IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 April 2024, accepted 22 April 2024, date of publication 6 May 2024, date of current version 14 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397013

RESEARCH ARTICLE

Open Urban mmWave Radar and Camera Vehicle Classification Dataset for Traffic Monitoring

JÜRGEN SOOM[1,2], MAIRO LEIER[1], KARL JANSON[1], AND JEFFREY A. TUHTAN[2], (Member, IEEE)

[1]Embedded AI Research Laboratory, Tallinn University of Technology, Harju County, 12618 Tallinn, Estonia
[2]Department of Computer Systems, Tallinn University of Technology, Harju County, 12618 Tallinn, Estonia

Corresponding author: Jürgen Soom (jurgen.soom@taltech.ee)

This work was supported by the European Union through European Social Fund through the Project IT Academy (IT Akadeemia in Estonian) (ICT) Program.

ABSTRACT Traffic monitoring systems featuring robust, multi-sensor fusion capabilities are rapidly growing in demand to observe traffic flow, reduce congestion and to detect and report traffic accidents. However, monitoring outdoor environments using cameras remains challenging due to complex weather conditions, including fog, rain, snow and variable lighting conditions. The presence of these weather conditions can significantly reduce vehicle detection and classification performance using machine learning methods. Unfortunately, openly available datasets for multi-sensor traffic monitoring development and testing remain limited, especially those featuring infrastructure-based cameras and millimeter wave (mmWave) radar. To address these challenges, we evaluate open camera and mmWave radar data using vehicle classification models for cars, trucks, vans and buses on embedded hardware. We also provide an open multi-sensor traffic monitoring dataset with more than 8,000 manually annotated frames as well as mmWave radar point clouds recorded in an urban environment under sunny, partially cloudy, cloudy, rainy and night conditions.

INDEX TERMS Object detection, edge computing, machine learning, camera, millimeter wave radar, traffic video.

I. INTRODUCTION

In 2020, the International Council on Clean Transportation (ICCT) reported 11.7 million new vehicle registrations in the 27 member states of the European Union and the United Kingdom. [1]. According to the report published by the European Automobile Manufacturers Association (ACEA), China alone had more than 25.5 million newly registered vehicles in 2022. From 2021 to 2022, the number of newly registered vehicles in India increased by 24.1% [2]. The Bureau of Transportation Statistics (BTS) 2021 survey shows that the USA currently has 275.9 million registered light-duty vehicles, motorcycles, trucks, and buses [3]. Analysts expect that the number of newly registered vehicles will continue to grow for the foreseeable future, reaching 2.21 billion worldwide by 2050 [4], [5]. With the growing number of vehicles around the world, developing and managing a city's transportation infrastructure has become a substantial and persistent challenge. Frequent problems include traffic jams, congestion, and environmental and noise pollution. To address these challenges, traffic monitoring systems are deployed to collect data about the traffic flow, including velocity, volume, Peak Hour Factor (PHF), density, headway, spacing, gap, and clearance [8]. This data is essential to improve the planning and development of transportation infrastructure. However, collecting the required information in outdoor environments still poses a major technology challenge, especially in the presence of fog, rain, snow, and during the night.

II. RELATED WORK

Sensors used for traffic monitoring applications are divided into three subcategories: in-roadway, side-roadway, and over-roadway [9]. The earliest traffic monitoring systems used in-roadway sensors and were primarily adapted for vehicle counting applications. Inductive loop detectors (ILD) monitor

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

passing vehicles by detecting changes in inductance. ILD sensors have excellent detection accuracy, ranging from 92% [10], [11], [12], [13] up to 99% accuracy with magnetic sensors [14], [15]. However, in-road sensor systems have several drawbacks: high installation and maintenance costs, pavement damage during installation, and limited lane coverage. Furthermore, installation and maintenance can cause traffic disruptions and congestion, since the road section must be closed during these activities [9].

Non-intrusive, side- and over-roadway traffic monitoring systems are typically comprised of acoustic sensors, light detection and range (LiDAR), or radio detection and range (radar). A key advantage over in-roadway sensing is their ability to monitor multiple lanes simultaneously [16], [17], [18]. In addition, the sensor can be installed over the road or on the side of the road without damaging the pavement or requiring closing any traffic lanes during installation and maintenance. Furthermore, these sensors are less dependent on the intensity and variability of illumination. However, reflective surfaces can cause large local changes in illumination intensity, reducing the detection accuracy, and in some cases, false detections. Nonideal weather conditions including fog, rain, or snow can also scatter or absorb radio waves, reducing both the operating range and the accuracy [19].

The first attempts at camera-based vehicle counting date back to 1978 by the Jet Propulsion Laboratory [21]. Cameras have since become the most widely used type of sensor for traffic monitoring, mainly because they offer several advantages: they can cover multiple lanes simultaneously, have flexible mounting options and require minimal maintenance [9]. The ability to survey the surroundings in high spatial and temporal detail provides cameras with a significant advantage over in-roadway sensors and they can detect and classify vehicles with over 95% accuracy [22], [23], [24]. Camera observations can also be applied for additional traffic monitoring use cases, such as assessing road conditions, detecting collisions and assisting rescue services. Despite their numerous advantages, camera-based vehicle detection accuracy remains highly dependent on weather and local illumination conditions. Fog can reduce visibility by scattering light and reducing contrast, making objects appear hazy and lacking details. Rain and snow can cause droplets to accumulate on the lens or sensor, adversely impacting the image quality. Low light conditions may produce unwanted levels of noise and blurring. Conversely, cameras may provide overexposed imagery during localized bright lighting conditions caused by reflection glare and headlights [9], [19]. Table 1 provides a comparative overview of the influence of weather conditions by sensor type. Recently, multi-sensor-based traffic monitoring systems have begun to gain more traction [28], [29], [33], [34], [35]. The primary motivation behind this trend is to improve detection accuracy and mitigate the shortcomings in complex, changing environments and weather conditions [30], [31], [32]. One major obstacle in developing, testing, and validating multi-sensor-based traffic monitoring systems is the limited number of openly available traffic monitoring datasets.

TABLE 1. Impacts of different weather conditions on cameras, radar and LiDAR sensors used for traffic monitoring. A score for each of the weather conditions and its impact on the traffic monitoring sensor type ranges from 0: negligible effect, to 5: severe impact, and was adopted for each combination in the table based on the method used in [19].

Sensor type	Light rain[1]	Heavy rain[2]	Mist[3]	Fog[4]	Snow	Strong light
Camera	3	4	5	4	2/3	5
Radar (24, 77 and 122 GHz)	0	1	2	0	2	0
LiDAR (850-950nm)	2	3	5	4	5	2

[1] < 4mm/hr
[2] < 25mm/hr
[3] Visibility < 0.1km
[4] Visibility < 0.5km
The effect level each phenomenon causes to sensors:
0 - negligible: influences that can almost be ignored
1 - minor: rarely cause detection error
2 - slight: occasionally cause minor errors
3 - moderate: cause perception error up to 30% of the time
4 - serious: cause perception error more than 30% but lower than 50% of the time
5 - severe: noise or blockage that causes false detection or detection failure

A. OVERVIEW OF MULTI-SENSOR INFRASTRUCTURE-BASED TRAFFIC MONITORING DATASETS

DAIR-V2X-I contains over 10000 annotated frames collected using a high-resolution camera and LiDAR. The dataset features ten classes, focusing on diverse weather and lighting variations [36].

A9 dataset consists of footage using a high-resolution camera and LiDAR, covering a variety of traffic situations. The anonymized and precision-timestamped footage was recorded at the three km-long Providentia++ testfield near Munich, Germany. The dataset features a total of four weather conditions and six vehicle classes [37].

LUMPI features over 200k frames, collected over several days during different weather and light conditions at a large junction in Hanover, Germany. The dataset includes 2D image information (videos) and 3D point clouds with labels of the traffic participants in the scene [38].

