Journal of Advances in Management, Engineering and Science (JAMES), 2026; 1(2): 41-52

Research Article

Real-Time Vehicle Detection and Tracking Based on CEEMDAN and LSTM Approach

D.R Prince Williams[1], Department of Computing and Information Sciences, University of Technology and Applied Sciences-Suhar, Sultanate of Oman, princeshree1@gmail.com

T. Prem Jacob[2], Department of CSE, Sathyabama Institute of Science and Technology, Chennai, India, premjac@yahoo.com

Received: 9th November 2025; Accepted: 2nd March 2026; Published: 30th April 2026
https://doi.org/10.65470/james.v1i02.22

Abstract – *This novel real-time vision system allows for the real-time analysis of color movies recorded by a forward-facing video camera in a moving vehicle on a highway. The system utilizes edge, color, and motion data to detect and track lane lines, road limits, and other vehicles on the road. Car recognition is achieved by matching input data online templates, identifying highway scene characteristics, and evaluating their associations. Vehicle detection can also benefit from temporal differencing and the tracking of motion metrics that are characteristic of cars. Preprocessing, segmentation, feature extraction, and model training are all utilized in the suggested approach. Preprocessing often distorts images and videos captured by cameras in busy public spaces. An algorithm employs the depth image to divide up areas that could be driven in. They used a feature extraction method based on the Haar Wavelet Transform. Once the features have been extracted, CEEMDAN-LSTM is used to train the models. The proposed method outperforms two of the most common alternatives, CEEMDAN and LSTM.*

Keywords— *Vehicle Detection, Ensemble Empirical Modal Decomposition (EEMD)·Long Short-Term Memory (LSTM).*

INTRODUCTION

One of the most critical new research challenges in autonomous vehicles, computer vision, and other kinds of intelligent transportation is how to detect and follow several vehicles. There are several motivating motivations for vehicle detection, including the need to reduce fatality rates and the associated costs. A major objective of these high-tech identifying and tracking systems is to reduce injuries sustained by passengers in motor vehicle collisions. There are currently a variety of challenges that make it difficult to detect and track many cars in real-time motion on the road. These include things like camera oscillations, changing lighting, camouflage, background noise, occlusion, collision, and motion speed. Evidence from several studies shows that the detection approaches were taken from motion-based detection, sensor-based detection, feature-based object detection, and so on. Various forms of including supervised and unsupervised machine learning, deep learning, convolutional neural networks, reinforcement learning, and others, have been proposed for the task of object detection and sliding window method, tracking neural network, fast R-CNN, R-CNN, and a single pass over the data are all examples of what a multi-scale deep convolutional

neural network can perform. Congestion on roads is a big problem in both developed and developing countries' most populous cities. Congestion in public transit systems is bound to worsen as cities grow to accommodate new residents. More in-depth information about management, planning, and traffic law is urgently needed. Cities can benefit from including the development of more precise traffic control programmes, traffic monitoring in several ways, the investigation of the temporal impacts of vehicle density on urban heat islands, the execution of traffic simulations, and the elucidation of the connection between air quality and combustion-related particulates. Traditional methods of traffic monitoring have included closed-circuit television (CCTV) systems at close range and airborne or satellite imaging at distant range. Closed-circuit television and aerial/space borne videos both provide unique insights into traffic patterns. Due to the vast differences between aerial and space-based imagery, the performance of CCTV systems is outside the scope of this article. The study of Earth has gained popularity in recent years. Having access to high-resolution images is essential. Vehicle detection is a crucial part of networks working on challenges like continuous monitoring, and the introduction of an Intelligent Transportation System, and autonomous vehicle research. Vehicle detection is at the center of the rapidly developing technological field of autonomous driving. The ever-increasing incidences of traffic accidents, robberies, and violations involving motor vehicles make it imperative that cities and towns install surveillance cameras to monitor the roadways. In order to identify vehicles in real time, the detector utilized in traffic surveillance systems must be lightning fast. Both traffic control and video monitoring have seen advancements. In the field of vehicle detection, spotting anomalies in traffic monitoring is a particularly fruitful area of research, including the detection of congestion, parking violations, and risky driving, among others. As deep learning neural networks for detection advance rapidly, machine learning methods for object detection networks are falling out of favor. Driving safely, operating in autonomous mode, pursuing other vehicles, and avoiding accidents all rely on drivers' awareness of their surroundings and the whereabouts of other vehicles. We came up with a method that uses the car's camera to track oncoming traffic and recognize cars headed in the same direction as the car. The main difficulty is in correctly detecting cars in environments with changing lighting and backgrounds. Few approaches have been demonstrated to function consistently in real time for in-car video, which must process the input while the vehicle is in motion, even though there are several documented methods for detecting and following moving objects in general. In this work, we outline an effort to create and roll out algorithms and systems with a strong focus on real-time data and a deep understanding of the road and traffic environments. It records upcoming traffic and road scenarios, making it useful for things like safe driving, traffic recording, and vehicle pursuit. Our team's mission is to always keep an eye on security footage in search of vehicles that are either approaching or being chased. The primary goals of ADAS, which are standard in most new automobiles, are to improve road safety, decrease accident rates, boost fuel economy, enhance passenger comfort, and enhance the driving experience overall. Several advanced driver assistance systems are stepping stones on the road to trustworthy, fully driverless automobiles of the future.

