Deep Learning Serves Traffic Safety: A Forward-looking Review

Abolfazl Razı1,*,†, Xiwen Chen1,†, Huayu Li2, Brendan Russo3, Yan Chen1, Hongbin Yu5

1 School of Computing, Clemson University, SC 29634
2 Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
3 Department of Civil Engineering, Northern Arizona University, Flagstaff, AZ 86011
4 The Polytechnic School, Arizona State University, Mesa, AZ 85212
5 School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287

Abstract: This paper explores Deep Learning (DL) methods that are used or have the potential to be used for traffic video analysis, emphasizing driving safety for both Autonomous Vehicles (AVs) and human-operated vehicles. We present a typical processing pipeline, which can be used to understand and interpret traffic videos by extracting operational safety metrics and providing general hints and guidelines to improve traffic safety. This processing framework includes several steps, including video enhancement, video stabilization, semantic and incident segmentation, object detection and classification, trajectory extraction, speed estimation, event analysis, modeling and anomaly detection. Our main goal is to guide traffic analysts to develop their own custom-built processing frameworks by selecting the best choices for each step and offering new designs for the lacking modules by providing a comparative analysis of the most successful conventional and DL-based algorithms proposed for each step. We also review existing open-source tools and public datasets that can help train DL models. To be more specific, we review exemplary traffic problems and mentioned steps for each problem. Besides, we investigate connections to the closely related research areas of drivers’ cognition evaluation, Crowd-sourcing-based monitoring systems, Edge Computing in roadside infrastructures, ADS-equipped AVs, and highlight the missing gaps. Finally, we review commercial implementations of traffic monitoring systems, their future outlook, and open problems and remaining challenges for widespread use of such systems.

1 Introduction

Despite recent advances in vehicles’ operational safety features, computer-assisted control, and technology-based traffic management systems, traffic safety remains one of the main challenges in today’s life. Every year, traffic crashes account for 20-50 million causalities and 1.35 million fatalities worldwide, making it one of the top-10 causes of death. Indeed, traffic crashes is the leading cause of death for people aged 5 to 29[1].

Using computational intelligence and computer tools for enhancing traffic safety has gained a lot of attention in recent years. Mainstream technological trends include (i) implementing vehicle safety features such as forward collision warning, blind-spot detection, lane departure warning, backup camera, and autonomous emergency braking[2], (ii) simulation-based road infrastructure design such as Site3D, RoadEng[3], and OpenRoads Designer[4], and (iii) intelligent traffic flow management systems such as GPS-based navigation tools. An example of the last category is Google’s road-user interpretive software that can infer the common road behavior of other drivers that allows Engine Control Units (ECUs) to make better route decisions[5].

The power of Artificial Intelligence (AI) has been utilized by many car manufacturers to develop safety features[3]. The recent advances in DL methods for video processing backed by low-cost and high-speed computational platforms such as Graphics/Tensor Processing Units (GPUs/TPUs) have accelerated the pace of developing AI-based features both at vehicle and infrastructure levels[6].

1.1 Architectural View

Traffic Safety analysis can be viewed as a modular and multi-faceted problem that involves many aspects. As shown in Figure 2, the overall analysis platform can be viewed as a software pipeline where the collected information undergoes different processing steps until it is translated to navigation commands, advisory messages, or overall guidelines for improving traffic safety.

This paper provides a comprehensive review of the popular methods, tools, software packages, and datasets developed by the scientific community for each sub-problem. We also highlight the open problems and future challenges on each frontier. Our main emphasis is on the role of DL methods in enhancing traffic flow (e.g., improving efficiency) and mitigating traffic safety risks.

In contrast to the primary trend of processing individual events from an involved vehicle’s perspective, we consider traffic safety at both the vehicle-level and network-level by processing videos captured by an external observer. The main information source is captured video by vehicle-mounted cameras and roadside cameras, but we also will review other sensor information that can be used for traffic management. We believe that enabling advanced traffic safety analysis and monitoring platforms will play a crucial role in future smart cities.

It is noteworthy that several review papers have been published to review methods and tools used for video-based traffic analysis. A summary of these papers is provided in Table 1. However, most review papers have limitations in certain aspects.

Some survey papers (e.g.,[7,12]), focused merely on solving traffic-related tasks (such as perception) while not covering safety assessment methods. Some other papers (e.g.,[7,12]) do not provide a comprehensive summary of DL methods, which recently
Fig. 1: A framework of Computer Vision (CV)-based traffic safety analysis pipeline.

Fig. 2: The data sources and computing resources used in video-based traffic analysis.
Table 1 Summary of related review papers. The paper with * means although this paper is related but outdated. $L_\sqrt{}$ denotes a topic is covered in fewer details.

| Paper          | Year | Human-driven Vehicle | AVs | Safety Assessment Analysis | CV-based Method | Deep Learning Method | Sensors | Datasets | Network Analysis | Vehicular Edge Computing | Behavioral & Driver Cognition |
|----------------|------|----------------------|-----|---------------------------|----------------|---------------------|--------|----------|------------------|--------------------------|-----------------------------|
| [7] Hu et al.  | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [8] Mozaffari et al. | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [9] Grigorescu et al. | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [10] Yurtsever et al. | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [11] Janai et al. | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [12] Badue et al. | 2019 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [13] Kumaran et al. | 2019 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [14] Wang et al. | 2018 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [15] Nguyen et al. | 2016 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [16] Shirazi et al. | 2015 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [17] Mukhtar et al. | 2013 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| [18] Morris et al. | 2012 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |
| Ours           | 2020 | √                    | √   | √                         | √              | √                   | √      | √        | √                | √                        | *                           |

Table 2 Comparison of video-processing for traffic analysis from two standpoints of (i) an external observer’s and (ii) an internal node’s perspectives.

| Perspective | Advantages | Disadvantages | Scenarios |
|-------------|------------|---------------|-----------|
| External observers | Easy to obtain; Easy to measure traditional metrics; Can consider overall information for an event | Labor cost; Observer variation; No input from driver’s cognition; No insight into component-level data in AVs | Site-based and network-based observation analysis |
| Internal nodes | Can both obtain driver behaviors and external conditions; In-depth understanding of the interactions between involved objects | Small data size; Time-consuming; privacy problem | Naturalistic driving studies; AV studies |

Fig. 3: The example of Perspectives: (a) Internal Perspective and (b) External Perspective.

has become the dominant approach in both industry and academia research. Other papers (e.g., [10][12]), which review driving techniques for vehicles equipped with automated driving systems (ADS), keep their attention solely on the DL methods developed for AVs while not investigating the practicality of these methods on human-driven vehicles, which still are the most widely used vehicles. A few papers (e.g., [7][13]) put their primary focus on DL methods but without special emphasis on vision-related tasks particularly useful for traffic analysis. A different set of papers (e.g., [8]) limit their analysis to the perspective of internal nodes, also known as the first-person perspective. Although CV-based tasks are covered by several papers [14, 15], they do not provide sufficient details on this subject from different perspectives. Most of the papers, including [9][11][13][15], although very informative, only review the DL algorithms that are used or can be used for traffic analysis and do not provide any sort of comparative analysis, which does not help choosing the right method for different real-world traffic problems. The relevant surveys we include are from 2013 and after; however, we want to mention some papers, including [16][17] are outdated, but highly related to our survey with the topics of CV-based traffic safety analysis. Table 3 provides more specific details on the main focus and the shortcoming of the recently published survey papers.

In addition to covering newly published methods, our paper covers the shortcomings of previous surveys and considers the driving safety problem from different perspectives. More specifically, we list exemplary problems in driving safety analysis; we review requirements and challenges from an external observer’s perspective; we review data sets and important industrial developments; we make connections to closely related areas of utilizing crowd-sourcing, and edge and cloud computing for bulk processing; we highlight connections to behavioral science, insurance industry, and other policy maker entities.

The rest of this paper is organized as follows. In Section 2 information acquisition equipment and methods are reviewed. A list of sample problems in driving safety analysis is provided in Section 3. Section 4 includes discussions about different stages of video pre-processing for safety analysis by highlighting historical milestones, successful methods, current trends, and remaining challenges for each category. Section 5 reviews DL methods for video processing with application to traffic safety analysis. A list of commonly used datasets with applications to traffic monitoring and traffic safety analysis is provided in Section 6. Section 7 list a set of key safety metrics used for assessing the potential for crash occurrence and crash severity. Finally, Section 8 provides different key points such as remaining challenges and issues, potential applications for safety analysis methods, the future outlook of safety monitoring systems, and connections to AVs and insurance evaluations.
2 Data Acquisition

In this section, we investigate the role of data acquisition in developing safety-related algorithms. First, we review data modalities and hardware used for information acquisition. Next, we list the most commonly used datasets for testing the developed algorithms.

2.1 Sensor and Data Acquisition Equipment

Most traffic analysis platforms rely on data collected by different types of sensors, including cameras, Global Positioning Systems (GPS), Radio Detection And Ranging (Radar), and Light Detection And Ranging (LiDAR). These sensors (except GPS) can be used in vehicles or on external observer systems such as roadside infrastructures, drone-based aerial platforms, etc. The following is a short description of sensors followed by a summary provided in Table 1.

Video Cameras are the most widely used means of information collection. Modern cars are heavily equipped with cameras in various parts to capture imagery for processing. Using thermographic cameras for night vision has become more common than ever. Camera feeds can be used by the driver (e.g., backup camera) to minimize safety risks, or by the control computer in an AV for automated driving. The imagery can also be used by more advanced AI platforms for driver’s cognition assessment in real-time mode (e.g., driver drowsiness assessment [19], and distraction awareness [20]). Volume collections of imagery can be stored for further analysis for various purposes, including vehicle detection, perception on AVs, etc., without sufficient detail. A traditional radar is a single source-detector, which does not have the spatial resolution required for precise scene/environment description; however, in recent years, imaging radars have been developed by adopting multiple-input, multiple-output (MIMO), and radar-on-chip technologies. Although such advanced technology is still expensive, attempts are made to reduce the cost of broader adoption [24].

LiDAR uses laser beam reflection to enable accurate positioning down to centimeter’s scale [25]. The 3D scanning of multiple laser beams provides a 3D point cloud image (3D map) of the surrounding obstacles with accuracy much higher than regular radars [26]. LiDARs send out a near-infrared laser beam and detect reflections from the object; thus, it can still operate in dark conditions, in contrast to visual sensors. Its use is less common than radars for a few reasons, including its higher cost, relatively sparse spatial resolution, especially for a limited number of scanning laser beams, extremely narrow FoV, and computational complexity of 360° scanning, noting that low-complexity point cloud methods are still under development. Solid-state LiDARs have been developed to reduce the cost, while enhancing the spatial resolution [27].

