Two Stage Lane Localization Algorithm for Enhanced Lane Modelling

Albert Author[1] and Bernard D. Researcher[2]

Abstract— **Lane marker identification is crucial for developing Intelligent Transportation Systems and Autonomous Vehicles. Although several image-processing-based algorithms are being used to pre-process the image at different steps of information extraction, like noise removal, region of interest selection etc., there is no efficient algorithm for enhancing the lane-modeling stage in particular. This paper proposes a two-stage pre-processing algorithm for implementing a divide and conquer approach to enhance the existing lane-modeling algorithms for efficient lane marker identification. The proposed methodology can generate the possible zones at which the lane markings can be present, thereby reducing the error margin in not detecting lanes to a greater extent. The current work uses Hough Transform (HT), Probabilistic Hough Transform (PHT) as a reference lane-modeling technique. A pre-processing image with curved lane marking is a test input to the HT, PHT algorithms and two-stage enhanced HT, PHT algorithms to extract lane markings. The proposed pre-processing algorithm for lane modeling has preserved more lane marker information in terms of curvature than the conventional HT and PHT.**

I. INTRODUCTION

With drastic development in the automobile industry, Intelligent Transportation Systems have become one of the fast-growing fields. The development of autonomous vehicles can drastically reduce road accidents, reduce traffic congestion and improve social and economic life in general[1]. The problem of road and lane perception is a crucial enabler for further development of autonomous vehicles [2]. The general lane marking identification algorithms involve three broad components, namely image data acquisition, pre-processing, and lane modeling [3].

A. Motivation & System Description

Image acquisition can be implemented using different modes, namely lidar, radar, optical camera, but due to the practical complications of lidar and radar-based systems, there are very few practical implementations [4].

The pre-processing step generally includes the region of interest selection followed by enhancement of the selected region and finally feature extraction from the image before finally passing to the lane modeling step.

There are many region of interest selection methodologies like Inverse Perspective Mapping geometry-based region extraction, which involves extracting a particular segment of image [5]. It is found out that the ambient noise due to different objects and moving vehicles is drastically reduced using the inverse perspective mapping method, which generates a bird's eye view of the lane images. Generating bird's eye view is found to drastically improve the functional and computational efficiency of later stages of the process [6].

Region enhancement involves pre-processing the valid region and enhancing the image quality through image filtering. Gaussian smoothening and sharpening filters are applied to smooth the color gradients of the image. These filters are particularly chosen depending upon the ambient lighting conditions to avoid manually induced gradient distortions [7].

The feature extraction is implemented to extract the gradient map of the images. The image's white and the yellow lane regions are extracted using color-based segmentation. Different color models such as HSI, HSV, YCbCr, YUV, along with the brightness of the image, are considered as factors, and range based filtering is performed to extract the lane markings, which are further processed by the threshold to remove any pseudo dark patches [8]. The lane modelling step follows the Feature Extraction. Hough Transform and Probabilistic Hough Transform are currently adapted as lane modelling techniques [9].

The existing works mainly cover the different ways to enhance the lane marking identification components, there is no particular pre enhancement step for the lane modelling stage in general. Although there are certain works which use divide and conquer approach through Simple Linear Iterative Clustering and to enhance the performance of lane modelling techniques like Hough Transform [10], [11], they were not computationally efficient which motivated us to current implementation.

B. Contribution of Current work

The current work involves an image processing-based lane localization to enhance and extract the lane marking zones. For input images, a reference data set is considered as an input source [12]. To reduce the complexity in calculations, firstly, inverse perspective mapping is applied to the input images. Color-based feature extraction is used to extract the lane line marking followed by image thresholding to avoid pseudo-dark patches. The feature extracted image is then passed to the currently developed two-stage algorithm to further enhance the feature extracted image before sending it to the lane-modeling stage for achieving effective and efficient lane modeling.

The two stage algorithms adopts a super-pixel based lane localization to implement dive and conquer approach for enhancing the lane modelling algorithms in general.

