Smart Driving Assistance Using Deep Learning

S. N. Baba Shankar, B. Karthik Reddy, B. Koushik Reddy,
Venuthurla Venkata Pradeep Reddy, and H. B. Mahesh(✉)

Department of Computer Science, PES University, Bengaluru, India
hbmahesh@pes.edu

Abstract. This research paper presents a smart driving assistance system that utilizes deep learning for lane detection and departure alert, traffic sign detection, and voice alerts. The system uses a combination of computer vision techniques and neural networks to detect lanes and traffic signs in real-time. The lane departure alert feature alerts the driver when the vehicle begins to drift out of its lane, and the traffic sign detection feature identifies and alerts the driver of any traffic signs that are relevant to the current road. The voice alert feature provides an additional layer of safety by audibly alerting the driver of any detected traffic signs. The proposed system has been evaluated on a dataset of real-world driving scenarios and has shown promising results in terms of accuracy and efficiency.

Keywords: Smart Driving Assistance · Deep Learning Lane Detection · Traffic Sign Detection · Voice Alerts Autonomous Vehicles · Convolutional Neural Networks (CNNs) · Image and Video Analysis · Computer Vision Real-time Alerts · Lane Departure Warning · Lane Marking Detection · Lane Curvature Detection · Traffic Sign Identification Perspective Mapping · Recurrent Neural Networks (RNN) · Long Short-Term Memory (LSTM) · Object Detection Classification · Driver Safety Obstacle · Detection Navigation Systems · Adaptive Cruise Control · Collision Avoidance System

1 Introduction

The field of autonomous vehicles has seen rapid growth in recent years, with advancements in technology allowing for increased road safety and efficiency. One important aspect of autonomous driving is detecting and responding to various road features and obstacles. One such feature is lane detection, which allows the vehicle to stay within the designated lanes on the road. Another important aspect is traffic sign detection, which enables the vehicle to recognize and respond to traffic signs such as speed limits or stop signs.

One approach to achieving these goals is the use of deep learning algorithms. These algorithms can learn and recognize patterns in large amounts of data, making them well-suited for tasks such as image and video analysis. In this research, we propose the use of deep learning techniques for the development of a smart driving assistance system that includes lane detection and departure alert, traffic sign detection, and voice alerts regarding traffic signs detected.

S. Aurelia et al. (Eds.): ICCSST 2023, CCIS 1973, pp. 402–418, 2024.
https://doi.org/10.1007/978-3-031-50993-3_32

Our system is designed to improve safety and efficiency on the roads by providing real-time alerts and assistance to drivers. The lane detection and departure alert feature helps to prevent accidents caused by drifting out of the lane, while the traffic sign detection feature ensures that the driver is aware of and compliant with all relevant traffic laws.

Additionally, the system includes voice alerts that notify the driver of any traffic signs detected, further improving their awareness of their surroundings.

We will be using deep learning techniques for image and video analysis, such as Convolutional Neural Networks (CNNs) to train our model for detecting traffic signs and lanes. The data for training will be collected from cameras mounted on the vehicle and will include a variety of different lighting and weather conditions to ensure robust performance in real-world scenarios.

In this research, we will also discuss the evaluation methods and metrics we will be using to evaluate the performance of our system and the challenges that need to be addressed for the successful implementation of our smart driving assistance system**.**

2 Literature Survey

[1] proposed a new approach to traffic sign detection using convolutional neural networks (CNNs). The authors use the Hue and Saturation of pixels to extract the sign area from an image, with a focus on the red border of the sign. The model achieves high accuracy and efficiency but requires more GPU resources for training, and real-time implementation is not yet available.

[2] present a method for robust lane marking detection using boundary-based inverse perspective mapping. The authors process video data to convert color to grayscale and then to black and white using binarization.

They then calculate the region of interest for road lines using edge detectors and Hough transformations. The offset between the car and the lane lines is calculated and displayed on the screen, with green indicating no offset and red indicating an offset greater than 0.

[3] proposed a lane detection method that employs key point estimation and point instance segmentation. The authors input a 512×256-pixel video frame into the network, compress it to reduce computation, and classify pixels by class. The method achieves accurate detection of curved lanes, but the added computation of classifying all pixels in the image is a drawback.

[4] assess the performance of MDEffNet on an Indian dataset using various preprocessing methods, including rescaling images to 48×48, extracting the region of interest, converting to HSV, and applying histogram equalization to the V channel. The authors select hyperparameters such as 3×3 max-pooling for CNN and kernel size for MDEffNet through experimentation.

[5] proposed a traffic sign identification technique that uses deep learning and image processing. The technique involves preprocessing images to highlight important details, localizing signs using the Hough Transform, and classifying them using a CNN, achieving high accuracy in recognizing circular traffic signs.

Advantages include its high recognition rate and ability to perform various computer vision tasks. However, limitations may include the need for large training datasets, the potential for overfitting, and reduced performance in adverse weather conditions or when signs are obscured.

[6] proposed Perspective Mapping (IPM), a technique used in autonomous navigation to accurately detect lane curvature. Gaussian filters are used to identify lane markings and the IPM technique converts the view to a top-level view for accurate lane detection. Line detection algorithms, such as canny edge detection and hough transformations, are used to detect lane lines, and the curvature of the road is calculated using the IPM-detected lane lines. While the technique provides a clear idea of road curvature and increases prediction accuracy, it may not be accurate in hilly areas or on steep roads.

[7] proposed a model that utilizes Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) for video processing using the TUSimple dataset. The model includes an Encoder CNN, a Decoder CNN, and an LSTM network that sequentially processes the frames to extract features and learn from them. The model outperforms baseline architectures that use a single image as input, but the network requires high computational power to process the images.

3 Datasets

The CULane dataset, a large-scale dataset for scholarly work on traffic lane recognition, was used for the feature of lane detection. The dataset was collected using cameras installed on six different cars operated by various drivers in Beijing. The dataset is divided into three sets: a training set of 88,880 images, a validation set of 9,675 images, and a test set of 34,680 images. The test set is further divided into 8 categories, comprising challenging and normal scenarios [8].

