Road Recognition and Lane Detection using Deep Learning

Moumita Chanda[1], Sadiya Rahman[1], Md. Tanvir Ahmed[1] and Md. Fazlul Karim Patwary[1]

[1] Institute of Information Technology, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh
patwary@juniv.edu

Abstract. An autonomous vehicle needs to be familiar with its surroundings. The safety of the transportation system is greatly enhanced by advanced driving assistance systems (ADASs). Road detection is one of the steps that a driving car must do. Is it possible for a computer to recognize a road in a single photograph for this purpose? This question is addressed using the lane detecting techniques. Roads and lanes are tough for machine learning to differentiate because of training a machine to recognize a road. Over the past few decades, a number of lane identification technologies have been created and integrated into various autonomous cars. It is still very difficult to create lane recognition technology that can effectively identify a road lane in a range of road conditions. This research provides a composite approach for road detection from image processing using convolutional neural networks by testing 150 photographs that include a road, jungle, muddy road, and barriers. It will decide if an image contains a road or not. In this essay, we first establish whether a road exists. The second step is to find a lane on the finished road. The benefit of the proposed technology is that if there is a road, the automobile can continue to move forward; otherwise, it will stop.

Keywords: Autonomous car, Advanced Driving Assistance Systems, CNN, Road recognize, Lane detection, Hough Transform.

1 Introduction

Every year, over 1.35 million people are died in traffic accidents. Every day, over 3,700 people lose their lives on the highways. 20-50 million people have been broken or rendered immobile. Road accidents affect people aged 15 to 44 [1]. The problem of traffic accidents in Bangladesh cannot be overlooked. According to a survey conducted in January 2021, 427 traffic accidents occurred, with 484 people killed and 673 injured. In comparison to the same period last year, the accident rate was roughly 25.58 percent [2]. Careless driving, over speeding, being overburdened, overtaking, disobeying rules, endangering competitors, extended periods of driving without rest, drug and alcohol usage, unsafe roads, and escaping safety are all factors that contribute to the increased frequency of accidents. Seat belts and other safety equipment to lane driving and improper overtaking, driver distraction [3]. To reduce road accidents, people should develop a positive attitude toward driving, wear safety equipment such as seats and helmets, take breaks between long drives, be aware of the road environment, and refrain from using drugs while driving [4].

If a machine learns to drive, it will not become weary while driving, will not be distracted from their work, and will not be required to wear any safety equipment. Reduced traffic accidents will be helpful. An autonomous vehicle is capable of Transportation patterns should be replaced. An autonomous vehicle senses its surroundings and operates without human intervention. Autonomous vehicles are safer than human-driven vehicles [5]. We don't need a person to drive the autonomous automobile at any moment, and we don't need to be present in the vehicle at all because it operates like a skilled human driver. The road environment is sensed by autonomous vehicles, which then take action utilizing actuators. The security control functions of an autonomous car, such as steering, acceleration, and braking, are performed without the assistance of the driver [6]. Autonomous vehicles operate by rapidly sensing vast amounts of data. The moment has come for self-driving automobiles to keep up with the times. Many advancements in self-driving cars have been made through investing in a variety of firms in order to be the first to market. Wemo, a Google product, has automatically accumulated millions of miles [7]. Tesla is an autonomous vehicle that can stay in its lane while driving. It was created by humans. There have been numerous instances in which a self-driving or autonomous vehicle has failed to drive on a road. An autonomous vehicle failed to drive on a highway in Arizona. A Tesla automobile collided with a truck on the highway in Florida [8]. Though autonomous vehicles have made progress, considerable work has to be done to improve driving competence. Uber, Google, and General Motors are among the companies that have developed self-driving automobiles. They use cameras, light detection and ranging (LIDAR), and sensors to build these autos. These vehicles are built with advanced autonomous technologies that provide high accuracy performance. Deep networks in machine learning have shown promise in detecting road conditions [9]. Deep learning produced amazing results in the object detection competition, and picture recognition is now being used for self-driving cars. The Convolution Neural Network (CNN) has been successful in detecting objects such as roads, lanes [10], persons [11], text, and so on. By configuring the output layer according to image recognition, CNN is employed not only for picture classification but also for detection. CNN provides accurate picture categorization results. For image recognition, CNN is the most powerful. CNN has been used in trading for almost 20 years, but their assumption has blown up in recent years. CNN is gaining popularity as a result of two innovative advancements.

