Hindawi

Research Article

Computer Vision and the IoT-Based Intelligent Road Lane Detection System

R. Shashidhar ⓘ,[1] **B. N. Arunakumari ⓘ,**[2] **A. S. Manjunath,**[3] **Neelu Jyoti Ahuja,**[4] **Vinh Truong Hoang ⓘ,**[5] **Kiet Tran-Trung ⓘ,**[5] **and Assaye Belay ⓘ**[6]

[1]*Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysuru 570006, India*
[2]*Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru 560064, India*
[3]*Department of Computer Applications, JSS Science and Technology University, Mysuru 570006, India*
[4]*Department of Systemics, School of Computer Science, University of Petroleum & Energy Studies, Dehradun, India*
[5]*Faculty of Computer Science, Ho Chi Minh City Open University, Vietnam, 97 Vo Van Tan, Ward Vo Thi Sau, District 3, Ho Chi Minh City 70000, Vietnam*
[6]*Department of Statistics, Mizan-Tepi University, Tepi, Ethiopia*

Correspondence should be addressed to Assaye Belay; abstat23@gmail.com

Many technical improvements have recently been made in the field of road safety, as accidents have been increasing at an alarming rate, and one of the major causes of such accidents is a driver's lack of attention. To lower the incidence of accidents and keep safe, technological innovations should be made. One way to accomplish this is with IoT-based lane detection systems, which function by recognizing the lane borders on the road and then prompting the turning of the road. Because of the various road conditions that one can encounter when driving, lane detection is a difficult problem. An image processing-based method for lane detection has been proposed in this paper. In this regard, each frame from the video is extracted and image processing techniques are applied for the detection of lanes. The frame which is extracted from the video is then subjected to a Gaussian filter for the removal of noise. Subsequently, color masking has been used to process the frame to detect only the road lanes, whose edges are obtained by applying the canny edge detection algorithm. Afterward, the Hough transform has been applied to the region of interest to extrapolate the lines. Finally, the path is plotted along the lines, and turns are predicted by using the concept of vanishing points.

1. Introduction

In today's world, numerous technical advancements are coming in the field of the automobile industry, and everything is getting automated. One of the examples is self-driving cars (cars that can navigate without drivers). One of the ways to achieve the same is to interface the IoT-based road lane detection system with the cars, which works by recognizing the lane borders or boundaries on the road and helps the car to navigate and guides it to take turns.

Many systems, such as lane departure warning, adaptive cruise control, lane change aid, turn to assist, time to lane change, and fully autonomous driving systems, have been actively researching IoT and vision-based road lane recognition. Although specialized systems for recognizing specific road types have made great progress, little work has been achieved in presenting a general method to detect a range of road conditions. In straight and curved, white and yellow, single and double, solid and broken, and pavement or highway lane limits, an effective lane identification system will navigate autonomously or aid drivers. Intelligent vehicles and smart infrastructure work together in intelligent transportation systems to create a safer environment and better traffic conditions. Lane detection is a critical component of intelligent vehicle systems. As a result, we want to demonstrate a reliable road lane marker detecting system.

However, a more compelling incentive to develop intelligent vehicles is to increase safety by automating or

partially automating driving responsibilities. Among these functions, road detection is critical in driving assistance systems that offer data on the lane structure and the vehicle position relative to the lane. The most compelling reason to equip vehicles with autonomous capabilities is to meet safety requirements. As a result, a system that alerts the driver about the danger has the potential to save a significant number of lives. Computer vision, which has become a powerful tool for perceiving the surroundings and has been widely used in many applications by intelligent transportation systems, is one of the primary technologies involved in these (ITS). Lane detection is often described as the localization of certain primitives such as road markings on the surface of painted roads in several proposed systems. This restriction simplifies the detection process; nonetheless, two conditions can obstruct it: the presence of other vehicles in the same lane that partially obscure the road markings ahead of the vehicle as well as the existence of shadows cast by trees, buildings, and other structures. This study proposes a vision-based method for detecting and tracking structured road boundaries with a slight curvature in real-time even in the presence of shadows.

Lane borders can be identified in real-time by using a camera fixed on the front of the vehicle, which captures the view of the road. In this study, we will use a prerecorded video of the road as an input, extract all of the frames as a set of images, and then extract the characteristics from each of those images, which will be used to detect lanes and predict turns.