A custom dataset was created using publicly available German Traffic signs, with f our classes: prohibitory, dangerous, mandatory, and other. The dataset was augmented by enlarging images, adjusting brightness, and adding noise to improve the training of the model and increase the accuracy of the results. The dataset contains 43 different traffic signs.

4 Methodology

Frame by frame, the footage is processed. Every frame is given as an input, and from every frame, information about lanes and traffic signs is retrieved and shown on the dashboard (Fig. 1).

4.1 Lane Detection

The lane detection model uses a novel row-based selection technique that makes use of global picture characteristics as part of its lane identification strategy. A 3-D tensor of

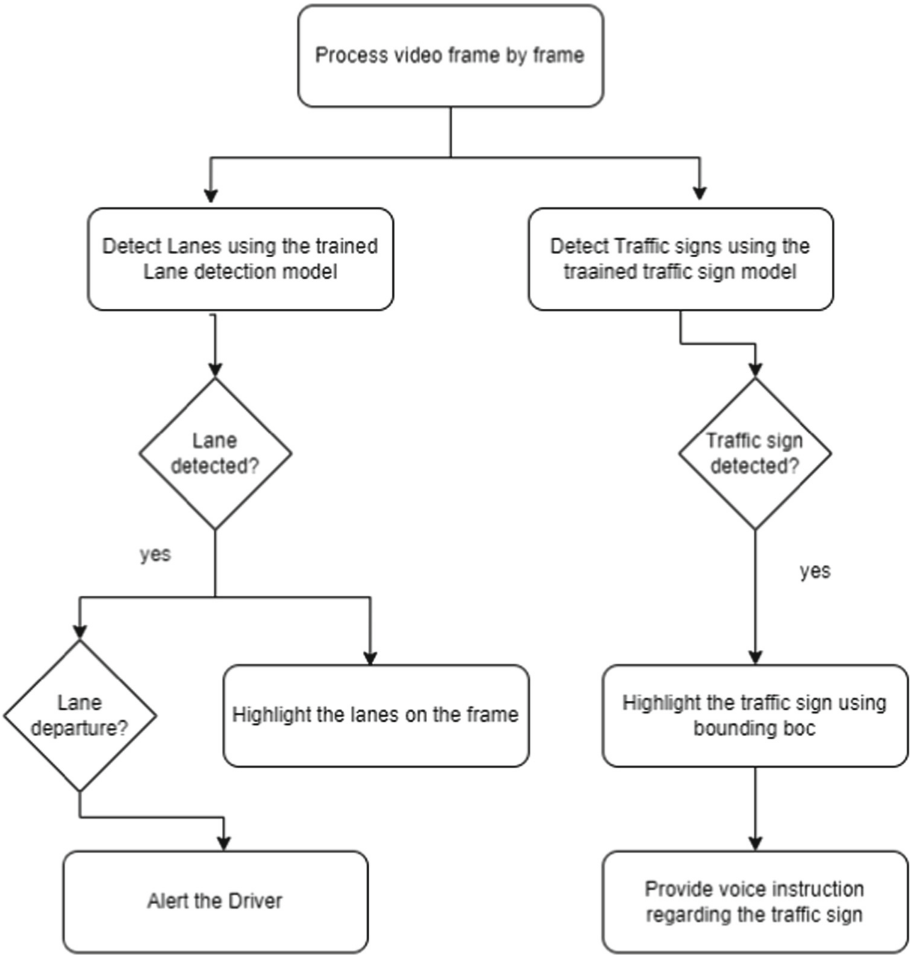

Process video frame by frame
Detect Lanes using the trained Lane detection model
Detect Traffic signs using the traained traffic sign model
Lane detected?
Traffic sign detected?
yes
yes
Lane departure?
Highlight the lanes on the frame
Highlight the traffic sign using bounding boc
Alert the Driver
Provide voice instruction regarding the traffic sign

Fig. 1. The flow of activities in the project

size C H W, where C, H, and W stand for the number of channels, rows, and columns, respectively, serves as the model's input. The first slice of this tensor is subjected to the convolution layer, which consists of C kernels of size C x w, where w is the kernel width. Then, this tensor is used to produce H slices.

Unlike standard convolutional neural networks, which simply relay the output of the convolution layer onto the following layer (CNNs). Instead, a new slice is created by combining it with the following tensor slice and sending it to the following convolution layer. The softmax layer, the top layer of the model, is where this procedure is carried out to calculate the likelihood that each pixel in the C, H, and W tensors corresponds to a detected lane anchor.

Each frame from the video feed is processed by various methods of the lane detector class. A method from the lane detector model processes each frame to produce the output.

The first step in the process involves passing the input frame to a custom function, which applies some preprocessing on the input image using the other function which involves converting the image to RGB format and then converting it into a numpy array for ease of processing. The image is then resized to a specific shape and normalized to reduce skewness.

The transformed image is then sent to the pre-trained model, which returns the output in the form of numpy arrays containing the probabilities of the pixels in the image. These arrays are then passed to a method, which processes the probabilities received from the model to form the lanes from the detected points in each row of the image as the anchors.

The row-based selection approach of the lane detection model is based on global picture attributes. In other words, depending on global characteristics, the model chooses the proper lanes for each preset row. In order to depict lanes as a collection of horizontal places, preconfigured rows and row anchors are both used. Gridding is the initial stage in depicting a location. The location is separated into several cells on each row anchor. This makes it possible to compare picking certain cells over particular row anchors to picking lanes.

To account for the form, it is required to determine where the lane is on each row anchor. In order to determine locations from the categorization forecast, it makes sense to find the greatest response peak. For each lane index i and row anchor index j, the position Loci,j can be stated in the manner shown below:

$$\mathrm{Loc}_{i,j} = \mathrm{argmax}_k P_{i,j,k} \text{ s.t. } k \in [1, w] \tag{1}$$

This method enables the model to accurately detect the lane positions on the road, which allows for the marking of the lanes, coloring them, and also finding the curvature of the road using the center points of the bottom and topmost lanes. Additionally, the model can detect if the car has switched lanes by storing a previous center value and comparing it with the current center.

Overall, the lane detection model's innovative approach to lane detection has great potential in improving driving safety and reducing the incidence of accidents caused by drivers drifting out of their lanes.

