Grenze International Journal of Engineering and Technology, June Issue

A Review on Lane Line Detection and Forward Collision Warning using Artificial Intelligence

Dr. M. Kavitha[1], Dr. R. Menaka[2], Nishanthini T V[3], Mohan Kumar P[4], Swathi M[5] and Vaishali T J[6]
[1]ASP, Velalar College of Engineering and Technology/Information Technology, Erode, Tamil Nadu
Email: misskavi84@gmail.com
[2]AP, Velalar College of Engineering and Technology/Information Technology, Erode, Tamil Nadu
Email: menaka.murugesan@gmail.com
[3-6]Velalar College of Engineering and Technology/Information Technology, Erode, Tamil Nadu
Email: nishanthinitv10@gmail.com, mohankumarmdk007@gmail.com, swathimadeshwaran005@gmail.com, itbvaishalitj@gmail.com

***Abstract*— The article introduces the Autonomous Self-Driving Assistance System (ASDAS), a groundbreaking technology poised to transform vehicle safety by seamlessly integrating Lane Line Detection and Forward Collision Warning. Within the Lane Line Detection module, advanced computer vision algorithms combine image processing techniques and machine learning models to swiftly and accurately identify lane boundaries in real-time, enabling autonomous vehicles to navigate within predefined lanes for secure and predictable trajectories, crucial for intricate traffic scenarios. Simultaneously, the Forward Collision Warning module utilizes sensor data such as radar and lidar inputs to monitor the vehicle's surroundings, assessing collision risks with sophisticated algorithms that predict and analyze relative speeds and distances of nearby objects. In response to potential collision threats, the system issues timely warnings to the vehicle's onboard control system, prompting immediate corrective actions. This innovative integration of technologies represents a significant step forward in enhancing vehicle safety and efficiency in the evolving landscape of autonomous transportation.**

***Index Terms*— Lane Line detection, Forward collision warning, Autonomous driving assistance system, and Image/video processing.**

I. INTRODUCTION

Lane detection involves identifying white road lines, while lane tracking assists vehicles in adhering to their intended path by using previously detected lane markers to control motion. Despite their critical role in autonomous driving, comprehensive studies in this domain are lacking in the literature. This research aims to fill this gap by conducting a qualitative comparative analysis of existing lane detection and tracking algorithms, providing insights into knowledge disparities.

The review examines prior studies, offering insights into the algorithms used for lane detection and tracking. It emphasizes a comparative analysis to identify strengths, weaknesses, and gaps in current knowledge. Additionally, the study provides an overview of prominent datasets used for algorithm testing and the evaluation metrics applied. Existing research predominantly focuses on straight roads, neglecting intricate geometries like clothoid roads.

Grenze ID: 01.GIJET.10.2.252

Challenges such as variable weather, camera quality, indistinct lane markings, and unpaved roads complicate lane detection and tracking. Additional challenges include overtaking vehicles causing occlusion, high-speed scenarios, and intense illumination.

Most studies use custom datasets, indicating a need for standardized datasets as autonomous vehicle development progresses. As technology advances, future research is expected to produce more reliable algorithms rigorously tested against real-time datasets. In advanced driver assistance systems, radar and lidar excel in long-range obstacle detection but face challenges in object recognition and cost. Visual sensors, especially cameras, are popular for identifying lanes and drivable areas through road surface marking detection.

The convergence of sensor technologies and algorithm refinement is expected to lead to more effective solutions in lane detection and tracking as autonomous driving capabilities evolve.

II. LANE LINE DETECTION

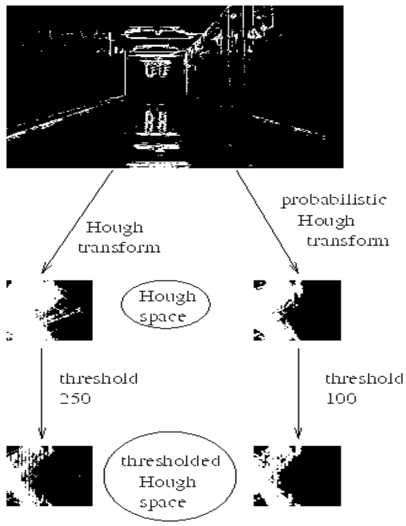
A. HOUGH LINE TRANSFORM ALGORITHM

The Hough Transform stands out as a widely adopted technique for shape detection, proving particularly effective when the shape can be mathematically represented. This method is versatile and capable of detecting shapes even when they are slightly broken or distorted. The focus here is on its application to line detection.

