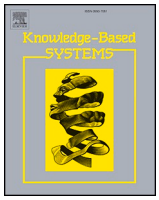
Knowledge-Based Systems 315 (2025) 113334

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



New generation thermal traffic sensor: A novel dataset and monocular 3D thermal vision framework

Arnd Pettirsch [a], Alvaro Garcia-Hernandez [a,*]

[a] Institute of Highway Engineering, RWTH Aachen University, 52074, Aachen, Germany

ARTICLE INFO

Keywords:
Thermal imagery
traffic data collection
monocular 3d detection
roadside cameras
projection

ABSTRACT

Applications like traffic safety analysis require highly accurate trajectory data in world coordinates of traffic participants. While systems like LiDAR or stereo cameras can provide such data, they are costly, sensitive to weather and lighting conditions, and may raise privacy concerns. Thermal roadside cameras offer a robust, privacy-compliant alternative. However, monocular thermal cameras face challenges due to the ambiguous relationship between pixel coordinates and world coordinates. Existing methods for monocular 3D detection from RGB roadside cameras often rely on simplifications or the complex task of depth estimation, which limits their effectiveness. Additionally, no dataset currently exists for monocular 3D detection using thermal roadside imagery. This work introduces a dataset of 9,591 thermal images annotated in 3D world coordinates, including detailed camera calibration and surface models. It proposes a lightweight neural network architecture leveraging a projection-based method to incorporate road surface information. By detecting bottom-center contact points in image space and projecting them into 3D, the presented framework efficiently estimates object's position, dimensions, and orientations in 3D. The presented approach outperforms homography-based methods by 25 percentage points in mean average precision (mAP). It achieves real-time performance with 54 FPS on a GPU server and 17 FPS on an NVIDIA Jetson Xavier NX, making it suitable for edge deployment. Unlike RGB-based systems, our method ensures data privacy and remains effective in diverse weather and lighting conditions, enabling reliable trajectory analysis and near-miss detection for traffic safety applications. Readers can find the dataset here: https://doi.org/10.17632/tw6ghtv624.1. The code used in this work is available here: https://github.com/4rnd25/new_generation_thermal_traffic_sensor.

1. Introduction

Advances in modern object detection algorithms and hardware development have enabled access to larger volumes of traffic data and the use of live data for traffic management applications, forecasting, and maintenance models [1]. This traffic data is often limited to information on the presence of objects, counting data, or 2D trajectories in pixel coordinates. Some applications require more accurate trajectory data, for example, traffic safety analysis with Surrogate Safety Measures [2] or systems that estimate the pavement loads [3]. This requires 3D detection, which involves determining an object's dimensions, position, and orientation in three-dimensional space [4].

Some sensors, such as LiDARs or stereo cameras, are able to detect this kind of data in the full detection range [5]. Nevertheless, those kinds of sensors are costly and sensitive to weather conditions (rain, fog, snow) or, in the case of stereo cameras, to lighting conditions. For some types of studies, situations with poor lighting may be particularly interesting. For example, in safety analysis, it is known that situations with poor lighting are crucial for pedestrian safety [6]. Thermal imagery overcomes those disadvantages since it is more robust against different weather conditions and does not need external lighting [7]. Moreover, it does not collect personal data and fulfills all privacy data protection rules common in many countries. This paper describes a method to extract highly accurate 3D data from monocular thermal cameras mounted in the traffic infrastructure.

Monocular 3D detection includes two tasks. On the one hand, the detection of objects in the image plane and, on the other hand, the transfer of those detections to the real object's position, dimension, and orientation in a global coordinate system such as UTM [8]. This requires a camera calibration, which can be obtained using various methods (e.g., target-based techniques employing calibration boards, targetless approaches leveraging natural scene features, or online methods that

* Corresponding author.
E-mail address: alvaro@isac.rwth-aachen.de (A. Garcia-Hernandez).

https://doi.org/10.1016/j.knosys.2025.113334
Received 14 January 2025; Received in revised form 1 March 2025; Accepted 9 March 2025
Available online 10 March 2025

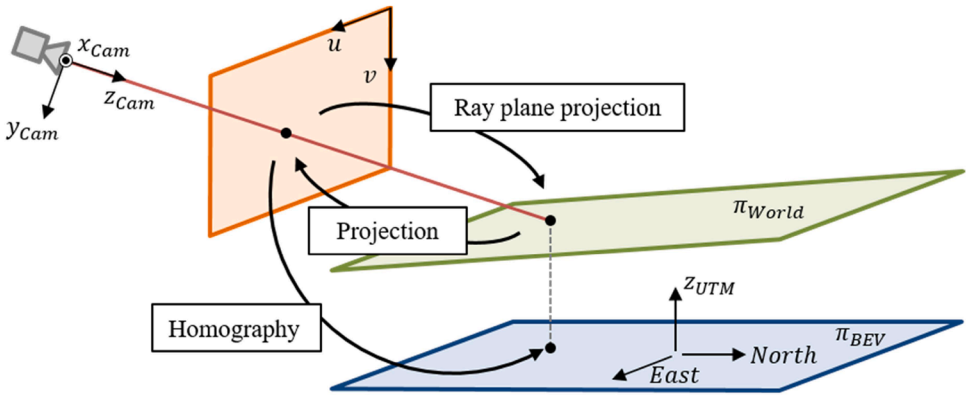


Fig. 1. Description of the various coordinate systems used (pixel coordinates, camera coordinates, world coordinates), the planes (image plane, world plane and BEV plane) and the transformation methods. (Figure based on [26]).

adapt in real-time) [9]. Those are not considered in detail in this work. The connection between the 2D image and the 3D scene is generally ambiguous [10]. This makes the monocular 3D object detection task challenging. However, some methods based on machine learning exist to solve this task. All those methods highly rely on the amount of training data [11]. If we evaluate the publicly available datasets for monocular object detection in the application of traffic detection from roadside cameras, one can find some datasets like the Rope3D dataset [12], the DAIR-V2X [13], and the A9 Intersection Dataset [14] available online. However, those existing datasets lack detailed elevation information of the road surface and the surrounding environment and are limited to the field of RGB images. To the best of our knowledge, no dataset exists for monocular 3D object detection in traffic scenes using thermal imaging.

While the importance of high-quality data cannot be overstated, developing robust algorithms for 3D object detection is also critical. In the area of detection with roadside cameras, as well as in other areas, such as autonomous driving, researchers have proposed methods based on sensor fusion [15], viewpoint fusion of stereo [16] or even multiple cameras [17]. In the field of monocular 3D detection, methods include pseudo-LiDAR point clouds [17] or predefined vehicle models [18], which require extra training data or are limited to the amount of available templates. Roadside camera research follows two main paths.

One type of algorithm, such as [19], is based on depth estimation and is similar to those for applications such as autonomous driving. Yang et al [20], for example, evaluated their work on the roadside datasets DAIR-V2X [13] and Rope3D [12], as well as the NuScenes datasets [21] for autonomous driving. These algorithms leverage knowledge transfer from different domains but overlook a key characteristic of roadside cameras: their fixed positions, which allow them to effectively utilize ground surface information. This leads to the second type of algorithm, those that transfer the detection from the image plane to the world coordinate system using known infrastructure and perspective transformation. However, currently, to the best of our knowledge, all of them rely on a broad variety of assumptions and simplifications. Rezaei et al [22], for example, built their system up on 2D detection with geometric constraints based on fixed dimensions and simplified bottom point estimation. Nevertheless, their homography-based transfer of image coordinates to world space demonstrated low center position errors [22]. Similar homography-based approaches were used e.g., in [23], while their work lacks a dimension and orientation estimation and projects non-bottom cuboid centers to the bottom plane, leading to non-valid projections. Clausse et al [24] further used optimization between masked objects and back projection to obtain refined positions and orientations but also rely on predefined object dimensions. A more flexible model that also estimates the box dimensions is CenterLoc3d [25]. However, their projection simplifies the surrounding environment to a flat earth with third dimension equal to 0 for every bottom data point, which limits its accuracy on uneven roads [25]. Furthermore, the methods described above were not developed for edge devices, making them difficult to use in mobile traffic sensors.

This work addresses two key challenges in monocular 3D detection for thermal roadside cameras: the absence of algorithms for monocular 3D detection tailored to thermal roadside cameras and the lack of real-time, lightweight models for accurately estimating 3D object positions, dimensions, and orientations. To bridge these gaps, we introduce the first thermal 3D traffic dataset with 9,591 annotated images, including 3D world coordinates, camera calibration data, and surface models. We propose a projection-based detection method that directly maps object bottom centers to world coordinates, eliminating complex depth estimation and improving accuracy. Our YOLOv7-based framework enables real-time monocular 3D detection making it suitable for edge deployment. Unlike RGB-based models, our approach remains privacy-compliant and robust under varying lighting and weather conditions. The framework is particularly valuable for traffic monitoring, enabling accurate trajectory analysis, speed estimation, and near-miss detection.

2. Methodology

2.1. Problem description

Traffic objects can be approximated by cuboids that tightly enclose them, defined by their dimensions, rotation, and translation. For roadside cameras, objects are assumed to rest on a locally flat ground plane, simplifying rotation to the yaw angle. This paper addresses the simultaneous detection of a cuboid's coordinates (x_{center}, y_{center}, z_{center}), the dimensions (w, h, l) and the yaw angle (θ) using images from a monocular roadside thermal camera with known ground surface geometry.

2.2. Coordinate transformation

This work aims to map objects from the image coordinate system to a global world coordinate system, yielding two representations: their actual position on the road surface, in the following description, simplified as a single plane π_{World}, and their top-down projection onto the ground plane the birds-eye-view (BEV) denoted as π_{BEV}.

Three coordinate systems are involved (see Fig. 1). The image coordinate system originates at the top-left corner, with axes u (horizontal) and v (vertical). The camera coordinate system, centered at the camera lens, uses axes x (right), y (down), and z (forward along the optical axis). For geospatial localization, the Universal Transverse Mercator (UTM) system divides the Earth into 60 longitudinal zones, each using a cartesian coordinate system with easting (E), northing (N), and elevation (Z) measured from a central meridian specific to each zone [8].

