sensors

MDPI

Article

Dual-View Single-Shot Multibox Detector at Urban Intersections: Settings and Performance Evaluation

Marta Lenatti [1,†], Sara Narteni [1,2,†], Alessia Paglialonga [1], Vittorio Rampa [1] and Maurizio Mongelli [1,*]

[1] CNR-IEIIT, 10129 Turin, Italy
[2] Department of Control and Computer Engineering (DAUIN), Politecnico di Torino, 10129 Turin, Italy
* Correspondence: maurizio.mongelli@ieiit.cnr.it
† These authors contributed equally to this work.

Abstract: The explosion of artificial intelligence methods has paved the way for more sophisticated smart mobility solutions. In this work, we present a multi-camera video content analysis (VCA) system that exploits a single-shot multibox detector (SSD) network to detect vehicles, riders, and pedestrians and triggers alerts to drivers of public transportation vehicles approaching the surveilled area. The evaluation of the VCA system will address both detection and alert generation performance by combining visual and quantitative approaches. Starting from a SSD model trained for a single camera, we added a second one, under a different field of view (FOV) to improve the accuracy and reliability of the system. Due to real-time constraints, the complexity of the VCA system must be limited, thus calling for a simple multi-view fusion method. According to the experimental test-bed, the use of two cameras achieves a better balance between precision (68%) and recall (84%) with respect to the use of a single camera (i.e., 62% precision and 86% recall). In addition, a system evaluation in temporal terms is provided, showing that missed alerts (false negatives) and wrong alerts (false positives) are typically transitory events. Therefore, adding spatial and temporal redundancy increases the overall reliability of the VCA system.

Keywords: smart mobility; object detection; video content analysis; single-shot multibox detector

check for updates

Citation: Lenatti, M.; Narteni, S.; Paglialonga, A.; Rampa, V.; Mongelli, M. Dual-View Single-Shot Multibox Detector at Urban Intersections: Settings and Performance Evaluation. *Sensors* **2023**, *23*, 3195. https://doi.org/10.3390/s23063195

Academic Editor: Chih-Yang Lin

Received: 16 February 2023
Revised: 14 March 2023
Accepted: 15 March 2023
Published: 16 March 2023

1. Introduction

Nowadays, the *smart city* paradigm is changing the asset of the urban environment thanks to the rapid growth of digital technologies and communication infrastructures. By interconnecting people and things, smart cities scenarios provide more efficient, fast, ubiquitous, and accessible services to citizens [1]. In this context, *smart mobility* applications are empowered by the high speed and low latency properties of 5G networks [2], being suitable for ensuring road safety [3] and monitoring dangerous situations [4]. The huge amount of sensor data and the availability of fast computing resources at the edge of the 5G networks have paved the way to advanced deep learning (DL) models for real-time video content analysis (VCA) scenarios [5].

Both real-time localization and object classification methods from video streams are mandatory requirements for VCA solutions. To this aim, different DL architectures based on convolutional neural networks (CNNs) have recently been proposed [6]. However, among the most widely exploited approaches, you-only-look-once (YOLO) and single-shot multibox detectors (SSD) algorithms stand out for their performance and computing efficiency [7]: the former is indeed one of the fastest and most accurate networks for real-time object detection [8], while the latter is a benchmark for real-time multi-class object detection at different scales [9].

In this paper, we consider a driver alert scenario, where an urban intersection is monitored by two cameras and an SSD-based object detection model is trained to identify, localize and, eventually, signal the presence of obstacles to public transportation vehicles approaching the surveilled area. Particular focus will be given to investigating the advantages of using two cameras instead of a single one, in terms of object detection and alert generation performance. To this purpose, the VCA model will be evaluated using qualitative and tailored quantitative approaches, exploiting both spatial and temporal redundancy.

The paper is organized as follows. First, we discuss relevant literature on the topic. Then, we recall the SSD-based method adopted, and we thoroughly describe the on-field implementation. Finally, we present and discuss the results in terms of object detection performance and the related alert generation performance.

2. Related Works

Object detection and/or tracking via multiple camera sensors is a widespread topic in computer vision research. Multi-view 3D object recognition [10] consists in reducing complex 3D object classification tasks to simpler 2D classification tasks by rendering 3D objects into 2D images. Real objects are surrounded by cameras posed at different viewpoints with configurations leading to multi-view proposals, such as MVCNN [11], GVCNN [12], View-GCN [13] and RotationNet [14] architectures. These methods use the most successful image classification networks, i.e., VGG, GoogleNet, AlexNet, ResNet, as backbone networks. Then, global 3D shape descriptors are obtained by aggregating selected multi-view features through approaches that account for both content and spatial relationships between the views.

Transfer learning approaches prove extremely useful, especially when dealing with scarcely available data. To this end, several open source datasets for object detection in urban traffic optimization and management have recently become available. These datasets focus either on pedestrian or vehicle tracking and detection, combining inputs from multiple cameras and extending visual coverage (e.g., [15,16]).

An overview of recent multi-camera solutions for object detection is presented below. In [17], a novel multi-view region proposal network that infers the vehicles position on the ground plane by leveraging multi-view cross-camera scenarios is presented, whereas an end-to-end DL method for multi-camera people detection is studied in [18]. In [19], a vehicle detection method that applies transfer learning on two cameras with different focal length is proposed. The processing consists of two steps: first, a mapping relationship between input images from the cameras is calculated offline through a robust evolutionary algorithm; then, CNN-based object detection is performed online. More specifically, after a vehicle region is detected from one camera, it is transformed into a binary map. This map is then used to filter CNN feature maps computed for the other camera's image. It is important to outline that finding the relationship between the two cameras is crucial to solve the problem of duplicated detection, as different cameras may focus on the same vehicles. The same problem is raised in [20], where the authors present a novel edge-AI solution for vehicles counting in a parking area monitored via multiple cameras. They combine a CNN-based technique for object localization with a geometric approach aimed at analyzing the shared area between the cameras and merging data collected from them. Multi-camera object detection is also investigated in [21], which presents an autonomous drone detection and tracking system exploiting a static wide-angle camera and a lower-angle camera mounted on a rotating turret. In order to save computational resources and time, the frame coming from the second camera is overlaid on the static camera's frame. Then, a lightweight version of YOLOv3 detector is developed to perform the object detection. Another recent work on multi-camera fusion for CNN-based object classification [22] devised three fusion strategies: early, late and score fusion. A separate CNN was first trained on each camera. Afterward, feature maps were stacked together and processed either from the initial layers (early fusion) or at the penultimate layers (late fusion). In addition, score fusion was performed, by aggregating the softmax classification scores in three possible ways: by summing, or by multiplying, the scores across cameras, or by taking the maximum score across them. Results showed that late and score fusion led to an accuracy improvement, with respect to early fusion and single camera proposals. Multi-camera detection has gained increasing importance in several areas besides smart mobility applications. For example, several solutions have recently been proposed in the area of fall detection for remote monitoring of fragile patients. In [23], multi-camera fusion is performed by combining models trained on single cameras together into a global ensemble model at the decision-making level, providing higher accuracy with respect to local single-camera models and avoiding computationally expensive cameras calibration. The dual-stream fused neural network method, proposed in [24], first trains two deep neural networks to detect falls by using two single cameras and then merges the results through a weighted fusion of prediction scores. The obtained results overcome the existing methods in this domain.