A line, in this context, can be expressed as $y=mx+c$ or in parametric form as $\rho=x\cos\theta+y\sin\theta$. In this representation, denote the line's perpendicular distance from the origin as ρ, and measure the angle formed by this perpendicular line and the horizontal axis in a counter-clockwise direction. Notably, the direction may vary based on the chosen coordinate system, with the representation used in OpenCV adopting a specific convention.

In the parameter space of (ρ, θ), the Hough Transform employs a 2D accumulator array initialized to zero. Rows correspond to ρ, and columns correspond to θ. The array size is contingent upon the desired accuracy, with the example of 1-degree accuracy resulting in 180 columns for angles. Assuming one-pixel accuracy for ρ, we determine the number of rows based on the diagonal length of the image.

The probabilistic Hough Transform improves the regular Hough Transform by being more efficient; it focuses on randomly selecting a subset of points for line detection. This judicious selection of points proves sufficient for the task, and by adjusting the threshold, the algorithm becomes adept at identifying lines while minimizing computational complexity. The comparison between the Hough Transform and the Probabilistic Hough Transform in Hough space is visually depicted in the accompanying image below, illustrating the efficacy of the probabilistic approach in streamlining the line detection process.



B. LANE DETECTION BASED ON THE TUSIMPLE DATASET

The Tusimple dataset comprises a total of 3626 training images and 2782 testing images. Each sequence image consists of 20 consecutive frames captured within a 1-second interval. Notably, the initial 19 frames are devoid of lane markings, while the 20th frame is annotated with the ground truth for lanes. The dataset exhibits diversity across four main conditions: various weather conditions, daytime scenarios, different lane numbers (ranging from

2 lanes to 4 lanes or more), and varying traffic environments [47,48]. In this study, a set of 150 images, each capturing diverse conditions, was randomly selected from the Tusimple dataset. Subsequently, lane detection experiments were conducted using the algorithm proposed in this research.

C. DATASET PREPARATION

Instead of using a pre-trained model, the lane detection model is trained using a custom dataset. For this purpose, images of different lane conditions are captured. A Huawei Nova 5 handphone is used as the recording device. The device is placed in front of the vehicle to avoid unnecessary jitters in the video quality. The recording starts when the car starts to move. In total, 25 videos have been collected: 13 morning videos, 4 evening videos, and eight rainy videos that vary in length, driving conditions, and environments. In this way, we can ensure that the trained model can detect lanes in different driving conditions and situations. After that, the videos are cropped to 1980x1080 pixels. MakeSense AI is used to turn each frame into its own bounding boxes and polygon annotations, and the annotations and bounding boxes are exported into a COCO.json file.

LANE DETECTION EVALUATION

Various models have undergone scrutiny for lane detection, with evaluations conducted on conventional YOLO, YOLACT, and Mask RCNN. The performance of distinct lane detection approaches is presented, revealing insights into their effectiveness. The extended processing time of Mask RCNN can be attributed to the prolonged duration taken by the detection mask, coupled with a notable occurrence of false detections, leading to redundant objects being identified.

This significantly hampers the processing speed. In contrast, YOLACT exhibits a faster frames-per-second (fps), outpacing MASK RCNN. YOLACT demonstrates enhanced accuracy in object detection, reducing false positives compared to MASK RCNN. YOLOv5 emerges as the fastest due to its focus on detecting object bounding boxes without the time-consuming task of outlining object masks. However, the quality of the generated mask is comparatively inferior. Notably, YOLACT emerges as the optimal choice, delivering superior performance in both speed and accuracy during testing. Consequently, the YOLACT model is selected for subsequent evaluations. The mean average precision serves as a technique for computing the conditional probability of observing data given a model, factored in with an initial probability or belief about the model.

On the other hand, the intersection over union (IOU) measures the degree of overlap between the expected and actual bounding boxes. Frames per second (fps) provide a metric for determining the number of frames the system can generate per second.

D. ALGORITHMS PERFORMANCE COMPARISON

In assessing the lane detection algorithm's efficacy, a comparative analysis was conducted, juxtaposing the proposed algorithm with other methodologies prevalent in the existing literature. The table presents a comprehensive overview, delineating the statistical comparison of algorithm performance.