AAU RainSnow dataset was collected using conventional RGB and thermal infrared cameras. It features scenes with rain and snowfall, captured from 22 five-minute videos from seven different traffic intersections. The illumination of the scenes varies from broad daylight to twilight and night. Scenes show glare from car headlights, reflections from puddles, and raindrop blur to the camera lens. In total, the data contains 2200 annotated frames [43].

Radar LAB created an automatic radar-camera dataset generation toolkit for sensor-fusion applications to minimize labor costs for recording and processing camera and radar data simultaneously. However, the dataset is not openly available [45].

UTIMR applies an urban traffic imaging using millimeter-wave radar system. Information about the features of the dataset is very limited and is not openly available [46].

TJRS TS focuses on trajectory tracking using millimeter-wave radar sensors. The verification data was captured with cameras attached to UAV [44]. The weather conditions and amount of data collected are not specified. The dataset is not open access, but is available upon request.

CIM (this work) complements existing open datasets by providing novel camera and mmWave radar data, covering multiple weather conditions, locations, and vehicle classes common to urban traffic monitoring locations. The specifics of the dataset are discussed in more detail in Section IV-A. A comparison of the CIM dataset with current openly available infrastructure-based multi-sensor traffic monitoring datasets is summarized in Table 2.

B. OBJECTIVES

The objectives of this work are two-fold: First, we provide a camera and radar-based vehicle detection and classification pipeline and evaluate its performance using embedded hardware. Second, we provide a new open infrastructure-based multi-sensor traffic dataset featuring nearly 8400 manually annotated frames, including five weather conditions and four vehicle classes. This article is organized as follows: In **Section III**, we provide a detailed overview of the selected embedded hardware options, camera sensor, and mmWave radar. The methods applied for camera and mmWave radar vehicle detection and classification are provided in **Section IV**. The results are provided in **Section V**. Finally, in **Sections VI and VII**, the advantages, limitations, and future research directions are discussed.

III. HARDWARE

A. EMBEDDED HARDWARE

Two embedded hardware configurations were chosen to evaluate the proposed vehicle detection and classification pipeline, and a summary of the hardware specifications is provided in Table 3. The author's first choice was the Nvidia Jetson Nano, which represents a typical platform for AI and machine learning applications. The Jetson Nano hardware features a Quad-core Cortex-A57 microprocessor control unit (MCU), 128-core Maxwell architecture-based graphics processing unit (GPU), 4GB of LPDDR4 and can support MIPI CSI-2 cameras. Although the board lacks built-in FLASH memory, the hardware supports flash storage devices. The second choice was the Nvidia Jetson Orin Nano 4GB version, emulated using the Jetson AGX Orin Developer Kit, chosen as it is the successor to the Nvidia Xavier platform. The board features a Hexa-core MCU based on Cortex-A78AE architecture running at 1.5 GHz. In addition, the Orin Nano board has a 512-core GPU based on the Ampere architecture with 16 dedicated Tensor Cores. Both systems support software-based power consumption monitoring, which was used in this work to evaluate their efficiency. The selection of the Jetson platform by the authors can be attributed to two primary factors. First, the Nvidia Jetson platform is a widely recognized and extensively documented commercial device that has gained widespread adoption in both academic and commercial research communities. Secondly, the worldwide chip shortage has put alternative platforms, such as Raspberry Pi, in short supply for many users, making the Jetson platform the most viable option for the authors during the time this work was carried out.

B. RADAR

Unlike optical sensors, such as cameras or LiDAR, radar sensors use radio waves to detect objects. By measuring the differences between arrival times and the phase shift of the radio wave signals reflected from an object's surfaces, radar sensors estimate the distance and speed of the target. Automotive millimeter-wave (mmWave) radars do not produce a high-resolution three-dimensional scan of the environment like LiDAR. Instead, they have a more limited output of some 200 measurement points. In this work, the Texas Instruments AWR1843BOOST mmWave radar development board [53] was implemented. The kits are based on the Texas Instruments AWR1843 automotive 77GHz radar sensor. The development board also includes a signal processor to translate the analog radar data into digital radar point clouds.

1) RADAR CONFIGURATION

The radar used in this work can be configured using multiple parameters to meet the needs of the selected application. Finding the best combination of these parameters depends highly on the application, and the parameters used in this work are provided in Table 4. These parameters in Table 4 were optimized from field data testing as part of the Advanced Traffic Sensor Pilot project conducted by the Embedded AI Research Lab at Tallinn University of Technology in cooperation with Thinnect OÜ in 2021 and 2022. The configured range resolution of the radar is 0.59 m, and the velocity resolution is 1.3 km/h, with a maximum measurable velocity of 83 km/h. This resolution was achieved for objects at a maximal distance of 30 m. However, the pilot project experiments indicated that the radar can feasibly recognize objects at distances up to 100 m. The radar uses a relatively narrow beam azimuth of 15°, transmitting measurement points 15 times per second. Measurements from stationary objects were filtered to reduce clutter and simplify processing. The filter subtracts the mean value of the input samples after applying a two-dimensional Fast Fourier Transform [20].

C. CAMERA

Compared to radar and LiDAR, camera sensors are able to record multiple features such as color, shape, and luminance, which is advantageous for object detection and classification applications. However, camera performance can be negatively impacted by commonly-occurring weather and environmental conditions such as rain, fog, snow and low lighting at night [19]. In this work, a Sony IMX-219-120

TABLE 2. Comparison of open multi-sensor infrastructure-based datasets for vehicle detection and classification. CIM (this work) provides the largest open dataset for camera and mmWave radar to-date.

Name	Content	Resolution	Weather conditions	Vehicle classes	Sensors	Availability
DAIR-V2X-I [36]	10084	1920x1080	Sunny Cloudy Nighttime Rain	Passenger car Truck Bus Van Motorcycle	Camera LiDAR	Open
A9 [37]	1098	1920x1200	Cloudy Snow Fog Sunny	Passenger car Truck Van Bus Motorcycle Trailer	Camera LiDAR	Open
LUMPI [38]	200k	1640x1232 1920x1080	Sunny Cloudy Night Rain	Passenger car Truck Van Bus Motorcycle Trailer	Camera LiDAR	Open
Rope3D [39]	50k	1920x1080	Clear Rain Night Dawn/Dusk	Passenger car Motorcycle Van Bus Truck Bicycle Tricycle Barrow	Camera LiDAR	Open
IPS300+ [42]	14198	1920x1080	N/A	Passenger car Bicycle Tricycle Bus Truck Engineer Car	Camera LiDAR	Open
RainSnow [43]	2200	640x480	Snow Rain Night Blizzard	Passenger car Bus Truck Van	Camera Thermal Camera	Open
TJRD TS [44]	N/A	N/A	N/A	Passenger car Bus Truck Van	Camera mmWave Radar	On request
Radar LAB [45]	8035	N/A	Clear Partially-cloudy	Passenger car	Camera Radar	Not available
UTIMR [46]	N/A	N/A	N/A	Small car[1] Medium car[1] Large car[1]	Camera Radar	Not available
CIM (this work)	8393	1920x1080	Sunny Partially cloudy Rain/Sleet Cloudy Night	Passenger car Van Truck Bus	Camera mmWave Radar	Open

[1] Vehicles are classified by length: small car (L < 4.3 m), medium-sized car (4.3 m < L < 7 m), and large bus (L > 8 m).