The first is the Enormous Scale Visual Recognition Challenge (ILSVRC), which consists of large labeled data sets that have just become available for training and validation [12]. Second, on the multiple graphics processing units, the CNN machine learning method is implemented (GPU). Can a machine or an automobile identify whether or not a road exists given an image? To detect a road, the system must first learn what it is and how it appears. There are various types of roads, including muddy roads, jungle roads, and high-way roads. For machines to drive an automobile, road image analysis is critical. To begin, an autonomous vehicle must understand which roads are ideal for driving. A CNN model was presented to teach a machine how to detect a road by perceiving the road surroundings. Many levels are completed in an autonomous vehicle. Road detection and classification are critical tasks in Computer Vision (CV). After detecting a route, an autonomous vehicle must find a lane to travel on. To detect lane on the road, we employed an algorithm. To detect lane, the Hough Transform is employed. When the boundary is specified with a small number of points, the Hough transform aids in finding the edges parameter. The canny edge detection approach also employs the Hough transform. It is used to distinguish between lines and non-lines. It draws lines and adds points to an image. A real-time training dataset is required to detect a road. The information is saved in jpeg and png formats. To begin, the machine must be taught about the visuals. We manually collected datasets, which took a long time. There are many different types of photos in this dataset, such as muddy roads, rail roads, highways, and jungles. After the computer has been trained on the images, the testing procedure is used. The neural network is fed with training images. All of the photos in the databases have their images compressed. Images are transmitted to a neural network, which classifies them into several categories. The photos are then forwarded to multiple network tiers. The output is then detected. The proposed method improved the discovered road's quality. The approach easily fits the enormous data collection. The model is capable of detecting roads more precisely.

1.1 Work Flow

A data collection containing various sorts of roads, such as muddy roads, highways, and jungles, will be used to create a dataset. The image is sent to a training data set, which instructs the system on how to navigate the route. Label all of the photos using the image data generator.

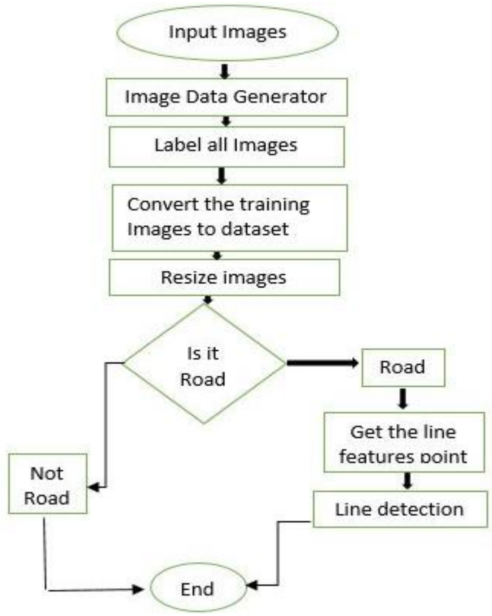


Fig. 1. Workflow

Convert the training images into a data collection that the neural network can use. Resize the photos from 0-255 to 0-1 pixels.