Drones are commonly used nowadays to implement aerial monitoring systems. Most external observer systems utilize sensors in roadside infrastructures. However, the use of drones, also known as Unmanned Aerial Vehicles (UAVs), is gaining more attention in different applications to enable fast, low-cost, and on-demand monitoring [28], and traffic analysis is not an exception. Particularly, drones can provide top-view and clearer occlusion-free images of the traffic flow when needed [29]. A network of cooperative drones can collectively cover relatively large areas [30]. Modern drones may be equipped with advanced sensing platforms, high-resolution cameras, and more importantly advanced features such as learning-based image processing, AI-based autonomous control, collision avoidance and auto landing, auto-calibration, real-time transmission, object tracking, and image restoration. The key challenges of drones are their limited payload, flight time, and communication range under study by several research teams. Aerial images pose new challenges to the research community, such as tackling image stabilization, small object recognition, and developing lightweight ML algorithms customized for top-view images.

Table 1

| Survey        | Content                                                                 | Drawback                                                                 | Perspective   | Application                      |
|---------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------|---------------|----------------------------------|
| [12] 2020     | Reviews DL approaches for Trajectory prediction based on the input representation, output type, and prediction method. | shallowly covers sensor information; less details are provided for performance comparison (only covers 10 detection methods on a general dataset and 4 semantic segmentation methods on CityScapes).  | Internal       | AVs; traffic analysis            |
| [13] 2020     | Reviews DL approaches for AVs, discusses open problems and current challenges. | Excludes video pre-processing methods (e.g. Video stabilization); provides little details on the performance analysis (9 methods on ImageNet and 7 methods on KITTI 3D).  | Internal       | AVs                             |
| [14] 2020     | Reviews CV-based approaches for AVs, discusses the AV industry milestones. | Covers limited DL methods (sign and light detection, pavement marking detection, MOT, excluding segmentation methods); does not give detailed performance comparisons. | Internal       | AVs                             |
| [15] 2020     | Reviews DL approaches for multiple traffic applications, including time series prediction, classification, and optimization. | Only includes tasks related to anomaly detection; does not include other CV tasks; does not provide comprehensive performance analysis. | External       | Traffic analysis                 |
| [16] 2018     | Reviews DL approaches for processing traffic data. | Covers a few CV-based problems (including vehicle detection, perception on AVs, etc.,) without sufficient detail. | External       | Traffic analysis                 |
vehicle count, speed, and flow volume, noting that Piezoelectric sensors are still used for weigh-in-motion measurement. Environmental Sensor Stations (ESS)\cite{34} on the roadway can collect atmospheric data, including air temperature and humidity, visibility distance, wind speed, wind direction, etc., as side parameters to be used for traffic analysis.

3 Sample Problems

Safety assessment is a qualitative process, which can be approached by a set of specific and objective sub-problems. For instance, the overall safety of a highway section can be assessed by a set of exemplary problems provided in Table\ref{table5}. Cloud-based software can process videos captured by the roadside infrastructure to extract statistical information from the observed events. The results of such analyses can be used to evaluate the highway’s safety profile and offer revisions to the traffic management guidelines to the transportation personnel. For instance, frequent roadblocks in specific sections of a highway may require widening the road, revisiting speed limits, prohibiting commercial vehicles, or planning traffic re-routing.

Table\ref{table5} represents typical processing steps required for each sample problem. As we discussed in this paper, some technical papers offer a solution for some problems, while more work is required to solve some other problems. Also, it is worth mentioning that some steps are necessary for each problem, while some other pre-processing steps such as denoising and video stabilization may or may not be included to balance between the accuracy and complexity.

In the next section, we provide alternative processing pipelines for zigzag driving as an exemplary application, along with a comparative analysis on the accuracy and complexity of each method.

3.1 Sample Task: Zigzag driving

Zigzag driving on highways is a single-vehicle-involved event defined as frequent shifts to left and right in a short period of time $t_z$. Figs.\ref{fig4}(c)-(e) show the three bases of zigzag driving during the short period $t_z$, in comparison with the normal driving (including the normal lane change shown in Fig.\ref{fig4}(a), and the normal straight driving shown in Fig.\ref{fig4}(b)). These zigzag driving incidents include i) zigzag driving only once without lane change (Fig.\ref{fig4}(c)), ii) zigzag driving once with lane change (Fig.\ref{fig4}(d)), and iii) zigzag driving once with multiple lane changes (Fig.\ref{fig4}(e)). The other cases of zigzag driving shown in Figs.\ref{fig4}(f)-(h) can be decomposed into three base cases. To solve such an easy problem, one may take different approaches, including:

1. MOT-TP: This solution uses multi-object tracking (MOT) to extract cars’ motor tracks. The overall process is as follows: Several sub-sequences of each trajectory are generated by using a sliding window with size $t_s$ (in order to save processing time, the stride can be set greater than 1). Then a sub-window with size $L$ is used to extract possible peaks and trough points to be processed by Non-Maximum Suppression (NMS) to obtain the turning points. When the number of the turning points is greater than the threshold $n_z$, this sequence will be determined as zigzag driving. Theoretically, MOT-TP is able to detect all cases of zigzag driving, regardless of the road pattern and without the need for detecting road lanes. The potential drawback of this method is that normal driving on sharp road curves may be considered zigzag driving.

2. LD-Markings: First, the lanes are detected by instance segmentation. Then, similar to the MOT-TP method, the motion trajectories are extracted by MOT. The zigzag driving is determined by comparing the trajectory sub-sequences with the lane markings. The trajectory is determined as Zigzag Driving when the number of lane changes is greater than a threshold $n_z$, within a given time interval. The accuracy of this method varies based on the performance of the utilized algorithms for trajectory extraction and lane detection. It is not suitable for dirt and unlaned roads. It also misses zigzag driving with no lane changes, such as the examples in Figs.\ref{fig4}(f-h).

3. Trj-Cls: The Trajectory classification method skips modeling the road and the surrounding environment; instead, it directly applies supervised or unsupervised classification to the extracted trajectories to identify zigzag driving events. Similar to the LD-markings, the accuracy of this method depends on the accuracy of MOT used for trajectory extraction. It is flexible and can utilize different classification methods. However, this data-driven method requires a relatively large manually annotated dataset to maintain reasonably high classification accuracy.

4. HQ-sampling: This method can be viewed as a simplified version of LD-Markings. It uses high-resolution data to accurately identify the plate license numbers (by Optical Character Recognition (OCR)). This solution can track each vehicle by employing plate license numbers to replace the tracking algorithms. Instead of using video, this high-quality data often is in the form of images, to be captured continuously at intervals $t_0$, which may lead to inaccurate trajectory extraction. Similar to the LD-markings method, HQ-sampling determines zigzag driving by comparing the detected number of lane changes with the threshold $n_z$. Its accuracy depends on the performance of the OCR, the lane detection algorithm, and the utilized sampling interval $t_0$. The main shortcoming of this method is high complexity for short intervals, compromised accuracy for long intervals. This method may also undercount the lane changes if the sampling interval is selected large (e.g., it only counts zero lane changes in Fig.\ref{fig4}(h) with the three sample points).

![Fig. 4: (a)-(d) are the trajectories with normal driving. (a) shows the normal lane changing, and (b) shows the normal straight driving. (c)-(e) are the three bases of zigzag driving. (c) is zigzag driving once without lane changing, (d) is zigzag driving once with lane changing once. (g) is zigzag driving once with lane changing twice. (f)-(h) are some examples of more aggressive and frequent zigzag driving. The blue circles in (h) denote the sampling points by HQ-sampling with a big interval $t_0$.](image)

| Method            | Complexity of Inference | Sources of Complexity | Accuracy |
|-------------------|-------------------------|-----------------------|----------|
| MOT-TP            | M                       | MOT+SW                | H        |
| LD-markings       | L                       | LD+MOT                | M        |
| Trj-Cls           | H                       | MOT+ML                | H        |
| HQ-sampling       | M/H                     | LD+OCR                | L/M      |

Table 4 Comparison of the solutions to detect zigzag driving. SW denotes sliding window used to search the peak and trough points.
Table 5 compares these four solutions in terms of accuracy and computation complexity. Note that the comparison is based on the assumption that the MOT algorithm performs well and a powerful trajectory smoothing method is deployed. Additionally, roadside video cameras are often fixed, meaning that lane detection is performed only once per scene and does not substantially affect the computation load.

As seen, there exist many alternative solutions for such a simple problem. Therefore, a thorough understanding of methods can help the researchers design the most effective processing pipeline for the problem at hand. It is worth mentioning that other trajectory-based tasks often have similar processing steps of (VS)-(DN)-(LD)-MOT-TE-(smoothing)-C, as shown in Table 5. It means that some parts of processing pipelines can be shared among different problems, or transferable inference models can be used for multiple tasks to lower the cost.

4 Video and image pre-processing

In this section, we review different stages of a typical video-based traffic analysis framework and highlight key developments, historical milestones, current trends, and existing challenges.

**Super Resolution:** The video and image super-resolution aims to reconstruct a Higher Resolution (HR) result from a Low Resolution (LR) observation. Super-resolution is a typical stage in image pre-processing and can be applied to traffic imagery to enhance the performance of the subsequent learning tasks, such as vehicle classification and license plate detection. One popular supervised learning method is the DL-based Single Image Super-Resolution (SISR) method which creates a mapping between the low and high-resolution images by training a deep Convolutional Neural Network (CNN). Most of the existing learning-based SISR methods are trained and evaluated using simulated datasets [51,52], where the LR images are generated by applying a hand-crafted degradation process into the HR samples. For instance, one may apply bi-cubic down-sampling to the original HR samples to obtain LR results. Recently, more advanced SISR [53] methods are developed for real-world applications with unknown and more complicated degradation processes, which can be used as a benchmark method for traffic image analysis as well.

In contrast to the simple spatial interpolation used in the SISR family for image processing, Video Super-Resolution (VSR) methods utilize both spatial and temporal relationships between consecutive frames to improve the quality of the reconstructed videos [55]. These methods are essential in processing roadside traffic videos, especially under poor visibility in foggy, rainy, and cloudy weather conditions.

An important application of SR methods is license plate detection for vehicle identification. Early works often focused on conventional signal processing methods. For instance, [57,58] deployed a Markov random fields-based method for plate detection. [59] proposed a Gaussian Mixture Model (GMM) to enhance the plate location and SR reconstruction. Compressed Sensing (CS)-based methods [60,61] can also be used to address this task by enforcing the sparsity of images in the frequency domain, which is equivalent to smoothness in the spatial domain. Recently, DL-based SR algorithms are proposed, which perform more accurately and efficiently. [63] is an example of such methods which use a CNN architecture to convert a low-resolution license plate into a high-resolution version. Some recent works [64,65] tend to use Generative Adversarial Networks (GAN) as their processing framework, which achieves a higher performance using a more reasonable real-time loss along with an adversarial loss, when inferring.