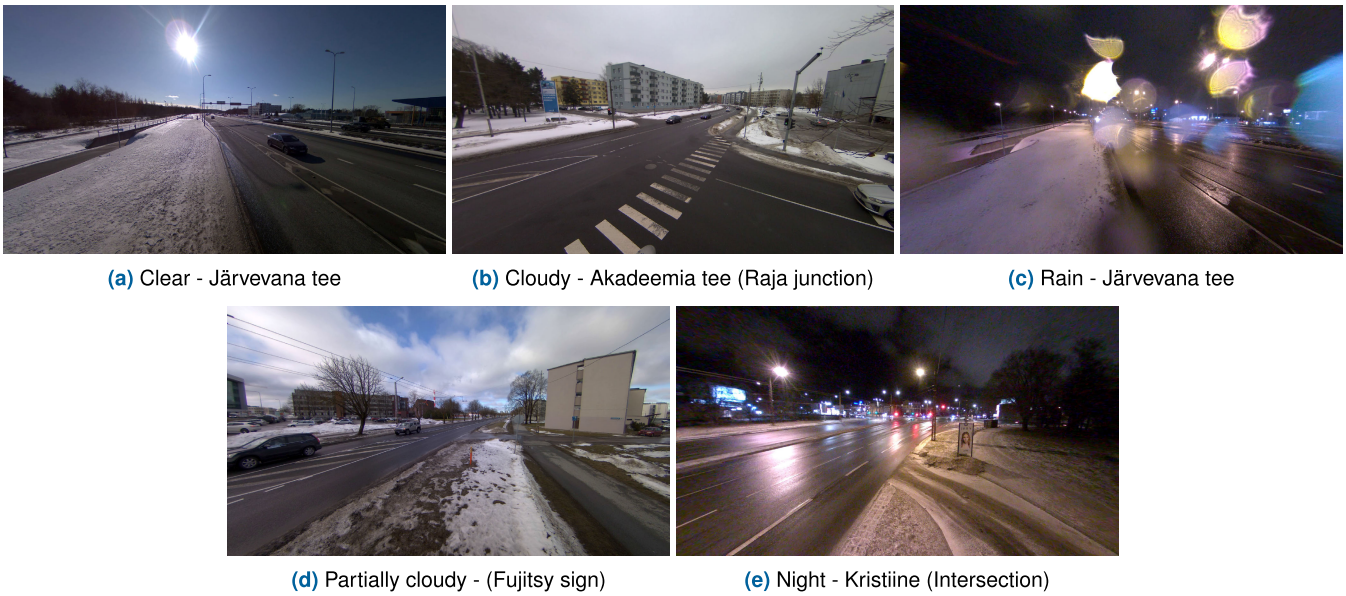

FIGURE 1. Examples from recording locations with different weather conditions. (a) Clear, ideal conditions of the roadway and vehicles. (b) Cloudy conditions, where some regions of the roadway have poor illumination at a distance. (c) Rain and other non-ideal conditions in which the camera lens may have water droplets and where sections of the roadway may have blurred imagery. (d) Partially cloudy, dynamic changes in near and far-field illumination occur on the roadway due to variations in cloud cover. (e) Night, considerable variability in the roadway illumination levels due to static street lighting in conjunction with automobile head and tail lights. Computer vision approaches for detection and classification remain challenging because at some locations different illumination and weather conditions may co-occur and vary substantially between frames.

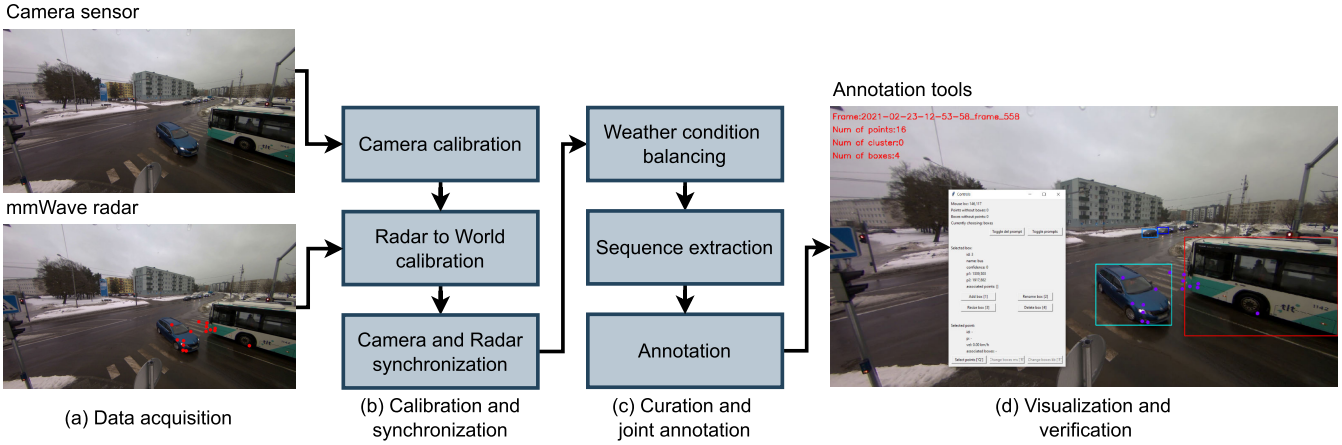

FIGURE 2. General overview of the CIM dataset curation workflow: First, the curator selects and undistorts the camera imagery. The point cloud data is then converted from 3D to the camera's 2D reference frame, as discussed in Section IV-C. Afterwards, the camera footage and radar data were synchronized. Next, the videos are sorted and balanced based on the weather and local illumination conditions following the National Oceanic and Atmospheric Administration (NOAA) guidelines [49]. After sorting, cleaning, and adjusting the recorded footage, the curator randomly selects and extracts a 100-frame sequence from each video. The sequence is then manually annotated for each vehicle within a frame using rectangular bounding boxes.

camera [52] was chosen primarily due to its compatibility with existing embedded hardware platforms and its wide angle of view. Table 5 provides an overview of the camera sensor specifications.

IV. MATERIALS AND METHODS

A. DATASET

Critical Infrastructure Monitoring (CIM, this work) is a new dataset collected in the Tallinn urban environment during the late winter and early spring, covering most weather and environmental conditions common to temperate and sub-polar regions. Fig. 1 provides an overview and examples of the weather conditions and recording locations. The original recordings feature over 41 hours of footage at a resolution of 3264 × 2464 px at 15 fps, which was fixed due to radar data acquisition limitations. The footage was undistorted and cropped to a resolution to 1920 × 1080 px. The CIM data curation workflow is depicted in Fig. 2. The final curated dataset contains 8393 manually annotated frames, including

TABLE 3. Overview of the Jetson Orin Nano and Jetson Nano hardware specifications.

Component/Feature	Nvidia Jetson Orin Nano [50]	Nvidia Jetson Nano [51]
Microprocessor	Hexa-core ARM A78AE @ 1.5 GHz	Quad-core ARM A57 @ 1.43GHz
Graphics processing unit	512-core Ampere with 16 Tensor cores	128-core Maxwell
Memory	4 GB LPDDR5	4 GB LPDDR4
Storage	MicroSD / NVMe	MicroSD
Camera	2x MIPI CSI-2 connectors	2x MIPI CSI-2 connectors
Network	Wi-Fi (802.11ac), GbE	GbE

TABLE 4. Summary of the AWR1843BOOST mWave radar hardware configuration used in this work, including brief descriptions of each parameter.

Parameter	Value	Description
Frame-rate	15	Number of measurements per second
Azimuth	15°	Horizontal angle of the radar beam
Range resolution	0.586 m	Maximum range measurement error
Max unambiguous range	30 m	Maximum range that radar can accurately measure
Velocity resolution	1.33 km/h	Maximum velocity measurement error
Max velocity	82.98 km/h	Maximum velocity radar can accurately measure
Clutter removal	Enabled	Removes radar reflections from stationary objects

TABLE 5. Summary of the Sony IMX-219-120 camera sensor specifications.

Parameter	Value
Resolution	3280 x 2464
Aperture (F)	2.2
Focal Length	1.79 mm
Angle of View (diagonal)	120°
Distortion	< 13.6%

TABLE 6. Label counts per vehicle class in each recording location. The largest number of samples belong to Fujitsy sign, featuring 14294 samples for passenger car, 382 for bus, 685 for van, and 289 for truck class.