Lane Structural Loss: To represent the position relations of lane points, we employ loss functions. One of these functions states that as lanes are continuous, neighbouring row anchors' lane points ought to be near to one another. Our solution preserves the continuity constraint while utilising a classification vector to determine the position of the lane by confining the spread of classification vectors within the closed row anchors.

$$\begin{aligned} L_{shp} = &\sum_{i=1}^{C}\sum_{j=1}^{h-2} \| (Lo\, c_{i,j} - Lo\, c_{i,j+1}) - \\ &(Lo\, c_{i,j+1} - Lo\, c_{i,j+2}) \| \end{aligned} \tag{2}$$

Data Augmentation: A method of augmentation using rotation, vertical shift, and horizontal shift is used to avoid the problem of over-fitting in a classification-based net- work caused by lane structure. In addition, the lane is extended to the picture boundary to preserve the lane structure and improve performance on the validation set.

Upon running the lane detection model, the system will display the detected lanes that the vehicle is currently traveling on in real-time on the screen. The output will provide the coordinates of points on the lanes, which can be used to generate a visual representation of the lanes in the form of highlighted points or dots on the current frame.

These dots can then be used to draw a solid line that accurately represents the position of the detected lanes in the scene.

In addition to identifying the current lanes, the model may also provide useful information such as the width and curvature of the lanes. This information can be used to infer the vehicle's position on the road and help with decision- making, such as lane changes or automatic steering. The system can also track the movement of the detected lanes over time and predict their future positions, providing a useful tool for autonomous driving or driver assistance systems.

4.2 Lane Departure Alert

The objective of this module is to provide an effective and reliable warning to the driver when the car departs from its current lane. The user has the option to enable or disable this function.

To calculate the center of the current lane, this module measures the distance between the left and right lanes the vehicle has traveled in. A significant variation in the center between the succeeding frames indicates that the automobile is leaving the current lane. This triggers a beep alarm to advise the driver of the lane departure. After conducting several tests, a threshold of 200 was determined to be the optimal difference between centers for detecting lane departures. If (previousCentre-presentCentre)>=200 implies that the vehicle has departed the lane (Fig. 2).

During testing, it was observed that the video processing was running behind when the warning was generated. To address this issue, the module was designed using threading to enable video processing and alert creation to happen simultaneously and separately. This design ensures that the warning is generated in real-time and does not cause any delay in the video processing.

The module also includes a feature for providing driving instructions to the driver. This feature detects the curvature of the road by locating the center of the lane between the starting and finishing positions. The deviation between the two centers is used to adjust the driving instructions based on the road curvature. Specifically, if the difference between the top center and bottom center is greater than 180, there is a right curve ahead and a left turn, and if none of these conditions are met, the road is straight (Fig. 3).

In conclusion, this module provides an efficient and reliable lane departure warning system that enhances road safety.

The addition of the driving instructions feature makes it even more valuable to drivers by providing them with real- time information about the road ahead.

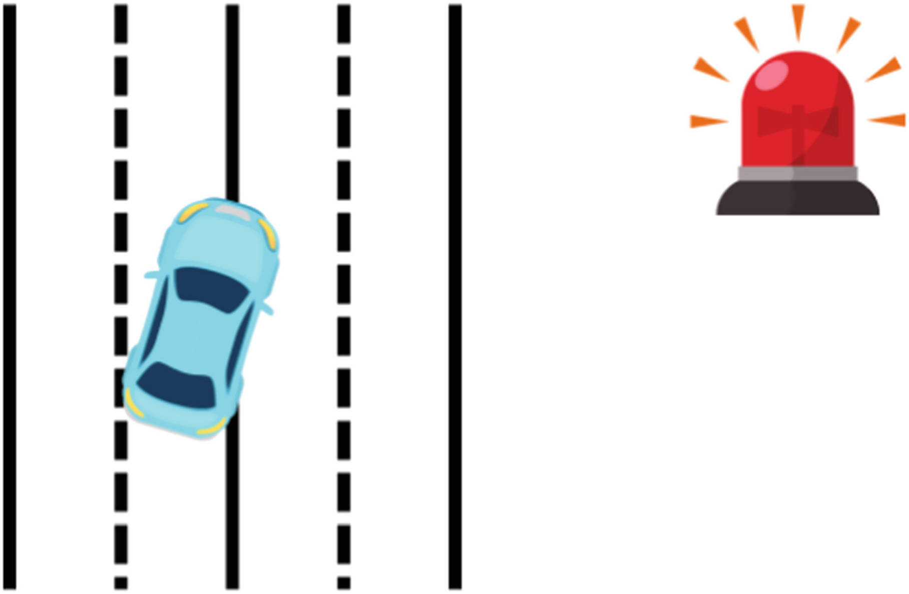
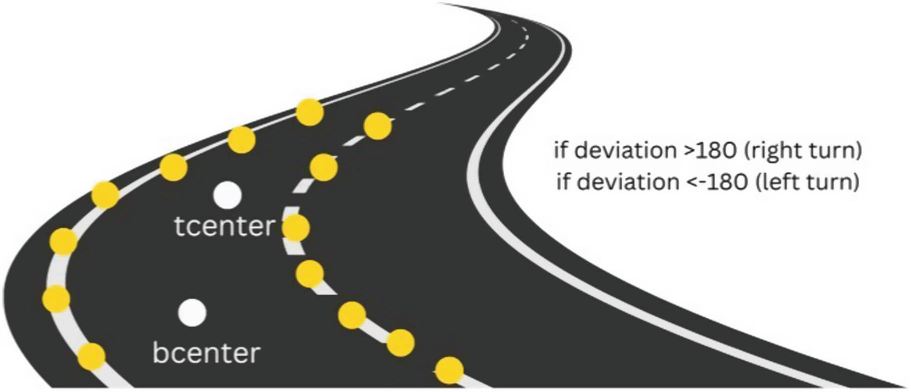


Fig. 2. Showcasing the lane departure alert system



if deviation >180 (right turn)
if deviation <-180 (left turn)
tcenter
bcenter

Fig. 3. Representation of lane centers

4.3 Traffic Sign Detection

The traffic sign and lane detection system involves processing each frame of the video using trained models. The overall methodology is as follows:

- Extract frames from the video.
- For traffic sign detection, feed each frame to the trained model, which divides the frame into possible regions and classifies each region as a traffic sign. A confidence threshold of 50% is set for traffic sign classification, and only the even- numbered frames are processed to speed up the processing of the video.
- For lane detection, apply various algorithms such as canny edge detection, hough line transformations, and ResNet 18 to detect the lane, mark the region, and calculate the curvature of the road.