Method	Algorithm	Average detection accuracy (%)	Average processing time (ms)/Frame
Traditional	Spatial Ray Features	94.40	45.0
	Hough Transform	95.70	65.04
Deep Learning	Fast draw Resnet	95.00	65.3
	convLSTM	97.25	42.0

III. FORWARD COLLISION WARNING

Our forward collision warning system utilizes gradient images and lane marking detection results as input, with cast shadows proving crucial in identifying front-facing vehicles, particularly in daylight conditions. We propose a fusion of an adaptable Region of Interest (ROI) and a consistent trapezoidal shape for effective vehicle detection. The adaptable ROI dynamically adjusts based on the width of the vertical edge pair of the vehicle, employing erosion and dilation with 3x3 and 11x11 convolution masks, respectively, for morphological operations. Vertical edge pair identification is facilitated through the histogram of vertical projection, employing multiple intervals with distinct widths to mitigate noise impact.

A. MACHINE LEARNING APPROACH

Dimililer et al. (2020) proposed a machine-learning strategy for predicting vehicles in both images and videos. The outlined approach involves the following key steps in developing a software pipeline for vehicle detection in a video:

1. Conduct a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and subsequently train a Linear Support Vector Machine (SVM) classifier (Dimililer et al., 2020).
2. Implement a sliding-window technique and leverage the trained classifier to search for vehicles within images (Dimililer et al., 2020).
3. Apply the developed pipeline to a video stream, generating a heat map of recurring detections frame by frame. This process aids in the rejection of outliers and facilitates the tracking of detected vehicles (Dimililer et al., 2020).
4. Estimate bounding boxes for the identified vehicles, providing a visual representation of the detected objects (Dimililer et al., 2020).

In this undertaking, we are looking over the methodologies outlined in the research by Dimililer et al. (2020) for vehicle detection and tracking. This is done using OpenCV and Python, which encompasses both images and videos, offering an automated system capable of detecting and tracking vehicles seamlessly. The dataset utilized comprises 8792 vehicle images and 8968 non-vehicle images, sourced from the GTI vehicle image database and the KITTI vision benchmark suite. Our approach involves the application of Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set, and we trained a Support Vector Machines (SVMs) classifier.
The implementation includes a sliding-window technique, leveraging the trained classifier to systematically search for vehicles in images. Upon obtaining the object detection results, we incorporated an object tracker. This tracker continually follows the target by re-detecting it in sequential frames following the initial detection. The presence of identified vehicles is visually represented by enclosing them within bounding boxes.

B. CLASSICAL ALGORITHM: MAZDA ALGORITHM

The algorithm employs a kinematics analysis-based approach to ascertain the critical braking distance. However, the computed distance proves to be excessively conservative for braking purposes and is instead utilized as a critical warning distance. This indicates the presence of a crucial alarm distance.

$$D_w = \frac{1}{2}\frac{v^2}{a_1} - {v - v_0}/{a_2} + vt_1 + v_0 t_2 + d_0 \qquad (1)$$

Here, d0 represents the head offset distance, denotes the speed of the rear car, indicates the relative speed between the vehicles, t1 accounts for the system delay, t2 reflects the driver reaction time, a1 is the maximum brake deceleration of the rear car, and a2 represents the maximum brake deceleration of the front car.
Symbols in your equation should be defined before the equation appears or immediately following. Use “(1),” not “Eq. (1)” or “equation (1),” except at the beginning of a sentence: “Equation (1) is ...”

