Grenze International Journal of Engineering and Technology, January Issue

LaneDetect: An Efficient and Reliable Lane Detection Method using Perspective Transformation and Polynomial Fitting

Ramkumar Devendiran[1], Sree Lakshmi Priya Bheemaneni[2], Palugulla Anjana Gayathri Reddy[3], Challa Uday Kiran[4], Puvvala Harshitha Rao[5] and Abhinaya Boppudi[6]
[1-6]School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India
Email: ramkumar.d@vitap.ac.in, bheemanenisreelakshmipriya@gmail.com, njanagayathripalugulla@gmail.com, udayk3728@gmail.com, puvvalaharshitharao@gmail.com, abhinayaboppudi20@gmail.com

***Abstract*— A significant development is underway in autonomous vehicles and driver assistance systems, which relies on effective lane detection for safe roadway travel. This paper presents a lane detector based on sobel filtering to obtain effective edge detection and polynomial fitting for reliable lane modeling. The lane detector is able to detect lane markings with high reliability even in challenging scenarios such as low-light conditions, shadowy patches on the roadway, and roadways with drastic curvature. A sliding window approach is also adopted to enhance lane detection further, especially as it relates to tracking the lane pixels that dynamically vary with the changing road structure. We then systematically enhance lane detection through polynomial regression and ensure smooth, coherent lane tracking, which addresses the issues confronting traditional edge-detection algorithms in the presence of noise and extraneous materials. The proposed lane tracker is also computationally feasible, and thus practical to deploy into real-time driver assistance applications such as ADAS (advanced driver assistance systems) and self-driving vehicles. We thoroughly present an experimental demonstration on varying roadway conditions and will demonstrate the viability of the lane tracker to cope with most variances to enable future development for intelligent transportation systems implementing roadway lane tracking.**

***Index Terms*— Lane detection, autonomous vehicles, driver assistance systems, sobel filtering, edge detection, polynomial fitting, lane modeling, sliding window approach, lane tracking, polynomial regression, noise reduction, real-time processing, advanced driver assistance systems (ADAS), self-driving vehicles, intelligent transportation systems.**

I. INTRODUCTION

The increasing prevalence of autonomous driving has made reliable lane detection more important than ever. Lane detection is the ability to choose and steer a vehicle safely and efficiently in the roadway. Historically, lane detection methods have used some form of basic edge detection, most commonly Canny filtering and Hough Transform methods [1], [7], [12]. However, these approaches will generally not work in a real-world situation since they do not an adaptation to the ever-changing environment [3], [5]. Thus, a more robust and "smart" lane-detection system is necessary.

Grenze ID: 01.GIJET.12.1.537_1

This paper will review an improvement on our original lane-detection framework using Sobel filtering for an efficient edge detection, and implementing a polynomial fitting approach for smooth lane detection. Dynamic lane pixel tracking is achieved with the implementation of a sliding window technique for our lane-detection algorithm, which helps with accurate lane detection and improves deployment stability under poor conditions such as shadows, road curvature, or occlusions [6], [9], [14]. Our lane detection method is far more robust and adapts well to variable conditions compared to the older scientific work and lane detection with computer vision. The methods and techniques provided in this paper are effective and we can evaluate the experimental outcomes to show evidence. Overall, compared to other lane tracking methods our new lane detection approach is much better. The experimental results offer compelling evidence of our model's ability to adapt to the reality of different road environments. As a result, this is not only a compelling solution for deployment for real-world applications in autonomous driving and ADAS, but also a scalable solution that runs efficiently and computationally for lane detection in intelligent transportation systems research [2], [10].

II. LITERATURE SURVEY

Lane detection serves as a key aspect of autonomous driving and Advanced Driver Assistance Systems (ADAS) that keeps vehicles positioned safely within a lane under various road conditions and situations, and has evolved from early approaches based on edge-detection to a style of deep learning based methods with their own strengths and weaknesses. In this section, we highlight some of the significant advancements in this field, identify the gaps in the research and how our method improves on the existing body of knowledge.