RELATED WORKS

There is a direct correlation between the precision of vehicle detection and the success of traffic monitoring and intelligent surveillance [1]. Because of recent developments in sensors, GPUs, and deep learning, study of autonomous or self-driving applications has gained traction [2]. Accurate, real-time detection of traffic objects (such as traffic lights, other vehicles etc.) is crucial for autonomous vehicles to make the proper control decisions and provide the required level of safety [3]. Autonomous vehicles use a variety of sensors, including cameras and light detection and ranging, to detect and avoid such hazards. Extreme weather conditions, such as sleeting rain, thick fog, dusty blasts, low lighting, and snowstorms have a substantial impact on camera image quality. Poor visibility makes it harder to identify vehicles, which increases the likelihood of accidents. It is possible to develop efficient picture enhancement technologies that boost aesthetic appeal or discriminative features, leading to better clarity. To improve the effectiveness of vehicle detection and tracking systems [4], it is helpful to provide these systems with high-quality images. Several techniques for vehicle detection have been developed in the realm of computer vision. For instance, deep learning has been implemented successfully in self-driving cars and other applications that use image sensors for traffic object detection [5]. After confirming the existence of vehicles, details about their positions and speeds can be gleaned. However, installing these sensors might be a pain. Costs are considerable both initially and over time [6]. Autonomous driving relies heavily on sensors like radar, light detection, and camera and ranging (LiDAR) devices. From a secure distance, autonomous vehicles' sensors are supposed to provide precise data. [7] When light is scarce, traditional cameras struggle to perform, but LiDAR sensors thrive. Elaborate method for tracking the location, velocity, and acceleration of a target dimensions and contours of automobiles by means of a hierarchical filtering system (one Rao-Blackwellized particle filter, one Bayes filter, and [8] and the Kalman filter, and one Particle filtering using a scaling series. Data on traffic flows in and out of a certain area includes average speeds, breakdowns by vehicle type or total volume, entry and departure locations, and travel times between thirty key nodes. [9] Tracking and detecting moving vehicles allows for the collection of this information. That's why it's crucial to have robust and efficient vehicle detection and tracking technologies in place for transportation networks. The feature vector can be created using either static or dynamic features how many pixels make up the whole image [10]. The detection algorithms can recognize vehicles by their distinctive shapes and hues follow a simple three-step process. To kick things off, they put on a show background modelling or deletion-based background subtraction historical context. After the initial backdrop removal stage, detection of any residual blobs and their locations. Then, we use to the pull out the important (customer-defined) features found spots and where they were located [11]. In the final phase, Classification receives the found blobs and extracted features. Methods for categorizing vehicles. These algorithms work well in videos where the music never stops playing online and appropriate weather conditions. When compared to other entities, however, they do poorly. Rapid changes in the background, varying weather and color, and form-related problems [12]. Computer programmes that analyse motion data, like dynamic Occupancy,

