

$$z_{j_1 \cdots j_P} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} a_{j_1 \cdots j_P i_1 \cdots i_M} x_{i_1 \cdots i_M} = \boldsymbol{\mathcal{A}}_{j_1 \cdots j_P :\, \cdots :} \circledast_M \boldsymbol{\mathcal{X}} \tag{A1}$$

From the CPD (Section 2.4.2), each subtensor $\boldsymbol{\mathcal{A}}_{j_1 \cdots j_P :\, \cdots :}$ can be approximated as a rank-$R^{(j_1, \cdots, j_P)}$ tensor, as Equation (A2) shows, where $\boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ is its r-th rank-one tensor, while Equation (A2) holds if $R^{(j_1, \cdots, j_P)}$ is the decomposition rank of $\boldsymbol{\mathcal{A}}_{j_1 \cdots j_P :\, \cdots :}$.

$$\boldsymbol{\mathcal{A}}_{j_1 \cdots j_P :\, \cdots :} \approxeq \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :} \tag{A2}$$

Substituting Equation (A2) into Equation (A1) results in Equation (A3):

$$z_{j_1 \cdots j_P} \approxeq \left\langle \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :}, \boldsymbol{\mathcal{X}} \right\rangle = \left(\sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :} \right) \circledast_M \boldsymbol{\mathcal{X}} \tag{A3}$$

By exploiting the distributive property of the Einstein product, tensor $\boldsymbol{\mathcal{X}}$ can be distributed along the summation, as Equation (A4) shows.

$$\begin{aligned} z_{j_1 \cdots j_P} &\approxeq \left(\sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :} \right) \circledast_M \boldsymbol{\mathcal{X}} \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \left(\boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P : \cdots :} \circledast_M \boldsymbol{\mathcal{X}} \right) \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} a^{(r)}_{j_1 \cdots j_P i_1 \cdots i_M} x_{i_1 \cdots i_M} \right) \end{aligned} \tag{A4}$$

Note that, as $\boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :}$ is a rank-one tensor, it can be factorized into the tensor product of N vectors (see Definition 7), as shown in Equation (A5), where $\mathbf{v}^{(j_1, \cdots, j_P, r, m)} \in \mathbb{R}^{I_m}$ denotes its mth vector, also called the factor vector, which is related to the m-mode.

$$\boldsymbol{\mathcal{A}}^{(r)}_{j_1 \cdots j_P :\, \cdots :} = \mathbf{v}^{(j_1, \cdots, j_P, r, 1)} \otimes \cdots \otimes \mathbf{v}^{(j_1, \cdots, j_P, r, M)} \tag{A5}$$

Since $\mathcal{X}$ is also a rank-one tensor, i.e., $\mathcal{X} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(M)}$, it follows that $x_{i_1 \cdots i_M} = \prod_{m=1}^{M} x_{i_m}^{(m)}$, and $a_{j_1 \cdots j_P i_1 \cdots i_M}^{(r)} = \prod_{m=1}^{M} v_{i_m}^{(j_1, \cdots, j_P, r, m)}$, respectively. Hence, Equation (A4) can be rewritten as follows:

$$\begin{aligned} z_{j_1 \cdots j_P} &\approxeq \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} a_{j_1 \cdots j_P i_1 \cdots i_M}^{(r)} x_{i_1}^{(1)} \cdots x_{i_M}^{(M)} \right) \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} v_{i_1}^{(j_1, \cdots, j_P, r, 1)} \cdots v_{i_M}^{(j_1, \cdots, j_P, r, M)} x_{i_1}^{(1)} \cdots x_{i_M}^{(M)} \right) \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} \prod_{m=1}^{M} v_{i_m}^{(j_1, \cdots, j_P, r, m)} x_{i_m}^{(m)} \right) \end{aligned} \tag{A6}$$

By leveraging the independence of the terms involved in the summations, it can be rearranged as a sum of the products of inner products, as shown in Equation (A7).

$$\begin{aligned} &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \sum_{i_1=1}^{I_1} v_{i_1}^{(j_1, \cdots, j_P, r, 1)} x_{i_1}^{(1)} \cdots \sum_{i_M=1}^{I_M} v_{i_M}^{(j_1, \cdots, j_P, r, M)} x_{i_M}^{(M)} \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \langle \mathbf{v}^{(j_1, \cdots, j_P, r, 1)}, \mathbf{x}^{(1)} \rangle \cdots \langle \mathbf{v}^{(j_1, \cdots, j_P, r, M)}, \mathbf{x}^{(M)} \rangle \\ &= \sum_{r=1}^{R^{(j_1, \cdots, j_P)}} \prod_{m=1}^{M} \langle \mathbf{v}^{(j_1, \cdots, j_P, r, m)}, \mathbf{x}^{(m)} \rangle \end{aligned} \tag{A7}$$

From Equation (A7), two cases can be distinguished:

i The tensor ranks of all subtensors are equal.

ii The tensor ranks of all subtensors are different.

Appendix A.1.1. The Tensor Ranks of Subtensors Are Equal

Since the tensor rank $R^{(j_1, \cdots, j_P)}$ is equal across subtensors, Equation (A6) becomes Equation (A8), where $R = R^{(j_1, \cdots, j_P)} \forall j_p \in [J_p], p \in [P]$.

$$z_{j_1, \cdots, j_P} \approxeq \sum_{r=1}^{R} \prod_{m=1}^{M} \langle \mathbf{v}^{(j_1, \cdots, j_P, r, m)}, \mathbf{x}^{(m)} \rangle \tag{A8}$$

From the inner product $\langle \mathbf{v}^{(j_1, \cdots, j_P, r, m)}, \mathbf{x}^{(m)} \rangle = \sum_{i_m=1}^{I_m} v_{i_m}^{(j_1, \cdots, j_P, r, m)}$, it should be noted that the term $v_{i_m}^{(j_1 \cdots j_P, r, m)}$ is indexed by $P+2$ indices $j_1 \in [J_1], \cdots, j_P \in [J_P], r \in [R]$, and $i_m \in [I_m]$. Let $u_{j_1 \cdots j_P r i_m}^{(m)} = v_{i_m}^{(j_1, \cdots, j_P, r, m)}$ be the $(j_1, \cdots, j_P, r, i_m)$th entry of a $(P+2)$th-order tensor $\mathcal{U}^{(m)} \in \mathbb{R}^{J_1 \times \cdots \times J_P \times R \times I_m} \forall m \in [M]$, here called the mth factor tensor, whose $(P+2)$-mode fiber $\mathcal{U}_{j_1 \cdots j_P r:}^{(m)} = \mathbf{v}^{(j_1, \cdots, j_P, r, m)}$. From this tensor, the inner product of Equation (A8) can be rewritten as the Einstein product $\mathcal{U}_{j_1 \cdots j_P r:}^{(m)} \circledast_1 \mathbf{x}^{(m)}$, as Equation (A9) shows:

$$z_{j_1 \cdots j_P} \approxeq \sum_{r=1}^{R} \prod_{m=1}^{M} \mathcal{U}_{j_1 \cdots j_P r:}^{(m)} \circledast_1 \mathbf{x}^{(m)} \tag{A9}$$

To simplify this proof, we restrict the order P of $\mathcal{S}$ as the unidimensional case, i.e., $P = 1$; however, it can be easily generalized to any arbitrary order. As a consequence, the tensor $\mathfrak{U}^{(m)} \in \mathbb{R}^{J_1 \times R \times I_m}$ and $\mathfrak{Z}$ become the vector $\mathbf{z} \in \mathbb{R}^{J_1}$, which is given by Equation (A10):

$$\mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_{J_1} \end{bmatrix} \approxeq \begin{bmatrix} \sum_{r=1}^{R} \prod_{m=1}^{M} \mathfrak{U}^{(m)}_{1r:} \circledast_1 \mathbf{x}^{(m)} \\ \vdots \\ \sum_{r=1}^{R} \prod_{m=1}^{M} \mathfrak{U}^{(m)}_{J_1 r:} \circledast_1 \mathbf{x}^{(m)} \end{bmatrix} \tag{A10}$$

The summations and products on Equation (A10) can be rewritten as the product $\mathfrak{U}^{(m)}_{j_1 r:} \circledast_1 \mathbf{x}^{(m)}$ using the Hadamard product, as Equation (A11) shows:

$$\mathbf{z} \approxeq \sum_{r=1}^{R} \bigodot_{m=1}^{M} \begin{bmatrix} \mathfrak{U}^{(m)}_{1r:} \circledast_1 \mathbf{x}^{(m)} \\ \vdots \\ \mathfrak{U}^{(m)}_{J_1 r:} \circledast_1 \mathbf{x}^{(m)} \end{bmatrix} = \sum_{r=1}^{R} \bigodot_{m=1}^{M} \mathbf{z}^{m,r} \tag{A11}$$