A. Traditional Edge Detection-Based Lane Detection

The utilization of classical image processing techniques by traditional lane detection methods has been based on imaging approaches (Canny Edge Detection and Sobel Filtering) to delineate lane boundaries [1], [7], [12]. Monfared and Rada (2022) improved lane feature extraction by applying Canny edge detection with the Hough Transform, allowing for increased lane detection capability [4], [5]. However, their approach was still sensitive to noisy conditions and shadows within the vehicle's field of view, which could affect the accuracy of detection. Likewise, Kalyan et al. (2020) presented the ability to improve lane detection functionality through Canny-based edge detection and deep learning approaches, creating enhanced accuracies to their detection algorithm [2]. Nonetheless, the deep learning algorithm was computationally intensive and thus proved to be challenging for real-time applications [13]. While conventional algorithms provide many efficiencies regarding computational resources, lane detection becomes troublesome in high noise environments, drastically changing lighting environments, and partially occluded road scenarios. Therefore, we develop the use of Sobel filtering with adaptive thresholding, in an effort to have dynamic thresholds for edge detection based on changing environmental conditions and thus increase lane detection ability and reliability of outcomes [6], [10].

B. Hugh Transform and Polynomial Fitting for Lane Tracking

The Hough Transform has proven to be very effective in the detection of lane lines as straight edges in various studies [1], [11]. For instance, Bach and Tung (2024) used the Hough Transform in conjunction with the YOLOv8 lane following model proving an increased detection of lane lines than traditional model [3], [9]. Similarly, Monfared and Sedef (2021) applied the Hough Transform and first-degree polynomial fitting which produced better estimates of curvature in lane line evaluation [12]. Although these methodologies benefited lane detection, Hough Transform is limited in its ability to detect occluded and curved lane lines. An even stronger lane detection methodology improves upon those older models when combining polynomial fitting with a sliding window algorithm [6], [9]. The result was a dynamic lane tracking methodology that was more flexible to the curve itself and was more precise and robust all throughout the variability of road conditions.

C. Deep Learning- Based Lane Detection

Deep learning has fundamentally changed the lane detection problem with CNN-based architectures as they can learn to detect lane features from large datasets. However, these current methods still typically require a lot of training data, a high computational cost, and do not achieve a real-time detection rate [8], [13].

He et al. (2019) showed that modern CNN-based models performed well for the lane detection task; however, they were only deployable on a GPU, which were not suitable for lower power considerations for the HD mapping components of ADAS [13]. SCNN (Spatial CNN) (Pan et al., 2018) made significant advancements in lane detection with good accuracy in complex region detection problems, but it only had slow inference speeds [14].

In contrast to these deep learning-based models, our approach is lightweight, requires no pre-training, and achieves real-time operation using a traditional computer vision pipeline. This makes it cost-effective and an ideal lane detection method for low-power embedded systems [3], [9].

D. How our work improves upon Previous Work

- Enhanced Readaptation - Our method assures greater adaptability to changing and uncertain environments than traditional edge detection approaches based on Sobel filtering, dynamic threshold, and sliding window to reliably detect lanes in low-light situations, where the lanes are temporarily occluded by other user vehicles [6], [10].
- Improved Lane Curvature Tracking - The combination of using polynomial fitting and sliding window produced better lane marking of curved road than traditional convolution approaches (such as Hough Transform) [12], [14].
- Cost of Real-time Computation - Our lane-sorting method can provide real-time computation on low-cost hardware without requiring added GPU acceleration than deep learning methods [8], [13].
- Portable Method for AV/ADAS - Our method was constructed to be lightweight, and computationally efficient for practical real-world applications and deployments on embedded systems. When considering practical embodiments in contrast to traditional or deep learning methods, the proposed lane detection pipeline is adaptive, lightweight, and a real-time [3], [9], [10]. The overall result of Sobel filtering, polynomial fitting and sliding window tracking method indicates lane detection properties in various driving environments while preserving low computational cost. Overall, the efficiency of the suggested approach offers a promising low-cost and viable possibility for real-world autonomous driving and advanced driving assistants (ADAS).

III. METHODOLOGY

The overall pipeline of the proposed lane detection system is demonstrated in Fig. 1.