Optical Flow, and Background Modelling grid and tracking stipels were developed to issues rooted in observable physical traits and social stratification that algorithms need to deal with from the past[13].Vision-based vehicle object recognition can be broken down into two primary groups at the moment: basic machine vision approaches and cutting-edge deep learning methods. The vehicle's motion is typically utilized to separate it from the still background in traditional machine vision systems. This method can be broken down into three distinct groups: continuous video frame difference, optical flow, and background subtraction [14]. Using the video frame difference approach, we can calculate the discrepancy by comparing the pixel values of two or three consecutive video frames. Furthermore, the threshold separates the non-moving backdrop from the moving foreground area [15].Traditional machine vision methods for vehicle detection may be faster, but they do not guarantee a high quality result positive result when there is periodic motion in the background, when there is variation in the brightness of the image, and when there are slow-moving cars or intricate scenery. Excellent CNN is becoming more and better at object recognition, but CNN pays special attention to changes in scale when recognizing objects [16]. Radar has the potential to be an effective tool for autonomous perception. To enhance the depth estimation, the sparse and noisy radar dots are superimposed onto camera photos using the method outlined. Using radar's Doppler frequency changes, as explained, is one approach to detect pedestrians that lidar cannot. The DEF team developed an early fusion detector that included lidar, cameras, and radar. Video-based supervision frameworks are useful in the field of traffic monitoring. In order to extract useful and accurate traffic data for traffic picture investigation and traffic stream control, such as vehicle direction, vehicle count,[17][28] vehicle stream, vehicle tracking, traffic density, vehicle arrangement, traffic path changes, license plate recognition, vehicle speed, and so on, experts have been studying vision-based ITS, transportation planning, and traffic designing applications for some time now. [18] All visual processing algorithms are mapped for real-time execution on reliable field programmable gate array devices using appropriate error correction codes. The rising importance of safety has prompted researchers to investigate cutting-edge surveillance techniques. Intelligent surveillance systems can be used for a variety of purposes, including as detecting and monitoring moving objects [19], segmenting videos into their constituent parts, identifying and classifying objects, and even seeing events and representing and comprehending user activity. [20] The ability to identify and follow moving vehicles is essential for traffic management. It is used to determine the gravitational location of moving objects in video in order to prevent potential collisions in crowded areas. Automatic parking, autonomous vehicle frameworks, estimating traffic parameters, and predicting trip times all rely heavily on this technique [21][29]. It is critical to traffic monitoring that vehicles be recognized accurately and reliably. ITSs and vehicle tracking raise several unanswered questions. Due to the great variability in vehicle looks and placements, it is difficult to construct a consistent detection model. Two distinct kinds of point cloud representations underpin lidar-based object detection. Lidar data is usually given as point clouds, which makes it straightforward to work with using architectures that are designed for unordered point sets [22][26]. This architecture paves the way for end-to-end learning

with raw point clouds. Background identification is accomplished by combining point-wise features extracted by Point-Net and later by PointRCNN. Then, it makes recommendations and refines the final detection results. By first segmenting points into pillars and then calculating features for each pillar independently, PointPillars [23][27] uses PointNet to generate a pseudo picture. Afterwards, the image is understood by an SSD detection head and a CNN backbone. Regrettably, point-wise features cannot be learned about regions covered by severe weather since no points exist there. Another option is to use vowelized variants of traditional image detection methods to examine lidar point clouds. In order to generate an occupancy map, PIXOR segments points at different heights. The voxel model allows MVDNet to utilize commonly used image data, like that from cameras and lidar, which can be seamlessly incorporated. Visual sensors' contrast [24]and viewable range are reduced in haze and fog, which in turn reduces the quality of data from lidar and cameras. On one side, we have cutting-edge picture dehiscing technologies that may be valuable for educational purposes. Whether these methods rely on learnt [25] or hand-crafted priors, they construct a trainable model from start to finish or estimate a transmission map between good and bad images. Though lidar point cloud denoising is an important subject, it has been under-discussed. Denoising methods that were developed for thick 3D point clouds cannot be directly applied to lidar point clouds due to their scarcity in order to remove fog spots. Denoising is accomplished by DROR through the utilization of points' dynamic spatial proximity. Inadvertently erasing reflections of specific objects is possible due to its lack of semantic information. However, present denoising methods are unable to compensate for lidar's visibility loss caused by fog in the absence of additional data. In contrast, MVDNet combats hazy conditions by combining weather-sensitive lidar point clouds with high-resolution radar. Due to the redundant data they give, multimodal sensors can withstand external and internal noise-induced sensor distortions.

METHODOLOGY

Autonomous vehicles rely heavily on infrastructure like precise localization, environmental monitoring, and preplanned routes. There are substantial obstacles to the environment perception of autonomous automobiles due to the complexity of road situations, especially those with mixed traffic. Detecting vehicles is an important aspect of environmental perception that ensures a UGV can be operated safely.

A. Preprocessing

The cameras do not record balanced images or videos in high-traffic areas. Due to the presence of potentially offensive elements in the media. Moreover, it's possible that environmental and meteorological impacts will lower the quality of the photographs. These are some of the possible causes of problems with color perception. We overcame these confounding factors by removing all distractions from the photos, leaving only the target and its background [22]. In addition, our network's preprocessing includes procedures, and the image is more harmonious now that the distractions have been taken away. Another method used to enhance the image's color quality is a color contrast technique that makes use of biased correction and statistical estimation. The final product is a clearer and more precise depiction of the car.

