Complex & Intelligent Systems (2023) 9:7171–7195
https://doi.org/10.1007/s40747-023-01117-0

ORIGINAL ARTICLE

Turning traffic surveillance cameras into intelligent sensors for traffic density estimation

Zijian Hu[1] · William H. K. Lam[1] · S. C. Wong[2] · Andy H. F. Chow[3] · Wei Ma[1,4]

Received: 3 February 2023 / Accepted: 8 May 2023 / Published online: 22 June 2023

Abstract

Accurate traffic density plays a pivotal role in the Intelligent Transportation Systems (ITS). The current practice to obtain the traffic density is through specialized sensors. However, those sensors are placed in limited locations due to the cost of installation and maintenance. In most metropolitan areas, traffic surveillance cameras are widespread in road networks, and they are the potential data sources for estimating traffic density in the whole city. Unfortunately, such an application is challenging since surveillance cameras are affected by the **4L** characteristics: **L**ow frame rate, **L**ow resolution, **L**ack of annotated data, and **L**ocated in complex road environments. To the best of our knowledge, there is a lack of holistic frameworks for estimating traffic density from traffic surveillance camera data with 4 L characteristics. Therefore, we propose a framework for estimating traffic density using uncalibrated traffic surveillance cameras. The proposed framework consists of two major components: camera calibration and vehicle detection. The camera calibration method estimates the actual length between pixels in the images and videos, and the vehicle counts are extracted from the deep-learning-based vehicle detection method. Combining the two components, high-granular traffic density can be estimated. To validate the proposed framework, two case studies were conducted in Hong Kong and Sacramento. The results show that the Mean Absolute Error (MAE) for the estimated traffic density is 9.04 veh/km/lane in Hong Kong and 7.03 veh/km/lane in Sacramento. The research outcomes can provide accurate traffic density without installing additional sensors.

Keywords Traffic surveillance camera · Camera calibration · Vehicle detection · Traffic density estimation

Introduction

Accurate and real-time traffic density is the essential input to the Intelligent Transportation Systems (ITS) with various traffic operation and management tasks [1, 2]. Many cities have expended considerable efforts in installing traffic detectors to obtain traffic density and other traffic-related information in recent years. However many ITS applications are still data-hungry. Using Hong Kong as an example, the current detectors (*e.g.,* loop detectors) only cover approximately 10% of the road segments, which is not sufficient to support the network-wide traffic modeling and management framework. How to collect the real-time traffic density in an accurate, efficient, and cost-effective manner presents a long-standing challenge for not only the research community but

✉ Wei Ma
wei.w.ma@polyu.edu.hk

Zijian Hu
zijian.hu@connect.polyu.hk

William H. K. Lam
william.lam@polyu.edu.hk

S. C. Wong
hhecwsc@hku.hk

Andy H. F. Chow
andychow@cityu.edu.hk

[1] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom , Kowloon, Hong Kong SAR, China

[2] Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong SAR, China

[3] Department of Systems Engineering, City University of Hong Kong, Kowloon Tong, Kowloon, Hong Kong SAR, China

[4] Research Institute for Sustainable Urban Development, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China

also the private sector (*e.g.,* Google Maps) and the public agency (*e.g.,* Transport Department).

Various sensors and devices can be employed to estimate the traffic density directly or indirectly on urban roads. A review of existing studies on traffic density estimation is shown in Table. 1. Point sensors (e.g., inductive-loop detectors, pneumatic tubes, radio-frequency identification (RFID), *etc.*) are widely used for traffic density estimation [3], and they are robust to environment changes (*e.g.,* weather, light) for stable 24/7 estimation. Some advanced techniques such as Vehicular Ad hoc Network (VANET) [4] and Unmanned Aerial Vehicle (UAV) [5], can complement point sensors and contribute to traffic density estimation. However, the uniform challenge for these sensing technologies is that they may not suitable for traffic density estimation in the entire urban network due to the deployment and maintenance cost of sensors.

Traffic surveillance cameras are an essential part of an urban traffic surveillance system. Cameras are often used for visual inspection of traffic conditions and detection of traffic accidents by traffic engineers sitting in the Traffic Management Centers (TMCs). Such cameras are widely distributed in most metropolises, making it possible for large-scale traffic density estimation.

For example, in California, approximately 1,300 cameras are set up by Caltrans to monitor the traffic conditions on highways [6]; in Seoul, the TOPIS[1] system functions on 834 surveillance cameras; and in Hong Kong, the Transport Department uses about 900 surveillance cameras in its eMobility System.[2] With various camera-based traffic surveillance systems deployed globally, there is great potential to extract traffic information from camera images and videos. Combined with recent advanced technologies, several attempts have been made to vehicle information extraction (*speed and count*) [7, 8], vehicle re-identification [9, 10] and pedestrian detection [11, 12]. Furthermore, it is in great need to make use of the massive traffic surveillance camera data for traffic density estimation.

To look into the density estimation problem, we note that the traffic density k is computed as the number of vehicles N per lane divided by the length of a road L [13], as presented in Eq. (1).

$$k = \frac{N}{L} \tag{1}$$

Note that in several studies [14–18], the road length is assumed to be fixed and known. Hence estimating the traffic density is equivalent to counting the number of vehicles on the road. However, such an assumption has been relaxed in this study, as the road lengths in camera images are also unknown to us in different surveillance systems. Therefore, based on Eq. (1), the traffic density estimation from surveillance cameras can be decomposed into two sub-problems:

- **Camera calibration:** aims to estimate the road length L from camera images, in which the core problem is to measure the distance between the real-world coordinates corresponding to the image pixels.
- **Vehicle detection:** focuses on counting the vehicle number N, and it can be formulated as the object detection problem.

Both problems are separately discussed in the research field of Computer Vision (CV) [19, 20]. However, the challenges of traffic density estimation from surveillance cameras are unique.

The data collected from traffic surveillance cameras appeal to the **4 L** characteristics. Firstly, due to personal privacy concerns and network bandwidth limits, the camera images are usually in **L**ow resolution and **L**ow frame rate. For example, in Hong Kong, the resolution of the monitoring image is 320 × 240 pixels, and all images are updated every two minutes [15]. Secondly, it is onerous to annotate detailed information for each camera, and hence most of the collected data are **L**acking in annotations. Thirdly, surveillance cameras distributed across urban areas are often **L**ocated in complex road environments, where the roads are not simply straight segments (*e.g.*, curved roads, mountain roads and intersections). Overall, we summarize the challenges of the traffic density estimation using the surveillance cameras as **4 L**, which represents: **L**ow resolution, **L**ow frame rate, **L**ack of annotated data and **L**ocated in complex road environments.

The 4 L characteristics present great challenges to both camera calibration and vehicle detection problems. There is a lack of holistic frameworks to comprehensively address the 4 L characteristics for traffic density estimation using surveillance cameras. To further highlight the contributions of this paper, we first review the existing literature on both camera calibration and vehicle detection.

Literature review on camera calibration. Camera calibration aims to match invariant patterns (*i.e.,* key points) to acquire a quantitative relationship between the points on images and in the real world. Under the 4 L characteristics, conventional camera calibration faces multi-fold challenges: (1) The endogenous camera parameters (*e.g.,* focal length) can be different for each camera and are generally unknown. (2) Recognizing the brands and models of vehicles from low-resolution images is challenging, making it difficult to correctly match key points based on car model information; (3) Continuous tracking a single vehicle from low frame rate images is nearly impossible, which makes some of the existing algorithms inapplicable. (4) The invariant patterns

[1] Seoul Transport Operation & Information Service.

[2] https://www.hkemobility.gov.hk/en/traffic-information/live/cctv.

Table 1 A review of emerging sensing technologies for estimating traffic density

Sensors	Advantages	Disadvantages
Point sensors [3]	1. Steady data sources for 24/7 monitoring	1. Expensive and difficult for massive installation and maintenance
VANET [4]	1. No additional hardware required	1. Limited accuracy when the penetration rate of PVs islow
	2. Potential data sources covered a large-scale traffic network	2. Rare pilot studies have been conducted
UAV [5]	1. High flexibility and instant deployment	1. Challenging to long-time estimation with large perspective
	2. High-fidelity data sources	2. Expensive for massive deployment
Traffic surveillance cameras	1. Widespread in many cities	1. Low data quality leading to potentially inaccurate results
This paper	2. Steady data sources for 24/7 monitoring	2. Owing to privacy concerns, sometimes only images (and not videos) can be acquired

in images are challenging to locate. This difficulty is caused by both the locations of the surveillance cameras (usually at the top of buildings or bridges to afford a wide visual perspective for visual monitoring traffic conditions) and the low image resolution. Even a one-pixel shift of the annotation errors (errors when annotating the key points) will result in a deviation of tens of centimeters in the real world, which fails the camera calibration (Impact of annotation errors on calibration algorithms can be referred in Appendix. A). (5) Existing camera calibration algorithms assume straight road segments, but many surveillance cameras locate at more complex road environments (*e.g.,* curved roads, mountain roads, intersections), making the existing algorithms not applicable.

Existing camera calibration methods only solve a subset of the aforementioned challenges. In the traditional calibration paradigm, a checkboard with a certain grid length is manually placed under the cameras [19], and key points can be selected as the intersections of the grid. However, it is time- and labor-consuming to simultaneously calibrate all cameras in the entire surveillance system. A common method for traffic camera calibration without the need of special equipment is to estimate the camera parameters using the vanishing point method, which leverages the perspective effect. The key points can be selected either as road markings [21] or common patterns on vehicles on roads [22–24]. These works assume that both sides of the road are parallel straight lines or all vehicles drive in the same direction. However, this assumption is invalid for complex road environments, such as curved roads and intersections, where vehicles drive in multiple directions. Hence, it is difficult to generalize the method to all camera scenarios in different traffic surveillance systems. Another alternative method is the Perspective-n-Point (PnP) method, which does not rely on vanishing points, but estimates the camera orientation given n three-dimensional points and their projected two-dimensional points (Normally $n \geq 3$) in the image. Several algorithms have been proposed to solve the PnP problem [25–29], and they have been validated as feasible and efficient methods of traffic camera calibration using monitoring videos [30]. However, the PnP method requires prior knowledge of the camera focal length, which is unknown for many surveillance cameras in real-world applications. The PnP method can be further extended to the PnPf method, which considers the focal length as an unknown variable during the calibration [31–34], but it has rarely been successfully applied to the traffic surveillance camera in practice. An important reason is that PnPf is normally sensitive to annotation errors which can lead to a completely false solution. Because the images from traffic surveillance cameras are in low resolution, the PnPf method may not be applicable. Additionally, a recently reported method [35] calibrates the camera in complex road environments without knowing the focal length, but it requires that the key points are on a specific vehicle model *e.g.,* Tesla Model S, which is impractical for low-resolution and low-frame-rate cameras.

In summary, existing camera calibration methods may not be suitable under 4 L characteristics. The main reason is that the key points on the single vehicle cannot provide enough information for the calibration due to the 4 L characteristics. In contrast, if multiple key points on multi-vehicles are considered in the camera calibration method, the calibration results could be made more stable and robust. However, this is still an open problem for the research community.

Literature review on vehicle detection. For vehicle detection, current solutions leverage machine-learning-based models to detect vehicles from camera images, while many

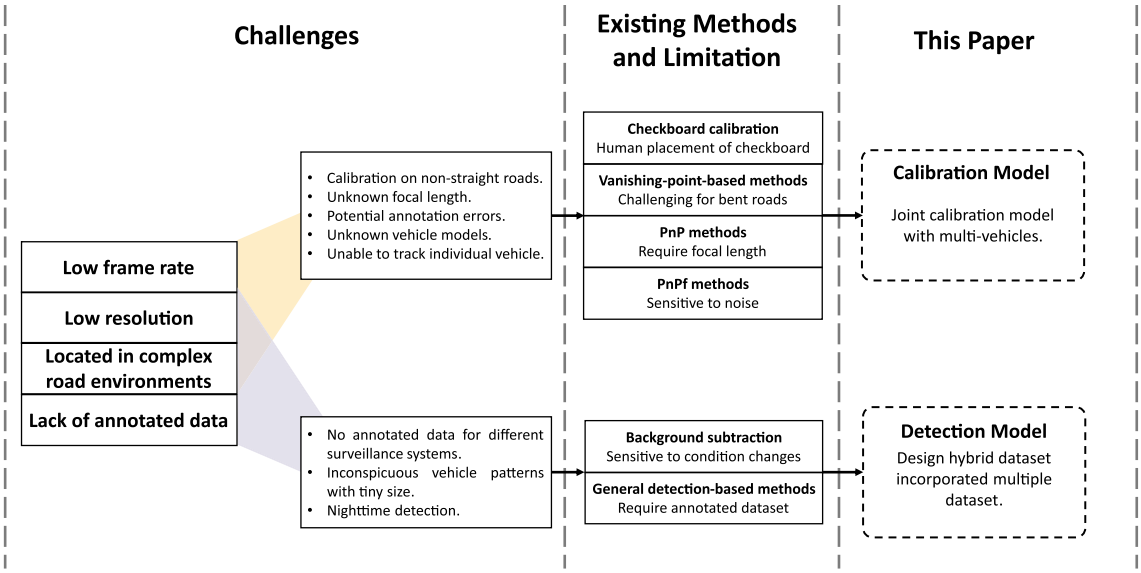


Fig. 1 Challenges and limitations of existing methods for traffic camera calibration and vehicle detection

challenges still remain: (1) The machine-learning models heavily rely on the annotated images for supervised training, and the labeled images are generally not available for each traffic surveillance system. (2) Vehicles only occupy several pixels in images due to the low resolution of images, making them difficult to be detected by the machine learning models; (3) during nighttime, the illumination conditions may hinder the detection of vehicles, presenting a challenge to 24/7 traffic density estimation.