All these proposals deal with high-intensity computational methods, while, on the contrary, real-time field-deployable applications impose computational complexity constraints as well. To solve this key issue, we propose here a simple but effective dual-view fusion and detection method and compare its performance with real field experiments [25]. In particular, our solution exploits a transfer learning approach, which consists in training the object detection model on a single camera, in updating it through an additional training by feeding the other camera's images, and then by fusing the single detection signals to generate alerts at the decision level. This speeds up the overall training time and saves computational resources, with respect to other existing decision-making level camera fusion approaches, such as [22,23].

3. Video Content Analysis System

3.1. Single-Shot Multibox Detector Model

The SSD network is composed of a *backbone* stage for feature extraction and a *head* stage for determining the output. The backbone is a feature pyramid network (FPN) [26], which is a CNN able to extract feature maps representing objects at different scales. It comprises a bottom-up pathway connected to a top-down pathway via lateral connections. The SSD head is a sequence of output maps, which determines the output of the network in the form of bounding box coordinates and object classes. Additionally, the SSD network exploits the concept of *priors* (also known as *anchor boxes*), a special kind of box whose predefined shape can guide the network to correctly detect objects of the desired class.

The SSD head is composed of multiple output maps (grids) with different sizes. Each grid decomposes the image into cells, and each cell expresses whether or not it belongs to a particular object, in terms of bounding box coordinates and object class. Lower resolution output maps (i.e., smaller size grids), having larger cells, can detect larger scale objects; in contrast, larger size output grids, having denser cells, are used to predict smaller objects. The use of multiple outputs improves the accuracy of the model significantly, while maintaining the ability to predict objects in real time.

3.1.1. Loss Function

The training of the SSD model is based on the minimization of the following loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{conf} + \mathcal{L}_{boxiness}, \tag{1}$$

where $\mathcal{L}_{loc}$ evaluates the object localization of the model, $\mathcal{L}_{conf}$ evaluates the object classification ability and $\mathcal{L}_{boxiness}$ term refers to the *boxiness*, i.e., the ability of discriminating boxes from background throughout SSD output grids.

Considering object localization, we define $\mathbf{y}_{gt} = (x, y, w, h)$ as the ground truth box coordinates vector for a generic object, with x, y expressing box center coordinates, w the box width and h the box height. Similarly, we denote with $\mathbf{y}_{pr} = (x_{pr}, y_{pr}, w_{pr}, h_{pr})$ the predicted box coordinates vector for that same object. A discrepancy between the real and predicted box positions is measured by the vector $\mathbf{a} \doteq |\mathbf{y}_{gt} - \mathbf{y}_{pr}|$, with coordinates $(a_1, a_2, a_3, a_4) = (|x - x_{pr}|, |y - y_{pr}|, |w - w_{pr}|, |h - h_{pr}|)$.

The $\mathcal{L}_{loc}$ term is then computed through the pseudo-Huber loss function [27]:

$$\mathcal{L}_{loc} = \sum_{i=1}^{4} \delta^2 \left(\sqrt{1 + (\frac{a_i}{\delta})^2} - 1 \right), \tag{2}$$

with δ being a fixed quantity that controls the steepness of the function. The pseudo-Huber loss provides the best performance, with minimal computational costs with respect to the Huber and other types of loss functions [28]. In this study, δ was set to 1.5, following preliminary training runs.

Referring to object classification, let y_c be the true class label for each class $c = 1, \ldots, N$, where N is the number of classes. Additionally, let $\hat{p}_c$ be the corresponding class probability estimates. The second loss term, $\mathcal{L}_{conf}$, is then a cross-entropy loss, computed as follows:

$$\mathcal{L}_{conf} = -\sum_{c=1}^{N} y_c \, log(\hat{p}_c) \tag{3}$$

After prediction, the SSD model also outputs an estimate of the boxiness, expressed as a real value $b_{pr} \in [0, 1]$, which can be interpreted as the model confidence in recognizing whether any object is present in each cell of the network output grids. Consequently, the quantity $b_{bg} = 1 - b_{pr}$ defines the level of confidence of each cell to be part of the background.

The last term $\mathcal{L}_{boxiness}$ relies on a focal loss function [29], which is chosen for its ability to penalize the false positives, i.e., the background points wrongly detected as objects by the model. The boxiness loss $\mathcal{L}_{boxiness}$ is then computed as

$$\mathcal{L}_{boxiness} = -\left[\alpha \, b_{bg}^{\gamma} log(b_{pr}) + (1 - \alpha) \, b_{pr}^{\gamma} \, log(b_{bg}) \right], \tag{4}$$

where the parameter α acts as a weight for those cells being covered by a box and $1 - \alpha$ acts as weight for the background cells; the parameter γ controls the shape of the function. Higher values of γ require lower loss values to better distinguish boxes from background (i.e., to have $b_{pr} > 0.5$). The attention of the model is thus devoted to the harder-to-detect samples.

3.1.2. Network Parameters, Training and Testing

Non-maximum suppression (NMS) [30] was performed to refine the predictions of the model. Indeed, it may often occur that multiple boxes are predicted for the same ground truth object. The NMS algorithm filters out the predicted boxes based on the class confidence and the intersection over union (IoU) method [31] between them. In particular, for a given SSD output grid and class, for each real object, the predicted box (if any) with the highest class confidence is picked. This box is then chosen as a reference to compute the IoU between itself and all the other predicted boxes, keeping only those with a value below a threshold. In our case, we fixed this threshold at 0.1. Choosing such a low value allows to filter out boxes characterized by even small overlaps with the reference one, therefore reducing the presence of false positives.