B. Segmentation

The depth image is used by the image segmentation algorithm to separate potential parking spots. A depth image can be used to calculate the relative distances of objects in a scene. It is a common assumption of segmentation algorithms that distance values within an object should be uniform [23]. According to this theory, it's expected the first derivative of depth values to be small for a given object but to get greater near its boundaries. The following expressions are used to compute the q and w components of the first derivative, respectively.

$$Der\_Q(q,w) = \frac{Depth(q+n,w) - Depth(q-n,w)}{2n} \quad (1)$$

$$Der\_W(q,w) = \frac{Depth(q,w+n) - Depth(q,w-n)}{2n} \quad (2)$$

where the depth image value for the pixel at coordinates (q, w) is Depth (q, w), and n∈{1,2,3,⋯ ,h}.

C. Feature Extraction

Wavelets approximate multi-resolution functions that can be used to dissect a signal or image hierarchically. Successful applications include those in the fields of object identification, facial recognition, and image retrieval, among many others. Only two waveforms (mother wavelets) are required to perform a decomposition of a signal into wavelets [24]. By translating and scaling the two forms, they can make waves of varying amplitudes and wavelengths, a process known as creating a wavelet (wavelet basis). It defines the bare minimum requirement for conducting multi-resolution analysis as follows:

$$U_{-1} \subset U_0 \subset U_1 \cdots T^2 \quad (3)$$

More nuanced descriptions are possible in U_(s+1)space compared to U_s space. A scaling function, as well as its stretched and translated counterpart, is required to build a multi-resolution analysis.

$$\lambda_r^s(q) = 2^{\frac{j}{2}}\lambda(2^s q - r) \cdot r = 0, \dots, 2^s - 1 \quad (4)$$

Instead of using g $\lambda_r^s(q)$ and increasing (s) to increase the size of the subspace spanned by the scaling functions, defining a slightly different set of functions $\vartheta_r^s(q)$t that span the difference between the spaces spanned by various scales of the scale function can better describe or parameterize the important features of a signal. Wavelets are functions of the kind $U_{s+1} = U_s \oplus Y_s$, which are continuous in the wavelet space (Y_s).

$$\vartheta_r^s(q) = 2^{\frac{j}{2}}\vartheta(2^s q - r) \cdot r = 0, \dots, 2^s - 1 \quad (5)$$

The wavelet $\lambda_r^s(q)$ and the scaling function $\vartheta_r^s(q)$ together decide the specific wavelet transformation to be applied. In this investigation, we pick the Haar wavelet because it is the easiest to build and generates the least amounts of computational challenges. The transform also produces an orthogonal representation of the input images thanks to the Haar basis. For the Haar scale, the formula is as follows:

$$\lambda(q) = \begin{cases} 1 & for\ 0 \le q < 1 \\ 0 & otherwise \end{cases} \quad (6)$$

Additionally, the Haar wavelet is described as: $\vartheta(q)$

$$= \begin{cases} 1 & for\ 0 < q < \frac{1}{2} \\ -1 & for\ \frac{1}{2} \le q < 1 \\ 0 & otherwise \end{cases} \quad (7)$$

The outside and interior structure of an object can be shown with great accuracy using wavelets. Different features are good at capturing details at different magnifications. Coarse-scale features are useful for representing large regions, while fine-scale

characteristics are better suited for describing smaller, more localized areas. This data elucidates the object's multiscale structure

D. Model Training

1. CEEMDAN

While EEMD is effective at resolving the mode mixing problem, removing the Gaussian white noise injected into the method requires a significant amount of averaging to be performed. Even after reconstruction, there is still some noise present, and the number of modes is very sensitive to the nature and level of the noise. To remedy the drawbacks of EEMD, a new method called Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) was devised. When positive and negative auxiliary white noise pairs and adaptive noise components are added to the original time series data, CEEMDAN can reduce mode mixing and residual noise while preserving its cheap iteration cost and high convergence performance. The decomposition algorithm is then described in detail. Primary data from the time series is represented by J(l) the e th IMF is represented by $\overline{IMF_e}(l)$, the (s th) IMF is represented by $EMD_s()$, s (), τ_e a scalar coefficient used to modify the signal-to-noise ratio, and $\sigma^r(l)$) is white noise with a Gaussian distribution, as required by the normal distribution. A lengthy vector of time series is represented here by $J(l), \overline{IMF_e}(l), \sigma^r(l)$and i(l). For the first decomposition, we use the J(l) formed by combining the original time series J^r (l)with a white noise $\sigma^r(l)(r = 1,2, \dots, h)$ with signal-to-noise ratio 0, as given in Equation (8). Where l denotes the various time points, r the r th white noise addition, and h the total number of white noise additions.

$$J^r(l) = J(l) + \tau_0\sigma^r(l) \quad (8)$$

$J^r(l)$can be decomposed by EMD n times, yielding $J^r(l)$as a byproduct. Equation (9) is used to produce the first IMF of CEEMDAN, Equation (10) is used to obtain the first residual I_1 (l), and $EMD_1()$i is the first IMF obtained using EMD. White noise can, in theory, be rendered irrelevant by simply subtracting its average value, which is zero.