**Denoising:** This is another critical pre-processing task to compensate for imaging artifacts and obtain clear and noise-free images before feeding them into the subsequent learning modules. This is a critical step in processing traffic imagery, especially when taken in motion or under low illumination and poor environmental conditions like rainy, cloudy, and foggy weather.

Conventional methods typically use filtering, interpolation, and smoothing methods either in time, frequency, or wavelet domains to remove noise from the captured images. In contrast, newer methods use more advanced concepts such as sparsity in the frequency domain, dictionary learning to model common noise patterns, prior knowledge about the noise model, and noise pattern discovery to more elegantly remove the noise from the captured images.

Traditional methods suffer from several shortcomings, including (1) involving complex optimization methods in some cases, (2) the need for manual parameter setting (e.g., the scale factor of Gaussian spatial filtering), (3) and using a fixed model which deems inflexible in tackling different noise patterns and ignores the learnability of some noises.

DL-based video and image denoising algorithms take advantage of the neural networks to learn the spatial or temporal dependency between pixels to reconstruct clean samples by end2end training and inferring. Therefore, DL methods provide sufficient flexibility in adapting to different conditions. In most research works on image denoising, a synthetic Additive white Gaussian Noise (AWGN) model is adopted to simulate the noise and evaluate the algorithm. Using a synthetic AWGN noise model has the clear advantage of simplifying the testing phase and quantifying the noise impact. However, it might oversimplify the problem since the real-world noise models can be more complicated depending on the noise source. Further, one may benefit from exploiting common noise patterns for more structured noises. For instance, the noise caused by rainy conditions may need a different treatment than a noise caused by the camera lens scratch. For instance, [68,69] generates noisy and clean image pairs by controlling the ISO of the cameras. By these approaches, the collected data could be used to emulate the camera-related noise under real-world conditions.

Similar to SR methods, denoising methods are typically used for generating clean traffic images that could be used to improve the precision of higher-level tasks. For example, in [70], a low-rank decomposition image denoising method is proposed for restoring the noisy traffic image. Likewise, in [71], spatio-temporally denoised images are used to enhance the performance of the traffic incident detection algorithm.

**Video Stabilization:** Traffic videos may contain vibrations, especially when captured by vehicle dashboard-cameras while driving on rough roads. This may undermine the performance of the subsequent processing stages (e.g., vehicle detection, speed estimation, etc.). Digital video stabilization techniques are proposed to improve the visual quality of the captured videos [72,73]. The common spirit of most video stabilization methods is extracting trajectory of objects or their representative feature points between consecutive frames and re-aligning the frames to smooth out the trajectories, based on the assumption that noise-like high-frequency fluctuations, especially when shared among most image descriptors, are caused by the camera shakes. We can subdivide these methods into pixel-based and feature-based methods. The pixel-based methods typically use block matching [74,75], phase information [76,77], and optical flow [78,79] to estimate the camera motion. Feature point detection methods can be used to convert high-dimensional images into low-dimensional representations to reduce the computation overhead. In [80], Scale-Invariant Feature Transform (SIFT) [81], Speeded Up Robust Features (SURF) [82], and other feature point extraction methods are compared for evaluating their impacts on video stabilization. Due to the computational complexity of video stabilization methods, it is often performed offline, which may not satisfy the requirements of real-time video-processing tasks. Recently, DL-based methods [83,84] enable fast and accurate online stabilization in almost a real-time fashion by instant processing of each incoming video frame with low latency.

Most DL-based methods utilize CNN and similar architectures for video stabilization using various sources of traffic data. In [80,83], video stabilization was used for aligning the car-mounted camera captured videos. In [89], the stabilized UAV captured videos were used to obtain higher accuracy for traffic video analysis.
Table 5 List of sample problems to assess the overall safety of a highway segment. The short codes include VS: video stabilization, DN: denosing, SR: super resolution, MOT: multi-object tracking, OD: object detection, IS: instance segmentation, TE: trajectory extraction, SS: semantic segmentation, EP: environment prior. C: classification. The step in brackets "[ ]" denotes this step is not required but potentially can enhance the performance. The tasks with "*" meaning that these tasks can be solved by obtaining the relationships (distance, velocity) among vehicles and determining a threshold from the relationships.

| Event                                      | Processing Steps                                      | Examples |
|--------------------------------------------|-------------------------------------------------------|----------|
| Vehicle Stopped on Shoulder                | (VS)-(DN)-LD-MOT                                     | [35]     |
| Car Crash                                  | (VS)-(DN)-OD/MOT-(IS)-C                               | [36, 37] |
| Emergency Vehicle on The Road              | (VS)-(DN)-fine-grained OD                            | [38]     |
| Careless/Evasive Lane Change               | (VS)-(DN)-LD-MOT-TE-(smoothing)-C                     | [39, 40] |
| Debris in Roadway                          | (VS)-(DN)-LD-(SR)-OD/(SS)                            | [41, 43] |
| Traffic Blocked (Slowed-down)              | (VS)-(DN)-MOT-(smoothing)-TE-(SE)                     | [44, 45] |
| Sharp Braking                              | (VS)-(DN)-MOT-(smoothing)-TE                          | [46]     |
| Passing Red Traffic Light                  | (VS)-(DN)-(OD for lights)-MOT                         | [47]     |
| Traffic Composition                        | (VS)-(DN)-OD/MOT                                      | [48]     |
| *Vehicle Driving Obviously Excessively Slow| (VS)-(DN)-MOT-TE-(smoothing)                         | -        |
| *Vehicle Driving Obviously Excessively Fast| (VS)-(DN)-MOT-TE-(smoothing)                         | -        |
| Vehicle Driving in the Prohibited Area     | (VS)-(DN)-LD-OD                                      | [49]     |
| Invalid Car in HOV                         | (VS)-(DN)-LD-OD                                      | -        |
| Improper (Careless) Entering the Road      | (VS)-(DN)-LD-MOT-TE-(smoothing)-C                     | [50]     |
| Improper (Careless) Exiting the Road       | (VS)-(DN)-LD-MOT-TE-(smoothing)-C                     | [50]     |
| Zigzag Driving                             | (VS)-(LD)-MOT-TE-(smoothing)-C                        | [39]     |
| *Distance Violation to Front Vehicle       | (VS)-(DN)-MOT-(smoothing)-C                           | -        |
| *Distance Violation to Side Vehicle        | (VS)-(DN)-MOT-(smoothing)-C                           | -        |
| Violation of The Lines                     | (VS)-(DN)-LD-MOT-(smoothing)-TE                       | [49]     |
| Pedestrian on The Road                     | (VS)-(DN)-OD                                         | -        |
| Bike/Motorcycle on The Road                | (VS)-(DN)-OD                                         | -        |
| Trailer [Oversized Car] on The Road        | (VS)-(DN)-OD                                         | -        |

5 Deep Learning for video processing

Deep learning methods are heavily used for video processing for their outstanding power in solving different problems such as object detection, object recognition, event recognition, and other video understanding tasks in general. DL methods can be considered the brainpower of most AI platforms developed for traffic safety analysis.

5.1 DL-based Classification Methods in Traffic Analysis

Classification is one of the most fundamental tasks in computer vision. The use of classification in this context can be implemented for object classification (e.g., classifying objects between vehicles, pedestrians, motorcycles, traffic signs, etc.), trajectory and traffic light and railroad crossing barrier status check, and scene detection (e.g., buildings, roads, road lanes, roadside infrastructures, etc.). Often, it is the backbone or the feature extractor part of the detection networks (e.g., SSD [90]) and segmentation networks (e.g., Mask R-CNN [91]). It can be developed at different levels, such as the basic level for classifying different object types and fine-grained classification into sub-categories (such as differentiating traffic signs or identifying vehicle classes among sedans, SUVs, trucks, etc.) based on the semantic content of the input. The well-known baseline classification methods include AlexNet (2012) [92], VggNet (2014) [93], GoogleNet (2015) [94], ResNet (2016) [95], MobileNets (2017) [96], DenseNet (2017) [97], etc.

Since the use of classification methods for most traffic-related problems is straightforward and noting the fact that there exist comprehensive reviews on classification methods, we skip the review of classification methods and refer the interested reader to [98–101]. Here, we only review fine-grained classification methods that are custom-built or customized for traffic-related problems, as presented in Table 6 and Fig.5.

5.2 DL-based Object Detection Methods in Traffic Analysis

Object detection is another key stage in DL-based processing pipelines for driving safety analysis. Object detection simply means locating different objects in images and video frames, potentially with complex backgrounds, by drawing bounding boxes around the objects of interest. It can coexist or be integrated with object classification and labeling. An illustrative example of vehicle detection using 2 benchmark methods is shown in Fig. 6(a)(b).

Notable examples of object detection in the context of driving safety analysis include detecting surrounding vehicles, humans, traffic signs, and obstacles. It also can be part of more complicated tasks such as traffic distribution and composition analysis, improper lane crossing events, trajectory extraction, speed estimation, moving object tracking, path planning, and detecting vehicles on road shoulders, etc., as presented in Table 6.

Another use case of object detection is removing Personal Identifiable Information (PII), such as masking human faces and license plate numbers before publishing traffic video, as shown in Fig. 6(c).

Although there exist some datasets for traffic analysis from the roadside cameras [119–121], still there is a critical need for
larger datasets that cover different zones, urban, suburban, and rural setups, residential and high-risk zones, railroads, and environmental conditions.

Some notable ongoing research problems include solving the trade-off between the algorithm’s accuracy and speed, realizing small object detection, distributed and federated learning, model sharing among roadside servers, and implementing lightweight models and embedded devices appropriate for autonomous vehicles, etc.

Compared to the conventional object detection algorithms such as Viola-Jones detector [122], the Histogram of Oriented Gradients (HOG detector) [123], and Deformable Part-based Models (DPM) [124], CNN-based methods substantially improve the recognition success rate.

From the implementation point of view, the DL-based algorithms can be divided into one-stage and two-stage methods. The two-stage detectors first generate Regions of Interests (RoIs) and then send the region proposals down the pipeline for object classification and bounding-box regression. R-CNN series [118, 125–128] comprises the most popular two-stage algorithm family.