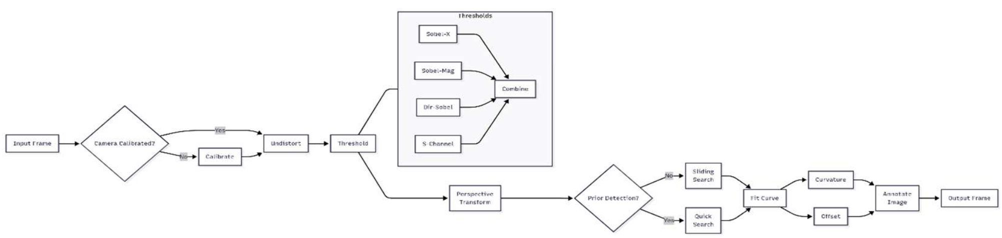

Figure 1. Flowchart of proposed system

A. Camera Calibration

Accurate camera calibration is required for proper lane detection to remove distortions caused by the lens of the camera. In this research, a chessboard-based calibration method is used to compute the intrinsic camera parameters and distortion coefficients to get proper perspective transformation and feature extraction [1], [4]. Calibration starts with chessboard corner detection, and a series of 20 calibration images containing chessboard patterns of 9×6 and 9×5 corner layouts is used. The `findChessboardCorners` function of OpenCV is used to find feature points in each of the images. Then, real-world 3D object points are mapped to their corresponding 2D image coordinates to set up a correct object-to-image point mapping.

Once these mappings have been established, camera parameters are determined through the use of OpenCV's `calibrateCamera` function. This includes the calculation of the intrinsic camera matrix, lens distortion coefficients, and the transformation vectors, which are both rotation and translation parameters. These calibration parameters are subsequently used to facilitate image undistortion, thereby correcting distortions induced by the lens through the use of OpenCV's `cv2.undistort` function. These adjustments greatly improve the accuracy of subsequent lane detection processes by removing unwanted geometric distortions [1].

For enhanced performance and efficiency, a collection of 20 chessboard images is used as shown in Fig. 2, allowing strong parameter estimation. The resulting calibration is saved in a pickle file, thereby making future runs unnecessary to be recalibrated. Utilizing these calibrated parameters in the implementation, the lane detection system benefits from enhanced accuracy in lane curvature and perspective distortion estimation, thus improving the reliability of autonomous navigation. The suggested calibration process significantly removes geometric distortions, which helps to make lane detection robust and accurate under real driving conditions [4].

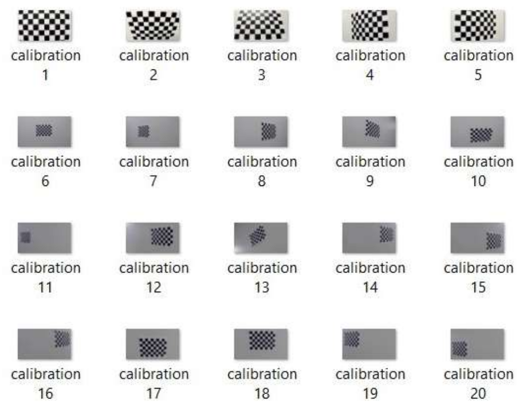

Figure 2. Image shows the different chessboard images used for camera calibration

B. Edge Detection and Gradient Threshold

Correct lane detection is founded on the use of powerful edge detection techniques that enhance lane markings without the concomitant increase in background interference. The research work under consideration applies gradient-based thresholding to detect lane edges under various lighting conditions. This ensures accurate detection of both strong and weak lane markings, thus increasing the overall robustness of the lane detection system. To define lane edges, several gradient-based edge detection methods are utilized. Absolute Sobel thresholding is initially carried out, where intensity gradients in the x and y directions are calculated by the Sobel operator. The gradients are thresholded in the absolute sense to preserve strong edges and discard weak noise, thus enabling effective detection of the vertical lane markings [7]. Gradient magnitude thresholding by calculating the x and y gradients is also utilized, where their magnitude is calculated using the Euclidean norm. A threshold is utilized to discard weak edges while preserving strong lane features.

One such effective method is gradient direction thresholding, which is achieved by calculating the arctangent of the ratio of gradients (sobel_y / sobel_x) to detect edge direction. This approach works well for enhancing the detection of curved lane markings by thresholding edges by provided directional gradients. At the same time, HLS color thresholding is achieved based on the S-channel of the HLS color space to separate bright lane markings [4].