1) YOLO Model: The general workflow of the YOLO (You Only Look Once) model involves dividing the input frame into grids of equal size, labeling objects in the grids to a particular class based on probability, and drawing a bounding box over the grids of similar classes to locate the object correctly.
2) CNN Model: To identify 43 different categories, a CNN (Convolutional Neural Network) model with four dense layers and five convolutional layers, each with a RELU activation function, is created using Keras and TensorFlow. The softmax activation function is present in the output layer. In order to categorise things with multiple labels, YOLOv3 employs binary cross-entropy loss. The chance that each label the identified item falls within is returned by this technique. An imbalanced bespoke dataset of 43 classes is used to train the model. The confidence threshold is established at a number that is neither too high nor too low to prevent overfitting and underfitting.
3) Data: Models have been trained using the YOLO algorithm and the German traffic dataset with some customization. If the user enables the traffic sign feature in the frontend, the traffic sign detection system will be enabled.
4) Output: Each item located in the bounding boxes is given a predicted label and a confidence score by the YOLO model. The outcome of each surrounding box is the label with the highest score. The bounding boxes are eliminated if the confidence score is less than the user-defined threshold. The algorithm highlights the traffic signs with bounding boxes and labels them with the type of sign it has determined in order to convey its level of confidence (Figs. 4 and 5).

Traffic Sign Voice Alert

This module serves as a critical component in the overall system's objective to alert drivers when a traffic sign is detected on the road. As such, it receives the label of the class that was identified by the traffic sign detection module as input to inform the user. To ensure efficient and accurate communication, each label of the traffic sign is stored in a file and assigned a unique id. The label-to-id mapping is then stored in a dictionary for easy reference. This allows for quick and easy retrieval of the appropriate audio file corresponding to the detected traffic sign, ensuring a seamless and efficient communication process between the module and the driver.

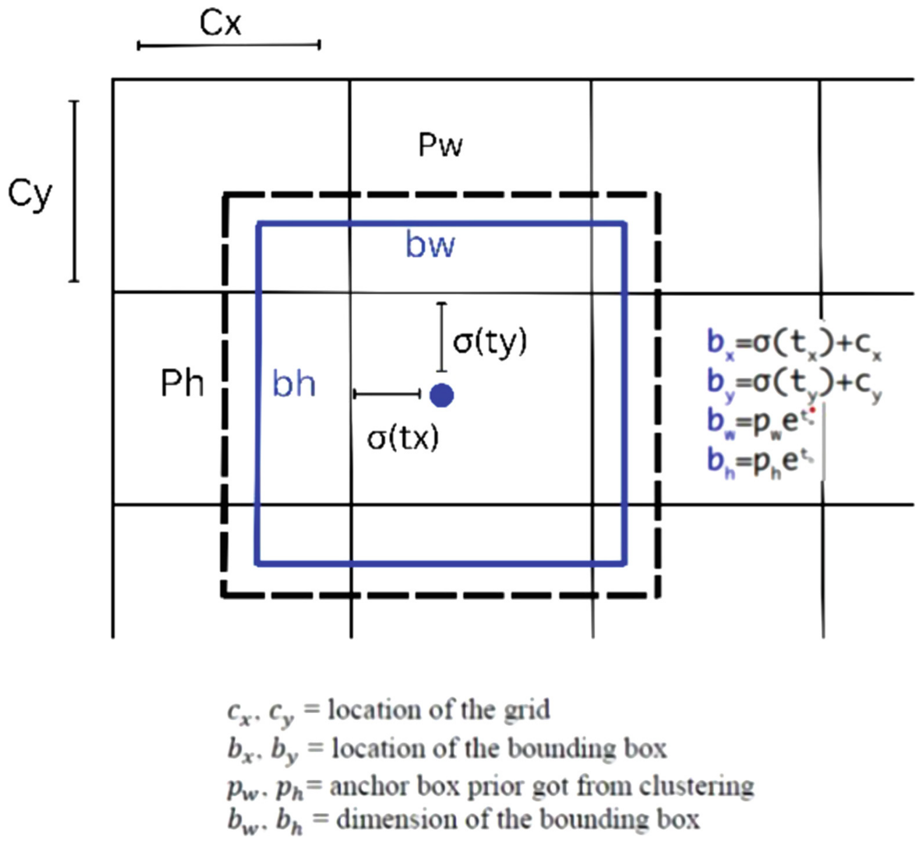

Cx
Cy
Pw
bw
Ph
bh
σ(ty)
σ(tx)
$b_x=\sigma(t_x)+c_x$
$b_y=\sigma(t_y)+c_y$
$b_w=p_we^{t_w}$
$b_h=p_he^{t_h}$
c_x, c_y = location of the grid
b_x, b_y = location of the bounding box
p_w, p_h= anchor box prior got from clustering
b_w, b_h = dimension of the bounding box

Fig. 4. Representation of bounding boxes around the detected object

One important consideration is the possibility of detecting the same traffic sign repeatedly in consecutive frames, which can lead to unnecessary and repetitive voice instructions. To address this, the module has been designed to ensure that instructions about a traffic sign are only given once and that subsequent instructions are different from the preceding one. This helps to reduce driver distraction and improve overall system efficiency.

To further improve accuracy, the voice instruction is only given if the traffic sign is detected in at least 3 consecutive frames. This ensures that the detected sign is consistent and not a false positive or a temporary artifact in the video feed.

However, it has been observed that generating voice instructions can significantly increase the processing time, particularly when the video feed is large and high resolution. To overcome this challenge, the module has been designed to utilize threads, allowing voice instruction creation and traffic sign recognition to be performed simultaneously.

This helps to optimize system performance and ensure timely communication with the driver.