Vehicle detection from surveillance cameras has been extensively studied for many years. Background subtraction was initially considered an efficient algorithm to extract vehicles from the background [14, 15, 18]. The underlying assumption in background subtraction is that the background of multiple images is static, and can therefore be obtained by averaging multiple images. However, this assumption may be improper when the illumination intensity of different images varies significantly, such as at night or on windy days. Recent studies have focused on detection-based algorithms since they are more resistant to background changes. General object detection frameworks can be used to detect vehicles from images [20, 36–38], while as they are not tailored for vehicle detection, the performance is not satisfactory. In the transportation community, work [39] applied a convolutional neural network (CNN) for vehicle detection in low-resolution traffic videos; [17] combined two classical detection frameworks for accuracy consideration, and [40] extended to automatically segment the region of interest (ROI) based on optical flow. Recently, [16] generated a weighted mask to compensate for size variance caused by the perspective effect. They subsequently combined a CNN with Long-Short-Term Memory (LSTM) to exploit spatial and temporal information from videos [41]. Shen et al. [42] took advantage of the K-means GIoU algorithm and then added a detection branch for fast convergence and small object detection. However, the performance of existing detection models degrade drastically when annotated data are lacking. Though the few-shot learning [43] may be incorporated to compensate for transferring the model into new scenes without or with a few annotated data, the unified performances under different camera conditions during daytime and nighttime cannot be guaranteed. To develop a generalized vehicle detection model in various surveillance systems, we augment the training data by incorporating traffic-related datasets in a systematic way.

Overall, the challenges to traffic density estimation under 4 L characteristics are summarized in Fig. 1.

For road length estimation, we aim to calibrate the surveillance camera with an unknown focal length using low-quality image slices obtained under complex conditions. For vehicle number estimation, we focus on developing a training strategy that is robust for low-resolution images acquired during both daytime and nighttime without annotating extra images.

This paper proposes a holistic framework that turns traffic surveillance cameras into intelligent sensors for traffic density estimation. The proposed framework mainly consists of two components: (1) camera calibration and (2) vehicle detection. For camera calibration, a novel method of multi-vehicle camera calibration (denoted as `MVCalib`) is developed to utilize the key point information of multiple vehicles simultaneously. The actual road length can be estimated from the pixel distance in images once the camera is calibrated. For vehicle detection, we develop a linear-program-based approach to hybridize various public vehicle

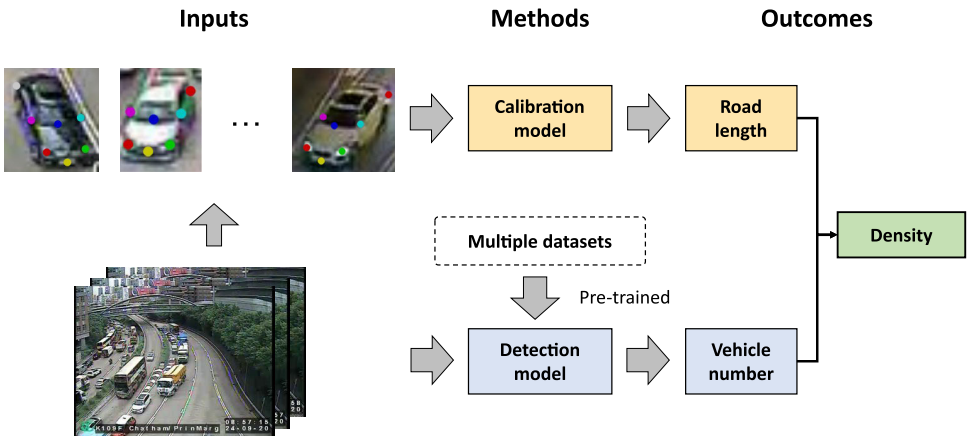


Fig. 2 Framework of traffic density estimation with surveillance cameras

datasets to balance the images during daytime and nighttime under various conditions, and these public datasets are originally used for different purposes. A deep learning network (YOLO-v5) is trained on the proposed hybrid dataset. The trained network can achieve decent detection accuracy during both daytime and nighttime in various surveillance camera systems without extra training on those surveillance cameras, which exempts additional effort from annotating labels on images.

Two case studies with ground truth have been conducted to evaluate the performance of the proposed framework. Results show that the estimation accuracy for the road length is more than 95%. Vehicle detection can reach an accuracy of 88% during daytime and nighttime, under low-quality camera images.

To summarize, the major contributions of this paper are as follows:

- It provides a holistic framework for 24/7 traffic density estimation using traffic surveillance cameras with **4L** characteristics: **L**ow frame rate, **L**ow resolution, **L**ack of annotated data, and **L**ocated in complex road environments.
- It first time develops a robust multi-vehicle camera calibration method `MVCalib` that collectively utilizes the spatial relationships among key points from multiple vehicles. The proposed method can be used to calibrate surveillance cameras under the 4**L** characteristics.
- It systematically designs a linear-program-based data mixing strategy to synergize image datasets from different cameras and to balance the performance of the deep-learning-based vehicle detection models under different traffic scenarios.
- It validates the proposed framework in two traffic surveillance camera systems in Hong Kong and Sacramento, and the research outcomes create portals for rapid and massive deployment of the proposed framework in different cities.

Methods

In this section, we first introduce the overall framework, and the camera calibration model and vehicle detection model are then elaborated separately.

The overall framework

The framework of the traffic density estimation model is shown in Fig. 2.

Camera images are first collected from public traffic surveillance camera systems, and then key points on vehicles are annotated. The camera calibration model uses the annotated data to derive a relationship between points on images and in the real world. If we can acquire the skeleton of the road, the road length can be further computed after calibration. For vehicle detection, the camera image data are fed to a deep-learning-based vehicle detection model pre-trained on a hybridized dataset, which is used to count vehicles on the road. Combining the road length and vehicle number information, we can estimate the high-granular traffic density information on the road.

Camera calibration

In this section, we present the proposed camera calibration method, `MVCalib`. The background about camera calibration is first reviewed, then the detailed information about the proposed camera calibration model will be elaborated subsequently.

Overview of camera calibration problems

A simplified pinhole camera model is widely used to illustrate the relationship between three-dimensional objects in the real world and the projected two-dimensional points on the camera images. Given the location of a certain point in the real world $[X, Y, Z]^T \in \mathbb{R}^3$, the projected point on the

camera image can be represented as $[u, v]^T \in \mathbb{R}^2$. The relationship between $[X, Y, Z]^T$ and $[u, v]^T$ is defined in Eq. (2) and (3).

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

$$sp_i = K\,[R|T]\,P_i \tag{3}$$

where Eq. (3) is the vectorized version of Eq. (2). $K = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}$ encodes the endogenous camera parameters, where f denotes the focal length of the camera. w and h represent the width and height of images. $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ and $T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$ are the rotation matrix and translation vector of the camera, respectively. Hence, $[f, R, T] \in \mathbb{R}^{13}$ are the 13 parameters to be estimated in the problem of camera calibration. Once the camera parameters f, R, and T are calibrated, the location of projection points on an image can be deduced from the coordinates in the real-world system.

The key points on vehicles in two-dimensional images and the three-dimensional real world are typically common features such as headlights, taillights, license plates, *etc.* Existing camera calibration methods assume that the key points of a specific vehicle model (*e.g.,* Tesla Model S, Toyota Corolla) are known. Under the 4L characteristics, camera images are too blurry for us to distinguish vehicle models. Hence, in the proposed method, a set of vehicle model candidates is built to serve as the references of three-dimensional points. The dataset of two-dimensional and three-dimensional key points for the ith vehicle in images and in the real world can be represented as

$$\begin{aligned} \sqrt{}^i &= \left\{ p_1^i, p_2^i, \ldots, p_k^i, \ldots, p_{M_i}^i \right\} \\ \mathcal{P}_j^i &= \left\{ P_{j,1}^i, P_{j,2}^i, \cdots, P_{j,k}^i, \cdots, P_{j,M_i}^i \right\}, \\ & \quad i \le n, j \le m, \end{aligned} \tag{4}$$

where i represents the vehicle index in images and j represents the index of vehicle models. $\sqrt{}^i$ represents the set of two-dimensional key points of the ith vehicle on camera images, and $\mathcal{P}_j^i$ denotes the sets of three-dimensional key points of the ith vehicle in real world assuming the vehicle model is j. n and m represent the number of vehicles and the number of vehicle models in the real world, respectively. M_i denotes the number of key points on the ith vehicle. More specifically, p_k^i represents the location of the kth key point on vehicle i in the image, and $P_{j,k}^i$ represents the three-dimensional coordinates of the kth key point on the vehicle i assuming that the vehicle model is j.

The `MVCalib` method

In this section, we present the proposed multi-vehicle camera calibration method `MVCalib`. The pipeline of `MVCalib` is shown in Fig. 3.

`MVCalib` proceeds through three stages: candidate generation, vehicle model matching and parameter fine-tuning. In the candidate generation stage, the solution candidates for each vehicle are generated separately based on conventional camera calibration methods. In the vehicle model matching stage, a specific model is assigned to each vehicle in the camera images. In the parameter fine-tuning stage, joint information on multiple vehicles is utilized to fine-tune the camera parameters. The fine-tuned value of f, R, T will be carried out to estimate the road length for the traffic density estimation.

Candidate generation. In the candidate generation stage, we first apply the conventional camera calibration method to the key points on each vehicle, assuming that its vehicle model and the focal length of the camera are known. Mathematically, for the ith vehicle, the coordinates of M_i pairs of key points in two-dimensional space $\sqrt{}^i$ and in three-dimensional space $\mathcal{P}_j^i$ under the jth model are known. Given a default value of focal length $\hat{f}$, the parameters of rotation matrix R and translation vector T can be estimated through the Efficient PnP algorithm (EPnP) [27] with a random sample consensus (RANSAC) strategy [44].

The EPnP method is applied to all pairs of (i, j), and hence a total number of $m \times n$ times of estimation using EPnP are conducted. The estimated camera parameters (candidates) are denoted as $\tilde{\psi}_j^i = \left\{ \hat{f}, \tilde{R}_j^i, \tilde{T}_j^i \right\}$, which represents the focal length, rotation matrix and translation vector for the ith vehicle of the jth model.

Vehicle model matching. In the vehicle model matching stage, the most closely matched vehicle model is determined to minimize the projection error from the real world to the image plane for each vehicle i. Mathematically, we aim to select the best vehicle model j from $\tilde{\psi}_j^i$ to obtain the camera parameter ψ^i for each vehicle. In the candidate generation stage, the focal length is fixed to a default value, which may contribute to errors in the projection. Therefore, in this stage, we adjust the focal length to a more accurate value and refine the parameter estimation. To this end, we formulate an optimization problem with the objective of minimizing the projection loss from the three-dimensional real world to two-dimensional camera images, as presented in Eq. (5).

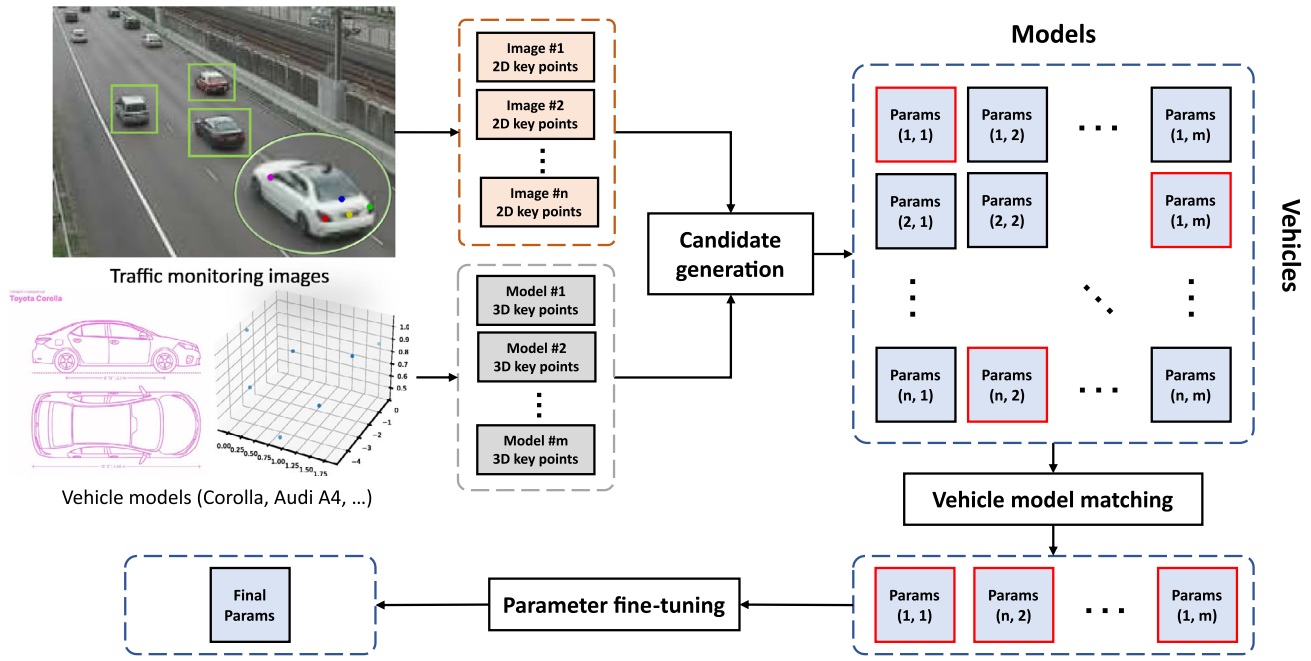


Fig. 3 The pipeline of the `MVCalib` method for camera calibration

$$
\begin{aligned}
L_v\left(\psi_j^i\right) = \min_{\psi_j^i} \sum_{k=1}^{M_i} \left\| p_k^i - \frac{1}{s_{j,k}^i} K\left(f_j^i\right)\left[R_j^i | T_j^i\right] P_{j,k}^i \right\|_2 \\
\text{s.t.} \quad \psi_j^i = \left\{ f_j^i, R_j^i, T_j^i \right\} \\
s_{j,k}^i = R_j^i\Big|_3 \cdot P_{j,k}^i + T_j^i\Big|_3 \\
f_j^i \geq 0, \forall i \leq i, j \leq m
\end{aligned}
\tag{5}
$$

where $L_v(\cdot)$ defines the projection loss from the three-dimensional real world to two-dimensional images for the key points on vehicles. $s_{j,k}^i = R_j^i\Big|_3 \cdot P_{j,k}^i + T_j^i\Big|_3$ is the scale factor for the combination of the kth key point on the ith vehicle with the jth model. $R_j^i\Big|_3$ represents the third row of the rotation matrix and $T_j^i\Big|_3$ denotes the third element of the translation vector. The focal length of a camera f_j^i should be greater than 0.

