e-ISSN: 2146 - 9067

International Journal of Automotive Engineering and Technologies

journal homepage:

https://dergipark.org.tr/en/pub/ijaet

Original Research Article

A new approach for camera supported machine learning algorithms based dynamic headlight model's design

hosted by Turkish JournalPark ACADEMIC

Şafak Yaşar[1], Mustafa Şahin[2], Onur Akar[3, *]

[1] Department of Electrical Engineering, University of Afyon Kocatepe, Afyon, Türkiye.
[2] Department of Electronics and Automation, University of Health Sciences, Istanbul, Türkiye.
[3, *]Department of Electronics and Automation, Marmara University, Istanbul, Türkiye.

ARTICLE INFO

Orcid Numbers
1. 0000-0003-0759-8933
2. 0000-0003-2284-8507
3. 0000-0001-9695-886X

Doi: 10.18245/ijaet.1168186

* Corresponding author
onur.akar@marmara.edu.tr

Received: Aug 29, 2022
Accepted: Oct 27, 2022

Published: 30 Dec 2022

Published by Editorial Board Members of IJAET

© This article is distributed by Turk Journal Park System under the CC 4.0 terms and conditions.

ABSTRACT

Traffic accidents continue to be a significant issue in modern society. Accidents usually happen on dark, mountainous, narrow, steep and curved roadways. One of the primary causes of such accidents is the drivers' weak sight brought on by the headlights of moving vehicles. In this study, a dynamic headlight model was designed using camera supported machine learning algorithms to improve the drivers' vision during night drive. In this design, the issues of enabling a lighting field supported by image processing programmed with machine learning, dynamic adjustment of the high beam headlights' LED cells in response to the vehicle approaching from the opposite direction, traffic-sign recognition system, lane-keeping system, and automatic adjustment of headlight angles were addressed. In this direction, a novel dynamic headlight model that will reduce the risk of accidents caused by lighting was presented, and its analyses were performed.

Keywords: Vehicle headlight system, Machine learning, Traffic-sign recognition, Lane-keeping, Vehicle recognition

1. Introduction

Traffic accidents are the primary problem affecting human life. More than 25% of all the accidents in the USA [1], and about 15% of the accidents involving death in Germany arise from rear-end collisions [2]. When numerous accidents and their causes were analyzed, it was observed that they generally arise from a lack of perfection in human sight and attention. Moreover, the role of topography in the occurrence of these accidents is an inevitable fact. The insufficiency of lighting and the presence of blind spots at points where the topography is uneven are also among the greatest causes of the accidents. Due to blind spots, such areas are unable to be seen by looking ahead, or by looking via the wing mirrors [3,4]. In addition, the curves, poor infrastructure, and insufficient lighting on the roadways also cause the blind spots. Driving is principally a visual task, because the vision function contributes to 90% of the conditions required for driving [5]. Considering the importance of vision function during driving, the direct effect of the quality and flexibility of the vehicle's lighting on the accident risk cannot be ignored [6].

As seen in Figure 1, on double-lane roads, especially during night time and in dark environments, the light of the vehicles approaching from the opposite direction is able to cause the drivers to go blind for a few seconds.

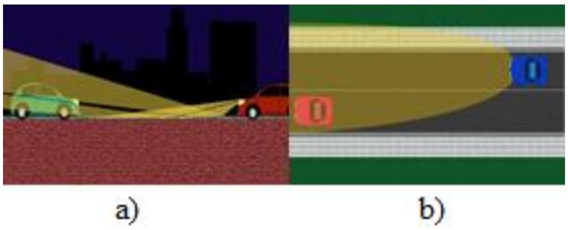

Fig. 1. a) reflection of headlights, b) lighting caused by the vehicles approaching from the opposite direction.

In intercity transportation, the drivers have to consider changing the high beam to low beam for the vehicles approaching from the opposite direction. And this circumstance causes problems for the drivers in the opposite direction. The states of instantaneous blindness as well as attention deficit are able to occur during the change of beams [7-9].

Fig. 2. a) Conventional headlight, b) Rotatable headlight vision.

At irregular road curves, the drivers are unable to be prepared for the risk they will face. This is because the vehicle's headlights create a lit area in the direction of the vehicle, as also seen in Figure 2. Lighting in the direction of the vehicle causes the formation of a dark area during driving through a curve. And such a dark area threatens the drive's safety and causes accidents.