$$\overline{IMF}_1(l) = \frac{1}{h}IMF_1^r(l) = \frac{1}{h}EMD_1\left(J^r(l)\right) \quad (9)$$

$$I_1(l) = J(l) - \overline{IMF}_1(l) \quad (10)$$

The adaptive noise term is the first IMF generated by EMD, and it is just white noise σ^r (l) with a ratio of signal to noise of τ_1. The first residual I_1 (l) is then multiplied by the adaptive noise term to produce a new time series. By using Equation (11) to a new time series, they can determine the second IMF of CEEMDAN, and Equation (12) may be used to determine the second residual $I_2(l)$.

$$\overline{IMF}_2(l) = \frac{1}{h}\sum_{r=1}^{h} EMD_1\left(I_1(l) + \tau_1 EMD_1\left(\sigma^r(l)\right)\right) \quad (11)$$

$$I_2(l) = I_1(l) - \overline{IMF}_2(l) \quad (12)$$

A new time series is obtained by adding the adaptive noise component to the residual term. Decomposing CEEMDAN into σ^r (l),(r=1,2,…,h),τ_e,(e=2,3,…,E) yields the kth IMF of CEEMDAN. Here is a look at the finer points of Equations (13) and (14).

$$\overline{IMF}_e(l) = \frac{1}{h}\sum_{r=1}^{h} EMD_1\left(I_{e-1}(l) + \tau_{e-1}EMD_{e-1}\left(\sigma^r(l)\right)\right) \quad (13)$$

$$I_e(l) = I_{e-1}(l) - \overline{IMF}_e(l) \quad (14)$$

The CEEMDAN algorithm terminates when the residual term reaches a value that does not lie between the two extremes. It is possible to treat the final residual I(l) as a distinct trend term. Equation (15) demonstrates the relationship between the generated

full IMF and I(l) when applied to the initial time series.

$$J(l) = \sum_{e}^{E} \overline{IMF_e}(l) + I(l) \quad (15)$$

2. LSTM

Recurrent neural networks (RNNs) include the specialized long short-term memory (LSTM) network. The flow of information from the input layer to the hidden layer and back again is predetermined in standard feedforward neural networks. One key feature of RNNs that sets them apart from feedforward neural networks is that their recurrent cells can recall the state of each preceding element in the sequence. While feedforward neural networks are often trained using the gradient descent method, recurrent neural network models can experience exponential growth or decay during training. If the gradient abruptly disappears, it is hard to make the necessary adjustments to the weights and continue the workout. Extremely big explosive gradients, however, will carry out substantial adjustments to network parameters, which can lead to unanticipated outcomes. To filter and process information based on past states, LSTM uses memory cells instead of recurrent ones, as in an RNN [25]. The use of gates makes this a reality. An LSTM's bare-bones, unrolled memory cell. The memory cell's r_l, ft, and Ot gates provide for control of data entry and exit. The amount of input data that must be stored in the cell state at the present time (l) is determined by the input gate r_l, and the cell state is updated using the intermediate value v_l in accordance with Equations (16) and (17).

$$r_l = \omega(Y_r \cdot [n_{l-1}, q_l] + d_r) \quad (16)$$

$$v_l = \tan n(Y_a \cdot [n_{l-1}, q_l] + d_A) \quad (17)$$

The forget gate g_l, as shown in Equation (18), determines how many cell states must be maintained between time instants l-1 and l.

$$g_l = \omega\left(Y_g \cdot [n_{l-1}, q_l] + d_g\right) \quad (18)$$

As shown in Equation (19), the cell state is updated from A_(l-1)to A_l by taking away some of the prior data and adding the filtered intermediate value v_l.

$$A_l = g_l * A_{l-1} + r_l + v_l \quad (19)$$

The amount of the current cell state that must be transmitted to the new hidden state is specified by the output gate P_l as shown in Equations (20) and (21).

$$P_l = \omega\left(Y_p \cdot [n_{l-1}, q_l] + d_p\right) \quad (20)$$

$$n_l = P_l + tann(A_l) \quad (21)$$

where q_l is the input at time l, A_l and $A\_((l-1))$ are the model output states at times l and l-1, n_(l-1) and n_l are the outputs of the hidden layer at times l-1, and v_l is the cell input state at times l and t; r_l, g_l, and P_l are the outputs of the input gate, forget gate, and output gate at times l and l-1 and Y_p. This proposed used a variant of gradient descent known as the Adam optimization technique. Its adjustable learning rate is one of its most appealing features, which contributes to its widespread use in weight matrix computations. Adam is an efficient optimization method that takes the best parts of AdaGrad and RMSProp and merges them into one. It's also easy to design, uses minimal resources, and has good computational efficiency. Adam performs admirably on datasets with very noisy data and sparse or unstable gradients since the gradient transformation has no effect on parameter updates in Adam. Overfitting is a typical problem with LSTM models; it causes the trained model to do well on the training set but badly on the test set. Therefore, it is essential to prevent overfitting when training.

