Instant-level vehicle speed and traffic density estimation using deep neural network

Obafemi Jinadu, Victor Oludare, Srijith Rajeev, Landry Kezebou, Karen Panetta, et al.

Obafemi Jinadu, Victor Oludare, Srijith Rajeev, Landry Kezebou, Karen Panetta, Sos Agaian, "Instant-level vehicle speed and traffic density estimation using deep neural network," Proc. SPIE 12526, Multimodal Image Exploitation and Learning 2023
, 125260J (15 June 2023); doi: 10.1117/12.2663643

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Instant-Level Vehicle Speed and Traffic Density Estimation Using Deep Neural Network

Obafemi Jinadu[a], Victor Oludare[a], Srijith Rajeev[a], Landry Kezebou[a], Karen Panetta[a], Sos Agaian[b],
[a]Tufts University, Medford, MA, 02155
[b]City University of New York, New York, NY 100161

ABSTRACT

The growing network of highway video surveillance cameras generates an immense amount of data that proves tedious for manual analysis. Automated real-time analysis of such data may provide many solutions, including traffic monitoring, traffic incident detection, and smart-city planning. More specifically, assessing traffic speed and density is critical in determining dynamic traffic conditions and detecting slowdowns, traffic incidents, and traffic alerts. However, despite several advancements, there are numerous challenges in estimating vehicle speed and traffic density, which are integral parts of ITS. Some of these challenges include variations in road networks, illumination constraints, weather, structure occlusion, and vehicle user-driving behavior. To address these issues, this paper proposes a novel deep learning-based framework for instant-level vehicle speed and traffic flow density estimation to effectively harness the potential of existing large-scale highway surveillance cameras to assist in real-time traffic analysis. This is achieved using the state-of-the-art region-based Siamese MOT network, SiamMOT which detects and associates object instances for multi-object tracking (MOT), to accurately estimate instant level vehicle speed in live video feeds. The UA-DETRAC dataset is used to train the speed estimation model. Computer simulations show that the proposed framework a) allows the classifying of traffic density into light, medium, or heavy traffic flows, b) is robust to different types of road networks and illuminations without prior road information, and c) shows good performance when compared to current state-of-the-art methods using adequate performance metrics.

Keywords: vehicle speed estimation, multi-object tracking, vehicle detection, deep learning, traffic density estimation, intelligent transportation system

1. INTRODUCTION

One of the leading causes of road incidents in the United States is speeding. In 2020, the National Highway Traffic Safety Administration (NHTSA) reported 11,258 speed-related causalities, which translates to 29% of total traffic fatalities [1]. Speeding is a type of aggressive driving behavior often due to impatience, anonymity, and traffic congestion [1]. Much work has been done to advance traveler safety in the Intelligent Transportation Systems (ITS) domain; [2] proposed a scalable and efficient algorithm for vehicle model detection; [3] explored video action recognition to tackle the problem of highway incident detection and classification from live surveillance footage by introducing the HWID12 (Highway Incident Detection) dataset; [4] proposed a high precision, deep neural network approach to detecting wrong-way driving (WWD) on highway roads. In maritime border security; [5] proposed the first underwater tracking benchmark dataset; and [6] proposed augmented reality as a tool for 3D navigation.

This work focuses on traffic control with ITS, the potential for ITS to mitigate issues of collisions, congestion, and other road incidents hinges on the ability to harness the large amount of data provided by the vast number of road traffic cameras. Although, traffic video feeds provide sufficient information related to traffic flow and density, weather conditions, and road incidents. They are often installed at high points and capture low-resolution videos with different vehicle scales due to network bandwidth limitations, lack of persistent storage, and privacy concerns, making traffic density estimation challenging. An integral part of any ITS is vehicle speed estimation which is necessary to assess the density of traffic and predict the possibility of an incident and other forms of road anomalies. However, many existing systems are designed for ideal scenarios with sufficient illumination and favorable weather conditions, which do not accurately represent the real-world scenarios that often involve less-than-ideal weather, such as haze, snow, and rain. Therefore, this article selects a dataset that offers diverse and challenging scenarios, which more accurately mirror the real-world conditions for speed and traffic density estimation.

Multimodal Image Exploitation and Learning 2023, edited by Sos S. Agaian, Vijayan K. Asari,
Stephen P. DelMarco, Proc. of SPIE Vol. 12526, 125260J · © 2023 SPIE
0277-786X · doi: 10.1117/12.2663643

This article proposes a framework that extracts features from a tracker, synthesizes pixel features from each tracklet (a unique tracked object) and uses these features to estimate vehicle speed and traffic density. The proposed framework comprises three modules: i) feature extraction & synthesizing module, ii) speed estimation module, and iii) traffic density estimation module which classifies traffic density per image sequence into heavy, medium, or light traffic. The output of the multi-object tracking (MOT) model feeds into the feature extraction module, and pixel speed and pixel distance features are synthesized for each tracklet. These features are input to the speed and traffic density estimation models trained on the UA-DETRAC dataset[7]–[9]. The key contributions of this paper are:

a) A novel framework is proposed for jointly estimating vehicle speed and traffic density on roads in an end-to-end manner that can adapt to any off-the-shelf multi-object tracker.

b) The advantages and limitations of adopting multi-object tracking for motion modeling and linear regression in tandem on instant speed estimation tasks are explored.

The remainder of this paper is organized as follows. The related work is reviewed in detail in Section II. The description of each algorithmic component is covered in Section III. Section IV presents the experimental results. Finally, section V concludes the study.

2. RELATED WORK

Several methods have been proposed to address the problem of vehicle speed estimation. These methods can be categorized into conventional computer vision and machine learning-based methods.

2.1 Conventional approaches

Llorca et al. [10] proposed a novel two-camera-based approach where each camera has different focal lengths and orientations. They used the vehicle's license plate as the reference point to evaluate the relative distance of the vehicles with respect to the cameras. Viet-Hoa et al. [11] proposed a geometric setup using an equilateral triangle as a reference object on the ground. The triangular image is then used to estimate the camera parameters. Camera parameters, together with an optical flow algorithm, are used to approximate the motion vectors of traffic video frames and calculate the moving speed of vehicles of traffic video frames and calculate the moving speed of vehicles. Kanagamalliga and Vasuki [12] proposed a contour-based approach to vehicle tracking. They perform movement estimation and object tracking using optical flow and Gabor features-based contour model. Sandeep et al. [13] adopted a normalized self-adaptive optical flow to estimate the direction of traffic flow and filter out noise using a standard Gaussian filter and self-adaptive window to identify moving object areas. El Bouziady et al.[14] adopted Speed Up Robust Features (SURF) for vehicle detection and used geometric derivatives to get vehicle speed from vehicle depth variation. Ibrahim et al. [15] used an adaptive background subtraction technique for object detection and performed tracking by monitoring the vehicle's entrance and exit of the scene. Speed was measured by counting the number of frames a vehicle takes to exit.