In the study [10], a system for automatic vehicle headlight control was modeled and experimentally tested. The conditions of the vehicle were monitored with a vehicle speed sensor and an electronic level. Variations in the projection angle of the headlight system were tested concurrently during the study. The model consisted of a control box and an electronic level that could be changed in a way that allowed parameter adjustment under different operational and functional requirements, allowing the system to have an extensive range of applications in different vehicles. In the study [11], an effective method for detecting the preceding vehicle during night drive by a vehicle equipped with a camera was integrated was presented and applied to an embedded system. In the referred study, vehicle detection was performed by the means of vehicle headlights and taillights through the use of image segmentation and model analysis techniques. In the study [12], a system for checking whether the headlights of a vehicle are in a high beam or low beam state according to the input from a forward-looking video camera was designed. The system was operated by the use of a low-cost Complementary Metal Oxide Semiconductor camera mounted on the vehicle's windshield and behind the rear-view mirror. No metadata such as the vehicle's speed or steering angle was used. The only input to the system consisted of a series of 752x480 pixel color images at 30 Hz. The most significant point of the system was the detection of vehicles' headlights. Principally, the received image was thresholded, and then an algorithm was operated to find the possible light sources. The thresholding was actualized in the red area in order to make sure of the detection of the distant taillights. In the study [13], a Wireless Sensor Network based controller was developed for transmitting the sensor data faster and more efficiently among the vehicles in order to reduce the number of accidents arising from temporary driver blindness. Low delay time allowed faster headlight intensity adjustment among the vehicles, significantly reducing the cause of temporary blindness. The intensity of the headlight of the vehicle approaching from the opposite direction was detected. Considering the adjusted threshold headlight intensity, a system capable of automatically reducing the headlight intensity of the vehicle approaching from the opposite direction was designed through a wireless sensor network. Thus, it was intended to reduce the temporary blindness time caused by overexposure to headlights. In the study [14], a system for autonomous vehicles estimating the headlight distance through the use of deep neural networks was designed. A completely new computer vision and machine learning application was presented in order to automatically detect the incorrectly positioned headlights by estimating the angles of inclination from the images of a camera connected to the vehicle. It was intended to obtain high performance in terms of accuracy and integrity through end-to-end training of a deep neural network.

In the study [15], the design and integration of lane change warning operating under different road and lighting conditions, adaptive headlight, and windshield wiper systems were presented. In the system, Raspberry pi was used for video processing, and Arduino Mega was used as the operating unit for Adaptive Headlight and Wiper System for Automobile Safety (AHAWS). Lane Departure Warning System (LDWS) algorithm receives the video input frame by frame and filters the frame. The edges were detected by the use of sharp edge detection, and the decision of lane detection was made with the Hough transform by means of OpenCV Python. When the vehicle departed from the determined lanes, the driver was warned by a sound. In the case of an integrated system, the AHAWS algorithm had received three parameters, being road curvature, environmental light intensity, and rainfall intensity, provided by LDWS. The headlight's position was adjusted according to the road curvature value, and the light intensity of the headlight was adjusted according to the environmental light and frequency of the windshield wiper. In study [16], a system for checking the headlight position while the vehicle is passing on uphill and downhill roads by the use of an MPU6050 accelerometer sensor and changing the vehicle's light mode according to the light approaching from the opposite direction was designed.

Considering all these studies, the majority of the studies are addressing the issues of adjustment of the headlight's light intensity, lane-keeping system, and headlight position either one by one or two at a time. But, in a night drive, it is required to bring together many features. And the novelty in this study is the design and presentation of the standard headlight technology used in current vehicles with a more sophisticated approach. The current design ensured lighting in the direction of the angle of vision at inclined and curved roads regardless of topography through headlight position control using the flexibility of Light-Emitting Diodes (LED)s in order to create a more secure driving environment under low light conditions, as well as presented compulsory low beam and high beam headlights. In addition, traffic-sign detection technology based on machine learning was integrated into the system in order to eliminate the possibility of non-recognition of the traffic-signs by the driver. The features such as adjustment of the headlight's position to up-down and right-left, vehicle detection, flexibility of the use of low beam as well as high beam without discomforting the driver approaching from the opposite direction, traffic-sign recognition and displaying of it on the driver information display eliminated many factors that may cause accidents. Moreover, the models formed through machine learning for the nighttime vehicle detection and traffic-sign recognition functions directly increased the system's accuracy rate.

2. Material and Methods

2.1. Software and hardware characteristics

The software used was Python 3, and OpenCV was used for real-time image processing, and the Keras library was used to detect objects and form a model trained by datasets. The image processing system was a 2.20 GHz, 8-core Intel Core i7-2670QM CPU. Python software was compiled using the Pycharm IDE. Real-time images for vehicle, traffic-sign, and lane detection were captured by a USB webcam. 2 units of 8x4 LED board was used for the high beam and low beam headlights, and servomotor was used for the movement of adaptive headlights.

2.2. Vehicle Recognition

A vehicle recognition system was formed by means of a model trained with the Common Objects in Context dataset. A dataset covering the visuals of the headlights and taillights was used in order to train this model that can detect the vehicles with a high accuracy rate. The X-coordinate values detected by the vehicle detection model were formed by the Python software. These coordinate values were sent to the Arduino microcontroller over the serial port in order to turn on and off the LEDs of the vehicle's headlights.

All the visuals were tagged with the LabelImg application as seen in Figure 3 in order to form a special object detection model. After tagging the headlights and taillights, these visuals were allocated for training by 80% and for testing by

20%. A good object recognition classifier requires hundreds of images. In addition, the quality of the images is also important for the good learning of the classifier [17-19].

Fig.3. LabelImg application vehicle headlight tagging

2.3. Traffic-sign recognition

Traffic-signs provide valuable information for the drivers and for others using the road. Ignoring them is capable of being fatal. During long trips, the drivers often miss the road safety signs, and thus, many accidents happen due to such negligence. For this reason, a model which would recognize the traffic-signs on a real-time basis based on deep learning methods and which would display the same to the driver via an LCD display was formed. A flowchart of this system is shown in Figure 4.

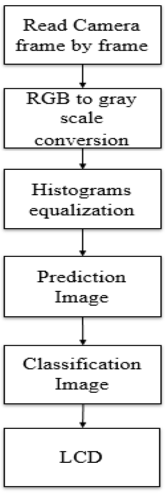

Fig.4. Flowchart of the traffic-sign recognition system

This deep learning model was built using artificial neural networks. The Keras library is a neural network library written in Python, and it was preferred as it supports Convolutional Neural Network (CNN) for computer vision models. The input values are obtained by training a neural network with weight values adjusted during training at hidden layers. After completion of the training, estimations are obtained from the new input values to be given to the model. A basic Artificial Neural Network (ANN) is shown in Figure 5.

