Vehicle Type and Speed Pattern Recognition

Ashwathi Ashok Tharun Subramanya Vedhesh Aandiyappan Saravanan
Department of Computer and Information Science
University of Michigan
Dearborn, MI, USA
ashh@umich.edu, tharuns@umich.edu, vedas@umich.edu

***Abstract*—Traffic monitoring is important for understanding the road usage and improving safety. Many existing system depends on the physical sensors which are expensive and difficult to maintain. In this project we developed a video based method to detect vehicles, identify their types and estimate the speed using traffic camera footage. Moving vehicles are separated from the background using background subtraction and contour filtering. Using a centroid based tracking approach, each vehicle is tracked across frames. For classification, a convolutional neural network is used to distinguish vehicles from non-vehicles and to classify vehicles into categories such as car, bus, truck and motorcycle. Vehicle speed is estimated by measuring pixels displacement across frames and converting it to units using the video frame rate and lane width calibration. The system was tested on multiple traffic videos with different lighting and traffic conditions. The results show that the approach can accurately detect and track vehicles, provide reliable vehicle type classification and estimate vehicle speeds with reasonable consistency. This work proves that camera based methods can be an effective alternative for traffic analysis tasks.**

I. Introduction

Road traffic monitoring is essential for improving the road safety. Information such as vehicle count, type and speed helps in traffic planning and identifying unsafe driving behavior. Many existing system rely on physical sensors placed on or near the road. Although these systems can be accurate, they are expensive to install, difficult to maintain and not always practical for large areas. This project focuses on studying how pattern recognition and neural network techniques can be used to analyze traffic videos. The aim is to build a system that can detect vehicles, follow them across frames, identify their types and estimate their speeds using only video input. By combining basic image processing methods with learning based classification, this works explores a practical approach to traffic analysis using camera footage and highlights the challenges involved in the real time video data.

II. Related Work

Vision based traffic analysis has been explored in many studies for tasks such as vehicle detection, tracking, classification and speed estimation. A detailed overview of camera based vehicle speed estimation methods is presented in [1]. This survey describes common processing steps used in video based systems and points out that converting pixel movement into real world distance is one of the main difficulties especially when the camera is not fully calibrated.

Many studies have shown that the vehicle speed can be estimated using just one fixed camera. These methods usually combine vehicle detection, tracking, and basic information about the road. Bell et al. used vehicle detection along with SORT tracking and applied homography to convert image positions into real road distances for calculating the speed [2]. Revaud et al. also studied speed estimation using a single camera and explained that camera calibration and the shape of the road have a strong effect on how accurate the speed results are [3]. Earlier, Temiz et al. estimated vehicle speed using an uncalibrated camera by measuring known distances on the road. Their work showed that it is possible to estimate speed without using extra sensors, but it requires careful setup and accurate distance measurements in advance [4].

Tracking is needed to make sure the same vehicle is followed from one frame to the next so its movement can be measured correctly. SORT is a commonly used tracking method that combines a Kalman filter with the Hungarian algorithm to match detections across frames. It is simple and fast which makes it suitable for real time use when detections are available in every frame [5]. DeepSORT builds on this idea by adding appearance information which helps keep the correct vehicle identity when vehicles overlap or when traffic is dense [6].

Vehicle detection in traffic videos is often done using object detection models because they can detect vehicles quickly while still giving good accuracy. YOLOv2, also called YOLO9000, is commonly used for real-time detection and has been used in many traffic analysis systems that combine detection with tracking [7]. In fixed camera setups, background subtraction is also widely used to detect moving vehicles. Zivkovic and van der Heijden proposed an adaptive Gaussian mixture model that updates pixel values over time, and this method is still commonly used in surveillance video processing [8].

Vehicle classification has been done using both traditional methods and newer learning based methods. HOG is a commonly used feature descriptor and is often combined with a linear SVM for object recognition in earlier pattern recognition work [9]. In more recent studies, convolutional neural networks have been used to classify vehicles directly from images, even when the video quality is low and fine details are not very clear [10]. These studies show that learning

based models handle changes in lighting and camera angle better while feature based methods are still useful as a basic comparison.

In traffic videos, it is often difficult to separate moving vehicles from the background. Problems like shadows, changes in lighting, and leftover objects in the scene can easily lead to false detections. Cucchiara et al. discussed these issues and explained how shadows and ghost objects interfere with moving object detection in surveillance footage [11]. This work makes it clear that background modeling must be handled carefully, especially for outdoor videos where lighting conditions keep changing.