*This work was not supported by any organization

[1]Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands albert.author@papercept.net

[2]Bernard D. Researcheris with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA b.d.researcher@ieee.org

C. Organization of the paper

This paper is further organized into following sections. In Section II, the authors provide an overview on related works on the primary problem. In Section III the proposed work on lane marking localization is discussed. In Section IV, the authors present the different experimental results. Finally, in Section V the authors conclude the paper.

II. RELATED WORKS

This section discusses current relevant works. A substantial amount of work has been implemented in the disciplines of the region of interest selection, feature extraction, and super-pixels-based segmentation.

A. Input Data-Set

For the current references input images, The caltech lane-lines data collection is used as input pictures for the current investigation. The caltech Lanes data-set contains four films shot along streets in Pasadena, California, at various times of the day. [12].

B. Inverse Perspective Mapping

There have been numerous studies done on the topic of lane identification utilizing image processing techniques. The input image is separated into two separate categories: Images from the dashboard-mounted camera in front of the car and a bird's eye modified version of these images through inverse perspective mapping (IPM).

IPM is commonly used to improve computing efficiency and reduce environmental noise. IPM is a geometrical procedure that creates a bird's eye view of the lanes from the input picture as a perspective mapped image together with the perspective mapping matrix using the Perspective mapping function. [6], [9], [13].

C. Feature Extraction

The feature-based extraction is given the perspective mapped picture as an input. Color-based feature extraction is used here.The picture is initially transformed into either the HSV or LAB color spaces. Despite the fact that color parameters are mainly dependent on lightning circumstances, yellow and white lane markers inhabit distinct color ranges. To extract the lane markers, these color frames are filtered.The image morphological closure followed by image dilation is used on the feature extracted picture to avoid incomplete recognition and decrease mistakes [8], [10].

D. Super-Pixel based Segmentation

One of the most popular super-pixel segmentation techniques is the Simple Linear Iterative Clustering Algorithm or SLIC.Its effectiveness as a pre-processing algorithm may be attributed to its ease of use, and ability to create super-pixels that meet the requirements of good border adherence and low adjacency.In the five-dimensional CIELAB color and image space, slic clusters pixels using a confined k-means optimization in a multiple iterative way to segment the images [11], [9].

Overcoming the limitations of SLIC in terms of computational efficiency, SNIC is non-iterative, explicitly enforces connectivity and computationally cheaper while reaching desirable results. SNIC relies excessively on the priority queue to achieve non-iterative optimization. It employs a complex region expanding approach in which clustering centroids are developed via online averaging to produce superpixels. [14]

Although the SNIC improved the computational efficiency to a much greater extent than SLIC, the run-time exceeded 15 seconds per input frame, posing a significant limitation in adapting the super-pixel-based segmentation in lane marker identification. Since gray images are the primary source of information, the 5-dimensional vectors used in SNIC would be unnecessary. Below are the contributions of the present paper to enhance super-pixel-based techniques for lane modeling. 1) 3-D Distance Metric instead of 5D metric to increase computational efficiency. 2) Each pixel's seed allocation is limited to 8-Neighbors, unlike SNIC. 3) The static pre-processing component of SNIC is separated as a stationary phase to remove computational redundancy in SNIC for multiple image inputs.

III. PROPOSED METHODOLOGY

The proposal involves a two-stage approach to extract the lane marker information. The pre-processing stage for the current implementation involves two essential steps. 1. Inverse Perspective Mapping 2. Feature Extraction. The algorithm takes the pre-processed image as an input. There are two phases in the proposed algorithm, 1. Static Phase and 2. Dynamic Phase The overall block diagram of the proposed algorithm is as shown in Figure 1.