These methods often rely on carefully crafted features for vehicle detection and custom setups for estimating speed, which is often not feasible in real-world scenarios, especially under weather conditions such as rainy, hazy, snowy, or foggy conditions. Another drawback of conventional approaches for vehicle detection is its inability to accurately and continuously detect small and/or occluded vehicles in traffic which could affect the efficiency of the traffic density estimation algorithm. Furthermore, these methods often require a reference point or object to perform measurements which is not feasible for every camera in an ITS network.

2.2 Machine learning-based approach

Modern machine/deep learning methods address these challenges by taking advantage of recent advancements in Convolutional Neural Networks (CNNs) [16] . Most machine learning-based vehicle speed estimation methods comprise i) vehicle detection, ii) vehicle tracking, and iii) speed calculation modules.

The introduction of large-scale datasets for vehicle detection such as UA-DETRAC [8], Boxy [17], EAGLE [18], and VAID [19] have paved the way for the development of highly efficient deep learning based-models for vehicle detection from land and aerial videos. CNN-based object detectors have gained significant improvements for various computer vision-related tasks. Ren et al. [20] proposed Faster R-CNN to provide real-time object detection with Region Proposal Networks (RPN) which simultaneously predicts object bounds and outputs objectiveness scores at each position. Redmon et al. [21] proposed an extremely fast approach for object detection called You Only Look Once (YOLO). A single neural network is used to directly predict bounding boxes and class probabilities in full-sized images in one pass. Wang et al. [22] proposed a High-Resolution Network (HRNet) that maintains high-resolution representations throughout the network process. The high-to-low resolution convolution streams are connected in parallel with repeated information exchange across resolutions.

Several object tracking algorithms have been proposed with various degrees of efficiency. Kalman filters have been adopted for single and multiple objects tracking problems [13], [23]. Correlation filter-based approaches such as KCF [24] and BACF [25] learn discriminative features between the target and the surrounding environment. BACF further models the changes in background and foreground of the object over time, resulting in improved tracking accuracy. Simple Online and Realtime Tracking (SORT)[26], a tracking-by-detection framework where detection results are associated across frames through motion modelling. SiamMOT, a Siamese Multi-object tracking network builds upon SORT and Faster-RCNN using a region-based Siamese tracker for instance-level motion modelling [27] where motion is modelled either implicitly using multi-layer perceptron (MLP) to implicitly estimate motion between two frames or explicitly which includes the usage of a channel-wise cross-correlation map for data association. Specifically, there is a correlation between each location of a search frame feature map with the target frame feature map. This paper adopts the SiamMOT network with explicit motion modelling (EMM).

This data-driven approach to estimating the speed of vehicles often relies on some assumptions, such as i) prior knowledge of the maximum speed limit of the road, ii) static camera positions [28] and iii) road length. The model's input are videos and the vehicle's maximum speed in each video footage for training and the predicted speed is a function of the learned local movement and the maximum speed. However, techniques have been proposed which are invariant to some of these assumptions such as the efficient chained centre network (ECCNet)[29], which is a unified framework for accomplishing vehicle detection, tracking speed estimation in parallel. This paper follows a simple mathematical approach invariant to road length and maximum speed limit of the road, where the pixel-per-second distance traveled by a tracked vehicle is computed and fed into a regression model trained on the UA-DETRAC dataset, and the regression model in training intuitively learns the weights required to map pixels to meters and make instant level speed predictions.

3. PROPOSED METHOD

This work focuses on a data-driven approach to speed and traffic density estimation, where speed estimation is tackled as a regression problem leveraging the output of a MOT as the input to our pipeline. Figure 1 shows the proposed framework. A trained state-of-the-art region-based Siamese Multi-object Tracking network, SiamMOT is used [27]. For detection, the tracking network leverages Faster-RCNN with a standard DLA-34 [30] with feature pyramid [31] as its backbone with an explicit motion modeling (EMM) implementation [27]. Features from each output tracklet of the SiamMOT tracker are extracted, and pixel features are synthesized. These features are input datapoints to the speed estimation and traffic density estimation modules which are machine learning models trained on the UA-DETRAC dataset. Both modules act in parallel to make instant-level predictions in real time.