Motion between frames has also been estimated using optical flow. The Lucas–Kanade method tracks motion by comparing pixel intensity changes between consecutive frames, and it has been used for vehicle tracking and speed estimation in some systems [12]. The method itself is simple, but it is sensitive to noise, which means preprocessing is needed to avoid unstable or incorrect motion estimates.

Many traffic models use OpenCV because it already has most of the tools needed for video processing. Background subtraction, feature extraction, tracking and basic image processing were done using functions thus they are readily available, thus it make it easier to test and build a complete traffic monitoring pipeline [13].

With the use of deep learning, object recognition performance has improved compared to older methods. Simonyan and Zisserman showed that deeper convolutional networks can achieve better classification accuracy by learning more detailed features from images [14]. Krizhevsky et al. also showed that deep CNNs trained on large datasets perform better than traditional handcrafted feature methods which influenced the use of CNNs for vehicle classification tasks [15].

Faster R-CNN and Fast R-CNN use a region-based approach where possible object regions are identified first and then classified [16], [17]. These methods usually give good detection accuracy but they are quite heavy. Because of this, they are not always suitable for real-time traffic analysis, especially when faster or motion-based methods are available.

Background subtraction is often used in surveillance videos to detect moving objects. Piccardi reviewed different background modeling techniques and explained where they work well and where they fail in real scenes [18]. Feature-based methods such as SURF have also been used for detection and tracking since they are robust to scale and rotation changes, but they require more computation compared to simpler features [19].

Datasets collected from busy urban scenes also show common detection challenges. CityPersons contains many crowded scenes with occlusions and different camera viewpoints [20]. Although it was created for pedestrian detection, it highlights issues such as dense scenes, partial visibility, and annotation difficulty, which are also relevant in traffic video analysis.

III. Dataset

Various datasets were used to build and test this model to support vehicle detection, tracking, classification and speed estimation. Separate datasets were selected for image based model training and also a video based system evaluation in order to monitor both controlled and real world scenarios.

A. Vehicle vs. Non-Vehicle Image Dataset

A public dataset from Kaggle was used to train a binary classifier for vehicle detection on road with two classes vehicle (8792) and non-vehicle (8968). Vehicle images includes vehicles from different view points meanwhile non vehicles images has background scenes like roads, building, and empty environments.

All images sizes were altered to a fixed resolution and normalized before training. This dataset provide a balanced and diverse. The trained binary classifier was later used to test and validate motion based detection in video frames and reduce false positive.

B. Traffic Video Datasets for Tracking and Speed Estimation

Two traffic video datasets were used to vehicle tracking and speed estimation performance.

The first video dataset was from Kaggle consists of traffic footage (mp4) captured using a fixed roadside camera. The video in dataset shows vehicles moving in a roadway under a stable lighting condition. This dataset was used to evaluate vehicle detection, multi-object tracking and speed estimation and as the camera view point is fixed the vehicle motion is smooth and predictable.

Fig. 1: Sample frame from the input traffic video used for analysis.

The second video dataset was sourced from publicly available short video clips (mp4) from social media for testing it in real world scenario. This video contains multiple types of vehicles with traffic flowing in different patterns. It was used to check the speed estimation pipeline under less controlled conditions too.

C. Multi-Class Vehicle Classification Dataset

Multi-class Vehicle image dataset from Kaggle was used to train a CNN for vehicle type detection. The dataset has labeled images (8219) from four vehicle categories like car, bus, truck and motorcycle. Bounding box annotations were used to crop the images focusing the vehicle region and were resized and normalized before training the model. The classification model provides reasonably stable predictions when combined with confidence thresholding and temporal voting across frames.

IV. Methodology

The methodology followed in this project is a video processing pipeline to detect the vehicles, track their movement, classify their type and estimate their speed. The system takes the traffic video footage as input and processes it frame by frame. Starting with preprocessing the video data to improve quality and reduce noise. Vehicles are then detected using motion based techniques and then tracking individual vehicles across consecutive frames. Once reliable tracks are obtained, each vehicle is classified into its corresponding type using a trained convolutional neural network. Finally the movement of tracked vehicles is analyzed over time to estimate their speed which is further refined to obtain stable and meaningful results. Each stage of the pipeline is tested independently before being integrated into the complete system.