Location	Car	Bus	Van	Truck
Akadeemia tee (Raja junction)	6438	170	50	187
Fujitsy sign	14294	382	685	289
Järvevana tee	2729	14	750	198
Kristiine (Intersection)	7471	416	571	383
Total	**30932**	**982**	**2056**	**1057**

radar point cloud data (Table 2). The distribution of sample count across all the vehicle classes and the number of samples for each vehicle class captured at each location is shown in Table 6. Each frame in the CIM dataset is represented using XML and JSON files. The XML file contains information about the vehicle class, followed by bounding box data, using (X_{min},Y_{min},X_{max},Y_{max}) formatting. The contents of the JSON and the fields in each measurement point object are

TABLE 7. Description of the JSON fields for radar point objects included in the open CIM dataset.

Field Name	Description
x	Measurement point coordinate in 3D space, relative to the radar (X component)
y	Measurement point coordinate in 3D space, relative to the radar (Y component)
z	Measurement point coordinate in 3D space, relative to the radar (Z component)
velocity	Speed at the measurement point. The value is given in km/h.
screen_coords	Pixel coordinates radar measurement point in camera's coordinate system (location of the point in camera image)
is_line	Specifies if the measurement point's coordinates are given as a line (always set to False in this work)
corresponding_box	ID of the object in the image the point is associated with. Defaults to -1 (no object)

documented in Table 7. CIM is a openly available under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License [48]. The dataset is hosted via Zenodo: 10.5281/zenodo.8301276

B. CAMERA-BASED VEHICLE DETECTION

In order to detect and classify vehicles using the camera system, we trained an object detection model based on YOLOv7 architecture [55]. Several CNN architectures were considered, such as SSDLite [25] and ResNet [26]. Ultimately, the YOLOv7 architecture was chosen based on the Average precision (AP) and inference time ratio. The model was trained to recognize four vehicle classes: passenger car, bus, truck, and van. The vehicle classification model was developed using a two-step process. As illustrated in Fig. 3, five randomly subsampled models were trained, tested and validated. The CIM dataset contains a total of 8393 frames, and was split into separate datasets for training and validation. The training dataset used for training and testing included 7218 labeled frames, leaving 1175 labeled frames for the validation dataset. The 7218 frames were further divided into five datasets for training and testing using random subsampling (see Fig. 3 (2)). Each of the five datasets contained 6714 frames for training and 504 for testing. All five models were then trained over a total of 50 epochs. In order to select the best-performing model, the five trained models were validated using the same hold-out data in the final stage (see Fig. 3 (3)). The results of the validation phase are further discussed in Section V-A.

C. RADAR-BASED VEHICLE CLASSIFICATION

A radar-based vehicle classifications model was developed using the mmWave radar point cloud coordinates converted from the sensor frame of reference using the extrinsic and intrinsic parameters of the camera. The extrinsic parameters

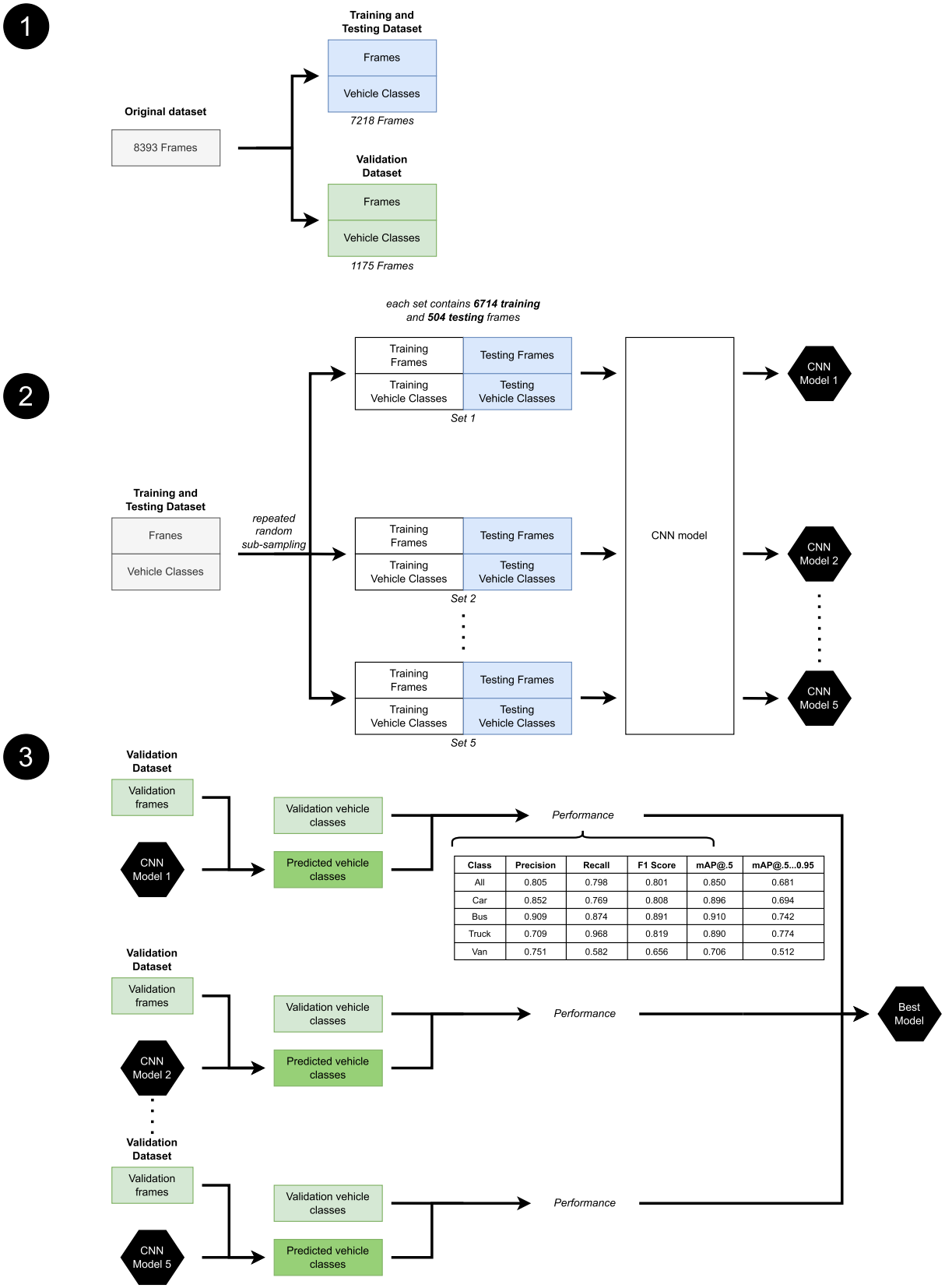

FIGURE 3. 1) Illustration of the hold-out procedure for training, testing, and validating the vehicle detection model. 2) Usage of repeated sub-sampling was applied for testing and training and was used to train models with identical CNN architecture. 3) Finding the best-performing model using the validation dataset, adapted from [27].

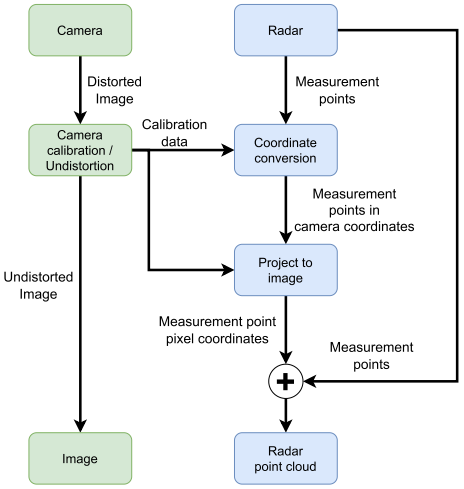

FIGURE 4. Workflow used to calibrate and align the camera pixel locations with the mmWave radar point cloud.

allow conversion between the camera and the world coordinate system. Extrinsic parameters are used to rotate and translate the radar point cloud coordinates to match the camera coordinate system. Intrinsic parameters represent the internal properties of the camera sensor including lens distortion. An overview of the coordinate conversion process is provided in Fig. 4, and can be broken down into the following four processing steps:

1) Calibration is carried out to retrieve the camera's intrinsic and extrinsic parameters using tools from the robot operating system (ROS 2).
2) Coordinate conversion from the radar point cloud reference frame to the camera as the reference system by rotating and translating the radar point cloud into the camera's coordinate system.
3) Lens distortion effects are reduced using the intrinsic parameters obtained during the camera calibration stage.
4) The radar point cloud is projected from the 3D space into the camera's 2D image plane. This results in a common reference frame for both the camera pixel coordinates and the radar point cloud data.