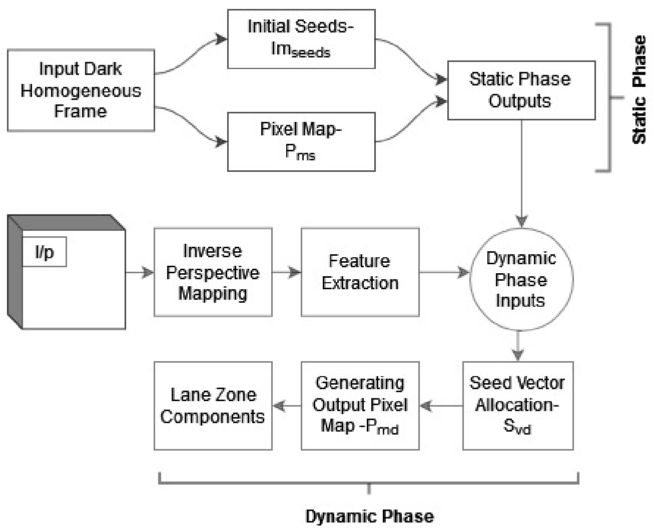


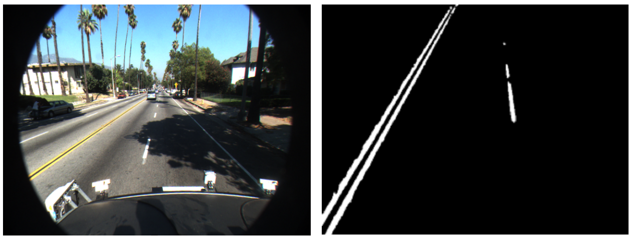
Fig. 1: Block Diagram

A. Pre-Processing of Input images

The input images from caltec lane lines data-set are passed to the inverse perspective block followed by the feature extraction block to generate the required input images for the algorithm.

By remapping each pixel of the resulting two-dimensional picture to a new place and constructing a new image on an inverted two-dimensional plane, IPM provides a birds-eye perspective view of the road.

The HSV colour space is utilised to segregate chromaticity and brightness in the feature extraction block. It aids in the reduction of colour saturation, which has been shown to be more accurate and effective in identifying and blocking shadows in the RGB space. As the Hue component is insensitive to shadows and water regions, it makes more sense to identify the road using HSV.



(a) Input for Static Phase (b) Input for Dynamic Phase

Fig. 2: Inputs for proposed algorithm

The process is designated to generate gray scale images. The pre-processed output for an example input is shown in Figure 2. Figure 2a is a color input image which is passed through the pre-processing component to generate the gray scale image as shown in Figure 2b

B. Static Phase

A static step is the first component of the proposed two-stage algorithm. Given the in-variant nature of the input images in terms of dimensions, the initial seed map and initial pixel label map will remain constant for every successive image frame.

The static reference picture Im_{static} is a black image that is homogeneous in appearance and has the same size as the successive input frames. Using the appropriate seed count in the x and y dimensions, the initial seeds Im_{seeds} are generated. The initial seeds Im_{seeds} are a collection of distinct seeds that may be obtained on a consistent grid over the reference picture Im_{static}, similar to snic [14].

The static algorithm 1 may be broken down into three primary parts. 1) Creating a basic pixel hash-map, 2) Calculating the Initial seed neighbor for each pixel using the distance measure, 3) Creating a eight-seed-neighbourhood map for each pixel.

The first step generates a pixel label map P_{ms} of pixel 'P' which has the pixel vector $P_v = \{P_x, P_y, P_i\}$ and seed label component$P_v[label] \leftarrow P_l$ for every valid pixel in the image Im_{static}, where P_x is the x-coordinate, P_y is the y-coordinate, P_i is the default pixel intensity which is initially set to -1 and P_l is the initial seed label for the current pixel.