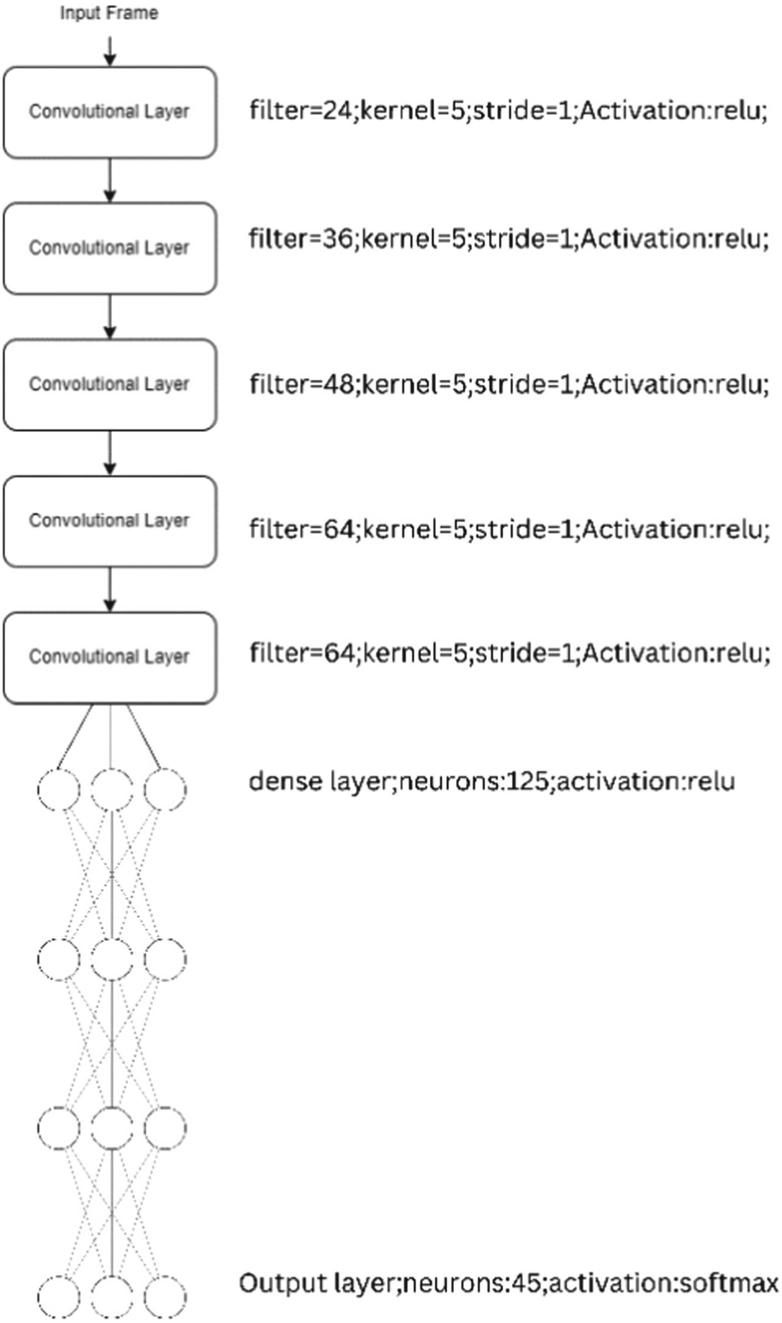

Input Frame
Convolutional Layer
filter=24;kernel=5;stride=1;Activation:relu;
Convolutional Layer
filter=36;kernel=5;stride=1;Activation:relu;
Convolutional Layer
filter=48;kernel=5;stride=1;Activation:relu;
Convolutional Layer
filter=64;kernel=5;stride=1;Activation:relu;
Convolutional Layer
filter=64;kernel=5;stride=1;Activation:relu;
dense layer;neurons:125;activation:relu
Output layer;neurons:45;activation:softmax

Fig. 5. The architecture used for object detection and classification

Overall, this module plays a critical role in the system's objective of ensuring driver safety by providing timely and accurate instructions about detected traffic signs on the road. By everaging various techniques such as mapping unique ids to traffic sign labels, limiting repetitive instructions, and utilizing threads for efficient processing, the module helps to improve the overall effectiveness and efficiency of the system.

4.4 Results

The implementation of deep learning algorithms for lane detection, lane departure alert, traffic sign detection, and voice alerts in a smart driving assistance system has produced positive results. The following are the key findings from the study:

4.5 Lane Detection

When the traffic lanes are graphically represented as dots, the two lanes the motorist is using will be highlighted. The accuracy rates on the training dataset and testing dataset for the neural network, which was trained using the CUlane dataset, were 93% and 88%, respectively. The lane departure warning feature has been developed effectively, and the model can perform well in all traffic and weather circumstances (Fig. 6).

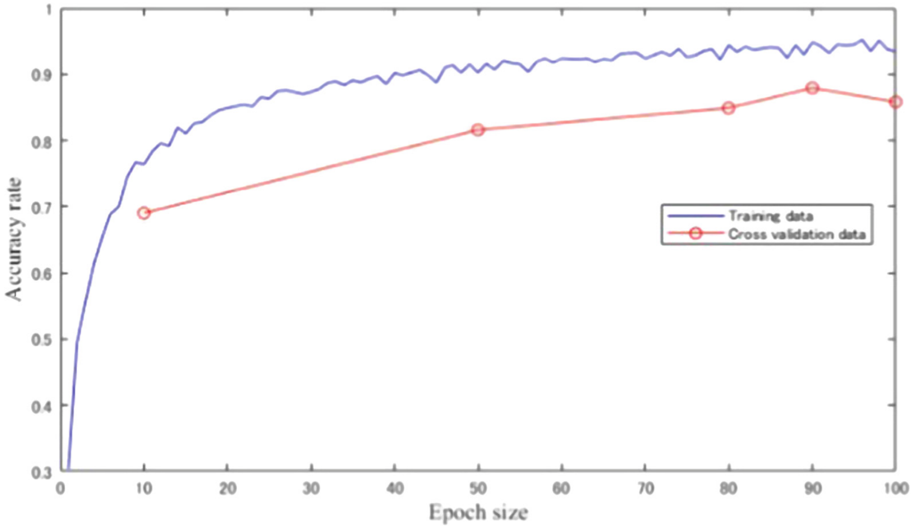


Fig. 6. Training and Testing accuracy of Lane detection model

4.6 Lane Departure Alert

The lane departure alert system was able to effectively alert the driver when the vehicle was deviating from its lane. This was demonstrated through various tests, where the system accurately alerted the driver when the vehicle started to stray from its lane. The alert system was also found to have a low rate of false alarms, which is an important factor in ensuring that the driver is not constantly disturbed by unnecessary alerts.