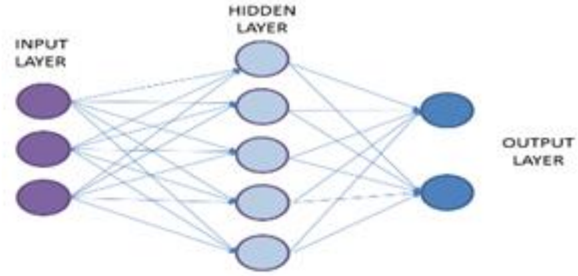

Fig. 5. ANN model

In the model, two different activation functions, Relu and Softmax, were used. The cause of usage of activation functions is to prevent direct conversion of the value forming in the neural networks. If the arising value is sent directly as an output signal, the system comes out to be a linear construct. However, in many cases, the construct is expected to learn the non-linear positions. In such cases, it is required to use the activation functions. Today, the Relu function is frequently used in deep learning models. The Relu function converts the received negative values directly to 0 without performing any process. The function value is given in Equation (1).

$$R_{(z)} = \max(0, z) \tag{1}$$

Where;
Max: Maximum
$R_{(z)}$: Relu function
z: input
For this reason, using the Relu activation function with negative values is not being a correct preference as indicated in Fig. 6 [20].

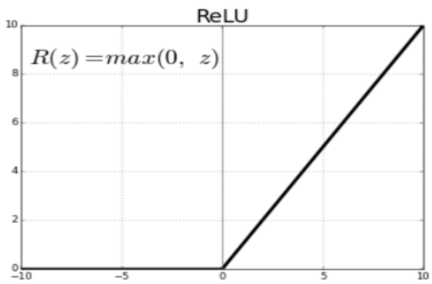

Fig. 6. Relu function's graph

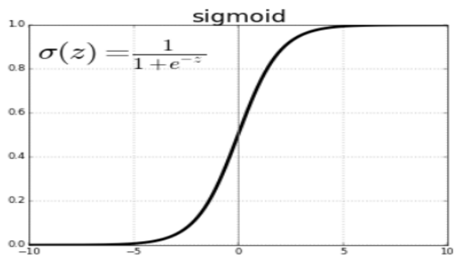

Fig. 7. Sigmoid function graph

In the final layer, the Sigmoid function is used instead of the Relu functions as indicated in Fig. 7 [20].

Sigmoid gets values between 0 and 1, and the values, whose function value correspond to the final layer in Equation (2), got in the operation by this function.

$$\sigma_{(z)} = \frac{1}{e^{-z}+1} \tag{2}$$

Where;

$\sigma_{(z)}$: Sigmoid function value

e^{-z}: Euler input

As a result of the operation in the function, the state of belonging to any class was obtained with values between 0 and 1. This circumstance is particularly useful for classification problems.

2.4. Lane detection

The lane-keeping system estimates the vehicle's steering angle and assists in the adjustment of the headlights' position. The system equipped with a camera is able to perform real-time lane detection by the use of OpenCV library of the Python software language.

2.4.1. Image filtering

The Gaussian Blur filtering is the operation of blurring the image by multiplying the core matrix and the pixels of the image, and the 2D Gaussian function is expressed with Equation (3) [21].

$$G_{(x,y)} = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{3}$$

Where;

$G_{(x,y)}$: 2D Gaussian function

x: horizontal

y: vertical

In the Equation, the x and y values are distances from the center on the horizontal and vertical axes, and the term σ expresses the standard deviation of the Gaussian distribution.

2.4.2 Canny Edge Detection

Canny edge detection algorithm uses four filters to calculate the transverse, vertical, and horizontal edges in the blurry image and rounds up the gradient direction to one of the four angles representing vertical, horizontal, and two diagonal directions (0º, 45º, 90º, and 135º). The gradient of the image is calculated using the formula in Equation (4), and the edge pixels with large changes in gray region values are defined [22]. It was concluded by estimating the first derivative in both vertical direction (G_y) and horizontal direction (G_x) [23]. Thus, the regions in which the gray intensity changes the most were determined with the algorithm. G_x and G_y represent the horizontal and vertical gradients, respectively.

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{4}$$

Where;

G: Gradient

G_x: Horizontal convolution masking vector

G_y: Vertical convolution masking vector

2.4.3. ROI (Region of interest)

ROI means working only on the desired pixel regions in the image. When it is required to detect the lane lines, the detection of numerous objects in the image makes lane detection difficult. Thus, while the vehicle is en-route, scanning the lane lines in a specific region in the frame is more accurate for detection.

2.4.4. Hough transform

The Hough transform is an algorithm developed to detect a form that can be expressed mathematically. For defining a line, the equations of the algorithm are expressed with (5) and (6) [24]. In this study, the detection of a linear form was performed.

$$y = mx + c \tag{5}$$

Where;

m: The slope of the line

The angle θ that the line makes with the origin and consequently its distance to the origin are sufficient for expressing the line.

$$\varrho = x\cos\theta + y\sin\theta \tag{6}$$

Where;

ϱ: The vertical axis value of the point in Hough space

θ: The horizontal axis value of the point in Hough space

This technique scans the θ and ϱ values of all the points on the image. Thus, if these values cohere on the pixels, it is understood that that a line is present on that pixel.