2 Literature Review

Many research projects relevant to this subject have been overseen in English as well as other languages. All of these research projects are unaffordable. An autonomous car does a fantastic job of detecting roads and lanes. Despite these efforts, further research is needed to improve the current systems. In this paper, many machine learning and deep learning algorithms were used. Many researchers developed their model using different approaches. The following section goes over some of these approaches. [13] Presented a method for detecting lanes in urban environments. Because of wear and tear, occlusions, and complex road geometry, the lane markings are not clearly visible. In the image mapped from the opposite angle, the particle filter is applied from the base row to the peak. Filters are adjusted in this way that tracks multiple lanes. [14] Offers a fast and effective method for detecting urban street lane markers in real time. It starts by creating the best perspective of the road image possible, using reverse point of view mapping to reduce the influence of the view. A selective-based two-dimensional Gaussian component is used to filter the view at that point. The filter is tuned specifically for bright lines in dark backgrounds with specific widths. Using the q percent quantile value from the filtered image and removing any values below the limit to the line markers. Following the detection of straight lines, RANSAC line fitting using the Hough transform is performed. It establishes the foundation for the RANSAC spline fitting stage. At that point, a post-processing step is performed to localize the spline and extend the image. At that point, a post-processing step is performed to localize the spline and extend the image. The calculation does not go as planned. It can detect the limits of any path within the image, but not the present one.[15] Suggested a lane detection method that works well at night. The Region of Interest is used initially by the algorithm (ROI). The sky and other immaterial things are removed from the Region of Interest (ROI). In the next stage, a gray scale image is created by averaging three color channels. To elongate the dashed lanes Temporal blurring is used. To extract the bright objects adaptive thresholding is done. Left and right halves are the parts of resultant binary image. On each half for detecting the straight, the Low-Resolution Hough Transform is applied.

Table 1. Method Dataset Detected portion.

Paper Author	Method	Data-set	Detects
[M. Rezaee]	DNN	Fredericton	Road
[Tiago]	DNN	Cityscapes	Road region
[Jose M]	Vision based	250 images	Shadow Road pixels
[Dan Ciresan]	DNN	26640 images	Small labels
[M.J ValadanZoej]	CNN	Ikonos,QuickBird images	Road pixel
[Jian Wang]	SVM	290 images	Road region
[Hui Kong]	LASV	1003 images	Road
[VolodymyrMnih]	NN	500, 50 square kilometer video	Shadow road
[Jae-Seol Lee]	CNN	KITTI	Road
[B. Southall]	Vision based	Video	vehicle position
[Alberto Broggi]	Artificial vision	Public data-set	Obstacles
[Guoliang Liu]	PPF-kernel	DRIVSCO	Kernel density
[Zheng Li]	DTMARF	Cell55	Unstructured road
[Neethu John]	CNN	192 images	Road region
[Hironobu Fujiyoshi]	CNN	1000 images	Objects

A Gaussian kernel is used to extract lane markers with repeated matched filter to extract lane markers. [16] suggested a lane detection method based on machine learning. To locate the correct lane markings. Both fundamental and measurable highlights of the extricated shining form are related to the Neural Organize in [17]. Convolutional neural systems demonstrated dominating execution for picture upgrade and path placement.[18] Proposed a video-based land detection system for use at night. For picture pre-processing, versatile splay ROI and Hough change the strategy steps incorporate the Gabor channel administrator [19].A route location and following calculation generated by [20] can handle challenging conditions such as fuzzy path markers, path ebbs and flows, and part paths. A solution has been developed based on the parallel nature of lane markers [21]. To begin, we undertake Converse Point of View Mapping. After that, the IPM image is converted to a grayscale image Normalized Cross Correlation is now used to filter the image. It identifies a group of straight lines using the Polar Randomized Hough Transform (RHT). Each line with the best match has high score coordinates or peaks. Each best-fitting line in space has high-scoring coordinates or peaks. If two lines have the same value, they can be paired to see if they are parallel lines peaks. A model called Dual-View CNN is proposed to handle lane detecting [22] [Kaiming He, et al. 2016]. (DVCNN). For example, [23] divides lane detection into two steps. Binary segmentation is used to distinguish between lane pixels and background pixels. Create a framework to accomplish lane border and road area segmentation at the same time [24]. [25] Mapping pixels onto a ground plane by a learnt homo graphy is a powerful method for regularizing individual lane curve fitting. Recent work has enhanced segmentation by adding a history of images merely conditioning on the current frame [26]. Clustering methodologies include K-means Clustering, Isodata Clustering, Histogram-Based Clustering and Recursive Variation, and Global-Theoretic Clustering [27].