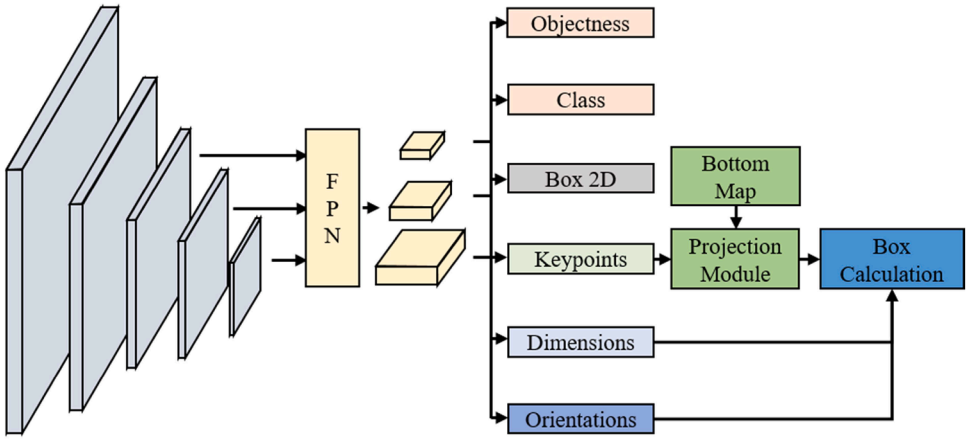


Fig. 2. The neural network architecture processes images sequentially from left to right, starting with the Yolov7-tiny backbone and neck. Heads for classification, 2D bounding box regression, bottom-center keypoint detection, and dimension and orientation regression, follow this. The projection module maps image keypoints to world space, enabling the calculation of 3D boxes using the dimensions and orientations.

2.2.1. Correcting image distortion

All transformations use undistorted pixel coordinates. Image distortion has two primary sources: radial and tangential distortion. Radial distortion bends straight lines due to lens shape and varying light refraction in the lens center and lens edges. Radial distortion is modeled by (1) using distortion coefficients k_1, k_2 and k_3 with x and y being undistorted coordinates and $r^2 = x^2 + y^2$ [27]

$$\begin{pmatrix} x_{distorted} \\ y_{distorted} \end{pmatrix} = \begin{pmatrix} x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \\ y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \end{pmatrix} \tag{1}$$

Tangential distortion occurs from lens misalignment, tilting the image plane. It is modeled by parameters p_1 and p_2 as described in (2) [27]

$$\begin{pmatrix} x_{distorted} \\ y_{distorted} \end{pmatrix} = \begin{pmatrix} x + \left[2p_1 xy + p_2\left(r^2 + 2x^2\right)\right] \\ y + \left[p_1\left(r^2 + 2y^2\right) + 2p_2 xy\right] \end{pmatrix} \tag{2}$$

The five distortion coefficients are determined during the calibration process as described in section 2.2.5.

2.2.2. Ray plane projection

As shown in Fig. 1, given a known plane in the real world, pixel coordinates displaying an object touching the ground plane can be projected into 3D space using the plane equation and the ray equation, which describes the ray passing through the camera center and the image point. Based on the work of Hartley and Zisserman in [26], each image point corresponds to a ray of potential matching points in space. For finite cameras, this ray can be described as (3). This requires the projection matrix ($P = (M|p_4)$), which contains camera parameters and is determined during calibration (see section 2.2.5) [26]

$$X(\mu) = \begin{pmatrix} M^{-1}(\mu x - p_4) \\ 1 \end{pmatrix} \tag{3}$$

$x = (u, v, 1)$ describes the pixel point in homogeneous coordinates and $X(\mu) = (x(\mu),\ y(\mu),\ z(\mu),\ 1)$ represent normalized world coordinates. The last row simplifies to $1 = 1$ which allows calculations in Euclidean coordinates. Combining this with the general plane equation in point-normal form (4), where P_0 is a point on the plane and n is the plane's normal vector, yield (5). This relationship provides the scaling factor (μ), combined with (6), where C is the camera center, allowing the projection point $X_{projected}$ computation (7) [26]

$$(x - P_0)\cdot n = 0 \tag{4}$$

$$\left(M^{-1}(\mu x - p_4) - P_0\right)\cdot n = 0 \tag{5}$$

$$-M^{-1}p_4 = C \tag{6}$$

$$X_{projected} = M^{-1}\left(\frac{(P_0 - C)\cdot n}{M^{-1}x\cdot n}x - p_4\right) \tag{7}$$

2.2.3. Planar homography

Planar homography offers an alternative method for projecting pixel coordinates onto a real-world plane. This work applies it to project points onto the plane $z = 0$ and normal vector $(0, 0, 1)$ representing the BEV. For points lying on the same plane in world space, the relationship (8) becomes valid, where x and y are world coordinates in the BEV, u and v are their corresponding pixel coordinates and H is the homography matrix. Given at least four coplanar points in the real world and their corresponding pixel coordinates, the homography matrix can be computed automatically based on a RANSAC optimization as described in [26].

$$s\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{8}$$

2.2.4. Back projection

A 3D world point in homogeneous coordinates ($x = (x, y, z, 1)$) is transformed to the camera coordinate system using rotation (R) and translation (t). The camera calibration matrix (K) projects it onto the image plane. Combining intrinsic (K) and extrinsic ($R,\ t$) parameters into a single projection matrix (P) enables direct projection of x to its image point ($x' = (u,v,1)$) up to a scaling factor (λ), as shown in (9) [26]

$$\lambda\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = Px = K[R|t]x \tag{9}$$

2.2.5. Camera Calibration

P described in (9) has 12 parameters which leads excluding scaling to 11 degrees of freedom. In combination with the five distortion coefficients (see 2.2.1) 16 parameters must be determined to calibrate the camera system and enable all mentioned transformations. This requires at least 8 pixel points (each having two coordinates) to be matched to corresponding world coordinates. This work used manual matching with georeferenced Airborne Laser Scanning (ALS) images from [28] by marking key points like street markings, lamps, etc. in both images. As many points as possible have been used. Nonlinear optimization refined all parameters by minimizing the reprojection error using equation (9), as further described in [26,29].

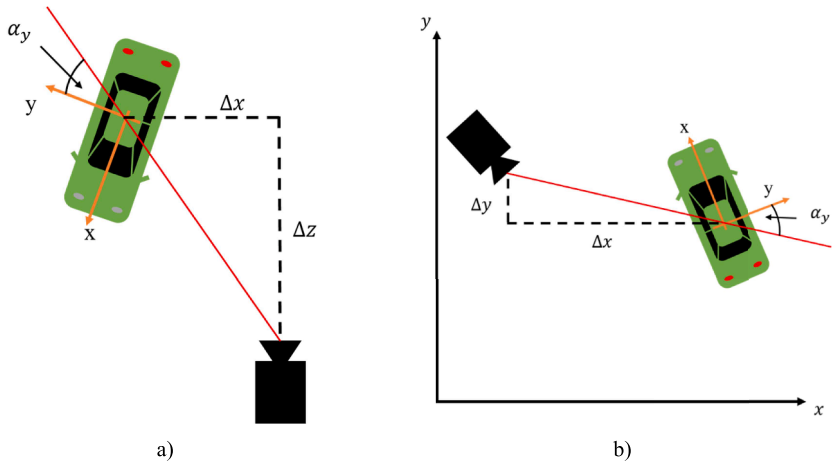


Fig. 3. Definition of the observation angle alpha based on [38] in camera coordinate system (a) and in the global coordinate system (b).

2.3. 3D-Detection Framework

As shown in Fig. 2, the proposed method comprises four main components: the backbone network (blue-gray and yellow), the detection heads (including the classification branch, 2D bounding box branch, keypoint branch, and dimension and orientation regression branch), the projection module (green), and the box calculation module (blue).

2.3.1. Backbone

The proposed methods crucially builds on image keypoint detection. Therefore, based on the success of [30], an anchor-based backbone for keypoint detection YOLOv7-tiny [31] was chosen. The family of YOLO (name after You only look once) is a single stage object detection architecture firstly published by Redmon et al [32]. Recent versions have proven to outperform comparable architectures like SSDLite [33] or EfficientDet [34] in the tradeoff between speed and accuracy [35,36]. In particular, YOLOv7-tiny has proven fast inference on edge devices [35] meeting the practical demands of traffic analysis - such as real-time traffic management requiring fast edge-device evaluation or remote studies relying on battery-powered devices. More recent versions of the YOLO family like YOLOv8 [37] are anchor-free and therefore not feasible for the proposed keypoint detection method.

The proposed method uses Yolov7-tiny's backbone and neck network. The backbone shown in grey in Fig. 2 receives an 640 × 640 image as input and uses convolutional layers, max-pooling and ELAN (Efficient Layer Aggregation Network) to produce three primary feature maps of shapes 80 × 80 × 64, 40 × 40 × 128 and 20 × 20 × 256. The neck further enhances the feature maps using Spatial Pyramid Pooling (SPP) and feature fusion via upsampling and concatenation, improving multi-scale feature aggregation. The refined output consists of three feature maps (80 × 80 × 64), (40 × 40 × 128) and (20 × 20 × 256). Similar to the original Yolov7 implementation the head is processed on the feature maps using a 1 × 1 convolutional [37]

2.3.2. Objectness and Class branch

Similar to the original YOLOv7 implementation, each anchor predicts both classification scores (p_{cls_x}) and an objectness score $\left(p_{obj}\right)$, with the final detection confidence obtained by multiplying these values (12). The raw objectness output (o_{obj}) and class logits (o_{cls_x}) are passed through a sigmoid activation function (σ) (10) and (11), ensuring the confidence remains within the range of 0 to 1 while allowing the assignment of multiple classes to a single object [31]

$$p_{obj} = \sigma\left(o_{obj}\right) \tag{10}$$

$$p_{cls\_x} = \sigma(o_{cls\_x}) \tag{11}$$

$$p = p_{obj} \cdot p_{cls} \tag{12}$$

2.3.3. 2D-Bounding Box Branch

The integration of 2D bounding box regression enables the use of existing fast non-maximum suppression (NMS) algorithms. This not only facilitates real-time performance but also improves detection in cases of occluded objects. Furthermore, the system's practical utility is enhanced by providing 2D detection as a fallback option when camera calibration is unavailable. Following the YOLOv7 approach, the box center (x,y) is predicted as an offset from the grid center (G_x, G_y), with a sigmoid activation ensuring it remains within the grid (13), (14). o_x, o_y are the network outputs and s the stride. The box dimensions (w,h) are scaled relative to anchor boxes, which sizes A_w and A_h are optimized at the start of training based on the YOLOv7 implementation (15), (16). o_w, o_h are the raw network outputs [31]

$$x = (2 \cdot (\sigma(o_x) - 0.5) + G_x) \cdot s \tag{13}$$

$$y = \left(2 \cdot \left(\sigma\left(o_y\right) - 0.5\right) + G_y\right) \cdot s \tag{14}$$

$$w = (2 \cdot \sigma(o_w))^2 \cdot A_w \tag{15}$$

$$h = (2 \cdot \sigma(o_h))^2 \cdot A_h \tag{16}$$

2.3.4. Keypoint Branch

Following [30], for each anchor box a specific keypoint in pixel coordinates is predicted, defined as the 3D center of the object's bottom plane projected into the image. Similar to the 2D bounding box center, the keypoints $\left(x_{kpt}, y_{kpt}\right)$ are detected as offset from the grid center (17), (18). Since this bottom center point can differ strongly from the anchor center, no sigmoid is applied to the network output $\left(o_{x_{kpt}}, o_{y_{kpt}}\right)$ allowing the offset to become bigger than 1 and smaller than 0. Since the keypoint is not calculated based on its closet anchor but based on the anchor belonging to the center of the same object.