One-stage detectors directly treat object detection tasks as a regression and classification problem. These methods are divided into anchor-free and anchor-based methods. In anchor-based methods, a set of bounding boxes with different predefined sizes are required to capture the scale and aspect ratio of the objects. Some famous implementations include You Look only Once (YOLO) family (e.g., YOLOv2 [129], YOLOv3 [130], and YOLOv4 [131]), and YOLOv5 as well as the Single Shot multi-box Detector (SSD) series [90]. Recently, anchor-free methods are getting more attention to avoid defining anchor-related hyperparameters and to ease complicated computations. The main ideas include implementation by dense prediction (e.g., DenseBox [132], Fully Convolutional One-Stage (FCOS) object detectors [133], RetinaNet [134]) as well as implementation by keypoints and center points (e.g., CornerNet [135, 136], CenterNet [137,138], ExtremeNet [139]).

Generally speaking, two-stage methods can achieve higher accuracy but at lower speeds than the one-stage methods. Some recent one-stage methods (including YOLO V4 [131] and SSD [90]) solve the trade-off between the accuracy and speed by realizing a more efficient network structure. A summary of these algorithms’ performance is presented in Fig. 7.

The applications of object detection methods in traffic video analysis mostly relate to understanding the objects surrounding the road users, such as vehicles, plates, and traffic signs. Most research works in this area adopt one of the aforementioned algorithms for object detection, as summarized in Table 7. The metrics used for object detection, recognition, and image segmentation are outlined in Table 10. We note that some works (such as [140–144]) tend to fine-tune the CNN framework according to the task requirement, and the recent mainstream traffic works begin to deploy the R-CNN series, YOLO series, and SSD widely. It means these algorithms can stand the test of practice, but it does not mean that the other algorithms are not favorable. It also can be due to the relatively low complexity of this task, or due to the difficulty of deploying recent detectors in real-world scenarios.

5.3 Visual Tracking

Visual tracking refers to capturing the movement of specific objects by processing video frames. Traffic video processing by autonomous vehicles or traffic monitoring systems is perhaps the most popular application of visual tracking. With the advent of AVs and technology-assisted driving systems, this research area has become a hot topic. The mainstream algorithms can roughly be categorized into Correlation Filter-based Trackers (CFT) and non-CFT methods [139]. The two challenging issues are the natural difficulty of reliable visual tracking and the lack of well-annotated datasets, especially for driving safety analysis.

*Note that the authors of YOLOv5 are different from the previous versions*
Fig. 6: (a),(b) are the examples of vehicle detection by two alternative methods YOLOv5\cite{117} and Faster R-CNN\cite{118}. They output the location and category of each object with detection confidence. Cars are shown by orange bounding boxes, trucks are shown by lime bounding boxes, and persons are shown by red bounding boxes. As shown in (a),(b), YOLOv5 performs worse on small vehicle detection while Faster R-CNN has missed objects in the near zone. (c) is the PII removed data.

Table 7: Examples of object detection tasks for traffic analysis. If the dataset is not indicated, it means the current work uses the dataset generated by its authors. \(\triangle\) means that this work uses fine-grained detection problem. '*' Means this work is based on YOLOv5. It is worth mentioning that YOLOv5 may not be considered as a member of YOLO family.

| Task             | Methods         | Paper: [Ref] Authors (year) | Performance: Accuracy [Dataset] |
|------------------|-----------------|-----------------------------|---------------------------------|
| Vehicle Detection| R-CNN family    | 145 Espinosa, et al.(2017)  | 70% 84.43% on DETRAC dataset [151] |
|                  |                 | 146 Wang, et al.(2017)      | 100% |
|                  |                 | 147 Sohn, et al.(2017)      | 95.5% |
|                  |                 | 148 Zhang, et al.(2017)     | 97%  |
|                  |                 | 149 Peppa, et al.(2018)     | 98%  |
|                  |                 | 150 Yu et al.(2017)        | 99.73% |
|                  | RYOLO family    | 152 Sang, et al.(2018)      | 100%  |
|                  |                 | 153 Kim, et al.(2019)       | 99.51% |
|                  |                 | 154 Kasper-Eulaers et al.(2021)* | 93% car, 63% truck front, 52% truck back (winter condition) |
|                  |                 | 155 Nayak et al.(2019)      | 97.73% |
|                  | SSD             | 156 Zhang, et al.(2019)     | 85.9% on UA-DETRAC [151] |
|                  |                 | 157 Chen, et al.(2020)      | 96.5% |
|                  |                 | 158 Cao, et al.(2020)       | 97.18% on KITTI |
|                  | CNN             | 159 Peppa, et al.(2018)     | 98.2%  |
|                  | Pre-processing+CNN | 160 Chen, et al.(2014)  | 99.7%  |
|                  | CNN             | 161 Zhou et al.(2016)      | 62.8% on UTS(self), 64.44% PASCAL VOC2007\cite{159}, and 79.41% on LISA 2010\cite{160} |
|                  | SSD             | 162 Rene, et al.(2020)     | 92%  |
|                  |                 | 163 Hu, et al.(2020)       | 90.12% |
|                  |                 | 164 Panjniko, et al.(2020) | 94%  |
|                  | CNN             | 165 Kim, et al.(2017)      | 97.67% on GAP-LP dataset\cite{166} |
|                  |                 | 166 Krasnov, et al.(2020)  | 97.9%  |
|                  |                 | 167 Xie, et al.(2018)      | 99.32% on UCSD 97.38% on PKU [167] |
|                  | CNN             | 168 Chen, et al.(2019)     | 98.22% on AOLP |
| Plate Detection  | R-CNN family    | 169 Lee, et al.(2016)      | 99.94% |
|                  | YOLO family     | 172 Kessentini, et al.(2019) | 97.67% on GAP-LP dataset\cite{166} 91.46% on Radar |
|                  |                 | 173 Kharazae, et al.(2020) | 97.9%  |
|                  | CNN             | 174 Xie, et al.(2018)      | 98.32% on UCSD 97.38% on PKU [167] |
|                  | Pre-processing+CNN | 175 Chen, et al.(2019)  | 98.22% on AOLP |
| Traffic Sign Detection | R-CNN family | 176 Quan, et al.(2016)  | 92.58% |
|                  |                 | 177 Shao, et al.(2019)     | 69.56% on GTSDB+CTSD\cite{179} |
|                  |                 | 178 Zhang, et al.(2020)    | 98.7% on GTSDB |
|                  |                 | 179 Zuo, et al.(2017)      | 34.49% mAP on CCP2016 |
|                  |                 | 180 Wu, et al.(2019)       | nAP 91.75% GTSDB |
|                  |                 | 181 Peng, et al.(2016)     | nAP 90% on GTSDB |
|                  | YOLO family     | 182 Zhang, et al.(2017)    | 96.69% on CTSD and GTSDB |
|                  |                 | 183 Tai, et al.(2020)      | 99.1% mAP |
|                  |                 | 184 Liu et al.(2021)*      | 97.2% mAP |
|                  |                 | 185 Qin et al.(2021)*     | 94.3 % mAP |
|                  | SSD             | 186 Gao, et al.(2019)      | 91.0% on GTSDB 75% on TT100K |
|                  | CNN             | 187 You, et al.(2020)      | 91.0% on GTSDB 75% on TT100K |
|                  | MSER+SVM+CNN    | 189 Yang, et al.(2015)     | 98.24% GTSDB 98.77% CTSD |
|                  | FullyConv       | 190 Zhu, et al.(2016)      | 91% TT100K |
|                  | CNN             | 191 Wu, et al.(2013)       | AUC 99.73% "danger", 97.62% "mandatory" on GTSDB |
|                  | CNN             | 192 Khustianov, et al.(2017) | 99.94% on GTSDB |

Visual tracking methods can be categorized into discriminative methods\cite{196,199} and generative methods\cite{194,198} from the modeling standpoint. Generative tracking methods comprise the following sequential steps: (i) extract the target features to learn the appearance model, which represents the target, (ii) search through the image area to find areas that best match the model, using pattern matching. The target information carried by generative models is often richer than that of the discriminative methods. Also, generative models make it easier to meet the evaluation criteria and the real-time requirements of target tracking when processing a massive amount of data. The key components of generative methods
include target representation and target modeling. The limitations of this approach include (i) background information of the image is not fully utilized, and (ii) the appearance of the target in different video frames may include substantial randomness and diversity, which affects the stability of the model.

Discriminative methods turn the tracking problem into a classification problem, and rely on training a classifier to distinguish between the target and the background. The target area is considered the positive sample in the current frame, and the background area is the negative sample. Discriminative methods turn the tracking problem into a classification problem, which can simultaneously utilize the information from the target and background. ML methods can be employed to train a binary classifier to distinguish between the target (considered as the positive sample) and the background (considered as the negative samples), and can be updated online as more video frames are accumulated. The trained classifier is used to find the optimal area in the next frame. Discriminative methods are often more robust than generative models when facing appearance and environmental changes.

Correlation Filter (CF) methods and DL methods are often considered discriminant methods. In recent years, tracking methods based on correlation filtering [199–204] have gained a lot of attention from computer vision researchers because of their fast speed and reasonable performance. The correlation filter is initialized by the given target in the first frame of the input. A classifier is trained by regressing the input features to the target Gaussian distribution. The response peak in the predicted distribution is found in the follow-up tracking step to locate the target’s position. Correlation filtering methods, when combined with deep features and CNN architectures (e.g., R-CNN series [203–207], SSD-based methods [208]) exhibit outstanding performances and hence have gradually become a dominant approach in this field.

In addition to CF-based trackers, some other DL frameworks, even without using correlation filters, can also achieve an excellent performance. More elegant methods, including [209–214] tried to directly employ sequential learning models such as Long Short Term Memory (LSTM) or Recurrent Neural Networks (RNN) in their network structures after CNN-based feature extractors to capture the temporal information of video frames that represent object motions. Siamese Network structure (shown in Fig. 11) offers a paradigm shift for visual tracking. The core idea of Siamese networks [215–221] is training twin networks to identify the similarity between two different images, such as the same object in consecutive video frames. These methods address both the similarity knowledge learning and the real-time operation requirements with acceptable tracking accuracy.

Many contemporary traffic monitoring systems still tend to use conventional tracking methods such as filtering and convolution methods due to their efficiency, low complexity, and stability. For examples, methods based on Kalman filtering [222–224], Gaussian Mixture Models (GMM) [225], Hidden Markov Models (HMM) [226], and SIFT-based methods [227–229] are used in traffic monitoring systems. However, using DL-based methods is gaining traction in recent years to perform traffic monitoring tasks due to the emergence of powerful and low-cost processing platforms, making DL methods more affordable and near real-time.

Hybrid methods that enable tracking by detection are another possibility. For instance, a DL method can be used for fast and accurate object detection, followed by a second estimator based on conventional methods, Kalman filtering, and Kanade–Lucas–Tomasi (KLT) feature tracker for tracking purposes. A summary of some important implementations is presented in Table 8.