3 Image Pre-Processing

3.1 Smoothing images

It is the most often used image processing procedure. Image blurring is another name for it. Smoothing and blurring are techniques for removing noise from photographs. We can use a variety of linear filters to smooth photos. Linear filters are simple and quick to create. Other types of filters are available in OpenCV.

Homogenous filter, Gaussian filter, Median filter, and Bilateral filter are the four types. Low-pass and high-pass filters can be used to filter images in a one-dimensional signal. The low pass filter blurs the images while removing noise. The high-pass filter aids in the detection of image edges.

Homogeneous filter: The simplest approach of smoothing/blurring photos is to use a homogenous filter. All pixels in a homogeneous filter have roughly equal weight, which is why they are called homogeneous. Each pixel in the output is the average of its kernel neighbors. Kernel is a tiny matrix that is used to blur, sharpen, and detect edges. Because matplotlib reads photos in RGB format while OpenCv reads images in BGB format, we must first transform the image from BGB color to RGB color. The kernel is then defined. For the homogenous filter, we used the 2D filter approach. As the kernel was increased, the image noises steadily decreased. The image gets smooth and noise is reduced after using a 2D filter.

Algorithm:
Input: A simple image with noise
Output: Detected Image without noise
Step-1: Take the input image containing noise to be filtered.
Step-2: Convert the image from BGB color to RGB color.
Step-3: Apply 2-D filter method on the image
Step-4: Define kernel
Step-5: Reduce noise and enhance the image

Blur Algorithm: The photos are blurred using the blur algorithm. Images are smoothed out by blurring them. It uses pixels in the area of a central pixel. After that, it uses kernel to average all of the pixels before exchanging with the central pixel. Image smoothing and noise reduction are achieved by increasing kernel size. X Y is the Blur formula.

Algorithm:
Input: Input image
Output: Detected Image without noise
Step-1: Take the input image
Step-2: Initialize pixels size
Step-3: Average all the pixels by kernel
Step-4: Replace the average pixels with central pixel
Step-5: Blur the image by increasing kernel size
Step-6: Show the blurred image.

Blur filter method is easy and simple to understand. But it has a disadvantage. Though weight of all the pixels inside the kernel is same so it overblur the image. For this reason, important edges missed out and lane cannot be detected accurately. Gaussian blur filter is solution for this problem. The blur filter approach is basic and straightforward. However, it has a drawback. Though the weight of all the pixels inside the kernel is the same, the image is overblurred. As a result, key edges are ignored and lane detection is inaccurate. This problem can be solved with a Gaussian blur filter.

Gaussian blur filter: The Gaussian blur filter has different kernel weights in both the x and y directions. Pixels in the middle of the kernel have a higher weight than pixels on the sides, which have a lower weight. It outperforms the blur algorithm. It’s made to remove high-frequency noise from images using a Gaussian distribution. This method blurs the image more organically than the Blur filter method. More images will be saved efficiently.

Algorithm:
Input: A simple image with noise
Output: Detected Image without noise
Step-1: Take the input image
Step-2: Initialize pixels size
Step-3: Apply Gaussian equation
Step-4: Put kernel size
Step-5: Increase kernel size
Step-6: Get the blurred Image
Gaussian filter method gives nice result but slightly slower than Blur filter method.

Median filter: The median filter replaces the value of each pixel with the median of its neighbors. To use this method, first set the kernel size. Then replace the central pixel with the neighboring pixel's median. Because the center pixel is replaced with a pixel that exits the image, it is effective for removing white and black pixels.P'(x,y)→ medianP(x+i, y+j)—(i,j)CR.