$$x_{kpt} = \left(o_{x_{kpt}} + G_x\right) \cdot s \tag{17}$$

$$y_{kpt} = \left(o_{y_{kpt}} + G_y\right) \cdot s \tag{18}$$

2.3.5. Bottom Map & Projection Module

Using the methods from Section 2.2, a bottom map $(B(u,v))$ is generated in advance, linking each pixel (u,v) to its world coordinates (x, y, z_{Bottom}). Combining this map with the estimated image keypoint allows

calculation of world coordinates using equation (19).

$$B(u,v) = (x, y, z_{Bottom}) \tag{19}$$

2.3.6. Dimension branch

Similar to other monocular 3D detection frameworks like [38] box dimensions and orientations are encoded using a tuple of three parameters $(\delta_l, \delta_w, \delta_h)$. These parameters are converted into cuboid dimensions, based on (20), using the class-wise mean dimensions $(\bar{l}, \bar{w}, \bar{h})$ previously computed from the training set, and the natural exponential function (e) [38]

$$\begin{pmatrix} l \\ w \\ h \end{pmatrix} = \begin{pmatrix} \bar{l} \cdot e^{\delta_l} \\ \bar{w} \cdot e^{\delta_w} \\ \bar{h} \cdot e^{\delta_h} \end{pmatrix} \tag{20}$$

The three parameters $(\delta_l, \delta_w, \delta_h)$, slightly differing from [38], are calculated as shown in (21) based on the network's specific outputs (o_l, o_w, o_h) and the sigmoid activation. This ensures the parameters are centered at 0 and constrained within the range of -1 to 1.

$$\begin{pmatrix} \delta_l \\ \delta_w \\ \delta_h \end{pmatrix} = 2 \cdot \sigma \begin{pmatrix} o_l \\ o_w \\ o_h \end{pmatrix} - 1 \tag{21}$$

2.3.7. Orientation Regression Branch

Building on the work by [38], the observation angle, rather than the yaw angle (φ), is estimated. Based on reference [38] the observation angle represents the angle between the object's axis (orthogonal to the direction of travel) and the ray from the camera to the object, projected onto the ground plane, as illustrated in Fig. 3 on the left side. To avoid the transfer to camera coordinate system, in this work the observation angle is defined based on the global coordinate system as described in (23) and Fig. 3 on the right side. Due to the periodic nature of orientation, the estimated observation angle (α_y) is encoded as $(\sin(\alpha_y), \cos(\alpha_y))$ using (22) based on the network outputs o_{sin} and o_{cos}. The yaw angle is then derived using (23) based on the x- and y-coordinates in global world coordinate system.

$$\begin{pmatrix} \sin(\alpha_y) \\ \cos(\alpha_y) \end{pmatrix} = \begin{pmatrix} o_{sin} \Big/ \sqrt{{o_{sin}}^2 + {o_{cos}}^2} \\ o_{cos} \Big/ \sqrt{{o_{sin}}^2 + {o_{cos}}^2} \end{pmatrix} \tag{22}$$

$$\varphi = \alpha_y + \arctan\left(\frac{x}{y}\right) \tag{23}$$

2.3.8. Box Calculation Module

Using the height (h), length (l), and width (w) estimated by the neural network, the eight corners $(Corners_{Obj})$ of the cuboid could be determined in a coordinate system centered at the cuboid's bottom center, as defined by (24).

$$Corners_{Obj} = \begin{pmatrix} 1/2 & -1/2 & 0 \\ 1/2 & 1/2 & 0 \\ -1/2 & 1/2 & 0 \\ -1/2 & -1/2 & 0 \\ 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 \\ -1/2 & -1/2 & 1/2 \end{pmatrix} * \begin{pmatrix} l & w & h \end{pmatrix} \tag{24}$$

These corners $(Corners_{Obj})$ are transformed into the global coordinate $(Corners_{UTM})$ system using the Rotation (R) based on the estimated yaw angle (φ) and the bottom center coordinates (X_{Bottom_Center}) provided by the projection module (25).

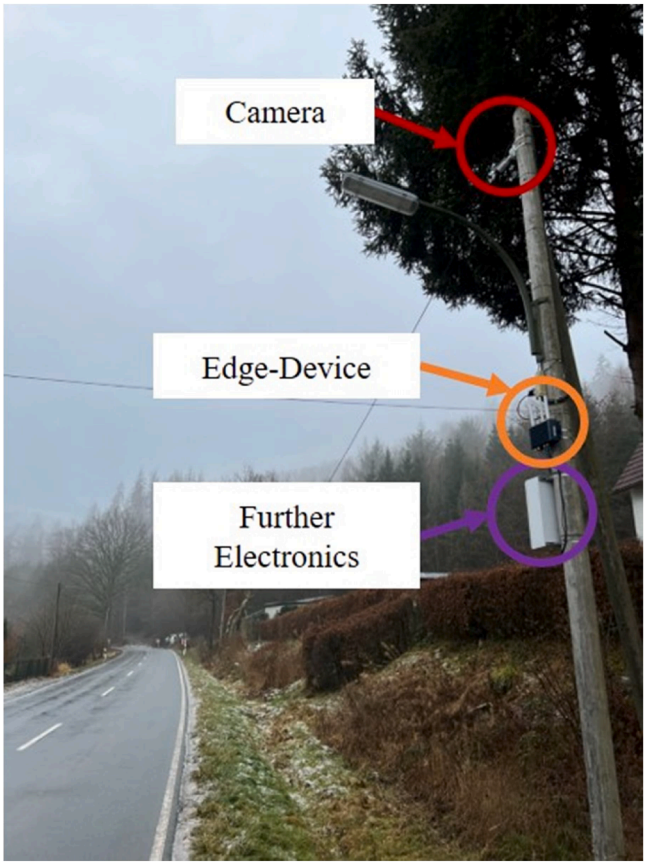


Fig. 4. Photo of one of the thermal images cameras used in this study mounted at a rural road

$$Corners_{UTM} = Corners_{Obj} * R(\varphi) + X_{Bottom\_Center} \tag{25}$$

2.3.9. Loss function

The total loss L consists of five components, three of which - objectness, classification, and 2D bounding box regression - are identical to the original YOLOv7-tiny implementation [31]. These losses are calculated as described in (26), (27) and (28). In case of the objectness loss (L_{Obj}) the parameter y_i is a binary label indicating whether an objects belong to this anchor. In case of the class loss $((L_{Cls})$ y_i is either 0 or 1 depending of the objects class. p belongs to the predicted objectness score or class probability respectively. p_{Box}, t_{Box} belong to the predicted and target box coordinates.

$$L_{Obj} = -[y_i \, log(p) + (1 - y_i) log(1 - p)] \tag{26}$$

$$L_{Cls} = -\sum_{i=1}^{C} y_i \, log(p_i) \tag{27}$$

$$L_{reg\_2D\_Box} = \sum_{i=1}^{n} 1 - IoU(p_{Box}, \; t_{Box}) \tag{28}$$

The total loss combines all five terms, weighted by factors $\alpha, \beta, \gamma, \delta$ and ε:

$$L = \alpha L_{obj} + \beta L_{cls} + \gamma L_{reg\_2D\_Box} + \delta L_{kpt} + \varepsilon L_{Corner} \tag{29}$$

Due to the projection module, there is no learned relationship between the keypoint and the object's position in world coordinates. Separate losses are used for these tasks. Inspired by [30], the Object Keypoint Similarity (OKS) term was applied for keypoint loss. OKS normalizes the accuracy by a scaling factor (s), based on the size of the

Fig. 5. Different locations used in the dataset with different weather and traffic situations (a) Loc. 1: Dry weather with many pedestrians in inner city (b) Loc. 2: Rain in inner city (c) Loc. 3: Sun in inner city (d) Loc. 4: Rain / wet surface at rural road

Table 1
Dataset split distribution per Camera location

Location	Street type	#Images Train	#Images Val	#Images Test	#3DBoxes (total)
Loc. 1	Inner city	3,072	1,024	1,022	35,744
Loc. 2	Inner city	1,731	577	576	10,545
Loc. 3	Inner city	631	210	209	3,475
Loc. 4	Rural road	330	110	108	679
Total		5,764	1,921	1,906	50,443

2D bounding box. Using the distance (d) between the true and predicted keypoint the loss is:

$$L_{kpt} = 1 - \ OKS = 1 - \ e^{\left(\frac{d}{2s^2}\right)} \tag{30}$$

The cuboid corners are for the loss calculated in local coordinates centered at the ground contact point (24), without adding $\boldsymbol{X}_{Bottom\_Center}$ for each object. Following [38], the mean Euclidean distance between the ground truth and predicted corners serves as the loss (31). This loss depends only on the yaw angle and object dimensions. The yaw angle is calculated using the ground truth ground position and the estimated observation angle (α_y), to maintain separation between 3D regression and keypoint estimation. The corner loss, based on each ground-truth (x_{GT}, y_{GT}, z_{GT}) and predicted corner points $(x_{pred}, y_{pred}, z_{pred})$ is:

$$L_{corn} = \ \frac{1}{8} \ \sum_{1}^{8} \sqrt{\left(x_{GT} - x_{pred}\right)^2 + \left(y_{GT} - y_{pred}\right)^2 + \left(z_{GT} - z_{pred}\right)^2} \tag{31}$$

2.4. Experimental Design and Data collection

2.4.1. Data collection

Two types of Axis thermal imaging cameras, AXIS Q1952-E 10 mm [39] and AXIS Q1942-E 10 mm [40], were installed at four locations to collect data. Both cameras have a resolution of 640 × 480 pixels and record at 30 frames per second. Images were decoded in 8-bit format, with pixel values ranging from 0 (coldest) to 255 (hottest). Cameras were mounted on street lamps or masts, 5-8 meters above the ground and positioned a few meters from the road, as shown in Fig. 4. Three cameras were placed in inner-city locations, and one on a rural road. Recordings were conducted under varying times and weather conditions, creating a heterogeneous dataset. Fig. 5 provides example images from each location. Approximately 60% of the images from each camera were used for training, 20% for validation, and 20% for testing. A more in detailed dataset contribution is provided in. Table 1.