5.4 Semantic Segmentation

Semantic segmentation, where objects of different types are separated, can be considered the heart of many video processing tasks. For instance, vehicle detection, vehicle tracking, and environment perception in a crowded environment with interlaced and overlapping objects can be powered by semantic segmentation when regular segmentation methods fail to separate objects from complex backgrounds. The purpose of semantic segmentation is to label each pixel of an image to represent different categories (e.g., cars, pedestrians, roadside infrastructures, traffic signs, etc.). For instance, semantic segmentation can be employed by an autonomous vehicle for background modeling, identifying road boundaries and free spaces, and detecting lane markings and traffic signs. Semantic segmentation can also be used by an external traffic monitoring system for analyzing the behaviors of human-driven and self-driving vehicles in specific zones and times. To avoid reliance on massive data collection and expensive annotations, semi-supervised and weakly-supervised learning methods [244–247] are developed for low-cost implementation with reasonable performance.

Early works tended to deploy existing classification algorithms at the patch level for semantic segmentation. Since 2014, Fully Convolutional Network (FCN) [239] was introduced that allows spatially dense prediction tasks by translating famous DL architectures such as AlexNet, VGG net, and GoogLeNet into fully convolutional network architectures. Afterward, many upgraded architectures such as U-net [250] and SegNet [251] were proposed, which build upon the concept of FCN and utilize auto-encoder architectures for semantic segmentation with small training datasets using data augmentation methods. This architecture is further updated to multi-stage auto-encoder networks in [252]. In some work [253–256], atrous/dilated convolution architectures are used to keep spatial resolution and expand the receptive fields. The authors of [257] expanded the receptive fields by employing large kernel with its proposed global convolution.

In [258–260], a feature fusion method is deployed that allows the framework to learn global features merged with more local features. For these methods, Conditional Random Fields can be used to enhance the output. Other works [261–263] are based on a Recurrent Neural Network (RNN) structure that can better tackle sequence-related tasks. Additional models including [263] based on graph convolutional network, [264] based on pyramid pooling model, and [265] based on learning relations between the object region and pixels have competitive performance. Furthermore, [256–260] focused on semantic segmentation with 3D point cloud data, which have great potentials for autonomous driving and traffic safety analysis. More specifically, the methods [256–260] are evaluated by indoor scene dataset, while [268] is evaluated by urban scene captured by scanners, something potentially more relevant for traffic analysis. The performance of some models is shown in Fig. 12.

The applications of semantic segmentation frequently appear in the world of AVs. Some examples include (i) scene understanding, which involves understanding the traffic environment with road users, (ii) free space estimation to determine the available spaces on the road that a vehicle is allowed to use with no collisions, and (iii) Stixel representation, which assigns each pixel with a 3D depth information. A summary of related works is presented in Table 8.
Fig. 8: Network architecture for R-CNN family of localization and segmentation.

Table 8 Some traffic-related visual tracking method that use deep learning. The default dataset is generated by the author of each work. The default performance metric is accuracy unless specified otherwise.

| Paper: [Ref] Authors (year) | Methods | Performance |
|-----------------------------|----------|-------------|
| [231] Qiu, et al. (2018)    | YOLO+KLT | 76.4% Recall 88.2% Precision |
| [232] López-Sastre, et al. (2019) | Faster R-CNN | 30.5AP on subset:M-30 66.2%AP on subset:M-30-HD 38.1%AP on subset:Urban1 of GRAM-RTM |
| [233] Scheidegger, et al. (2018) | CNN+PMBM filter[234] | 80.04% MOTA on KITTI |
| [235] Zou, et al. (2019)    | Siamese network+SPP+MDP | 75.29% MOTA campus 76.06% MOTAurban 78.14% MOTA highway on KITTI |
| [236] Li, et al. (2019)     | FPN+tracking loss | 83.2% IDF1 on nvidia AI city |
| [237] Nikodem, et al. (2020) | hourglass | MOTA 97+% MCTA 91+% |
| [238] Zhao, et al. (2018)   | SSD+dual Kalman filters | Shown in the form of figures |
| [239] Wang, et al. (2019)   | YOLOv3+SIFT | 81.3% MOTA |
| [240] Usmankhujaev, et al. (2020) | yolov3+kalman filter | 100% Daytime Wrong cases detection 89.83% Daytime(flipped) Wrong cases detection 86.11% Nighttime(flipped) Wrong cases detection |
| [241] Kwan, et al. (2018)   | ResNet | For compressive measurements |
| [242] Fernández-Sanjurjo, et al. (2019) | CNN detector+DCF+Kalman filter | 86.96% MOTA on MOT 15 |
**Table 9** Some semantic segmentation works in the traffic field.

| Task                      | Paper: [Ref] Authors (year) | Methods                                      | Performance                                  |
|---------------------------|-----------------------------|----------------------------------------------|----------------------------------------------|
| Scene understanding       | [269] Romera, et al.(2017)  | Autoencoder                                  | pixel accuracy>95% Cityscapes               |
|                           | [270] Lyu et al.(2019)      | Edge Detection Network+Fusion                | 63.2% mIoU Cityscapes                       |
|                           | [271] Deng, et al.(2017)    | CNN+pyramid pooling module                   | 54.5% mIoU Cityscapes                       |
|                           | [272] Sáez, et al.(2018)    | Autoencoder                                  | 59.3% mIoU Cityscapes                       |
|                           | [273] Kendall, et al.       | Bayesian Autoencoder                          | 63.1% mIoU CamVid                          |
| Free space estimation     | [41] Ohgushi, et al.(2020)  | Autoencoder                                  | 21.9% mIoU                                 |
|                           | [42] Hua, et al.(2019)      | Autoencoder+fusion+optical flow              | Path planning accuracy                      |
|                           | [274] Levi, et al.(2015)    | CNN                                         | 0.15m Indoor and 0.27m Outdoor              |
|                           | [43] Deepika, et al.(2017)  | SegNet                                       | 89.12% maxF SEGMENTATION on KITTI           |
| Stixel representation     | [275] Schneider, et al.(2016)| FCN+SGM                                     | 7.8 Disparity Error on KITTI 15.2% on Ladicky |
|                           | [276] Cordis, et al.(2017)  | FCN+SGM+graphical model                      | 83.1% exact Disparity accuracy              |

**Table 10** Some metrics to evaluate traffic works.

| Metric                           | Definition                                                                 |
|----------------------------------|---------------------------------------------------------------------------|
| Accuracy                         | Correct predictions                                                       |
|                                  | Total predictions                                                         |
| Recall (R)                       | $\frac{TP}{TP+FN}$ ($\ast$)                                              |
| Precision (P)                    | $\frac{TP}{TP+FP}$                                                        |
| F1 score                         | $2 \times \frac{P \times R}{P+R}$                                        |
| Pixel Accuracy (PA)              | $\frac{TP+TN}{TP+TN+FP+FN}$                                               |
| Normalized error (mean)          | Normalized error=$\frac{|d-d_{gt}|}{d_{gt}}$.                               |
|                                  | $d$ is the calculated distance and $d_{gt}$ is the ground truth of distance. |
| Average Precision (AP)           | $AP = \int_0^1 P(R)dR$ P-R CURVE (for one class)                         |
| mean Average Precision (mAP)     | $\text{mean}(AP_i)$                                                       |
| Intersection over Union (IoU)    | $B_p \cap B_{gt}$                                                         |
|                                  | $B_{gt}$: ground truth bounding box                                       |
|                                  | $B_p$: predicted bounding box                                             |
| Disparity accuracy               | $\text{disparity}_{\text{inlier}}\%$. The disparity $\leq 3\text{px}/5\%$ is true |
| Stixel-wise percentage accuracy  | Similar to PA                                                              |
| Identification F1-Score (ID : F1)| $ID : F1 = \frac{2IDTP}{2IDTP+IDFP+IDFN}$                                |
| multiple object tracking accuracy (MOTA) | $MOTA = 1 - \frac{FN+FP+\Phi}{\Phi}$                                    |
|                                  | $\Phi$ denotes the number of fragmentations                                |
| Multi-camera Tracking Accuracy (MCTA) | $M_w$: within-camera identity mismatches                                   |
|                                  | $T_w$: true within-camera detections                                       |
|                                  | $M_h$: handover mismatches                                                 |
|                                  | $T_h$: true handover detections                                            |
| RMSE                             | $RMSE = \sqrt{\sum_{i=1}^N (p_i - a_i)^2 / N}$                           |
5.5 Instance Segmentation

Instance segmentation, which deals with detecting and delineating distinct objects of interest in images and video frames, can be considered as one of the most difficult tasks in computer vision. It goes one step beyond semantic segmentation and not only labels the pixels based on their object categories, but also distinguishes between different object instances of the same type. This is crucial importance for traffic imagery analysis in dense zones, where different objects (e.g., vehicles, pedestrians) have to be identified and located in video frames for optimal decision making by AVs, or to extract safety metrics by monitoring systems. In contrast to semantic segmentation, instance segmentation only needs to find the edge contour of the object of interest with no need for bounding boxes, hence it can realize a more accurate object detection when assessing the behaviors of vehicles. Building a reliable and real-time method, for instance, segmentation, especially for crowded zones and under highly distorted traffic videos (e.g., in rainy and cloudy weather conditions), can be challenging.

Instance segmentation can be divided into one-stage and two-stage methods. Two-stage methods often require generating region or object proposals followed by a classification-based segmentation performed over the features extracted from the selected regions or bounding boxes around object proposals. To generate region proposal, [91, 278, 279] predict a bounding box for each instance, while [259, 280, 282] develop pixel-wise coarse segmentation masks.

Since instance segmentation partitions the image by masking the detected objects and associating each pixel with a distinct object, some two-stage detectors such as Faster R-CNN [113] can execute instance segmentation task after some post-processing, e.g., by adding a branch for mask predictions.

One-stage instance segmentation methods do not utilize a separate stage for generating region proposals; rather, they apply the segmentation directly to the original images. Some methods [283–286] inspired by one-stage detectors (such as YOLO [129]) directly predict bounding boxes. However, these anchor-based methods heavily rely on predefined anchors, which may be affected by many factors such as the predefined anchor boxes’ aspect ratio and scales. Another approach is developing anchor-free methods using dense prediction or centerpoint/keypoints. For instance, [287–289] relied on FCOS [133] as their dense prediction detector, while ExtremeNet [139] is a keypoint-based detector that can roughly perform the segmentation task. The difference between the accuracy of one-stage and two-stage methods is not as great as one may expect. Indeed, recent one-stage methods such as CenterMask [288] can perform real-time inference with accuracy as high as two-stage methods, or even better. The performance of some instance segmentation models is shown in Fig.13.