This system will not depend much on external factors. It only needs a clear picture of the road. This system is not justmeant for self-driving cars; we can even implement it in normal cars which will give a warning to the driver when he crosses the line on a highway because of drowsiness or some other reasons. It will also help the driver to automatically turn on the indicator if he wishes to cross the line. The major objective of the proposed work is to extract frames from a video and apply image processing techniques to detect the lanes. Join all the lanes and form two lines at the boundaries of the path. Plot the path between the lanes. Predict the turn-based on the length of the boundary lines. Display the turnover of the image. Display the possibility of overtaking based on the type of road lane. Contributions of the proposed work is to develop and implement a road lane detection algorithm using the basic image processing techniques and plot a path between the road lanes in which an autonomous vehicle has to move by predicting the turn direction and also predict the possibility of overtaking based on the type of road lane.

The major contributions of the proposed work are as follows: (i) Conventional image processing techniques such as edge detection, Hough transform, and line extrapolation to detect the road lanes are very accurate and easily implementable than machine learning methods and also comparatively faster; (ii) we proved an IoT and vision-based approach capable of reaching a real-time performance in the detection and tracking of structured road boundaries with a slight curvature, which is robust enough in the presence of shadow conditions; and (iii) proved machine learning as high error-susceptible than in our work.

The remaining sections are as follows: Section 2 described the literature survey of the exciting work; Section 3 describes the proposed methodology in detail; Section 4 mentioned the result and comparison table; and Section 5 concludes the proposed work with future scope.

2. Related Work

Vehicle wrecks continue to be the largest source of death and injury around the globe, costing the lives of tens of thousands of people each year and hurting millions more. The majority of these transportation-related deaths and injuries happen on America's roadways. As autonomous cars are ready to fill up the roads, the chances of the crash rate will increase more, which is a most concerning issue. This paper proposes a vision-based solution to real-time recognition and tracking of structured road boundaries with small curvature that is strong enough in the face of shadow situations.

In complicated driving settings, a robust lane recognition model based on vertical spatial features and contextual driving information is proposed [1–4]. Two developed blocks such as the feature merging block [5] and the information exchange block illustrate a more effective utilization of contextual information and vertical spatial features, allowing detection of more unclear and occluded lane lines. Hough transform applied to detect lane markers after picture noise filtering devised utilizing an image processing computer language platform is another efficient way [6]. Using a pair of hyperbolas that are fit to the boundaries of the lane and extracted using the Hough transform, a vision-based lane identification strategy capable of approaching real-time operation with robustness to illumination change and shadows [7] is also described. Two approaches to recognizing automobiles on the road are appearance-based and feature-based [8–10]. Optical flow, backdrop subtraction, and frame subtraction are the most often utilized methods for vehicle detection [11]. Background subtraction is the most well-known and widely utilized approach today [9]. Furthermore, tail light pairs can be used for night-time vehicle detection [11]. Other methods for recognizing automobiles on the road include texture descriptors [10], Haar filters [12], and others. A deep architecture [13] that can run in real-time while giving accurate semantic segmentation also exists. Our architecture is built around a new layer that employs residual connections and factorized convolutions. In addition, a multitask deep convolutional network [14] recognizes both the presence of the target and its geometric features with the region of interest. For structured visual identification, a recurrent neural layer [15] is used to deal with the spatial distribution of observable cues relating to an object whose shape or structure is difficult to characterize directly. Machine learning is not flawless despite its many benefits and widespread appeal. Data acquisition, time and resources, and result interpretation are some of the factors, which make

machine learning high error-susceptible. In the modern era, IoT is proposed for the traffic system. Several sensors are integrated using a microcontroller for a smart traffic system in an emergency [16–20].

To solve this problem, we intend to use predictable image processing techniques such as edge detection, Hough transforms, and line extrapolation to detect the road lanes, which is very accurate and easily implementable than other methods, and also comparatively faster. The most important point is that the above-mentioned techniques only detect the lanes but the latter also predicts the turns by extrapolating the lanes.