Note that the entries of vector $\mathbf{z}^{(m,r)} \in \mathbb{R}^{J}$ are inner products, i.e., $\mathfrak{U}^{(m)}_{j_1 r:} \circledast_1 \mathbf{x}^{(m)} = \langle \mathfrak{U}^{(m)}_{j_1 r:}, \mathbf{x}^{(m)} \rangle$. It resembles the standard matrix times vector multiplication, which is carried along the first and third mode of $\mathfrak{U}^{(m)}$ with vector $\mathbf{x}^{(m)}$, as illustrated by Equation (A12):

$$\mathbf{z} \approxeq \sum_{r=1}^{R} \bigodot_{m=1}^{M} \begin{bmatrix} \langle \mathfrak{U}^{(m)}_{1r:}, \mathbf{x}^{(m)} \rangle \\ \vdots \\ \langle \mathfrak{U}^{(m)}_{J_1 r:}, \mathbf{x}^{(m)} \rangle \end{bmatrix} = \sum_{r=1}^{R} \bigodot_{m=1}^{M} \begin{bmatrix} \sum_{i_m=1}^{I_m} u^{(m)}_{1ri_m} x^{(m)}_{i_m} \\ \vdots \\ \sum_{i_m=1}^{I_m} u^{(m)}_{J_1 ri_m} x^{(m)}_{i_m} \end{bmatrix} = \sum_{r=1}^{R} \bigodot_{m=1}^{M} \mathfrak{U}^{(m)}_{:r:} \circledast_1 \mathbf{x}^{(m)} \tag{A12}$$

Finally, also note that the outer summation is carried out along the r index, and it can be expressed by the Einstein product of the Hadamard product of Einstein products $\mathfrak{U}^{(m)} \circledast_1 \mathbf{x}^{(m)}$ with a vector $\mathbf{1} \in \mathbb{R}^{J_1}$, whose entries are 1, as shown in Equation (A13), which concludes the proof.

$$\mathbf{z} \approxeq \sum_{r=1}^{R} \bigodot_{m=1}^{M} \mathfrak{U}^{(m)}_{:r:} \circledast_1 \mathbf{x}^{(m)} = \left[\bigodot_{m=1}^{M} \mathfrak{U}^{(m)} \circledast_1 \mathbf{x}^{(m)} \right] \circledast_1 \mathbf{1} \tag{A13}$$

Appendix A.1.2. The Tensor Ranks of Subtensors Are Different

Here, we also restrict the order P of $\mathcal{S}$ as the unidimensional case. Let R be the maximum decomposition rank across subtensors, as Equation (A14) shows:

$$R = \max_{j_1 \in [J_1]} R^{(j_1)} \tag{A14}$$

Then, Equation (A6) can be rewritten as depicted in Equation (A15).

$$z_{j_1} \approxeq \sum_{r=1}^{R} \begin{cases} \prod_{m=1}^{M} \langle \mathbf{v}^{(j_1,r,m)}, \mathbf{x}^{(m)} \rangle, & r \leq R^{(j_1)} \\ 0, & r > R^{(j_1)} \end{cases} \tag{A15}$$

We also define a tensor $\mathfrak{U}^{(m)} \in \mathbb{R}^{J_1 \times R \times I_m}$, whose (j_1, r, i_m)th entry $u^{(m)}_{j_1 r i_m}$ is given as follows:

$$u^{(m)}_{j_1 r i_m} = \begin{cases} v^{(j_1,r,m)}_{i_m}, & r \leq R^{(j)} \\ 0, & r > R^{(j)} \end{cases} \tag{A16}$$

Note that Equation (A15) is consistent with Equation (A7). Consequently, Equation (A14) also holds for the second case, which completes this proof.

Appendix A.2. Low-Rank Multi-View Tensor Data Fusion Layer for Unidimensional Latent Spaces

A particular case of the MV-DTF layer is for unidimensional latent spaces, i.e., $P = 1$, and $\dim(\mathcal{S}) = J$. In this scenario, the multilinear transformation $\mathcal{T} : \mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_M} \to \mathbb{R}^J$ is induced via the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{J \times I_1 \times \cdots \times I_M}$. From the Einstein product (Definition 11), the j-th entry of $\mathbf{z} = \boldsymbol{\mathcal{A}} \circledast_M \boldsymbol{\mathfrak{X}}$ is given by Equation (A1), where $\mathbf{z} \in \mathbb{R}^J$, and $j \in [J]$.

$$z_j = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} a_{ji_1 \cdots i_M} x_{i_1 \cdots i_M} = \boldsymbol{\mathcal{A}}_{j: \, \cdots :} \circledast_M \boldsymbol{\mathfrak{X}} \tag{A17}$$

Using the CPD, each j-th subtensor $\boldsymbol{\mathcal{A}}_{j: \, \cdots :} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ can then be decomposed into $R^{(j)}$ rank-one tensors, as shown in Equation (A18), where $\mathbf{v}^{(j,r,m)} \in \mathbb{R}^{I_m}$ is the m-mode factor vector for the r-th rank-one tensor $\mathbf{v}^{(j,r,1)} \otimes \cdots \otimes \mathbf{v}^{(j,r,M)}$, and $r \in [R^{(j)}]$.

$$\boldsymbol{\mathcal{A}}_{j: \, \cdots :} \approxeq \sum_{r=1}^{R^{(j)}} \mathbf{v}^{(j,r,1)} \otimes \cdots \otimes \mathbf{v}^{(j,r,M)} \tag{A18}$$

Then, similar to Equation (A7), subtensor $\boldsymbol{\mathcal{A}}_{j: \, \cdots :}$ can also be expressed as a sum of products of inner products, as Equation (A18) shows.

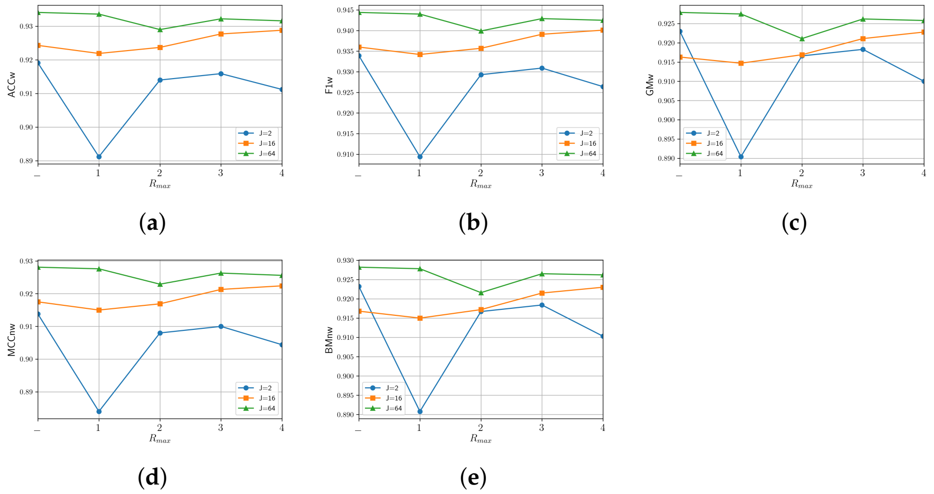
$$z_j \approxeq \sum_{r=1}^{R^{(j)}} \prod_{m=1}^{M} \langle \mathbf{v}^{(j,r,m)}, \mathbf{x}^{(m)} \rangle \tag{A19}$$

Consequently, it yields the same tensor forms derived in Appendixes A.1.1 and A.1.2, i.e.,

$$\mathbf{z} \approxeq \left[\bigodot_{m=1}^{M} \boldsymbol{\mathfrak{U}}^{(m)} \circledast_1 \mathbf{x}^{(m)} \right] \circledast_1 \mathbf{1} \tag{A20}$$

where $\boldsymbol{\mathfrak{U}}^{(m)} \in \mathbb{R}^{J \times R \times I_m}$ is the m-mode factor tensor associated with the m-th view, whose third-mode fiber $\boldsymbol{\mathfrak{U}}^{(m)}_{jr:} = \mathbf{v}^{(j,r,m)}$.