4.7 Traffic Sign Detection

The YOLO model for traffic sign detection was able to detect and classify different traffic signs with a high level of accuracy. The model was trained on a large dataset of traffic sign images, and it was found to be able to detect and classify traffic signs with an accuracy of over 90%. This is an important capability for a smart driving assistance system, as it enables the driver to quickly identify and respond to road signs and regulations.

4.8 Voice Alerts

The voice alert system successfully informed the driver of the detected traffic sign. The voice alert was implemented using text-to-speech software, and it was found to be effective in communicating the detected traffic sign to the driver in a clear and concise manner. This can help to reduce the amount of time that the driver spends looking away from the road, and it can also help to ensure that the driver is aware of important road information, such as speed limits and road conditions (Fig. 7).

Here is the sample outcome of the proposed system in (Fig. 11).

5 Discussion

The use of deep learning algorithms in a smart driving assistance system has shown promising results in enhancing driving safety. This technology employs a combination of advanced features such as lane detection, lane departure alert, traffic sign detection, and voice alert to create an integrated system that can significantly improve the overall driving experience (Fig. 8).

One of the key benefits of this technology is the ability to detect and alert drivers of potential lane departures, which is a leading cause of road accidents. The lane departure alert system utilizes deep learning algorithms to detect the vehicle's position relative to the lane markers and provides audible and visual alerts if the vehicle deviates from its intended path.

Another important feature is traffic sign detection, which uses deep learning algorithms to recognize and interpret various road signs such as speed limits, stop signs, and pedestrian crossings. This system provides timely alerts to the driver, ensuring they are aware of any road hazards and adhere to traffic regulations.

Moreover, voice alert systems are integrated into the technology to provide drivers with real-time notifications on the driving conditions, allowing them to make informed

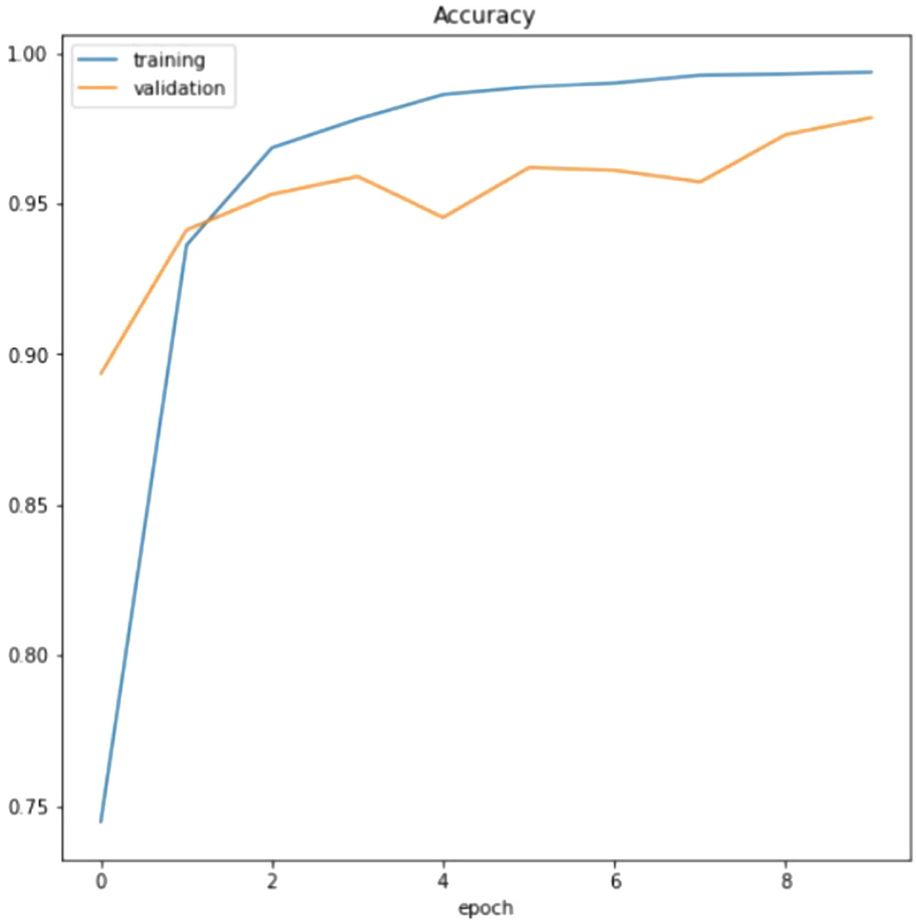


Fig. 7. Accuracy of the YOLO v4 model

decisions on the road. This feature can be particularly useful for long-distance drivers, reducing fatigue and improving their concentration levels (Figs. 9 and 10).

While the results of this study are encouraging, there is still room for improvement in the technology. For instance, incorporating obstacle detection and avoidance into the lane detection algorithm would provide even more comprehensive driving assistance, reducing the risk of collisions. Additionally, advanced deep learning models, such as convolutional neural networks, could be used to further increase the accuracy of the traffic sign detection system.