Algorithm:
Input: Image P of size M × N
Output: Image P' of same size as P
Step-1: Take the input image
Step-2: Initialize the pixels size
Step-3: Replace central pixel with median of the neighbor pixel
Step-4: Increase the kernel size
Step-5: Get filtered output.
By increasing kernel size, many detail of an image will be removed. It smooths the edge in the image.

Bilateral filter: A bilateral filter is used to minimize visual noise and fine details. When we need to preserve the sharpness of edges in photos, we can use the Bilateral filter to sharpen the edges even if they are blurred. All pixels in the same neighborhood have the same weight and represent the same object. The weighted combination of neighbor pixel values determines the output pixel value. Bilateral filter method reduce noise and sharps the edges even if blurred

Algorithm:
Input: Image R of size M × N
Output: Image S of same size as R
Step-1: Take the input image
Step-2: Initialize the size of the pixel
Step-3 Pixel represents object
Step-4: Apply bilateral equation
Step-5: Increase diameter
Step-6: Get filtered Image
Bilateral filter method reduce noise and sharps the edges even if blurred image.

(a) (b) (c)

(d) (e) (f)

Fig. 2. (a) Original image, (b) Homogeneous filtered image, (c) Blur filtered image, (d) Gaussian Blur filtered image, (e) Median filtered image, (f) Bilateral filtered image

Smoothing photographs is accomplished using any of these ways. Because the same weighted pixels are over blurred in the Bilateral filter approach, crucial edges are lost and the lane cannot be recognized accurately. The Gaussian blur approach lowers image noise but is a little slow. The features of an image are lost with the median filter method, making it unable to discern lines properly. According to our findings, the Bilateral filter is the best at not only smoothing images but also sharpening edges.

3.2 Edge detection

Edge detection is a common image processing technique for detecting points in an image where there are discontinuities. Inside an image, an image gradient is utilized to smooth down the edges. In a picture, an image gradient is used to modify the intensity or color. In OpenCv, there are two types of edge detection algorithms. Gradient and Gaussian methods are the two options. These techniques operate in grayscale mode.

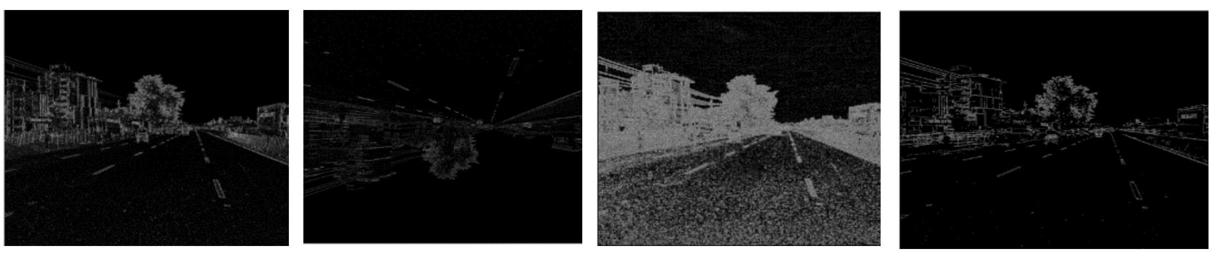
Gradient method: In an image, the gradient approach computes first-order derivations. Gradient methods include the Sobel X and Sobel Y detectors. To the image, apply a gradient mask. The magnitude is then calculated using the Sobel equation. Sobel X:The X-direction is where the intensity changes direction. dX=1 is the order of derivatives X, and dY=0 is the order of derivatives Y. Sobel Y:The Y-direction is when the intensity changes direction. dX=0 is the order of X derivatives, while dY=1 is the order of Y derivatives.