3. Material and Methods

3.1. Dataset. The source of image frames given to the system is data collected from the camera located between the front windscreen and the rear-view mirror. The taken image frames can be separated into foreground and background fields when the camera lens is parallel to the ground. By selecting an adequate ROI, you may reduce not just the search area in photos but also the interference from unwanted items. We have so far relied on datasets that were freely available on the Internet. https://www.kaggle.com/soumya044/advanced-lane-detection.

3.2. Methods. The four main core methods used for this work are as follows: First is color masking; second is edge detection; the third is region of interest selection; and the fourth method is Hough line transform extrapolation.

Different methods modalities are found useful for the task of road lane detection, each having its pros and cons. This section elucidates a simple yet effective strategy for doing so, explains the major characteristics and usual applications, and breaks down the road lane detection task into functional modules, enumerating the approach to each module's implementation.

3.2.1. Proposed Work. To implement the proposed method, MATLAB's Image Processing Tool has been used, and the block diagram of is shown in Figure 1. To detect the lanes and extract the features of the lanes in every frame of the video, we use color masking, edge detection, ROC selection, Hough transform selection, and other techniques.

The frame read from the video is subjected to a Gaussian filter to remove the noise. Then it is masked with yellow and white colors to detect the road lanes properly. Then the edges of the lanes are obtained from the masked images by applying the canny edge detection technique. Then the region of interest, i.e., road lanes will be selected and then the Hough transform will be applied to extrapolate the lines. Finally, the path will be plotted along the lines and turns will be predicted by observing the vanishing points from the extrapolated lines.

If the right vanishing point is greater than the left vanishing point, then the system will predict that there is a right turn ahead and vice versa. The source of image frames given to the system is data collected from the
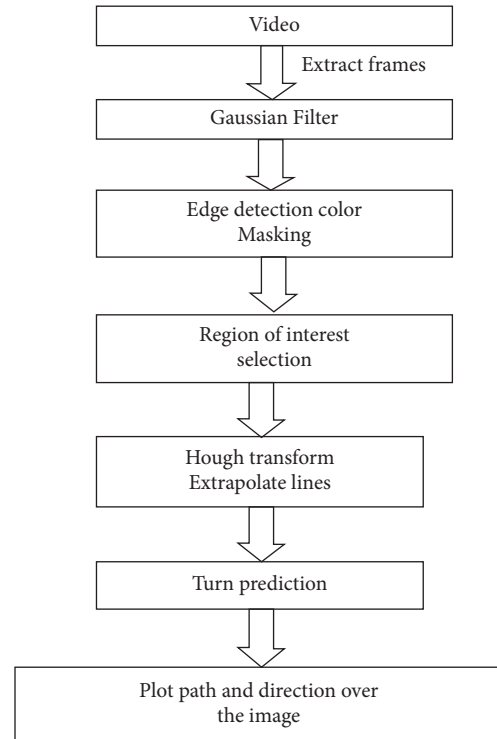

FIGURE 1: Block diagram.

camera located between the front windscreen and the rear-view mirror. The taken image frames can be separated into foreground and background fields when the camera lens is parallel to the ground. By selecting an adequate ROI, you may reduce not just the search area in photos but also the interference from unwanted items. As of now, we have used the datasets which were available online. We have surfed and searched for the dataset videos which would perfectly match the appropriate application which we are thinking of.

The four main core concepts that we intend to use in this method are color masking, edge detection, region of interest (ROI) selection, and Hough line transform extrapolation.

3.2.2. Gaussian Filter. A linear filter is a Gaussian filter. It is frequently used to blur or minimize the noise in an image. For "unsharp masking," two of them are employed and subtracted (edge detection). The Gaussian filter blurs edges and reduces contrast by itself. Figure 2shows the frame of the image after the application of the Gaussian filter to Figure 3.

The Gaussian filter function is defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-x^2 + y^2/2\sigma^2}. \tag{1}$$

The amount of filtering is inversely proportional to the sigma (variance) value. More frequencies are muted when sigma is less, and vice versa. The coordinates of the pixel are denoted by x and y.