In the context of traffic analysis, instance segmentation can be used not only to identify and locate vehicles but also to obtain...
5.6 Video-based Event Recognition

Video-based event recognition extends the role of DL methods from object detection and identification paradigms into a more intricate problem of understanding events. It provides endless possibilities to explore the interactions among objects in an interactive environment rather than focusing on disjoint object-based tasks. Indeed, without event detection and analysis, the majority of video information remains unexploited and underutilized. Note that many safety metrics relate to the interactions of vehicles with one another and with the environment. For instance, traffic sign interpretation can be recast as an image-based object classification problem, while more intricate violations of traffic safety such as unsafe lane changing behavior without signaling require modeling interactions between vehicles and their surrounding environments. Such challenging problems are still in their infancy stages and apparently require heavy investment by the research community. In essence, the event recognition problem is also related to another well-investigated problem of video-based human action recognition, and similar tools and algorithms can be adopted here.

An alternative method of event analysis is using conventional analysis approaches by extracting object-based information and manually feeding them into statistical and reasoning models such as Hidden Markov Models (HMM) for safety analysis. However, with the recent advances in developing powerful DL methods, they enable a more automated and direct way of evaluating the behaviors of involved vehicles. The most naive way of event recognition can be realized by extracting static features from video frames using methods like SIFT [299], SURF [82], the Local Binary Patterns (LBP) [300, 301], HOG detector [123], Binary Robust Invariant Scalable Keypoints (BRISK) [302], Features from Accelerated Segment Test (FAST) [303, 304] and GIST (a very low-dimensional scene representation) [305, 306], and then performing object detection followed by a time-series analysis for even recognition. Some other methods combine the feature extraction and time-series analysis stage into directly extracting temporal features using methods like motion spatio-temporal features (Motion SIFT (MoSIFT) [307], Spatio-Temporal Interest Points (STIP) [308], and Dense trajectories [309, 310]), then perform the classification task.

More contemporary event detection algorithms benefit from DL methods to automate this process, and a dominant method is directly applying a 3D convolutional network to process non-anomalous frames [311]. Another approach is deploying CNN to extract spatial features and then performing sequential analysis using methods such as RNN/LSTM [312, 318] to preserve the temporal features.

Most of these methods use supervised learning methods to detect a set of predefined events. Therefore, developing generic methods for understanding safety risks from driving profiles and tackling unseen types of safety violations has a long way ahead.

Event recognition is of particular importance for traffic safety analysis since it can be used for detecting abnormal events and traffic violations and their associations with crash rates [319], car behavior analysis [320, 321], and pedestrians’ crossing identification [316, 322, 323]. Recently, some works [37, 324, 325] try to predict anomaly actions using Generative Adversarial Networks (GAN). The core idea is predicting the future video frames for a normal user with rational behavior from the history of normal sequences to identify severe abnormalities by comparing the observed video frame against the anticipated one.

For vehicle-level analysis, when the goal is detecting plain and simple events, conventional methods achieve a reasonable performance. For instance, [326] uses HMM to detect traffic abnormality, and [327] deployed a topic model to recognize crashes from surveillance videos. However, using DL methods can be used to analyze more complex events and achieve higher performance records. Some generic platforms like Retina [328], provide a query-based approach to translate the challenging task of event recognition into a sequence of object detection and classification problems.

We summarize important DL-based event recognition methods in Table 12 and present some popular datasets for event recognition tasks in Table 13. One observation is that unsupervised, semi-supervised, and self-supervised models are becoming more prevalent in recent works [37, 319, 329] to mitigate the costly and tedious job of video annotation and simplifies volume video processing.

5.7 Sensor Information Processing

It is notable that there are several studies and datasets devoted to sensor information analysis. Different types of commonly used sensors were provided previously in section 4. Of particular and increasing interest is the point cloud mapping collected from LiDARs, not just because they offer more accurate distance measurement, but also they are considered as separate and independent sensing of the environment to that of videos to ensure accurate perception thus road safety. 3D point cloud mapping from LiDAR is different from grid-based 2D images, thus different treatment strategies are pursued. LiDAR point cloud semantic segmentation works use deep neural networks, initially treating point cloud as construct graph [352], followed by the development of multi-layer perceptrons to learn from raw cloud data directly [350, 367]. More recently, the spherical projection has been employed to map LiDAR sequential scans to depth images, and improved segmentation [343, 346]. Since the focus of this paper is video-processing for traffic safety analysis, we refer the interested readers to [474, 350].

5.8 Network-level Analysis

Traffic flow problems can be formulated as a network of mobile nodes, and studies on individual crash analysis can be extended to the more complex setup of network-level analysis.

There exist a few research paradigms that consider network-level analysis. One bold example is the transportation network design and related family of problems. Transportation network design belongs to the category of operations research and can be divided into the Road Network Design Problem (RNDP) and Service Network Design Problem (SNDP) according to their features and functions [351]. The works [352, 354] of RNDP aim to optimize the performance of urban networks according to some criteria such as topology, capacities, and flow accessibility. The works [355–359] of SNDP aim to address the planning of operations for freight transportation carriers, such as station locations, route planning, and operation frequency.

Traffic prediction studies often employ statistical techniques (such as Kalman filtering [360], hidden Markov model [361], Bayesian inference [362] and DL (such as LSTM, CNN) methods [356, 359]) to infer the network state and produce optimal strategies for different conditions.

Network-level analysis can be used for traffic safety analysis as well. Some key objectives would include finding correlations between traffic flow, safety metrics geo-maps, and crash rates. For example, one may expect a direct relation between the traffic composition (density and variety of vehicles) and the number of crashes at different parts of highways. One may also expect relations between the traffic flow and crash rates of nearby intersections or segments. Network-level analysis can shed light on these highly unexplored research areas.

There exist four general approaches for network-level analysis [367] including traditional statistical models (e.g., [368]), endogenety/heterogeneity models (e.g., [369, 370]), data-driven methods [371, 372], and causal inference models. Furthermore, we note that some models [373, 374] exploit previously collected crash data with road information (such as average daily traffic (ADT), lane width, speed limited, shoulder width) to estimate crash frequencies at intersections or segments. Some models [375, 376] explore risk probabilities by processing geometric features. Recently, new
Table 11 Some instance segmentation methods used for traffic imagery analysis. If the dataset is not indicated, it means that a proprietary dataset generated by the authors is used.

| Task                      | [REF] authors (Year) | Methods     | Performance                                                                 |
|---------------------------|-----------------------|-------------|-----------------------------------------------------------------------------|
| Obtain comprehensive      | [290] Zhang, et al.(2015) | CNN+MRF    | 59.0% object recall, 83.1% of the randomly sampled foreground pixel pairs ordered on KITTI |
| vehicle information       | [291] Mou, et al.(2018) | ResFCN      | 95.87% F1 score on ISPRS                                                    |
|                           | [292] Zhang, et al.(2020) | Mask R-CNN  | 97% vehicle types accuracy                                                  |
|                           | [293] Huang, et al.(2018) | Mask R-CNN  | 1.333 front 4.698 side-way KITTI                                          |
| Lane detection            | Neven, et al.(2018)    | multi-task CNN | 96.4% accuracy                                                              |
|                           | Roberts, et al.(2018)  | SegNet      | IoU Mapillary 82.9% CityScapes 85.2% KITTI 83.8% author’s dataset 95%       |
| 3D reconstruction         | Hadi, et al.(2020)     | Mask R-CNN  | AP@50% 8.862% CityScapes 43.949% IDD[297] 5.643% WildDash[298]           |

Table 12 Some recent DL works for video-based traffic abnormal event detection. Unless otherwise specified, the non-indicated numbers is the accuracy.

| Paper: [Ref] Authors (year) | Methods                  | Performance                      |
|-----------------------------|--------------------------|----------------------------------|
| [37] Nguyen, et al.(2020)   | GAN                      | F1 score 0.9412 AI City Challenge 2019 |
| [329] Yao, et al.(2019)     | Auto Encoder-Decoder with GRU | 60.1% AUC on A3D Dataset       |
| [356] Tian, et al.(2019)    | YOLO-CA                  | 90.02% AP CAD-CVIS               |
| [330] Kim, et al.(2020)     | 3D conv                  | 82%                              |
| [331] Ijjina, et al.(2019)  | 3D conv                  | 71%                              |
| [332] Shah, et al.(2018)    | DSA-RNN[333] Faster R-CNN | 47.25% on CADP                   |
| [334] Srinivasan, et al.(2020) | Detection Transformer[335] | 78.2% on CADP                    |
| [336] Suzuki, et al.(2018)  | QRNN+DeCAF[337]          | 99.1% mAP                        |
| [319] Giannakeris, et al.(2018) | Faster RCNN              | F1-score 0.33 RMSE 227 on NVIDIA CITY Track 2 |
| [338] Arceda, et al.(2018)  | YOLO+ViF[339]+SVM        | 89%                              |
| [340] Biradar, et al.(2019) | YOLOv2+ CNN              | F1-score 0.3838 RMSE 93.61 on NVIDIA CITY Track 3 |
| [355] Xu, et al.(2018)      | Mask-RCNN+ ResCNN        | F1-score 0.8649 RMSE 3.6152 on NVIDIA CITY Track 2 |
| [341] Doshi, et al.(2020)   | YOLO+KNN+K-means         | F1-score 0.5926 RMSE 8.2386 on NVIDIA CITY track 4 |
| [320] Zhou, et al.(2016)    | 3D Conv                  | 95.2% on U-turn dataset          |
| [321] Franklin, et al.(2020) | YOLOv3                   | 100% and 95.34% for input video 1 and video 2 |

models[37][38] take advantage of advanced sensors techniques and high-performance computation frameworks to infer real-time crash risks and take proactive strategies. Safety analysis frameworks can perform integrative analysis by incorporating different static and dynamic data modalities (i.e., imagery and sensor inputs) from different points of view to comprehend the network status and derive realistic distributions for safety factors. An immediate benefit of such networks would be assessing the contribution of different factors on safety distributions and providing advisory to improve roadways and infrastructure design as well as developing safety enforcement and public education campaigns. A key challenge is developing strategies and scheduling policies for data aggregation to provide required modalities for network-level safety analysis at the minimum cost possible. Also, data aggregation is constrained by the utilized networking infrastructure and communication protocols between the vehicles and roadside infrastructure. Study of such networks are out of the scope of this work, and we refer the interested users to[372, 383, 383–389].

Recently, the ideas of using data augmentation and physics-informed neural networks (PINN) are proposed to mimic the dynamics of complex systems while mitigating the need for manual annotation of massive datasets. We believe that the power of PINN and data augmentation is not yet fully utilized in this context. There is a great potential to develop surrogate models for traffic flow and risk analysis. The authors also believe that elegantly designed graph neural networks can play an essential role in modeling network-level events and trends.