Result and Discussion

As a result of the development of deep convolutional networks, substantial headway has been achieved in the object detection difficulty. However, a fast-processing, highly accurate object detector is crucial for usage in autonomous vehicles.

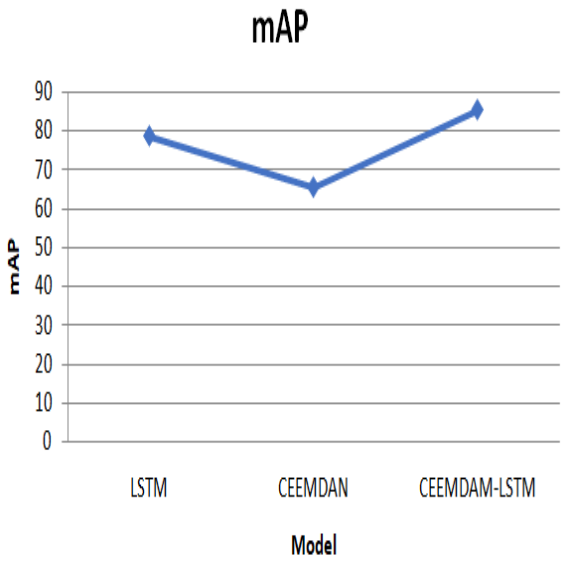

Fig1.Comparision Graph of CEEMDAN-LSTM Model

Figure 1 shows that CEEMDAN-LSTM has the highest mAP of the three models, at 85.32 percent, while the other two models only manage 78.6 and 65.4 percent, respectively. When compared to the LSTM and CEEMDAN datasets, the mAP achieved by the combined dataset network is lower.

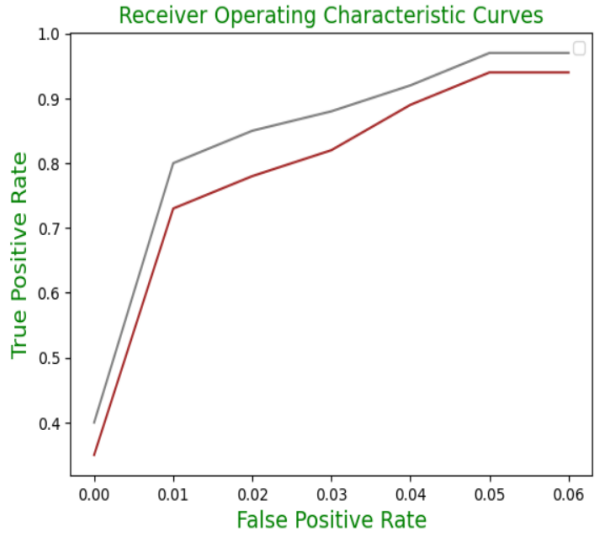

Fig. 2. Receiver Operating Characteristic Curves

As shown in Figure 2, the ROC curve is illustrated by a vertical axis corresponding to the true positive rate and a horizontal axis corresponding to the false positive rate. CEEMDAN-LSTM's detection result is depicted by the brown solid line, while detection improved by tracking results is depicted by the grey solid line. As can be seen in Figure 2, the true positive rate increases by roughly 0.4 when the false positive rate is 0.01.

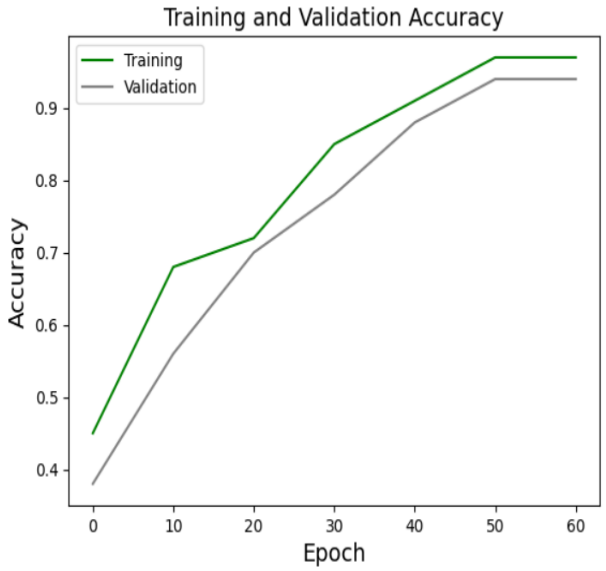

Fig. 3. Training and Validation Accuracy of CEEMDAN-LSTM Model

The CEEMDAN-LSTM model they proposed achieved 97.16 percent accuracy on validation data. Accuracy for both validation and training data is shown in Figure 3.