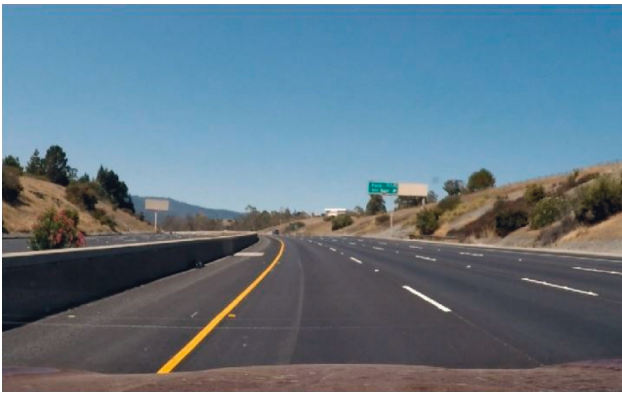Figure 2: Picture frame after applying the Gaussian filter.
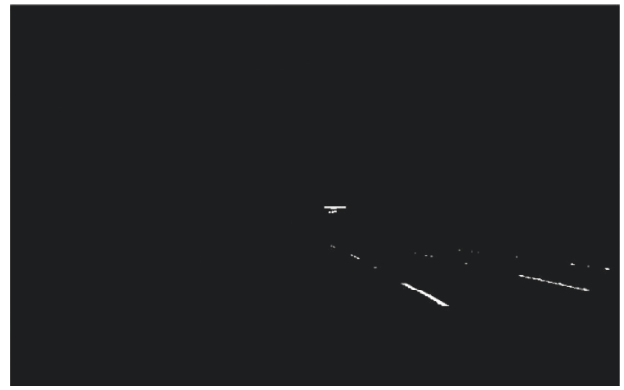

Figure 3: Original picture frame.

Figure 4: Picture frame after applying color masking.

3.2.3. Color Masking.

Color masking is the conversion of a particular color pixel in an image or relative color pixels of an image into white and other colors into black by altering the pixel values, as shown in Figure 4. It is similar to thresholding, but here, instead of a grey scale image, we will take a color image.

For example, to detect the white color in an image, the RGB value of white color (R, G, B) = (255,255,255) is applied. If   (R, G, B) = (250 − 255, 250 − 255, 250 − 255), then the color mask of the white color is given by

$$P(x, y) = \begin{cases} 1, & if \; C(x, y, z) = (R, G, B), \\ 0, & otherwise. \end{cases} \qquad (2)$$

3.2.4. Edge Detection.

The term "detection" refers to the process of identifying rapid changes in the intensity of a pixel in a picture. In this work, the Canny edge detection method is used to identify the edges, whose output is as shown in Figure 3 when applied over Figure 4.

The Canny edge detection algorithm is a gradient-based edge detection technique; to use this method, first we have to optimize the signal-to-noise ratio using a Gaussian filter or any other filtering technique Algorithm 1.

Canny edge detection algorithm steps:

3.2.5. Region of Interest Selection.

There are different methods for a region of interest selection; in this work, the approach of selecting the polygon in the image formed from a certain set of predetermined points is followed, as shown in Figure 5.

3.2.6. Hough Transform.

The Hough transform is an image processing technique for isolating features of a specific shape in an image. The Hough transform is most typically used to detect regular curves like lines, ellipses, circles, and so on.

The Hough transform can be used to find lines in a picture using the parametric representation of a line:

$$\rho = x * \cos(\theta) + y * \sin(\theta), \qquad (3)$$

where ρ is the distance along a vector perpendicular to the line from the origin to the line, and θ is the angle formed between the vector and the x-axis.

3.2.7. Turn Prediction.

Fitting a trapezium to the visual data gathered in a single frame is commonly used to direct turn prediction from the top down. Both lanes, which are modeled as a 2D path with left and right borders, employ similar model-fitting approaches. By adopting a smooth path model with limits on its breadth and curvature, the noisy bottom-up path detection is enhanced. In the temporal

Step 1:
(i)   Smoothing: Using the Gaussian filter
(ii) $f_x = f * G_x$, where $G_x = (-x/\sigma^2).G(x, y)$
(iii) $f_y = f * G_y$, where $G_y = (-y/\sigma^2).G(x, y)$
Step 2:
(i)   Gradient magnitude: Sobel or Prewitt
     $|G| = \sqrt{G_x^2 + G_y^2},$
     $\approx |G_x| + |G_y|.$
Step 3:
(i)   Edge direction
     $\theta = \tan^{-1}(G_y G_x)$
Step 4:
(i)   Resolve Edge Direction
Step 5:
(i)   Nonmaxima suppression: keep all local maxima in the gradient and remove everything else
(ii)   Gives a thin edge line
Step 6:
(i)   Double or hysteresis thresholding

ALGORITHM 1: Canny edge detection Algorithm.

integration stage, this path representation is often fine-tuned by comparing it to earlier frames. A path is often described by its boundary points or its centerline and lateral extension at each centerline position, which defines the borders uniquely.