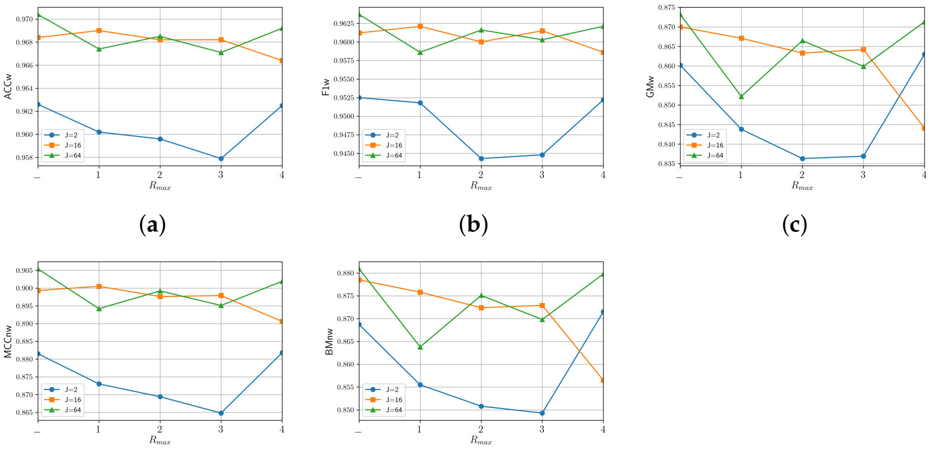
Appendix B. Results



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

(**d**) (**e**)

Figure A1. Mean values for 30 runs of performance metrics achieved in the occlusion detection task with test video V2 and the multi-view multitask model. The value $R = -$ denotes the results when the MV-DTF layer was employed, whereas the other values correspond to the LRMV-DTF layer. (**a**) Weighted accuracy. (**b**) Weighted F1. (**c**) Weighted GM. (**d**) Weighted normalized MCC. (**e**) Weighted normalized BM.



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

(**d**) (**e**)

Figure A2. Mean values for 30 runs of performance metrics achieved in the vehicle-size classification task with test video V2 and the multi-view multitask model. The value $R = -$ denotes the results when the MV-DTF layer was employed, whereas the other values correspond to the LRMV-DTF layer. (**a**) Weighted accuracy. (**b**) Weighted F1. (**c**) Weighted GM. (**d**) Weighted normalized MCC. (**e**) Weighted normalized BM.

Table A1. Mean and standard deviation of metrics for different pairs of (J, R) values for 30 runs for occlusion detection. The value $R = -$ denotes the results when the MV-DTF layer was employed, whereas the other values correspond to the LRMV-DTF layer.

Video	(J, R)	ACCw	F1w	GMw	MCCnw	BMnw
V1	$(2, -)$	0.9122 ± 0.0075	0.9361 ± 0.0060	0.9281 ± 0.0090	0.9007 ± 0.0084	0.9289 ± 0.0093
	$(2, 1)$	0.9043 ± 0.0073	0.9284 ± 0.0070	0.9093 ± 0.0193	0.8885 ± 0.0110	0.9098 ± 0.0193
	$(2, 2)$	0.9158 ± 0.0094	0.9384 ± 0.0074	0.9278 ± 0.0107	0.9028 ± 0.0105	0.9283 ± 0.0108
	$(2, 3)$	0.9155 ± 0.0043	0.9381 ± 0.0037	0.9270 ± 0.0087	0.9024 ± 0.0059	0.9275 ± 0.0090
	$(2, 4)$	0.9128 ± 0.0091	0.9345 ± 0.0086	0.9144 ± 0.0226	0.8973 ± 0.0131	0.9151 ± 0.0226
V1	$(16, -)$	0.9137 ± 0.0206	0.9304 ± 0.0178	0.8858 ± 0.0368	0.8957 ± 0.0245	0.8883 ± 0.0345
	$(16, 1)$	0.9094 ± 0.0114	0.9279 ± 0.0096	0.8857 ± 0.0217	0.8903 ± 0.0142	0.8875 ± 0.0204
	$(16, 2)$	0.9088 ± 0.0142	0.9268 ± 0.0128	0.8815 ± 0.0305	0.8901 ± 0.0170	0.8840 ± 0.0284
	$(16, 3)$	0.9145 ± 0.0137	0.9320 ± 0.0122	0.8926 ± 0.0282	0.8966 ± 0.0163	0.8944 ± 0.0264
	$(16, 4)$	0.9162 ± 0.0153	0.9332 ± 0.0130	0.8940 ± 0.0259	0.8978 ± 0.0187	0.8955 ± 0.0246
V1	$(64, -)$	0.9158 ± 0.0203	0.9314 ± 0.0171	0.8844 ± 0.0328	0.8984 ± 0.0243	0.8871 ± 0.0311
	$(64, 1)$	0.9140 ± 0.0160	0.9306 ± 0.0139	0.8858 ± 0.0289	0.8957 ± 0.0191	0.8880 ± 0.0271
	$(64, 2)$	0.9208 ± 0.0193	0.9357 ± 0.0165	0.8931 ± 0.0341	0.9050 ± 0.0225	0.8955 ± 0.0318
	$(64, 3)$	0.9177 ± 0.0152	0.9338 ± 0.0120	0.8930 ± 0.0229	0.9007 ± 0.0186	0.8948 ± 0.0219
	$(64, 4)$	0.9151 ± 0.0183	0.9313 ± 0.0157	0.8865 ± 0.0326	0.8976 ± 0.0219	0.8890 ± 0.0307
V2	$(2, -)$	0.9191 ± 0.0132	0.9339 ± 0.0117	0.9230 ± 0.0168	0.9138 ± 0.0148	0.9232 ± 0.0169
	$(2, 1)$	0.8912 ± 0.0376	0.9094 ± 0.0326	0.8904 ± 0.0400	0.8840 ± 0.0387	0.8908 ± 0.0399
	$(2, 2)$	0.9140 ± 0.0139	0.9293 ± 0.0124	0.9166 ± 0.0181	0.9080 ± 0.0159	0.9167 ± 0.0182
	$(2, 3)$	0.9159 ± 0.0089	0.9309 ± 0.0076	0.9183 ± 0.0100	0.9100 ± 0.0095	0.9184 ± 0.0101
	$(2, 4)$	0.9112 ± 0.0138	0.9264 ± 0.0126	0.9100 ± 0.0209	0.9044 ± 0.0157	0.9103 ± 0.0205
V2	$(16, -)$	0.9243 ± 0.0121	0.9360 ± 0.0109	0.9163 ± 0.0171	0.9175 ± 0.0131	0.9168 ± 0.0167
	$(16, 1)$	0.9219 ± 0.0089	0.9342 ± 0.0078	0.9147 ± 0.0118	0.9150 ± 0.0097	0.9150 ± 0.0116
	$(16, 2)$	0.9237 ± 0.0098	0.9357 ± 0.0088	0.9169 ± 0.0135	0.9169 ± 0.0108	0.9172 ± 0.0133
	$(16, 3)$	0.9277 ± 0.0100	0.9391 ± 0.0092	0.9211 ± 0.0148	0.9213 ± 0.0108	0.9215 ± 0.0144
	$(16, 4)$	0.9288 ± 0.0085	0.9401 ± 0.0075	0.9228 ± 0.0113	0.9224 ± 0.0093	0.9230 ± 0.0111