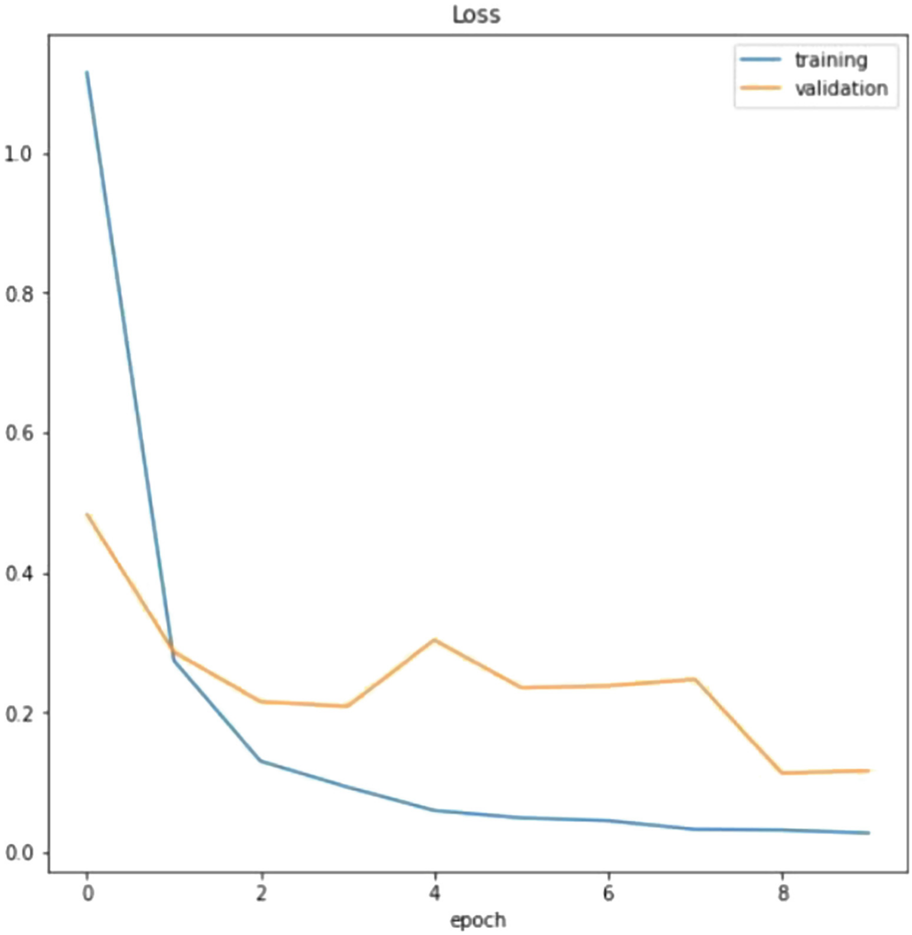
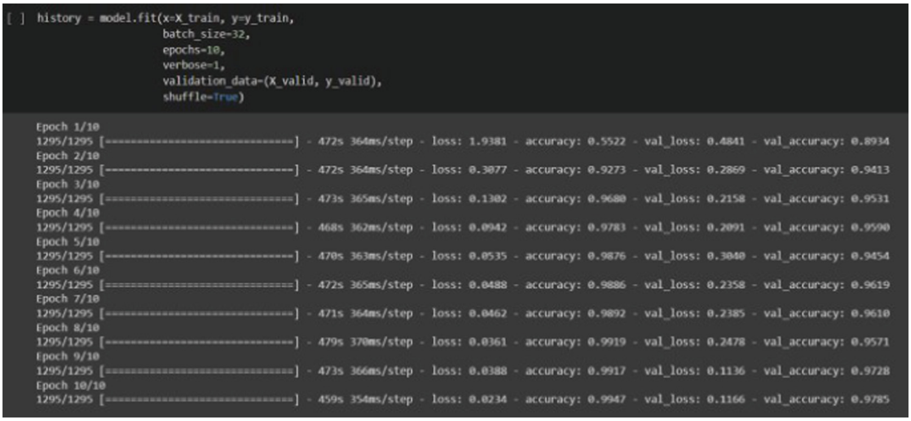


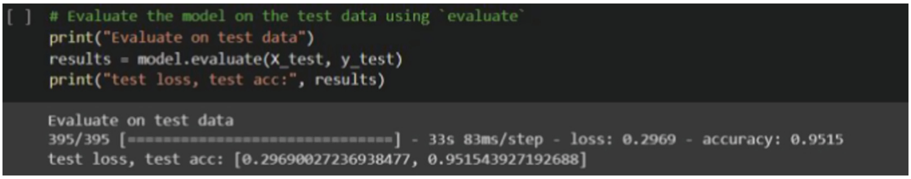
Fig. 8. Loss of YOLO v4 nodel



```
[ ] history = model.fit(x=X_train, y=y_train,
                        batch_size=32,
                        epochs=10,
                        verbose=1,
                        validation_data=(X_valid, y_valid),
                        shuffle=True)

    Epoch 1/10
    1295/1295 [==============================] - 472s 364ms/step - loss: 1.9381 - accuracy: 0.5522 - val_loss: 0.4841 - val_accuracy: 0.8934
    Epoch 2/10
    1295/1295 [==============================] - 472s 364ms/step - loss: 0.3077 - accuracy: 0.9273 - val_loss: 0.2869 - val_accuracy: 0.9413
    Epoch 3/10
    1295/1295 [==============================] - 473s 365ms/step - loss: 0.1302 - accuracy: 0.9680 - val_loss: 0.2158 - val_accuracy: 0.9531
    Epoch 4/10
    1295/1295 [==============================] - 468s 362ms/step - loss: 0.0942 - accuracy: 0.9783 - val_loss: 0.2091 - val_accuracy: 0.9590
    Epoch 5/10
    1295/1295 [==============================] - 470s 363ms/step - loss: 0.0535 - accuracy: 0.9876 - val_loss: 0.3040 - val_accuracy: 0.9454
    Epoch 6/10
    1295/1295 [==============================] - 472s 365ms/step - loss: 0.0488 - accuracy: 0.9886 - val_loss: 0.2358 - val_accuracy: 0.9619
    Epoch 7/10
    1295/1295 [==============================] - 471s 364ms/step - loss: 0.0462 - accuracy: 0.9892 - val_loss: 0.2385 - val_accuracy: 0.9610
    Epoch 8/10
    1295/1295 [==============================] - 479s 370ms/step - loss: 0.0361 - accuracy: 0.9919 - val_loss: 0.2478 - val_accuracy: 0.9571
    Epoch 9/10
    1295/1295 [==============================] - 473s 366ms/step - loss: 0.0388 - accuracy: 0.9917 - val_loss: 0.1136 - val_accuracy: 0.9728
    Epoch 10/10
    1295/1295 [==============================] - 459s 354ms/step - loss: 0.0234 - accuracy: 0.9947 - val_loss: 0.1166 - val_accuracy: 0.9785
```