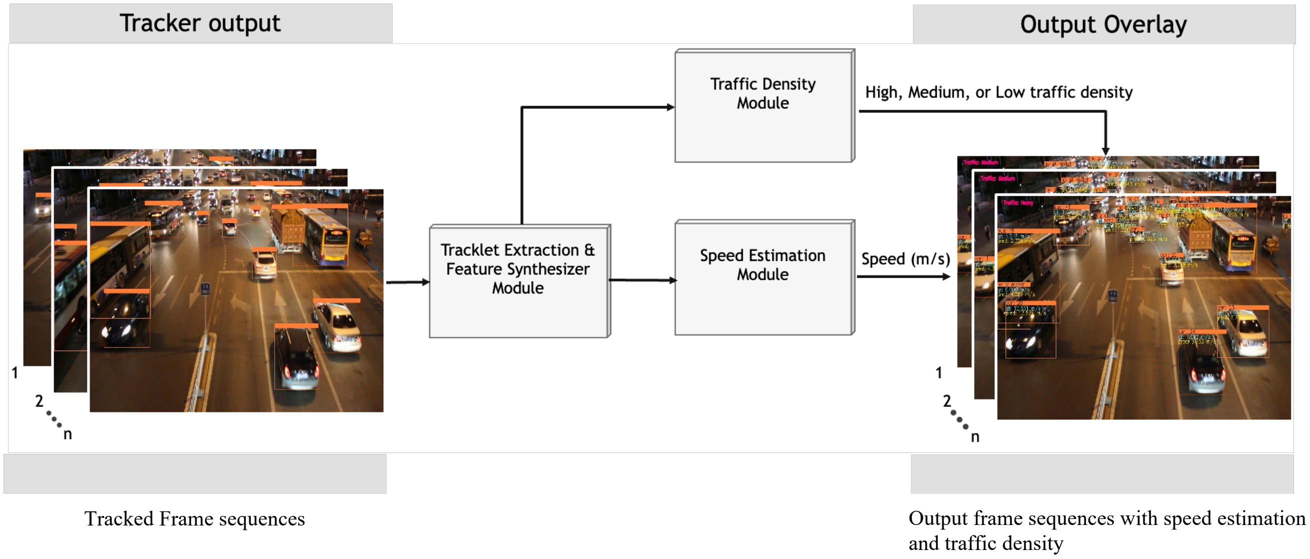

Tracker output
Output Overlay
Traffic Density Module
High, Medium, or Low traffic density
Tracklet Extraction & Feature Synthesizer Module
Speed Estimation Module
Speed (m/s)
1
2
n
1
2
n
Tracked Frame sequences
Output frame sequences with speed estimation and traffic density

Figure 1: Speed and traffic density estimation pipeline; image frames in the left correspond to tracker output and image frames to the right correspond to tracked images with predicted speed and traffic density.

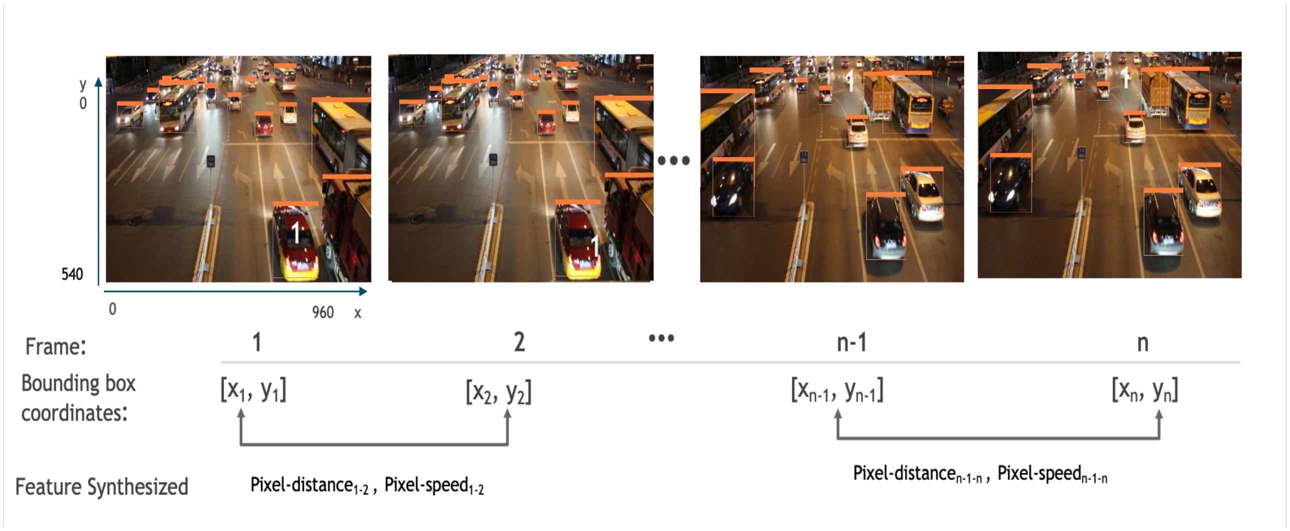

y
0
540
0
960
x
1
1
1
1
Frame:
1
2
•••
n-1
n
Bounding box coordinates:
$[x_1, y_1]$
$[x_2, y_2]$
$[x_{n-1}, y_{n-1}]$
$[x_n, y_n]$
Feature Synthesized
Pixel-distance$_{1-2}$, Pixel-speed$_{1-2}$
Pixel-distance$_{n-1-n}$, Pixel-speed$_{n-1-n}$

Figure 2: The Tracklet extraction & Feature Synthesizer module which takes in the tracker output and generates features such as pixel distance and pixel speed which are input to the speed estimation and traffic density estimation modules.

3.1 Tracklet Extraction & Feature Synthesizer Module

This module takes in the tracker output and synthesizes features which go into the speed estimation module and the traffic density module as shown in Figure 2. The output of the tracker for each tracklet consists of:

i. The four bounding box coordinates (x_{min}, y_{min}, x_{max}, x_{max}).

ii. The unique ID for each tracked vehicle (tracklet ID).

iii. Tracklet class.

iv. Confidence score.

From the bounding box coordinates the area and centroids of each bounding box is extracted. For every unique tracklet, a full motion trajectory is captured.

The pixel distance traveled between two consecutive frames ($n - 1$ & n) by an object is computed using the 2-D Euclidean distance of their centroids, given by:

$$pixel\ distance = \sqrt{(x_{centroid_n} - x_{centroid_{n-1}})^2 + (y_{centroid_n} - y_{centroid_{n-1}})^2} \tag{1}$$

The pixel speed is also obtained with the expression below:

$$pixel\ speed = \frac{pixel\ distance * \ frame - rate}{number\ of\ frames} \tag{2}$$

$$x_{centroid}, y_{centroid} = \frac{x_{max} + x_{min}}{2}, \frac{y_{max} + y_{min}}{2} \tag{3}$$

Where, number of frames = 2. Note that different number of frames values can be considered. For example, for a video with a framerate of 25, the pixel distance for the 25 frames making 1 sec would be accumulated and aggregated to give a broadcasted single value for every 25 frames. However, to achieve instant level predictions i.e., predictions for every frame, uniquely synthesized features were considered. Obtaining the pixel distance and speed features for n instances of a vehicle would give n-1 features, to account for this, forward and backward fills were applied i) backward to estimate the first instance given the last n-1 instances and ii) forward to estimate the nth instance given the first n-1 instances, backward fill produced better results.