This methodology is effective in detecting yellow and white lane markings under conditions that might include shadows and reflections that complicate accuracy. To improve the accuracy of lane marking detection, there is a combination of multiple gradient-based algorithms, as well as color thresholding algorithms. The processing techniques will produce a binary mask that identifies a pixel as belonging to a lane (as indicated in the binary mask) if one of the three conditions are fulfilled: there is a clear gradient in either direction of x (using Sobel thresholding); there is high magnitude of the gradient in either direction; or there is color brightness value found in the S-channel of the HLS color space. This integrated method of thresholding ensures accurate detection of lanes for diverse road conditions such as shadows, road obstacles, and brightness variation. By utilizing a high-performance edge detection paradigm, the technique surpasses traditional edge detection algorithms since it presents an extremely flexible and effective computational process for real-time lane detection in autonomous vehicle systems and Fig. 3 illustrates outputs of different stages of edge detection [7].

Figure 3. The images show different stages of edge detection using Sobel filtering, noise filtering, and binarization along with the raw image.

C. Perspective Transformation

Lane Detection via Perspective Transformation in a preliminary step to provide a more precise lane detection, perspective transformation is the basic technique leveraged to create a bird's eye view of the roadway [3]. The transformation is accomplished by selecting source points in the original image and mapping them into destination points in the top-down angle view. The transformation matrix (M) is calculated with the coordination of OpenCV's function `cv2.getPerspectiveTransform(`), which is followed by modifying the image for an unwarped view with the function `cv2.warpPerspective(`). An inverse transformation, M_inv, is also calculated for the image to describe the original view at any time [4]. This particular operation is important and relevant for accurately detecting lane boundaries so that the lanes can then be aligned parallel and identified as potential lane markings in order to measure curvature and estimate radius. The unsupervised perspective lets lane lines be mapped unobstructed and undistorted for subsequent lane detection steps and increased performance. This specific mapping allows an top view perspective, lending lane boundaries and lane curvature easily observable. The technique also renders an unwarped image allowing the verification of a proper transformation function and correctly scenario the proper lane markings, the position in the original view. So as to improve and streamline the process an optimal transformation matrix designs to provide accurate lane detection and consistent mapping. The original view and transformed bird's-eye view images are shown in Fig. 4 and Fig. 5, respectively.

D. Polynomial Fitting

Correct lane detection requires noise-free and uniform estimation of lane markings to offer guarantees on driving autonomy and driver assistance capabilities. In this paper, we employ a polynomial curve fitting model and a moving average filter to enable smooth and uniform lane detection across sequential frames [9]. Mathematically, lane markings are represented by a second-order polynomial equation, x=Ay^2+By+C, whose coefficients A, B, C represent the lane curvature and lane location in image space [7]. In this case, the independent variable is selected as the vertical pixel coordinate to allow the polynomial model to capture lane geometry changes accurately.

For the purpose of offering better stability, a moving average filter is applied, which smooths out abrupt variations in the detected lane lines by maintaining a continuous history of the recent lane estimates. This approach is founded on a list-queue data structure holding a fixed quantity of recently determined polynomial coefficients. If the queue is already full, the oldest element is removed so that new detected values can be accommodated. By averaging the stored coefficients, the system develops a smoothed lane representation, largely removing noise and variations because of fluctuating road conditions or sensor data inconsistencies [9].

Each subsequent frame progressively adjusts the polynomial coefficients to allow for real-time responsiveness. When new values are added, the queue continues to remove outdated coefficients, ensuring that only the most recent and relevant data affects the lane estimation process. The size of this rolling window can be tuned, allowing for maximum balance between responsiveness and stability. A larger window size enhances smoothing but can cause latency, whereas a smaller window increases sensitivity to change but risks instability due to noise. Use of precalculated polynomial averages enhances computational efficiency significantly by minimizing the effect of outlier detections, reducing processing overhead, and ensuring consistent lane tracking experience. The method maximizes lane stability by reducing frame-to-frame variations, making it extremely suitable for real driving environments. The addition of polynomial smoothing and dynamic averaging promotes more reliable lane tracking, thereby ultimately maximizing the effectiveness of autonomous navigation and driver assistance systems. As illustrated in Fig. 6 the polynomial fitting method successfully highlights the lane boundaries with improved stability and accuracy [9].