2.4.5. Detection of Lane Lines and Curvature Calculation

It is important to draw a histogram to detect exactly where the lane lines begin. When the histogram graph is examined, it is observed that the white pixels form two different peaks. In this graph, where the right and left lane lines will begin and end is clearly understood. The x coordinate of the histogram forms the x coordinate of the frame on which detection is performed. Thus, the beginning points, required for detecting lane lines on the image, would be obtained. The hot pixels, formed when the pixels on the image are converted to black and white format, are expressed as a second degree polynomial. As the right and left lines on the image are nearly vertical, the pixels have the same x value for more than one y values. For this reason, after obtaining second-degree polynomial coefficients, x values corresponding to y values are calculated by the use of Equation (7) for the values of y between 0 and the image's height.

$$x = f(y) = Ay^2 + By + C \tag{7}$$

Where;
A, B, C: Constant coefficients of the equation expressing the curve
Our function was expressed by Equation (7). By this, the first derivative and second derivative of the f(y) function were obtained. A curve's radius of curvature at a specific point is equal to the radius of the circle being tangential to that point. The radius of the curve [25] is expressed by Equation (8).

$$RC = \frac{\left[1+\left(\frac{dx}{dy}\right)^2\right]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}} \tag{8}$$

$$f(y)' = 2Ay + B \tag{9}$$

$$f(y)'' = 2A \tag{10}$$

Where;
$f(y)$: Representation of the resulting line in the imag
When the terms in the Equations (9) and (10) were placed in the Equation (8), the Equation (11), being the formula of radius of curvature, was obtained. This equation was used in the algorithm in Figure 8 in terms of software.

$$RC = \frac{\left[1+(2Ay+B)^2\right]^{\frac{3}{2}}}{|2A|} \tag{11}$$

Where;
$f(y)$: Representation of the resulting line in the imag

4. Implementation

This study was performed in order to show a more sophisticated approach to the standard headlight technology used in the current automobiles.

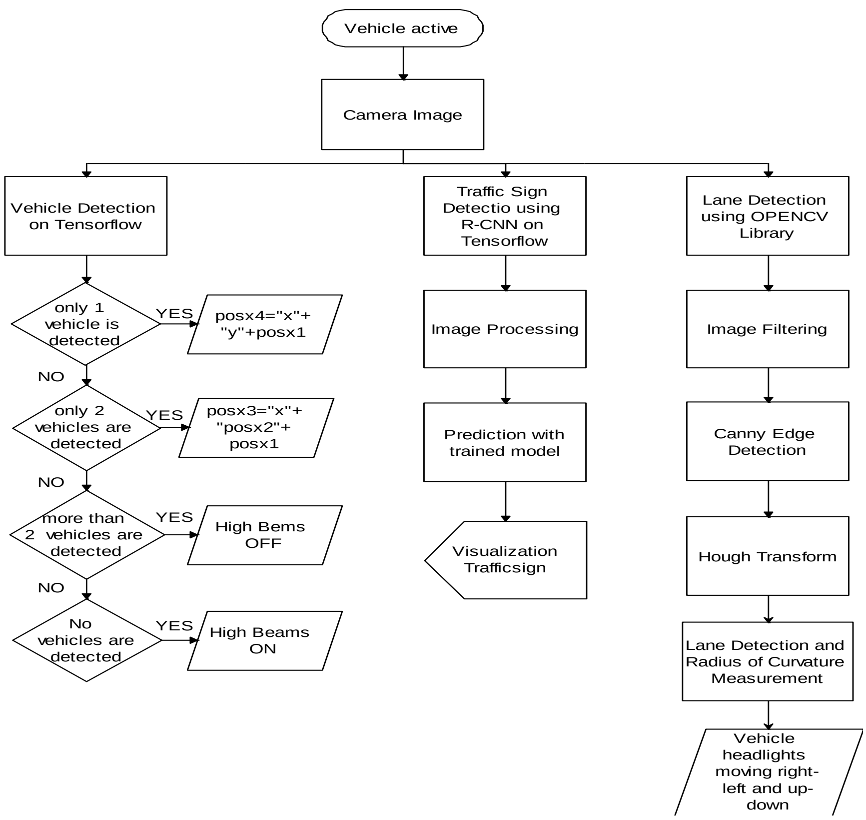

Fig. 8. System's flowchart.

As well as having compulsory low and high beam headlights, the flexibility of LEDs was used to create a more secure driving environment under low lighting conditions. The main objective of this technology is to light the whole road as much as possible without getting into the eyes of other drivers both approaching from the opposite direction and driving in front, and by virtue of the headlights' position control, it ensures lighting in the direction of the angle of vision on inclined and curved roads regardless of the topography. In addition, traffic-sign detection technology based on machine learning was integrated into the system in order to eliminate the possibility of non-recognition of the traffic-signs by the driver. The system includes a webcam. The referred webcam analyzes the road in advance in order to detect the vehicles and provides image flow to a trained algorithm whose main principle is detecting the headlights and taillights. More than 1,000 traffic images shot at nighttime was used to enable this model to detect the vehicles at nighttime. By the object detection model, the x coordinates of the detected vehicles are formed by means of a software. In order to activate the LEDs one by one, these coordinate values are received by the Arduino microcontroller over serial port. An LED module consists of 64 separate chips (4 rows by 16 columns). But 8 columns are used for high beam, and the remaining 8 columns are always turned on for low beam. While the field of the detected vehicles is dynamically kept dark, the remaining fields are lit by high beam headlights. As a result, the driver's field of vision has improved and the risk of an accident has decreased. In the case of the detection of more than two vehicles, the lighting efficiency decreases as the present eight columns will be divided to many parts, and thus, the high beams are completely closed. Another characteristic of the system is traffic-sign recognition based on machine learning. During night rides, the drivers often miss the road safety signs, and they don't care that much. For this reason, a model that would recognize the traffic-signs on a real-time basis based on machine learning and which would display the same on a display was formed. A traffic-sign recognition model was formed as being trained with 34,800 visuals.