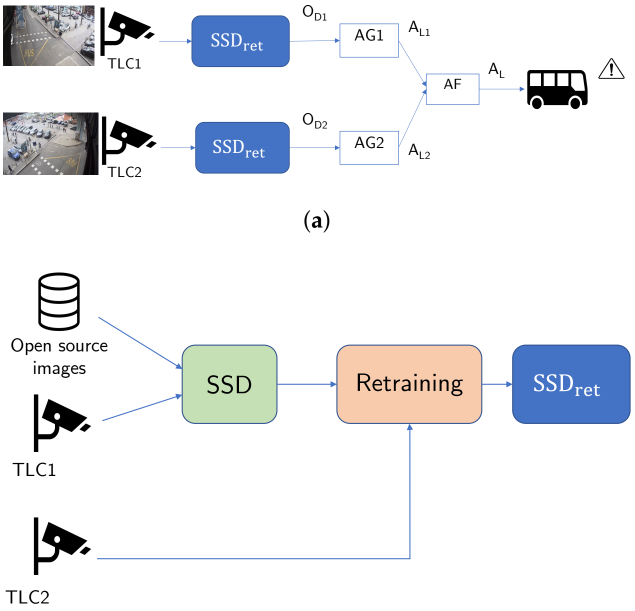
Table 1 summarizes the properties and parameters of the SSD model adopted in this work. The choice of the SSD output grids dimensions was guided by a preliminary analysis on a range of suitable values, performed to individuate a proper balance between model accuracy and computational complexity. Additionally, the selection of regions of interest from the foreground area, as better detailed in the scenario definition, required lower sized grids, able to capture bigger foreground objects. The network is trained to recognize three classes of objects: ‘vehicle’, ‘rider’, and ‘pedestrian’.

Table 1. Parameters and properties of the adopted SSD.

Output grids	24 × 40, 12 × 20, 6 × 10 and 3 × 5
Priors	1 × 1, 2 × 1, 4 × 1, 1 × 4 and 1 × 2
# trainable parameters	5000
Learning rate	10^{-4}
δ	1.5
α	0.85
γ	2
IoU threshold	0.1

3.2. VCA Architecture

We define here the main pipeline of the VCA system for alert generation, whose inference and training/retraining flowcharts are sketched in Figure 1. The first pipeline (Figure 1a) sketches the object detection blocks employed to generate alerts (inference phase) by exploiting image fusion on both cameras. The second pipeline (Figure 1b) focuses on retraining the baseline SSD by adding *TLC2* images via transfer learning, thus obtaining a final model, i.e., SSD_{ret}. More specifically, the inference block diagram shows the real-time processing pipeline adopted to generate the alarm signal A_L by fusing together the single-view alerts A_{L1} and A_{L2} produced by the alert generation blocks AG_1 and AG_2 that are fed by the output of the SSD_{ret} object detectors attached to the single camera $TLC1$ and $TLC2$, respectively. The two cameras have a broad field of view, but in order to define the area of potential danger to be monitored, a region of interest (ROI) is determined and adapted for each camera. The alert A_L is then employed to alert the driver by activating visual and acoustic alarms on the bus console. The final inference stage A_L is designed to integrate the two independent outputs of the single alert generators related to each camera view and to perform information fusion at the decision level with the aim of increasing the overall reliability and accuracy of the system.



(**a**)

(**b**)

Figure 1. Flowchart of the procedures exploited for the proposed VCA system, sketching the alert generation and fusion (**a**) based on model SSD_{ret}, obtained via a retraining process (**b**). (**a**) Inference. (**b**) Retraining.

Figure 1b shows the retraining procedure adopted to update the baseline SSD network of the single-view system (that uses only $TLC1$ data) by including also images from the $TLC2$ camera. In fact, the baseline SSD model (i.e., the green block in Figure 1b) is preliminarily trained on a set of images extracted from three open-source datasets (Open Images Dataset [32], ETH Pedestrian Dataset [33] and EuroCity Dataset [34]) that contain annotated images of urban traffic scenes. Afterward, the images captured by *TLC1* were added to these datasets to complete the training of the baseline SSD model. To further improve the flexibility, reliability and, in particular, the detection accuracy of the VCA system, the baseline SSD model was later retrained on a set of 10,000 additional images acquired from the *TLC2* camera. The term retraining refers to the procedure of updating the parameters of a previously trained model based on the addition of new data by transfer learning methods [35]. From now on, we will refer to the final retrained model as $\mathrm{SSD}_{\mathrm{ret}}$ (blue block in Figure 1b). The generalizing capabilities of the baseline SSD and the retrained $\mathrm{SSD}_{\mathrm{ret}}$ models were assessed using a test dataset consisting of frames extracted from a 1-h video, for both cameras. Both videos were first synchronized and cut to align the start and end time stamps, then converted from the h263 format to the mp4 format using the FFmpeg tool [36] (with compression factor 1.25). Finally, 1000 frames were extracted for each recording.

3.3. Data Labeling

YOLOv5x [37], one of the state-of-the-art YOLO networks for object detection in real-time applications, was adopted to define ground truth boxes, i.e., to label the objects actually present in each image. For this purpose, YOLOv5x was applied on each image of the training, retraining, and test datasets in order to recognize objects of the classes ‘Car’, ‘Bus’, ‘Truck’, ‘Motorcycle’, ‘Bicycle’, and ‘Pedestrian’. Then, these classes were grouped into three more generic classes, namely ‘Vehicle’, ‘Rider’, and ‘Pedestrian’. Ground truth boxes were provided in the YOLO format (x_{center}, y_{center}, width, height), and subsequently converted in the SSD format (x_{min}, y_{min}, width, height). The results of this automatic labeling step were then manually inspected to verify the presence of sufficiently accurate ground truth boxes. In the presence of detection errors inside the monitored area, the corresponding images were removed from the dataset. Based on the ground truth boxes, we also defined the number of ground truth alerts, which were raised any time at least one ground truth box was detected within the ROI.