Fig. 9. Training the YOLO v4 detection model



```
[ ] # Evaluate the model on the test data using `evaluate`
    print("Evaluate on test data")
    results = model.evaluate(X_test, y_test)
    print("test loss, test acc:", results)

    Evaluate on test data
    395/395 [==============================] - 33s 83ms/step - loss: 0.2969 - accuracy: 0.9515
    test loss, test acc: [0.29690027236938477, 0.951543927192688]
```

Fig. 10. Testing loss and accuracy of the YOLO v4 model

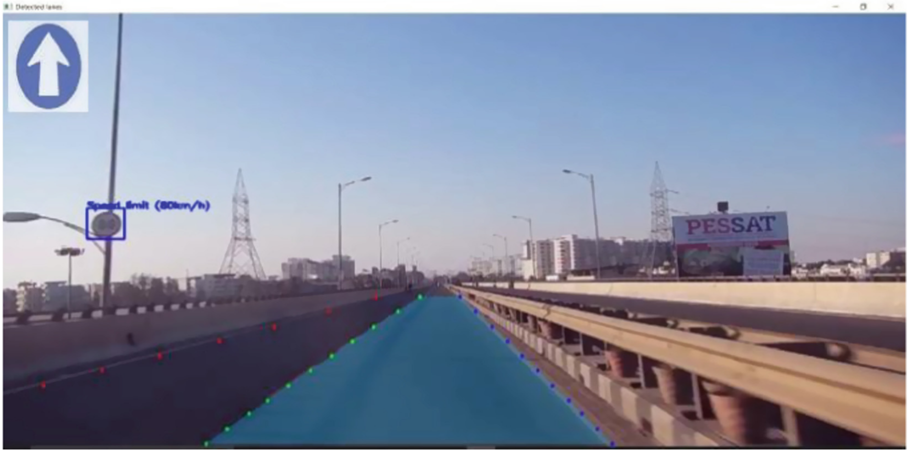


Fig. 11. Sample output of the proposed system

6 Conclusion

In conclusion, this research paper has presented a smart driving assistance system that incorporates deep learning techniques for lane detection and departure alert, as well as traffic sign detection. The system utilizes YOLO, a state-of- the-art object detection algorithm, to detect traffic signs and provide voice alerts to the driver. The proposed system has been tested and validated on real-world driving scenarios, and the results show that it is able to accurately detect lanes and traffic signs, and provide useful alerts to the driver in a timely manner. The proposed system has the potential to improve the safety and efficiency of driving significantly and can be integrated into existing vehicles to provide an additional layer of safety for drivers. However, it is important to note that plagiarism is not acceptable in any academic work and it is crucial to always give credit to the sources used in the research.

7 Future Work

Here are some of the suggested future works that can be done on the smart driving assistance.

- Integration with navigation systems for real-time lane guidance.
- Implementation of driver drowsiness detection and alert.
- Object detection and tracking for road obstacles such as pedestrians and cyclists.
- Integration with adaptive cruise control for improved driving comfort and safety.
- Further improvement in traffic sign detection accuracy through fine-tuning of YOLO.
- Expansion of traffic sign detection to include additional signs and symbols.
- Improved voice alert system for better clarity and natural language processing.
- Development of an alert system for road conditions such as ice or heavy rain.
- Integration with autonomous driving technologies.
- Analysis of driving behavior data for personalized driving assistance.
- Real-time weather condition updates for improved driving safety.
- Integration with vehicle sensors for improved lane departure alerts.
- Expansion of the system to support multiple languages.
- Development of a mobile application for real-time monitoring and control of the system.
- Utilization of machine learning algorithms for continuous improvement in system performance.
- Implementation of a collision avoidance system.
- Integration with car infotainment systems for improved user experience.
- Implementation of an emergency response system for accidents.
- Development of a cloud-based platform for centralized data analysis and management.
- Expansion of the system to support additional vehicle models and makes.

Acknowledgments. The authors wish to express their sincere gratitude to the following individuals who provided invaluable support and guidance throughout the development of this study: Prof. Mahesh Basavanna from the Department of Computer Science and Engineering at PES University, for his unwavering assistance and encouragement; Dr. Shylaja S S, Chairperson of the Department of Computer Science and Engineering, for her expertise and support; and Dr. B.K. Keshavan, Dean of Faculty at PES University, for his valuable assistance. The authors also extend their heartfelt appreciation to Dr. M. R. Doreswamy, Chancellor of PES University; Prof. Jawahar Doreswamy, Pro-Chancellor of PES University; and Dr. Suryaprasad J, Vice-Chancellor of PES University, for providing countless opportunities and enlightenment at every stage of the study. Lastly, the authors wish to acknowledge the continuous support and encouragement provided by their family members, friends, and the technical/office staff of the CSE Department, without whom this study could not have been completed.

References

1. Dhar, P., Abedin, M.Z., Biswas, T., Datta, A.: Traffic sign detection - a new approach and recognition using convolution neural network. In: IEEE Region 10 Humanitarian Technology Conference, pp. 416–419 (2017)
2. Ying, Z., Li, G.: Robust lane marking detection using boundary-based inverse perspective mapping. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (1921) (2016)

3. Ko, Y., Lee, Y., Azam, S., Munir, F., Jeon, M., Pedrycz, W.: Key points estimation and point instance segmentation approach for lane detection. IEEE Trans. Intell. Transp. Syst. **23**, 8949–8958 (2021)
4. Sanyal, B., Padhy, A., Mohapatra, R.K., Dash, R.: Real-time Indian TSR using MDEffNet. In: 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), pp. 1–5 (2022)
5. Sun, Y., Ge, P., Liu, D.: Traffic sign detection and recognition based on convolutional neural network. Chinese Automation Congress (CAC), pp. 2851–2854 (2019)
6. Seo, D., Jo, K.: Inverse perspective mapping based road curvature estimation. In: IEEE/SICE International Symposium on System Integration, pp. 480–483 (2014)
7. Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., Wang, Q.: Robust lane detection from continuous driving scenes using deep neural networks. IEEE Trans. Veh. Technol. **69**, 41–54 (2020)
8. Pan, J., Shi, P., Luo, X., Wang, X., Tang: Spatial as deep: Spatial CNN for traffic scene understanding. In: AAAI Conference on Artificial Intelligence (AAAI) (2018)