To solve the optimization problem $L_v(\psi_j^i)$, we employ the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [45], which is an evolutionary algorithm for non-linear and non-convex optimization problems, to search for the optimal parameter ψ_j^i for each combination of vehicle and vehicle model. As the performance of the CMA-ES depends on the initial points, we start by searching for the parameters from $\tilde{\psi}_j^i$. For vehicle i, we assign the vehicle model with the minimal projection loss L_v, as presented in Eq. (6).

$$
\psi^i = \left\{ f^i, R^i, T^i \right\} = \operatorname{argmin} L_v\left(\psi_j^i\right), \quad \forall i \leq n \tag{6}
$$

Parameter fine-tuning. In this stage, we combine the key point information on multiple vehicles and further fine-tune the information to obtain the final estimation of the camera parameters ψ. In previous stages, we made use of the key point information on every single vehicle and applied the estimated camera parameter ψ^i to each vehicle i separately. Ideally, if ψ^i is perfectly estimated, we can project the key points on all vehicles in camera images back to the real world using ψ^i, and those key points should exactly match the key points on the vehicle models. Based on this criterion, we can select the camera parameters from ψ^i and further fine-tune them to obtain ψ.

To this end, we back-project the two-dimensional points in camera images to the three-dimensional real world by using the parameter $\psi^{i'}$ for vehicle i' as an "anchor". Mathematically, given an ith vehicle, the coordinates of the kth key point on the camera image and in the real world can be represented as p_k^i and P_k^i, respectively. Note that P_k^i is a member of $\{P_{j,k}^i|, 1 \leq j \leq m\}$ as the vehicle model is fixed in the vehicle model matching stage. To back-project p_k^i to the real-world space using $\psi^{i'}$, we solve a system of equations derived from Eq. 2, as shown in Eq. (7).

$$
\begin{cases}
\left(\tilde{u}_k^i \left[R^{i'} | T^{i'}\right]\Big|_3 - \left[R^{i'} | T^{i'}\right]\Big|_1\right) \cdot \begin{bmatrix} \hat{P}_k^i(\psi^{i'}) \\ 1 \end{bmatrix} = 0 \\
\left(\tilde{v}_k^i \left[R^{i'} | T^{i'}\right]\Big|_3 - \left[R^{i'} | T^{i'}\right]\Big|_2\right) \cdot \begin{bmatrix} \hat{P}_k^i(\psi^{i'}) \\ 1 \end{bmatrix} = 0
\end{cases}
\tag{7}
$$

where $\tilde{u}_k^i = \frac{u_k^i - \frac{w}{2}}{f^{i'}}$, $\tilde{v}_k^i = \frac{v_k^i - \frac{h}{2}}{f^{i'}}$, (u_k^i, v_k^i) is the two-dimensional coordinate of the kth key point on vehicle i in the camera images, and $\hat{P}_k^i(\psi^{i'})$ represents the back-projected point on the ith vehicle of the kth key point given the camera parameter $\psi^{i'}$ of anchor vehicle i'.

The primary loss between back-projected points and real-world points is defined in Eq. (8).

$$\begin{aligned} L_p\left(i, \psi^{i'}|\alpha\right) &= \sum_{k_1<k_2<M_i} \xi_l\left(i, k_1, k_2, \psi^{i'}\right) \\ &\quad + \alpha\xi_r\left(i, k_1, k_2, \psi^{i'}\right), \\ \xi_l\left(i, k_1, k_2, \psi^{i'}\right) &= \left\| \left\| \overrightarrow{\hat{P}_{k_1}^i(\psi^{i'})\hat{P}_{k_2}^i(\psi^{i'})} \right\|_2 - \left\| \overrightarrow{P_{k_1}^i P_{k_2}^i} \right\|_2 \right|, \\ \xi_r\left(i, k_1, k_2, \psi^{i'}\right) &= \sqrt{1 - \cos\left(< k_1, k_2 >\right)^2} \\ \cos\left(< k_1, k_2 >\right) &= \left(\frac{\overrightarrow{P_{k_1}^i P_{k_2}^i} \cdot \overrightarrow{\hat{P}_{k_1}^i(\psi^{i'})\hat{P}_{k_2}^i(\psi^{i'})}}{\left\| \overrightarrow{P_{k_1}^i P_{k_2}^i} \right\|_2 \cdot \left\| \overrightarrow{\hat{P}_{k_1}^i(\psi^{i'})\hat{P}_{k_2}^i(\psi^{i'})} \right\|_2} \right), \end{aligned} \tag{8}$$

where $\xi_l\left(i, k_1, k_2, \psi^{i'}\right)$ and $\xi_r\left(i, k_1, k_2, \psi^{i'}\right)$ represent the distance and angle loss between the back-projected points and real-world points, and α is a hyper-parameter that adjusts the weight of each loss. $\left\| \overrightarrow{P_{k_1}^i P_{k_2}^i} \right\|_2$ and $\left\| \overrightarrow{\hat{P}_{k_1}^i(\psi^{i'})\hat{P}_{k_2}^i(\psi^{i'})} \right\|_2$ are vectors that consist of any two real-world and back-projected points on the same vehicle i. k_1 and k_2 represents two non-overlapping indices of the key points on the same vehicle i. The distance loss represents the gap between the Euclidean distance of the back-projected points and one of the real-world points, while the angle loss can be regarded as the sine value of the angle between two vectors formed with the back-projected points and real-world points. We further aggregate the loss $L_p\left(i, \psi^{i'}|\alpha\right)$ for different vehicles i based on their relative distance. In general, if a vehicle is further from the anchor vehicle, then the loss in the back-projected points is larger, and we have less confidence in these points. Therefore, smaller weights are assigned to vehicles that are further from the anchor vehicle.

The objective of minimizing the fine-tuning loss for all vehicles is formulated to consider different weights due to the relative distance, as presented in Eq. (9).

$$\begin{aligned} L_f\left(\psi^{i'}|\alpha, \tau\right) &= \sum_{i<n} \omega\left(\hat{C}_i, \hat{C}_{i'}|\tau\right) L_p\left(i, \psi^{i'}|\alpha\right) \\ \omega\left(\hat{C}_i, \hat{C}_{i'}|\tau\right) &= \frac{\exp\left(\tau \left\| \overrightarrow{\hat{C}_i \hat{C}_{i'}} \right\|_2\right)}{\sum_{i''<n} \exp\left(\tau \left\| \overrightarrow{\hat{C}_{i'} \hat{C}_{i''}} \right\|_2\right)} \\ \hat{C}_i &= \frac{1}{M_i} \sum_{k<M_i} \hat{P}_k^i(\psi^{i'}) \\ \hat{C}_{i'} &= \frac{1}{M_{i'}} \sum_{k<M_{i'}} \hat{P}_k^{i'}(\psi^{i'}) \end{aligned} \tag{9}$$

where $\hat{C}_i$ is the centroid of all back-projected key points on the ith vehicle, and $\hat{C}_{i'}$ is the centroid of all back-projected key points on the anchor vehicle i'. $\omega\left(\hat{C}_i, \hat{C}_{i'}|\tau\right)$ is the weighting function for vehicle i using the vehicle i' as an anchor. The temperature τ is a hyper-parameter that controls the distribution of the weighting function. When $\tau = 0$, the weighting function uniformly averages the loss for all vehicles; when $\tau < 0$, more attention will be paid to vehicles that are close to the current vehicle, and vice versa.

To obtain the final estimation of the camera parameters, we minimize the objective $L_f\left(\psi^{i'}|\alpha, \tau\right)$ in Eq. (9) for each selection of anchor vehicle. The optimal estimation is selected as that with the minimal loss, as shown in Eq. (10).

$$\psi = \text{argmin}_{i'<n} L_f\left(\psi^{i'}|\alpha, \tau\right) \tag{10}$$

Vehicle detection

In this section, we present the vehicle detection model, which counts the number of vehicles on road segments from camera images. The state-of-the-art vehicle detection models adopt Deep Learning (DL) based methods to train the model on a vehicle-related dataset. The training process of DL models usually requires massive data. Owing to the 4 L characteristics, the quantity of annotated camera images for a specific traffic surveillance system cannot support the complete training of a modern DL-based vehicle detection model. In addition, it is inefficient to train new models for each traffic surveillance system separately. Therefore, we adopt the transfer learning scheme to first train the model on traffic-related public datasets and then apply the model to specific surveillance camera systems [46].

Existing public datasets are designed for a range of purposes, such as vehicle re-identification (reID), autonomous driving, vehicle detection, *etc.* [16, 47–52]. The camera images in different datasets have different endogenous attributes (*e.g.*, focal length, type of photosensitive element, resolution, *etc.*) and exogenous attributes (*e.g.*, perspective, illumination, directions, *etc.*) Additionally, the datasets differ in size. A summary of the existing traffic-related public datasets is presented in Table. 2, and snapshots of some of the datasets are shown in Fig. 4.

We categorize the camera images from these datasets into different traffic scenarios, which include the time of day (daytime and nighttime), congestion level, surrounding environment, *etc.* Each traffic scenario represents a unique set of features in the camera images, so if a DL model is trained for one traffic scenario, it might not perform well on a different scenario. Given the 4 L characteristics, the camera images in a large-scale traffic surveillance system may cover multiple traffic scenarios, so it is important to merge and balance

Table 2 A summary of traffic-related image datasets

Name	Size	Resolution	Camera angle	Original usage
BDD100K	100,000	1280 × 720	Front	Autonomous driving
BIT Vehicle	9,850	Multiple	Inclined top	Vehicle reID
CityCam	60,000	352 × 240	Inclined top	Vehicle detection
COCO	17,684	Multiple	Multiple	Object detection & segmentation
MIO-TCD-L	137,743	720 × 480	Inclined top	Vehicle detection & classification
UA-DETRAC	138,252	960 × 540	Inclined top	Vehicle detection

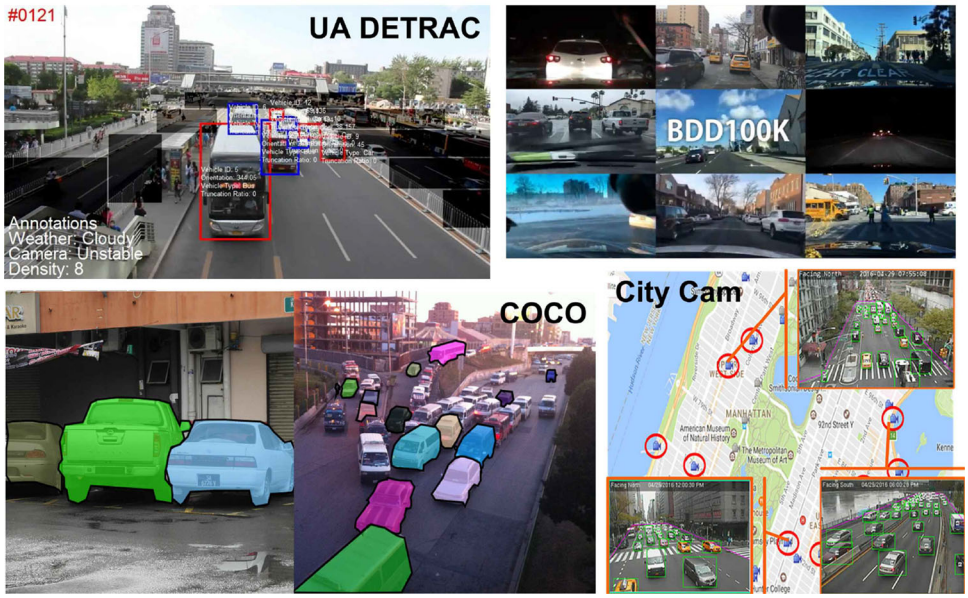


Fig. 4 A glance of various traffic-related image datasets used in this study

the different datasets summarized in Table. 2 for training the vehicle detection model.

To this end, we formulate a Linear Program (LP) to hybridize a generalized dataset called the **LP hybrid dataset**, by re-sampling from multiple traffic-related public datasets. The LP hybrid dataset balances the proportion of images from each traffic scenario to prevent one traffic scenario from dominating the dataset. For example, if most camera images are captured during the daytime, then the trained vehicle detection model will not perform well on the nighttime images. If different traffic scenarios are comprehensively covered, balanced, and trained, the robustness and generality of the detection model will be significantly improved.

Following the above discussion, the pipeline for the vehicle detection model is presented in Fig. 5.

One can see that the multiple traffic-related datasets are fed into the LP to generate the LP hybrid dataset, and the dataset will be used to train the vehicle detection model. The trained model can be directly applied to different traffic surveillance systems.

As stated above, the hybrid detection dataset is formulated as an LP, the goal of which is to maximize the total number of images in the dataset, written as

$$\max \sum_{\mu=0}^{u-1} \sum_{v=0}^{v-1} q_{\mu,v} \tag{11}$$

where u denotes the number of datasets, and v represents the number of traffic scenarios. $q_{\mu,v}$ are decision variables that denote the number of images to be incorporated into the LP hybrid dataset from dataset μ for traffic scenario v.

The constraints of the proposed LP are constructed based on two principles: (1) The difference between the numbers of images from different traffic scenarios should be limited within a certain range. (2) The number of images contributed by each dataset should be similar. Mathematically, the constraints are presented in Eq. (12).