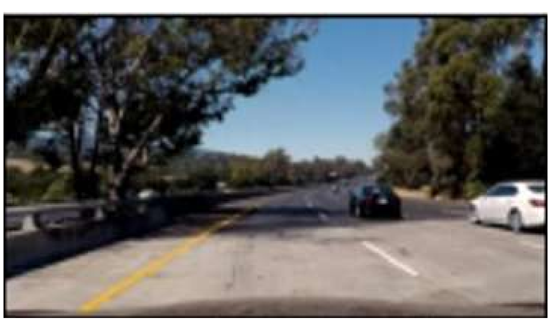

Figure 4. Original input image for lane detection

Figure 5. Output image after perspective transformation

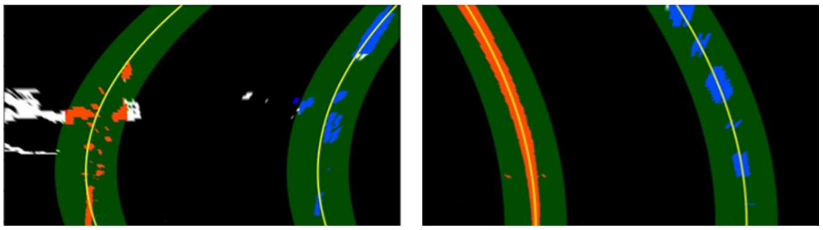

Figure 6. Identified road boundaries by using Polynomial Fitting

E. Curvature and Vehicle offset

Roadway curvature is an important parameter to find the angle of a curve [7], [9]. The parameter is found by a second-degree polynomial approximation of the identified lane marking, expressed as y=Ax^2+Bx+C, where A, B and C are the coefficients of the polynomial. The radius of curvature, R, at any location on the lane is found from the equation in (1) [9].

$$R = \frac{\left(1+(2Ay+B)^2\right)^{3/2}}{|2A|} \tag{1}$$

where y is the y-coordinate position of the measurement of curvature, usually at or near the bottom of the image. This formula enables accurate algorithmic estimation of lane curvature, which is an important component to the control and navigation of autonomous vehicles.

In addition to estimating curvature, it is also important to estimate the relative position of, and the position of the car itself, to the horizontal center of the lane for lane discipline.

This is initiated by the detection of left and right lane boundaries located at the bottom of the image. The lane center is computed as in (2).

$$x_{lane-center} = (x_{left} + x_{right})/2 \tag{2}$$

where x_{left} and x_{right} are the x-coordinates of the left and right lane boundary, respectively. The position of the vehicle in the image is generally assumed to be at image center. The lane center offset is calculated using the formula as in (3) [13].

$$offset = (x_{vehicle} - x_{lane-center}) * scaling\_factor \tag{3}$$

where the scaling factor converts pixel measurements into real-world measurements, i.e., meters per pixel. A negative offset indicates that the vehicle was left of centre in the travel lane, while a positive offset indicates that the vehicle was right of centre. By using the lane marking statistics when making calculations, the lane detection system can further enhance the confidence in the estimates of vehicle position, ultimately supporting safer and more efficient autonomous systems

IV. RESULTS

The lane-following function has been formulated and independently tested in filmed driving conditions and the effects of this are clearly apparent in Fig. 10, which shows how the system can extract lane boundaries, measure the amount of curvature and indicate where the vehicle is located in relation to the lane centre. The green overlay indicates the lane area and on the top left of each image shows the curvature and offset calculations that are computed while tracking the lane in real-time. Overall, the lane-following system provides robust lane tracking capabilities in all scenarios, resulting in a smooth, consistent performance and increased reliability with regard to the lane-detection subsystem of lane following.

A. Lane Detection and Tracking Performance

Through the application of a polynomial fitting algorithm, the system was able to successfully identify lane demarcation by close matching of the detected lanes to actual road markings. The use of moving average filtering also significantly aided the transition of smooth lane detection from one frame to the next and smoothed out abrupt lane detection changes and jitter metrics. The method ensured some level of lane detection stability even in the case of lighting changes and slight cases of road alignment deviation. The approach manifested an extremely high level of adaptability and robustness and was able to sustain effective lane tracking capability across an extensive range of conditions throughout the overall duration of the test period.