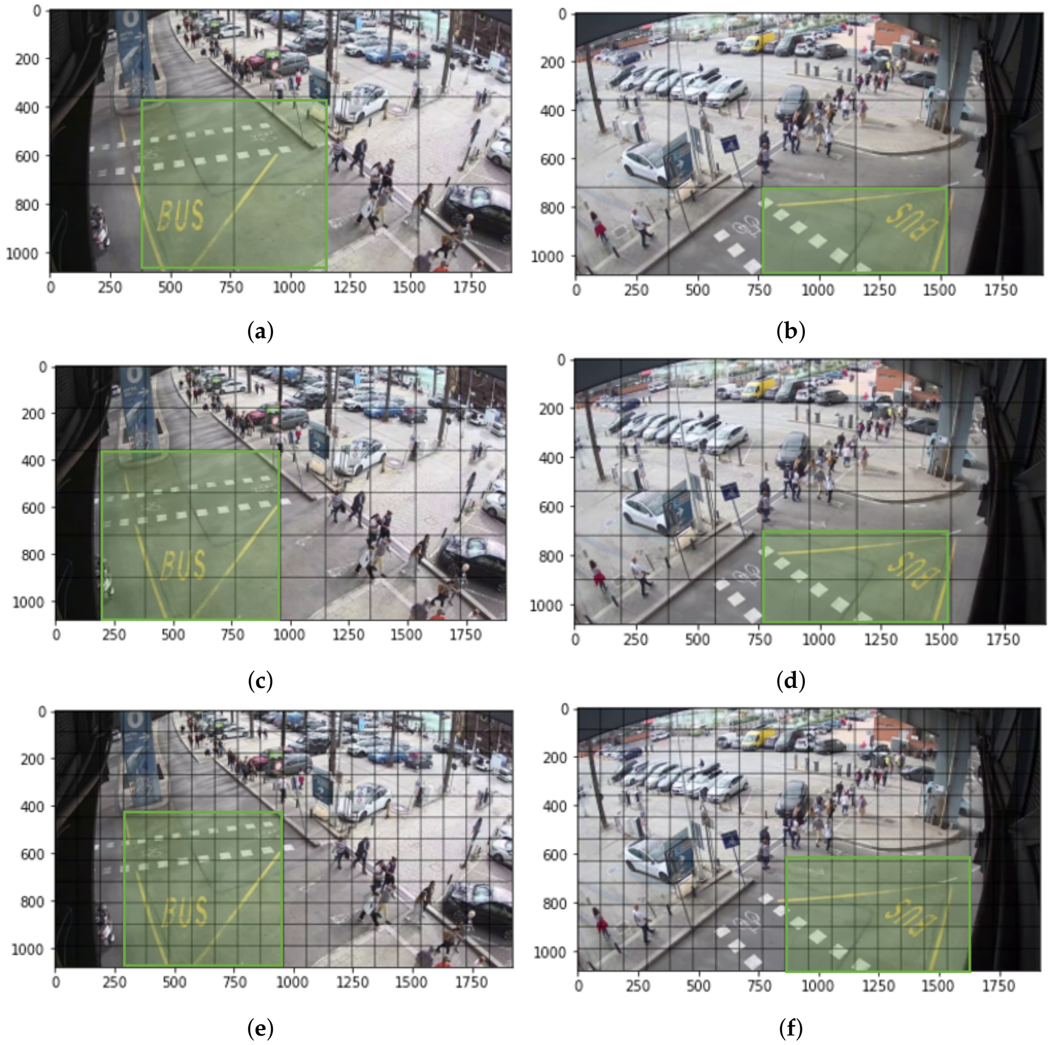
4. Driver Alert Use Case

4.1. Scenario Definition

Piazza Caricamento is one of the locations with the highest concentration of pedestrian and road traffic in the historic center of Genoa, Italy, as it connects the east and west areas of the city and, above all, it is located nearby the main tourist attractions (e.g., the aquarium, the pedestrian area on the harbor, and the most important architectural and artistic sites of the city). The area monitored by the proposed VCA system is the intersection between the pedestrian area of the harbor, the vehicular access to the parking lot, and the access roads to the underground tunnel below Piazza Caricamento corresponding to the latitude and longitude coordinates 44.4110720922656 N, 8.928069542327654 E (expressed in decimal degrees). A dedicated public transportation bus lane, which is characterized by limited visibility, interconnects with the monitored intersection. The area is often crowded with pedestrians and vehicles frequently passing through to access the car parking. Hence, potential collisions with buses coming from their dedicated lane represent a real risk scenario that makes Piazza Caricamento a suitable location to implement a VCA system. The proposed solution consists in an automatic system able to detect the presence of pedestrians and/or vehicles inside the area via VCA processing, and to generate an appropriate alert to the bus approaching the intersection. Real-time monitoring is performed via two Bosh DINION IP Bullet 6000I HD, 2, 8–12 MM cameras, which are professional surveillance HD cameras compliant with the SMPTE 296M-2001 standard [38] and ONVIF profiles G and

S [39] to guarantee the interoperability with the AI components. We will refer to these cameras as *TLC1* and *TLC2*.

As previously noted, only objects within each ROI of the cameras can generate an alert to be sent to the driver. As a result, the two ROIs strictly overlap. Since our SSD model involves multiple output grids, the ROI was resized for each of them based on their dimension. Figure 2 displays the fields of view covered by the two cameras and reports the selected ROIs for each adopted grid.



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

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

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

Figure 2. Regions of interest (ROIs) inside the monitored area (green rectangles), for each considered SSD output grid on *TLC1* (left column) and *TLC2* (right column). (**a**) ROI on *TLC1* for output grid of size 3 × 5. (**b**) ROI on *TLC2* for output grid of size 3 × 5. (**c**) ROI on *TLC1* for output grid of size 6 × 10. (**d**) ROI on *TLC2* for output grid of size 6 × 10. (**e**) ROI on *TLC1* for output grid of size 12 × 20. (**f**) ROI on *TLC2* for output grid of size 12 × 20.

As further emphasized in the following sections, the main goal of our work is to understand to what extent the joint use of two cameras can represent an added value for the VCA task with respect to the use of a single camera (either *TLC1* or *TLC2*).

4.2. Performance Evaluation

Two types of performance figures will be considered to evaluate the VCA monitoring system, namely *object detection performance*, that is the ability of the system to correctly identify different classes of objects inside the ROI, and *alert generation performance*, that is the ability of the system to trigger an alert if and only if at least one object is present in the monitored area.

For the sake of simplicity, the system performance results were assessed considering only 3 grids (i.e., 12 × 20, 6 × 10 and 3 × 5) with priors of size 1 × 2 (more suitable for identifying people) and priors of size 2 × 1 (more suitable for identifying vehicles).

Finally, the VCA system performance results were evaluated also in terms of computation time required for object detection and alert generation. The average inference time per frame was assessed locally on a host equipped with an Intel Core i5 dual-core processor at 2.6 GHz, 8 GB RAM memory banks, and running the macOS 10.15.7 operating system.

4.2.1. Object Detection Performance

The ability of each component of $\mathrm{SSD}_{\mathrm{ret}}$ (according to the aforementioned grids and priors) to identify objects of different classes inside the ROI was evaluated by calculating the average confusion matrix over the whole test dataset, for each camera, namely the average number of correctly identified objects ($\mathrm{TP}_{\mathrm{obj}}$), the average number of undetected objects ($\mathrm{FN}_{\mathrm{obj}}$), and the average number of objects detected but not actually present in the ground truth image ($\mathrm{FP}_{\mathrm{obj}}$). The obtained values were then compared with the average number of real objects per image. Then, in order to measure the object detection performance from a comprehensive point of view, precision ($\mathrm{PRE}_{\mathrm{obj}}$) and recall ($\mathrm{REC}_{\mathrm{obj}}$) were assessed for each considered frame, both individually for single grids and priors, and aggregating all outputs. Precision measures the number of correctly identified objects to the total number of detected objects, whereas recall measures the number of correctly detected objects to the total number of ground truth objects. These metrics were then averaged across all the frames in the test dataset (i.e., 1000 frames).