The last step is to predict the lane heading for the vehicle. This is calculated using the position of the vehicle for lanes at any given point. The pixel coordinates of the extrapolated lane lines are stored in the form of a matrix. First, the thick 2D plane of both lanes is converted to a 1D line at the center of the plane, using the concept of projection, i.e., projection of $\overline{b}$ on $\overline{a} = (\overline{a}.\overline{b}/|\overline{b}|)$.

First, the central point of the lane is calculated using the position of both the left and right lanes. Then, these central coordinates are stored in another matrix. Further, the perpendicular distance between the two lanes is calculated by measuring the projection of one lane over the other. This gives the central point of the road. Then, the vanishing point is found by taking a cross product of the obtained two vectors.

4. Algorithm

The algorithm for the proposed work is given as Algorithm 2.

The radius of curvature is calculated using the derivatives of the polynomial equation estimated using the polyfit parameters. Both the first and the second derivatives of the curve are used for calculating the radius of curvature for both the right and the left lanes separately. The radius of curvature is given as follows:

$$R_{\text{curve}} = \frac{\left[1 + (dx/dy)^2\right]^{3/2}}{\left|d^2x/dy^2\right|}. \tag{4}$$

Then, this radius of curvature and the position of the center of the vehicle, that is, the center of the image is used to predict the lane turn. The difference between the lane center

FIGURE 5: Picture frame after selecting the ROI.

and the image center is calculated. This is calculated for both lanes. The vector cross product is calculated to obtain a vanishing point.

$$\overline{A} \times \overline{B} = |\overline{A}||\overline{B}|\sin\,\theta. \tag{5}$$

The ratio of the vanishing point and the frame size results in a new parameter called the vanishing ratio. This value of the vanishing Ratio is compared to predict the turns as follows:

If vanishing ratio is 0.5, then "Straight" is predicted.

If $0.47 \leq$ vanishing ratio $M < 0.49$, then the "Left" turn is predicted.

If the vanishing ratio> 0.5, then the "Right" turn is predicted.

5. Result and Discussion

This section shows how the proposed road lane recognition system performs in real-world scenarios. The left and right lane markers are detected and extracted by the algorithm. The suggested algorithm is put to the test using video pictures captured with an onboard vision sensor.

Step 1: A couple of videos are used as data to detect the lanes in the videos. The video is loaded to read each frame using the function *VideoReader* of Matlab.

Step 2: The Gaussian filter is applied to every frame in the image to remove noises in the image using the function *imgaussfilt3*.

Step 3: Yellow and white color masking is done to the frame.

Note: Lanes on the road are present in yellow and white colors.

Step 4: Then the *Canny* edge detection algorithm is applied to identify the edges of the yellow and white color regions in the image.

Step 5: The unwanted small group of pixels is removed using the function *bare open*.

Step 6: The region of interest is selected using the function *roipoly* in every frame of the video. This yields the region of the image where only lanes are present.

Step 7: The Hough transform is applied to the interested region to detect the lanes using the function *hough*. Further, Hough peaks function is used to determine the parameter space's peak values. The potential lines in the input image are represented by these peaks.

Step 8: The Hough lines function is used to locate the endpoints of line segments in the Hough transform that correspond to peaks. This function fills in minor gaps in line segments automatically.

Step 9: The slope of the left and right lanes is calculated, which is further used to extrapolate the broken road lanes if any.

Step 10: The cross product of both the vector lines is calculated to obtain the vanishing point, and its ratio to the frame size yields the vanishing ratio.

Step 11: The turn is predicted based on the obtained magnitude of the vanishing ratio (VR).

(i) Straight if VR ≃ 0.5.

(ii) Left if $0.47 \leq VR < 0.49$.

(iii) Right if $VR > 0.5$.