A. Data Collection

Multiple datasets were used at different stages of the system. A publicly available image dataset from Kaggle was used for vehicle detection, which includes both vehicle and non-vehicle images. The dataset is balanced and was suitable for training a binary classifier to distinguish vehicles from background regions [10].

Real traffic videos were used for tracking and speed estimation. These include motorway footage and bridge-view traffic videos where multiple vehicles are visible at the same time. The videos contain variations in lighting conditions and camera viewpoints, which makes them useful for checking whether tracking remains stable and whether speed estimation behaves consistently in real traffic scenarios [1], [2].

For vehicle type classification, a labeled vehicle detection dataset was used. Although the dataset contains several vehicle categories, only four classes were considered in this work: car, bus, truck, and motorcycle. Vehicle regions were cropped directly from the labeled images so that the dataset contains only vehicle patches. These cropped images were then used to train the vehicle classification model, following approaches commonly used in vehicle recognition studies [9], [10].

B. Data Preprocessing

Image data and video data were handled differently since they are used for different parts of the system. Doing this avoided issues during training and made the video processing more stable.

All images used for training were resized to 128×128 to make sure the network always receives the same input size. Pixel values were divided by 255 to bring them into the range $[0, 1]$. The dataset was split into training and validation sets using a fixed random seed so the split remains the same across runs. Caching, shuffling, and prefetching were used to speed up training and avoid effects caused by loading images in a fixed order [10], [15].

In the video frames, only the lower portion was processed since vehicles appear on the road area. Background subtraction was applied using the MOG2 method to separate moving vehicles from the background [8], [18]. Shadow pixels produced by the background model were removed. Median filtering was applied to reduce noise, and thresholding was used to obtain a clear foreground mask.

After foreground extraction, opening and dilation were applied to remove small noisy regions and connect broken vehicle shapes. Contours were then filtered using area, bounding box size, and aspect ratio so that non-vehicle regions such as lane markings and shadows were removed before tracking [11], [18].

For vehicle type classification, bounding boxes from labeled images were used to crop vehicle regions. These cropped patches were resized to match the CNN input size and saved based on vehicle class. Using cropped vehicle regions instead of full images helped the classifier focus only on vehicle appearance rather than background information [9], [10].

C. Vehicle Detection

Vehicle detection in this project is carried out using both motion-based processing and a binary convolutional neural network. These two steps are used together so that moving vehicles can be identified while reducing background noise and unwanted motion.

The first step involves detecting moving regions using background subtraction. The MOG2 background subtraction method is applied to a selected region of interest in each video frame to separate moving foreground objects from the static background. This method works well for traffic videos where the camera position does not change. Shadow regions produced by the background model are removed so that they do not get detected as vehicles, which helps reduce false detections in outdoor scenes [8], [18].

After the foreground regions are obtained, contour detection is used to find candidate moving objects. Several filtering steps are applied at this stage to remove unwanted detections. Very small contours are ignored since they are usually caused by noise or distant motion. Bounding boxes are further filtered using width, height, and aspect ratio checks to remove objects such as lane markings, shadows, and irregular shapes that do not resemble vehicles [11].

To further improve detection reliability, each remaining bounding box is passed through a trained binary convolutional

neural network that classifies the region as either a vehicle or a non-vehicle. The bounding box region is cropped from the original frame, resized to match the CNN input size, and normalized before being evaluated by the model. Only detections with a high confidence score are kept for further processing. This additional classification step helps reduce false positives and ensures that only valid vehicle regions are forwarded to the tracking and speed estimation stages [9], [10].

D. Vehicle Tracking

After vehicles are detected in each frame, they are tracked across consecutive frames so that the same vehicle can be followed over time. Tracking allows the movement of each vehicle to be observed continuously, which is required later for estimating speed.

A centroid-based tracking method is used in this work. For every detected vehicle, the center of the bounding box is calculated and treated as the vehicle's centroid. These centroid points are compared between consecutive frames, and matching is done based on how close the current centroid position is to the previous one. The Euclidean distance between centroid positions is used to associate new detections with existing tracks, which is similar to the idea used in simple real-time tracking methods such as SORT [5].

Each tracked vehicle is assigned a unique ID, and the history of its centroid positions is stored. When a detection matches an existing track, the centroid location and bounding box information are updated. If a track does not receive a matching detection for a few frames, it is temporarily marked as lost. Tracks that remain unmatched beyond a fixed threshold are removed so that objects that have left the scene or false detections are not tracked further [5].