4.2.2. Alert Generation Performance

The ability of $\mathrm{SSD}_{\mathrm{ret}}$ to generate alerts when an object is inside the ROI was assessed by calculating the confusion matrix over the entire test dataset, considering two possible outputs of the system, namely the presence of an alert (*alert* = 1) or its absence (*alert* = 0), for each input image. The following elements of the confusion matrix were considered: the total number of correctly generated alerts ($\mathrm{TP}_{\mathrm{alert}}$), the total number of ground truth alerts not triggered by the system ($\mathrm{FN}_{\mathrm{alert}}$), the total number of alerts incorrectly triggered by the system ($\mathrm{FP}_{\mathrm{alert}}$), and the total number of non-alert situations in which the alert is correctly not triggered by the system ($\mathrm{TN}_{\mathrm{alert}}$). It is also important to underline that, in light of the technological implementation of the alerts triggering system of each camera, incorrect alerts (either $\mathrm{FN}_{\mathrm{alert}}$ or $\mathrm{FP}_{\mathrm{alert}}$) were only triggered when no true positives had already been generated for the same image.

As previously described, SSD models provide different outputs from output maps of different sizes. Therefore, system performance was first evaluated by considering alerts detected individually by each grid and prior and then by evaluating the total amount of alerts identified by the aggregation of all grids and all priors. Alert generation performance was evaluated both individually on the two cameras (*TLC1* and *TLC2*, separately) and then on their fusion. In the latter case, an alert is generated when at least one of the two cameras detects an object within the ROI.

Since the frames considered in our use case are temporally continuous, we also decided to evaluate if the presence of $\mathrm{FN}_{\mathrm{alert}}$ and $\mathrm{FP}_{\mathrm{alert}}$ could be considered a transient phenomenon or not. Hence, we computed also the $\mathrm{FN}^{*}{}_{\mathrm{alert}}$ and $\mathrm{FP}^{*}{}_{\mathrm{alert}}$, representing the false negatives and false positives occurred at least in two consecutive frames. Any $\mathrm{FN}_{\mathrm{alert}}$ or $\mathrm{FP}_{\mathrm{alert}}$ events present in just one frame were therefore considered spurious and avoided by waiting for the next frame before performing inference.

5. Results

5.1. Object Detection Performance

A base model was trained on a set of images composed by *TLC1* images and external images from open-source datasets on mobility scenarios. The base model was then retrained on a dataset extracted from *TLC2* recordings yielding $\mathrm{SSD}_{\mathrm{ret}}$. The procedure of retraining (on *TLC2* images only) an already pre-trained model offers several advantages over training from scratch (using *TLC1* and *TLC2* images). Notably, retraining was faster than the full training. Specifically, the time required to retrain the model was more than 10 times shorter than the original training time of the baseline SSD (i.e., 42 h). Table 2 reports the obtained object detection performance for each camera, each grid, and each prior separately in terms of mean confusion matrix over the entire test dataset. Average precision and average recall were also computed.

Table 2. Mean and standard deviation (between parentheses) of TP_{obj}, FP_{obj}, FN_{obj} and percentage of PRE_{obj} and REC_{obj} for each camera, grid, and prior of the SSD_{ret} model.

	TLC1						**TLC2**					
	#Real Objects	$\mathbf{TP_{obj}}$	$\mathbf{FP_{obj}}$	$\mathbf{FN_{obj}}$	$\mathbf{PRE_{obj}}$	$\mathbf{REC_{obj}}$	**#Real Objects**	$\mathbf{TP_{obj}}$	$\mathbf{FP_{obj}}$	$\mathbf{FN_{obj}}$	$\mathbf{PRE_{obj}}$	$\mathbf{REC_{obj}}$
Grid: 12 × 20 Prior: 1 × 2	0.19 (0.80)	0.10 (0.55)	0.04 (0.20)	0.08 (0.87)	55%	54%	0.71 (1.43)	0.24 (0.79)	0.22 (0.57)	0.40 (0.96)	43%	31%
Grid: 12 × 20 Prior: 2 × 1	0.02 (0.31)	0.02 (0.24)	0.08 (0.36)	0.005 (0.13)	11%	66%	0.34 (1.18)	0.24 (0.99)	0.28 (0.65)	0.10 (0.61)	24%	67%
Grid: 6 × 10 Prior: 1 × 2	0.05 (0.35)	0.05 (0.30)	0.15 (0.42)	0.02 (0.23)	19.76%	63.46%	0.13 (0.48)	0.07 (0.36)	0.07 (0.27)	0.06 (0.28)	37%	43%
Grid: 6 × 10 Prior: 2 × 1	0.005 (0.08)	0.005 (0.10)	0.18 (0.47)	0.00 (0.00)	1.6%	100%	0.08 (0.40)	0.01 (0.14)	0.08 (0.35)	0.07 (0.43)	15%	21%
Grid: 3 × 5 Prior: 1 × 2	0.01 (0.14)	0.00 (0.05)	0.07 (0.25)	0.01 (0.13)	4.11%	42.86%	-	-	1.70 (0.59)	-	0%	-
Grid: 3 × 5 Prior: 2 × 1	0.001 (0.03)	0.00 (0.00)	0.08 (0.28)	0.001 (0.03)	0%	0%	0.07 0.33	0.04 0.23	1.63 0.56	0.03 (0.19)	1.3%	61.44%

According to Table 2, it appears that the *TLC1* images contain fewer ground truth objects inside the ROI than the TLC2 ones. However, no ground truth events filmed by *TLC2* are captured by the 3 × 5 grid with 1 × 2 prior. Hence, it was not possible to calculate TP_{obj}, FN_{obj} and recall in that case. Since the number of false positives is on average higher than the number of false negatives, PRE_{obj} is lower than REC_{obj}, except when considering a 12 × 20 grid with 1 × 2 prior. In addition, we can observe how grids with a larger number of cells (i.e., 12 × 20 and 6 × 10) are generally able to detect more objects than the smallest grid (i.e., 3 × 5). This may be due to the fact that objects within the ROI are typically in the background and thus more easily detected by denser grids, characterized by smaller cell sizes.

The global object detection performance results of SSD_{ret} on both cameras were then evaluated in terms of precision and recall, reported in Table 3. These values were obtained by considering all the grids and priors used to define the model's architecture (as defined in Table 1). *TLC1* yielded a low precision of about 17% and a satisfying recall, equal to about 90%. In contrast, *TLC2* yielded a much higher precision of about 73% and recall similar to *TLC1* (i.e., about 89%).