As an augmentation step to address potential imperfections of the tracker, such as misses caused by occlusion, for every tracklet's trajectory path, there is a check to ensure there are no missing frames. In the event of missing frames, linear interpolation is applied to forecast the missing frame's centroid values given by the following equations.

$$x_{centroid\_missing} = x_{centroid_1} + \frac{(x_{centroid_2} - x_{centroid_1})(frame\_id\_missing - frame\_id_1)}{\left(frame_{id_2} - frame_{id_1}\right)} \tag{4}$$

$$y_{centroid\_missing} = y_{centroid_1} + \frac{(y_{centroid_2} - y_{centroid_1})(frame\_id\_missing - frame\_id_1)}{(frame\_id_2 - frame\_id_1)} \tag{5}$$

3.2 Speed Estimation Module

The output of the tracklet extraction and feature synthesizer module feeds into the speed estimation module. This model consists of a regression machine learning model that takes in the following features:

i. x and y centroid coordinates.

ii. Area of bounding box.

iii. Tracklet ID.

iv. Tracklet class.

v. Pixel distance (pixels).

vi. Pixel speed (pixels/sec).

The model is tasked with predicting speed in m/s given these features. The detailed training experimental procedure of the speed estimation model is given in the experiments and results section.

3.3 Traffic Density Module

Similar to the speed estimation module, the output of the tracklet extraction and feature synthesizer module feeds into the traffic density module. In this case, the following features are extracted:

i. x and y centroid coordinates.

ii. Count of tracklets/vehicles in a frame.

Since traffic density is a measure of the number of vehicles per unit road length, a one-time calibration step is performed where the distance between the points where a vehicle enters and exits the camera's view i.e., the first and last frame the tracklet appears is computed. This distance is computed for every tracklet and the largest distance which corresponds to the longest straight path is extracted and used to approximate the road length of the video. This is mathematically given by:

$$Road\ length\ \approx max[\sqrt{\left(x_{centroid_n} - x_{centroid_1}\right)^2 + \left(y_{centroid_n} - y_{centroid_1}\right)^2}\ ]_{1:T} \tag{6}$$

Where, $(x_{centroid_1}, x_{centroid_n})$ and $(y_{centroid_1}, y_{centroid_n})$ correspond to the x and y-centroid coordinates of the first and last frames of T tracklets of which the maximum distance is chosen to be the road length. The traffic density is then given by:

$$Traffic\ Density = \frac{vehicle\ count\ per\ frame}{road\ length} \tag{7}$$

The traffic density feature is fed into an unsupervised clustering algorithm which is tasked with classifying the traffic of each video frame into light, medium or heavy traffic density. Procedural training details are given in the results and experiments section.

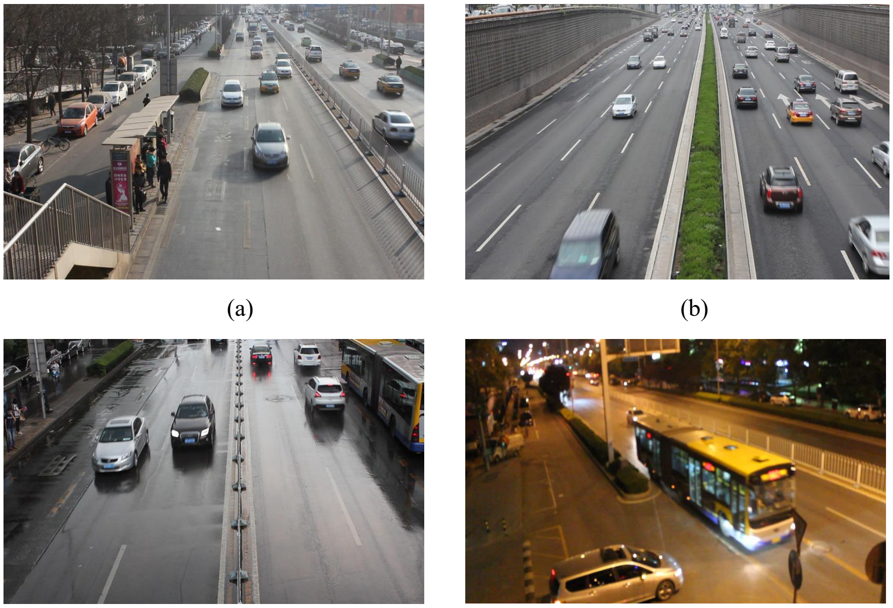
4. EXPERIMENTS AND RESULTS

4.1 UA-DETRAC Dataset

The university of Albany Detection and TRACking (UA-DETRAC)[7]–[9] benchmark dataset is a publicly available large scale dataset for performance evaluation of detection and MOT models. It consists of 100 challenging videos with more than 140,000 image frames with a 960 x 540 resolution and a frame rate of 25 fps captured from diverse real-world traffic scenes such as rainy, cloudy, sunny, night scenes. It comprises of 8,250 richly annotated vehicles and 1.21 million vehicle bounding box labels. The data is collected at 24 different locations at Beijing and Tianjin, China. Figure 3 shows samples from the UA-DETRAC training set. Table 1 shows the number of videos per scene in the 60 videos of training data and 40 videos of test data along with the total number of frames.

Table 1: UA-DETRAC data size breakdown

Dataset	**Number of Videos by Scenes**				**Number of Frames**
	Sunny	**Cloudy**	**Rainy**	**Night**	
Train	15	19	10	16	83,790
Test	8	11	9	12	56,340



(a)

(b)

(c)

(d)

Figure 3: Image samples from UA-DETRAC in a) sunny scene, b) cloudy scene c) rainy scene and d) night scene

4.2 Speed Estimation Model

To build the speed estimation model, annotations of the UA-DETRAC dataset of all 60 videos were extracted. Key ground truth annotation features are the bounding box coordinates, vehicle class and ID, trajectory length, truncation ratio and "speed". Similar to the process outlined in the tracklet extraction & feature synthesizer module, pixel distance and speed are generated for each annotation instance. Exploratory analysis reveals that the synthetically generated features, particularly pixel and distance speed have strongly positive correlations with the ground truth speed as shown in the correlation heat map in Figure 4. The provision of the groundtruth speed values permits the speed estimation problem to be addressed with a data-driven supervised learning approach with speed as the target label. The model's objective is to minimize the error between the true speed values and the model's predictions. Training is performed on a vast number of algorithms suitable for regression. For the training 10-fold and 5-fold cross-validations were carried out yielding near identical results and the performance metrics considered are the Mean Absolute Error (MAE) which gives the absolute difference between model prediction and ground truth value, Root Mean Squared Error (RMSE) which is the square root of the mean of the square of all of the error [32], it provides insight on how the error is distributed[33] and the coefficient of determination (R^2) which captures the variations in the dependent variable (speed) that is predictable from the independent variables (features). These metrics are shown in equations 8-11.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{I=1}^{n} e_i^2}{\sum_{I=1}^{n} (y_i - \bar{y})^2} \tag{10}$$

$$e_i = y_i - \widehat{y_\iota} \quad (11)$$

Where, $(y_i , \widehat{y_\iota})$ is the speed prediction, groundtruth pair and $\bar{y}$ is the mean of ground truth speed. The performances of the speed estimation models explored are given in Table 2, with the linear regression model selected.

