Volume 9, Issue 6, June – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1657

Lane Line and Object Detection Using Yolo v3

Shiva Charan Vanga[1]
CSE(AI&ML)
CMR Engineering College Hyderabad, Telangana

Rajesh Vangari[2]
CSE(AI&ML)
CMR Engineering College Hyderabad, Telangana

Shyamala Vasre[3]
CSE(AI&ML)
CMR Engineering College Hyderabad, Telangana.

Dheekshith Rao Nayini[4]
CSE(AI&ML)
CMR Engineering College Hyderabad, Telangana

A. Amara Jyothi[5]
CSE(AI&ML)
CMR Engineering College Hyderabad, Telangana

Abstract:- In the contemporary age, creating autonomous vehi- cles is a crucial starting point for the advancement of intelligent transportation systems that rely on sophisticated telecommu- nications network infrastructure, including the upcoming 6g networks. The paper discusses two significant issues, namely, lane detection and obstacle detection (such as road signs, traffic lights, vehicles ahead, etc.) using image processing algorithms. To address issues like low accuracy in traditional image processing methods and slow real-time performance of deep learning-based methods, barriers for smart traffic lane and object detection algorithms are proposed. We initially rectify the distorted image resulting from the camera and then employ a threshold algorithm for the lane detection algorithm. The image is obtained by extracting a specific region of interest and applying an inverse perspective transform to obtain a top-down view. Finally, we apply the sliding window technique to identify pixels that belong to each lane and modify it to fit a quadratic equation. The Yolo algorithm is well-suited for identifying various types of obstacles, making it a valuable tool for solving identification problems. Finally, we utilize real-time videos and a straightforward dataset to conduct simulations for the proposed algorithm. The simula- tion outcomes indicate that the accuracy of the proposed method for lane detection is 97.91% and the processing time is 0.0021 seconds. The proposal for detecting obstacles has an accuracy rate of 81.90% and takes only 0.022 seconds to process. Compared to the conventional image processing technique, the proposed method achieves an average accuracy of 89.90% and execution time of 0.024 seconds, demonstrating a robust capability against noise. The findings demonstrate that the suggested algorithm can be implemented in self-driving car systems, allowing for efficient and fast processing of the advanced network.

Keywords:- *Component, Formatting, Style, Styling, Insert.*

I. INTRODUCTION

Lane detection models are created to recognize and monitor the lanes on a road or driving surface by analyzing input images or videos. The objective of lane detection is to aid in various applications, including lane departure warning sys- tems, self-driving cars, and advanced driver assistance systems (adas). Lane detection models commonly employ computer vision techniques and advanced deep learning algorithms to identify and delineate the lane boundaries. Identify applicable funding agency here. If none, delete this.

Object segmentation models strive to divide and isolate individual objects within an image or video by assigning pixel-level labels to distinct object regions. Unlike object detection, which primarily focuses on identifying and marking objects as rectangular boxes, object segmentation offers a more comprehensive understanding of the object's shape and boundaries. It is widely utilized in numerous applications, including autonomous navigation and image and video editing.

- *Objective*

The objective of autonomous vehicle models is to enable vehicles to navigate and operate on roads without human in- tervention, utilizing sensors, algorithms, and decision-making systems to ensure safe and efficient transportation while min- imizing human error and improving mobility.

- *Existing System*

The CNN-RNN framework employs Convolutional Neural Networks (CNNs) to encode images into vectors, known as image features, which are then decoded into captions by Recurrent Neural Networks (RNNs). Initially, CNNs transform images into feature vectors, and these vectors are fed into RNNs to generate the captions, with support from the NLTK library for processing the actual captions. Despite needing more training time due to its sequential nature, the CNN-RNN model usually experiences

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1657

less loss and typically produces more accurate and relevant captions. However, the model de- mands significant training time and faces the risk of vanishing gradients, which can impair the learning process over long sequences.

In contrast, the CNN-CNN framework utilizes CNNs for both encoding images and decoding them into captions. CNNs first encode the image into feature vectors, and these features are then mapped to a vocabulary dictionary to generate words that form the caption, using models from the NLTK library. This framework has the advantage of a generally faster training process compared to the CNN-RNN framework. However, the CNN-CNN model tends to have a higher loss, leading to less accurate and relevant captions, which is not ideal for generating meaningful captions.

Comparatively, the CNN-CNN model trains more quickly than the CNN-RNN model. While the CNN-RNN model is more advanced and yields more accurate and relevant captions with less loss, the CNN-CNN model, despite its faster training time, often results in higher loss and less precise captions. Therefore, the CNN-RNN model, experiencing lower loss, is preferable for generating high-quality captions despite its longer training time. Both frameworks have their advantages and disadvantages: the CNN-RNN model is favored for its accuracy and relevance in caption generation, even though it requires more training time and can have issues with vanishing gradients, while the CNN-CNN model offers faster training but at the cost of higher loss and reduced accuracy in captions.