Table 3. Global object detection performance of SSD_{ret} for each camera by considering all the grids and priors as defined in Table 1. Precision: PRE_{obj}; Recall: REC_{obj}.

	TLC1	**TLC2**
PRE_{obj}	17%	73%
REC_{obj}	90%	89%

5.2. Alert Generation Performances

Alert generation performance was first evaluated separately on the two cameras and then considering the fusion between the alerts generated by the two, as shown in Table 4. The results reported in Table 4 are consistent with those shown in Table 2 since grids with a larger number of cells (i.e., 12 × 20 and 6 × 10) are able to generate more alerts than the smallest grid (i.e., 3 × 5). In particular, with the exception of the 3 × 5 grid that mostly detects vehicles, most of the alerts seem to be raised by objects that correspond to the prior of size 1 × 2 (i.e., pedestrians in the ROI). From these results, we can observe that both the number of ground truth alerts and the number of correctly predicted alerts ($\mathrm{TP}_{\mathrm{alert}}$) increase when considering the data fusion of both cameras (*fusion(TLC1,TLC2)*), compared to the individual *TLC1* and *TLC2*. Figure 3 shows an example of an alert correctly detected by *TLC2* but not by *TLC1*. This image would therefore constitute an FN event considering only *TLC1*, but it is correctly classified as a TP event when *fusion(TLC1,TLC2)* is considered.

Table 4. Number of ground truth alerts and $\mathrm{TP}_{\mathrm{alert}}$ for each grid and prior using single camera processing (*TLC1*, *TLC2*) and data fusion of both cameras (*fusion(TLC1,TLC2)*).

	TLC1		**TLC2**		**Fusion(TLC1,TLC2)**	
	Ground Truth Alerts	$\mathbf{TP}_{\mathbf{alert}}$	**Ground Truth Alerts**	$\mathbf{TP}_{\mathbf{alert}}$	**Ground Truth Alerts**	$\mathbf{TP}_{\mathbf{alert}}$
Grid: 12 × 20 Prior: 1 × 2	62	54	41	27	76	61
Grid: 12 × 20 Prior: 2 × 1	9	6	29	25	34	28
Grid: 6 × 10 Prior: 1 × 2	66	49	29	23	76	57
Grid: 6 × 10 Prior: 2 × 1	8	8	3	0	11	8
Grid: 3 × 5 Prior: 1 × 2	3	2	2	0	5	2
Grid: 3 × 5 Prior: 2 × 1	7	4	3	0	10	4

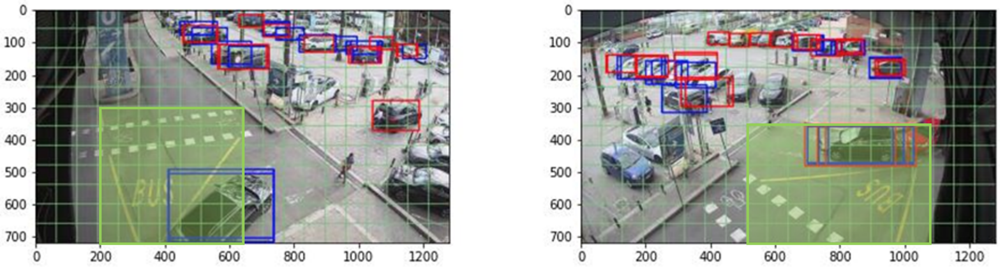


Figure 3. Example of the same object correctly detected within the ROI (green area) by *TLC2* (**right**), but missed by *TLC1* (**left**). Ground truth boxes are shown in blue, while predicted boxes are shown in red.

If we focus, for example, on the 12 × 20 grid and the 1 × 2 prior (Table 4), we can observe that *TLC1* alone detects 62 ground truth alerts (54 $\mathrm{TP}_{\mathrm{alert}}$), while *TLC2* detects 41 ground truth alerts (27 $\mathrm{TP}_{\mathrm{alert}}$) and *fusion(TLC1,TLC2)* detects 76 ground truth alerts (61 $\mathrm{TP}_{\mathrm{alert}}$). These results confirm how different grids and priors are able to identify different objects, and consequently generate different alerts. For this reason, we finally evaluated the global alert generation performance results, obtained by combining all the outputs provided by different priors and grids and by considering the temporal continuity of the frames. The results of this global evaluation are reported in Table 5.

Table 5. Ground truth alerts, TP_{alert}, TN_{alert}, FP_{alert}, FN_{alert}, $FP^{*}{}_{alert}$ and $FN^{*}{}_{alert}$ obtained from all grids and priors, on single cameras (*TLC1*, *TLC2*) and their fusion (fusion(TLC1,TLC2)).

	Ground Truth Alerts	TP_{alert}	TN_{alert}	FP_{alert}	FN_{alert}	$FP^{*}{}_{alert}$	$FN^{*}{}_{alert}$
TLC1	89	77	865	46	12	2	0
TLC2	74	59	908	18	15	1	2
fusion(TLC1,TLC2)	125	105	827	48	20	3	3

The estimated average elapsed time during the inference phase for the whole alert generation process on a single camera is about 0.46 s per frame, while the elapsed time of the decision fusion is about $1.8 \cdot 10^{-6}$ s and may be neglected. Thus, the total inference time of the multi-camera VCA system (not parallelized) is about 0.92 s per frame.

6. Discussion

A VCA monitoring system based on a SSD architecture was implemented and evaluated in terms of its ability to detect objects in the surveilled area and its related ability to generate alerts. Specifically, the VCA system foresees possible dangerous situations inside a intersection through the use of a multi-camera deep learning-based object detection system. The choice to merge data at the decision level was motivated by its simplicity, which allows to operate within the time constraints dictated by a real-time application. In addition, the system built in this way can easily compensate for the lack of one of the two possible inputs, ensuring robustness against possible failures or damages to the system.