The initial seed label for every pixel is generated using the euclidean distance measure as given by equation 1. Here a priority que q_p is generated and the euclidean distance E_{ds} between every seed and current pixel is pushed. (x_s, y_s) are the euclidean positions of each seed and (x_p, y_p) are euclidean positions of the pixel in the image Im_{static}. Once the iteration completes over the entire seed collection Im_{seeds} , the least distance value is popped out of the priority-que q_p and the corresponding label P_l is assigned to the pixel vector P_v.

$$E_{ds} = \sqrt{(x_s - x_p)^2 + (y_s - y_p)^2} \tag{1}$$

The second stage is creating a list of eight eligible neighbours $8n_p$ for each pixel 'P' in the picture Im_{static}. Here for every pixel, a list P_n is initialised with a zero value. Now for every eight neighbour of the current pixel, a validity check is conducted to make sure that the neighbors $\in Im_{static}$. The list P_n is then updated onto the corresponding pixel vectors P_v as $P_v[n] \leftarrow P_n$ in the pixel label map P_{ms}

The initial Pixel label map for the homogeneous frame similar to the IPM image of Figure 2b is as shown in Figure 3. Here $9X9$ seeds are considered in x and y dimensions. The pixels on the top left of the frame are labelled with lower seed label and the seed label keeps incremented by a unit value for each seed position. Here pixels of each square unit can be collectively regarded as a super pixel unit.

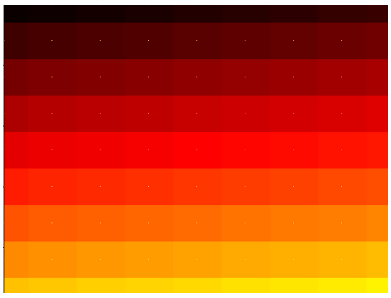


Fig. 3: Static Pixel Map

C. Dynamic Phase

The dynamic phase is the two-stage algorithm's run-time component. Before moving on to the lane modeling step, each feature extracted picture goes through the dynamic phase. The default pixel neighborhood map P_{ms} , the static image seeds Im_{seeds} that are created initially in the static phase, and the current feature extracted picture $Im_{dynamic}$ are the input components for this phase.

The dynamic algorithm is divided into three major components. 1) Generating the default dynamic pixel hash-map P_{md}, 2) Generating the initial seed vectors S_{vd} 3) Assigning pixels to their appropriate seeds.

The initial stage is to create a dynamic pixel map P_{md} from the static map P_{ms} and input picture $Im_{dynamic}$. The pixels in the input image are thresholded using a threshold lambda λ dependent on the lighting circumstances [15]. A three-dimensional vector $P[vector] \leftarrow \{P_x, P_y, P_i\}$ is now assigned to the dynamic pixel hash map for each accessible Pixel P in the thresholded picture $Im_{dynamic}$, the pixel intensity P_i is now set from the input image $Im_{dynamic}$ together with the initial label $P[label] \leftarrow P_l$ and the seed

Algorithm 1 Static Phase

Input: Im_{static}, Im_{seeds}
Output: P_{ms}
initialize:
  $P_{ms}[:] \leftarrow 0$
procedure
  Step-1: Generate pixel label map
  for Pixel P in Im_{static} **do**
    initialize empty priority-que q_p
    for seed in Im_{seeds} **do**
      Calculate Euclidean distance
      $D_s \leftarrow E_{ds}$
      push D_s **onto** q_p
    end for
    pop least distance L_{ds} **from** q_p
    $P_l \leftarrow label - of - L_{ds}$
    $P_i \leftarrow -1$
    $P_v \leftarrow [P_x, P_y, P_i]$
    $P_v[label] \leftarrow P_l$
    $P_{ms}[P] \leftarrow P_v$
  end for
  Step-2: Generate 8-Seed Neighbours
  for Pixel P in Im_{static} **do**
    $P_n \leftarrow 0$
    for n' in $8n_p$ of P $\in Im_{static}$ **do**
      $Check \leftarrow validity - of - n'$
      if $Check$ **then**
        $P_n \leftarrow n'$
      end if
    end for
    $P_{ms}[P][n] \leftarrow P_n$
  end for
end procedure

neighborhood map $P[n] \leftarrow 8n_p$ at the positions P_x and P_y which are generated in the static phase.