C. Radar

Implementing RADAR (Radio Detection and Ranging) in cars for forward collision warning involves the integration of radar sensors into the vehicle's safety and driver assistance systems. RADAR is commonly used in automotive applications to detect objects in the vehicle's path and provide warnings or take preventive measures in the case of an imminent collision. Here are the key steps in the implementation:
1. *Sensor Selection*: Choose a RADAR sensor suitable for automotive applications. Automotive RADAR systems often operate in the 24 GHz or 77-81 GHz frequency bands. Select the appropriate sensor type, such as continuous wave (CW) radar or frequency-modulated continuous wave (FMCW) radar, based on performance requirements.
2. *Sensor Placement*: Strategically place the RADAR sensor on the vehicle. Common locations include the front grille, behind the front bumper, or integrated into the front fascia.
3. *Sensor Calibration*: Calibrate the RADAR sensor to ensure accurate measurements and alignment. Calibration is crucial for precise distance and speed estimation.
4. *Signal Processing*: Develop signal processing algorithms to analyze the RADAR returns. This involves processing the received signals to extract information about the range and speed of detected objects.
5. *Object Detection and Tracking*: Implement algorithms for detecting and tracking objects based on RADAR data. This includes identifying targets, estimating their range and speed, and associating measurements across multiple RADAR scans.
6. *Data Fusion*: Integrate RADAR data with information from other sensors, such as cameras and lidar, using sensor fusion techniques. Combining data from multiple sensors enhances the overall perception and reliability of the system.
7. *Filtering and Smoothing*: Apply filtering and smoothing techniques to the RADAR measurements to reduce noise and improve the accuracy of object tracking. Kalman filters or more advanced filters are commonly used for this purpose.

8. *Collision Risk Assessment*: Implement algorithms to assess collision risks based on the detected objects. Determine whether the vehicle is on a collision course with an object and evaluate the severity of the potential collision.
9. *Forward Collision Warning*: Provide warnings to the driver when a potential forward collision is detected. Warnings can be visual, auditory, or haptic, depending on the vehicle's human-machine interface (HMI) design.
10. *Collision Mitigation*: In advanced systems, integrate collision mitigation measures such as pre-charging brakes, adaptive cruise control adjustments, or even automatic emergency braking to actively prevent or mitigate collisions.
11. *Testing and Validation*: Conduct thorough testing and validation of the RADAR-based forward collision warning system in various driving conditions, including scenarios with different object sizes, speeds, and environmental factors.
12. *Compliance with Standards*: Ensure that the RADAR system and the forward collision warning system comply with relevant safety standards and regulations for automotive applications.General Formula for Radar Range Measurement:
The basic formula for calculating the range (distance) of an object using radar is:

$$R = c\,\tau/2$$

where:
R is the range of the object,
c is the speed of light (approximately 3 X 10 8 meters per second),
τ is the time delay of the radar signal.
This formula assumes a one-way travel of the radar signal. The actual implementation may involve additional considerations, such as signal processing and calibration factors.

D. Radar

Implementing LIDAR in cars for forward collision warning involves integrating LIDAR sensors into the vehicle's safety system, coupled with advanced object detection methodologies. LIDAR provides high-resolution 3D point cloud data, which is valuable for detecting and tracking objects in the vehicle's path. Here's a step-by-step guide for implementing LIDAR for forward collision warning with an emphasis on object detection methodology:
1. *Sensor Selection:* Choose a LIDAR sensor suitable for forward collision warning applications. Consider factors such as range, field of view, resolution, and update rate. Solid-state LIDARs or mechanical LIDARs can be used based on the application requirements.
2. *Sensor Placement:* Strategically place the LIDAR sensor on the vehicle, typically on the front grille or integrated into the front bumper. Ensure the placement provides a clear and comprehensive view of the road ahead.
3. *Sensor Calibration*: Calibrate the LIDAR sensor to ensure accurate measurements. Calibration involves aligning the LIDAR sensor with other sensors on the vehicle and compensating for any offsets, distortions, or misalignments.
4. *Data Processing*: Develop algorithms for processing the raw LIDAR data. Convert the point cloud data into a format suitable for object detection and tracking. Consider downsampling or filtering techniques to manage data density.
5. *Object Detection*: Implement object detection algorithms to identify obstacles, vehicles, pedestrians, and other relevant objects in the LIDAR point cloud. Common techniques include clustering, segmentation, and machine learning-based approaches.
6. *Object Tracking*: Implement object tracking algorithms to monitor the movement of detected objects over time. Kalman filters or more advanced tracking methods can be used to predict object trajectories.
7. *Collision Risk Assessment*: Develop algorithms to assess collision risks based on the detected and tracked objects. Consider factors such as relative speed, distance, and the trajectory of the objects.
8. *Forward Collision Warning System Integration*: Integrate the LIDAR-based object detection and collision risk assessment into the vehicle's forward collision warning system. Ensure seamless communication with other safety systems.
9. *Driver Alert System*: Implement a driver alert system that provides timely warnings to the driver in case of an imminent collision. The warnings can be visual, auditory, or haptic, depending on the vehicle's human-machine interface design.
10. *Adaptive Cruise Control*: Integrate LIDAR data into adaptive cruise control systems to adjust the vehicle's speed based on the detected objects and the desired following distance.