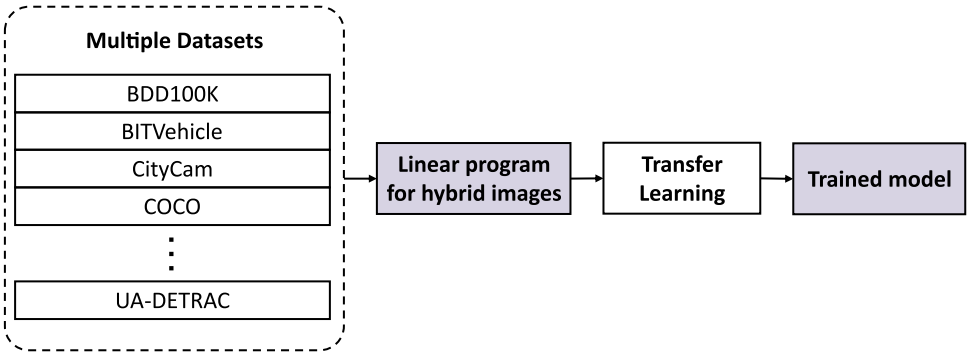


Fig. 5 The pipeline of vehicle detection

$$
\begin{aligned}
& q_{\mu,v} - \frac{(1+\beta)\sum_{\mu=0}^{u-1} q_{\mu,v}}{\sum_{\mu=0}^{u-1} \delta(0, Q_{\mu,v})} && \leq 0, \\
& \forall 0 \leq \mu < u, 0 \leq v < v, Q_{\mu,v} \neq 0 \\
& q_{\mu,v} - \frac{(1-\beta)\sum_{\mu=0}^{u-1} q_{\mu,v}}{\sum_{\mu=0}^{u-1} \delta(0, Q_{\mu,v})} && \geq 0, \\
& \forall 0 \leq \mu < u, 0 \leq v < v, Q_{\mu,v} \neq 0 \\
& \sum_{\mu=0}^{u-1} q_{\mu,v} - \frac{1+\gamma}{v} \sum_{\mu=0}^{u-1} \sum_{v=0}^{v-1} q_{\mu,v} && \leq 0, \\
& \forall 0 \leq v < v \\
& \sum_{\mu=0}^{u-1} q_{\mu,v} - \frac{1-\gamma}{v} \sum_{\mu=0}^{u-1} \sum_{v=0}^{v-1} q_{\mu,v} && \geq 0, \\
& \forall 0 \leq v < v \\
& q_{\mu,v} && \leq Q_{\mu,v}, \\
& \forall 0 \leq \mu < u, 0 \leq v < v \\
& q_{\mu,v} && \geq 0, \\
& \forall 0 \leq \mu < u, 0 \leq v < v
\end{aligned}
\tag{12}
$$

where the former two constraints adjust the image contribution from different datasets, while the latter two balance the number of images from different traffic scenarios. $Q_{\mu,v}$ represents the total number of data for traffic scenario v in dataset μ, and $q_{\mu,v} \leq Q_{\mu,v}$ enforces that the selected number of images should be smaller than the total number of images. β is the maximum tolerance parameter for the upper and lower bound of the image number in different traffic datasets given certain scenarios, and γ is another maximum tolerance parameter limiting the difference between the numbers of images selected from different scenarios. $\delta(0, Q_{\mu,v})$ is defined as $\delta(0, Q_{\mu,v}) = \begin{cases} 0, & Q_{\mu,v} = 0 \\ 1, & Q_{\mu,v} \neq 0 \end{cases}$. Combining the objective in Eq. (11) and constraints in Eq. (12), we can formulate the LP hybrid dataset that maximizes the number of data and balances the contributions of data from different datasets as well as traffic scenarios.

The vehicle detection model is built on top of You Only Look Once (YOLO)-v5, a widely used object detection model [53]. YOLO-v5 is initially pre-trained on the full COCO dataset, and we adopt the transfer learning scheme to inherit the pre-trained weights and tune the weight parameters on the LP hybrid dataset. The YOLO-v5 network is a general framework for detecting and classifying objects simultaneously. In the vehicle detection context, we only need to box out the vehicles from the background images regardless of vehicle type. Hence we reshape the output dimension into one with randomly initialized parameters. As the LP hybrid dataset contains camera images in various traffic scenarios, we can build a generalized detection model suitable for various traffic surveillance systems in different countries.

Table 3 The comparison for the selected traffic cameras used for case studies in Hong Kong and Sacramento

Attributes	HK	Sac
Resolution	320 × 240 pixels	720 × 480 pixels
Update rate	2 min	1/30 s
Orientation	Vehicle head	Vehicle tail
Road type	Urban road	Highway
Speed limit	50 km/h	105.3 km/h

Numerical experiments

In this section, we conduct numerical experiments on the proposed camera calibration and vehicle detection methods to evaluate the performance of two traffic surveillance camera systems.

Experimental settings

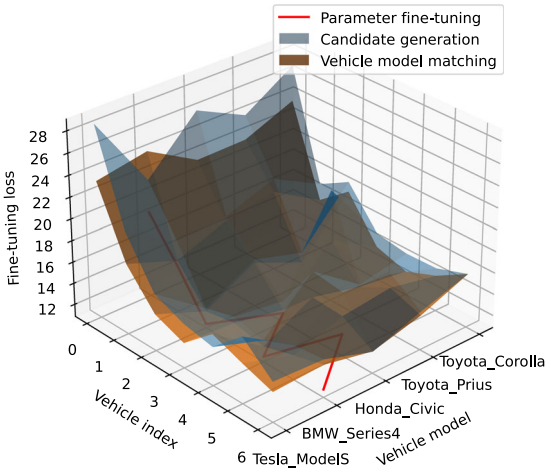
To demonstrate that the proposed framework can be applied to traffic density estimation in countries with different traffic surveillance systems, two case studies of traffic density estimation are conducted, Hong Kong (HK) and Sacramento, California (Sac) where the ground true data can be obtained at both sites. A comparison of these two cameras is shown in Table. 3.

- HK: Camera images in Hong Kong are obtained from HKeMobility[3] at the Chatham Road South, Kowloon, Hong Kong SAR, with the camera code K109F. Images containing seven vehicles are selected from June 22nd to June 25th, 2020.
- Sac: Camera images in Sacramento are obtained from Caltrans system[4] at Capital City Freeway at E Street, Sacramento, CA, the US. Images containing seven vehicles are selected from February 17th to December 18th, 2022.

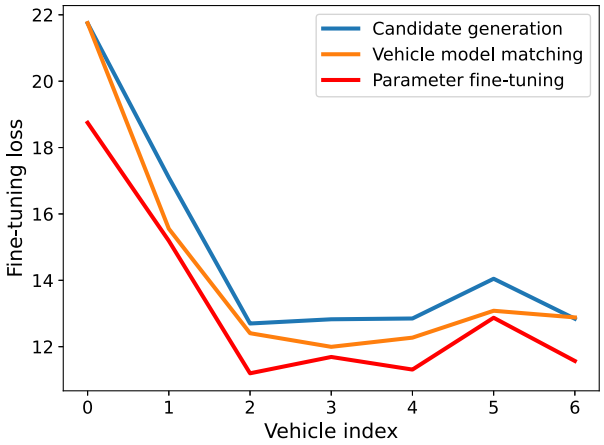
For camera calibration, we select all the vehicles that are not shadowed by other vehicles, and those vehicles are annotated with eight key points: left headlight, right headlight, front license plate center, front wiper center, left wing mirror, right wing mirror, back left corner and back right corner. Any key points not visible in an image are excluded. Besides, five popular vehicle models are involved with three-dimensional information: Toyota Corolla, Toyota Prius, Honda Civic, BMW Series 4 and Tesla Model S. The three-dimensional key points for those models are measured from the *Dimensions*.[5] α in Eq. (9) is set to 6.

For vehicle detection, all of the datasets are summarized in Table. 2 are incorporated. The ratio factors γ and β in Eq. (12) are set to 0.25. The LP hybrid dataset is divided into a training set (80%) and a validation set (20%). A total of 3,812 camera images are annotated to test the performance of the model trained on the LP hybrid dataset.

All experiments are conducted on a desktop with Intel Core i9-10900K CPU @3.7GHz × 10, 2666MHz × 2 × 16GB RAM, GeForce RTX 2080 Ti × 2, 500GB SSD. The camera calibration and vehicle detection models are both implemented with Python. For the camera calibration model, OpenCV [54] is used for computing Eq. (2) and running the EPnP algorithm [27]. In the candidate generation stage, the focal length is fixed at 350 millimeters. The CMA-ES algorithm [45] is executed with the Nevergrad package [55]. The numbers of iterations of CMA-ES in the vehicle model matching and parameter fine-tuning stages are set to 4,000 and 20,000, respectively. When tuning the vehicle detection model, we set the number of training epochs to 300, and other hyperparameters take the default settings[6] of the original YOLO-v5. The Adam optimizer [56] is adopted with a learning rate of 0.001. In the testing stage, the inference speed is more than 100 fps (frames per second). Supposed the camera images are updated every 2 min, which means the frequency of the camera image is 1/120 fps, then a single server can simultaneously process the images from over 12,000 cameras. As surveillance cameras are only required to be calibrated once with few input parameters, the city-wide real-time traffic density estimation can be achieved using the proposed framework.



(a) Fine-tuning losses with all indices and models.



(b) Minimal fine-tuning losses among all vehicle models for each vehicle.

Fig. 6 Fine-tuning losses with parameters in the three stages for camera calibration in HK (vehicle index is defined in Eq. (4)

Experimental results

In this section, we compare the proposed camera calibration and vehicle detection models with existing baselines, respectively.

Camera calibration

To evaluate the performance of the camera calibration method, we first compare the fine-tuning loss defined in Eq.

[3] https://www.hkemobility.gov.hk/tc/traffic-information/live/cctv.

[4] https://cwwp2.dot.ca.gov/vm/iframemap.htm.

[5] https://www.dimensions.com.

[6] https://github.com/ultralytics/yolov5.

(9) among baseline models for the two cameras in HK and Sac. Based on the calibration results, we estimate the road length from the camera images, and the length estimated by each model is compared with the actual length.

To demonstrate the necessity of the three steps in MVCalib, Fig. 6 plots the fine-tuning loss defined in Eq. (9) for the three stages: candidate generation, vehicle model matching and parameter fine-tuning. In particular, Fig. 6a includes the losses of all the vehicle index and vehicle model pairs for the first two stages, and Fig. 6b plots the loss based on the matched vehicle model with the minimal fine-tuning loss. One can see that the fine-tuning loss defined in Eq. (8) decreases after each stage, which indicates that the CMA-ES can successfully reduce the loss in each stage.

We then measure the lengths of road markings on the camera images, as the road markings are invariant features on the road, and their lengths can be determined from measurements or official guidebooks. Detailed road marking information for the HK and Sac studies is shown in Fig. 7.

In Fig. 7a, the length of the white line is 1 m and the interval between the white lines is 5 ms, which are obtained from field measurements. On the camera images, a total of 14 points are annotated at the midpoints of white lines, resulting in 12 line segments of the same length (shown in Fig. 7b). Hence each line segment corresponds to 6 ms in the real world. For the camera images in Sac, we likewise use the actual lengths of the lane markings on the Capital City Freeway as the ground truth. According to the Manual on Uniform Traffic Control Devices (MUTCD) [57], the length of a white line is 10 feet (approximately 3.05 ms) and the interval is 30 feet (approximately 9.14 ms) (shown in Fig. 7c). On the camera images, we annotate 14 points resulting in 12 line segments (shown in Fig. 7d), elongated in 40 feet (approximately 12.19 ms) for each segment.

We compare our method with existing baseline models including EPnP [27], UPnP, UPnP+GN (UPnP fine-tuned with the Gauss-Newton method) [31], GPnP and GPnP+GN (GPnP fine-tuned with the Gauss-Newton method) [32]. The calibration results are shown in Table. 4. The estimated lengths of the road markings on camera images with the actual lengths are compared and three metrics are employed for benchmark comparison: Rooted Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The calculation of MAE, RMSE, and MAPE is shown in Eq. 13.

$$RMSE = \sqrt{\frac{1}{N_{rm}} \sum_{i=0}^{N_{rm}} (l_{rm}^i - \hat{l}_{rm}^i)^2}$$

$$MAE = \frac{1}{N_{rm}} \sum_{i=0}^{N_{rm}} |l_{rm}^i - \hat{l}_{rm}^i|$$

$$MAPE = \frac{1}{N_{rm}} \sum_{i=0}^{N_{rm}} \left| \frac{l_{rm}^i - \hat{l}_{rm}^i}{\hat{l}_{rm}^i} \right|, \tag{13}$$

where N_{rm} represents the number of roadmarks, l_{rm}^i and $\hat{l}_{rm}^i$ are the estimated and actual length of the ith roadmark, respectively. At each stage of MVCalib, we compare its result with baseline methods in terms of their ability to solve the PnPf problem. To conduct an ablation study gauging the contribution of each stage, we run MVCalib with only the first stage (candidate generation), with the first two stages (up to vehicle model matching), and with all three stages. The three models are referred to as MVCalib CG, MVCalib VM, and MVCalib, respectively. In fact, the MVCalib CG is equivalent to the EPnP method.

One can see from Table. 4 that UPnP (GN) and GPnP (GN) yield unsatisfactory solutions owing to the low image quality. As they take the focal length into account, the complexity of the problem is significantly increased, and hence they require high-resolution images and more numerous and accurate annotation points.

As for the ablation study, we compare MVCalib CG, MVCalib VM, and MVCalib to evaluate the contribution of each stage. In the vehicle model matching stage, if we optimize the focal length with other parameters simultaneously, the estimation results are greatly improved relative to MVCalib CG, demonstrating that the estimation of focal length is necessary and important for the calibration of traffic surveillance cameras. In the full MVCalib, we also incorporate the joint information of multi-vehicle under the same camera. MVCalib achieves the best result among all models. For the surveillance camera in HK, the average error is only approximately 40 cms for estimating the six-meter road markings, less than 10% in MAPE. while in Sac, the average error is about 1 m for the forty-foot road markings, less than 10% in MAPE.

Besides, MVCalib outperforms the other models in terms of all three metrics, which means that the calibration results are close to the ground truth. Snapshots of calibration results of surveillance cameras in HK and Sac are shown in Fig. 8, where the distance between any two red dots is one meter.

Owing to the perspective effect, the distance between red dots on images appears closer when they are more distant from the camera. Through visual inspection, we note that the estimation of focal length is reasonable and the skew of perspective error is small. Additional experiments regarding the convergence and sensitivity of the MVCalib are further presented in Appendix. B, and the choice of τ is discussed in Appendix. D.

(a) Real size of road markings in HK.

(b) Testing points from camera images in HK.

(c) Real size of road markings in Sac.

(d) Testing points from camera images in Sac.