II. PROPOSED SYSTEM

The YOLO (You Only Look Once) model has gained significant attention and adoption in the field of autonomous vehicles due to its real-time object detection capabilities. Here is a more comprehensive overview of how the YOLO model can be utilized in autonomous vehicles:

- *Object Detection*

The primary purpose of using YOLO in autonomous vehi- cles is for accurate and efficient object detection. The model can identify and localize various objects in the vehicle's surroundings, such as vehicles, pedestrians, cyclists, traffic signs, and other obstacles.

- *Real-Time Processing*

YOLO's architecture allows for real-time processing, en- abling the vehicle to continuously perceive and react to the changing environment in real-time. This capability is crucial for autonomous vehicles to make immediate decisions and navigate safely.

- *Single-Pass Architecture*

YOLO operates on a single-pass architecture, where it analyzes the entire image in a single forward pass through the neural network. This design enables quick inference times, making it suitable for real-time applications such as autonomous driving.

- *Speed and Efficiency*

YOLO is known for its high processing speed, making it well-suited for resource-constrained systems like autonomous vehicles. Its efficiency comes from shared convolutional layers that extract features once and predict object attributes simul- taneously, resulting in faster inference times.

- *Multi-Class Object Detection*

YOLO can detect and classify multiple objects in a single image. It predicts both the bounding box coordinates and the class probabilities for each object, providing rich information about the detected objects.

- *Integration with Sensor Data*

YOLO can incorporate sensor data from various sources, such as cameras, LiDAR, or radar, to enhance object detection. By fusing information from different sensors, YOLO can improve the accuracy and reliability of

- *Training on Diverse Datasets*

To optimize YOLO for autonomous driving, the model can be trained on large-scale datasets that include annotated im- ages and videos of different driving scenarios. This allows the model to learn and generalize object detection across various road conditions, lighting conditions, and object appearances.

- *Fine-grained Object Localization*

YOLO provides precise bounding box coordinates for each detected object, enabling accurate localization. This informa- tion is crucial for autonomous vehicles to plan and execute safe maneuvers, such as lane changing, overtaking, and maintaining safe distances from other objects.

- *Integration with Decision-Making Systems*

YOLO's output, including object bounding boxes and class probabilities, can be used as input for the vehicle's decision- making system. By leveraging the detected objects, the au- tonomous vehicle can make informed decisions, such as path planning, trajectory prediction, and collision avoidance.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1657