To keep the tracking results stable, only tracks that appear consistently across multiple frames are treated as valid. Tracks that exist for only a short duration are ignored, since they are often caused by noise, shadows, or partial detections. This helps reduce sudden changes in vehicle IDs and improves the reliability of tracking, which is important for later vehicle classification and speed estimation. Similar ideas are also used in improved tracking methods such as DeepSORT to reduce identity switches in crowded scenes [6].

E. Vehicle Classification

After tracking, each vehicle is assigned a type based on how it looks in the video frames. The vehicle types considered in this work are car, bus, truck, and motorcycle.

A convolutional neural network trained on cropped vehicle images is used for classification. For every tracked vehicle, the bounding box region is cropped from the video frame and resized to 128×128 pixels. The cropped image is normalized and passed through the trained multi-class CNN, which produces probability scores for the four vehicle classes. CNN-based models have been commonly used for vehicle classification in surveillance videos and similar datasets [10], [15].

Predictions are collected across multiple frames instead of relying on a single frame. Only predictions with high confidence are considered, and a simple voting method is used to store the predicted classes over time for each vehicle track. After enough confident predictions are gathered, the class with the highest count is assigned as the final vehicle type. This helps reduce errors caused by partial views, blur, or short-term noise in individual frames [10].

A few additional checks are applied to handle obvious mistakes. For instance, very large bounding boxes are unlikely to represent motorcycles and are corrected when such cases appear. Using confidence thresholds, frame-based voting, and simple rule checks together helps keep the classification results stable. The final vehicle type labels are then used in the later speed estimation and visualization steps.

F. Speed Estimation

Vehicle speed is calculated using the movement of tracked vehicles across frames. For each vehicle, the stored centroid positions from consecutive frames are used to measure how far the vehicle moves over time.

Centroid displacement is calculated over a short window of frames. Using the frame rate of the video, this displacement is converted into speed measured in pixels per second. Speed values are computed only for tracks that exist for multiple frames so that short or unstable tracks do not affect the results. This type of frame-based motion calculation is commonly used in monocular speed estimation approaches [1], [2].

The raw speed values are not always smooth due to small tracking errors or noise. To handle this, a moving average filter is applied to the speed values over time. This reduces sudden spikes and produces smoother speed curves for each tracked vehicle [4].

Pixel-based speed values are converted into real-world units using a simple calibration step. A known distance on the road, such as lane width, is measured in pixels from the video frame. This measurement is used to compute a pixel-to-meter conversion factor. Using this factor, speeds are converted from pixels per second to meters per second and then to miles per hour (mph), following common vision-based speed estimation practices [1], [3].

The estimated speeds are saved along with vehicle IDs and frame timestamps. Speed values are also displayed directly on the video frames, which allows visual checking of the results while keeping the numerical values available for later analysis.

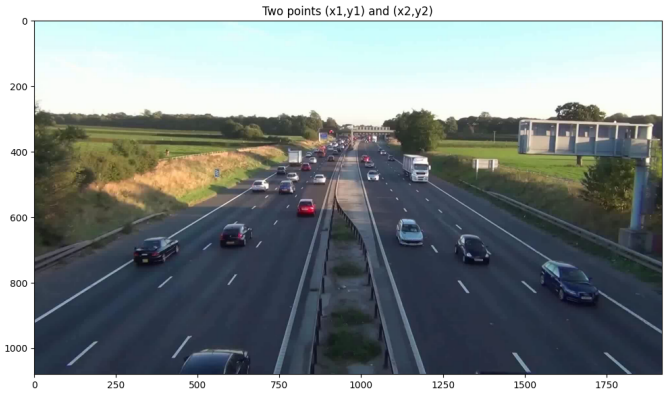

Fig. 2: Reference points selected in the scene for approximate lane-width calibration.

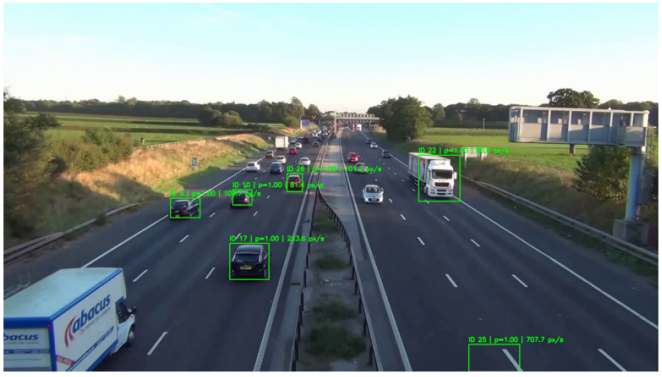

Fig. 3: Detected vehicles with persistent track IDs and estimated speeds overlaid on the video frame.