Fig. 7 Driving lanes from real world and camera images in HK and Sac

Table 4 The comparison of results of surveillance camera calibrated by different methods in HK and Sac (unit for RMSE and MAE: meter)

Method	HK			Sac		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
UPNP	25.80	22.21	370.03%	9.66	8.86	72.66%
UPNP+GN	2.02	0.62	10.36%	9.40	9.34	76.67%
GPNP	3.14	2.76	46.15%	2.12	1.71	14.09%
GPNP+GN	2.24	1.98	33.15%	2.03	1.64	13.48%
MVCalib CG	1.68	1.49	24.91%	4.48	4.46	36.61%
MVCalib VM	0.98	0.77	12.83%	2.27	1.88	15.45%
MVCalib	**0.49**	**0.43**	**7.22%**	**1.28**	**1.05**	**8.62%**

Bold indicates the best performance

Table 5 The allocation of the vehicle-related dataset in the LP hybrid dataset

Dataset	# images in daytime	# images at nighttime	Total # images
BDD100K	8319	8398	16,717
BITVehicle	7325	0	7325
CityCam	8459	0	8459
COCO	7111	7,619	14,730
MIO-TCD-L	8892	7413	16,305
UA-DETRAC	7955	5407	13,362
Total	48,061	28,837	76,898

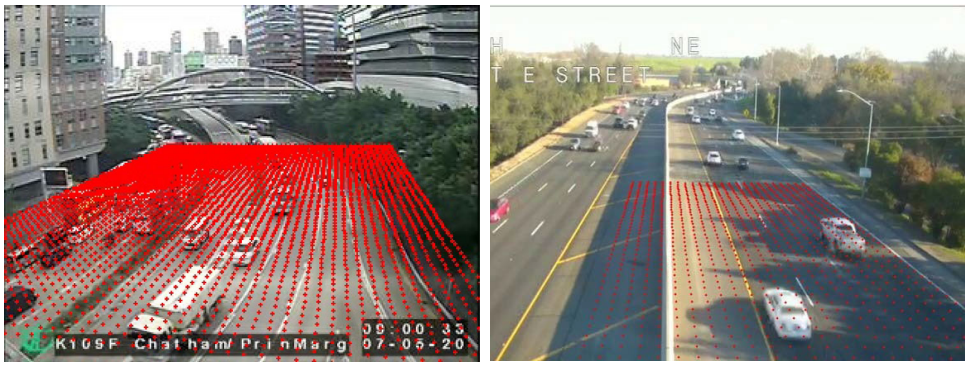


Fig. 8 Snapshots of calibration results in HK and Sac

(a) The Snapshots of calibration result in HK.

(b) The Snapshots of calibration result in Sac.

Table 6 Evaluation results for different detection models on images during the daytime

Name	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Dataset size
BDD-100K	0.361	0.364	0.326	0.144	100,000
BITVehicle	0.255	0.009	0.062	0.035	9,850
CityCam	0.412	0.938	0.881	0.538	60,000
COCO	**0.978**	0.017	0.556	0.340	17,684
MIO-TCD-L	0.737	0.885	0.899	0.578	137,743
Pretrained	0.455	0.899	0.838	0.552	0
UA-DETRAC	0.775	0.693	0.758	0.488	138,252
Spaghetti	0.605	0.948	**0.927**	**0.608**	**434,993**
Random	0.588	0.942	0.919	0.595	76,898
LP hybrid	0.583	**0.949**	0.921	0.594	76,898

Bold indicates the best performance

Vehicle detection

In the detection model, two traffic scenarios are considered: daytime and nighttime. A total of 76,898 images are hybridized in the LP-hybrid detection dataset after solving for the LP in Eq. (11) and (12). The detailed allocation of the 76,898 images is presented in Table. 5.

To evaluate the generality of the vehicle detection model trained on the LP hybrid dataset, we also train the YOLO-v5 individually with the BDD100K, BITVehicle, CityCam, COCO, MIO-TCD-L, and UA-DETRAC datasets for benchmark comparison. Additionally, an integrated dataset incorporating all of the aforementioned datasets without balancing the numbers of images in the daytime and at nighttime is also considered, called the Spaghetti dataset, is also compared. Moreover, we down-sample five datasets named Random dataset whose sizes are the same to the LP hybrid dataset to ablate the influence of image number. The mean of Random dataset performance with be considered in the final results. For the model trained on each dataset, we report the vehicle detection accuracy on the testing data. Several metrics are used in evaluating the performance of the vehicle detection models, including precision, recall, AP@0.5, and AP@0.5:0.95. Interpretation about these metrics is in Appendix. C. In this paper, the threshold for IoU (Intersection over Union) is 0.45 and the threshold for object confidence is 0.25, which are the default settings in YOLO-v5.

Tables 6 and 7 present the evaluation results for the models trained with the LP hybrid and other datasets for daytime and nighttime, respectively. The model trained on the LP hybrid dataset reaches the highest recall rate and also achieves a desirable precision rate. The high recall rate means the model trained on the LP hybrid dataset is confident to find as many vehicles as possible in surveillance images. In daytime detection, the gaps between the model trained on the LP hybrid dataset and the ones trained on the Spaghetti and Random dataset based on recall rate are not conspicuous, since most images in the training set are shot during daytime. When it comes to nighttime detection, the recall rate has elevated by 2-3% compared to the model trained on Spaghetti and Random dataset, which demonstrates that the proposed hybrid strategy is adept at catching vehicles at night while maintaining a decent detection performance during daytime. For the metrics of mAP@0.5 and mAP@0.5:0.95, the model trained on the Spaghetti dataset achieves the best performance, but the gap between the models trained on the Spaghetti dataset, the

Table 7 Evaluation results for different detection models on images during the nighttime

Name	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Dataset size
BDD-100K	0.443	0.316	0.302	0.124	100,000
BITVehicle	0.058	0.001	0.018	0.010	9,850
CityCam	0.402	0.793	0.713	0.412	60,000
COCO	**0.949**	0.003	0.397	0.223	17,684
MIO-TCD-L	0.805	0.746	0.817	0.511	137,743
Pretrained	0.387	0.862	0.781	0.471	0
UA-DETRAC	0.708	0.573	0.629	0.365	138,252
Spaghetti	0.689	0.872	0.882	**0.546**	**434,993**
Random	0.674	0.864	0.871	0.536	76,898
LP hybrid	0.653	**0.89**	**0.886**	0.545	76,898

Bold indicates the best performance

Random dataset and the LP hybrid dataset for mAP@0.5 is less than 1% and the gap for mAP@0.95 is less than 2%. For images at nighttime, the model on the LP hybrid dataset outperforms those trained on the Spaghetti dataset and the Random dataset on mAP@0.5, also indicating that the proposed LP hybrid dataset can improve the detection performance at night.

Moreover, the Spaghetti dataset contains more than 430,000 images, which takes more than 21 days for training. The LP hybrid dataset is a strategic sample from the Spaghetti dataset whose size is about 76,000, one-sixth of the Spaghetti dataset. It only takes 6 days to train the model on the LP hybrid dataset, but it reaches a competitive performance with the model trained on the Spaghetti dataset. The Random dataset is a random sample from the Spaghetti dataset with the same size as the LP hybrid dataset. While it maintains a decent performance in daytime detection, nighttime detection has been weakened since it contains fewer images at night. The size of the rest dataset varies a lot. However, based on the performance, the models trained on these datasets may not be able to transfer into new scenes.

In a nutshell, compared with a dataset of similar size, the LP Hybrid dataset can balance the accuracy for different scenarios (e.g., daytime, nighttime, etc.), which has important real-world implications. Compared with other datasets with larger sizes, the LP Hybrid data can achieve similar performance in different scenarios, but it takes less time for training the detection model. The LP hybrid approach is a simple and easy-to-implement approach to boost detection accuracy for rare traffic scenarios in the face of limited computational resources.

Case study I: surveillance cameras in Hong Kong

In this section, we conduct a case study of traffic density estimation using camera images on Chatham Road South, Hong Kong SAR. Given the study region, we divide the roads into four lanes (numbered along the x-axis), and define vehicle locations along the y-axis, as shown in Fig. 9.

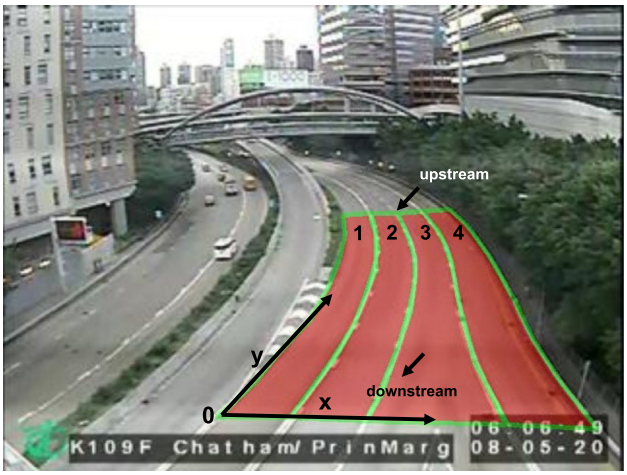


Fig. 9 Driving lanes for traffic density estimation beneath the surveillance camera

The length of each lane can be estimated from the images using the calibration results, and the number of vehicles can be counted using the vehicle detection model. The traffic density in each lane can be estimated by dividing the number of vehicles by the length of each lane at each location and time point. To evaluate the estimated density, a high-resolution (1920×1080 pixels per frame) camera is installed shooting the same region with different directions, and the camera video is acquired in this case study as a ground truth. The video recorded by this camera, shows the traffic conditions over 21 h from 11:30 PM, September 23rd to 8:30 PM, September 24th, 2020.

An overview of the vehicle detection results is presented in Fig. 10. Figure 10A displays a snapshot of vehicle detection using the model trained on the LP hybrid dataset of images taken in the daytime. By boxing out identical study regions in the traffic surveillance camera images and high-resolution videos (shown in Fig. 10B, C), the estimated number of vehi-

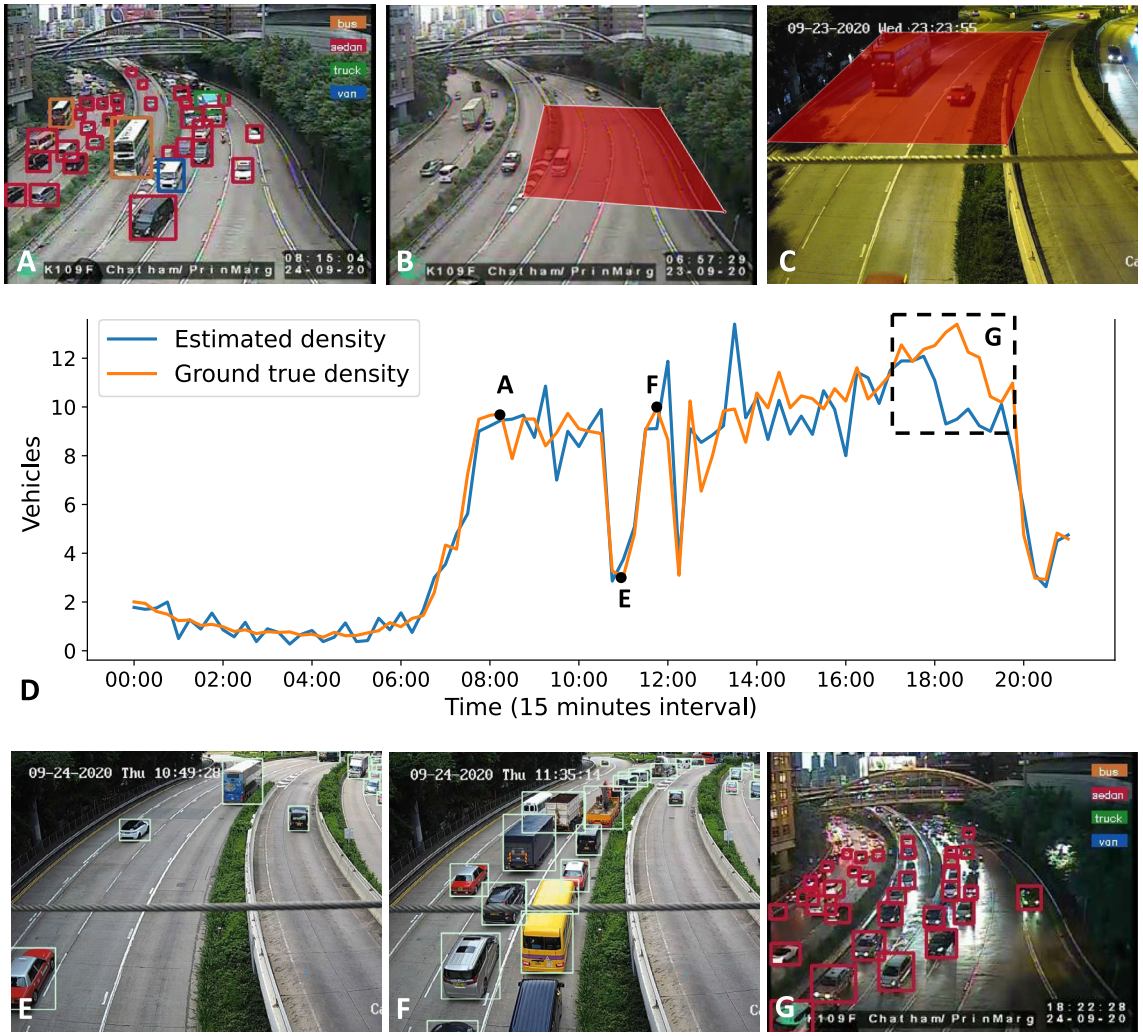


Fig. 10 Overview of the vehicle detection results in HK

Table 8 The RMSE, MAE and MAPE of the estimated density for different lanes from surveillance cameras in HK (unit: RMSE, MAE: veh/km/lane)

Lane ID	RMSE	MAE	MAPE
Lane #1	16.94	12.65	19.60%
Lane #2	12.98	9.23	27.48%
Lane #3	11.11	7.77	41.24%
Lane #4	8.70	6.53	50.44%
Average	12.43	9.04	34.69%

cles can be compared with the ground truth in Fig. 10D. We select four points or regions in Fig. 10D, which are shown in Fig. 10A, E, F and G. Figure 10A shows the beginning of the morning peak when the vehicle number significantly increases. Lanes #1 and #2 in the study region (numbered from the left) become visibly congested in the camera images. Points E and F are a pair of points that depicts contrasting traffic conditions when the traffic density fluctuates dramatically in a short time interval. If we inspect images taken around 11:00 AM and 11:30 AM, respectively on September 24, 2020, which are the corresponding points E and F. In Fig. 10E, it can be seen that there are few vehicles on the road, and hence the traffic density is relatively low at point E. However, at point F, there is a sharp increase in the demand on the road. The traffic condition oscillates owing to the traffic signals downstream, which causes the pronounced changes between points E and F. Figure 10G depicts the traffic conditions at the evening peak when the vehicle number reaches the daily maximum. The evening peak fades away quickly and disappears at approximately 8:00 PM.