Table A1. *Cont.*

Video	(J, R)	ACCw	F1w	GMw	MCCnw	BMnw
V2	(64, −)	0.9341 ± 0.0080	0.9444 ± 0.0073	0.9279 ± 0.0115	0.9281 ± 0.0086	0.9282 ± 0.0113
	(64, 1)	0.9336 ± 0.0082	0.9440 ± 0.0076	0.9275 ± 0.0120	0.9276 ± 0.0089	0.9278 ± 0.0118
	(64, 2)	0.9290 ± 0.0103	0.9399 ± 0.0093	0.9211 ± 0.0148	0.9229 ± 0.0111	0.9216 ± 0.0144
	(64, 3)	0.9322 ± 0.0102	0.9429 ± 0.0091	0.9262 ± 0.0139	0.9263 ± 0.0111	0.9265 ± 0.0136
	(64, 4)	0.9316 ± 0.0091	0.9425 ± 0.0084	0.9258 ± 0.0136	0.9256 ± 0.0097	0.9262 ± 0.0132
V3	(2, −)	0.9134 ± 0.0102	0.9329 ± 0.0077	0.9270 ± 0.0069	0.9092 ± 0.0077	0.9281 ± 0.0068
	(2, 1)	0.9151 ± 0.0116	0.9342 ± 0.0097	0.9265 ± 0.0136	0.9091 ± 0.0128	0.9271 ± 0.0138
	(2, 2)	0.9143 ± 0.0092	0.9343 ± 0.0075	0.9322 ± 0.0076	0.9119 ± 0.0081	0.9337 ± 0.0074
	(2, 3)	0.9187 ± 0.0107	0.9376 ± 0.0087	0.9329 ± 0.0099	0.9142 ± 0.0105	0.9338 ± 0.0099
	(2, 4)	0.9227 ± 0.0110	0.9407 ± 0.0083	0.9360 ± 0.0071	0.9182 ± 0.0095	0.9369 ± 0.0069
V3	(16, −)	0.9300 ± 0.0097	0.9438 ± 0.0083	0.9265 ± 0.0143	0.9210 ± 0.0110	0.9267 ± 0.0141
	(16, 1)	0.9260 ± 0.0169	0.9406 ± 0.0138	0.9225 ± 0.0199	0.9168 ± 0.0191	0.9228 ± 0.0198
	(16, 2)	0.9304 ± 0.0110	0.9446 ± 0.0090	0.9297 ± 0.0139	0.9222 ± 0.0115	0.9300 ± 0.0136
	(16, 3)	0.9273 ± 0.0102	0.9423 ± 0.0087	0.9281 ± 0.0164	0.9196 ± 0.0109	0.9286 ± 0.0160
	(16, 4)	0.9324 ± 0.0107	0.9460 ± 0.0086	0.9307 ± 0.0126	0.9241 ± 0.0115	0.9310 ± 0.0125
V3	(64, −)	0.9283 ± 0.0093	0.9419 ± 0.0085	0.9220 ± 0.0168	0.9192 ± 0.0103	0.9225 ± 0.0162
	(64, 1)	0.9299 ± 0.0083	0.9437 ± 0.0073	0.9264 ± 0.0144	0.9214 ± 0.0091	0.9268 ± 0.0141
	(64, 2)	0.9285 ± 0.0136	0.9416 ± 0.0111	0.9197 ± 0.0172	0.9195 ± 0.0151	0.9203 ± 0.0169
	(64, 3)	0.9226 ± 0.0148	0.9372 ± 0.0120	0.9159 ± 0.0177	0.9132 ± 0.0161	0.9165 ± 0.0175
	(64, 4)	0.9313 ± 0.0066	0.9449 ± 0.0054	0.9282 ± 0.0106	0.9229 ± 0.0070	0.9285 ± 0.0104
V4	(2, −)	0.9204 ± 0.0057	0.9450 ± 0.0041	0.9404 ± 0.0041	0.9056 ± 0.0055	0.9413 ± 0.0040
	(2, 1)	0.9099 ± 0.0130	0.9369 ± 0.0093	0.9307 ± 0.0089	0.8951 ± 0.0115	0.9318 ± 0.0086
	(2, 2)	0.9099 ± 0.0146	0.9372 ± 0.0107	0.9324 ± 0.0093	0.8960 ± 0.0123	0.9337 ± 0.0087
	(2, 3)	0.9149 ± 0.0140	0.9406 ± 0.0106	0.9342 ± 0.0141	0.8996 ± 0.0143	0.9351 ± 0.0141
	(2, 4)	0.9170 ± 0.0156	0.9420 ± 0.0113	0.9352 ± 0.0117	0.9015 ± 0.0149	0.9360 ± 0.0114
V4	(16, −)	0.9370 ± 0.0070	0.9556 ± 0.0047	0.9450 ± 0.0047	0.9203 ± 0.0071	0.9451 ± 0.0046
	(16, 1)	0.9325 ± 0.0079	0.9524 ± 0.0052	0.9412 ± 0.0043	0.9155 ± 0.0078	0.9415 ± 0.0043
	(16, 2)	0.9369 ± 0.0080	0.9556 ± 0.0053	0.9455 ± 0.0053	0.9205 ± 0.0079	0.9457 ± 0.0052
	(16, 3)	0.9368 ± 0.0063	0.9556 ± 0.0040	0.9458 ± 0.0035	0.9204 ± 0.0061	0.9461 ± 0.0035
	(16, 4)	0.9361 ± 0.0064	0.9550 ± 0.0043	0.9447 ± 0.0048	0.9194 ± 0.0065	0.9449 ± 0.0048
V4	(64, −)	0.9423 ± 0.0049	0.9590 ± 0.0032	0.9470 ± 0.0044	0.9259 ± 0.0052	0.9471 ± 0.0044
	(64, 1)	0.9401 ± 0.0052	0.9572 ± 0.0036	0.9438 ± 0.0063	0.9231 ± 0.0057	0.9439 ± 0.0063
	(64, 2)	0.9393 ± 0.0085	0.9570 ± 0.0058	0.9449 ± 0.0065	0.9228 ± 0.0090	0.9451 ± 0.0064
	(64, 3)	0.9377 ± 0.0094	0.9557 ± 0.0063	0.9432 ± 0.0066	0.9209 ± 0.0098	0.9433 ± 0.0065
	(64, 4)	0.9387 ± 0.0067	0.9567 ± 0.0046	0.9455 ± 0.0052	0.9220 ± 0.0072	0.9456 ± 0.0052

Values highlighted in red denote the worst values achieved on every group. Values highlighted in blue denote the best values achieved on every group.

Table A2. Mean and standard deviation of metrics for different pairs of (J, R) values for 30 runs in vehicle-size classification. The value $R = -$ denotes the results when the MV-DTF layer is employed, whereas the other values correspond to the LRMV-DTF layer.

Video	(J,R)	ACCw	F1w	GMw	MCCnw	BMnw
V1	(2, −)	0.9761 ± 0.0069	0.9666 ± 0.0119	0.9150 ± 0.0333	0.9441 ± 0.0160	0.9196 ± 0.0300
	(2, 1)	0.9719 ± 0.0103	0.9641 ± 0.0132	0.8953 ± 0.0480	0.9345 ± 0.0251	0.9022 ± 0.0410
	(2, 2)	0.9710 ± 0.0092	0.9520 ± 0.0264	0.8932 ± 0.0459	0.9299 ± 0.0257	0.9021 ± 0.0370
	(2, 3)	0.9675 ± 0.0101	0.9529 ± 0.0209	0.8747 ± 0.0432	0.9224 ± 0.0262	0.8840 ± 0.0371
	(2, 4)	0.9764 ± 0.0062	0.9668 ± 0.0119	0.9135 ± 0.0284	0.9444 ± 0.0157	0.9179 ± 0.0254
V1	(16, −)	0.9782 ± 0.0089	0.9699 ± 0.0139	0.9211 ± 0.0405	0.9489 ± 0.0220	0.9252 ± 0.0353
	(16, 1)	0.9786 ± 0.0064	0.9719 ± 0.0084	0.9216 ± 0.0275	0.9503 ± 0.0151	0.9250 ± 0.0250
	(16, 2)	0.9783 ± 0.0061	0.9690 ± 0.0120	0.9196 ± 0.0268	0.9490 ± 0.0148	0.9236 ± 0.0244
	(16, 3)	0.9786 ± 0.0070	0.9716 ± 0.0087	0.9236 ± 0.0312	0.9503 ± 0.0167	0.9269 ± 0.0280
	(16, 4)	0.9750 ± 0.0075	0.9673 ± 0.0116	0.9078 ± 0.0346	0.9417 ± 0.0189	0.9128 ± 0.0296