The second stage is to generate the initial seed vector map S_{vd}, which are used to allocate pixels to their associated seed places. The seed positions P_x and P_y from the original seed map Im_{seeds} and the intensity value P_i from the current input frame $Im_{dynamic}$ are taken into account here. For every valid seed 'S' $\in Im_{dynamic}$, a three-dimensional vector $S \leftarrow \{P_x, P_y, P_i\}$ is assigned at the positions P_x and P_y. The Seed 'S' is now pushed onto the seed vector map $S_{vd}[(P_x, P_y)] \leftarrow S$

$$D_{ps} = |S_x - P_x| + |S_y - P_y| + |S_i - P_i| \tag{2}$$

In the last stage, each valid pixel 'P' $\in$ the input frame $Im_{dynamic}$ is linked to its super-pixel seed 'S'. For each pixel 'P', a priority map q_p is built and set to zero. The manhattan distance is now computed as given by equation 2 for each seed neighbor vector $S_{vd}[S]$ of the current pixel vector P_v, where S_x, S_y, S_i are the seed vector components and P_x, P_y, P_i are the pixel vector components. The distance result D_{ps} is added to the priority queue. The nearest distance $L_d s$ from the priority queue is chosen when the method is completed, and all other recorded values are erased. The allotted seed label is then updated in the dynamic pixel map $P_{md}[P][label] \leftarrow label$. Similar to the seed update procedure in SNIC, the seed vector $S_{vd}[S]$ is now updated with the newly allocated pixel by taking the vector average of assigned seed vector $S_{vd}[S]$ and the current pixel vector P_v as given by equation 3.

$$S_{vd}[S] = (S_{vd}[S] + P_v)/2 \tag{3}$$

After generating the dynamic pixel map P_{md}, the associated seed vectors S of S_{vd} and S $\in Im_{dynamic}$ in the dynamic P_{md} and static pixel maps P_{ms} are matched to create the super-pixels seed labels of interest. This array of seed labels can now be utilised to isolate and produce the super-pixel segments using the pixel map generated in the dynamic phase. A dynamic pixel map for a homogeneous frame similar to IPM image of Figure 2b is as shown in Figure 4. Here $9X9$ seeds are considered in x and y dimensions.

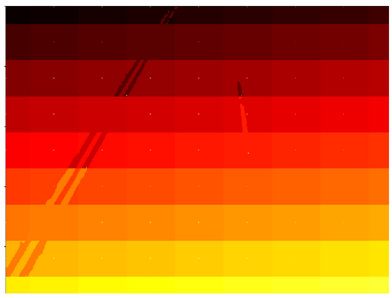


Fig. 4: Dynamic Pixel Map

IV. Experimental Results

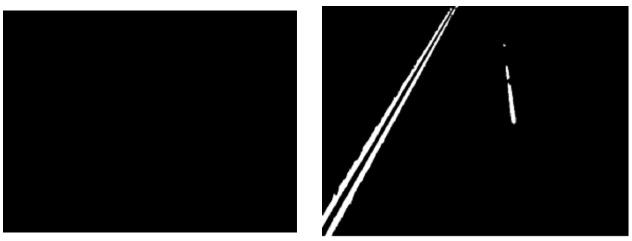
This section describes the working efficiency of the the proposed two-stage algorithm on the input video pictures from the lane-lines data set. The current experiments were run on a computer having a Intel Core i5-9300H processor with a base clock at 2.4 GHz using 8 GB of RAM and NVIDIA GeForce GTX 1650 GPU. For a given input IPM frame, the suggested pipeline efficiently localized the lane zones. Figure 7 depicts a succession of pictures created in response to an input to the proposed algorithm. Figure 5a is the input to static phase. It depicts a homogeneous dark input image with the exact dimensions as the subsequent input IPM frames.