4.1. Design of adaptive headlights

In the adaptive headlight system, while driving through a curve, the sensors ensure lighting in a manner suitable to the alignment of the curve considering the vehicle's speed and steering angle. In addition, when it is driven uphill, a standard headlight ensures better lighting. And this makes it difficult for the driver to see the vehicle approaching from the opposite direction. In the adaptive system, the capacity to move the headlight system up and down was added according to the vehicle's position. The adaptive headlight system becomes active only when the vehicle is moving. It was designed so that the system would not be active when the vehicle is stopped and when it is in reverse gear position, even if the steering wheel is moved. The purpose of addressing it in this manner is not to disturb the drivers approaching from the opposite direction. Fig. 9 (a) shows the lighting range of the conventional headlight system, and Fig. 9 (b) shows the lighting range of the adaptive headlight. As is seen, while the conventional headlight system ensures lighting in the direction of the vehicle, the adaptive headlight ensures a lighting range along the direction of the road. As the lighting range is not according to the axis of the road in the conventional headlight system, the beams surpass the road's axis.

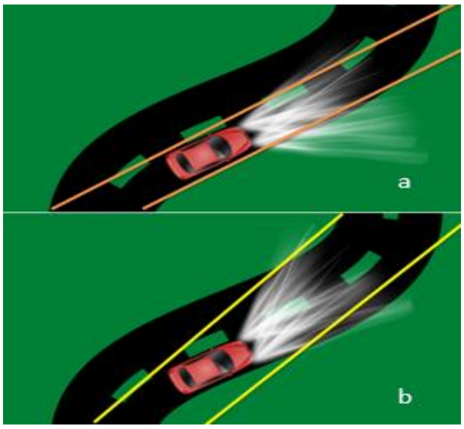

Fig. 9. Lighting ranges of (a) conventional headlight, (b) adaptive headlight

The hardware system of the project's adaptive headlight is shown in Fig 10. The servomotor number 1 turns the headlight system to the right and left with the value received from the steering angle sensor. The servomotor number 2 moves the position of the headlights towards up and down during the uphill and downhill ride of the vehicle. As seen in Fig.10, x-coordinate information of the vehicle or vehicles detected by the algorithm of vehicle recognition from the camera was transmitted to the Arduino microcontroller via serial communication protocol. If the number of vehicles determined by the Arduino microcontroller is 1 or 2, then LEDs as much as the calculated dynamic area are turned off. As the vehicle comes closer, such a dimmed dynamic area changes. For instance, in the case of the detection of more than two vehicles, the LED module of the high beam is completely dimmed. In addition, if no vehicle can be detected,the LED module of the high beam is completely activated.

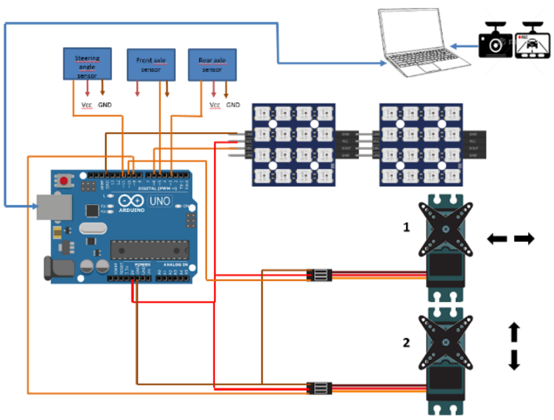

Fig. 10 Hardware system of adaptive headlight

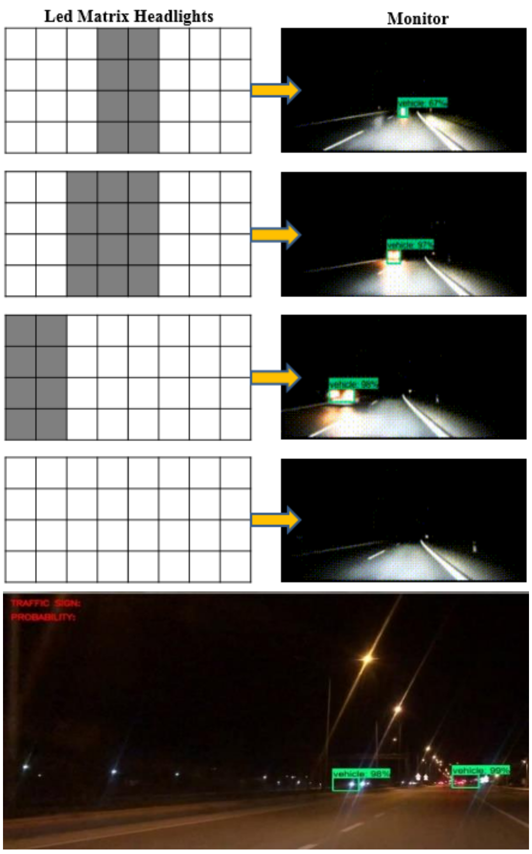

Fig.11. Vehicle recognition and lighting system

4.2. Implementation of vehicle recognition

The X-coordinate values detected by the vehicle detection model were formed by the Python software. These coordinate values were sent to the Arduino microcontroller over the serial port in order to turn on and off the LEDs of the vehicle's headlights. An LED module consists of 2 separate chips of 4 rows and 8 columns. 32 chips were used for high beam, and the other 32 chips were used for low beam.