Algorithm of Sobel Operator:
Input: Image R of size M × N
Output: Image S of same size as R
Step-1: Take the input image
Step-2: Apply gradient mask Xx, Yy to the input image
Step-3: Apply sobel equation and the gradient
Step-4: Find the magnitude of the gradient
Step-5: The magnitude is the output.

Gaussian method: In a picture, the Gaussian approach computes second order derivations. Gaussian methods include the Canny edge and Laplacian detector. Laplacian method: The second derivatives of an image are computed using the Laplacian method. A datatype is used in the Laplacian method to change a picture from white to black. The laplacian approach, which is used to detect all of the edges within an image, uses kernel size. It generates better results if we reduce the kernel size. Algorithm: Input: Image R of size M × N Output: Image S of same size as R Step-1: Take the input image Step-2: Transform image white to black Step-3: Apply laplacian equation Step-4: Initialize kernel size and increase kernel size Step-5: Display image and detect Edges Canny edge method: An edge detector operator is the Canny edge detector. It detects a wide spectrum of images using a multi-stage technique. There are 5 steps:

Algorithm:
Input: Image R of size M × N
Output: Image S of same size as R
Step-1: Take the input image
Step-2: Smooth the image with 2D Gaussian method: n * I
Step-3: Gradient calculation: n *I
Step-4: Non-maximum suppression: It is used to get rid of superiors response to edge detection: —— n * I—
Step-5: Double threshold: It helps to determine potential edges.
Step-6: Edge tracking by Hysteresis: It suppresses all the other edges which are weak and not connected to strong edges and finally detect the edges.

The Laplacian edge detection approach has poor performance at image corners and is sensitive to noise along an edge. The most effective and extensively utilized detection methods are canny detection methods. It is less noisy than previous approaches. However, these identification algorithms do not discern which edges relate to the boundary. The solution to this problem is the Hough transform.



(a) (b) (c) (d)

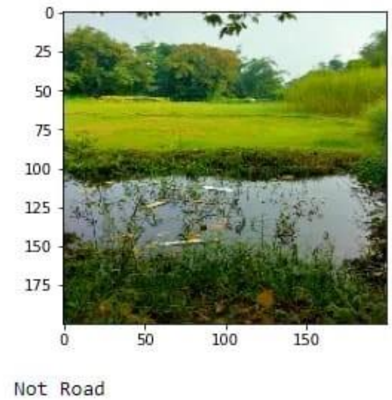
Fig. 3. (a) SobelX image, (b) SobelY image, (c) Laplacian image, (d) Canny Edge image.

4 Result

Road Detection
Library: Tensorflow, Numpy
Model: Sequential
Activation function: Relu,Sigmoid
Optimizer: Adam
Epochs: 2

We utilize Tensorflow and Numpy as libraries from dataset to hand-pick is it road or not. Tensorflow is a data-flow-based math library. It is a machine learning software library that is free and open-source. NumPy is an array-based programming language. It includes functions that work with linear algebra, the Fourier transform, and matrices. In the activation function, ReLU and sigmoid are utilized. ReLU stands for Rectified Linear Unit, and it returns 0 if it gets any negative input, but it returns any positive number x. f(x)=max(0,x) The sigmoid function is also known as the logistic function. Its transform value ranges from 0.0 to 1.0.

(a)



(b)

Fig. 4. Output (a) Road, (b) Not Road

Hough transform: When the boundary is defined with a minimal number of parameters, the Hough transform aids in finding the edges. The canny edge detection approach employs the Hough transform. It is used to distinguish between lines and non-lines. It draws lines and adds points to an image. In the xy plane, it does not operate. It transforms the xy plane into the mc plane. Line equation: y=mx + c , c=-mx + y There are two spaces available. The first is picture space, whereas the second is parameter space. Consider one image space line and one specific location on that line (Xi, Yi).