Before the mmWave radar vehicle classification model could be trained, we first generated a labeled point-cloud dataset. As described in Table 7, the raw radar point cloud information includes a bounding box ID, which was used to match the points in each box to a corresponding vehicle label taken from the camera system. This was done manually for each frame by extracting the vehicle class and the bounding box coordinates (x_{min},y_{min},x_{max},y_{max}). Afterwards, we stored the number of points for each radar bounding box with the corresponding vehicle label. The outcome of the point cloud annotation process resulted in a dataset featuring 423 unique vehicle bounding box samples, summarized in Table 8. Due to the low number of truck and bus samples obtained by the radar, the truck and bus classes were merged into a single class

TABLE 8. Example datasets used for the mmWave radar based classification. Each sample contains the number of radar points, the coordinates, the point velocity and the corresponding vehicle class label.

#	Number of points	Radar point 2D coordinate pairs	Velocity (km/h)	Vehicle Class Label
1	6	[(867, 519) ... (910, 514)]	8.952	Car
2	3	[(970, 553), (1010, 556), (949, 554)]	6.445	Car
3	8	[(1219, 589) ... (1378, 574)]	0.716	Truck/Bus
.....				
423	23	[(1244, 561) ... (1244, 552)]	2.865	Truck/Bus

corresponding to large non-car vehicles. The radar dataset was divided into training and validation datasets using a ratio of 80:20, leaving 338 samples for training and 85 samples for validation. The validation dataset was checked to feature equal amount of samples labeled as *car* and *truck/bus*. Three distinct approaches were examined and assessed in an effort to classify vehicle types, relying only on the radar point cloud information: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Feedforward Neural Network (FNN).

SVM is one of the most commonly used supervised machine learning algorithms for classifying, regressing, or detecting outliers. The algorithm works by finding a hyperplane, separating the data points of one class from the other. Maximizing the distance between classes in a multidimensional space. These characteristics have enabled researchers to accurately distinguish vehicles, pedestrians, and other objects from point cloud information obtained using mmWave radar [17], [58].

The proposed SVM model utilizes a radial basis function (RBF) kernel. The RBF kernel measures the similarity between two data points in infinite dimensions. The kernel function is defined as:

$$K(x_1, x_2) = \exp(-\gamma \cdot \|x_1 - x_2\|^2) \tag{1}$$

where γ controls the 'spread' of the kernel. The closer the kernel function is to zero, the larger the Euclidean distance between two points $\|x_1 - x_2\|^2$. The larger the distance between the two points, the more likely they are dissimilar [61].

Neural networks are a category of machine learning algorithms renowned for their capacity to discern intricate patterns and relationships within data, solidifying their position as one of the most widely employed methods for addressing classification and regression tasks. These networks have been effectively deployed to perform object classification using exclusively point cloud data, enabling more streamlined and precise object categorization without reliance on conventional image-based information [59], [60].

The designed FNN model incorporates two hidden layers. The first hidden layer contains eight units, while the second only features four. Using dense layers, each individual neuron is connected to every neuron in the subsequent layer, creating a densely connected network structure. With the exception of the output layer, which employs a softmax activation

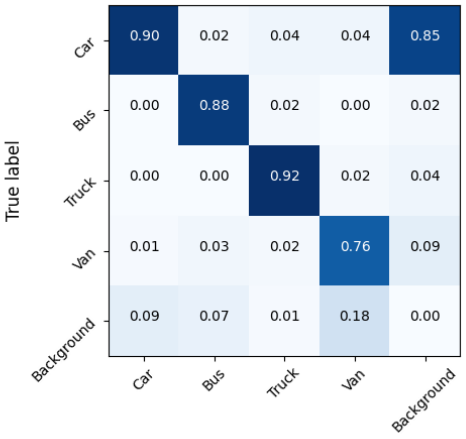

FIGURE 5. Confusion matrix of best performing camera classification model (Model 3).

function, all layers utilize the ReLu activation function. Additionally, dropout regularization was implemented to avoid overfitting the network.

KNN is a simple yet highly effective algorithm, which operating on the principle that similar data points are in close proximity to each other in the feature space. It looks at the K closest data points and assigns the new point the majority class (for classification). To find the optimal K value, we utilized the elbow method [56], [57]. The elbow method involves plotting the error rate of the KNN model as a function of different K values.

The results of the radar-based vehicle classification models are further discussed in Section V-A.

V. RESULTS

A. VEHICLE DETECTION AND CLASSIFICATION USING CAMERA

To ensure a robust evaluation of the CNN model for camera-based vehicle classification, a hold-out validation dataset containing 1175 frames was used. As described in Section IV-B and visualized in Fig 3, the curated dataset was split using a ratio of 70:30. The first portion of the split dataset was used to train and test the model, using a randomizer to generate a new training and testing data for each run. The remaining portion of the camera dataset was set aside to validate the performance and to determine the best performing model. Table 9 summarizes the CNN model performance across all vehicle classes, using the evaluation parameters described in the previous section. Of the five trained models, Model 3 performed best, albeit by a small margin, achieving the highest mAP@0.5…0.95 score of 0.681 across all vehicle classes. The confusion matrix of Model 3 is depicted in Fig. 5.

B. VEHICLE DETECTION AND CLASSIFICATION USING RADAR

Three separate approaches were tested to classify vehicle types using only information collected by the mmWave radar. The results were validated using a hold-out dataset

TABLE 9. Performance summary after validation for each of the five trained models for all vehicle classes, and for all classes combined. Model 3 was chosen as the overall best model based on the mAP@0.5…0.95 evaluation parameter.

Model	Class	Precision	Recall	F1 score	mAP@0.5	mAP@0.5...0.95
Model 1	All	0.793	0.786	0.789	0.834	0.642
	Car	0.825	0.769	0.796	0.867	0.650
	Bus	0.855	0.851	0.853	0.890	0.703
	Truck	0.747	0.952	0.837	0.905	0.759
	Van	0.744	0.572	0.647	0.675	0.457
Model 2	All	0.783	0.708	0.748	0.78	0.581
	Car	0.846	0.724	0.780	0.861	0.641
	Bus	0.840	0.712	0.771	0.799	0.568
	Truck	0.735	0.895	0.807	0.868	0.717
	Van	0.711	0.501	0.588	0.591	0.400
Model 3	All	0.805	0.798	0.801	0.850	0.681
	Car	0.852	0.769	0.808	0.896	0.694
	Bus	0.909	0.874	0.891	0.910	0.742
	Truck	0.709	0.968	0.819	0.890	0.774
	Van	0.751	0.582	0.656	0.706	0.512
Model 4	All	0.74	0.81	0.773	0.823	0.637
	Car	0.792	0.812	0.802	0.874	0.667
	Bus	0.796	0.792	0.794	0.848	0.676
	Truck	0.733	0.975	0.837	0.917	0.765
	Van	0.637	0.659	0.648	0.651	0.441
Model 5	All	0.681	0.651	0.665	0.679	0.466
	Car	0.753	0.732	0.742	0.803	0.556
	Bus	0.829	0.560	0.668	0.696	0.466
	Truck	0.633	0.779	0.698	0.754	0.560
	Van	0.508	0.533	0.520	0.463	0.276

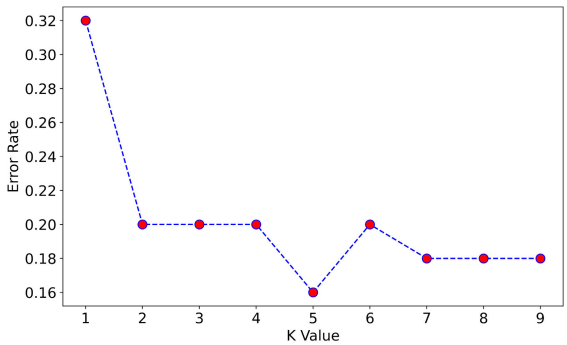

FIGURE 6. The optimal K value was determined using Elbow method. A K value of 5 resulted in the lowest error rate of 0.16.

featuring 50 samples, of which 25 belong to passenger cars and 25 to bus/truck class, to validate the performance of the radar-based classification model. The evaluation outcomes are summarized in Table 10.