Compared to the estimated and ground true traffic density, the developed model succeeded in tracking the growth of the morning peak and detecting the fluctuation of traffic conditions. However, at point G, some of the vehicles are miss-detected in the evening peak. This may have been

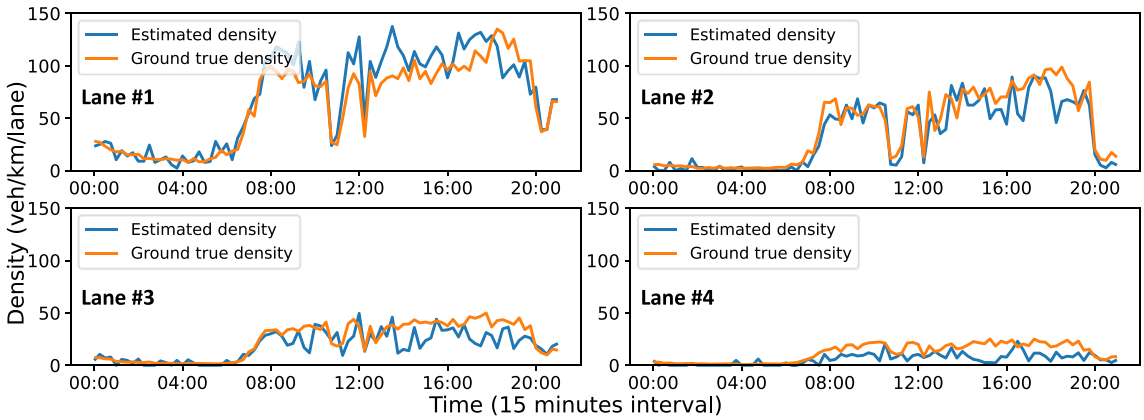


Fig. 11 The variation of traffic density from 00:00 AM to 9:00 PM in HK

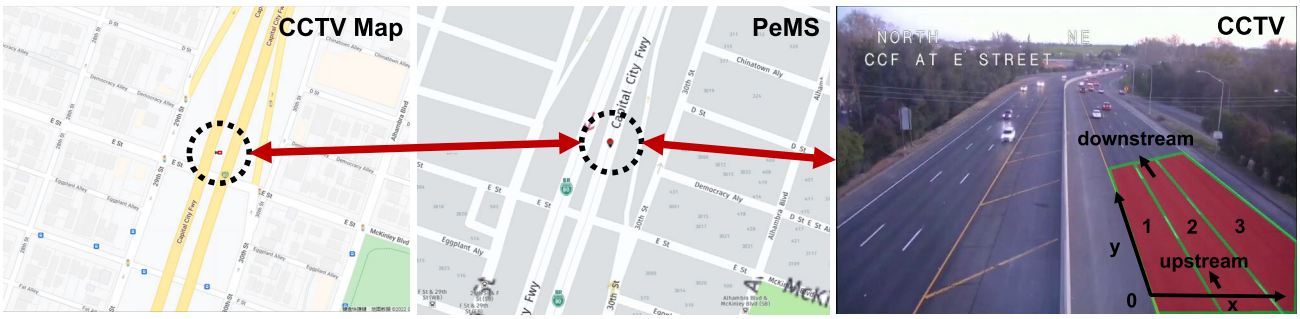


Fig. 12 The location of the matched surveillance camera and double-loop detector in Caltrans

caused by dazzles from the headlights and the light reflected from the ground, which make it difficult for the vehicle detection model to identify the features of vehicles. This phenomenon is a common issue in Computer Vision (CV), which will be left for future research. Overall, the estimated result is close to the ground truth most of the time, which demonstrates that the detection model can accomplish an accurate detection despite the low resolution and low frame rate of the images.

The RMSE, MAE, and MAPE of the estimated traffic density for each lane and the entire road are presented in Table. 8, and a comparison of traffic densities from estimation and the ground truth is shown in Fig. 11.

One can see that the estimated density approximates the actual density, and the density fluctuation is accurately captured. The MAPE is relatively high because this metric is sensitive when the density is small. For example, if the true density is 2 veh/km/lane, while the estimated density is 1 veh/km/lane, then the MAPE is 50%. The traffic density of Lane #1 is overestimated with an MAE of approximately 12 veh/km/lane, while the traffic density of Lanes #2, #3 and #4 are underestimated with an MAE of approximately 9, 7, and 6 veh/km/lane, respectively. The possible causes of the under estimation and over estimation are two-fold: (1) The frame rate is not sufficient enough to support an individual estimation for each lane. Since the image will be updated once every two minutes, an average of 7.5 images will be accumulated in a time interval of 15 min. The estimation may result in a biased estimation since the small-size samples happen to capture the non-recurrent patterns of the traffic density. (2) The determination of the lane of each vehicle may be biased, as the lane occupied by each vehicle is determined by the center of the bounding box. When the road is curved in images and the vehicle is large, the center of the bounding box may shift to another lane, affecting the accuracy of the estimations in both lanes.

Case study II: surveillance cameras in Sacramento

To demonstrate the generality of the proposed framework, another case study is conducted using a surveillance camera in the Caltrans system. The monitoring video data is collected from the camera on the Capital City Freeway, Sacramento, CA (shown in Fig. 12 left).

A 24-hour video is downloaded, covering the period from 12:00 AM on February 17th to 12:00 AM on February 18th, 2022. Similar to the procedures for HK, key points on vehicles are annotated manually for camera calibration. The ground

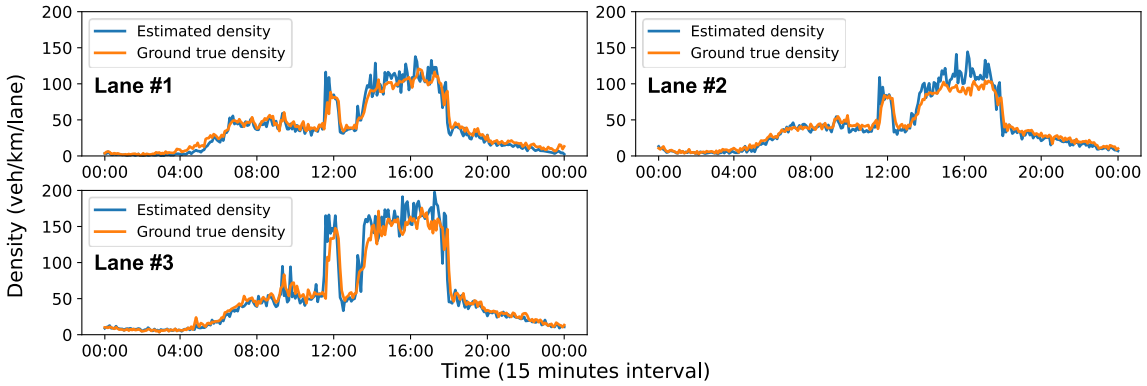


Fig. 13 The variation of traffic density in 24 h in Sac

Table 9 The RMSE, MAE and MAPE of the estimated density in different lanes from surveillance cameras in transition and non-transition time in Sac (unit for RMSE, MAE: veh/km/lane)

Lane ID	Transition time			Non-transition time		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Lane #1	17.38	11.60	17.80%	8.84	5.76	24.53%
Lane #2	16.62	12.04	19.23%	9.75	5.85	16.48%
Lane #3	31.33	22.62	24.85%	14.03	7.44	14.23%
Average	21.78	15.42	20.63%	10.87	6.35	18.31%

true density data are obtained from a double-loop detector at the same location (shown in Fig. 12 center) within the same time period. The detector data are obtained from the PeMS system, which includes the average traffic speed, density, and flow data. Given the study region, we can also divide the roads into three lanes (numbered along the x-axis), and define vehicle locations along the the y-axis, as shown in Fig. 12 right.

The accuracy of the estimated traffic density is shown in Fig. 13. The blue curve represents the estimated density, while the orange curve represents the ground truth. The gaps between these curves are small from visual inspection, and the proposed framework can successfully detect sudden changes happened in traffic density. Furthermore, the estimation accuracy does not deteriorate at nighttime. In this scenario, the sunset ended about 6:00 PM, but the proposed framework can follow the evening peak even after 6:00 PM. To evaluate the estimation accuracy in transition time from free-flow to congestion or from congestion to free-flow regimes, we divide the 24 h into two density regimes, transition time and non-transition time. The transition time is from 11:00 AM to 1:00 PM and from 5:00 PM to 7:00 PM. The non-transition time consists of rest time intervals. From Table. 9,

the average MAPE in transition time and non-transition time are 20.63% and 18.31%, respectively. Though the MAPE in transition time is 2% higher than that in non-transition time, MAPEs remain at the same level in transition time and non-transition time, meaning that the method can capture the transition in traffic density.

Conclusions

In this paper, we propose a framework for traffic density estimation using traffic surveillance cameras with 4 L characteristics, and the **4L** represents **L**ow frame rate, **L**ow resolution, **L**ack of annotated data, and **L**ocated in complex road environments. The proposed density estimation framework consists of two major components: camera calibration and vehicle detection. For camera calibration, a multi-vehicle calibration method named MVCalib is developed to estimate the actual length of roads from camera images. For vehicle detection, the transfer learning scheme is adopted to fine-tune the deep-learning-based model parameters. A linear-program-based data mixing strategy that incorporates multiple datasets is proposed to synergize the performance of the vehicle detection model in different traffic scenarios.

The developed camera calibration and vehicle detection models are compared with existing baseline models in terms of the performance on real-world surveillance camera data in Hong Kong and Sacramento, and both models outperform the existing state-of-the-art models. The MAE of camera calibration is less than 0.4 ms out of 6 ms, and the accuracy of the detection model is approximately 90%. We further conduct two case studies in Hong Kong and Sacramento to evaluate the quality of the estimated density. The experimental results indicate that the MAE for the estimated density is 9.04 veh/km/lane in Hong Kong and 7.03 veh/km/lane in Sacramento. Comparing the estimation results in the two study regions, we also observe that the performance of the proposed density estimation framework degrades under low-quality images and high-illumination-intensity environments. Considering the robustness of surveillance cameras and the estimation accuracy, we think the performance is

acceptable for current transport industries. The proposed framework has great potential for large-scale traffic density estimation from surveillance cameras in cities across the globe and it could provide considerable and fruitful information for traffic operations and management applications.

In future research, we would like to extend the proposed framework to estimate other traffic state variables such as speed, flow, and occupancy. In the camera calibration method, the key points of each vehicle are manually labeled, which can be further automated [30]. In addition to the vehicle detection model, a vehicle classification model could also be developed to estimate traffic density by vehicle type. Moreover, it would be of practical value to develop a fully automated and end-to-end pipeline to deploy the proposed density estimation framework in different traffic surveillance systems.

Acknowledgements The work described in this study was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU R5029-18, R7027-18, and PolyU/25209221), a grant from the Research Institute for Sustainable Urban Development (RISUD) at the Hong Kong Polytechnic University (Project No. P0038288), and a grant from the Otto Poon Charitable Foundation Smart Cities Research Institute (SCRI) at the Hong Kong Polytechnic University (Project No. P0043552). The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

Data Availability The data that support the findings of this study are available from the corresponding author upon request.

Appendix A Impact of Annotation Errors of Key Points on the MVCalib Algorithm

To discuss the impact of annotation errors of key points on the proposed MVCalib algorithm, we set different experiments in HK and Sac. In HK, we randomly shift the location of the annotated key points on images by 1 to 5 pixels while fixing the real-world three-dimensional points. For each shift group, we randomly shift annotation points in the up, down, left and right directions 5 times. So, there is a total of 25 sets for the input. For each group, we take the average value for the RMSE, MAE and MAPE. In Sac, since the width and height of the images are doubled compared to those in HK, we random shift the annotation points in images for 2, 4, 6, 8 and 10 pixels (double size of 1 to 5 pixels) while fixing the real-world three-dimensional points for fairness. Other settings are the same as HK.

Table 10 The comparison of RMSE, MAE and MAPE of surveillance camera calibrated by different methods in HK and Sac (unit for RMSE and MAE: meter)

HK				Sac			
Pixel shifts	RMSE	MAE	MAPE	Pixel shifts	RMSE	MAE	MAPE
±0	**0.49**	**0.43**	**7.22%**	±0	1.28	1.05	8.62%
±1	0.98	0.84	14.15%	±2	**1.12**	**0.93**	**7.70%**
±2	1.15	1.04	17.36%	±4	3.42	3.22	26.42%
±3	1.48	1.35	22.61%	±6	5.98	5.83	47.86%
±4	2.38	2.28	38.16%	±8	7.01	6.92	56.78%
±5	2.90	2.84	47.37%	±10	7.05	6.97	57.22%

Bold indicates the best performance

All the results are shown in Table. 10. It is shown that, for the camera in HK, the RMSE, MAE and MAPE positively correlate with the number of pixel shifts. When there is no pixel shift, the RMSE, MAE and MAPE are minimal among all groups. Errors enlarge with the increase of pixel shifts. The calibration performance remains decent when the average shift is less than 4 (MAPE $< 25\%$). In contrast, the calibration error significantly increases and becomes unacceptable when the average shift is larger than 4 pixels. For the camera in Sac, the calibration error reaches the minimum when the pixel shift is 2. This may be because the original annotations of key points are not accurate. When the number of pixel shifts is greater than or equal to 4, the calibration results cannot be used for practical applications (MAPE $> 25\%$). It is reasonable to conjecture that the proposed MVCalib can sustain about 1% annotation error in the location of the key points.

Appendix B Convergence and Sensitivity of the MVCalib Algorithm

In this section, we discuss the convergence and sensitivity of the proposed MVCalib algorithm.