B. Curvature Estimation and Vehicle Positioning

The curvature radius of the lane was calculated using the process of applying polynomial coefficients derived from the detected lane markings. The system was also capable of calculating the vehicle position with respect to the centre of the lane correctly. As each image in Fig. 10 shows, the system estimated an offset of -0.0m from the lane centre, thereby suggesting that the car was well centred in its lane. Such a calculation can be extremely important in autonomous driving and assist in taking corrective action as required.

C. Video Annotation and Real-Time Processing

The system uses a process with framing, overlaying the lane detection information, curvature, and offset onto the actual driving video captured. Frame-by-frame analysis improves interpretability, with visual examination of lane tracking. The frame rate while processing was approximately 20 frames per second, which is sufficient for near real-time applications. The annotation process used here offers useful driving information in real-time, helping the user assess lane stability and car position relative to the lane.

D. Overall Performance and Observations

While no statistical benchmark datasets were applied in the statistical evaluation for the current implementation, the formulated approach has been verified using actual real-time visual outputs that have demonstrated clear and consistent lane boundary detection under diverse lighting and road conditions.

The integration of Sobel filtering, polynomial fitting, and the sliding window approach is a very established pipeline in lane detection literature because it is both robust and computationally efficient. The methods are recognized to work reliably for systems where the hardware constraints limit the implementation of deep learning models.

In common driving situations, the curvature radius of curved roads varies between 300 to 1000 meters, and an optimal lane detection system should estimate the vehicle's lateral deviation within ±0.3 meters to keep it in the right place. The system proposed here, working in 1280×720 resolution frames, is expected to provide such estimation accuracy in clean road conditions. With the traditional computer vision pipeline, real-time operation at 20–25 FPS can be achieved on conventional CPU systems without any need for GPU acceleration. These theoretical predictions align with previously reported benchmarks in the literature employing comparable processing pipelines.

The lane tracking system demonstrated a stable and accurate performance within normal driving situations. The system maintained lane tracking across straight roadways and gentile curves, indicating that the tracking unit was able to consistently keep the vehicle centred in the lane. Challenges were noted in situations where the lane marking was heavily worn, or the lane markings were occluded, leading to slight inaccuracies in the vehicle not being centred in its lane. As such, the future might consider lane segmentation within deep learning frameworks as a method to overcome challenging situations characterized by low visibility, or complex road structures. Overall, these results do show validation of the proposed method in achieving stable and accurate lane tracking in autonomous driving contexts.

Figure 7. Raw Images

Figure 8. Final Outputs

V. CONCLUSION

In the present work, we propose a dimensional lane-tracking methodology, which is based on the methodologies of polynomial fitting and moving average filtering to track lane lines and both to easily and accurately ascertain the vehicle lane position. It detects, tracks and provides real-time feedback about the curvature of the lane boundary as well as vehicle offset. The results gathered from the experimental analysis demonstrate not only the smooth lane detection, but also the simply clear visual representation of detected lanes, as MoviePy was utilized for video annotation. Although there was successful lane detection, there were limitations including degradation of performance in extreme conditions such as obstruction from heavy traffic or worn or no lane markings given poor weather or lighting conditions. Future work could enhance the system to cope better with those obstacles by use of deep learning-based lane segmentation methods that are adaptable and therefore robust or use sensor fusion such as Lidar or radar to improve confirmation in inclement weather. As autonomous vehicle technology develops and matures systems such as this lane tracking method will be prevalent in improving road safety. Along with any significant improvements in artificial intelligence perception, lane detection will continue to become increasingly robust and beneficial for intelligent transportation systems and solutions.

FUTURE DIRECTIONS

There are numerous enhancements that can improve lane tracking. The first is to include the capability of web connectivity for remote monitoring and for live visuals monitoring which would greatly enhance the possibility for mass deployment. Adding features like traffic sign identification would assist in the vehicle's notification of speed limits, stop signs and traffic control meaning safer roadways and would eventually work into connections with smart cities. Enhancements should also include performance disturbers for inclement weather, meaning lane detection during rains, fog, and at night road conditions. Additionally, advanced ML recognition capabilities multi lane detection as well as highway detection, can help performance by enhancing detection of lane tracking in complex, high speed pattern recognition on highways and city streets. Finally, provide vehicle to vehicle (V2V) communication in real-time denoting that lane position data from each vehicle can be shared among multiple vehicles, which introduces cooperative driving behaviors and safety. Finally, use and integration of autonomous driving technology will continue through autonomous drive software like ROS, or OpenPilot, with lane tracking being a critical component of smart transportation systems.