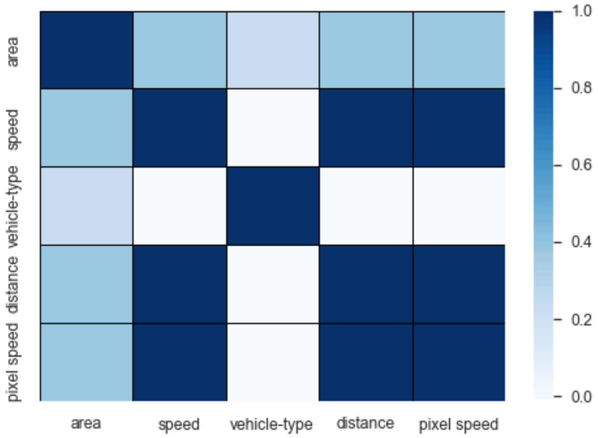


Figure 4: Heatmap showing correlation between synthesized features and ground truth speed, the strongest correlations are between pixel speed, pixel distance and ground truth speed.

Table 2: results of speed estimation models on training (using 10-fold cross-validation) and testing.

Mode	Model	MAE ⬇	RMSE⬇	R^2 ⬆
Training (10-fold cross validation)	Linear Regression [34], [35]	0.0018	0.0024	0.9999
	Lasso Regression [34]	0.1094	0.1695	0.9995
	Huber Regressor [34]	0.0609	0.0255	0.9996
	MLPRegressor [33]	1.0972	1.8792	0.7484
	K Neighbors Regressor [34]	1.2941	2.7783	0.8558
Test	Linear Regression [34]	0.0016	0.0023	0.9999

4.3 Clustering Model

To build the clustering model, the pre-processing procedure carried out on the traffic density module was performed on the annotations of the training set to generate the traffic density feature. KMeans unsupervised learning was carried out on the training data to classify traffic density into heavy, medium, or light. The clustering model was evaluated using the J-squared error function [36] given by:

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} ||x_n - \mu_k \,||^2 \quad (12)$$

This represents the sum of the squares of the distances of each traffic density data point to its assigned cluster. Where, x_n is a traffic density datapoint, μ_k is the centroid of cluster k and r_{ik} is a one-hot vector encoding of the K clusters. Intuitively, since the clustering model is tasked with grouping traffic density into heavy, medium, and light a K = 3 would be natural. Experiments were carried out on the following cluster sizes K = 3, 4, 5, 6 with K = 6 giving the least amount of intersection between clusters. Clusters 4, 1, and 2 were grouped as light, clusters 3 and 0 were grouped as medium and cluster 5 was grouped as heavy based on traffic density values as shown in Figure 5.

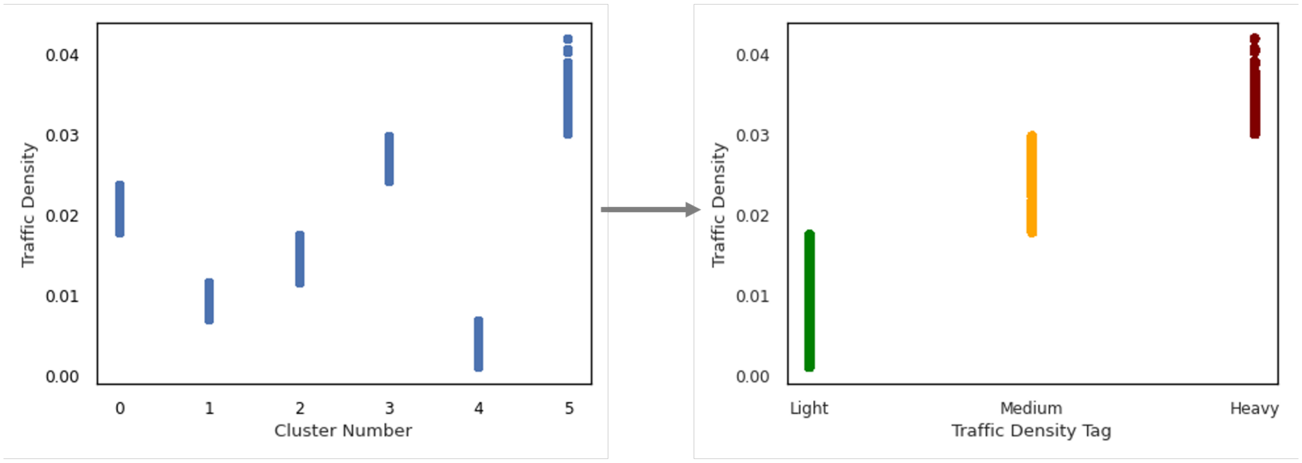


Figure 5: KMeans clustering for traffic density estimation with K = 6; Clustering model output mapping on the left side, 6 clusters, right side corresponds to the transformation from cluster integers to traffic density tags. Cluster 5 corresponds to heavy traffic density, clusters 0 and 3 correspond to medium traffic density and cluster 4, 1 and 2 correspond to light traffic density (where traffic density is given in vehicle count per pixel road length).

4.4 Implementation details

Training, validation and testing of the speed and traffic density estimation models were carried out on the sklearn [35] and pycaret libraries [34]. The results for the trained multi-object tracking network, SiamMOT with the DLA-34 used as the backbone for Faster-RCNN reported by [27] is given in table 4. The tracking performance metrics used are the Multiple Object Tracking Accuracy (MOTA) and IDF1 [37].