2.4.2. Data annotation

Annotation was performed manually using a self-developed tool (see Fig. 6), which processes video frames at adjustable frequency and enables users to place 3D boxes in the images. An initial box with initial dimensions and yaw angle is placed on the known road surface. This box, defined in global coordinates, is projected onto the image plane using camera calibration (see section 2.2). When the user adjusts the box position in the pixel plane, the bottom center is mapped to the road surface (as detailed in section 2.2) and re-projected onto the image plane. Changes in dimensions or yaw angle are applied in global coordinates, with immediate updates to the image projection. This approach ensures all object bottom centers align with the ground surface and resolves ambiguities between pixel and world coordinates. Once the projection tightly fits the object, the user confirms the box. Each annotated box, paired with a class label (motorcyclist, car, truck, bus, pedestrian, cyclist, or e-scooter rider), constitutes the ground-truth data. It is notable that this approach requires that the object's bottom center is part of the image. Annotations were made for all objects which bottom

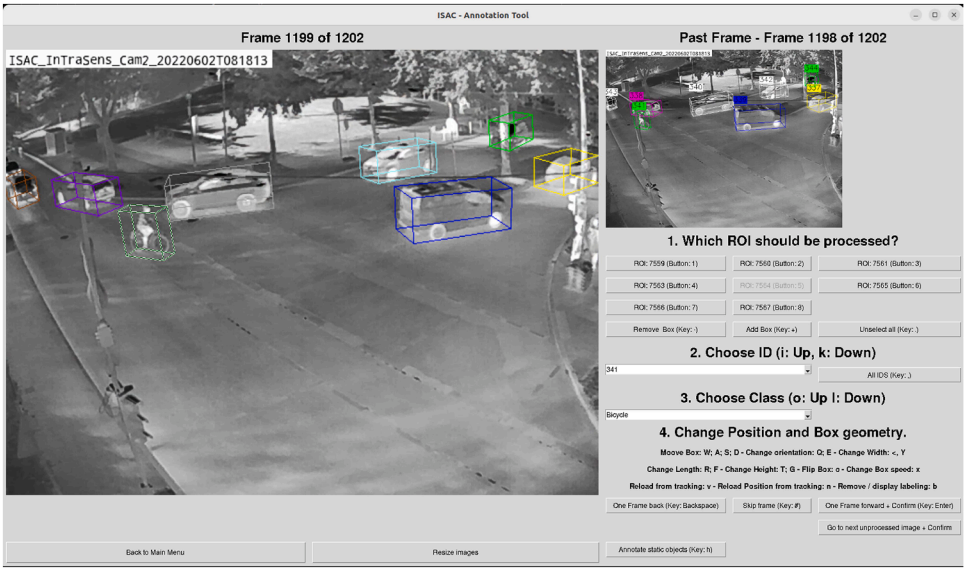


Fig. 6. Example from the annotation tool during the annotation process. The left side shows the current frame, on the right side there is the past frame as well as all possible actions for the user.

center lies in the image, even if the objects were obscured, but which nevertheless stood out clearly from the background and were judged by the user to be clearly meaningful.

2.4.3. Infrastructure data

The surface information in this work is derived from Airborne Laser Scanning (ALS) data provided by geoportal.nrw [28], a type of data widely available in many countries [41]. Since not all points in the ALS data belong to the road surface the data is first manually filtered. Points beside the road and points above the road caused by trees or other elevated features were filtered. After filtering, the data is processed using Delaunay triangulation [42] to create a surface mesh and further refined using Open3D's Laplacian smoothing filter [43]. Areas besides the road were assigned to the closest triangulated plane. While many advanced filtering algorithms for ALS data have been developed, as for example [44], exploring these techniques in detail exceeds the scope of this work. Simplified surface models with significantly reduced data points were evaluated to assess the robustness of the presented methodology in terms of variations in filtering and point cloud density.

2.4.4. Implementation details

This work builds upon the YOLOv7 implementation by its original authors [32] and Pytorch3D evaluation implementation [39]. Additional algorithms, data processing techniques, and evaluation methods were integrated, significantly modifying the base code. The source code, with all training, testing evaluation scripts and hyperparameters, as well as documentation and setup introductions for all experiments are available at: https://github.com/4rnd25/new_generation_thermal_traffic_sensor.

Training and evaluation used 640 × 640 images, resized using the methods in [34]. Non-maximum suppression (NMS) was not class-agnostic and utilized a score threshold of 0.001 and an IoU threshold of 0.65. Models were trained for 300 epochs, with the best model selected based on the fitness score from [32], adapted to include mAP based on 3D-IoU from the validation set.

Training and inference times were measured on a server with an NVIDIA Quadro RTX 5000 GPU and an Intel Xeon E5-2640 v3 CPU (2.60 GHz, 16 cores). Inference time was also tested on an NVIDIA Jetson Xavier NX Edge GPU in 15W 6-core Power mode with Jetson Clocks activated.

2.5. Dataprocessing and anlysis

2.5.1. Dataset analysis

The dataset was divided into three distance classes for evaluation. Following the approach of DAIR-V2X [13] the distance classes used are 0-30m, 30-50m and 50-100m.

2.5.2. Evaluation metrics

2.5.2.1. Overall 3D performance. Many 3D detection tasks utilize average precision (AP) as an evaluation metric. AP represents the area under the curve in a Precision-Recall diagram. Precision (P) measures the ratio of true-positive (TP) detections to all detections, including false positives (FP), while recall (R) measures the ratio of true positives to all ground truth objects, including false negatives (FN). These are calculated, based on reference [45] as:

$$P = \frac{TP}{TP + FP} \tag{32}$$

$$R = \frac{TP}{TP + FN} \tag{33}$$

The AP is derived by interpolating precision values at specific recall levels (r) and is calculated based on [45] as:

$$AP = \frac{1}{R} \sum_{r \in \{0, 0.1, \ldots, 1\}} P_{interp}(r) \tag{34}$$

AP evaluates classification quality but relies on the matching metric between ground truth (GT) and predicted detections for further information. This work uses the 3D-IoU and bird's eye view (BEV) IoU metrics. The 3D-IoU is defined in reference [45] as:

$$IOU_{3D} = \frac{Volume\ of\ Intersection}{Volume\ of\ Union} \tag{35}$$

The BEV IoU simplifies this by projecting objects to the ground plane calculating the Areas of Intersection and Union of the object's ground planes, calculated based on reference [45]:

$$IOU_{BEV} = \frac{(Area\ of\ Intersection)_{BEV}}{(Area\ of\ Union)_{BEV}} \tag{36}$$

The combination of these metrics provides a comprehensive

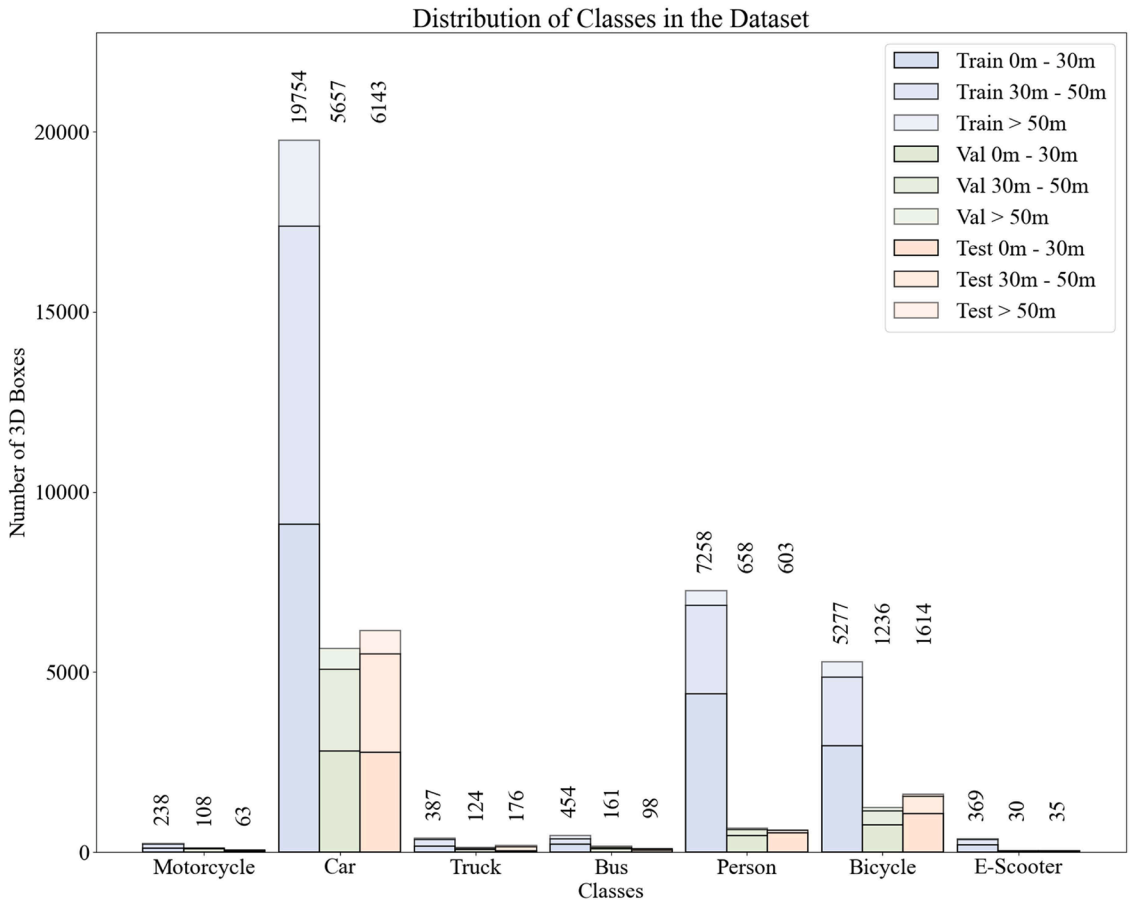


Fig. 7. Distribution of the data set based on the number of 3D boxes with respect to the classes encountered and the difficulty of recognizing the objects.

evaluation, with 3D-IoU assessing overall algorithm performance and BEV IoU addressing practical applications like traffic safety, where object relationships matter more than height.

2.5.2.2. Detailed evaluation of Box quality. Similar to [22] the quality of detected boxes is further evaluated using the mean squared error (MSE) of the 3D bottom center position (MSE_{pos}) using predicted (X_{Pred}) and ground truth position (X_{GT}) and the heading angle (MSE_{Yaw}) using predicted (φ_{Pred}) and ground truth yaw angle (φ_{GT}). As well as the mean absolute percentage error (MAPE) of the 3D box sizes ($MAPE_{Size}$) based on predicted (V_{Pred_i}) and ground-truth volumes (V_{GT_i}). These are defined as:

$$MSE_{pos} = \frac{1}{n}\sum_{1}^{n} \|X_{Pred} - X_{GT}\|^2 \tag{37}$$

$$MSE_{Yaw} = \frac{1}{n}\sum_{1}^{n} \|\varphi_{Pred} - \varphi_{GT}\|^2 \tag{38}$$

$$MAPE_{Size} = \frac{1}{n}\sum_{1}^{n} \left|\frac{V_{Pred\_i} - V_{GT\_i}}{V_{GT\_i}}\right| \cdot 100 \tag{39}$$

3. Results and Discussion

3.1. Dataset analysis

The training, validation, and testing subsets were analyzed for class distribution and object distance from the camera. The Car class dominates across all subsets, reflecting its prevalence in traffic data. In contrast, classes like Motorcycle, Truck, Bus, and E-Scooter are rarely

Table 2
Comparison of the proposed dataset with different roadside datasets from the RGB and thermal field.