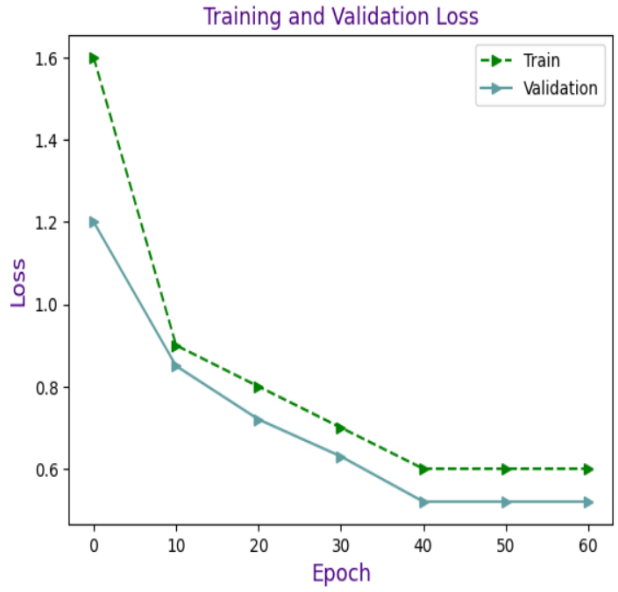

Fig. 4. Training and Validation Accuracy of CEEMDAN-LSTM Model

Conclusion

The annual increase in traffic accidents indicates an increasing threat to both human life and property. Due to the potentially devastating outcomes of automobile collisions, driver assistance systems (DAS) have recently garnered a great deal of attention. The ability to recognize vehicles is crucial for any driver assistance system. Due to the environment's complexity and the broad diversity of vehicles, real-time vehicle detection is a challenging task. Preprocessing often distorts images and videos captured by cameras in busy public spaces. An algorithm employs the depth image to divide up areas that could be driven in. They used a feature extraction method based on the Haar Wavelet Transform. In the end, CEEMDAN-LSTM is using to train the model from the retrieved characteristics. The proposed method achieves an accuracy of roughly 97.16%, which is higher than that of the CEEMDAN and LSTM models.

Acknowledgement

The author would like to appreciate the effort of the editors and reviewers.

Author Contributions

All authors are equally contributed.

Conflict of Interests

The authors declare that they have no conflicts of interest.

Ethics Approval

There are no human subjects in this article and informed consent is not applicable.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] R. Umanesan, M. N. Rao, S. Bhavana, M. A. A. Walid, P. U. Trivedi, and P. N. K. Reddy, "Innovative Method for Predicting Vehicle Speed Detection using Fuzzy Local Information Means and CNN," in 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Jun. 2023, no. Icscss, pp. 746–752, doi: 10.1109/ICSCSS57650.2023.10169523.

[2] L. Chen et al., "Surrounding Vehicle Detection Using an FPGA Panoramic Camera and Deep CNNs," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 12, pp. 5110–5122, 2020, doi: 10.1109/TITS.2019.2949005.

[3] K. Liu, W. Wang, R. Tharmarasa, and J. Wang, "Dynamic Vehicle Detection with Sparse Point Clouds Based on PE-CPD," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 5, pp. 1964–1977, 2019, doi: 10.1109/TITS.2018.2857510.

[4] H. Kuang, X. Zhang, Y. J. Li, L. L. H. Chan, and H. Yan, "Nighttime Vehicle Detection Based on Bio-Inspired Image Enhancement and Weighted Score-Level Feature Fusion," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 4, pp. 927–936, 2017, doi: 10.1109/TITS.2016.2598192.

[5] X. Hu et al., "SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 1010–1019, 2019, doi: 10.1109/TITS.2018.2838132.

[6] Q. Wang, J. Zheng, H. Xu, B. Xu, and R. Chen, "Roadside Magnetic Sensor System for Vehicle Detection in Urban Environments," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 5, pp. 1365–1374, 2018, doi: 10.1109/TITS.2017.2723908.

[7] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle Detection Techniques for Collision Avoidance Systems: A Review," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 5, pp. 2318–2338, 2015, doi: 10.1109/TITS.2015.2409109.

[8] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," Auton. Robots, vol. 26, no. 2–3, pp. 123–139, 2009, doi: 10.1007/s10514-009-9115-1.

[9] Z. Yang and L. S. C. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," Image Vis. Comput., vol. 69, pp. 143–154, 2018, doi: 10.1016/j.imavis.2017.09.008.