Table 3: SiamMOT performance as reported [27]

Dataset	**MOTA ↑**	**IDF1 ↑**
MOT17 [37]	65.9	63.3
HiEve [38]	51.5	47.9

4.5 Results

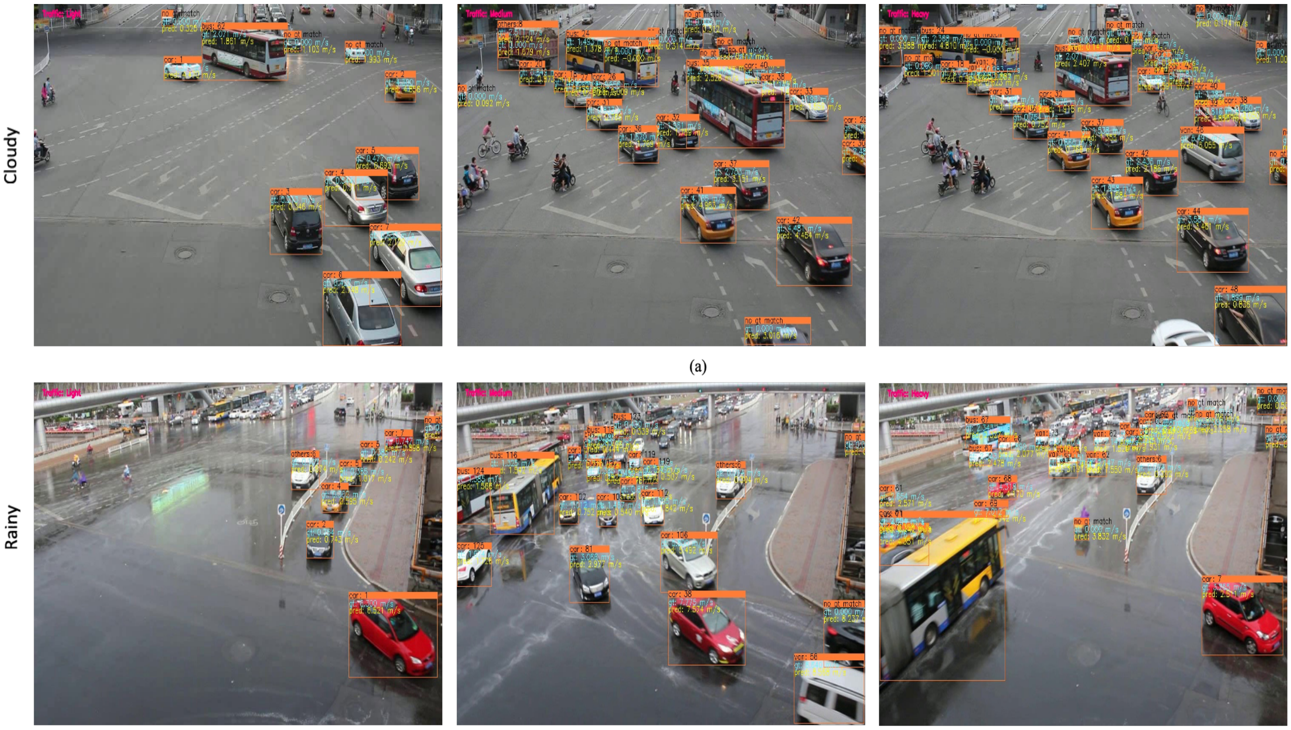
Table 5 quantifies the performance of the proposed methodology on the 40 test videos of the UA-DETRAC dataset in comparison with other state-of-the-art technique(s). Figure 6 shows images selected from the output of this methodology across the four scenes considered i.e., sunny, cloudy, rainy. Our approach outperforms [29] across all scenes.

Table 4: results of our approach in comparison with [29]

Method	**MAE ↓**	**RMSE ↓**	**R^2 ↑**
Ours	0.824	1.621	0.599
ECCNet[29]	3.100	3.930	-

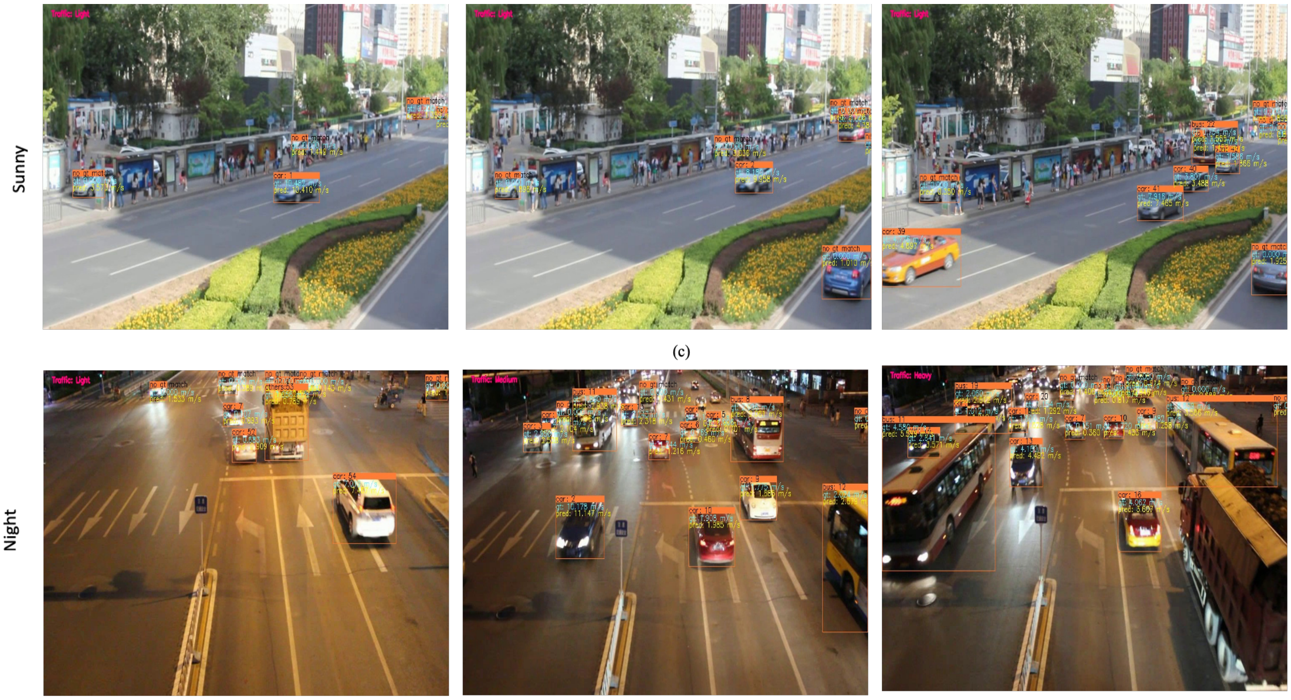
Table 5: results of our approach across different scenes in comparison with [29]