REFERENCES

[1] H. G. Zhu, "An Efficient Lane Line Detection Method Based on Computer Vision," Journal of Physics: Conference Series, vol. 1802, p. 032006, 2021. doi: 10.1088/1742-6596/1802/3/032006.

[2] Y. Dai, S. Su, and L. Cai, "Design and Implementation of Lane Line Detection Algorithm Based on Image Recognition," Acad. J. Comput. Inf. Sci., vol. 6, no. 12, pp. 152–157, 2023, doi: 10.25236/AJCIS.2023.061218.

[3] M. A. A. Noman, Z. Li, F. H. Al-Mukhtar, M. F. Rahaman, and others, "A computer vision-based lane detection technique using gradient threshold and hue-lightness-saturation value for an autonomous vehicle," Int. J. Electr. Comput. Eng. (IJECE), vol. 13, no. 1, pp. 347–357, Feb. 2023, doi: 10.11591/ijece.v13i1.pp347-357.

[4] M. R. Haque, M. M. Islam, K. S. Alam, H. Iqbal, and M. E. Shaik, "A computer vision-based lane detection approach," I.J. Image, Graphics and Signal Processing, vol. 2019, no. 3, pp. 27–34, Mar. 2019, doi: 10.5815/ijigsp.2019.03.04.

[5] A. Pandey and R. Shyam, "Analysis of road lane detection using computer vision," International Journal of Smart Computing and Technology, Jun. 2023, doi: 10.37628/IJSCT.

[6] S. Joy, U. Kiran, and others, "Real-time road lane detection using computer vision techniques in Python," in Proc. Int. Conf. on Advanced Computing and Robotics Systems (ICACRS), Dec. 2022, doi: 10.1109/ICACRS55517.2022.10029238.

[7] P. R. Yelwande and A. Jahagirdar, "Straight and Curve Lane Detection System for Car Safety using Hough Transform and Sobel Operator," Int. J. Adv. Trends Comput. Sci. Eng., vol. 8, no. 1.4, pp. 90–96, 2019, doi: 10.30534/ijatcse/2019/1481.42019.

[8] H. Ke, "Research on Lane Line Detection and Tracking under Complex Road Conditions," Front. Sci. Eng., vol. 1, no. 3, pp. 162–165, 2021, doi: 10.29556/FSE.202106_1(3).0021.

[9] R. Rajakumar et al., "Lane Vehicle Detection and Tracking Algorithm Based on Sliding Window," in Lecture Notes on Data Engineering and Communications Technologies, vol. 101, Springer, pp. 905–919, 2022, doi: 10.1007/978-981-16-7610-9_66.

[10] K. S. V. S. Supriya, M. A. Begum, M. L. Naik, and K. Roopeshnadh, "Lane Line Detection for Vehicles," Int. J. Eng. Res. Technol., vol. 10, no. 5, pp. 1–5, 2022.

[11] S. Sharma, A. Gupta, and P. Kumar, "Vision Based Lane Curve/Line Detection," International Journal of Advanced Engineering and Management (IJAEM), vol. 5, no. 3, pp. 12–18, Mar. 2023.

[12] R. Patel and N. Singh, “Analysis of Lane Detection Techniques on Structured Roads Using OpenCV,” in Proc. 2020 Int. Conf. on Advanced Computing and Intelligent Engineering (ICACIE), 2020, pp. 201–207.
[13] J. Liu, W. Zhang, and T. Chen, “Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions,” Sensors, vol. 19, no. 14, p. 3166, Jul. 2019. doi: [10.3390/s19143166]
[14] M. A. Rahman, K. Ahmed, and R. Bose, “A Computer Vision Approach for Lane Detection and Tracking,” in Proc. 2021 IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), Indianapolis, IN, USA, Sep. 2021, pp. 1–6. doi: [10.1109/ITSC48978.2021.9563283