Dataset	Scene Type	Sensors	Annotation in 3D	# Annotated Images	# Objects	Conditions
DAIR V2X-I [13]	Roadside city and highway	RGB Camera, Lidar	Yes	10k	493k	Diverse weather and lighting no further descried
Rope3D [12]	Roadside	RGB Camera, Lidar	Yes	50k	1.5 M	Day / night / dusk; rainy / sunny / cloudy
A9 Intersection dataset [14]	Roadside -Urban intersection	RGB Camera, Lidar	Yes	4,8k	57.4k	Night, day, heavy rain, sunny, cloudy
AAU RainSnow [46]	Roadside – Urban intersection	RGB and Thermal camera	Only 2d	2,2k	N/A	Rain and snow
Balon AT [47]	Roadside – Highway	Thermal camera	Only 2D	8k +	35k +	Cloudy only one location and time
Thermal3D (ours)	Roadside – Urban and rural road	Thermal camera	Yes	9,6k	50.4k	Sunny / rainy / cloudy; different traffic conditions

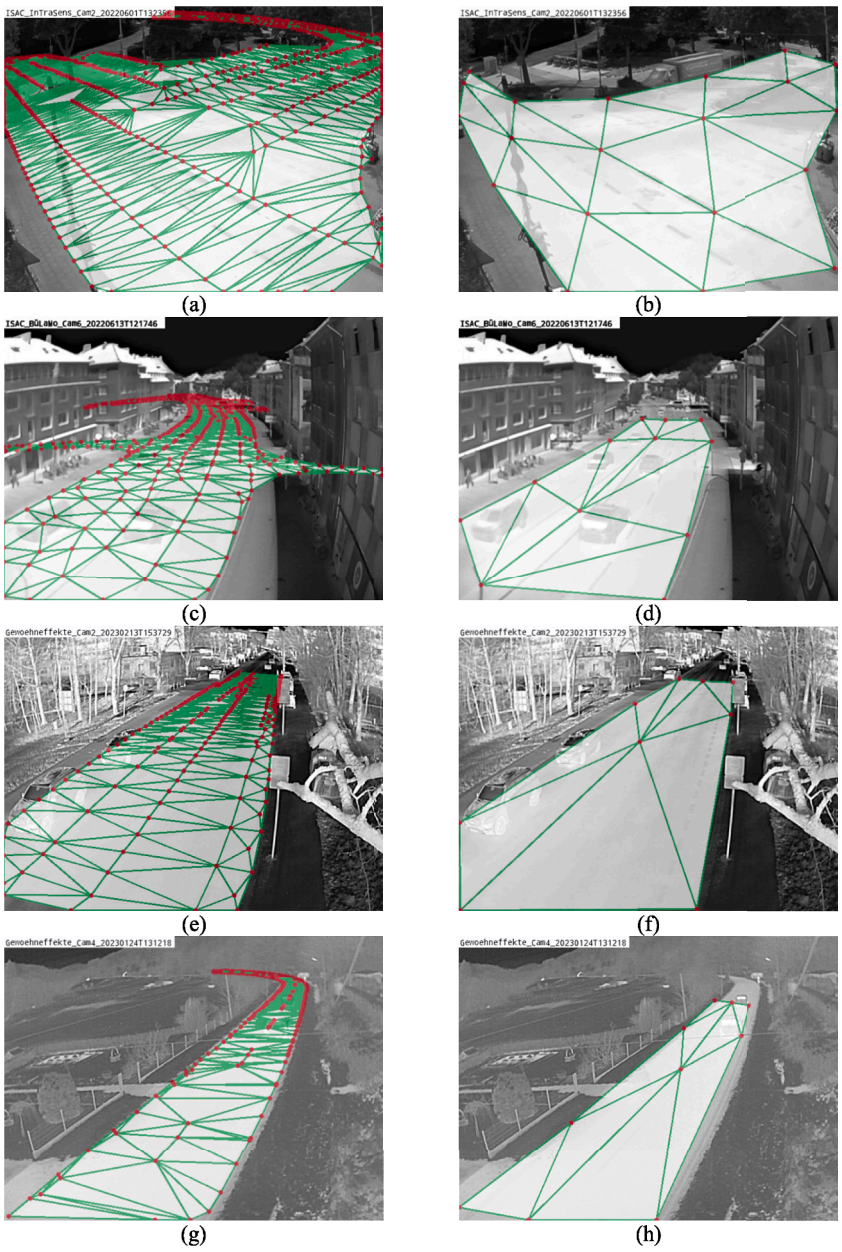


Fig. 8. Comparison of used surfaces information. (a) Whole filtered points cloud Loc 1 (b) Based on 20 points (c) All filtered points in Loc 2 (d) Loc. D with only 12 points used (e) Loc 3 with all filtered points (f) Loc 3 with only 9 Points used (g) Loc. 4 with all points (h) Loc 4 with only 10 points used

represented, especially in testing, making performance metrics for these classes less reliable. Fig. 7 highlights the class distribution and object distances, showing most objects are within 50 meters of the camera. The distances are nearly evenly split within this range, with 25,994 boxes within 30 meters and 19,614 beyond.

3.1.1. Comparison to State-of-the-art datasets

To the best of our knowledge, the Thermal3D dataset is the first roadside object detection dataset using thermal imagery with 3D annotations. Therefore, comparisons to existing datasets can only be made in terms of size and diversity.

As shown in Table 2, compared to existing roadside 3D object detection datasets, Thermal3D is similar in size to DAIR V2X-I [13] (9.6k images vs 10 k images) in terms of the number of images and to A9 Intersection in terms of annotated objects (50.4k to 57.4k). At the same time, Rope3D is significantly larger (50k images with 1.5 M objects). Regarding weather diversity, DAIR V2X-I, Rope3D, and A9 Intersection all include rainy, sunny, and cloudy conditions, though A9 Intersection uniquely includes heavy rain. While Thermal3D covers sunny, rainy, and cloudy weather across different traffic conditions, additional weather conditions such as heavy rain, fog, and snow could further improve dataset robustness and enhance the evaluation of detection methods.

In contrast, roadside thermal traffic datasets like AAU RainSnow and Balon AT are significantly smaller, with only 2.2k and 8k images, respectively. More importantly, unlike Thermal3D, these datasets provide only 2D annotations. This limitation extends to larger thermal datasets from other applications, like e.g., pedestrian detetion, such as CAMEL [48] and MF3D [49], which, despite their size, remain restricted to 2D annotations and are therefore not directly comparable to Thermal3D.

3.1.2. Data quality

In general, our dataset relies on human-made annotations and a

Table 3
Detection results in mAP based on BEV IoU. R stand for Ray-Plane Intersection, H for Homography and S are the corresponding sparser versions.

Model	Motorcycle	Car	Truck	Bus	Person	Bicycle	E-Scooter	Sum
	AP	AP	AP	AP	AP	AP	AP	mAP
ProjNet	36.8	55.1	28.8	24.7	30.9	37.6	34.4	35.5
ProjNet -sparse	36.4	53.9	27.4	24.5	32.4	36.8	32.0	34.8
Homography (baseline)	8.4	10.4	17.1	26.5	2.7	6.8	8.3	11.5
Homography sparse	9.5	11.6	17.0	24.5	1.9	7.1	8,3	11.4

single sensor. Although all annotations are double-checked, some missed detections may still occur due to human error. However, this challenge is inherent to any manually annotated dataset.

3.2. Bottom Map Creation

3.2.1. Plane –ray-intersection

Two plane-ray intersection methods were compared: Method A used the ALS point cloud preprocessed based on Section 2.2, and Method B incorporated road surface knowledge. Triangles were manually defined based on slope consistency. Each triangle in the triangulation was designed to terminate either at the roadway's center or lane boundaries. This ensured that each direction of travel was treated as a separate plane. At intersections, the triangles ended at the contact points where lanes intersected. To achieve this, the corner points of the triangles were strategically placed along lane edges and centers. The manual definition of triangles ensured strict adherence to these rules.

Fig. 8 illustrates the manually defined triangles in comparison to the triangles based on all data points for all locations. This approach requires precise coordinates in this world coordinate system. 20, 12, 9 and 10 points were used for the four locations. This amount is similar to number of points needed for camera calibration. Those, points can be derived from various sources (e.g., ALS data, Satellite systems, manually measured GNSS points) and need, based on (7), not to align with specific image pixels.

3.2.2. Homography based Methods

Similar to the plane-ray-intersection models, homography was used as an alternative to ray-plane intersection for projecting pixel coordinates to the world plane. Two approaches were explored: one utilizing all filtered ALS points within 50 meters of the camera (beyond which pixel-to-meter ratios become less accurate), and another using sparse triangulations with 20, 12, 9, and 10 points at the four specified locations.

3.3. General 3D Detection Performance

Table 3 provides some key insights in the general model performance. The table shows the AP based on a matching threshold of 0.5 for the classes with bigger objects (Car, Truck and Bus) and 0.25 for the smaller classes (Motorcycle, Person, Bicycle and E-Scooter) with matching feature IOU_{BEV}. Those different thresholds for objects from different size are commonly used as for example in [20]. The ProjNet (Ray-plane projection) methods achieves a significant improvement over homography-based approaches, with a mAP of 35.5 percentage points outperforming the best homography-based method by more than 24 percentage points (35.5% vs 11.5%). Homography is by definition only valid for the mapping between two planes [40]. A road is not naturally flat and has gradients and differences in height due to topography and drainage. This simplified assumption of homography is reflected in the performance drop. The gap in performance between the baseline of homography based methods and PRojNet is smaller for bigger objects like Trucks (28.8% vs 17.1%) and Busses (24.7% vs 26.5%). This can be explained by the fact that the positioning error does not scale with the size of the objects and thus is relatively less critical. The slightly better performance (26.5% vs 24.7%) of the homography methods for the bus class indicates random corrections due to incorrect projection and is unique to this specific class and this specific bottom map.

The sparse variant of the ProjNet achieves 34.8 % mAP demonstrating that a sparse point cloud with manually selected points performs similarly (35.5% to 34.8%) to that with significantly more points. (Manual) filtering and the availability of surface information that is as dense as possible has a minor influence. Generally, the distribution of the data in the training data set is clearly evident in the significantly best performance for the car class with a mAP of 55.1 % for RProjNet.

The methods based on ray-plane intersection allow the calculation of mAP based on IOU_{3D} (with the same thresholds as mentioned above) as shown in Table 4. The small performance drop (35.5% vs 30.5% and 30.5% vs. 29.2%) could be explained by the faults in the height estimation and seem to appear for classes which have by nature absolute more difference in their height distribution between objects like different models of cars (55.1% vs 46.0%), trucks (28.8% vs 20.6%), busses (24.7% vs 18.7%) in comparison to classes which include humans like motorcycle (36.8% vs 32.6%), person (30.9% vs 30.0%), Bicycle (37.6% vs 36.6%) and e-scooter (35.5% vs 30.5%). Fig. 9 shows two examples of the model based on the ray-plane intersection with the dense point-cloud from the test set. On the right side, the detection in UTM coordinates is shown, on the left side the back-projection to image.