When a vehicle or vehicles is detected, the area around such vehicles is dynamically kept dark, and the remaining areas are lit by high beams. In Fig. 11, when the vehicle detection algorithm detects more than two vehicles, it sends to the Arduino the information to completely turn off the high beams because when numerous vehicles are detected, the division of the area into more than one part creates an inefficient lighting range.

4.3. Implementation of traffic-sign recognition

As seen in Fig. 12, CNN was used in this deep learning model, and 34,800 visuals of 43 traffic-signs were used as an input dataset, and such an dataset was divided into two as training and test.

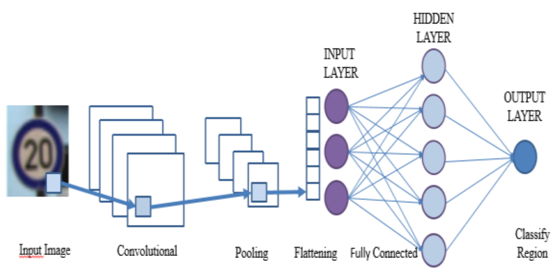

Fig12. Convolutional Neural Network (CNN) construct

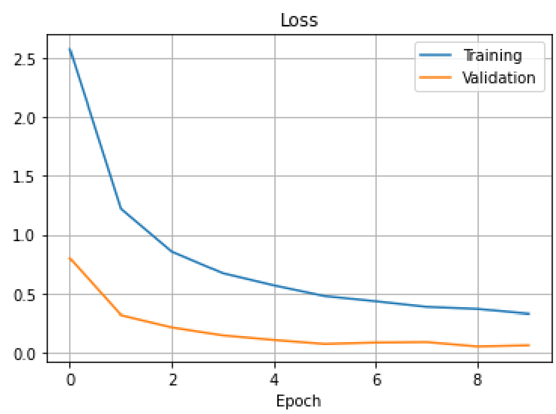

Fig.13. Dataset's training and validation loss graph

20% of this data was allocated for testing during the training of the model, and 20% of the referred 20% was allocated for validation, and the remaining 80% was allocated for training. As a result of the trainings, a 97.3% accuracy rate was obtained, and they are shown in Figures 13 and 14.

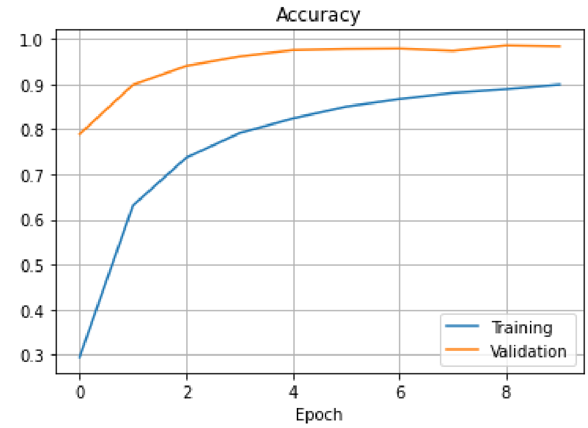

Fig.14. Dataset's training and validation accuracy graph

CNN processes the model image in various layers. A Convolutional Layer was used to determine the characteristics, a Pooling Layer was used to decrease the amount of weight and check the conformity, and a Flattening Layer was used to prepare data for a classic neural network. The frames received by the Convolutional Layer from the camera were first converted to black and white matrix format. And then, such matrices were subjected to a filtering operation, first in the dimension of 5x5, and then in the dimension of 3x3. Thus, the matrices obtained as the result of filtering formed the attribute map. In 2x2 dimension, Max. Pooling Layer was applied to attribute matrices obtained from the dimensions of 5x5 and 3x3. In this model, there were two 2x2 Max. Pooling Layers. Finally, flattening operation was performed in the Flattening Layer for the obtained matrices to be able to be used in the Fully Connected Layer, because the neural network receives the input data from a single dimension series.

4.4. Implementation of lane detection

4.4.1. Image processing process

As seen in Figure 15, first, brightness and contrast were adjusted, the image was converted to grayscale, and the white color was masked in order to perform good edge detection on the image. In order to correctly detect the lane lines, it is necessary to eliminate the noise and details in the image. Thus, the noise and details in the image were eliminated through the application of the Gaussian blur filter.

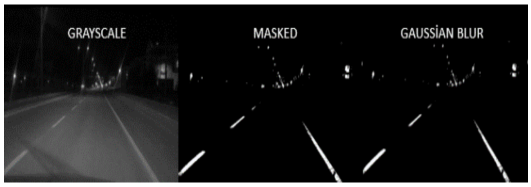

Fig.15. Steps of image filtering

By the use of the lighting in Equation (4) and Figure 8, which is the mathematical basis of Canny edge detection, the implementation output shown in Figure 16 was obtained.