Method	**Scene**	**MAE ↓**	**RMSE↓**	**R^2 ↑**
Ours	Sunny	1.016	2.341	0.528
	Cloudy	0.571	1.012	0.700
	Rainy	0.771	1.382	0.489
	Night	0.969	1.880	0.636
ECCNet [29]	Sunny	2.850	4.000	-
	Cloudy	2.770	3.510	-
	Rainy	3.180	3.900	-
	Night	3.590	4.480	-



(a)

(b)



(c)

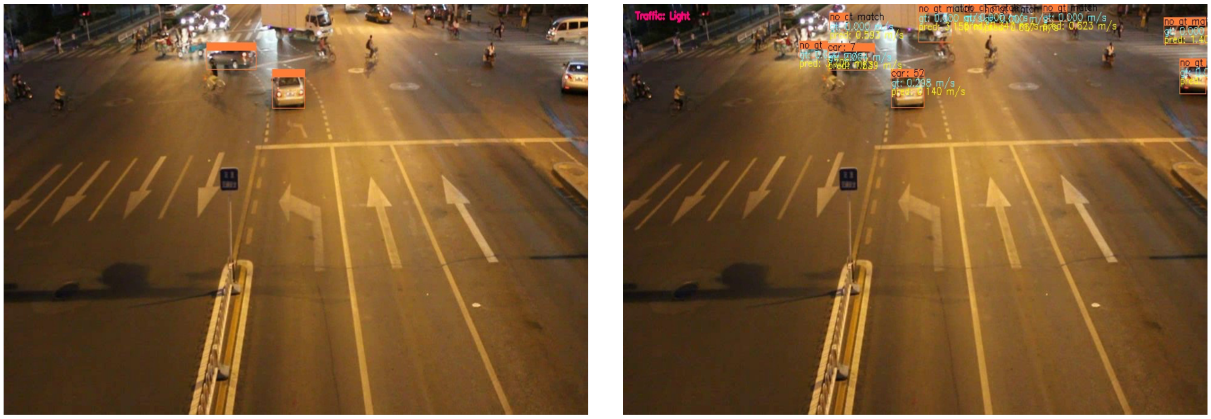
(d)

Figure 6: Videos/Images showing speed and traffic density estimation of our method in a) cloudy scene b) rainy scene c) sunny scene d) night scene with light. Full videos here: http://dx.doi.org/10.1117/12.2663643.1, http://dx.doi.org/10.1117/12.2663643.2, http://dx.doi.org/10.1117/12.2663643.3, http://dx.doi.org/10.1117/12.2663643.4

4.6 Discussion

The results reported in Table 2 for the speed estimation model do not reflect the performance of this methodology in real world scenarios. It depicts the potential performance of our framework given an ideal tracker since it is trained on groundtruth annotations where all the tracked instances are true values not values predicted by a tracker. This methodology is sensitive to tracker performance. For illustration, consider an object with trajectory path of 3 consecutive frames. In a scenario where the tracker misses the intermediate frame, the pixel distance computed would be larger leading to larger predictions. To alleviate this dependence on tracker output, in the event of these misses this framework interpolates the missing frame's coordinates as an augmentation strategy.

The tracker was able to detect and associate vehicles outside the scope of the ground truth annotations particularly in the sunny scene, some of these outputs were suppressed to match the ground truth as closely as possible. However, this slightly reduced performance as shown in Table 5.



(a) (b)

Figure 7 a) shows the bounding box on the ground truth annotations b) shows the predictions of the tracker and our model output. Certain regions not captured by the ground truth a) but captured by tracker b) some of these features are suppressed and the tag “no gt match” is used to represent some of these tracklets as seen in b).

5. CONCLUSION AND FUTURE WORK

This work introduces a simple yet effective, data-driven approach to speed and traffic density estimation that is adaptable to any multi-object tracker. This approach does not require meter-to-pixel mapping, prior road information, or other forms of rigorous calibration. It outperforms the existing approach across all scenes considered by a 73.4% reduction in MAE. Furthermore, to reduce the dependence and sensitivity of the proposed method on tracker performance, an augmentation strategy to forecast missing tracklet frames is introduced. As part of our future work, this model will be evaluated on more datasets, and image enhancement techniques will be explored to produce even better performance in poor weather conditions like haze, heavy rain, and fog to mention a few. Finally, a scale feature that could be used to generate a zoom factor will be incorporated into the model in training to add a penalization to tracked objects thereby regularizing the pixel features accordingly making it invariant to the effects of panning and zooming in real-world scenarios.

ACKNOWLEDGMENTS

This work is supported by the US Department of Transportation, Federal Highway Administration (FHWA), grant contract: **693JJ320C000023**. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Department of Transportation, Federal Highway Administration (FHWA).

REFERENCES

[1] National Highway Traffic Safety Administration (NHTSA),” *Speeding*, Jun. 23, 2022. https://www.nhtsa.gov/risky-driving/speeding.

[2] L. Kezebou, V. Oludare, K. Panetta and S. Agaian, ‘Few-Shots Learning for Fine-Grained Vehicle Model Recognition,’ in 2021 IEEE International Symposium on Technologies for Homeland Security (HST), 2021, pp. 1-9, doi: 10.1109/HST53381.2021.9619823.

[3] L. Kezebou, V. Oludare, K. Panetta, J. Intriligator, and S. Agaian, ‘Highway accident detection and classification from live traffic surveillance cameras: a comprehensive dataset and video action recognition benchmarking’ May, 2022. In Multimodal Image Exploitation and Learning 2022 (Vol. 12100, pp. 240-250). SPIE.

[4] L. Kezebou, V. Oludare, K. Panetta, and S. Agaian ‘A deep neural network approach for detecting wrong-way driving incidents on highway roads.’ April, 2021. In Multimodal Image Exploitation and Learning 2021 (Vol. 11734, pp. 183-197). SPIE.

[5] L. Kezebou, V. Oludare, K. Panetta and S. S. Agaian, ‘Underwater Object Tracking Benchmark and Dataset,’ in 2019 IEEE International Symposium on Technologies for Homeland Security (HST), 2019, pp. 1-6, doi: 10.1109/HST47167.2019.9032954.

[6] S. Rajeev, A. Samani, K. Panetta and S. Agaian, ‘3D Navigational Insight using AR Technology,’ in 2019 IEEE International Symposium on Technologies for Homeland Security (HST), 2019, pp. 1-4, doi: 10.1109/HST47167.2019.9033006.