3.4. Detailed 3D Detection Performance

The evaluation of *MSE* for position, heading angle, and *MAPE* strongly depends on the matching criteria. In this study, an IOU_{3D} threshold of 0.5 was applied for all object classes, and additionally an IOU_{BEV} with the same threshold to compare to homography-based methods. The results, summarized in Table 5, demonstrate that ProjNet consistently provide more precise localization compared to the baseline homography-based approaches (0.28m vs 0.41m). This improvement underscores a significant advancement over state-of-the-art homography-based methods like [22] and [23] although numerical comparisons reveal mixed outcomes. Comparing the versions using sparse and dense point clouds as base for the ray-plane intersection (0.26m to 0.26m with matching based on IOU_{3D}) and the homography (0.41m to 0.40m) show similar performances, further underlining that sparse points clouds are sufficient. Comparing the results of ProjNet

Table 4
Detection results in mAP based on 3D IoU.

Model	Motorcycle	Car	Truck	Bus	Person	Bicycle	E-Scooter	Sum
	AP	AP	AP	AP	AP	AP	AP	mAP
ProjNet	32.6	46.0	20.6	18.7	30.0	36.6	29.3	30.5
ProjNet -sparse	35.0	43.0	17.7	19.1	31.5	35.0	23.2	29.2

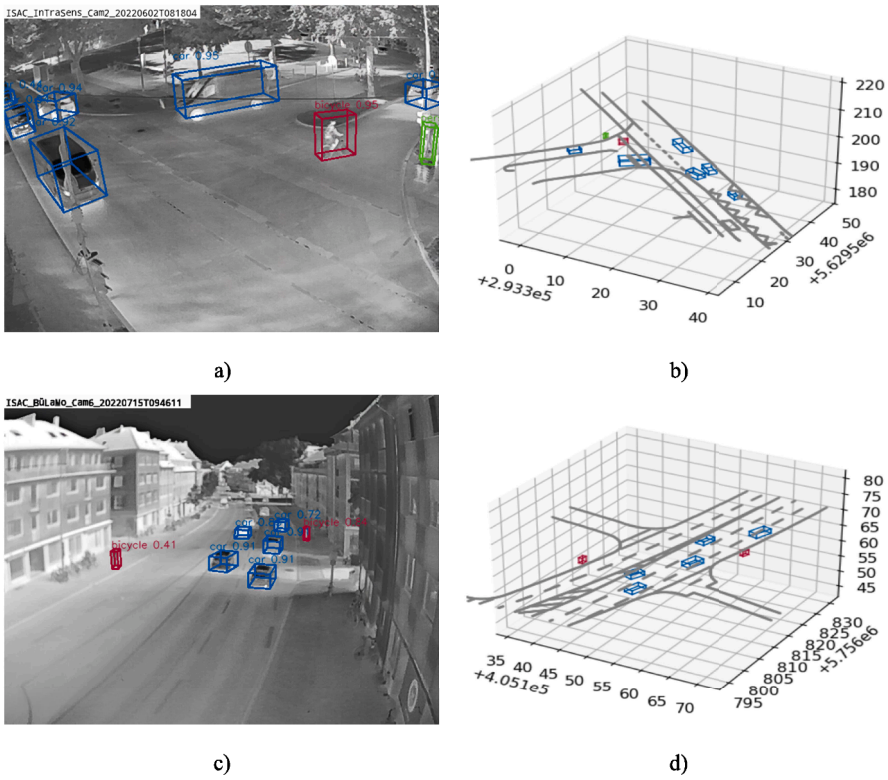


Fig. 9. Example detections from the test set. a) Showing the projections to the image plane b) The digital shadow of the intersection c) Projection of an inner city scene d) Corresponding digital shadow

Table 5
Comparison of mean squared position error for different classes and projection methods.

Model	Match-ing criteria	Motor-cycle MSE	Car MSE	Truck MSE	Bus MSE	Person MSE	Bicycle MSE	E-Scooter MSE	Sum MSE-all
3DNet [22]	N/A	/	/	/	/	/	/	/	0.13
Clausse [24]	N/A	/	/	/	/	/	/	/	0.61
ProjNet	IOU_{3D}	0.15	0.26	0.63	0.69	0.11	0.14	0.10	0.26
ProjNet - sparse	IOU_{3D}	0.13	0.27	0.66	0.72	0.11	0.14	0.10	0.26
ProjNet	IOU_{BEV}	0.16	0.29	0.74	0.77	0.12	0.14	0.11	0.28
ProjNet - sparse	IOU_{BEV}	0.15	0.30	0.75	0.74	0.12	0.14	0.11	0.29
Homo-graphy	IOU_{BEV}	0.32	0.39	0.88	0.93	0.12	0.18	0.14	0.41
Homo- graphy sparse	IOU_{BEV}	0.32	0.38	0.88	0.84	0.11	0.17	0.12	0.40

matched based on IOU_{3D} (0.26m) and IOU_{BEV} (0.28m) on sees small difference. The harder matching criteria IOU_{3D} lead to less matches but matches with higher similarity between detection and ground-truth and thus less distance.

In more detailed analysis, the proposed method outperforms [24] (0.26m vs 0.61m) but underperforms [22] (0.26m vs 0.13m), with these differences likely influenced by the matching criteria, which [22] and [24] do not specify. Furthermore, localization accuracy varies across object classes, with smaller objects such as Person (0.11m), Bicycle (0.14m), E-Scooter (0.10m), and Motorcycle (0.15m) achieving better results than larger objects such as Bus (0.69m) and Truck (0.63m). This disparity is likely due to the more straightforward localization of the bottom center point in pixel coordinates for smaller objects.

The Analysis of MSE trends across distance clusters (0 - 60m with 10m clusters size), as illustrated in Fig. 10 underlines the benefit of ProjNet's ray-plane intersection over homography based approaches for all clusters. Further it shows general trends like the increase in error with distance (e.g., from 0.19 at the second cluster to 0.33m in the last cluster for ProjNet with IOU_{3D}). The first cluster with objects potentially not fully visible in the image has no matches for the homography based methods and only few for ProjNet is excluded this trend. The observed trend is attributed to decreasing pixel resolution at greater distances, an effect that is logical and unavoidable.

The Heading-Angle and *MAPE* do not depend on the projection method but rather on the observation angle and object dimensions which means that homography based methods would lead to the same results. Therefore, only the ProjNet method with dense point cloud was evaluated. As shown in Table 6, in the *MSE* for heading angle, among the object classes, the Person class stands out with significantly poorer orientation estimation (1.83 in comparison to the second most 0.23), likely due to the lack of visible facial features in thermal imagery and in general limited features that indicate orientation of persons. Despite this, such errors are less impactful in practice as person exhibit minimal variation in width and depth. An overall heading angle accuracy of 0.1 radians (equivalent to 5.7°) is generally sufficient for practical applications. Compared to existing methods, the results are outperforming [24] (0.1 vs 0.14) but lag behind [22] (0.1 vs 0.015), the latter of which calculates orientation through tracking rather than direct estimation. Incorporating temporal context and tracking could improve the proposed method in the future.

For *MAPE*, shown in Table 7, most object classes achieve results within approximately 20%, with exceptions observed for E-Scooter

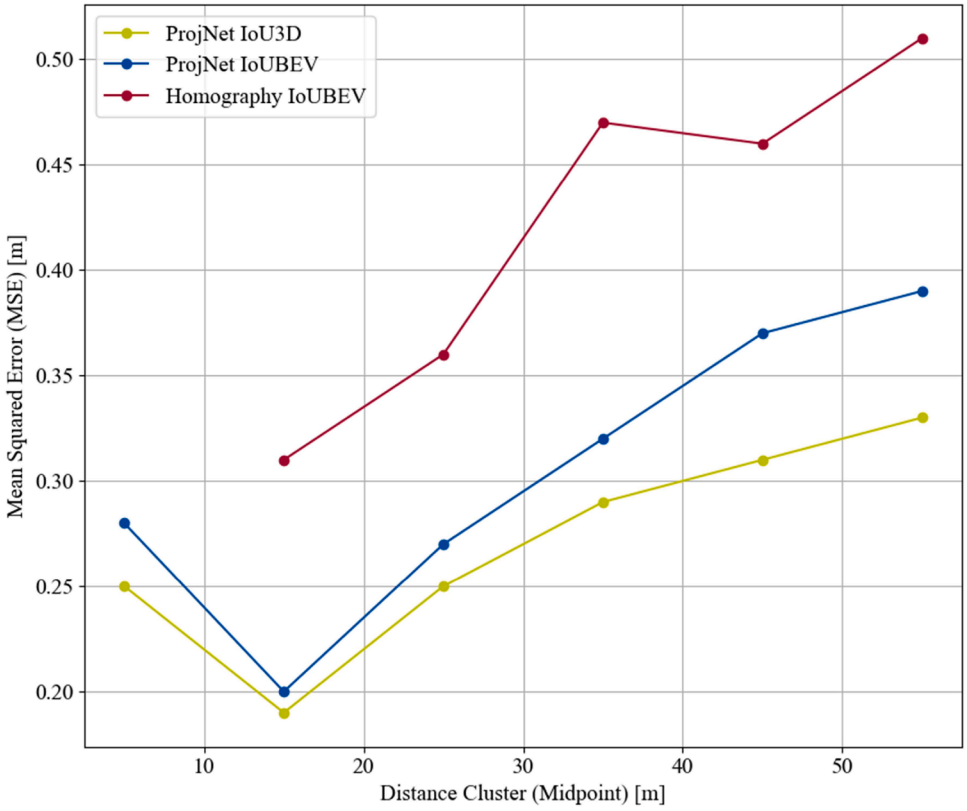


Fig. 10. Mean Squared Position Error illustrated for different distance clusters.

Table 6
Comparison of mean squared heading angle error for different classes.

Model	Motorcycle MSE	Car MSE	Truck MSE	Bus MSE	Person MSE	Bicycle MSE	E-Scooter MSE	Sum MSE-all
3DNet [22]	/	/	/	/	/	/	/	0.015
Clausse [24]	/	/	/	/	/	/	/	0.14
ProjNet	0.02	0.04	0.13	0.23	1.83	0.15	0.07	0.10

Table 7
Comparison of mean average percentage error of the object sizes for different classes.