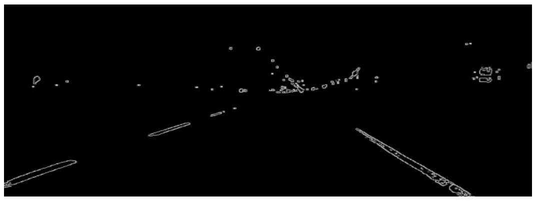

Fig.16. Canny edge detection

After the implementation of the Canny edge detection algorithm, the edges of numerous white-colored objects were detected in the image during the ride. For this reason, in order to remove from the image the edges except the lane lines, a bird's-eye view area was determined in Figure 17 in a manner as the camera's angle of vision would only show the front of the vehicle. Thus, the operation of scanning lines outside the road was prevented.

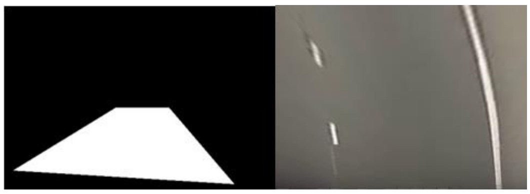

Fig.17. ROI on the left, bird's-eye view on the right

The Hough transform is based on the mathematical expressions in Equations (5) and (6), and Figure 18 was obtained by the use of the algorithm in Figure 8.

Fig.18. Image's output on which Hough transform was applied

Following the above operations, a binary image was obtained. A histogram was applied to the image in order to detect the position of the lane lines on the pixels. Thus, the highest two peaks on the histogram expressed the x position of the lane lines, and the lane lines were scanned on the image by considering such positions as reference.

Sliding windows, placed around the line centers in Figure 19, were used to find and track the lines from the beginning points to the top of the frame.

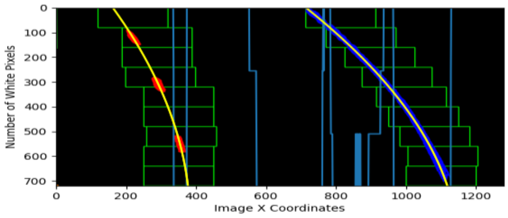

Fig.19. View of sliding windows

After operating the sliding windows function, the general scan function draws a yellow line that correctly tallies with the detected lanes. This line was used to measure the radius of curvature, which is important in estimating the steering angles. This curve was expressed by Equation (7).

As the values obtained following the completion of the above two operations were in pixels, these values were not converted to the unit of meter. The distance between the x-coordinates of the right and left lanes obtained from the histogram provides the width of the lane in pixels. The actual lane width in meters was divided by the lane width, which was obtained from the histogram and which was in pixels, the meter per pixel was then obtained. As seen in Figure 20, in the same manner for the y axis, when the actual lane length was divided by frame height, the meter per pixel was obtained.

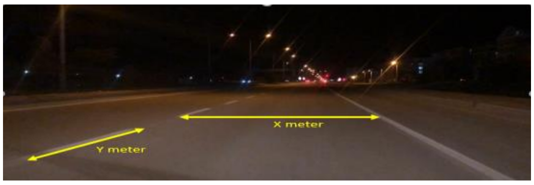

Fig.20. Actual lane width and length


Fig.21. Detection of lane lines

The average meter/pixel value along the x axis is calculated as X/Frame width, and it is calculated as Y/frame height along the y axis.

As a result of all these operation steps being applied, the detection of lane lines is able to be performed at nighttime, as seen in Figure 21.

4. Conclusions

Night rides are challenging and exhausting for the drivers, and as such causes give rise to an instantaneous lack of attention, they increase the risk of accident. In this study, a recognition, detection and warning system that increases the drivers' area of vision at nighttime rides and that eliminates their instantaneous absence was designed. These technologies carry out their tasks autonomously via software. High and low beam headlights were equipped with LED technology, and each cell was controlled according to recognition and detection algorithms. A dataset was formed from the relevant visuals for the performance of vehicle detection at nighttime, and these visuals were used in forming a model detecting objects by means of Keras. The control of each cell provided an advantage in eliminating the drivers' instantaneous blindness to the beams of vehicles approaching from the opposite direction. On inclined and curved roads, the headlight system designed is able to turn to right and left, and up and down, by means of servomotors in order to ensure lighting in the direction of vision. Another problem at night, besides the angle of vision, is the inability to clearly see the traffic-signs, or missing of such traffic-signs due to attention deficit and exhaustion. The camera detecting the vehicle approaching from the opposite direction at night detects the traffic-signs on the roadside by means of a model designed with the deep learning method and warns the driver via the display in the vehicle. This model designed was trained with 34,800 visuals and tested. As a result of trainings, a 97% accuracy rate was obtained.

CRediT authorship contribution statement

Declaration of Competing Interest

5. References

1. ScienceDaily, "Traffic accidents, Avoiding Rear-end Collisions", Online available: (24.08.2022), http://www.sciencedaily.com

2. Vorndran, I., "Unfallgeschehen im Straenverkehr 2006", Wirtschaft und Statistik

2007, Statistisches Bundesamt, Wiesbaden, Vol. 7, 2007, pp.679-687.