Table A2. *Cont.*

Video	(J,R)	ACCw	F1w	GMw	MCCnw	BMnw
V1	(64, −)	0.9790 ± 0.0064	0.9706 ± 0.0118	0.9256 ± 0.0278	0.9510 ± 0.0157	0.9288 ± 0.0254
	(64, 1)	0.9743 ± 0.0130	0.9634 ± 0.0211	0.9038 ± 0.0692	0.9385 ± 0.0360	0.9111 ± 0.0507
	(64, 2)	0.9745 ± 0.0105	0.9649 ± 0.0157	0.9066 ± 0.0476	0.9398 ± 0.0270	0.9124 ± 0.0395
	(64, 3)	0.9766 ± 0.0067	0.9682 ± 0.0084	0.9152 ± 0.0299	0.9454 ± 0.0159	0.9192 ± 0.0270
	(64, 4)	0.9775 ± 0.0074	0.9696 ± 0.0085	0.9197 ± 0.0334	0.9475 ± 0.0174	0.9234 ± 0.0299
V2	(2, −)	0.9626 ± 0.0034	0.9525 ± 0.0053	0.8602 ± 0.0286	0.8815 ± 0.0131	0.8687 ± 0.0246
	(2, 1)	0.9602 ± 0.0056	0.9518 ± 0.0084	0.8438 ± 0.0517	0.8730 ± 0.0233	0.8555 ± 0.0434
	(2, 2)	0.9596 ± 0.0053	0.9443 ± 0.0149	0.8363 ± 0.0625	0.8694 ± 0.0249	0.8508 ± 0.0494
	(2, 3)	0.9579 ± 0.0052	0.9448 ± 0.0114	0.8369 ± 0.0412	0.8648 ± 0.0217	0.8493 ± 0.0337
	(2, 4)	0.9625 ± 0.0029	0.9522 ± 0.0051	0.8630 ± 0.0280	0.8818 ± 0.0112	0.8715 ± 0.0242
V2	(16, −)	0.9684 ± 0.0046	0.9612 ± 0.0063	0.8700 ± 0.0405	0.8993 ± 0.0168	0.8785 ± 0.0351
	(16, 1)	0.9690 ± 0.0054	0.9621 ± 0.0076	0.8671 ± 0.0372	0.9005 ± 0.0190	0.8758 ± 0.0320
	(16, 2)	0.9682 ± 0.0040	0.9600 ± 0.0062	0.8633 ± 0.0311	0.8976 ± 0.0143	0.8724 ± 0.0266
	(16, 3)	0.9682 ± 0.0046	0.9615 ± 0.0057	0.8642 ± 0.0328	0.8979 ± 0.0163	0.8729 ± 0.0283
	(16, 4)	0.9664 ± 0.0052	0.9586 ± 0.0078	0.8441 ± 0.0447	0.8906 ± 0.0200	0.8565 ± 0.0371
V2	(64, −)	0.9704 ± 0.0044	0.9637 ± 0.0061	0.8732 ± 0.0296	0.9054 ± 0.0151	0.8809 ± 0.0259
	(64, 1)	0.9674 ± 0.0049	0.9586 ± 0.0115	0.8522 ± 0.0505	0.8942 ± 0.0207	0.8638 ± 0.0386
	(64, 2)	0.9685 ± 0.0040	0.9616 ± 0.0062	0.8665 ± 0.0304	0.8992 ± 0.0140	0.8751 ± 0.0259
	(64, 3)	0.9671 ± 0.0041	0.9603 ± 0.0059	0.8599 ± 0.0448	0.8951 ± 0.0160	0.8698 ± 0.0380
	(64, 4)	0.9692 ± 0.0037	0.9621 ± 0.0054	0.8713 ± 0.0453	0.9019 ± 0.0151	0.8798 ± 0.0387
V3	(2, −)	0.9475 ± 0.0074	0.9175 ± 0.0185	0.8752 ± 0.0491	0.8998 ± 0.0178	0.8851 ± 0.0419
	(2, 1)	0.9317 ± 0.0148	0.8884 ± 0.0281	0.7909 ± 0.0644	0.8630 ± 0.0327	0.8157 ± 0.0504
	(2, 2)	0.9377 ± 0.0128	0.8953 ± 0.0306	0.8096 ± 0.0600	0.8744 ± 0.0297	0.8323 ± 0.0417
	(2, 3)	0.9451 ± 0.0142	0.9163 ± 0.0253	0.8596 ± 0.0666	0.8945 ± 0.0309	0.8717 ± 0.0554
	(2, 4)	0.9452 ± 0.0158	0.9130 ± 0.0283	0.8605 ± 0.0754	0.8934 ± 0.0353	0.8726 ± 0.0635
V3	(16, −)	0.9635 ± 0.0091	0.9431 ± 0.0187	0.9192 ± 0.0393	0.9308 ± 0.0199	0.9236 ± 0.0350
	(16, 1)	0.9590 ± 0.0073	0.9401 ± 0.0124	0.9015 ± 0.0312	0.9226 ± 0.0145	0.9068 ± 0.0275
	(16, 2)	0.9620 ± 0.0078	0.9450 ± 0.0115	0.9163 ± 0.0337	0.9292 ± 0.0155	0.9202 ± 0.0299
	(16, 3)	0.9637 ± 0.0091	0.9470 ± 0.0136	0.9161 ± 0.0385	0.9320 ± 0.0180	0.9204 ± 0.0342
	(16, 4)	0.9621 ± 0.0091	0.9435 ± 0.0140	0.9074 ± 0.0327	0.9282 ± 0.0179	0.9125 ± 0.0288
V3	(64, −)	0.9614 ± 0.0090	0.9418 ± 0.0177	0.9054 ± 0.0362	0.9271 ± 0.0180	0.9110 ± 0.0306
	(64, 1)	0.9592 ± 0.0117	0.9362 ± 0.0243	0.8931 ± 0.0544	0.9216 ± 0.0259	0.9016 ± 0.0414
	(64, 2)	0.9620 ± 0.0114	0.9421 ± 0.0187	0.8999 ± 0.0441	0.9273 ± 0.0237	0.9066 ± 0.0363
	(64, 3)	0.9614 ± 0.0069	0.9398 ± 0.0116	0.8977 ± 0.0298	0.9265 ± 0.0134	0.9042 ± 0.0258
	(64, 4)	0.9596 ± 0.0068	0.9397 ± 0.0099	0.9014 ± 0.0284	0.9240 ± 0.0131	0.9069 ± 0.0251
V4	(2, −)	0.9492 ± 0.0036	0.9212 ± 0.0044	0.8336 ± 0.0126	0.8661 ± 0.0071	0.8466 ± 0.0100
	(2, 1)	0.9446 ± 0.0079	0.9143 ± 0.0122	0.8022 ± 0.0406	0.8506 ± 0.0246	0.8213 ± 0.0304
	(2, 2)	0.9475 ± 0.0052	0.9168 ± 0.0128	0.8105 ± 0.0343	0.8585 ± 0.0164	0.8289 ± 0.0247
	(2, 3)	0.9446 ± 0.0051	0.9146 ± 0.0096	0.8157 ± 0.0318	0.8531 ± 0.0151	0.8317 ± 0.0244
	(2, 4)	0.9477 ± 0.0043	0.9188 ± 0.0072	0.8280 ± 0.0104	0.8615 ± 0.0107	0.8413 ± 0.0088
V4	(16, −)	0.9481 ± 0.0046	0.9236 ± 0.0071	0.8526 ± 0.0256	0.8680 ± 0.0118	0.8621 ± 0.0201
	(16, 1)	0.9501 ± 0.0036	0.9252 ± 0.0061	0.8565 ± 0.0088	0.8723 ± 0.0074	0.8651 ± 0.0070
	(16, 2)	0.9498 ± 0.0042	0.9257 ± 0.0063	0.8555 ± 0.0169	0.8716 ± 0.0109	0.8643 ± 0.0138
	(16, 3)	0.9491 ± 0.0026	0.9254 ± 0.0030	0.8581 ± 0.0089	0.8708 ± 0.0043	0.8662 ± 0.0074
	(16, 4)	0.9499 ± 0.0035	0.9257 ± 0.0051	0.8494 ± 0.0199	0.8706 ± 0.0095	0.8591 ± 0.0162
V4	(64, −)	0.9493 ± 0.0043	0.9264 ± 0.0056	0.8636 ± 0.0106	0.8727 ± 0.0067	0.8711 ± 0.0083
	(64, 1)	0.9499 ± 0.0043	0.9267 ± 0.0087	0.8586 ± 0.0327	0.8728 ± 0.0134	0.8674 ± 0.0246
	(64, 2)	0.9498 ± 0.0044	0.9271 ± 0.0054	0.8623 ± 0.0091	0.8735 ± 0.0075	0.8700 ± 0.0075
	(64, 3)	0.9503 ± 0.0039	0.9272 ± 0.0047	0.8617 ± 0.0098	0.8742 ± 0.0059	0.8695 ± 0.0082
	(64, 4)	0.9483 ± 0.0054	0.9263 ± 0.0049	0.8635 ± 0.0142	0.8713 ± 0.0089	0.8708 ± 0.0116

Values highlighted in red denote the worst values achieved on every group. Values highlighted in blue denote the best values achieved on every group.