Model	Motorcycle MAPE	Car MAPE	Truck MAPE	Bus MAPE	Person MAPE	Bicycle MAPE	E-Scooter MAPE	Sum MAPE -all
3DNet [22]	/	10.3-19.7	15.0-18.4	10.1-12.3	/	/	/	/
ProjNet	32.5	18.8	20.4	17.5	22.6	23.7	9.0	19.3

Table 8
Comparison of inference times for the different models

Model	Server Info	Inference time Edge device	Inference time Server
3DNET [22]	Intel Core i7 12900K with NVIDIA RTX 5000 GPU		0.0278 (~36 FPS)
CenterLoc [25]	Intel Core i7-8700 CPU and one GTX 1080Ti GPU	/	0.0243 (~41 FPS)
Mohammed et al [23]	Intel(R) Xeon(R) CPU E3-1240 v3 @ 3.40 GHz, 16.0 GB NVIDIA Quadro K620	/	0.0333 (~30 FPS)
ProjNet	NVIDIA Quadro RTX 5000 GPU and an Intel Xeon E5-2640 v3 CPU (2.60 GHz, 16 cores)	0.058 (~17 FPS)	0.0185s (~54 FPS)

(9.0%) and Motorcycle (32.5%) due to their limited representation in the test set, warranting cautious interpretation. The Car (18.8%) and Bus (17.5%) classes benefit from a relatively large number of training samples and standardized dimensions, facilitating more accurate estimations. In comparison to 3D-Net, the proposed method delivers similar performance (19.3% mean MAPE vs 10.1-19.7%), although differences are challenging to analyze due to 3D-Net's sparse evaluation only for specific object models and the absence of reported matching criteria [22].

This comparison underscores that thermal imagery, combined with sufficient training data, can achieve performance comparable to optical state-of-the-art methods. Overall, the findings highlight the robustness of the proposed approach while identifying areas for further enhancement, such as improving performance for larger objects and leveraging temporal tracking. Such a model based tracking method e.g., based on a kalman-filter could help to handle missed detections, allow to improve

the dimension estimation by calculation of mean values along tracks and estimating the heading angle based on the objects movement.

3.5. Inference times

A comparison of the inference times with SOTA methods based on the provided data of those works is difficult. Since [21] and [22] report only raw detection times without postprocessing, which means the projection and calculation of the world coordinates. Furthermore, table 7 contains the reported inference times measured on different hardware.

The measured inference times of ProjNet in Table 8 are with a batch size of 1, evaluated on an NVIDIA Jetson Xavier NX and a GPU server. For our method the time includes the whole process including the projection and 3D box calculation. On the server, our model reaches 54 frames per Second (FPS) exceeding the camera's frame rate (30 FPS) and therefore processing faster. It outperforms the reported state-of-the-art frame rates of 30-41 FPS significantly, but as mentioned above different hardware was used. Especially, [22] reported their results on a significantly less powerful GPU. Therefore, the exceed of the camera frame rate should predominate for the evaluation.

A noticeable performance drop is observed on the edge device, with the model processing 17 FPS. Despite this reduction, the framework remains capable of evaluating every second frame, which is sufficient for most applications, particularly in inner-city locations with lower traffic speeds. This performance highlights the framework's practicality while allowing for further optimization, such as faster non-maximum suppression (NMS), TensorRT acceleration, or leveraging newer generations of NVIDIA Jetson hardware.

3.6. Conceptual Comparison with State-of-the-Art Methods

To the best of our knowledge, this work is the first to address monocular 3D detection using thermal imagery from roadside cameras while also achieving fast performance on edge devices. Due to this novelty, no direct state-of-the-art comparison is possible.

However, there is existing work on monocular 3D vision from roadside cameras in RGB images. There are mainly two kinds of algorithms. Depth-estimation-based methods, such as those evaluated on, monocular 3D detection datasets, such as DAIR-V2X [13] or Rope3D [12], typically emphasize depth estimation as the most challenging aspect of monocular 3D detection. These approaches aim to detect objects in images while simultaneously estimating their depth, which is an error-prone task for roadside cameras [20].

This differs significantly from projection-based methods which use methods of perspective transformation to directly calculate the depth. This removes completely a source of error which makes a comparison to depth-estimation based methods obsolete. However, it is only feasible for objects touching the ground in calibrated cameras with known information on the surrounding world surface. Importantly, no publicly available datasets for monocular roadside cameras currently provide detailed road surface information, making this work and the provided dataset a potential new benchmark for the field.

While a direct numerical comparison to state-of-the-art methods using perspective projection in overall detection performance is not feasible, evaluation parameters, such as Mean Squared Error (*MSE*) for position and heading angle or Mean Absolute Percentage Error (*MAPE*) for size estimation, offer some insights. The presented results fall within a comparable range to the limited existing works that share similar constraints. However, these outcomes depend heavily on the matching criteria and dataset properties which are not reported for the existing methods. Nevertheless, the other projection based methods are all based on homography. Therefore, homography based methods were chosen as the baseline for numerical comparison. The presented experiments were able to demonstrate that the presented ray-plane intersection approach is more accurate than homography-based approaches.

Fig. 11. Near miss between right turning Car and Bicycle.

3.7. Evaluation in regard to practical application

A key evaluation criterion for monocular 3D traffic sensors is their practical applicability. In real-world scenarios, such methods are often relevant when the relative positioning or temporal relationships between objects are critical—for instance, to measure speed or analyze interactions between traffic participants. However, the reported accuracies allow such analysis and the fast inference time, in particular the fact that it is the first work to have demonstrated this for edge devices, underlines this practical usability.

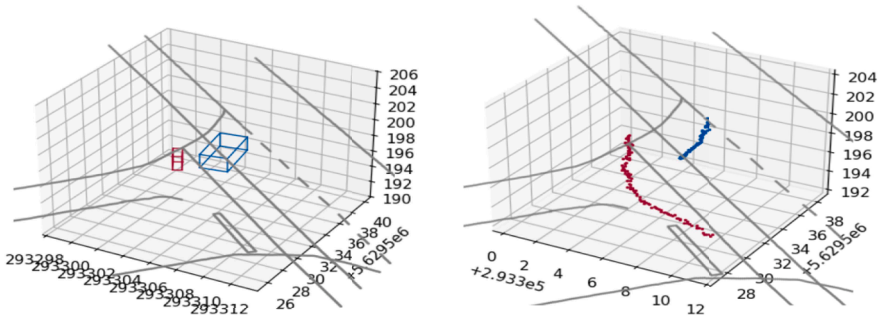
Our dataset does not yet include extreme weather conditions such as fog and snow. However, as shown in Fig. 2 and Table 1, it covers various real-world conditions, including sunny, rainy, and wet road surfaces, as well as urban and rural environments with different traffic densities. Nevertheless, it is notable that extreme weather is not yet included in the evaluation. Additionally, a check in additional traffic situations would underline the robustness of our method. Nevertheless it is notable that only the object classification, keypoint detection and 2D-bounding box detection relies on such external factors. In contrast to depth estimation, the used projection based method ensures an independent robust depth calculation.

Currently, our method is optimized for a detection range of approximately 50 meters per camera (as seen in Fig. 10 positioning error increases afterwards, making it well-suited for analyzing interactions and traffic behavior at smaller intersections or high-risk locations within larger intersections. Multi-sensor fusion could enhance the field of view, enabling broader scene coverage. Additionally, integrating LiDAR or other complementary sensors could improve detection accuracy and robustness, making the system more applicable to autonomous driving, where higher precision and redundancy are essential.

It is also important that detection alone is not sufficient for a comprehensive analysis, as it only provides individual object instances without continuity. Matching and tracking are essential to obtain complete trajectories, enabling a more accurate representation of object movements. By integrating filtering and fitting algorithms, the trajectory quality can be further refined, reducing noise and improving consistency. This combination allows for more reliable interaction analysis, particularly in dynamic traffic environments.

3.8. Future application

The proposed methodology enhances the capabilities of conventional traffic detectors by providing precise data on object positions, orientations, and dimensions beyond pixel coordinates. This enables accurate speed measurement and advanced spatial analysis for example



a) b)

Fig. 12. Digital shadow of the near miss. a) The near miss in world coordinates showing the true distances and dimensions b) The car's and bicycle's track showing the evasive action of the bicycle

for in-depth evaluations of accidents and near-miss incidents. Such analysis aims to identify patterns in near-misses to proactively prevent accidents.

Human-interpretable video snippets are critical for this analysis, emphasizing the advantages of the proposed method over radar and LiDAR systems. Detailed information on distances, speeds, and positions is essential for identifying and understanding near-miss scenarios. For instance, Fig. 11 illustrates a near miss between a right-turning vehicle and a cyclist, showing the system's capability to capture such events.

Fig. 12 highlights the advantages of the proposed approach. Without distortion from perspective effects, the absolute distance between two objects (left image) and the entire trajectory, including evasive actions, can be analyzed (right image). This provides significantly richer insights compared to traditional pixel-based analysis methods.

4. Conclusion

This work introduces a novel approach to 3D traffic monitoring using thermal imagery, addressing key challenges in real-time traffic monitoring. The key contributions and outcomes of this work are as follows:

- A ray-tracing-based approach to map pixel and world coordinates increasing the mAP in BEV by 25 percent points compared to homography-based methods used by prior monocular 3D frameworks.
- Real-time capable monocular 3D vision on edge devices. The proposed Yolov7-tiny based method achieves 54 FPS on GPU server and 17 FPS on Nvidia Jetson Xavier NX making it significantly more efficient than existing methods.
- A method for privacy-compliant 3D traffic monitoring that, unlike RGB-based methods, stays highly effective in diverse weather and lighting conditions
- A novel thermal 3D roadside dataset providing 9,591 annotated images including 3D world coordinates, camera calibration data and surface models filling a critical gap in traffic monitoring research.
- The proposed system provides a basis for accurate trajectory measurement and near-miss detection, making it well-suited for traffic monitoring to analyze road user behavior and finding patterns in near-misses for preventive safety research.

Future work will focus on enhancing the generalizability and robustness of the proposed method. The dataset will be expanded to include adverse weather conditions such as fog, snow, and heavy rain, as well as high-density traffic scenarios to better reflect real-world complexities. Additional robustness tests in varied environments will help validate the model's performance under challenging conditions. Improving trajectory accuracy through advanced tracking will further support applications in traffic flow analysis, accident prevention, and conflict detection. Integrating multi-sensor fusion with LiDAR, radar, or RGB cameras could enhance detection accuracy and multi-camera fusion could enhance the field of view. Automated techniques for generating surface models and further optimizing detection speed will be evaluated. These improvements will strengthen the framework's real-time, privacy-compliant 3D traffic monitoring role.

CRediT authorship contribution statement

Arnd Pettirsch: Writing – original draft, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Alvaro Garcia-Hernandez:** Writing – review & editing, Validation, Supervision, Resources, Project administration.

Declaration of competing interest

The authors declare no conflicts of interest related to this study. This research was conducted independently, and no financial or personal relationships with other people or organizations have influenced the work reported in this paper. The dataset and code developed for this study are publicly accessible to ensure transparency and reproducibility.

Acknowledgments

The study was part of the research project 'SmarteAmpel' founded by the German Federal Ministry of Economic Affairs and Climate Action (BMWK). Data was further collected in the project 'BueLaMo – Muensterland' founded by the German Federal Ministry of Education and Research (BMBF). and the contract research 'Untersuchung von Gewöhnungseffekten beim Einsatz von fluoreszierenden Materialien' for the German Federal Highway Research Institute and the project 'IntraSens' supported by RWTH Aachen University.