[7] L. Wen et al., ‘UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking,’ ArXiv151104136 Cs, Jan. 2020, Accessed: May 25, 2021. [Online]. Available: http://arxiv.org/abs/1511.04136.

[8] S. Lyu et al., ‘UA-DETRAC 2017: Report of AVSS2017 IWT4S Challenge on Advanced Traffic Monitoring,’ in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Aug. 2017, pp. 1–7. doi: 10.1109/AVSS.2017.8078560.

[9] S. Lyu et al., ‘UA-DETRAC 2018: Report of AVSS2018 IWT4S Challenge on Advanced Traffic Monitoring,’ in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Nov. 2018, pp. 1–6. doi: 10.1109/AVSS.2018.8639089.

[10] D. F. Llorca et al., ‘Two-camera based accurate vehicle speed measurement using average speed at a fixed point,’ in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Nov. 2016, pp. 2533–2538. doi: 10.1109/ITSC.2016.7795963.

[11] V.-H. Do, L.-H. Nghiem, N. Pham Thi, and N. Pham Ngoc, ‘A simple camera calibration method for vehicle velocity estimation,’ in 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Jun. 2015, pp. 1–5. doi: 10.1109/ECTICon.2015.7207027.

[12] K. S. and V. S., ‘Contour-based object tracking in video scenes through optical flow and gabor features,’ Optik, vol. 157, pp. 787–797, Mar. 2018, doi: 10.1016/j.ijleo.2017.11.181.

[13] S. S. Sengar and S. Mukhopadhyay, ‘Moving object area detection using normalized self-adaptive optical flow,’ Optik, vol. 127, no. 16, pp. 6258–6267, Aug. 2016, doi: 10.1016/j.ijleo.2016.03.061.

[14] A. El Bouziady, R. O. H. Thami, M. Ghogho, O. Bourja, and S. El Fkihi, ‘Vehicle speed estimation using extracted SURF features from stereo images,’ in 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Apr. 2018, pp. 1–6. doi: 10.1109/ISACV.2018.8354040.

[15] O. Ibrahim, H. ElGendy, and A. Elshafee, ‘Speed Detection Camera System using Image Processing Techniques on Video Streams,’ Int. J. Comput. Electr. Eng., vol. 3, p. 771, Dec. 2011, doi: 10.7763/IJCEE.

[16] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018. Recent advances in convolutional neural networks. Pattern recognition, 77, pp.354-377.

[17] K. Behrendt, ‘Boxy Vehicle Detection in Large Images,’ in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Oct. 2019, pp. 840–846. doi: 10.1109/ICCVW.2019.00112.

[18] S. M. Azimi, R. Bahmanyar, C. Henry, and F. Kurz, ‘EAGLE: Large-scale Vehicle Detection Dataset in Real-World Scenarios using Aerial Imagery,’ ArXiv200706124 Cs, Nov. 2020, Accessed: May 26, 2021. [Online]. Available: http://arxiv.org/abs/2007.06124.

[19] H.-Y. Lin, K.-C. Tu, and C.-Y. Li, ‘VAID: An Aerial Image Dataset for Vehicle Detection and Classification,’ IEEE Access, vol. 8, pp. 212209–212219, 2020, doi: 10.1109/ACCESS.2020.3040290.

[20] S. Ren, K. He, R. Girshick, and J. Sun, ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,’ ArXiv150601497 Cs, Jan. 2016, Accessed: May 26, 2021. [Online]. Available: http://arxiv.org/abs/1506.01497.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘You Only Look Once: Unified, Real-Time Object Detection,’ in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[22] J. Wang et al., ‘Deep High-Resolution Representation Learning for Visual Recognition,’ ArXiv190807919 Cs, Mar. 2020, Accessed: May 26, 2021. [Online]. Available: http://arxiv.org/abs/1908.07919.

[23] H. Ait Abdelali, F. Essannouni, L. Essannouni, and D. Aboutajdine, ‘An Adaptive Object Tracking Using Kalman Filter and Probability Product Kernel,’ Model. Simul. Eng., vol. 2016, p. e2592368, Mar. 2016, doi: 10.1155/2016/2592368.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, ‘High-Speed Tracking with Kernelized Correlation Filters,’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Mar. 2015, doi: 10.1109/TPAMI.2014.2345390.

[25] H. K. Galoogahi, A. Fagg, and S. Lucey, ‘Learning Background-Aware Correlation Filters for Visual Tracking,’ in 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017, pp. 1144–1152. doi: 10.1109/ICCV.2017.129.

[26] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B., 2016, September. Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). IEEE.

[27] Shuai, B., Berneshawi, A., Li, X., Modolo, D. and Tighe, J., 2021. Siammot: Siamese multi-object tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12372-12382).

[28] S. Hua, M. Kapoor, and D. C. Anastasiu, ‘Vehicle Tracking and Speed Estimation from Traffic Videos,’ in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2018, pp. 153–1537. doi: 10.1109/CVPRW.2018.00028.

[29] Yu, C., Yang, J., Jiang, S., Zhang, Y., Li, H. and Du, L., 2022. ECCNet: Efficient chained centre network for real-time multi-category vehicle tracking and vehicle speed estimation. IET Intelligent Transport Systems, 16(11), pp.1489-1503.

[30] Yu, F., Wang, D., Shelhamer, E. and Darrell, T., 2018. Deep layer aggregation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2403-2412).

[31] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[32] D. Christie and S. P. Neil, “8.09 - Measuring and Observing the Ocean Renewable Energy Resource,” in *Comprehensive Renewable Energy (Second Edition)*, Second Edition.Elsevier, 2022, pp. 149–175. Accessed: Apr. 21, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128197271000832

[33] Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE). Geoscientific model development discussions, 7(1), pp.1525-1534.

[34] “PyCaret: An open source, low-code machine learning library in Python.” Apr. 2020. [Online]. Available: https://www.pycaret.org

[35] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.

[36] *Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer, Berlin.*

[37] Milan, A., Leal-Taixé, L., Reid, I., Roth, S. and Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.

[38] Lin, W., Liu, H., Liu, S., Li, Y., Qian, R., Wang, T., Xu, N., Xiong, H., Qi, G.J. and Sebe, N., 2020. Human in events: A large-scale benchmark for human-centric video analysis in complex events. arXiv preprint arXiv:2005.04490.