B.1 Convergence

The MVCalib algorithm consists of a series of optimization problems that are solved by different algorithms. In the candidate generation stage, the EPNP algorithm [27] can approximately solve the problem through a bundle of systems of equations. In the vehicle model matching and parameter fine-tuning stage, two non-linear optimization problems are

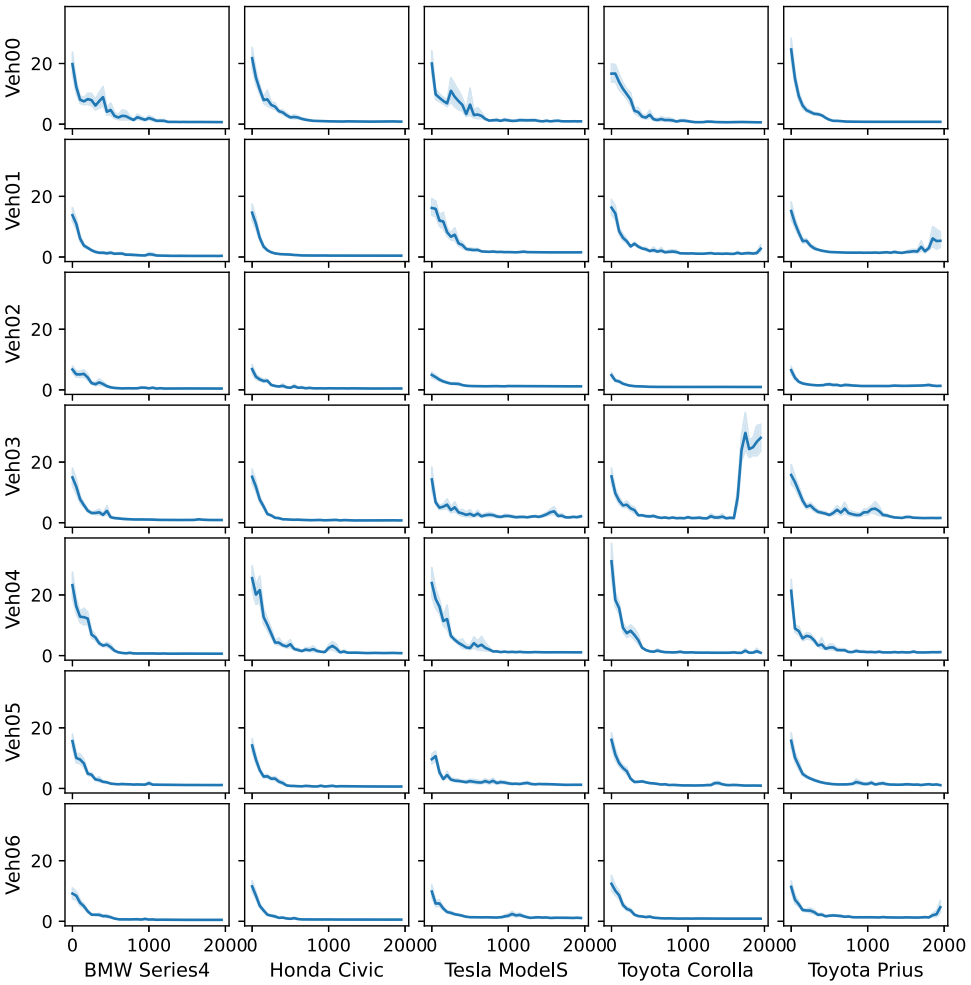


Fig. 14 The calibration losses in the vehicle model matching stage for the camera in HK. The x-axis is the iteration of optimization, while the y-axis means the loss

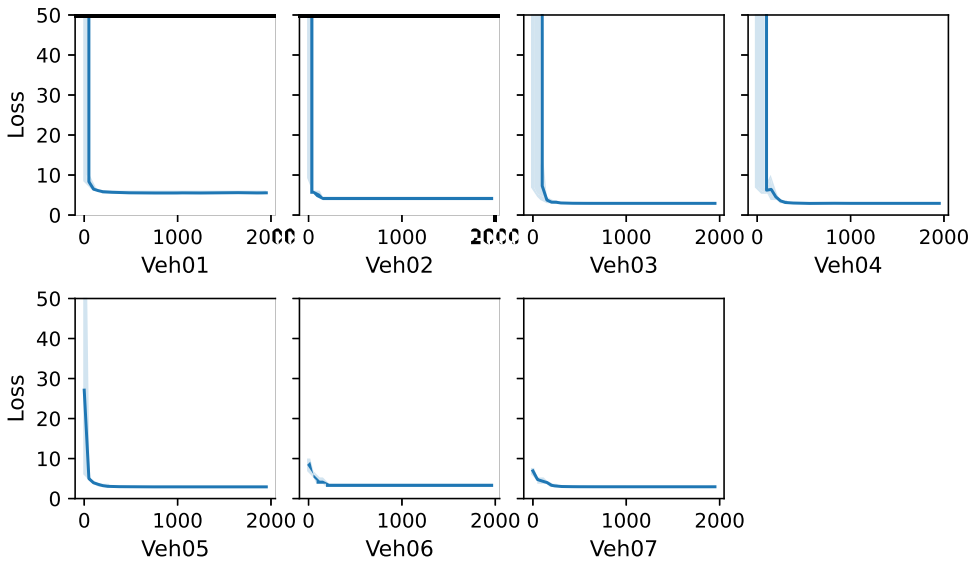


Fig. 15 The calibration losses for the camera in the parameter fine-tuning stage in HK. The x-axis is the iteration of optimization, while the y-axis means the loss

Table 11 The sensitivity of initial parameters in HK and Sac (unit for RMSE and MAE: meter, VMM: vehicle model matching, PF: parameter fine-tuning)

Group	Change	HK			Sac		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
Only VMM	0%	0.49	0.43	7.22%	1.28	1.05	8.62%
	10%	0.49	0.43	7.22%	1.33	1.23	10.13%
	20%	0.67	0.59	9.96%	1.72	1.66	13.68%
	30%	1.07	0.95	15.88%	2.34	2.27	18.64%
Only PF	0%	0.49	0.43	7.22%	1.28	1.05	8.62%
	10%	0.49	0.43	7.22%	0.99	0.86	7.12%
	20%	1.01	0.85	14.28%	1.40	1.17	9.61%
	30%	1.29	1.10	18.38%	1.29	1.03	8.47%
VMM + PF	0%	0.49	0.43	7.22%	1.28	1.05	8.62%
	10%	0.50	0.44	7.49%	1.31	1.18	9.75%
	20%	0.63	0.55	9.32%	1.88	1.82	14.93%
	30%	1.91	1.65	27.56%	5.28	4.98	40.90%

formulated, respectively, and there is no guarantee to find the global optimal. Since the gradients of variables are difficult to formulate in these optimization problems in the vehicle model matching and parameter fine-tuning stage, we consider a non-gradient-based optimization algorithm, named CMA-ES to solve these problems.

The calibration losses for the camera in HK in the vehicle model matching and parameter fine-tuning stage are shown in Figs. 14 and 15. It is clear to see that most losses with different combinations of vehicles and vehicle models converge in the vehicle model matching stage, and all losses converge to the local minimum in the parameter fine-tuning stage. The patterns of calibration losses in Sac for both vehicle model matching and parameter fine-tuning stages are similar to those in HK, so we do not discuss them respectively.

B.2 Sensitivity

The robustness of the MVCalib algorithm is important in real-world applications. In this section, we discuss the impact of initial parameters on the MVCalib algorithm. The MVCalib algorithm consists of three-stage calibrations. In the candidate generation stage, the problem is not solved by an optimization algorithm. Hence there is no initial parameter in this stage. In the vehicle model matching and parameter fine-tuning stage, problems are solved by two optimization algorithms. The initial parameters may individually or mutually affect the calibration results. Hence, we set up three groups of experiments to figure out the influence of initial parameters.

In the first group, we only change the initial parameters within the scale of 10%, 20% and 30% in the vehicle model matching stage. We repeat the process 5 times. In the second group, we only change the initial parameters in the parameter fine-tuning stage within the same scale 5 times. In the third group, both initial parameters in the vehicle model matching and parameter fine-tuning stages are changed 5 times within the same scales. Table. 11 shows the average results within the same scale in each group. It can be seen that, when the change scale is less than 20%, the calibration results do not degrade significantly in most groups in HK and Sac. When the change scale is over 30%, RMSE, MAE and MAPE are nearly doubled in most groups in HK and Sac, meaning that the camera calibrations results are not reliable. Therefore, the MVCalib algorithm can sustain the vibration of initial parameters within 20% of the original scale.

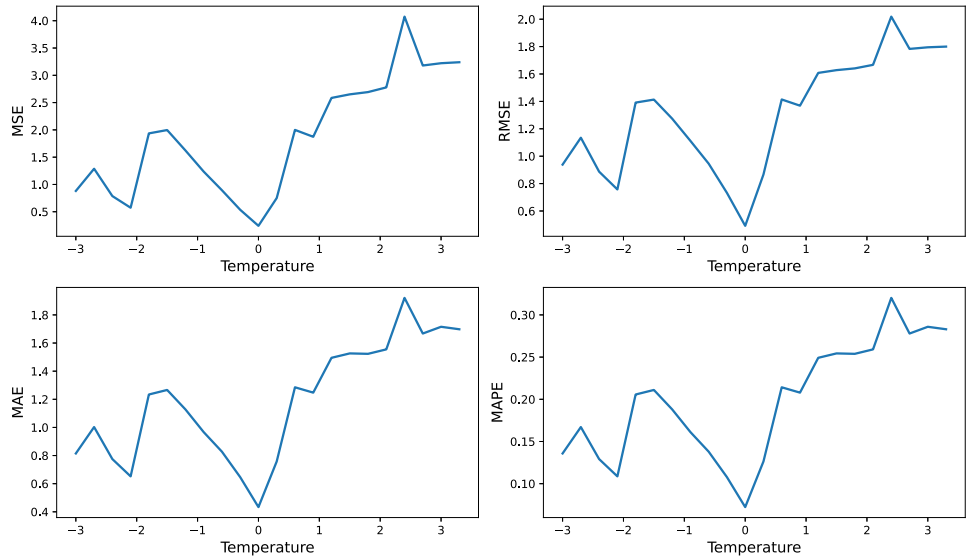
Appendix C The choice of τ in the MVCalib Algorithm

The choice of τ in Eq. 9 is important to the accuracy of the MVCalib algorithm. To find out the relationship between τ and calibration results, we set up the experiments as follows. For both datasets in Sac and HK, the value of *tau* ranges from -3.0 to 3.3 with a step size of 0.3. The calibration results are compared under different values of τ in HK and Sac.

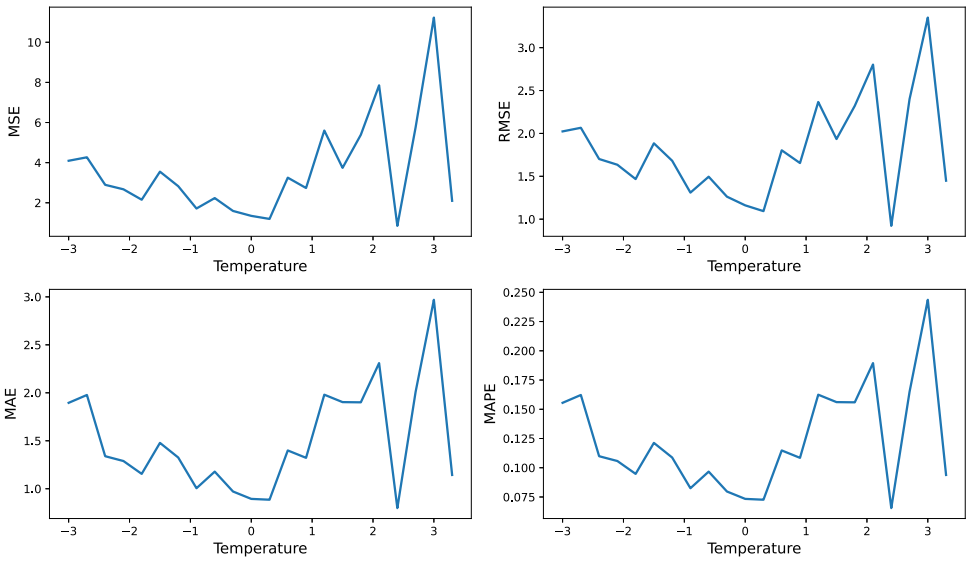
In Fig. 16, the values of τ are 0 and 2.4 in HK and Sac when the calibration losses reach the minimum. Though the value of τ should be set differently in different scenarios, when $\tau = 0$, we can get the best result in HK and the third-best result in Sac. Hence the value of τ is set to 0 in this study.

Appendix D The Interpretation of Metrics of Vehicle Detection

The metrics for evaluating the accuracy of vehicle detection models include precision, recall, PR-curve, mAP@0.5, and mAP@0.5:0.95. These metrics are commonly used to evaluate the quality of object detection models in CV. Before



(a) The calibration losses with different τ in HK



(b) The calibration losses with different τ in Sac .

Fig. 16 The calibration losses with different τ in HK and Sac

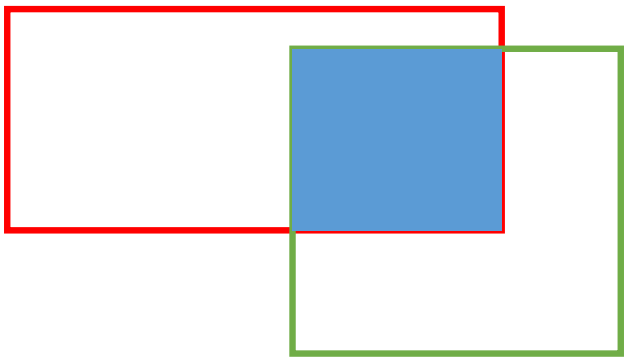
introducing the concept of the above metrics, there is a prerequisite metric called intersection over union (IoU), which defines the gaps between the estimated objection location and the ground truth. The outputs of the detection model are twofold. One is four corner coordinates that locates the object position in the image. The other is the confidence probability of the belonging category. If we overlap the estimated and the ground true bounding boxes, there will be an area of intersection (shown in Fig. 17a) and an area of union (shown in Fig. 17b), where the red and green rectangle means the estimated and ground true bounding boxes of an object, and the blue rectangle shows the intersection and union area, respectively. The intersection over union is defined as the quotient of the intersection area over the union area.

A threshold for IoU is set to decide if the bounding box is real or fake. If the IoU exceeds the threshold, we label it as True Positive (TP). Moreover, we can divide all circumstances into three categories, True Positive (TP), False Positive (FP), and False Negative (FN). The illustration about these circumstances is shown in Table 12.