3. Lee, YM., Sheppard, E., "The effect of lighting conditions and use of headlights on drivers' perception and appraisal of approaching vehicles at junctions", Ergonomics, Vol. 61, No. 3,2018, pp. 444-455, https://doi.org/10.1080/00140139.2017.1364790

4. Balaji, RD, "A Case study on Automatic Smart Headlight System for Accident Avoidance", Int. J. Comput. Commun. Informatics, Vol. 2 Iss. 1, 2020, pp.70–77, https://doi.org/10.34256/ijcci2012

5. Romansky, ML., Plummer, RW., Neumann, ES. "Environmental Stressors as Causal Factors for Driver Fatigue", Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 1979, pp.452–455, https://doi.org/10.1177/107118137902300169

6. Adrian, J., Hue, D., Porte, S., Brun, JL., "Validation of the driver ecological glare test", Journal of Safety Research, Journal of Safety Research 72, 2020, pp.139–143

7. Büthorn, R., Tadjine, H.H., Auerbach, B., Schulze, K., "Advanced Headlight System", 3D High Beam, 2013, pp. 77–88, https://doi.org/10.1007/978-3-319-00476-1_8.

8. Reagan, IJ., Frischmann, T., Brumbelow , ML., "Test track evaluation of headlight glare associated with adaptive curve HID, fixed HID, and fixed halogen low beam headlights" Ergonomics, Vol. 59, No. 12, 2016, pp.1586–1595, http://dx.doi.org/10.1080/00140139.2016.1154986

9. Gil, PA., Santos-Berbel, CD., Castro, M., "Driver glare exposure with different vehicle frontlighting systems", Journal of Safety Research, Volume 76, February 2021, pp.228-237, https://doi.org/10.1016/j.jsr.2020.12.018

10. Chen, YS., Chiu, SC., "New method of automatic control for vehicle headlights", Optik 157, 2018, pp.718-723, https://doi.org/10.1016/j.ijleo.2017.11.137.

11. Chen, Y.L., Chiang, C.Y., "Embedded vision-based nighttime driver assistance system, IEEE, International Symposium on Computer, Communication, Control and Automation, 2010, pp. 199–203, https://doi.org/10.1109/3CA.2010.5533586.

12. Connell, JH., Herta, BW., Pankanti, S., Hess, H., Pliefke, S., "A fast and robust intelligent headlight controller for vehicles, in: IEEE Intell. Veh. Symp.,Proc., 2011: pp. 703–708, https://doi.org/10.1109/IVS.2011.5940492.

13. Tripathy, AK., Kayande, D. , George,J., John, J., Jose, B., "Wi lights - A wireless solution to control headlight intensity", in: Proc. - Int. Conf. Technol. Sustain. Dev. ICTSD 2015, Institute of Electrical and Electronics Engineers Inc., 2015, https://doi.org/10.1109/ICTSD.2015.7095866.

14. Mayr, J., Giracoglu , C., Unger, C., Tombari , F., "Headlight Range Estimation for Autonomous Driving using Deep Neural Networks", 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France. June 9-12, 2019, pp.2251-2258. https://doi.org/10.0/Linux-x86_64.

15. Vijay, G., Ramanarayan, MN., Arunkumar PC., "Design and Integration of Lane Departure Warning, Adaptive Headlight and Wiper system for Automobile Safety", IEEE, 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT-2019), May 17th & 18th 2019

16. Muhammad, F., Yanto, DD., Martiningsih, W., Noverli, V., Wiryadinata, R., "Design of Automatic Headlight System Based on Road Contour and Beam from Other Headlights", Proceeding - 2020 2nd Int. Conf. Ind. Electr. Electron. ICIEE 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 112–115, https://doi.org/10.1109/ICIEE49813.2020.9276906.

17. Görmer, S., Müller, D., Hold, S., Meuter, M., Kummert, A., "Vehicle Recognition and TTC Estimation at Night based on Spotlight Pairing", Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, October 3-7, 2009, pp.196-201

18. Chen, YL., Chen, YH., Chen, CJ, Wu, BF., "Nighttime Vehicle Detection for Driver Assistance and Autonomous Vehicles", IEEE, Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), 2006, pp.1051-4651, https://doi.org/10.1109/ICPR.2006.858.

19. Connell, JH., Herta, BW., Pankanti, S.,

Hess, H., Pliefke, S., “A Fast and Robust Intelligent Headlight Controller for Vehicles” IEEE Intelligent Vehicles Symposium (IV), 2011, pp.703-708, https://doi.org/10.1109/IVS.2011.5940492.
20. Activation Functions in Neural Networks, Online available: (20.08.2022), https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
21. Andrei, MA., Boiangiu, CA., Tarbă, N., Voncilă, ML., “Robust lane detection and tracking algorithm for steering assist systems”, Machines. 10, 2022, https://doi.org/10.3390/machines1001001.
22. Sekehravani, EA., Babulak, E., Masoodi, M., “Implementing canny edge detection algorithm for noisy image”, Bull. Electr. Eng. Informatics. 9, 2020, pp. 404–1410, https://doi.org/10.11591/eei.v9i4.1837.
23. Wolf, C., Jolion, JM., Chassaing, F., “Text localization, enhancement and binarization in multimedia documents,” IEEE, 2002 International Conference on Pattern Recognition, Quebec City, Quebec, Canada, vol. 2, 2002, pp. 1037-1040, https://doi.org/10.1109/ICPR.2002.1048482
24. Fokkinga, M., “The Hough transform”, J. Funct. Program. 21, 2011, pp.129–133, https://doi.org/10.1017/S0956796810000341.
25. Radius of curvatur, Online available: (24.08.2022), https://www.intmath.com/applications-differentiation/8-radius-curvature.php.