We find a line in parameter space when we put the point (Xi, Yi) in the equation c=-mx + y c=-m*Xi + Yi. The line in parameter space is made up of numerous points, and the lines in image space pass through the point (Xi, Yi). When we draw more lines in parameter space, they intersect at a place called (m, c). If we take a point in image space that does not belong to a line, it will not cross at the parameter space point (m, c).

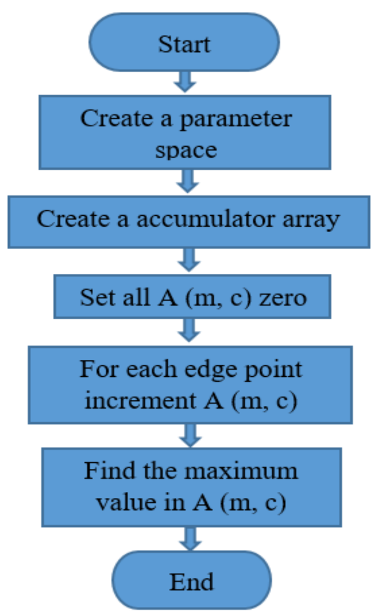


Fig. 5. Flow chart of Hough Transform

Line detection algorithm

1. Create a parameter space named (m, c).
2. Create a accumulator array A(m, c).
3. Set A(m, c)=0 for all (m, c).
4. For each edge point (Xi, Yi) A(m, c) = A(m, c) + 1 If (m, c) lies on the line: c = -m * Xi + Y
5. Find the maximum value in A(m, c) Here, 3 is the intersect point in parameter space

Big and many lines may be merged is possible in Hough transform. Small noises can be removed in image.

5 Conclusion

In this paper, a method is used to determine whether or not a road exists, as well as to identify a road lane. We will use machine learning approaches to develop a successful strategy for acquiring training data and lane marking. The lane detecting work is made easier with the help of machine learning. This procedure will be carried out using CNN. This strategy is expected to produce greater accuracy than current studies in this field. Data sets in which we manually acquire data. The results of this experiment show that the proposed approach effectively detects lanes in a variety of road conditions.

5.1 Limitation of the Existing Systems

Some limitations are found out:
(1) When lanes are abruptly changed in autonomous driving, even when there are no vehicles ahead or to the side, the passenger in the car may be anxious about why the lanes were changed.
(2) The blur filtering cannot be removed.
(3) There is opportunity for improvement, particularly in the loss function, because lowering the loss value does not necessarily result in a reduction in the error rate.
(4) Its inability to appropriately categorize different types of road surfaces or areas with strong lighting.
(5) From frame to frame, predict the position of boundaries and the color of the road surface: The influence of shade and water on the road surface is removed.
(6) The algorithm will treat the water area as non-road if it is too large and reflects light.
(7) Color constancy theory ignores the enlighten effect in order to eliminate shadows' negative effects.
(8) Asphalt, sand, and stones cannot be separated on unstructured roads.
(9) Compute the DSI separately for each color channel and compare the results.
(10) There is no system for detecting and correcting real-word errors that is 100 percent accurate.

References

1. https://www.thewanderingrv.com/car-accident-statistics/.
2. https://www.unb.com.bd/category/bangladesh/road-crashes-killed-484in-jan-2021/64511: text=At
3. http://jhtransport.gov.in/causes-of-road-accidents.html
4. https://www.geico.com/information/safety/auto/teendriving/top-ten-tips/
5. M. Kyriakidis, J. C. de Winter, N. Stanton, T. Bellet, B. van Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat, et al., “A human factors perspective on automated driving,” Theoretical Issues in Ergonomics Science, vol. 20, no. 3, pp. 223–249, 2019.
6. NHTSA Preliminary Statement of Policy ires¸an, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep big multilayer perceptrons for digit recognition,” in Neural networks: tricks of the trade, pp. 581–598, Springer, 2012.
7. M. Isaac, “Uber suspends tests of self-driving vehicles after arizona crash,” New York Times, 2017.
8. M. Rezaee and Y. Zhang, “Road detection using deep neural network in high spatial resolution images,” in 2017 Joint Urban Remote Sensing Event (JURSE), pp. 1–4, IEEE, 2017.
9. T. Almeida, B. Lourenc¸o, and V. Santos, “Road detection based on simultaneous deep learning approaches,” Robotics and Autonomous Systems, vol. 133, p. 103605, 2020.