Comparing the *TLC1* and *TLC2* cases, it can be seen that the former has a rather low precision in detecting objects. This result is further confirmed by the performance results of alert generation (Table 5). Provably, the precision of *TLC1* in terms of alert generation is lower than the corresponding *TLC2* precision (i.e., 62% and 77%, respectively). As a result, *fusion(TLC1,TLC2)* reaches a higher precision (i.e., 69%) with respect to *TLC1* alone. In contrast, the recall of *TLC1* in terms of alert generation is slightly higher than the corresponding *TLC2* recall (i.e., 86% and 80%, respectively). As a result, *fusion(TLC1,TLC2)* yields a higher recall (i.e., 84%) with respect to *TLC2* alone. In summary, by combining the two cameras, there is a significant increase in precision with respect to *TLC1* alone and a slight improvement in recall compared to *TLC2*. The monitoring system based on SSD_{ret} yields quite satisfactory alert generation accuracy when considering a single camera (i.e., about 94%).This means that the retraining phase did not erase what the model learned from *TLC1* images, i.e., there is no catastrophic forgetting [40]. Although accuracy remains almost stable (93%) when considering *fusion(TLC1,TLC2)*, the introduction of a second camera *TLC2* improves the overall safety by allowing the identification of a higher number of real dangerous situations (i.e., 125 ground truth alerts) within the area of interest. In fact, the combination of *TLC1* and *TLC2* enables the triggering of 40% more ground truth alerts than *TLC1* alone. The increase in the number of alerts is mainly due to the different framing of the two cameras, and thus the increased field of view of the object detection system. Consequently, also the absolute number of TP_{alert} increases (from 77 to 105) after the outputs of the two cameras are merged. Since we are dealing with a highly unbalanced dataset, where the number of dangerous situations is considerably lower than the number of safe situations, it could be useful to evaluate the F1-score. Specifically, it can be seen that the use of two cameras results in an F1-score of 75%, which is higher than that obtained by using *TLC1* alone (i.e., 73%).

By using not only spatial redundancy, i.e., the different views of the same monitored area captured by *TLC1* and *TLC2*, but also the temporal continuity of the frames, we can design a post-processing algorithm that uses the information of two or more consecutive video frames instead of a single one as assumed so far. In this case, the actual output alert signal is generated if it is triggered by at least two consecutive frames. By exploiting the temporal continuity, the amount of wrong predictions is reduced, as indicated in Table 5, where $FP^{*}{}_{alert}$ and $FN^{*}{}_{alert}$ (i.e., the number of FPs and FNs persisting in at least two consecutive frames) are consistently lower than FP_{alert} and FN_{alert}, respectively. This reduction in the number of false and missed alarms proves that FPs and FNs are generally spurious events that can be easily removed by considering a certain time window. However, it is worth noting that this method introduces a one-frame delay in the alert signal generation stage.

In addition, a local evaluation of the total inference time per frame was performed, demonstrating the ability of the proposed multi-camera VCA system to generate the alert in a sufficiently short time (less than 1 s), which is compatible with the system requirements to make a decision in real time. However, more precise evaluations will be needed following specific on-site deployment.

This study presents some limitations. First of all, the multi-camera system was evaluated using a single fusion technique directly applied at the decision level. In future studies, different data fusion techniques, including early and late fusion at different depths of the network, should be compared to evaluate possible further improvements in terms of the system reliability. Moreover, although the network was originally trained on a heterogeneous set of images from the experimental test-bed (*TLC1*) and open source datasets, the dataset used for retraining SSD_{ret} included only *TLC2* frames captured in daytime. Therefore, it will be necessary to evaluate the system's ability to generalize in different scenarios, such as its robustness in different weather and light conditions (e.g., day/night and sunny/rainy weather). Lastly, at the current stage, possible security issues following malicious attacks on the main components of the system (e.g., cameras, onboard units, and edge servers) have not been considered yet. In particular, the alert generation system could be vulnerable to adversarial attacks aimed at changing the output of the system, which could cause potential dangerous situations. In the future, it will be necessary to devise robust solutions to these types of attacks, such as considering the introduction of a Bayesian layer in the vision system [41].

7. Conclusions

This work focuses on the development and evaluation of an single-shot multibox detector-based object detection system applied to an urban scenario. In particular, we evaluated the effectiveness of adding a second camera (*TLC2*) in terms of detecting potential hazardous situations within the region of interest. The introduction of a second camera, in addition to the first one (*TLC2*) not only makes the video content analysis system more robust with respect to possible failures due to *TLC1* malfunctions but also leads to a higher number of correctly detected alarms thanks to a wider coverage of the surveilled area. Furthermore, the number of false negative (FN-type) events is reduced by considering temporal continuity in successive frames. In the specific smart mobility use case, FN-type errors were considered to be more important than false positive (FP-type) errors. Indeed, the number of negative events misclassified as positive (i.e., FP-type), will result in alarms that do not correspond to the presence of objects or obstacles in the region of interest. Such errors are considered less critical because they simply cause unnecessary alerts to be sent, without endangering the driver. However, in the long run, these redundant alarms may make the driver less confident in the system's ability to correctly identify dangerous situations. Future studies will focus on further validation of the proposed solution. Finally, the formalization of an algorithm that can leverage the temporal continuity provided by videos, instead of relying on individual frames, could be investigated.

Author Contributions: Conceptualization, M.M., A.P. and V.R.; methodology, M.L., M.M., S.N., A.P. and V.R.; software, M.L. and S.N.; validation, M.L., M.M., S.N. and V.R.; formal analysis, M.L. and S.N.; investigation, M.L., M.M., S.N. and V.R.; resources, M.M. and V.R.; data curation, M.L. and S.N.; writing—original draft preparation, M.L., M.M., S.N., A.P. and V.R.; writing—review and editing, M.L., M.M., S.N., A.P. and V.R.; visualization, M.L., S.N. and V.R.; supervision, M.M.; project administration, M.M.; funding acquisition, M.M. and V.R. All authors have read and agreed to the published version of the manuscript.

Funding: The work was carried out within the Genova 5G project, a tender (Call MiSE of 5/03/2020) by the Italian Ministry of Economic Development (MiSE) for the acquisition of technologies aimed at the safety of road infrastructures in the territorial area of Genova through experiments based on 5G technology. The project recently ended in October 2022.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study may be available upon request to the corresponding author.

Acknowledgments: The authors would like to thank Vodafone, being the administrative and technical coordinator of the Genova 5G project, as well as network operator and 5G technology enabler. The authors are also grateful to Aitek S.p.A. (Vanessa Orani, Stefano Delucchi and Bernardo Pilarz) for their assistance in the development of the VCA solution. The authors would also like to thank all partners involved in the project: Azienda Mobilità e Trasporti SpA, Genova, Comune di Genova, Leonardo, Start 4.0. Marta Lenatti is a PhD student enrolled in the National PhD in Artificial Intelligence, XXXVIII cycle, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma.