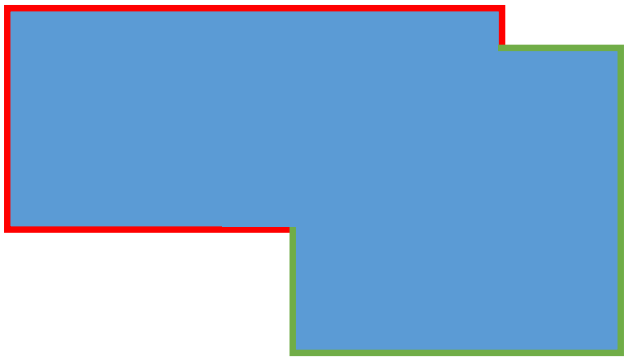
Additionally, the precision and recall can be calculated as

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned}, \tag{D1}$$

The precision and recall are a a pair of contradictory metric. When the precision is high, the recall is relative low,



(a) Area of intersection.



(b) Area of union.

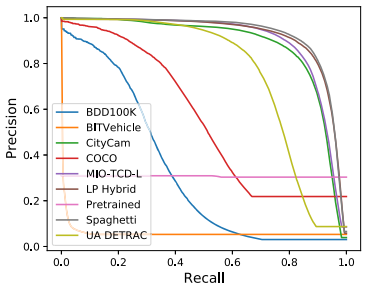
Fig. 17 Illustration of the intersection over union (IoU)

Table 12 Illustration of TP, FP, and FN

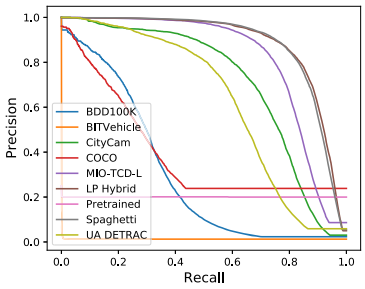
Categories	Comments
True Positive (TP)	The IoU between predicted and ground truth exceeds the threshold
False Positive (FP)	1. The IoU between predicted and ground truth is smaller than the threshold. 2. Estimated bounding boxes not overlapping with any ground true bounding boxes
False Negative (FN)	The object is not detected by the algorithm

vice versa. If we rank all the detection results according to the confidence probability, set different thresholds for confidence probability and re-calculate the precision and recall, a precision-recall (PR) curve can be plotted where the x-axis is the recall and the y-axis is the precision. An example is shown in Fig. 18. The Precision-Recall curves of different vehicle detection models are shown in Fig. 18.

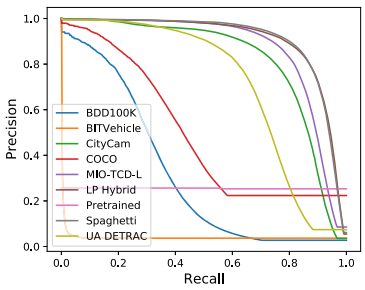
With the increasing of confidence threshold, the recall enlarges while the precision reduces. If the curve is close to the upper right corner of the figure, the performance of the model is good. Hence, it can be seen that the detection models trained with Spaghetti and LP hybrid datasets outperform other models trained with sole datasets. In particular, if we compare the curve between models trained with Spaghetti and LP hybrid datasets, the differences between these two models are marginal.



(a) The PR curve for images at daytime.



(b) The PR curve for images at nighttime.



(c) The PR curve for images throughout day and night.

Fig. 18 The PR curve for camera images on the testing set

The Average Precision (AP) is the area that below the PR curve, calculated as

$$AP = \int_0^1 PR(r)dr, \tag{D2}$$

where r is the recall and $PR(r)$ is the precision. The mAP@0.5 means the AP value when the IoU threshold is 0.5. Besides, the mAP@0.5:0.95 means the average of AP when the IoU threshold equals to 0.5, 0.55, 0.9, $\cdots$, 0.9, 0.95 separately. These two metrics are extensively used to evaluate

the performance of algorithms in object detection tasks in CV.

References

1. Smaragdis E, Papageorgiou M, Kosmatopoulos E (2004) A flow-maximizing adaptive local ramp metering strategy. Transp Res Part B: Methodolo 38(3):251–270. https://doi.org/10.1016/S0191-2615(03)00012-2
2. Wuthishuwong C, Traechtler A (2020) Distributed control system architecture for balancing and stabilizing traffic in the network of multiple autonomous intersections using feedback consensus and route assignment method. Complex Intell Syst 6(1):165–187. https://doi.org/10.1007/s40747-019-00125-3
3. Bodvarsson GA, Muench ST Effects of loop detector installation on the portland cement concrete pavement lifespan : case study on I-5. Technical Report dot:22405, Washington State Transportation Center
4. Panichpapiboon S, Pattara-atikom W (2008) Evaluation of a neighbor-based vehicle density estimation scheme. In: 2008 8th International Conference on ITS Telecommunications, pp. 294–298. https://doi.org/10.1109/ITST.2008.4740274
5. Zhu J, Sun K, Jia S, Li Q, Hou X, Lin W, Liu B, Qiu G (2018) Urban traffic density estimation based on ultrahigh-resolution uav video and deep neural network. IEEE J Select Top Appl Earth Observ Remote Sensing 11(12):4968–4981. https://doi.org/10.1109/JSTARS.2018.2879368
6. Gerfen J, Hockaday N, et al. (2009) Caltrans TMC coordination. Technical report
7. Wan Y, Huang Y, Buckles B (2014) Camera calibration and vehicle tracking: highway traffic video analytics. Transport Res C 44:202–213. https://doi.org/10.1016/j.trc.2014.02.018
8. Zhang B, Zhang J (2020) A traffic surveillance system for obtaining comprehensive information of the passing vehicles based on instance segmentation. IEEE Trans Intell Transport Syst. https://doi.org/10.1109/TITS.2020.3001154
9. Zapletal D, Herout A (2016) Vehicle Re-identification for automatic video traffic surveillance. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1568–1574 . https://doi.org/10.1109/CVPRW.2016.195
10. Xiong Z, Li M, Ma Y, Wu X (2020) Vehicle re-identification with image processing and car-following model using multiple surveillance cameras from urban arterials. IEEE Trans Intell Transport Syst. https://doi.org/10.1109/TITS.2020.3006047
11. Suryakala S, Muthumeenakshi K, Gladwin SJ (2019) Vision based vehicle/pedestrian detection in traffic surveillance system. In: 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0506–0510 . https://doi.org/10.1109/ICCSP.2019.8697954
12. Sipetas C, Keklikoglou A, Gonzales EJ (2020) Estimation of left behind subway passengers through archived data and video image processing. Transport Res C 118:102727. https://doi.org/10.1016/j.trc.2020.102727
13. Darwish T, Abu Bakar K (2015) Traffic density estimation in vehicular ad hoc networks: a review. Ad Hoc Netw 24:337–351. https://doi.org/10.1016/j.adhoc.2014.09.007
14. Ozkurt C, Camci F (2009) Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. Math Comput Appl 14(3):187–196
15. Wu Z, Lam WHK (2010) Using online CCTV image sequences for real-time traffic estimation. In: Proceeding of the 89th Annual Meeting of the Transportation Research Board (TRB)
16. Zhang S, Wu G, Costeira JP, Moura JMF (2017) Understanding traffic density from large-scale web camera data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
17. Biswas D, Su H, Wang C, Stevanovic A, Wang W (2019) An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD. Physics and Chemistry of the Earth, Parts A/B/C **110**, 176–184 . https://doi.org/10.1016/j.pce.2018.12.001. Sensing and Sensor Systems for Urban Environmental Studies
18. Jain V, Dhananjay A, Sharma A, Subramanian L (2012) Traffic density estimation from highly noise image sources. In: Transportation Research Board Annual Summit,
19. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330–1334. https://doi.org/10.1109/34.888718
20. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)
21. Song K-T, Tai J-C (2006) Dynamic calibration of pan-tilt-zoom cameras for traffic monitoring. IEEE Trans Syst Man Cybern B (Cybern) 36(5):1091–1103. https://doi.org/10.1109/TSMCB.2006.872271
22. Dubska M, Herout A, Sochor J (2014) Automatic camera calibration for traffic understanding. In: Proceedings of the British Machine Vision Conference. BMVA Press. https://doi.org/10.5244/C.28.42
23. Sochor J, Juránek R, Herout A (2017) Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement. Comput Vis Image Underst 161:87–98. https://doi.org/10.1016/j.cviu.2017.05.015
24. Tang X, Wang W, Song H, Zhao C (2023) Centerloc3d: monocular 3d vehicle localization network for roadside surveillance cameras. Complex Intell Syst. https://doi.org/10.1007/s40747-022-00962-9
25. Haralick RM, Lee D, Ottenburg K, Nolle M (1991) Analysis and solutions of the three point perspective pose estimation problem. In: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 592–598. https://doi.org/10.1109/CVPR.1991.139759
26. Quan L, Lan Z (1999) Linear N-point camera pose determination. IEEE Trans Pattern Anal Mach Intell 21(8):774–780. https://doi.org/10.1109/34.784291
27. Lepetit V, Moreno-Noguer F, Fua P (2008) EPnP: an accurate an solution to the PnP problem. Int J Comput Vision 81(2):155. https://doi.org/10.1007/s11263-008-0152-6
28. Hesch JA, Roumeliotis SI (2011) A Direct Least-Squares (DLS) method for PnP. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 383–390 . https://doi.org/10.1109/ICCV.2011.6126266
29. Li S, Xu C, Xie M (2012) A robust O(n) solution to the perspective-n-point problem. IEEE Trans Pattern Anal Mach Intell 34(7):1444–1450. https://doi.org/10.1109/TPAMI.2012.41
30. Bhardwaj R, Tummala GK, Ramalingam G, Ramjee R, Sinha P (2018) Autocalib: automatic traffic camera calibration at scale. ACM Trans Sen Netw. https://doi.org/10.1145/3199667
31. Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F (2013) Exhaustive linearization for robust camera pose and focal length estimation. IEEE Trans Pattern Anal Mach Intell 35(10):2387–2400. https://doi.org/10.1109/TPAMI.2013.36
32. Zheng Y, Sugimoto S, Sato I, Okutomi M (2014) A general and simple method for camera pose and focal length determination. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 430–437. https://doi.org/10.1109/CVPR.2014.62
33. Wu C (2015) P3.5P: Pose estimation with unknown focal length. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2440–2448. https://doi.org/10.1109/CVPR.2015.7298858

34. Zheng Y, Kneip L (2016) A direct least-squares solution to the pnp problem with unknown focal length. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1790–1798 . https://doi.org/10.1109/CVPR.2016.198
35. Bartl V, Špaňhel J, Dobeš P, Juránek R, Herout A (2020) Automatic camera calibration by landmarks on rigid objects. Mach Vis Appl 32(1):2. https://doi.org/10.1007/s00138-020-01125-x
36. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
37. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)
38. Lin T-Y, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)
39. Bautista CM, Dy CA, Mañalac MI, Orbe RA, Cordel M (2016) Convolutional neural network for vehicle detection in low resolution traffic videos. In: 2016 IEEE Region 10 Symposium (TENSYMP), pp. 277–281. https://doi.org/10.1109/TENCONSpring.2016.7519418
40. Yeshwanth C, Sooraj PSA, Sudhakaran V, Raveendran V (2017) Estimation of intersection traffic density on decentralized architectures with deep networks. In: 2017 International Smart Cities Conference (ISC2), pp. 1–6. https://doi.org/10.1109/ISC2.2017.8090799
41. Zhang S, Wu G, Costeira JP, Moura JMF (2017) FCN-rLSTM: Deep Spatio-Temporal neural networks for vehicle counting in city cameras. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)
42. Shen L, Tao H, Ni Y, Wang Y, Stojanovic V (2023) Improved yolov3 model with feature map cropping for multi-scale road object detection. Meas Sci Technol 34(4):045406. https://doi.org/10.1088/1361-6501/acb075
43. Tao H, Cheng L, Qiu J, Stojanovic V (2022) Few shot cross equipment fault diagnosis method based on parameter optimization and feature mertic. Meas Sci Technol 33(11):115005. https://doi.org/10.1088/1361-6501/ac8368
44. Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395. https://doi.org/10.1145/358669.358692
45. Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 312–317. https://doi.org/10.1109/ICEC.1996.542381
46. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359. https://doi.org/10.1109/TKDE.2009.191
47. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision - ECCV 2014. Springer, Cham, pp 740–755
48. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848
49. Yu F, Chen H, Wang X, Xian W, Chen Y, Liu F, Madhavan V, Darrell T (2020) BDD100K: A diverse driving dataset for heterogeneous multitask learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
50. Lyu S, Chang M-C, Du D, Li W, Wei Y, Coco MD, Carcagnì P, Schumann A, Munjal B, Dang D-Q-T, Choi D-H, Bochinski E, Galasso F, Bunyak F, Seetharaman G, Baek J-W, Lee JT, Palaniappan K, Lim K-T, Moon K, Kim K-J, Sommer L, Brandlmaier M, Kang M-S, Jeon M, Al-Shakarji NM, Acatay O, Kim P-K, Amin S, Sikora T., Dinh T, Senst T, Che V-G-H, Lim Y-C, Song Y-m, Chung Y-S (2018) UA-DETRAC 2018: Report of AVSS2018; IWT4S challenge on advanced traffic monitoring. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. https://doi.org/10.1109/AVSS.2018.8639089
51. Dong Z, Wu Y, Pei M, Jia Y (2015) Vehicle type classification using a semisupervised convolutional neural network. IEEE Trans Intell Transport Syst 16(4):2247–2256. https://doi.org/10.1109/TITS.2015.2402438
52. Luo Z, Branchaud-Charron F, Lemaire C, Konrad J, Li S, Mishra A, Achkar A, Eichel J, Jodoin P-M (2018) MIO-TCD: a new benchmark dataset for vehicle classification and localization. IEEE Trans Image Process 27(10):5129–5141. https://doi.org/10.1109/TIP.2018.2848705
53. Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomammana, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, J., Fang, J., Yu, L., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, Rai, P.: Ultralytics/yolov5: V5.0 - YOLOv5-P6 1280 Models, AWS, Supervise.ly and YouTube Integrations. https://doi.org/10.5281/zenodo.4154370
54. Itseez: Open Source Computer Vision Library. https://github.com/itseez/opencv (2015)
55. Rapin J, Teytaud O (2018) Nevergrad - A gradient-free optimization platform. GitHub
56. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings
57. (2010) Federal Highway Administration: Manual on Uniform Traffic Control Devices 2009 Edition,