[10] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1773–1795, 2013, doi: 10.1109/TITS.2013.2266661.

[11] M. Fernández-Sanjurjo, B. Bosquet, M. Mucientes, and V. M. Brea, "Real-time visual detection and tracking system for traffic monitoring," Eng. Appl. Artif. Intell., vol. 85, pp. 410–420, 2019, doi: 10.1016/j.engappai.2019.07.005.

[12] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles," Expert Syst. Appl., vol. 38, no. 3, pp. 1619–1631, 2011, doi: 10.1016/j.eswa.2010.07.083.

[13] F. Erbs, A. Barth, and U. Franke, "Moving vehicle detection by optimal segmentation of the dynamic stixel world," IEEE Intell. Veh. Symp. Proc., no. Iv, pp. 951–956, 2011, doi: 10.1109/IVS.2011.5940532.

[14] M. Al-Smadi, K. Abdulrahim, and R. A. Salam, "Traffic surveillance: A review of vision based vehicle detection, recognition and tracking," Int. J. Appl. Eng. Res., vol. 11, no. 1, pp. 713–726, 2016.

[15] Y. Liu, Y. Lu, Q. Shi, and J. Ding, "Optical flow based urban road vehicle tracking," Proc. - 9th Int. Conf. Comput. Intell. Secur. CIS 2013, pp. 391–395, 2013, doi: 10.1109/CIS.2013.89.

[16] Y. Zhang, B. Song, X. Du, and M. Guizani, "Vehicle Tracking Using Surveillance with Multimodal Data Fusion," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 7, pp. 2353–2361, 2018, doi: 10.1109/TITS.2017.2787101.

[17] H. Y. Cheng, C. C. Weng, and Y. Y. Chen, "Vehicle detection in aerial surveillance using dynamic bayesian networks," IEEE Trans. Image Process., vol. 21, no. 4, pp. 2152–2159, 2012, doi: 10.1109/TIP.2011.2172798.

[18] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-Based Vehicle Detection System With Consideration of the Detecting Location," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 1243–1252, 2012, doi: 10.1109/tits.2012.2188630.

[19] C. L. Huang and W. C. Liao, "A vision-based vehicle identification system," Proc. - Int. Conf. Pattern Recognit., vol. 4, no. C, pp. 364–367, 2004, doi: 10.1109/ICPR.2004.1333778.

[20] X. Ji, Z. Wei, and Y. Feng, "Effective vehicle detection technique for traffic surveillance systems," J. Vis. Commun. Image Represent., vol. 17, no. 3, pp. 647–658, 2006, doi: 10.1016/j.jvcir.2005.07.004.

[21] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," IEEE Trans. Intell. Transp. Syst., vol. 9, no. 1, pp. 148–159, 2008, doi: 10.1109/TITS.2007.911357.

[22] R. Nabati and H. Qi, "RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," Proc. - Int. Conf. Image Process. ICIP, vol. 2019-September, pp. 3093–3097, 2019, doi: 10.1109/ICIP.2019.8803392.

[23] Y. Lv and G. Yu, "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net Wenjie," Galactose Struct. Funct. Biol. Med., pp. 1–23, 2014.

[24] Pei Sun et al., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset Pei," Cpvr, 2020, [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/Sun_Scalability_in_Perception_for_Autonomous_Driving_Waymo_Open_Dataset_CVPR_2020_paper.pdf.

[25] H. Caesar et al., "Nuscenes: A multimodal dataset for autonomous driving," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. March, pp. 11618–11628, 2020, doi: 10.1109/CVPR42600.2020.01164.

[26] A. Tariq, M. Z. Khan, and M. U. Ghani Khan, "Real Time Vehicle Detection and Colour Recognition using tuned Features of Faster-RCNN," 2021 1st Int. Conf. Artif. Intell. Data Anal. CAIDA 2021, pp. 262–267, 2021, doi: 10.1109/CAIDA51941.2021.9425106.

[27] X. Li and Y. L. Murphey, "A real-time vehicle detection and tracking system in outdoor traffic scenes," Robot Vis. New Res., no. January 2004, pp. 81–112, 2009, doi: 10.1109/icpr.2004.1334370.

[28] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," Proc. IEEE Work. Appl. Comput. Vis., vol. 2002-Janua, pp. 171–176, 2002, doi: 10.1109/ACV.2002.1182177.

[29] L. Chen, X. Liu, C. Zeng, X. He, F. Chen, and B. Zhu, "Temperature Prediction of Seasonal Frozen Subgrades Based on CEEMDAN-LSTM Hybrid Model," Sensors, vol. 22, no. 15, 2022, doi: 10.3390/s22155742.