The static phase creates the initial seed locations employed in the dynamic phase. A sample input IPM frame for the dynamic phase is shown in Figure 5b, this phase leverages the proposed super-pixel based lane-localization to create six output segments, as shown in Figures 7a, 7b, 7c, 7d, 7e and 7f. Individual segments can now be submitted straight to the lane modeling stage for efficient lane modeling. The static phase was timed at 90 seconds during the start of the algorithm and for subsequent IPM inputs to dynamic phase had a run-time of 0.69 seconds.

Algorithm 2 Dynamic Phase

Input: $Im_{dynamic}$, Im_{seeds}, P_{ms}
Output: P_{md}
initialize:
 $P_{md}[:] \leftarrow 0$, $S_{ms}[:] \leftarrow 0$, $S_{vd}[:] \leftarrow 0$, $S_{md}[:] \leftarrow 0$

procedure
  Step-1: Generate Pixel Hash-Map
  for Pixel P in $Im_{dynamic}$ **do**
    if $Intensity$ of $P \geq \lambda$ **then**
      $P_{md}[P][vector] \leftarrow [P_x, P_y, P_i]$
      $P_{md}[P][label] \leftarrow P_{ms}[P][label]$
      $P_{md}[P][n] \leftarrow P_{ms}[P][n]$
    end if
  end for
  Step-2: Initialise the Seed Vectors
  for Seed S in Im_{seeds} **do**
    $P_i \leftarrow Im_{dynamic}[P_x, P_y].Intensity$
    $S \leftarrow [P_x, P_y, P_i]$
    $S_{vd}[(P_x, P_y)] \leftarrow S$
  end for
  Step-3: Allocate Pixels to Seeds
  for Pixel P in P_{md} **do**
    $neighbours \leftarrow P_{md}[P][n]$
    $P_v \leftarrow [P_{md}[P][vector]]$
    $q_p \leftarrow empty$
    for Seed S in $neighbours$ **do**
      $D_{ps} \leftarrow \sum(|S_{vd}[S] - P_v|)$
      $push\ D_{ps}\ onto\ q_p$
    end for
    $L_{ds} \leftarrow pop\ q_p$
    $P_{md}[P][label] \leftarrow L_{ds}[label]$
    Update Seed position and values
    $S_{vd}[label] \leftarrow (S_{vd}[label] + P_v)/2$
  end for
end procedure



(a) Input for Static Phase (b) Input for Dynamic Phase

Fig. 5: Inputs for proposed algorithm

Hough transform (HT) and Probabilistic hough transform (PHT) are applied to an input image with curved lane markings as shown in Figure 6. It is demonstrated that with the addition of proposed super-pixel based pre-processing stage, the overall detection efficiency has been increased. Figure 8a and Figure 8c are the output images after applying hough transform and probabilistic hough transform on the input frame and Figure 8b and Figure 8d are the combined hough-transform output generated for each resultant segment of the 2-phase algorithm. As seen in the Figure 8b and Figure 8d the curvature and lane marker information is better analysable when the existing lane-modelling algorithms are combined with the proposed pre-processing stage. The direct HT algorithm had a run-time of 4.2 milliseconds, PHT algorithm had a run-time of 4.4 milliseconds, the segment-wise HT had a run-time of 6.8 milliseconds followed by segment-wise PHT had a run-time of 12.0 milliseconds for eight super-pixel lane segments.