The Feedforward Neural Network (FNN) exhibited the highest F1 Score for the *Car* category at 0.86. However, the method struggled to classify the *Bus/Truck* class, yielding an F1 score of 0.67. In contrast, the Support Vector Machine (SVM) demonstrated respectable performance, achieving F1 scores of 0.75 for *Car* and 0.7 for *Bus/Truck*. The K-Nearest Neighbors (KNN) performed the best of the three tested approaches. As described in Section IV-C, we used the elbow method to determine the optimal K value. As depicted in Fig. 6, the lowest error rate of 0.16 was achieved using a K value of 5. The KNN succeeded in attaining an F1 score exceeding 0.8 for both classes. The performance of the KNN model is visually depicted in the confusion matrix in Fig. 7.

TABLE 10. Classification result of the mmWave radar various methods. Out of the four tested method, KNN performed best, achieving an F1 score of 0.85 classifying cars and 0.83 for the bus/truck class.

Method	Vehicle class	Precision	Recall	F1 score
SVM	Car	0.76	0.74	0.75
	Bus/Truck	0.69	0.71	0.70
FNN	Car	0.83	0.90	0.86
	Bus/Truck	0.74	0.61	0.67
KNN	Car	0.79	0.92	0.85
	Bus/Truck	0.90	0.76	0.83

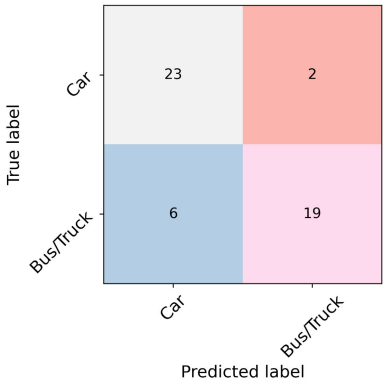

FIGURE 7. Radar point cloud classification model confusion matrix using the KNN. Out of 50 samples, 23 where correctly classified as Car and 19 as Bus/Truck, only 8 samples were incorrectly classified.

TABLE 11. Embedded hardware performance comparison when running the camera-based (YOLOv7) and mmWave radar (KNN) vehicle classification models.

Camera		
Evaluation parameter	Nvidia Jetson Orin Nano	Nvidia Jetson Nano
Frames per second	20	2 40
Power consumption (W)	8.9	4.9
Performance per Watt	2.25	0.41
mmWave Radar		
Evaluation parameter	Nvidia Jetson Orin Nano	Nvidia Jetson Nano
Frames per second	303	119
Power consumption (W)	5.8	3.6
Performance per Watt	52.24	39.67

C. HARDWARE COMPARISON–EFFICIENCY

As described in Section III-A, we evaluated the vehicle classification pipeline using two embedded hardware devices, the Jetson Nano and Jetson Orin Nano. A comparison of the results is provided in Table 11), where it was found that the Orin Nano performed the best overall, reaching up to 20 fps, while the Nano performed only 2 fps. Although the Nano (4.9 W) consumed around 4 watts less power than the Orin Nano (8.9 W), the Orin Nano (2.25 PPW) provided the highest overall performance-per-watt ratio relative to the Nano (0.41 PPW).

VI. DISCUSSION

The major contribution of this work is CIM, the open infrastructure-based multi-sensor traffic monitoring dataset. The primary motivation behind creating CIM was the limited availability of open annotated datasets featuring mmWave radar and camera imagery. Most open datasets, such as DAIR-V2X-I [36], A9 [37], LUMPI [38] and Rope3D [39] (summarized in Table 2) are based on LiDAR and camera systems. CIM, the dataset provided in this work exhibits several important features for real-world testing including multiple weather conditions and vehicle classes and features high image resolution and annotated, synchronized mmWave point clouds and corresponding vehicle labels. Datasets such as TJRD TS, Radar LAB, and UTIMR have limited availability, which CIM specifically addresses. As with all field datasets, CIM does have some shortcomings. The point cloud sample sizes are relatively sparse and are limited to cars and a limited number of buses and trucks due to the vehicle types passing during our field data collection campaign.

To mitigate this issue to the greatest extent, future studies using mmWave radar and infrastructure-mounted cameras could collect footage from a busy intersection during rush hour, allowing for the capture of more vehicle types within a single frame. Furthermore, a larger diversification of recording locations can help balance the sample count for each vehicle class. For example, the inclusion of industrial regions to collect more van and truck samples.

The second contribution of this work was to train and evaluate the performance of camera-based and mmWave radar classification in complex environments and weather conditions on embedded hardware. Weather conditions provided in the CIM dataset include overexposure from the headlights in a low-luminosity environment with low-contrast regions, which may lead to false negatives. Fog can reduce visibility by scattering light and reducing contrast, making objects appear hazy and lacking details. Rain and snow can cause droplets (see Fig. 1 (c)) to accumulate on the lens or sensor, affecting image clarity or suffering from clipping, and reducing detail in overexposed situations, increasing the amount of incorrect classifications, as well as the number of false negatives. Previous works have shown that combining multiple sensors can be effective [28], [29], [30], [31]. A recent work published by researchers at Guilin University of Electronic Technology [32] showed promising results. Using a camera and a mmWave radar, they achieved an average accuracy of 95.3% for vehicle detection. Expanding on this finding, we were able to show that our mmWave radar classification model with a few point cloud samples using KNN resulted in a minimum F1 score of 0.83 for two vehicle classes. Furthermore, the KNN managed to outperform both the SVM and FNN based classification models. However, the camera-based system was able to detect and classify cars, buses, trucks and vans with similar performance, with the best performing model obtaining an F1 score 0.805 for all classes combined.

Considering processing efficiency, one of the chosen embedded hardware options was able to process the camera data at a minimum rate of 20 fps, which far exceeded the recommended minimum of 15 fps. In situations and at

locations where restricted access to a reliable power supply is a limiting factor, we also show that embedded hardware can provide viable real-world solutions for vehicle detection under non-ideal weather conditions. Future work based on the open CIM dataset can be conducted by developing, testing and validating novel vehicle classification pipelines fusing the mmWave radar and infrastructure-mounted camera systems.

VII. CONCLUSION

We show how synchronized camera and mmWave radar traffic monitoring sensors can be applied for complimentary vehicle detection and classification on embedded hardware. In addition, we provide CIM, an open infrastructure-based camera and mmWave radar traffic monitoring dataset featuring several weather conditions and vehicle classes. The results show promise for future work related to multi-sensor classification systems using stationary camera and mmWave radar for traffic monitoring. Both models were found to be suitable for low-cost embedded hardware running in real-time. Future work will focus on expanding the open CIM dataset by investigating the use of a cascaded mmWave radar system to improve data collection in urban settings with multiple weather and environmental conditions. Further research is needed when combining camera and mmWave radar traffic monitoring systems, considering that computational performance and vehicle classification accuracy may be increased by including an additional dedicated computing accelerator to the edge hardware used in this work.