Table A3. Mean values of metrics achieved over 30 runs for vehicle-size classification, using different pairs of (J, R) values. The value $R = -$ denotes the results when the MV-DTF layer was employed, whereas the other values correspond to the LRMV-DTF layer. If the rank R value is not provided, the entire row corresponds to the results of the multitask single-view model.

Method	Metric				
	ACCw	**F1w**	**GMw**	**MCCnw**	**BMnw**
MT-SV (J = 8)	0.9483	0.9282	0.7878	0.8388	0.8087
MT-MV (J = 8, R = −)	0.9514	0.9286	0.8075	0.8494	0.8248
MT-MV (J = 8, R = 1)	0.9501	0.9279	0.8026	0.8455	0.8205
MT-MV (J = 8, R = 2)	0.9504	0.9285	0.8098	0.8476	0.8264
MT-MV (J = 8, R = 3)	0.9516	0.9288	0.8073	0.8498	0.8245
MT-SV (J = 10)	0.9493	0.9290	0.7909	0.8420	0.8112
MT-MV (J = 10, R = −)	0.9504	0.9276	0.8091	0.8471	0.8257
MT-MV (J = 10, R = 1)	0.9499	0.9273	0.8063	0.8456	0.8235
MT-MV (J = 10, R = 2)	0.9503	0.9282	0.8006	0.8490	0.8190
MT-MV (J = 10, R = 3)	0.9508	0.9284	0.8045	0.8476	0.8222
MT-SV (J = 12)	0.9483	0.9285	0.7874	0.8389	0.8084
MT-MV (J = 12, R = −)	0.9510	0.9283	0.8089	0.8487	0.8256
MT-MV (J = 12, R = 1)	0.9512	0.9288	0.8089	0.8492	0.8257
MT-MV (J = 12, R = 2)	0.9498	0.9272	0.8065	0.8455	0.8237
MT-MV (J = 12, R = 3)	0.9511	0.9279	0.7984	0.8469	0.8176

Values highlighted in red denote the worst values achieved on every group. Values highlighted in blue denote the best values achieved on every group.

References

1. Hou, Z.; Chen, Y. A real time vehicle collision detecting and reporting system based on internet of things technology. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1135–1139.
2. Ijjina, E.P.; Chand, D.; Gupta, S.; Goutham, K. Computer vision-based accident detection in traffic surveillance. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
3. Niu, Y.; Zhang, Y.; Li, L. Road Monitoring and Traffic Control System Design. In Proceedings of the 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 19–20 December 2009; pp. 1–4.
4. Desai, Y.; Rungta, Y.; Reshamwala, P. Automatic Traffic Management and Surveillance System. In Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 30–31 October 2020; pp. 131–133.
5. Chan, M.N.; Tint, T. A Review on Advanced Detection Methods in Vehicle Traffic Scenes. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 642–649.
6. Velazquez-Pupo, R.; Sierra-Romero, A.; Torres-Roman, D.; Shkvarko, Y.V.; Santiago-Paz, J.; Gómez-Gutiérrez, D.; Robles-Valdez, D.; Hermosillo-Reynoso, F.; Romero-Delgado, M. Vehicle Detection with Occlusion Handling, Tracking, and OC-SVM Classification: A High Performance Vision-Based System. *Sensors* **2018**, *18*, 374. [CrossRef] [PubMed]
7. Buch, N.; Velastin, S.A.; Orwell, J. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 920–939. [CrossRef]
8. Hsieh, J.W.; Yu, S.H.; Chen, Y.S.; Hu, W.F. Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 175–187. [CrossRef]
9. Moussa, G.S. Vehicle type classification with geometric and appearance attributes. *Int. J. Archit. Environ. Eng.* **2014**, *8*, 277–282.
10. Chen, Z.; Pears, N.; Freeman, M.; Austin, J. A Gaussian mixture model and support vector machine approach to vehicle type and colour classification. *ET Intell. Transp. Syst.* **2014**, *8*, 135–144. [CrossRef]
11. Al Okaishi, W.A.H.B.A.N.; Zaarane, A.; Slimani, I.; Atouf, I.; Benrabh, M. A Traffic Surveillance System in Real-Time to Detect and Classify Vehicles by Using Convolutional Neural Network. In Proceedings of the 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security, Casablanca, Morocco, 12–13 December 2019; pp. 1–5.
12. Wu, Z.; Sang, J.; Zhang, Q.; Xiang, H.; Cai, B.; Xia, X. Multi-Scale Vehicle Detection for Foreground-Background Class Imbalance with Improved YOLOv2. *Sensors* **2019**, *19*, 3336. [CrossRef]
13. Chen, X.-Z.; Chang, C.-M.; Yu, C.-W.; Chen, Y.-L. A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining. *Sensors* **2020**, *20*, 5731. [CrossRef]
14. Chen, Y.; Hu, W. Robust Vehicle Detection and Counting Algorithm Adapted to Complex Traffic Environments with Sudden Illumination Changes and Shadows. *Sensors* **2020**, *20*, 2686. [CrossRef] [PubMed]

15. Meng, T.; Jing, X.; Yan, Z.; Pedrycz, W. A survey on machine learning for data fusion. *Inf. Fusion* **2020**, *57*, 115–129. [CrossRef]
16. Castanedo, F. A review of data fusion techniques. *Sci. World J.* **2013**, *2013*, 704504. [CrossRef]
17. Polikar, R. Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45. [CrossRef]
18. Lahat, D.; Adali, T.; Jutten, C. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proc. IEEE* **2015**, *103*, 1449–1477. [CrossRef]
19. Lahat, D.; Adalý, T.; Jutten, C. Challenges in multimodal data fusion. In Proceedings of the 2014 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 1–5 September 2014; pp. 101–105.
20. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [CrossRef]
21. Yan, X.; Hu, S.; Mao, Y.; Ye, Y.; Yu, H. Deep multi-view learning methods: A review. *Neurocomputing* **2021**, *448*, 106–129. [CrossRef]
22. Xu, C.; Tao, D.; Xu, C. A survey on multi-view learning. *arXiv* **2013**, arXiv:1304.5634.
23. Zhang, Q.; Zhang, L.; Du, B.; Zheng, W.; Bian, W.; Tao, D. MMFE: Multitask multiview feature embedding. In Proceedings of the 2015 IEEE International Conference on Data Mining, Atlantic City, NJ, USA, 14–17 November 2015; pp. 1105–1110.
24. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
25. Zhang, Y.; Yang, Q. A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5586–5609. [CrossRef]
26. Zhang, Y.; Yang, Q. An overview of multi-task learning. *Natl. Sci. Rev.* **2018**, *5*, 30–43. [CrossRef]
27. Crawshaw, M. Multi-task learning with deep neural networks: A survey. *arXiv* **2020**, arXiv:2009.09796.
28. Yang, Y.; Hospedales, T. Deep multi-task representation learning: A tensor factorisation approach. *arXiv* **2016**, arXiv:1605.06391.
29. Fausett, L.V. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*; Pearson Education India: New Delhi, India, 2006.
30. Kolda, T.; Bader, B. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
31. Cong, F.; Lin, Q.H.; Kuang, L.D.; Gong, X.F.; Astikainen, P.; Ristaniemi, T. Tensor decomposition of EEG signals: A brief review. *J. Neurosci. Methods* **2015**, *248*, 59–69. [CrossRef] [PubMed]
32. López, J.; Torres, D.; Santos, S.; Atzberger, C. Spectral Imagery Tensor Decomposition for Semantic Segmentation of Remote Sensing Data through Fully Convolutional Networks. *Remote Sens.* **2020**, *12*, 517. [CrossRef]
33. Wimalawarne, K.; Sugiyama, M.; Tomioka, R. Multitask learning meets tensor factorization: Task imputation via convex optimization. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2825–2833.
34. Romera-Paredes, B.; Aung, H.; Bianchi-Berthouze, N.; Pontil, M. Multilinear multitask learning. *Int. Conf. Mach. Learn.* **2013**, *28*, 1444–1452.
35. Zhang, Z.; Xie, Y.; Zhang, W.; Tang, Y.; Tian, Q. Tensor multi-task learning for person re-identification. *IEEE Trans. Image Process.* **2019**, *29*, 2463–2477. [CrossRef]
36. Cao, B.; He, L.; Kong, X.; Philip, S.Y.; Hao, Z.; Ragin, A.B. Tensor-based multi-view feature selection with applications to brain diseases. In Proceedings of the 2014 IEEE International Conference on Data Mining, Shenzhen, China, 14–17 December 2014; pp. 40–49.
37. Sidiropoulos, N.D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* **2017**, *65*, 3551–3582. [CrossRef]
38. Vasilescu, M.A.O.; Terzopoulos, D. Multilinear analysis of image ensembles: Tensorfaces. In Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002; pp. 447–460.
39. de Almeida, A.L.; Favier, G.; Mota, J.C.M. PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. *Signal Process.* **2007**, *87*, 337–351. [CrossRef]
40. da Costa, M.N.; Favier, G.; Romano, J.M.T. Tensor modelling of MIMO communication systems with performance analysis and Kronecker receivers. *Signal Process.* **2018**, *145*, 304–316. [CrossRef]
41. Zhang, W.; Wu, Q.J.; Yang, X.; Fang, X. Multilevel framework to detect and handle vehicle occlusion. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 161–174. [CrossRef]
42. Pang, C.C.C.; Lam, W.W.L.; Yung, N.H.C. A novel method for resolving vehicle occlusion in a monocular traffic-image sequence. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 129–141. [CrossRef]
43. Wu, B.F.; Kao, C.C.; Jen, C.L.; Li, Y.F.; Chen, Y.H.; Juang, J.H. A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking. *IEEE Trans. Ind. Electron.* **2013**, *61*, 4228–4237. [CrossRef]
44. Yung, N.H.; Lai, A.H. Detection of vehicle occlusion using a generalized deformable model. *Detect. Veh. Occlusion Using Gen. Deform. Model* **1998**, *4*, 154–157.
45. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Vision-based occlusion handling and vehicle classification for traffic surveillance systems. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 80–92. [CrossRef]
46. Phan, H.N.; Pham, L.H.; Tran, D.N.N.; Ha, S.V.U. Occlusion vehicle detection algorithm in crowded scene for traffic surveillance system. In Proceedings of the 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, 21–23 July 2017; pp. 215–220.
47. Heidari, V.; Ahmadzadeh, M.R. A method for vehicle classification and resolving vehicle occlusion in traffic images. In Proceedings of the 2013 First Iranian Conference on Pattern Recognition and Image Analysis (PRIA), Birjand, Iran, 6–8 March 2013; pp. 1–6.