V. Results

The proposed system was tested on multiple traffic videos captured using fixed roadside cameras. The results focus on vehicle detection accuracy, tracking stability, vehicle type classification, and speed estimation consistency.

For vehicle placement, background subtraction combined with contour filtering was able to identify moving vehicles reliably in most frames. Small objects, shadows, and road markings were largely removed using area and aspect ratio constraints. The detection stage performed well under normal lighting conditions and moderate traffic density, though performance decreased slightly in scenes with heavy shadows or overlapping vehicles.

Tracking results show that the centroid-based tracking method was able to maintain consistent vehicle identities across frames for most vehicles. Stable tracks were observed for vehicles moving smoothly through the scene. Short track interruptions occurred when vehicles were partially occluded or moved very close to each other, but these cases were limited. The raw vs smooth speed estimations, reduces the size of raw speed estimates with smoothed values from a vehicle in a single track. The smoothing process reduces the

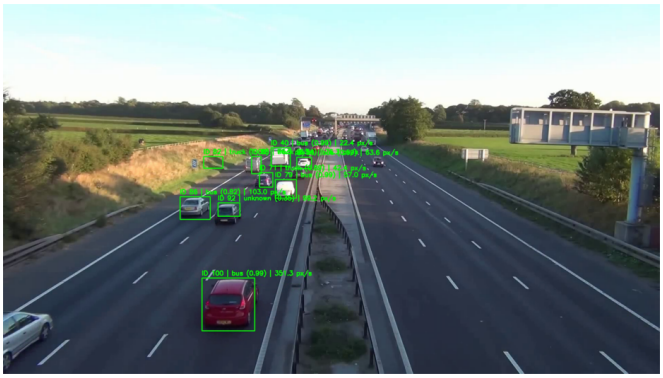

Fig. 4: Vehicle type predictions overlaid on tracked objects using the multi-class CNN with temporal voting.

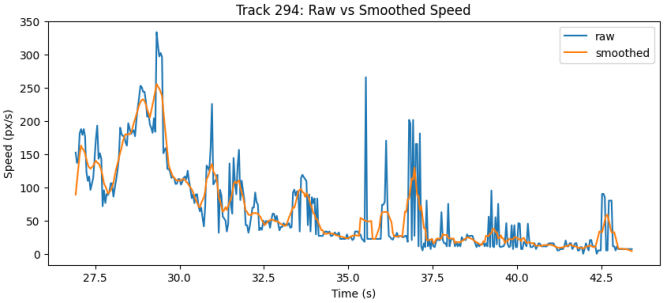

Fig. 5: Example of raw vs. smoothed speed estimates for a single tracked vehicle.

noise that was caused by small tracking errors and produces a more stable speed profile. Vehicle classification was evaluated using a convolutional neural network trained on cropped vehicle images. The binary classifier achieved high accuracy in separating vehicles from non-vehicle objects. The multi-class classifier was able to distinguish between cars, buses, trucks, and motorcycles with good consistency. Classification stability improved when predictions from multiple frames were combined using a voting approach instead of relying on single-frame predictions.

Speed estimation results were analyzed by plotting speed values over time for individual vehicle tracks. Raw speed values showed fluctuations due to small variations in centroid positions between frames. After applying a moving average filter, the estimated speeds became smoother and more consistent. Speed values were converted from pixels per second to real-world units using a reference lane width, producing reasonable speed ranges for highway traffic. The final output videos clearly display vehicle IDs, vehicle types, and estimated speeds, making it easy to visually verify the results.

The calibrated speed graph in mph shows the calculated vehicle speeds after applying pixel-to-meter calibration. The resulting speed values are well within a reasonable range for highway traffic, indicating consistent estimation behavior.

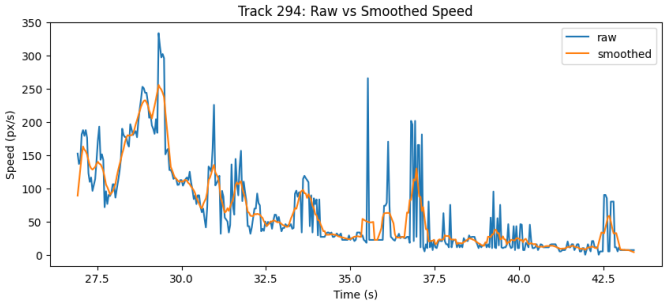

Fig. 6: calibrated speed in mph.