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

References

1. Founoun, A.; Hayar, A. Evaluation of the concept of the smart city through local regulation and the importance of local initiative. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 16–19 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
2. Savithramma, R.; Ashwini, B.; Sumathi, R. Smart Mobility Implementation in Smart Cities: A Comprehensive Review on State-of-art Technologies. In Proceedings of the 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 January 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 10–17.
3. Celidonio, M.; Di Zenobio, D.; Fionda, E.; Panea, G.G.; Grazzini, S.; Niemann, B.; Pulcini, L.; Scalise, S.; Sergio, E.; Titomanlio, S. Safetrip: A bi-directional communication system operating in s-band for road safety and incident prevention. In Proceedings of the 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), Yokohama, Japan, 6–9 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–6.
4. Wen, J.; He, Z.; Yang, Y.; Cheng, Y. Study on the factors and management strategy of traffic block incident on Hangzhou Province Highway. In Proceedings of the 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Vientiane, Laos, 11–12 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 67–71.
5. Mauri, A.; Khemmar, R.; Decoux, B.; Ragot, N.; Rossi, R.; Trabelsi, R.; Boutteau, R.; Ertaud, J.Y.; Savatier, X. Deep learning for real-time 3D multi-object detection, localisation, and tracking: Application to smart mobility. *Sensors* **2020**, *20*, 532. [CrossRef] [PubMed]
6. Jiao, L.; Zhang, R.; Liu, F.; Yang, S.; Hou, B.; Li, L.; Tang, X. New generation deep learning for video object detection: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3195–3215. [CrossRef] [PubMed]
7. Chen, Z.; Khemmar, R.; Decoux, B.; Atahouet, A.; Ertaud, J.Y. Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility. In Proceedings of the 2019 Eighth International Conference on Emerging Security Technologies (EST), Colchester, UK, 22–24 July 2019; pp. 1–6. [CrossRef]
8. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Qi, S.; Ning, X.; Yang, G.; Zhang, L.; Long, P.; Cai, W.; Li, W. Review of multi-view 3D object recognition methods based on deep learning. *Displays* **2021**, *69*, 102053. [CrossRef]
11. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
12. Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; Gao, Y. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 264–272.
13. Wei, X.; Yu, R.; Sun, J. View-gcn: View-based graph convolutional network for 3d shape analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1850–1859.

14. Kanezaki, A.; Matsushita, Y.; Nishida, Y. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5010–5019.
15. Chavdarova, T.; Baqué, P.; Bouquet, S.; Maksai, A.; Jose, C.; Bagautdinov, T.; Lettry, L.; Fua, P.; Van Gool, L.; Fleuret, F. WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5030–5039. [CrossRef]
16. Tang, Z.; Naphade, M.; Liu, M.Y.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.; Hwang, J.N. CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8789–8798. [CrossRef]
17. Wu, H.; Zhang, X.; Story, B.; Rajan, D. Accurate Vehicle Detection Using Multi-camera Data Fusion and Machine Learning. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 3767–3771. [CrossRef]
18. Chavdarova, T.; Fleuret, F. Deep multi-camera people detection. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 848–853.
19. Dinh, V.Q.; Munir, F.; Azam, S.; Yow, K.C.; Jeon, M. Transfer learning for vehicle detection using two cameras with different focal lengths. *Inf. Sci.* **2020**, *514*, 71–87. [CrossRef]
20. Ciampi, L.; Gennaro, C.; Carrara, F.; Falchi, F.; Vairo, C.; Amato, G. Multi-camera vehicle counting using edge-AI. *Expert Syst. Appl.* **2022**, *207*, 117929. [CrossRef]
21. Unlu, E.; Zenou, E.; Riviere, N.; Dupouy, P.E. Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSJ Trans. Comput. Vis. Appl.* **2019**, *11*, 7. [CrossRef]
22. Seeland, M.; Mäder, P. Multi-view classification with convolutional neural networks. *PLoS ONE* **2021**, *16*, e0245230. [CrossRef] [PubMed]
23. Ezatzadeh, S.; Keyvanpour, M.R.; Shojaedini, S.V. A human fall detection framework based on multi-camera fusion. *J. Exp. Theor. Artif. Intell.* **2022**, *34*, 905–924. [CrossRef]
24. Saurav, S.; Saini, R.; Singh, S. A dual-stream fused neural network for fall detection in multi-camera and 360° videos. *Neural Comput. Appl.* **2022**, *34*, 1455–1482. [CrossRef]
25. Narteni, S.; Lenatti, M.; Orani, V.; Rampa, V.; Paglialonga, A.; Ravazzani, P.; Mongelli, M. Technology transfer in smart mobility: The driver alert pilot of 5G Genova project. In Proceedings of the 11th World Conference on Information Systems and Technologies (WorldCIST'23), 1st Workshop on Artificial Intelligence for Technology Transfer (WAITT'23), Pisa, Italy, 4–6 April 2023; pp. 1–4.
26. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
27. Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; Barlaud, M. Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.* **1997**, *6*, 298–311. [CrossRef] [PubMed]
28. Barrow, D.; Kourentzes, N.; Sandberg, R.; Niklewski, J. Automatic robust estimation for exponential smoothing: Perspectives from statistics and machine learning. *Expert Syst. Appl.* **2020**, *160*, 113637. [CrossRef]
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
30. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
31. Kowalczyk, P.; Izydorczyk, J.; Szelest, M. Evaluation Methodology for Object Detection and Tracking in Bounding Box Based Perception Modules. *Electronics* **2022**, *11*, 1182. [CrossRef]
32. Krasin, I.; Duerig, T.; Alldrin, N.; Ferrari, V.; Abu-El-Haija, S.; Kuznetsova, A.; Rom, H.; Uijlings, J.; Popov, S.; Kamali, S.; et al. OpenImages: A Public Dataset for Large-Scale Multi-Label and Multi-Class Image Classification. Available online: https://storage.googleapis.com/openimages/web/index.html (accessed on 11 January 2023)).
33. Ess, A.; Leibe, B.; Schindler, K.; van Gool, L. A Mobile Vision System for Robust Multi-Person Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), Anchorage, Alaska, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008.
34. Braun, M.; Krebs, S.; Flohr, F.B.; Gavrila, D.M. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1844–1861. [CrossRef] [PubMed]
35. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. [CrossRef]
36. FFmpeg 5.0. Available online: https://ffmpeg.org/ (accessed on 5 July 2022).
37. Jocher, G. YOLOv5 by Ultralytics (Version 7.0)[Computer Software], 2020. Available online : https://zenodo.org/record/7347926/#.ZBGNcnZByUk (accessed on 10 November 2022)).
38. Informative, A.B.P.A. *ST 296:2011 - SMPTE Standard*; 1280× 720 Progressive Image Sample Structure—Analog and Digital Representation and Analog Interface. 2011.
39. ONVIF Profiles. Available online: https://www.onvif.org/profiles/ (accessed on 1 March 2023).

40. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **1999**, *3*, 128–135. [CrossRef]
41. Pang, Y.; Cheng, S.; Hu, J.; Liu, Y. Evaluating the robustness of bayesian neural networks against different types of attacks. In Proceedings of the CVPR 2021 Workshop on Adversarial Machine Learning in Real-World Computer Vision Systems and Online Challenges (AML-CV), Virtual Conference, 19–25 June 2021 .

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.