11. *Testing and Validation*: Conduct extensive testing and validation of the LIDAR-based forward collision warning system under various driving conditions, including scenarios with different object sizes, speeds, and environmental factors.
12. *Redundancy*: Consider redundancy by integrating multiple LIDAR sensors or combining LIDAR with other sensor modalities to enhance system reliability and safety.
13. *Compliance with Standards*: Ensure that the LIDAR-based forward collision warning system complies with relevant safety standards and regulations for automotive applications.
14. *Ongoing Maintenance and Updates*: Establish procedures for ongoing maintenance and software updates to ensure the continued performance and reliability of the LIDAR system.
15. *Power Management*: Implement power management strategies to optimize energy usage, especially in electric and hybrid vehicles.
By following these steps, you can implement a LIDAR-based forward collision warning system with robust object detection methodologies, contributing to improved safety on the roads

III. CONCLUSIONS

The synthesis of literature and research findings has shed light on the significant strides made in these critical areas, reflecting the evolution from traditional computer vision techniques to the incorporation of advanced machine learning and deep learning methodologies. For lane line detection, the review highlighted the diverse range of approaches, including color-based thresholding, edge detection, Hough transform, and the integration of deep learning models. In the realm of forward collision warning, the synthesis of literature elucidated the pivotal role of sensor technologies such as radar and LIDAR, coupled with sophisticated algorithms for object detection and collision risk assessment. The integration of sensor fusion techniques, machine learning algorithms, and real-time processing has elevated the precision and reliability of forward collision warning systems. The ongoing fusion of artificial intelligence, sensor technologies, and human-centric design principles holds the promise of creating vehicles that are not only aware of their surroundings but also capable of proactively mitigating collision risks.

REFERENCES

[1] Ni, Jianjun, Yinan Chen, Yan Chen, Jinxiu Zhu, Deena Ali, and Weidong Cao. 2020. "A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods" Applied Sciences 10, no. 8: 2749.
[2] Jaswanth Nidamanuri; Chinmayi Nibhanupudi; Rolf Assfalg; Hrishikesh Venkataraman. A Progressive Review: “Emerging Technologies for ADAS-Driven Solutions”. IEEE Transactions on Intelligent Vehicles (Volume: 7, Issue: 2, June 2022)
[3] Huei-Yung Lin, Jyun-Min Dai,Lu-Ting Wu and Li-Qi Chen. “A Vision-Based Driver Assistance System with Forward Collision and Overtaking Detection”. National Library of Medicine, National Center for Biotechnology Information. Sensors (Basel). 2020 Sep; 20(18): 5139.
[4] Jun Jie Wee, Connie Tee, and Michael Kah Ong Goh, “Forward Collision Warning for Autonomous Driving”. Journal of Logistics, Informatics and Service Science Vol. 9 (2022) No.3, pp.208-225.
[5] Jingwei Cao, Chuanxue Song, Shixin Song, Feng Xiao, and Silun Peng. “ Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments”. National Library of Medicine, National Center for Biotechnology Information. Sensors (Basel). 2019 Jul; 19(14): 3166.
[6] M.N Tondra, Ebrahim Karami, “Vehicle detection and tracking”. Research Gate publication/ 353300177 July 2021.
[7] Y. J. Zhang,1F. Du, J. Wang, L. S. Ke, M. Wang,1Y. Hu,1M. Yu, G. H. Li, and A. Y. Zhan, ” A Safety Collision Avoidance Algorithm Based on Comprehensive Characteristics.” Research Article Volume 2020 | Article ID 1616420.
[8] D. O Cualain, M. Glavin, E. Jones, P. Denny, “Distance Detection Systems for the Automotive Environment: A Review” ISSC 2007, Derry, Sept 13-14.
[9] Z. Sun, G.. On-road vehicle detection IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(5):694–711, 2006.
[10] Muhammad Sohail, Abd Ullah Khan, Moid Sandhu, Ijaz Ali Shoukat, Mohsin Jafri & Hyundong Shin, “ Radar sensor-based machine learning approach for precise vehicle position estimation” https://www.nature.com/articles/s44159-023-40961-5.