VI. Discussion and Future Work

The results demonstrate that a camera-based approach can be used to detect vehicles, track their movement, classify their types, and estimate their speeds using a single fixed camera. The system performs well in controlled traffic scenes and shows stable behavior when vehicles are clearly visible and moving in predictable paths.

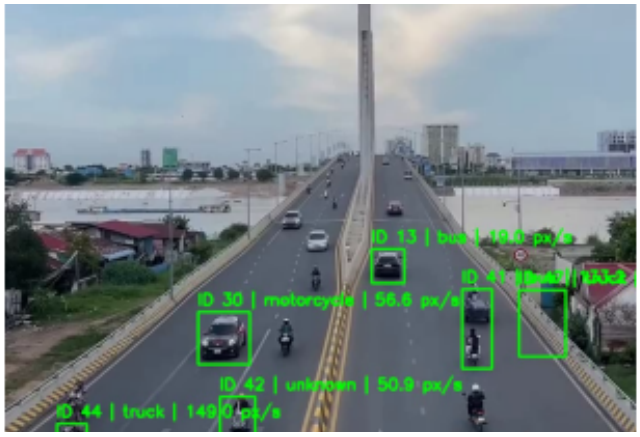

Fig. 7: Generalization test on a second traffic video with different viewpoint and vehicle scale.

One limitation of the current approach is its dependence on background subtraction, which can be sensitive to lighting changes, shadows, and camera vibration. Sudden illumination changes may introduce noise in the foreground mask and affect detection quality. In addition, the centroid-based tracking method may struggle in dense traffic scenarios where vehicles overlap or change lanes frequently.

Speed estimation accuracy depends on manual measurement of reference distances in the scene. Any error in selecting the lane width or reference points directly affects the final speed values. This limits the system's ability to generalize across different camera setups without additional calibration.

Future work could focus on improving detection robustness by using learning-based object detectors instead of background subtraction. Tracking performance could be improved by incorporating motion prediction models or appearance-based features to reduce identity switches. Automatic camera calibration techniques could also be explored to reduce reliance on manual measurements for speed estimation. Extending the system to handle nighttime videos and higher traffic density would further improve its practical usability.

VII. Conclusion

This project presented a video-based system for vehicle detection, tracking, classification, and speed estimation using traffic camera footage. By combining classical image processing techniques with learning-based classification models, the system is able to analyze traffic scenes using a single fixed camera.

The results show that vehicles can be detected and tracked with reasonable reliability, vehicle types can be classified into common categories, and speeds can be estimated using pixel displacement and simple geometric references. While the system has some limitations related to lighting conditions and manual calibration, it demonstrates that camera-based traffic analysis can be an effective alternative to sensor-based methods for certain applications.

Overall, this work highlights a practical approach to traffic monitoring using video data and provides a foundation for further improvements in accuracy and robustness.

References

[1] D. Fernández-Llorca, A. Hernández Martínez, and I. García Daza, "Vision-based vehicle speed estimation: A survey," *IET Intelligent Transport Systems*, 2021, doi: 10.1049/itr2.12079.

[2] D. Bell, W. Xiao, and P. James, "Accurate vehicle speed estimation from monocular camera," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-2, pp. 419–426, 2020.

[3] J. Revaud, M. Humenberger, and V. Lepetit, "Robust automatic monocular vehicle speed estimation for traffic surveillance," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021.

[4] M. S. Temiz, O. Altun, and S. G. Arica, "Real-time speed estimation from monocular video," in *Proc. International Conference on Signal Processing and Communications*, 2012.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *arXiv preprint arXiv:1602.00763*, 2016.

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *arXiv preprint arXiv:1703.07402*, 2017.

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2017.

[8] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[10] S. Tas, A. K. Ozdemir, and B. Aydin, "Deep learning-based vehicle classification for low-resolution surveillance images," *Sensors*, vol. 22, no. 3, 2022.

[11] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337–1342, 2003.

[12] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 674–679, 1981.

[13] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[17] R. Girshick, "Fast R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015.

[18] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, pp. 3099–3104, 2004.

[19] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[20] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.