VIII. DECLARATION OF COMPETING INTEREST

The authors declare that there are no competing interests.

ACKNOWLEDGMENT

The authors wish to thank Henry Juhanson and Uljana Reinsalu from the Embedded AI Research Laboratory, Tallinn University of Technology, for their assistance in collecting the CIM dataset.

REFERENCES

[1] S. Diaz, M. R. Bernard, Y. Bernard, G. Bieker, K. Lee, P. Mock, E. Mulholland, P. Ragon, F. Rodriguez, U. Tietge, and S. Wappelhorst, "European vehicle market statistics," ICCT, Washington, DC, USA, Tech. Rep., Dec. 2021, Accessed: Aug. 18, 2022. [Online]. Available: https://theicct.org/publication/european-vehicle-market-statistics-2021-2022/

[2] ACEA. (Jul. 18, 2022). *New Passenger Car Registrations and Annual GDP Growth in the EU*. Accessed: Oct. 3, 2023. [Online]. Available: https://www.acea.auto/figure/world-new-motor-vehicle-registrations-in-units/

[3] Bureau of Transportation Statistics. (Jan. 1, 2021). *National Transportation Statistics*. Accessed: Aug. 18, 2022. [Online]. Available: https://www.bts.gov/content/number-us-aircraft-vehicles-vessels-and-other-conveyances

[4] *US Energy Information Administration*. Accessed: Aug. 18, 2022. [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=50096

[5] ACEA. (Jul. 18, 2022). *New Passenger Car Registrations and Annual GDP Growth in the EU*. Accessed: Aug. 19, 2022. [Online]. Available: https://www.acea.auto/figure/new-passenger-car-registrations-and-annual-gdp-growth-in-the-eu/

[6] J. Arts, W. Leendertse, and T. Tillema, "Road infrastructure: Planning, impact, and management," *Int. Encyclopedia Transp.*, vols. 1–7, pp. 360–372, May 2021, doi: 10.1016/b978-0-08-102671-7.10448-8.

[7] H. J. Shatz, K. E. Kitchens, S. Rosenbloom, and M. Wachs, "Highway infrastructure and the economy: Implications for federal policy," RAND Corp., Santa Monica, CA, USA, Tech. Rep., 2011. [Online]. Available: https://www.rand.org/pubs/monographs/MG1049.html

[8] *University of Memphis*. Accessed: Jun. 26, 2023. [Online]. Available: http://www.ce.memphis.edu/4162/L1_Traffic_Flow_Parameters.pdf

[9] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73340–73358, 2020, doi: 10.1109/ACCESS.2020.2987634.

[10] Y.-K. Ki and D.-K. Baik, "Vehicle-classification algorithm for single-loop detectors using neural networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 6, pp. 1704–1711, Nov. 2006, doi: 10.1109/TVT.2006.883726.

[11] S. Meta and M. G. Cinsdikici, "Vehicle-classification algorithm based on component analysis for single-loop inductive detector," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2795–2805, Jul. 2010, doi: 10.1109/TVT.2010.2049756.

[12] H. A. Oliveira, F. R. Barbosa, O. M. Almeida, and A. P. S. Braga, "A vehicle classification based on inductive loop detectors using artificial neural networks," in *Proc. 9th IEEE/IAS Int. Conf. Ind. Appl. INDUSCON*, Sao Paulo, Brazil, Nov. 2010, pp. 1–6, doi: 10.1109/INDUSCON.2010.5740079.

[13] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, "A vehicle classification based on inductive loop detectors," in *Proc. 18th IEEE Instrum. Meas. Technol. Conf. Rediscovering Meas. Age Informat.*, Budapest, Hungary, 2001, pp. 460–464, doi: 10.1109/IMTC.2001.928860.

[14] B. Yang and Y. Lei, "Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor," *IEEE Sensors J.*, vol. 15, no. 2, pp. 1132–1138, Feb. 2015, doi: 10.1109/JSEN.2014.2359014.

[15] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, and P. Varaiya, "Traffic measurement and vehicle classification with single magnetic sensor," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1917, no. 1, pp. 173–181, Jan. 2005, doi: 10.1177/0361198105191700119.

[16] H.-S. Lim, H.-M. Park, J.-E. Lee, Y.-H. Kim, and S. Lee, "Lane-by-lane traffic monitoring using 24.1 GHz FMCW radar system," *IEEE Access*, vol. 9, pp. 14677–14687, 2021, doi: 10.1109/ACCESS.2021.3052876.

[17] Z. Zhao, Y. Song, F. Cui, J. Zhu, C. Song, Z. Xu, and K. Ding, "Point cloud features-based kernel SVM for human-vehicle classification in millimeter wave radar," *IEEE Access*, vol. 8, pp. 26012–26021, 2020, doi: 10.1109/ACCESS.2020.2970533.

[18] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, "Vehicle tracking and speed estimation from roadside LiDAR," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5597–5608, 2020, doi: 10.1109/JSTARS.2020.3024921.

[19] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS J. Photogramm. Remote Sens.*, vol. 196, pp. 146–177, Feb. 2023, doi: 10.1016/j.isprsjprs.2022.12.021.

[20] Texas Instruments. (Jan. 18, 2019). *MMwave SDK User Guide*. Accessed: Sep. 12, 2023. [Online]. Available: https://https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/7801.mmwave_5F00_sdk_5F00_user_5F00_guide.pdf

[21] Y. Liu, B. Tian, S. Chen, F. Zhu, and K. Wang, "A survey of vision-based vehicle detection and tracking techniques in ITS," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Dongguan, China, Jul. 2013, pp. 72–77, doi: 10.1109/ICVES.2013.6619606.

[22] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011, doi: 10.1109/TITS.2011.2119372.

[23] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2681–2698, Oct. 2016, doi: 10.1109/TITS.2016.2530146.

[24] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds, and S. Atkinson, "A vision-based pipeline for vehicle counting, speed estimation, and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7547–7560, Dec. 2021, doi: 10.1109/TITS.2020.3004066.

[25] W. Liu, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), Amsterdam, The Netherlands, 2016, p. 2137, doi: 10.1007/978-3-319-46448-0_2.

[26] L. Chen, S. Lin, X. Lu, D. Cao, H. Wu, C. Guo, C. Liu, and F.-Y. Wang, "Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3234–3246, Jun. 2021, doi: 10.1109/TITS.2020.2993926.

[27] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," 2018, *arXiv:1811.12808*.

[28] H. Bischof, "Autonomous multi-sensor vehicle classification for traffic monitoring," in *Data and Mobility* (Advances in Intelligent and Soft Computing), vol. 81, J. Dh, H. Hufnagl, E. Juritsch, R. Pfliegl, H. K. Schimany, H. Schnegger, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-15503-1_2.

[29] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May 2014, pp. 1836–1843, doi: 10.1109/ICRA.2014.6907100.

[30] I.-S. Weon, S.-G. Lee, and J.-K. Ryu, "Object recognition based interpolation with 3D LiDAR and vision for autonomous driving of an intelligent vehicle," *IEEE Access*, vol. 8, pp. 65599–65608, 2020, doi: 10.1109/ACCESS.2020.2982681.

[31] D. Roy, Y. Li, T. Jian, P. Tian, K. Chowdhury, and S. Ioannidis, "Multi-modality sensing and data fusion for multi-vehicle detection," *IEEE Trans. Multimedia*, vol. 25, pp. 2280–2295, 2023, doi: 10.1109/TMM.2022.3145663.

[32] W. Zhang, K. Liu, and H. Li, "Traffic vehicle detection by fusion of millimeter wave radar and camera," in *Proc. 3rd Int. Conf. Inf. Sci., Parallel Distrib. Syst. (ISPDS)*, Guangzhou, China, Jul. 2022, pp. 105–108, doi: 10.1109/ISPDS56360.2022.9874115.