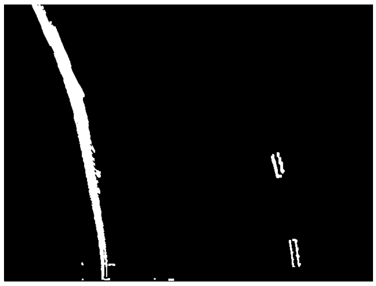


Fig. 6: IPM Image with Curved Path

(a) First Segment

(b) Second Segment

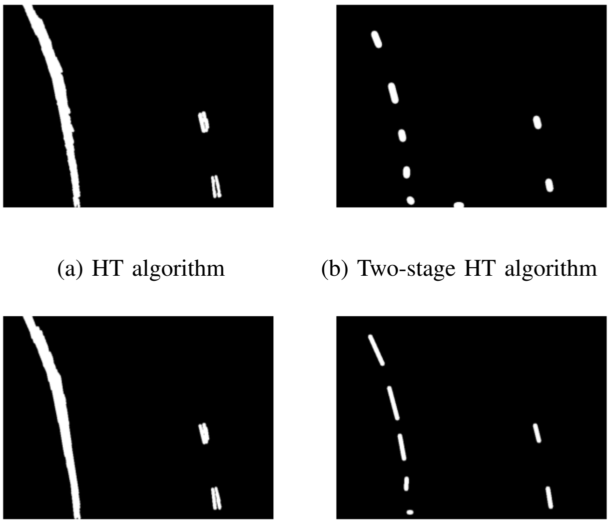
(c) Third Segment

(d) Fourth Segment

(e) Fifth Segment

(f) Sixth Segment

Fig. 7: Super pixel segments from dynamic phase



(a) HT algorithm (b) Two-stage HT algorithm

(c) PHT algorithm (d) Two-stage PHT algorithm

Fig. 8: Comparative Outputs

V. CONCLUSIONS

This study developed a two-step pre-processing approach to improve the lane-modeling stage. An inverse perspective mapped picture of a curving road is fed into Hough transform (HT), Probabilistic hough transform (PHT), and the two-stage upgraded versions of HT and PHT for testing and validation. The experimental findings show that our suggested pre-processing approach improves the detection effectiveness of the lane-modeling algorithms while keeping the lane curvature of the discovered components. The suggested algorithm's resilience and computing efficiency may be enhanced by merging many different lane-modeling techniques. The algorithmic approach can potentially be modified to reduce processing time further.

REFERENCES

[1] M. Divyaprabha, M. Thangavel, and P. Varalakshmi, "A comparative study on road safety problems," in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–7, 2018.

[2] S. Agrawal, I. K. Deo, S. Haldar, G. R. Kranti Kiran, V. Lodhi, and D. Chakravarty, "Off-road lane detection using superpixel clustering and ransac curve fitting," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1942–1946, 2018.

[3] S. Shirke and R. Udayakumar, "Lane datasets for lane detection," *2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0792–0796, 2019.

[4] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognition*, vol. 73, pp. 216–234, 2018.

[5] Q. Lin, Y. Han, and H. Hahn, "Real-time lane departure detection based on extended edge-linking algorithm," in *2010 Second International Conference on Computer Research and Development*, pp. 725–730, 2010.

[6] Y. Choi, J. H. Park, and H.-Y. Jung, "Lane detection using labeling based ransac algorithm," *International Journal of Computer and Information Engineering*, vol. 12, no. 4, pp. 245 – 248, 2018.

[7] D. Li, Y. Zhang, P. Wen, and L. Bai, "A retinex algorithm for image enhancement based on recursive bilateral filtering," in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, pp. 154–157, 2015.

[8] K.-Y. Chiu and S.-F. Lin, "Lane detection using color-based segmentation," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pp. 706–711, 2005.

[9] W. Chen, W. Wang, K. Wang, Z. Li, H. Li, and S. Liu, "Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 7, no. 6, pp. 748–774, 2020.

[10] M.-A. Andrei, C.-A. Boiangiu, N. Tarbă, and M.-L. Voncilă, "Robust lane detection and tracking algorithm for steering assist systems," *Machines*, vol. 10, no. 1, 2022.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," p. 15, 2010.

[12] M. Aly, "Real time detection of lane markers in urban streets," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 7–12, 2008.

[13] D. Nava, G. Panzani, and S. M. Savaresi, "A collision warning oriented brake lights detection and classification algorithm based on a mono camera sensor," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 319–324, 2019.

[14] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4895–4904, 2017.

[15] Y. Li, L. Chen, H. Huang, X. Li, W. Xu, L. Zheng, and J. Huang, "Nighttime lane markings recognition based on canny detection and hough transform," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 411–415, 2016.