48. Ke, L.; Tai, Y.W.; Tang, C.K. Deep Occlusion-Aware Instance Segmentation with Overlapping BiLayers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4019–4028.
49. Qi, J.; Gao, Y.; Hu, Y.; Wang, X.; Liu, X.; Bai, X.; Belongie, S.; Yuille, A.; Torr, P.H.S.; Bai, S. Occluded Video Instance Segmentation: A Benchmark. *arXiv* **2021**, arXiv:2102.01558. [CrossRef]
50. Saleh, K.; Szénási, S.; Vámossy, Z. Occlusion Handling in Generic Object Detection: A Review. In Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia, 21–23 January 2021; pp. 000477–000484.
51. Yuan, X.; Kortylewski, A.; Sun, Y.; Yuille, A. Robust Instance Segmentation through Reasoning about Multi-Object Occlusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11141–11150.
52. Feng, P.; She, Q.; Zhu, L.; Li, J.; Zhang, L.; Feng, Z.; Wang, C.; Li, C.; Kang, X.; Ming, A. MT-ORL: Multi-Task Occlusion Relationship Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9364–9373.
53. Zhan, X.; Pan, X.; Dai, B.; Liu, Z.; Lin, D.; Loy, C.C. Self-Supervised Scene De-Occlusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3784–3792.
54. Yan, X.; Wang, F.; Liu, W.; Yu, Y.; He, S.; Pan, J. Visualizing the Invisible: Occluded Vehicle Segmentation and Recovery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7618–7627.
55. Lin, J.P.; Sun, M.T. A YOLO-based traffic counting system. In Proceedings of the 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taichung, Taiwan, 30 November–2 December 2018; pp. 82–85.
56. Kim, K.J.; Park, S.M.; Choi, Y.J. Deciding the number of color histogram bins for vehicle color recognition. In Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan, 9–12 December 2008; pp. 134–138.
57. Ge, P.; Hu, Y. Vehicle Type Classification based on Improved HOG_SVM. In Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019), Dalian, China, 29–30 March 2019; pp. 640–647.
58. Kim, J.A.; Sung, J.Y.; Park, S.H. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Republic of Korea, 1–3 November 2020; pp. 1–4.
59. Naik, U.P.; Rajesh, V.; Kumar, R. Implementation of YOLOv4 algorithm for multiple object detection in image and video dataset using deep learning and artificial intelligence for urban traffic video surveillance application. In Proceedings of the 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 15–17 September 2021; pp. 1–6.
60. Pavani, K.; Sriramya, P. Comparison of KNN, ANN, CNN and YOLO algorithms for detecting the accurate traffic flow and build an Intelligent Transportation System. In Proceedings of the 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 23–25 February 2022; Volume 2, pp. 628–633.
61. Zhao, Y.; Lu, Z. Ford Vehicle Identification based on gray-level co-occurrence matrix and genetic neural network. In Proceedings of the 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity, London, UK, 30–31 July 2019; pp. 275–279.
62. Leotta, M.J.; Mundy, J.L. Vehicle surveillance with a generic, adaptive, 3d vehicle model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1457–1469. [CrossRef]
63. Sochor, J.; Herout, A.; Havel, J. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3006–3015.
64. Prokaj, J.; Medioni, G. 3-D model based vehicle recognition. In Proceedings of the 2009 Workshop on Applications of Computer Vision, Snowbird, UT, USA, 7–8 December 2009; pp. 1–7.
65. Shahin, O.R.; Alruily, M. Vehicle Identification Using Eigenvehicles. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies, Coimbatore, India, 20–22 February 2019; pp. 1–6.
66. Shi, C.; Wu, C. Vehicle Face Recognition Algorithm Based on Weighted Nonnegative Matrix Factorization with Double Regularization Terms. *Ksii Trans. Internet Inf. Syst.* **2020**, *14*, 2171–2185.
67. Ban, J.M.; Lee, B.R.; Kang, H.C. Vehicle recognition using NMF in urban scene. *J. Korean Inst. Commun. Inf. Sci.* **2012**, *37*, 554–564.
68. Ban, J.M.; Kang, H. Vehicle Recognition using Non-negative Tensor Factorization. *J. Inst. Electron. Inf. Eng.* **2015**, *52*, 136–146.
69. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
70. Wang, K.; Liu, Y.; Gou, C.; Wang, F.Y. A multi-view learning approach to foreground detection for traffic surveillance applications. *IEEE Trans. Veh. Technol.* **2015**, *65*, 4144–4158. [CrossRef]
71. Guo, H.; Wang, J.; Xu, M.; Zha, Z.J.; Lu, H. Learning multi-view deep features for small object retrieval in surveillance scenarios. In Proceedings of the 23rd ACM International Conference on Multimedia, New York, NY, USA, 26–30 October 2015; pp. 859–862.
72. Chu, W.; Liu, Y.; Shen, C.; Cai, D.; Hua, X.S. Multi-task vehicle detection with region-of-interest voting. *IEEE Trans. Image Process.* **2017**, *27*, 432–441. [CrossRef] [PubMed]