10. J. Wang, Z. Ji, and Y.-T. Su, “Unstructured road detection using hybrid features,” in 2009 International Conference on Machine Learning and Cybernetics, vol. 1, pp. 482–486, IEEE, 2009.
11. Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/ challenges/LSVRC/.
12. S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, “Robust lane detection in urban environments,” in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 123–128, IEEE, 2007
13. M. Aly, “Real time detection of lane markers in urban streets,” in 2008 IEEE Intelligent Vehicles Symposium, pp. 7–12, IEEE, 2008.
14. A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, “A layered approach to robust lane detection at night,” in 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Sy
15. S. Lim, D. Lee, and Y. Park, “Lane detection and tracking using classification in image sequences,” KSII Transactions on Internet and Information Systems (TIIS), vol. 8, no. 12, pp. 4489–4501, 2014.
16. J. Kim, J. Kim, J. G. Jang and M. Lee, “Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection,” Neural Networks, March, 2017.
17. X. He, Z. Duan, C. Chen, and F. You, “Video-based lane detection and tracking during night,” in CICTP 2019, pp. 5794–5807, 2019.
18. M. Mokhtarzade and M. V. Zoej, “Road detection from high-resolution satellite images using artificial neural networks,” International journal of applied earth observation and geoinformation, vol. 9, no. 1, pp. 32 –40, 2007.
19. Z. Kim, “Robust lane detection and tracking in challenging scenarios,” IEEE Transactions on intelligent transportation systems, vol. 9, no. 1 , pp. 16–26, 2008.
20. A. Borkar, M. Hayes, and M. T. Smith, “Polar randomized hough transform for lane detection using loose constraints of parallel lines,” in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1037–1040, IEEE, 2011.
21. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
22. Davy Neven, Bert De Brabandere, StamatiosGeorgoulis,MarcProesmans, and Luc VanGool. Towards end-to-end lane detection: an instance segmentation approach. In IEEE Intelligent Vehicles Symposium, pages 286–291. IEEE,2018.
23. J. Zhang, Y. Xu, B. Ni, and Z. Duan, “Geometric constrained joint lane segmentation and lane boundary detection,” in proceedings of the european conference on computer vision (ECCV), pp. 486–502, 2018.
24. D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, “Towards end-to-end lane detection: an instance segmentation approach,” in 2018 IEEE intelligent vehicles symposium (IV), pp. 286–291, IEEE, 2018.
25. Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” IEEE transactions on vehicular technology, vol. 69, no. 1, pp. 41 –54, 2019.
26. E. Al-Hujazi and A. Sood, “Range image segmentation combining edgedetection and region-growing techniques with applications sto robot binpicking using vacuum gripper,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 6, pp. 1313–1325, 1990.
27. Almeida, T., Louren ̧co, B., Santos, V.: Road detection based on simultaneous deep learning approaches. Robotics and Autonomous Systems 133,103605 (2020)
28. Jung, J., Park, M., Cho, K., Mun, C., Ahn, J.: Intelligent hybrid fusion algorithm with vision patterns for generation of precise digital road maps in self-driving vehicles. KSII Transactions on Internet and Information Systems (TIIS) 14(10), 3955–3971 (2020)
29. Yu, Z., Ren, X., Huang, Y., Tian, W., Zhao, J.: Detecting lane and road markings at a distance with perspective transformer layers. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6 (2020). IEEE.
30. Baheti, B., Innani, S., Gajre, S., Talbar, S.: Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 358–359 (2020).