6 Traffic Datasets

Table 13 provides a relatively complete list of datasets that can be used for different aspects of video-based analysis for different CV-based traffic tasks. Data for trajectory analysis is often extracted from still cameras or drone-captured videos and used to understand microscopic-level (such as car-following models) to macroscopic-level traffic flow (such as traffic wave models) [390]. Safety analysis of human-driven vehicles closely relates to video understanding tasks such as scene recognition, object detection, instance and semantic segmentation, developed for auto-driving vehicles. Traffic sign detection is another difficult problem due to using different variants for the same signs, uncontrolled and varying light conditions, the impact of weather conditions (rain, snow, fog, etc.), and the unpredictable traffic environment, especially when real-time processing is needed. Naturalistic Driving Studies often record and process the driver’s behavior, cognition, and perception of the surrounding environment without influencing or distracting the driver. This kind of
data is generally large-scale since it requires long-term observations with multiple observed objects. The plate detection task involves drawing a bounding box around every detectable plate. A sub-task of plate detection is Optical Character Recognition (OCR), which requires understanding and validating engraved license plate numbers. Another twist to this problem is the angle of view. While most images are taken by roadside cameras, drone-based imagery datasets provide top-view images that solve some problems but poses new image processing challenges like human detection and small object detection.

It is notable that general Computer Vision (CV) datasets, presented in Table.15 can be used to approximately evaluate the performance of developed algorithms if the desired traffic-specific dataset is not readily available. Here, we also want to mention some popular CV datasets (shown in Table.15 since the most recent state-of-the-art algorithms deploy them as their evaluation criteria.

**Crash Data:** Police-reported Crash Data, generally collected and stored by local, regional, state, and/or national government agencies after traffic crashes, can provide official, objective information about crash incidents. Sanitized crash data (e.g., personal information removed) is generally publicly available at the state level either online (e.g., Michigan crash data can be obtained at [391]) or by a public records request. National level crash data in the US is available from the National Highway Traffic Safety Administration (NHTSA) via the National Automotive Sampling System General Estimates System (NASS-GES), which provides details on nationwide fatal crashes. Crash data sets provide a wealth of information related to crash circumstances (including sequence of crash events and crash type), environmental conditions at the time of the crash, roadway characteristics at the crash scene, vehicle/road user information, and crash involved person-level characteristics (e.g., injury severity, age, gender, impairment, etc.). among other information. The Model Minimum Uniform Crash Criteria (MMUCC) provides a voluntary guideline for agencies with respect to the minimum data elements that should be included in crash databases and includes a description of 115 recommended data elements related to the incident, vehicle, person, roadway, and other categories. Crash data can be used to revise traffic and/or roadway plans, develop safety countermeasures, and explore associations between crashes and traffic safety violations or other non-crash safety metrics. It can also be used to incorporate prior knowledge when analyzing traffic safety events.

**Missing Datasets:** There is a critical need to develop new datasets for traffic analysis that cover the underexamined aspects. As mentioned later in section 8.1, some studies consider driving safety from the behavioral science perspective. For instance, eye motion tracking can be used to gauge the driver’s attention and detect distraction episodes. Very few datasets exist to facilitate such research. Among these, we found two datasets (The 100-Car Naturalistic Driving Study and SHRP 2 NDS dataset) that offered a relatively large number of samples for driver behavior analysis. However, we noted that in these datasets, the activities for one vehicle in a specific scene are non-repeating, meaning that these datasets cannot provide driver-specific information. The authors of this study are currently working to develop a small dataset for driver-specific anomaly detection by collecting aerial imagery from test drivers’ behavior on specific scenarios multiple times. This dataset would enable profiling drivers based on their reaction to traffic conditions and use it to find abnormal behaviors that can be indicators of driving issues and potential crash risks. Also, very few datasets record traffic events from different perspectives. Datasets that can offer roadside imagery, along with aerial imagery and car-mounted cameras for synchronous analysis of traffic views, can open new research directions, particularly for multi-modal video analysis. The visualization of some exemplary crash data from ADOT is presented in Fig.15.

7 Safety Metrics

An essential objective of driving safety analysis is extracting operational safety metrics, which are quantifiable measures extracted from traffic videos (or other data sources) that determine the relative risk of an event that may lead to a crash. Some important safety metrics that are used to analyze car crashes include:

1. **Temporal-based indicators:** Time to Collision (TTC), Extended Time to Collision (Time Exposed Time-to-Collision (TET)), Time Integrated Time-to-Collision (ITTC), Modified TTC (MTTC), Crash Index (CI), Time-to-Accident (TA), Time Headway (H), and Post-Encroachment Time (PET).

2. **Distance-based indicators:** Potential Index for Collision with Urgent Deceleration (PICUD), Proportion of stopping Distance...
Fig. 15: The visualization of some exemplary crash data. Each point in (a)-(c) denotes a crash, and (d)-(f) are the corresponding heatmaps.
(PSD), Margin to Collision (MTC), Difference of Space Distance and Stopping Distance (DSS), Time Integrated DSS (TIDSS), and Unsafe Density (UD); 3. Deceleration-based indicators: Deceleration Rate to Avoid a Crash (DRAC), Crash Potential Index (CPI), and Criticality Index Function (CIF).

Reviewing different safety metric is out of the scope of this paper and we refer the interested readers to recent papers on safety metrics \cite{429,430}. As part of the Institute of Automated Mobility (IAM) project, the authors of this paper are working toward extending safety metrics into network-level metrics and developing a taxonomy of metrics for safety metrics for AVs based on the level of access required of ADS data \cite{432}.

### 8 Miscellaneous Points

This section discusses some general facts about the driving safety analysis and reviews closely related research directions. We conclude by mentioning key areas and emerging topics that require further investigations.

#### 8.1 Study from human condition and psychology perspective:

It is notable that understanding and interpreting traffic patterns, especially for human-driven vehicles, involves behavioral and psychological factors. Some studies study traffic safety from a deeper perspective of assessing driver’s cognition and mental capacity. For example, \cite{433,434} use inventories or questionnaires to investigate the correlation between driving quality and general personality.
Table 14 Some popular traffic-related datasets.

| Category                        | Dataset                                                      | Size                                      | Features                                                                 |
|---------------------------------|--------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------|
| Naturalistic Study              | The 100-Car Naturalistic Driving Study (2006)               | 2,000,000 vehicle miles; 43,000 hours of data | Contains extreme cases of driving behavior; Event-based database         |
|                                 | SHRP 2 NDS dataset (2012)                                  | more than 2 PB of continuous naturalistic driving data | Multi-view video outputs; collected during a 3-year period from more than 3,500 participants, aged 16-98 |
|                                 | MOR-UAV/2020                                              | 89,783 moving object instances; 10,948 various scenarios; | Tasks for moving and non-moving objects; varied conditions of the environment; labeled by axis-aligned bounding boxes |
|                                 | Stanford Drone Dataset (2016)                             | ≥100 top-view scenes; 20,000 targets annotated trajectories and id | Real-world scenario-based; varied types of targets                       |
|                                 | The highD dataset (2018)                                  | 16. Shons of measurements for 110,000 vehicles from 6 locations | Real-world scenario-based; focus on highways contain naturalistic behavior of road users |
|                                 | The inD dataset (2019)                                    | contains more than 11500 road users; 10 hours measurement | Successor of highD; focus on intersections                               |
|                                 | Video-based Traffic events (Crash)                         | UC-F-Crimes (2018)                        | Includes realistic anomalies such as fighting, road crash; No spatial annotation |
|                                 | DAD (2016)                                                 | 1900 sequences, 128 hours                 |                                                                          |
|                                 | CADP (2018)                                                | 1416 sequences, 5.2 hours                 | Include multi-crash sequences                                             |
|                                 | NVIDIA AI CITY challenge (2018)                           | Track 1: 31 videos (about 9 hours in total) | Track 1: vehicle counting                                               |
|                                 |                                                           | Track 2: 56,277 images                    | Track 2: vehicle Re-id                                                   |
|                                 |                                                           | Track 3: 215.03 minutes videos            | Track 3: vehicle tracking                                               |
|                                 |                                                           | Track 4: 200 15mins videos 30fps          | Track 4: Abnormal detection                                             |
|                                 | Fine-grained Vehicle Classification                       | Standford Cars [105]                     | The cars images are taken from many angles; classes typically based on Make, Model, Year |
|                                 | COMPCARS [106]                                            | 16,185 images of 196 classes of cars      |                                                                          |
|                                 |                                                              | Web-nature: 136,726 (entire car)+27,618 (car parts), 163 car makes with 1,716 car models; Surveillance-nature: 5000 car images | Web-nature data are labeled with bounding boxes and viewpoints; the surveillance-nature images are in the front view |

IET Research Journals, pp. 1-30
© The Institution of Engineering and Technology 2015
8.2 Relation to AVs

AI platforms have been utilized in recent years to build AVs. AVs use different technologies such as regular cameras, radar, optical radar, and GPS, along with computer vision and learning methods, to realize autonomous driving. In 2015, Tesla started to commercialize ‘Autopilot’ features in its cars, and soon afterward, other manufacturers joined the race. Currently, there are over 250 autonomous vehicle companies, including automakers, technology providers, services providers, and tech start-ups, that are taking serious steps to make self-driven or driver-less cars a reality. According to [445], the top five autonomous vehicle companies are Waymo, General Motors’ Cruise division, Tesla, Baidu, and Argo AI (Ford Motor).

The official level classification system for autonomous driving proposed by the National Highway Traffic Safety Administration (NHTSA) defines the five autonomy levels including [448]:

- **Level 0 (no automation):** The human driver has full authority to operate the car, and can be assisted by warning and protection systems during driving.
- **Level 1 (driving support/hands on):** Provide driving support for one operation of the pay-off reel and acceleration/deceleration through the driving environment, and the human driver operates other driving actions.
- **Level 2 (partial automation/hands off):** The vehicle can fully control the car by acceleration, braking, and steering; however, the driver should keep monitoring the environment and be prepared to intervene immediately at any time.
- **Level 3 (conditional automation/eyes off):** The unmanned driving system completes all driving operations. According to the system request, the human driver provides an appropriate response.
- **Level 4 (highly automated/ mind off):** The unmanned driving system completes all driving operations. According to system requests, human drivers do not necessarily need to respond to all system requests and limit road and environmental conditions.
- **Level 5 (fully automated):** The unmanned driving system completes all driving operations. Human drivers take over when possible. Drive on all roads and environmental conditions.