73. Oeljeklaus, M.; Hoffmann, F.; Bertram, T. A fast multi-task CNN for spatial understanding of traffic scenes. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2825–2830.
74. Liu, S.; Johns, E.; Davison, A.J. End-to-end multi-task learning with attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1871–1880.
75. Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; Morency, L.P. Tensor fusion network for multimodal sentiment analysis. *arXiv* **2017**, arXiv:1707.07250.
76. Liu, Z.; Shen, Y.; Lakshminarasimhan, V.B.; Liang, P.P.; Zadeh, A.; Morency, L.P. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv* **2018**, arXiv:1806.00064.
77. Guo, Y.; Zhang, C.; Zhang, C.; Chen, Y. Sparse dnns with improved adversarial robustness. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 240–249.
78. Oh, Y.H.; Quan, Q.; Kim, D.; Kim, S.; Heo, J.; Jung, S.; Jang, J.; Lee, J.W. A portable, automatic data qantizer for deep neural networks. In Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques, Limassol, Cyprus, 1–4 November 2018; pp. 1–14.
79. Denil, M.; Shakibi, B.; Dinh, L.; Ranzato, M.A.; De Freitas, N. Predicting parameters in deep learning. *Adv. Neural Inf. Process. Systems.* **2013**, *26*, 1–9.
80. Mai, A.; Tran, L.; Tran, L.; Trinh, N. VGG deep neural network compression via SVD and CUR decomposition techniques. In Proceedings of the 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), Ho Chi Minh City, Vietnam, 26–27 November 2020; pp. 118–123.
81. Jaderberg, M.; Vedaldi, A.; Zisserman, A. Speeding up convolutional neural networks with low rank expansions. *arXiv* **2014**, arXiv:1405.3866.
82. Lebedev, V.; Ganin, Y.; Rakhuba, M.; Oseledets, I.; Lempitsky, V. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv* **2014**, arXiv:1412.6553.
83. Kim, Y.D.; Park, E.; Yoo, S.; Choi, T.; Yang, L.; Shin, D. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv* **2015**, arXiv:1511.06530.
84. Tukan, M.; Maalouf, A.; Weksler, M.; Feldman, D. No fine-tuning, no cry: Robust svd for compressing deep networks. *Sensors* **2021**, *21*, 5599. [CrossRef]
85. Tai, C.; Xiao, T.; Zhang, Y.; Wang, X. Convolutional neural networks with low-rank regularization. *arXiv* **2015**, arXiv:1511.06067.
86. Xu, Y.; Li, Y.; Zhang, S.; Wen, W.; Wang, B.; Dai, W.; Qi, Y.; Qi, Y.; Lin, W.; Xiong, H. Trained rank pruning for efficient deep neural networks. In Proceedings of the 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS), Vancouver, BC, Canada, 13 December 2019; pp. 14–17.
87. Novikov, A.; Podoprikhin, D.; Osokin, A.; Vetrov, D.P. Tensorizing neural networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 442–450.
88. Newman, E.; Horesh, L.; Avron, H.; Kilmer, M. Stable tensor neural networks for rapid deep learning. *arXiv* **2018**, arXiv:1811.06569.
89. Lee, D.; Kwon, S.J.; Kim, B.; Wei, G.Y. Learning low-rank approximation for cnns. *arXiv* **2019**, arXiv:1905.10145.
90. Padilla-Zepeda, E.; Torres-Roman, D.; Mendez-Vazquez, A. A Semantic Segmentation Framework for Hyperspectral Imagery Based on Tucker Decomposition and 3DCNN Tested with Simulated Noisy Scenarios. *Remote Sens.* **2023**, *15*, 1399. [CrossRef]
91. Kossaifi, J.; Lipton, Z.C.; Kolbeinsson, A.; Khanna, A.; Furlanello, T.; Anandkumar, A. Tensor regression networks. *J. Mach. Learn. Res.* **2020**, *21*, 4862–4882.
92. Zhu, J.; Li, X.; Jin, P.; Xu, Q.; Sun, Z.; Song, X. Mme-yolo: Multi-sensor multi-level enhanced yolo for robust vehicle detection in traffic surveillance. *Sensors* **2021**, *21*, 27. [CrossRef]
93. Cao, X.; Wu, C.; Yan, P.; Li, X. Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2421–2424.
94. Wen, X.; Yuan, H.; Yang, C.; Song, C.; Duan, B.; Zhao, H. Improved Haar wavelet feature extraction approaches for vehicle detection. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Bellevue, WA, USA, 30 September–3 October 2007; pp. 1050–1053.
95. Kuang, H.; Chen, L.; Chan, L.L.H.; Cheung, R.C.; Yan, H. Feature selection based on tensor decomposition and object proposal for night-time multiclass vehicle detection. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 71–80. [CrossRef]
96. Wang, W.; Zhang, M. Tensor deep learning model for heterogeneous data fusion in Internet of Things. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *4*, 32–41. [CrossRef]
97. Brazell, M.; Li, N.; Navasca, C.; Tamon, C. Solving multilinear systems via tensor inversion. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 542–570. [CrossRef]
98. Rogers, M.; Li, L.; Russell, S.J. Multilinear dynamical systems for tensor time series. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
99. Chen, C.; Surana, A.; Bloch, A.; Rajapakse, I. Multilinear time invariant system theory. In Proceedings of the 2019 Conference on Control and its Applications, Chengdu, China, 19–21 June 2019; pp. 118–125.
100. Pandey, D.; Leib, H. A tensor framework for multi-linear complex MMSE estimation. *IEEE Open J. Signal Process.* **2021**, *2*, 336–358. [CrossRef]

101. Greub, W.H. Tensor algebra. In *Multilinear Algebra*; Springer: Berlin/Heidelberg, Germany, 1978; pp. 60–83.
102. Lim, L.H. Tensors in computations. *Acta Numerica.* **2021**, *30*, 555–764. [CrossRef]
103. Panigrahy, K.; Mishra, D. Extension of Moore–Penrose inverse of tensor via Einstein product. *Linear Multilinear Algebra* **2022**, *70*, 750–773. [CrossRef]
104. Sagiroglu, S.; Sinanc, D. Big data: A review. In Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 20–24 May 2013; pp. 42–47.
105. Fan, J.; Han, F.; Liu, H. Challenges of big data analysis. *Natl. Sci. Rev.* **2014**, *1*, 293–314. [CrossRef]
106. Lu, H.; Plataniotis, K.N.; Venetsanopoulos, A.N. A survey of multilinear subspace learning for tensor data. *Pattern Recognit.* **2011**, *44*, 1540–1551. [CrossRef]
107. De La Torre, F.; Black, M.J. A framework for robust subspace learning. *Int. J. Comput. Vis.* **2003**, *54*, 117–142. [CrossRef]
108. Pearson, K. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
109. Spearman, C. "General Intelligence" Objectively Determined and Measured. *Am. J. Psychol.* **1904**, *15*, 201–292. [CrossRef]
110. Hyvärinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430. [CrossRef]
111. Hotelling, H. Relations between two sets of variates. In *Breakthroughs in Statistics*; Springer: New York, NY, USA, 1992; pp. 162–190.
112. Eckart, C.; Young, G. The approximation of one matrix by another of lower rank. *Psychometrika* **1936**, *1*, 211–218. [CrossRef]
113. Klecka, W.R.; Iversen, G.R.; Klecka, W.R. *Discriminant Analysis*; Sage: Thousand Oaks, CA, USA, 1980; Volume 19.
114. Sener, O.; Koltun, V. Multi-task learning as multi-objective optimization. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 525–536.
115. Samet, H.; Tamminen, M. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 579–586. [CrossRef]
116. Dillencourt, M.B.; Samet, H.; Tamminen, M. A general approach to connected-component labeling for arbitrary image representations. *J. ACM* **1992**, *39*, 253–280. [CrossRef]
117. Dougherty, E.R.; Lotufo, R.A. *Hands-On Morphological Image Processing*; SPIE Press: Bellingham, WA, USA, 2003; p. 59.
118. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2017**, *32*, 145–163. [CrossRef]
119. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **2021**, *64*, 107–115. [CrossRef]
120. Vervliet, N.; Debals, O.; Sorber, L.; De Lathauwer, L. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Process. Mag.* **2014**, *31*, 71–79. [CrossRef]
121. Phan, A.; Sobolev, K.; Sozykin, K.; Ermilov, D.; Gusak, J.; Tichavsky, P.; Glukhov, V.; Oseledets, I.; Cichocki, A. Stable low-rank tensor decomposition for compression of convolutional neural network. In Proceedings of the Computer Vision–ECCV 2020, 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXIX 16; Springer: Cham, Switzerland, 2020; pp. 522–539.
122. Fernando Hermosillo Reynoso—Youtube. Available online: https://www.youtube.com/watch?v=ZWWX4nojMos&list=PLKng1hWmrHM2wWBQXrA8zxOoPzFSj-Upj (accessed on 20 November 2023).
123. Fernando Hermosillo Reynoso—Github. Available online: https://github.com/fhermosillo/TDF (accessed on 20 November 2023).
124. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
125. Markoulidakis, I.; Kopsiaftis, G.; Rallis, I.; Georgoulas, I. Multi-class confusion matrix reduction method and its application on net promoter score classification problem. In Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference, Corfu, Greece, 29 June–2 July 2021; pp. 412–419.
126. Gonzalez-Ramirez, A.; Lopez, J.; Torres-Roman, D.; Yañez-Vargas, I. Analysis of multi-class classification performance metrics for remote sensing imagery imbalanced datasets. *J. Quant. Stat. Anal.* **2021**, *8*, 11–17. [CrossRef]
127. Luque, A.; Carrasco, A.; Martín, A.; de Las Heras, A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.* **2019**, *91*, 216–231. [CrossRef]
128. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]

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