ALGORITHM 2: Hough transform method.
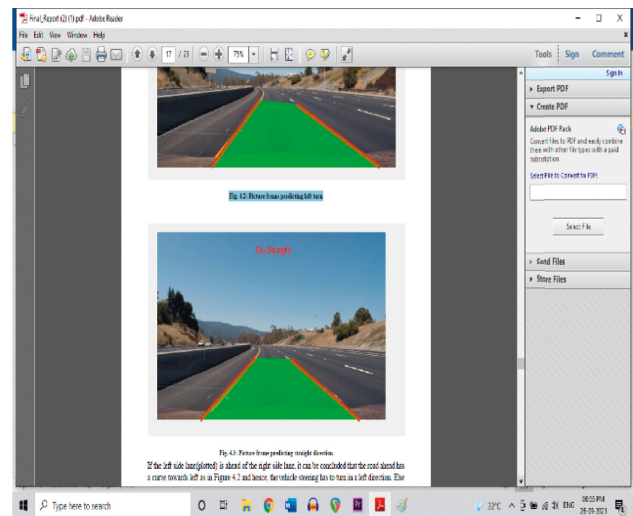

FIGURE 6: Picture frame predicting the right turn.

FIGURE 8: Picture frame predicting the straight direction.

5.1. Evaluation Metrics

5.1.1. Accuracy.
The accuracy of the proposed method is 92%. In the proposed technique, we used a custom dataset of around fifty frames; in fifty frames, forty-six frames showed the correct path, so the accuracy of the proposed method is 92%.

5.1.2. Error Rate.
The error rate of the proposed method is 8%. Out of fifty frames, forty-six frames show correct output, so we consider the reaming part as an error rate.

The capacity to evaluate algorithms is necessary for comparing and identifying the performance of various

FIGURE 7: Picture frame predicting the left turn.

TABLE 1: Comparison of the results of different proposed methods.

	Liu et al. [1]	Kim et al.[12]	Proposed method
Dataset	Custom dataset	Custom dataset	Custom dataset
Accuracy	91.1%	84.54%	92%
Error rate	8.9%	15.45%	8%

algorithms and methodologies. However, because of a lack of standardized test methodologies, performance measures, and publicly available datasets, this issue is extremely troublesome in the lane and road detection literature. Many works result in terms of efficiency only, but the complete breakdown of the number of frames tested, the number of frames successful, and the number of failure frames are not quoted for the existing methods. The lanes in front of a vehicle are successfully detected, and the path between those lanes is plotted in green color and lanes are marked in red color. It is clear from Figure 6 that the right sideline (plotted) is ahead of the left side lane, which indicates that there is a slight right curve in the road, so the steering of the vehicle should turn slightly towards the right side. This is predicted by finding the vanishing points and vanishing ratios of the two lanes.

If the left side lane (plotted) is ahead of the right side lane, it can be concluded that the road ahead has a curve towards the left, as in Figure 7, and hence, the vehicle steering has to turn in a left direction. Else if both the left and right lanes are equal in length, as in Figure 8, it can be inferred that the vehicle needs to proceed straight, with the steering in the steady position, since the vanishing ratio is 0.5. The direction in which the car has to travel is displayed over each frame. Table 1 shows the comparative result of the existing result and proposed result.

6. Conclusion

Lane detection is often complicated by varying roads, markings, clutter from other vehicles and complex shadows, lighting, occlusion from vehicles, and varying road conditions. In this work, we have presented a solution to the lane detection problem that shows robustness to these conditions.

This work focuses on road lane detection using basic image processing techniques. The lane detection algorithm presented has been tested on both straight and curved lanes. The algorithm passes through a series of low-level image processing to make useful information easier to be extracted. Then the road features are extracted with Canny edge detection. Hough transform is applied to find relevant lines that can be used as the left and right lane boundaries. Finally, the collected left and right lane boundaries are averaged to get the desired result of left and right boundaries and are shown on the original image. To reduce the high computational cost, the image is reduced to a smaller region of interest. The smaller threshold value is applied to edge detection for more complex environments. Experiments show that this algorithm can achieve fairly good performance in different kinds of road conditions.

The future scope of the work is that this model can be updated and tuned with more efficient mathematical modeling, whereas the classical OpenCV approach is limited and no upgrade is possible as the approach is not efficient. It is unable to give accurate results on the roads which do not have clear markings present on the roads. Also, it cannot work for all climatic conditions.