One of the biggest questions surrounding AVs is: How safe is safe enough? This is a controversial issue. Statistics [449] show that AVs are very safe, and AV-caused crashes are rare. It is still hard to get an accurate self-driving car death toll since some AVs are still developing or testing. Tesla claims that their ADS processing time is four times safer than regular cars, while operating in Autopilot mode. Their estimation is one fatality per 320 million miles driven. However, people still have doubts about the safety performance of autonomous vehicles. As reported in [450], a self-driving Uber car (a test vehicle) hit and killed a woman in Arizona partly because autonomous vehicles calculate safety metrics and use AI methods to react to safety issues appropriately. However, they see the events and the traffic from their own perspective. Developing network-level safety metrics can provide a holistic assessment of the overall traffic safety level when AVs join the regular traffic flows.

8.3 Relation to Vehicle Insurance Evaluation

The primary use of auto insurance is to provide financial protection against physical damage or bodily injury resulting from traffic collisions and against liability that could also arise from incidents in a vehicle. In general, the insurance company calculates the insurance premium based on different factors that affect the customer’s chance of being involved in a crash. The factors include age, car type, driving history, where the customers live, and other factors. Therefore, developing safety models that integrate these factors and predict crash rates can be used by insurance companies for more accurate estimations [452].

Recently, a fully autonomous vehicle insurance pricing [453] system was built based on such information. Moreover, a system [454] that uses vehicle operation data collected via mobile devices for vehicle insurance pricing. Studies on traffic safety metrics could provide more precise and targeted information to these systems to determine a more flexible and reasonable insurance policy for both customers and companies.

8.4 Crowd-sourcing for traffic analysis and driving safety

Noting the wide use of smartphones with accurate positioning systems, crowd-sourcing can be used to collect vehicle motion trajectories, crash incidents, etc., for network-level safety analysis. For instance, [455] uses crowd-sourcing to provide an early warning to drivers regarding road safety hazards due to construction work, defective street cuts, bumps, etc., using a cellphone-based App and embedded accelerometer readings. Indeed, part of the navigation software features in GoogleMAP and Waze is based on crowdsourcing [456]. However, most crowd-sourcing methods use the raw data; therefore, using more elegant DL-enabled safety analysis algorithms can substantially enhance the efficacy of the produces advisory messages.

8.5 Vehicular Edge Computing

When the computation load is beyond the local servers’ power, cloud computing is used. However, cloud computing may cause intolerable computation delays and interruptions due to networking delays. For these scenarios, edge computing is adopted by running computations on servers located at the network edge, to mitigate networking delays [457]. For instance, the idea of offloading heavy computations tasks by AVs to RSUs is proposed in [458]. These distributed nodes can share their computation ability that decentralizes the stress of the large network and reduces bandwidth consumption and response time among various servers and end-users. Users are allowed to access the physically closest servers to operate their real-time applications with good Quality of Service (QoS) and low latency [459].

Recently, the idea of deploying Vehicular Edge Computing (VEC) in Vehicular Ad Hoc Networks (VANETs) is proposed (Fig.14). Conventionally, VEC consists of three layers [460]: User Layer, Edge Layers, and Cloud Layer. The User Layer is composed of Vehicular Terminals (VTs), mainly as smart vehicles. Terminals perform sensing the environment, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, limited computation tasks, and part of the storage. Edge Layer mainly consists of Road-Side Units (RSUs), handle caching, computation, offloading and deliver low-latency diverse service. RSUs are widely distributed on the roads and have more powerful computation capacity and larger storage space. RSUs deploy wireless communication protocols (such as WLAN, 3GPP, 4G, 5G, etc.) to guarantee reliable links that connect the Cloud and User Layers. It should be noted that the majority of data computation and storage is carried out in this layer. Cloud Layer, which consisted of cloud servers, has the greatest computation capacity and the highest storage capacity. This layer is designed to support centralized control, data aggregation, and global management and optimization. These tasks often are complicated and time-consuming; however, latency-sensitive tasks should not be included. To achieve enhanced performance, VECs use several techniques, including Content caching, redundancy (the content the terminal potentially requests), and history content on the edge servers in the proximity of users. This reduces the traffic flow across the whole network and enhances the user experience [461][462].

In order to solve computation issues, task Offloading is used to transfer the excessive computations from local servers (potentially in vehicles computers) to the closest RSUs [455][464][465]. Software-Defined Networking (SDN) [460] separates the control plane and...
data plane that allows easy switches reconfiguration for flexible network management. It addresses a diverse range of problems for traffic safety. It facilitates real-time collecting and processing data from smart vehicles and infrastructures to optimize navigation to avoid congestion and warn the surrounded vehicles to avoid collisions or other incidents. Also, each participating VEC receives the information and is allowed to adjust its strategy (such as Platoon [467]). Moreover, beyond safety-related scenarios [468], VEC also make high traffic demand applications possible, such as video streaming [469, 470]. Augmented Reality (AR) [471], and in-vehicle Infotainment Service (such as online gaming) [472]. The current challenges and active research problems in VEC include transmission reliability, service capacity, integration, scalability, security and user privacy, and economic considerations.

### 6.6 Key Challenges and Issues

Although the use of DL methods for different aspects of driving safety analysis gains more momentum every year, there still exist numerous challenges and issues to be addressed.

**Modeling complexity:** Developing data-driven and mathematical frameworks to model traffic flow and safety risks remains a challenging issue. Part of the reasons is the difficulty of modeling the environment, a huge number of factors with interlaced roles, the impact of human factors and cognition, and the relations between different vehicles, which creates a complex system to model. Several studies tried to model complicated traffic conditions. Some studies including [473, 477] applied conventional methods to create surrogate safety models, while other works such as [474, 478, 485] deployed statistical models to analyze traffic data. More recent works use DL for modeling purposes. For instance, DL frameworks are utilized by [486, 492] for traffic prediction, by [491] for vehicle behaviour prediction, and by [415, 493, 493] for traffic classification.

Although these works achieve excellent performance, they only modeled some specific scenarios (e.g., intersection, ramp merge, etc.). A general model that can briefly represent real road traffic conditions, perhaps by integrating the existing models, is still considered an open research problem.

**Labor cost:** Labor cost is another limiting factor for developing learning-based traffic modeling frameworks. For instance, the 100-Car naturalistic driving study (2006) [486], one of the most popular traffic datasets, collected data from about 100 cars totally driving approximately 2,000,000 miles and 43,000 hours, and took about four years to complete. We agree that not every problem needs this huge of dataset; however, the traffic-related work needs this kind of data to create reliable automated analysis frameworks. An alternative solution is using unsupervised learning and data augmentation methods to mitigate the need for massive annotated datasets.

**Algorithm reliability and efficiency:** Although DL methods have shown superior performance in simple image-based tasks such as object recognition, object tracking, and instance segmentation, they can be prohibitively unreliable when it comes to modeling multi-factor and multi-faceted phenomena that involve extracting complicated tasks by processing videos in real-time. It is known that many DL-based algorithms deployed by traffic systems and AVs, such as [490, 492] for object detection, [494, 495] for stereo matching, [496, 497] for optical flow. With recent advances in high-computational processing platforms, this issue seems to be mitigated day by day. More and more studies provide evidence for AVs’ safety and accuracy of DL-based traffic monitoring systems to alleviate cultural barriers in using DL-powered technology and replacing humans with computers. However, still, some manufacturers like Tesla emphasize their products still require active driver supervision [498].

For conventional traffic analysis, the use of DL-based algorithms is not critical and does not directly compromize safety. However, the more widespread use of these algorithms can provide a better understanding of traffic safety in general. It can improve traffic risks by offering design hints to transportation infrastructures and real-time warnings to the cars on the road.

**Equipment support:** Scarcity of data and the lack of sufficient monitoring infrastructure is another drawback. For instance, in the state of Arizona, more than 441 public cameras are used by the ADOT to monitor traffic [499], but there are a total of 144,959 miles [500]. This means that a lot of roads have not yet been fully covered by surveillance cameras.

**Privacy and secrecy:** Another barrier of common spread use of DL-algorithm for safety analysis is that traffic video contains personal information such as human face and plate numbers, which raises privacy concerns. We believe that publishing more traffic video repositories with removed Personal Identifiable Information (PII) could substantially increase the rate of discovery without compromising people’s safety.

**Naturalistic driving data:** Although some naturalistic driving data were collected for traffic flow and transportation research, such as Next Generation Simulation (NGSIM) data [434], open-source naturalistic driving data for safety evaluation of vehicles are rarely reported. Since for vehicle safety analysis, some quantitative data of each vehicle are required, such as location, speed, acceleration, and heading angles, advanced technology need to be developed and applied to accurately obtain these measurements. When cameras are applied, computer vision and related to objective identification and tracking algorithms, based on ML or DL, need to be specifically developed, especially towards real-time processing. Some recent available naturalistic driving data obtained by drones, such as LevelX [422, 423], can provide necessary pre-processed data for safety analysis purposes. However, these data are processed offline and expensive.

### 9 Conclusion

This paper reviewed DL methods that can be used for different aspects of video-based traffic safety analysis. We reviewed methods, tools, and datasets that are recently developed by the research community and industry. We highlighted key achievements and mentioned areas that need further investigation. For example, we enumerate areas that require more advanced tools and also collecting well-annotated datasets. Some examples include but not limited to the need for developing DL algorithms, tools, and datasets for aerial traffic monitoring systems, more advanced video-based action recognition systems, integrative analysis of multi-modal traffic image and sensor data, extending individual safety metrics into network-level safety metrics, formal ways to develop traffic metric distributions, finding associations between network-level safety metrics and crash rate, developing online safety metric extraction tools, and developing end-to-end frameworks to translate safety risks into traffic advisory messages. We also made connections to closely related research areas, including AVs, Crowd-sourcing for traffic analysis, and driver’s behavioral patterns and psychological profiles, and the insurance industry. Our hope is that this paper will help computer scientists solve traffic safety problems, particularly areas that need further investigation. This paper also aimed to help traffic engineers and personnel to identify and use existing open-source tools for their problems.

### 10 Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### 11 Acknowledgment

We would like to thank Drs. Jason Pacheco, Larry Head, and Junsuo Qu from their thoughtful comments on this paper. Special thanks go to Greg Leeming from Intel for his insightful comments and continued support of this project. We are grateful to Arizona Commerce...
Žbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. The journal of machine learning research. 2016;17(1):2287–2318.

Luo W, Schwing AG, Urtasun R. Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. p. 5695–5703.

Fischer P, Dosovitskiy A, Ilg E, Hausser P, Hazrbaç C, Golkov V, et al. FlowNet: Learning Optical Flow with Convolutional Networks. arXiv. 2015.

Ranjan A, Black MJ. Optical Flow Estimation Using a Spatial Pyramid Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017:2720–2729.

Available from: https://www.tesla.com/autopilot

Available from: https://az511.gov/

Available from: https://www.fhwa.dot.gov/policyinformation/statistics/2017/hm60.cfm