[33] F. Cui, Q. Zhang, J. Wu, Y. Song, Z. Xie, C. Song, and Z. Xu, "Online multipedestrian tracking based on fused detections of millimeter wave radar and vision," *IEEE Sensors J.*, vol. 23, no. 14, pp. 15702–15712, 2023, doi: 10.1109/JSEN.2023.3255924.

[34] Y. Liu and Y. Liu, "A data fusion model for millimeter-wave radar and vision sensor in advanced driving assistance system," *Int. J. Automot. Technol.*, vol. 22, no. 6, pp. 1695–1709, Dec. 2021, doi: 10.1007/s12239-021-0146-8.

[35] Z. Wang, X. Miao, Z. Huang, and H. Luo, "Research of target detection and classification techniques using millimeter-wave radar and vision sensors," *Remote Sens.*, vol. 13, no. 6, p. 1064, Mar. 2021, doi: 10.3390/rs13061064.

[36] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie, "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 21329–21338, doi: 10.1109/CVPR52688.2022.02067.

[37] C. Creß, W. Zimmer, L. Strand, M. Fortkord, S. Dai, V. Lakshminarasimhan, and A. Knoll, "A9-dataset: multi-sensor infrastructure-based dataset for mobility research," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Aachen, Germany, Jun. 2022, pp. 965–970, doi: 10.1109/IV51971.2022.9827401.

[38] S. Busch, C. Koetsier, and J. Axmann. (2022). *Dataset: LUMPI: The Leibniz University Multi-Perspective Intersection Dataset*. [Online]. Available: https://doi.org/10.25835/z54qcu1b

[39] X. Ye, M. Shu, H. Li, Y. Shi, Y. Li, G. Wang, X. Tan, and E. Ding, "Rope3D: The roadside perception dataset for autonomous driving and monocular 3D object detection task," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 21309–21318, doi: 10.1109/CVPR52688.2022.02065.

[40] J. Sochor, J. Španhel, and A. Herout, "BoxCars: Improving fine-grained recognition of vehicles using 3-D bounding boxes in traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 97–108, Jan. 2019, doi: 10.1109/TITS.2018.2799228.

[41] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The Ko-PER intersection laserscanner and video dataset," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 1900–1901, doi: 10.1109/ITSC.2014.6957976.

[42] H. Wang, X. Zhang, Z. Li, J. Li, K. Wang, Z. Lei, and R. Haibing, "IPS300+: A challenging multi-modal data sets for intersection perception system," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2539–2545.

[43] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2802–2819, Aug. 2019, doi: 10.1109/TITS.2018.2872502.

[44] J. Wang, T. Fu, J. Xue, C. Li, H. Song, W. Xu, and Q. Shangguan, "Realtime wide-area vehicle trajectory tracking using millimeter-wave radar sensors and the open TJRD TS dataset," *Int. J. Transp. Sci. Technol.*, vol. 12, no. 1, pp. 273–290, Mar. 2023.

[45] F. Jin, A. Sengupta, S. Cao, and Y.-J. Wu, "Mmwave radar point cloud segmentation using GMM in multimodal traffic monitoring," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Washington, DC, USA, Apr. 2020, pp. 732–737, doi: 10.1109/RADAR42522.2020.9114662.

[46] B. Yang, H. Zhang, Y. Chen, Y. Zhou, and Y. Peng, "Urban traffic imaging using millimeter-wave radar," *Remote Sens.*, vol. 14, no. 21, p. 5416, Oct. 2022, doi: 10.3390/rs14215416.

[47] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Understand.*, vol. 193, Apr. 2020, Art. no. 102907.

[48] Commons Creative. (2021). *Attribution-Noncommercial-Noderivatives 4.0 International (Cc by-Nc-Nd 4.0)*. Accessed: Jun. 26, 2023. [Online]. Available: https://creativecommons.org/licenses/by-nc-nd/4.0/

[49] (2021). *National Oceanic and Atmospheric Administration*. Accessed: Jun. 26, 2023. [Online]. Available: https://forecast.weather.gov/glossary.php?word=Sky%20Condition

[50] Nvidia. (Aug. 22, 2022). *Jetson Nano Developer Kit*. Accessed: May 2, 2023. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/

[51] Nvidia. *Jetson Nano Developer Kit*. Accessed: Aug. 22, 2022. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[52] Sony. *IMX219*. Accessed: Aug. 22, 2022. [Online]. Available: https://www.gophotonics.com/products/CMOS-image-sensors/sony-corporation/21-209-imx219

[53] Texas Instruments. *AWR1843BOOST*. Accessed: Aug. 22, 2022. [Online]. Available: https://www.ti.com/tool/AWR1843BOOST

[54] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011, doi: 10.1109/TPAMI.2010.161.

[55] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[56] S. Raj and D. Ghosh, "Improved and optimal DBSCAN for embedded applications using high-resolution automotive radar," in *Proc. 21st Int. Radar Symp. (IRS)*, Warsaw, Poland, Oct. 2020, pp. 343–346, doi: 10.23919/IRS48640.2020.9253774.

[57] A. Manjunath, Y. Liu, B. Henriques, and A. Engstle, "Radar based object detection and tracking for autonomous driving," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Munich, Germany, Apr. 2018, pp. 1–4, doi: 10.1109/ICMIM.2018.8443497.

[58] T. D. Bufler and R. M. Narayanan, "Radar classification of indoor targets using support vector machines," *IET Radar, Sonar Navigat.*, vol. 10, no. 8, pp. 1468–1476, Oct. 2016, doi: 10.1049/iet-rsn.2015.0580.

[59] A. Diab, R. Kashef, and A. Shaker, "Deep learning for LiDAR point cloud classification in remote sensing," *Sensors*, vol. 22, no. 20, p. 7868, 2022, doi: 10.3390/s22207868.

[60] Y. Sun, H. Zhang, Z. Huang, and B. Liu, "R2P: A deep learning model from Mmwave radar to point cloud," 2022, *arXiv:2207.10690*.

[61] Scikit-Learn Developers. (Aug. 22, 2022). *Support Vector Machines*. Accessed: Apr. 2, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/SVM

[62] R. Shi, K. N. Ngan, and S. Li, "Jaccard index compensation for object segmentation evaluation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 4457–4461, doi: 10.1109/ICIP.2014.7025904.

[63] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Niteroi, Brazil, Jul. 2020, pp. 237–242, doi: 10.1109/IWSSIP48289.2020.9145130.

JÜRGEN SOOM received the B.Sc. degree in computer systems from Tallinn University of Technology, in 2017, and the M.Sc. degree, in 2020. He is currently an Early-Stage Researcher with the Embedded AI Research Laboratory. His current Ph.D. thesis is related to computer vision, machine learning, and embedded hardware.

MAIRO LEIER received the Ph.D. degree in computer systems from Tallinn University of Technology, in 2016. He was a Research Scientist with the Department of Computer Systems, Tallinn University of Technology, where he currently leads the Embedded AI Research Laboratory. His current research interests include machine learning on embedded systems, optimization techniques, and edge computing.

KARL JANSON received the Ph.D. degree in computer systems from Tallinn University of Technology, in 2021. He is currently with the Embedded AI Research Laboratory. His current research interests include machine learning, radars, embedded hardware, edge computing, and signal processing.

JEFFREY A. TUHTAN (Member, IEEE) received the B.Sc. degree in civil engineering from California Polytechnic University, San Luis Obispo, CA, USA, in 2004, and the M.Sc. and Dr.-Eng. degrees in water resources engineering and management from the Institute for Modelling Hydraulic and Environmental Systems, University of Stuttgart, Germany, in 2007 and 2011, respectively. Since 2016, he has been leading the Centre for Environmental Sensing and Intelligence, Department of Computer Systems, Tallinn University of Technology. His research interests include data-driven modeling and bio-inspired underwater sensing in extreme environments.