This technology is increasing the number of applications such as traffic control, traffic monitoring, traffic flow, and security. This can be integrated with a vision-based obstacle-detection algorithm, for example [8], for a collision-warning system. This can be further expanded to incorporate piecewise estimates of trajectory and curvature. Also, this system can be combined into a vehicle surround analysis system to create an intelligent vehicle driver assistance system. This lane detection algorithm can be applied with position tracking to reduce computational load. With real-world positioning applied to the image, distance and time to departure can be obtained.

With the combined parameters, a new lane departure warning system can be introduced.

Data Availability

This study used the data available here: https://www.kaggle.com/soumya044/advanced-lane-detection.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] W. Liu, F. Yan, J. Zhang, and T. Deng, "A robust lane detection model using vertical spatial features and contextual driving information," *Sensors*, vol. 21, no. 3, 708 pages, 2021.

[2] D. A. Goyal, D. M. Singh, and A. Srivatava, "Lane detection on roads using computer vision," *Vision International Journal of Engineering and Advanced Technology (IJEAT)ISSN*, vol. 9, pp. 1200–1205, October 2019.

[3] J. Jung and S. H. Bae, "Real-time road lane detection in urban areas using LiDAR data," *Electronics*, vol. 7, no. 11, 276 pages, 2018.

[4] C. Y. Low, H. Zamzuri, and S. A. Mazlan, "Simple robust road lane detection algorithm," in *Proceedings of the 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, pp. 1–4, Kuala Lumpur, Malaysia, June 2014.

[5] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Raz advanced technical center Israel, general motors R&D, 7 HaMada st., 46725 herzliya, Israel, "recent progress in road and lane detection: a survey"," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[6] K. Ghazali, R. Xiao, and J. Ma, "Road lane detection using H-maxima and improved hough transform," in *Proceedings of the Fourth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 205–208, Kuantan, Malaysia, September 2012.

[7] A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Proceedings of the International Conference on Computer and Communication Engineering*, pp. 82–88, Kuala Lumpur, Malaysia, May 2008.

[8] Z. Kim, "Real-time obstacle detection and tracking based on constrained Delaunay triangulation," *Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference*, pp. 548–553, Toronto, ON, Canada, September 2006.

[9] J. C. McCall and M. M. Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 533–537, Parma, Italy, June 2004.

[10] D. Li, D. Zhao, Q. Zhang, and Y. Chen, "Reinforcement learning and deep learning based lateral control for autonomous driving [application notes]," *IEEE Computational Intelligence Magazine*, vol. 14, no. 2, pp. 83–98, 2019.

[11] Y. Xing, C. Lv, L. Chen et al., "Advances in vision-based lane detection: algorithms, integration, assessment, and perspectives on ACP-based parallel vision," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 645–661, 2018.

[12] Z. W. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, 2008.

[13] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. E. Arroyo, "Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, 2017.

[14] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, 2017.

[15] W. Yang, X. Zhang, Q. Lei, D. Shen, P. Xiao, and Y. Huang, "Lane position detection based on long short-term memory (LSTM)," *Sensors*, vol. 20, no. 11, 3115 pages, 2020.

[16] G. R. Nagarjuna, R. Shashidhar, S. B. Puneeth, and B. N. Arunakumari, "IoT enabled smart traffic system for public and emergency mobility in smart city," in *Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 53–59, Palladam, India, October 2020.

[17] D. Koundal, "Texture-based image segmentation using neutrosophic clustering," *IET Image Processing*, vol. 11, no. 8, pp. 640–645, 2017.

[18] H. Yar, T. Hussain, Z. A. Khan, D. Koundal, M. Y. Lee, and S. W. Baik, "Vision sensor-based real-time fire detection in resource-constrained IoT environments," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–15, Article ID 5195508, 2021.

[19] G. Madhu, B. Lalith Bharadwaj, R. Boddeda et al., "Intelligent disease diagnosis model for energy aware cluster based IoT healthcare systems," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1189–1203, 2022.

[20] K. Bhalla, D. Koundal, S. Bhatia, M. Khalid Imam Rahmani, and M. Tahir, "Fusion of infrared and visible images using fuzzy based siamese convolutional network," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5503–5518, 2022.