Data availability

Data will be made available here: https://doi.org/10.17632/tw6ghtv624.1.

References

[1] Y. Modi, R. Teli, A. Mehta, K. Shah and M. Shah, "A comprehensive review on intelligent traffic management using machine learning algorithms," vol. 7, no. 128 (2022), December 2021. https://doi.org/10.1007/s41062-021-00718-3.

[2] A. Laureshyn, C. Johnsson, T.K.O. Madsen, A. Várhelyi, M. de Goede, Å. Svensson, N. Saunier, W. van Haperen, Exploration of a method to validate surrogate safety measures with a focus on vulnerable road users, in: Proceedings of the Road Safety & Simulation International Conference, 17-19 October 2017, 2017.

[3] M. Kaliske, R. Behnke, I. Wollny, Vision on a Digital Twin of the Road-Tire-Vehicle System for Future Mobility, Tire Science and Technology 49 (2021) 2–18, https://doi.org/10.2346/tire.21.190223. January.

[4] Q. Huang, K. Zhu, K. Wu, W. Hua, Y. Zhu, Multi-sensor Fusion for Perception in Complex Traffic Environments. Communication, Computation and Perception Technologies for Internet of Vehicles, Springer Nature Singapore, 2023, pp. 147–161, https://doi.org/10.1007/978-981-99-5439-1.
[5] P.P. Tasgaonkar, R.D. Garg, P.K. Garg, Vehicle Detection and Traffic Estimation with Sensors Technologies for Intelligent Transportation Systems, Sensing and Imaging 21 (2020), https://doi.org/10.1007/s11220-020-00295-2. June.
[6] N. Casado-Sanz, B. Guirao, A. Lara Galera, M. Attard, Investigating the Risk Factors Associated with the Severity of the Pedestrians Injured on Spanish Crosstown Roads, Sustainability 11 (2019) 5194, https://doi.org/10.3390/su11195194.
[7] T. Alldieck, C. Bahnsen, T. Moeslund, Context-Aware Fusion of RGB and Thermal Imagery for Traffic Monitoring, Sensors 16 (2016) 1947, https://doi.org/10.3390/s16111947. November.
[8] R.B. Langley, The UTM grid system, GPS world 9 (1998) 46–50.
[9] Z. Liu, Z. Chen, X. Wei, W. Chen, Y. Wang, External Extrinsic Calibration of Multi-Modal Imaging Sensors: A Review, IEEE Access 11 (2023) 110417–110441, https://doi.org/10.1109/ACCESS.2023.3322229.
[10] A. Mertan, D.J. Duff, G. Unal, Single image depth estimation: An overview, Digital Signal Processing 123 (2022) 103441, https://doi.org/10.1016/j.dsp.2022.103441. April.
[11] M. Ahmed, K.A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, M.Z. Afzal, Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments, Sensors 21 (2021) 5116, https://doi.org/10.3390/s21155116. July.
[12] X. Ye, M. Shu, H. Li, Y. Shi, Y. Li, G. Wang, X. Tan, E. Ding, Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, https://doi.org/10.1109/CVPR52688.2022.02065.
[13] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, Z. Nie, DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, https://doi.org/10.1109/CVPR52688.2022.02067.
[14] W. Zimmer, C. Creß, H. T. Nguyen and A. C. Knoll, "A9 Intersection Dataset: All You Need for Urban 3D Camera-LiDAR Roadside Perception," arXiv preprint arXiv: 2306.09266, June 2023. https://doi.org/10.48550/arXiv.2306.09266.
[15] W. Zimmer, J. Birkner, M. Brucker, H.T. Nguyen, S. Petrovski, B. Wang, A.C. Knoll, InfraDet3D: Multi-Modal 3D Object Detection based on Roadside Infrastructure Camera and LiDAR Sensors, IEEE Intelligent Vehicles Symposium (IV) (2023) 1–8, https://doi.org/10.1109/IV55152.2023.10186723. April.
[16] C. Guindel, D. Martín, J.M. Armingol, Traffic scene awareness for intelligent vehicles using ConvNets and stereo vision, Robotics and Autonomous Systems 112 (2019) 109–122, https://doi.org/10.1016/j.robot.2018.11.010. February.
[17] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, K.Q. Weinberger, Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8445–8453, https://doi.org/10.1109/CVPR.2019.00864.
[18] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, T. Chateau, Deep MANTA: A Coarse-to-Fine Many-Task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, https://doi.org/10.1109/CVPR.2017.198.
[19] L. Yang, K. Yu, T. Tang, J. Li, K. Yuan, L. Wang, X. Zhang, P. Chen, BEVHeight: A Robust Framework for Vision-based Roadside 3D Object Detection, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 21611–21620, https://doi.org/10.1109/CVPR52729.2023.02070.
[20] L. Yang, T. Tang, J. Li, P. Chen, K. Yuan, L. Wang, Y. Huang, X. Zhang and K. Yu, "BEVHeight++: Toward Robust Visual Centric 3D Object Detection," *arXiv preprint arXiv:2309.16179*, September 2023. https://doi.org/10.48550/arXiv.2309.16179.
[21] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuScenes: A multimodal dataset for autonomous driving, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, https://doi.org/10.1109/CVPR42600.2020.01164.
[22] M. Rezaei, M. Azarmi, F.M.P. Mir, 3D-Net: Monocular 3D object recognition for traffic monitoring, Expert Systems with Applications 227 (2023) 120253, https://doi.org/10.1016/j.eswa.2023.120253. October.
[23] A. Mohamed, M.M. Ahmed, L. Li, Automated Traffic Safety Assessment Tool Utilizing Monocular 3-D Convolutional Neural Network-Based Detection Algorithm at Signalized Intersections, in: I*nternational Conference on Transportation and Development 2024*, 2023, pp. 456–467, https://doi.org/10.1061/9780784485514.040.
[24] A. Clausse, S. Benslimane, A. de La Fortelle, Large-Scale extraction of accurate vehicle trajectories for driving behavior learning, in: 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 2391–2396, https://doi.org/10.1109/IVS.2019.8814095.
[25] X. Tang, W. Wang, H. Song, C. Zhao, CenterLoc3D: monocular 3D vehicle localization network for roadside surveillance cameras, Complex & Intelligent Systems 9 (2023) 4349–4368, https://doi.org/10.1007/s40747-022-00962-9. January.
[26] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2 ed., Cambridge University Press, 2004 https://doi.org/10.1017/CBO9780511811685.
[27] J. Zeng, R. Butler, J.J. van den Dobbelsteen, B.H.W. Hendriks, M. Van der Elst, J. Dauwels, Automatic Camera Pose Estimation by Key-Point Matching of Reference Objects, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5, https://doi.org/10.1109/ICASSP49357.2023.10095197.
[28] N. Geoportal, Geoportal NRW, 2024 [Online]. Available, https://www.geoportal.nrw/?activetab=portal [Accessed 23 12 2024].
[29] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on pattern analysis and machine intelligence 22 (2000) 1330–1334, https://doi.org/10.1109/34.888718.
[30] D. Maji, S. Nagori, M. Mathew, D. Poddar, Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2637–2646, https://doi.org/10.1109/CVPRW56347.2022.00297.
[31] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 7464–7475, https://doi.org/10.1109/CVPR52729.2023.00721. July.
[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.
[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520, https://doi.org/10.1109/CVPR.2018.00474.
[34] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, 2019, pp. 6105–6114, https://doi.org/10.48550/arXiv.1905.11946.
[35] T.M.N.B.T. Rashid, L.M. Fadzil, Comparative Review of Object Detection Algorithms in Small Single-Board Computers, International Journal on Recent and Innovation Trends in Computing and Communication 11 (2023) 244–252.
[36] O. Doll, A. Loos, Comparison of object detection algorithms for livestock monitoring of sheep in UAV images, Int. Workshop Camera traps, AI, and Ecology (2023), https://doi.org/10.24406/publica-2164.
[37] Ultralytics, YOLOv8: State-of-the-art Object Detection and Segmentation Models, 2023 [Online]. Available, https://docs.ultralytics.com/models/yolov8/#overview [Accessed 2025-02-15].
[38] Z. Liu, Z. Wu, R. Tóth, Smoke: Single-stage monocular 3d object detection via keypoint estimation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 996–997, https://doi.org/10.1109/CVPRW50498.2020.00506.
[39] Axis Communications AB., "Product Information Axis Q1952-E Thermal Camera," [Online]. Available: https://www.axis.com/de-de/products/axis-q1952-e. [Accessed 24 06 2024].
[40] Axis Communication AB., "Product Information Axis Q1942-E Thermal Network Camera," [Online]. Available: https://www.axis.com/de-de/products/axis-q1942-e. [Accessed 24 06 2024].
[41] E.P. Baltsavias, Airborne laser scanning: existing systems and firms and other resources, ISPRS Journal of Photogrammetry and Remote sensing 54 (1999) 164–198, https://doi.org/10.1016/S0924-2716(99)00016-7.
[42] Y. Ito, Delaunay Triangulation. Encyclopedia of Applied and Computational Mathematics, Springer Berlin Heidelberg, 2015, pp. 332–334, https://doi.org/10.1007/978-3-540-70529-1_314.
[43] Open3D-Team, open3d.geometry.TriangleMesh — Open3D 0.17.0 documentation, 2024 [Online]. Available: https://www.open3d.org/html/python_api/open3d.geometry.TriangleMesh.html [Accessed 2024-12-06].
[44] G. Sithole, G. Vosselman, Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 59 (2004) 85–101, https://doi.org/10.1016/j.isprsjprs.2004.05.004.
[45] W.-C. Hung, H. Kretzschmar, V. Casser, J.-J. Hwang, D. Anguelov, LET-3D-AP: Longitudinal Error Tolerant 3D Average Precision for Camera-Only 3D Detection, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, p. 8272, https://doi.org/10.1109/ICRA57147.2024.10609986. -7279.
[46] C.H. Bahnsen, T.B. Moeslund, Rain Removal in Traffic Surveillance: Does it Matter? IEEE Transactions on Intelligent Transportation Systems 20 (2019) 2802–2819, https://doi.org/10.1109/TITS.2018.2872502. August.
[47] T. Balon, M. Knapik, B. Cyganek, Real-Time Detection of Small Objects in Automotive Thermal Images with Modern Deep Neural Architectures, ACSIS 37 (2023) 29–35, https://doi.org/10.15439/2023F8409.
[48] E. Gebhardt, M. Wolf, Camel dataset for visual and thermal infrared multiple object detection and tracking, in: 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS), 2018, pp. 1–6, https://doi.org/10.1109/AVSS.2018.8639094.
[49] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, Z. Luo, Target-Aware Dual Adversarial Learning and a Multi-Scenario Multi-Modality Benchmark To Fuse Infrared and Visible for Object Detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 5802–5811, https://doi.org/10.1109/CVPR52688.2022.00571.