➢ *Activity Diagram*

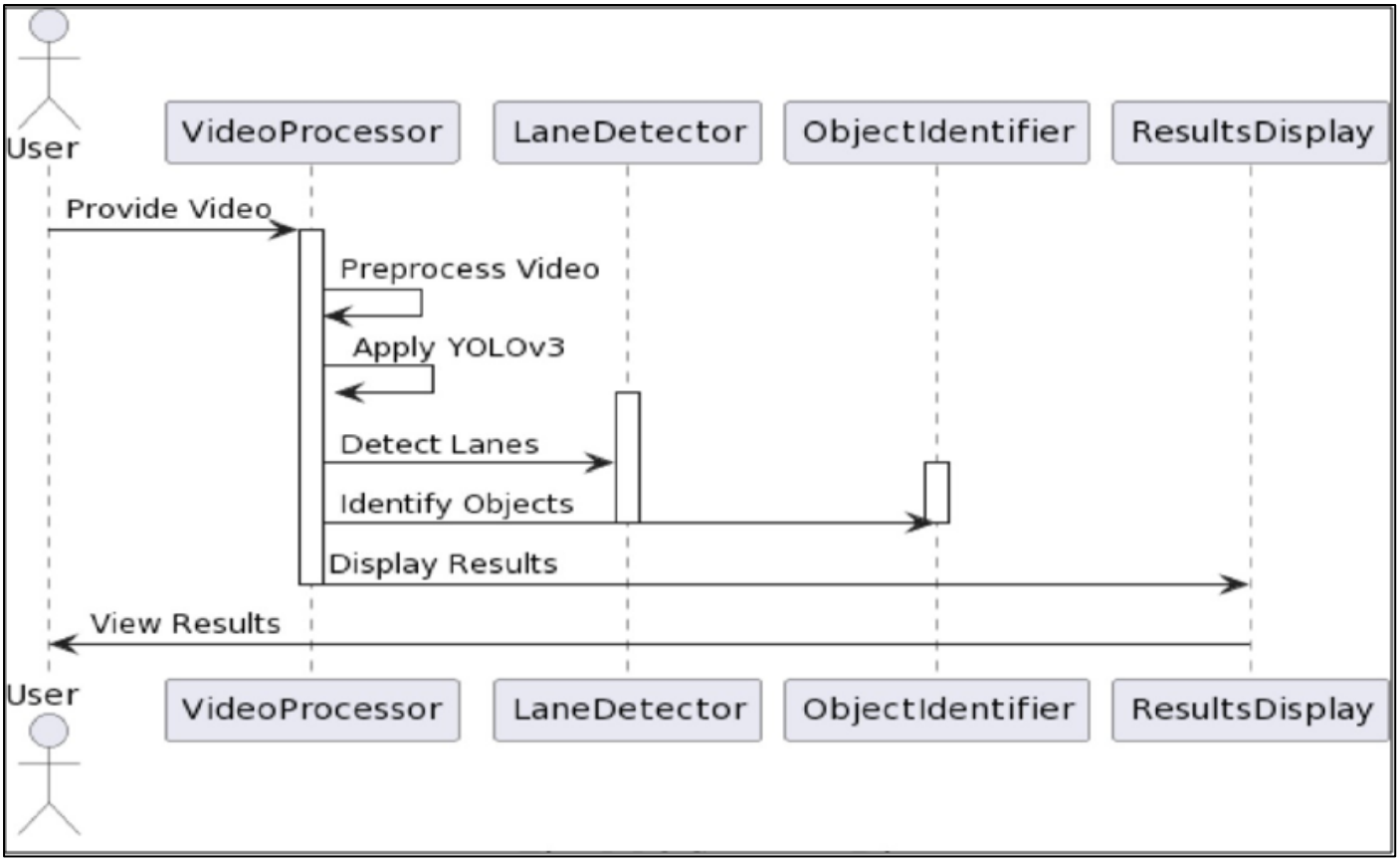

Fig 1 Activity Diagram

III. METHODOLOGY

The process of lane segmentation entails gathering a collec- tion of image-caption pairs, pre-processing the data, extract- ing visual features using a pre-trained convolutional neural network (cnn), and processing captions using text encoding. Techniques, designing a yolo architecture, training the model using the encoded data, and evaluating the generated captions. During inference, an image is encoded using the cnn, and the yolo generates features. The success of the lane.

IV. FEATURE EXTRACTION USING YOLO MODEL

In the YOLO (You Only Look Once) model, feature ex- traction is an integral part of the object detection process. The YOLO model performs both feature extraction and object detection simultaneously in a single pass through the network. Here's an overview of how feature extraction is performed using the YOLO model:

- Convolutional layers: The YOLO model commonly com- prises multiple convolutional layers that extract features from the input image. These layers perform convolutions, applying filters to capture local patterns and features at various scales and levels of abstraction.
- Pre-trained Backbone Network: YOLO frequently uti- lizes a pre-trained convolutional neural network (CNN) as its primary network structure. The backbone network, including darknet, resnet, or mobilenet, is trained on extensive image classification tasks, enabling it to extract general visual fea- tures from images.
- Shared Convolutional Layers: The first layers of the pre-trained backbone network are reused in the yolo model, enabling them to identify basic features such as edges, tex- tures, and colors. This common feature extraction technique enhances efficiency by eliminating the need for redundant calculations for various objects.
- Spatial Hierarchy: The YOLO model utilizes multiple layers of its backbone network to extract features at different scales and levels of abstraction. The deeper layers of the model capture more abstract, high-level features, while the earlier layers focus on finer details. This spatial hierarchy allows the model to gather features at various levels of detail.
- Feature Maps: Each layer of the YOLO model's backbone network produces feature maps. Feature maps retain spatial information but have reduced spatial resolution compared to the input image. These maps encode different semantic information and capture features relevant to object detection.
- Anchors: YOLO utilizes anchor boxes, which are prede- termined bounding boxes of various sizes and aspect ratios. The model employs these anchor boxes to estimate the co- ordinates of bounding boxes in relation to the anchor boxes. Anchors assist the model in managing objects of different sizes and forms.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1657

- Convolutional Predictions: YOLO performs convolutional predictions on the feature maps to produce predictions for bounding boxes and object class probabilities. These predic- tions are generated at various grid cells within the feature maps, enabling the model to identify objects at different positions.
- Prediction Layers: YOLO has prediction layers that analyze the feature maps and produce predictions for every grid cell. Each prediction layer generates a predetermined number of bounding boxes, along with the associated class probabilities and confidence scores.
- Upsampling and Concatenation: In some YOLO variants, the model may incorporate upsampling and concatenation operations to combine feature mapsfrom different scales. This fusion of features at multiple resolutions helps in capturing both fine-grained details and contextual information.
- Non-Maximum Suppression (NMS): Following the pre- diction generation, YOLO employs non-maximum suppression to eliminate redundant and overlapping bounding box pre- dictions. Nms chooses the most certain and non-intersecting boxes, enhancing the accuracy of the object detection out- comes.

➢ *Output*

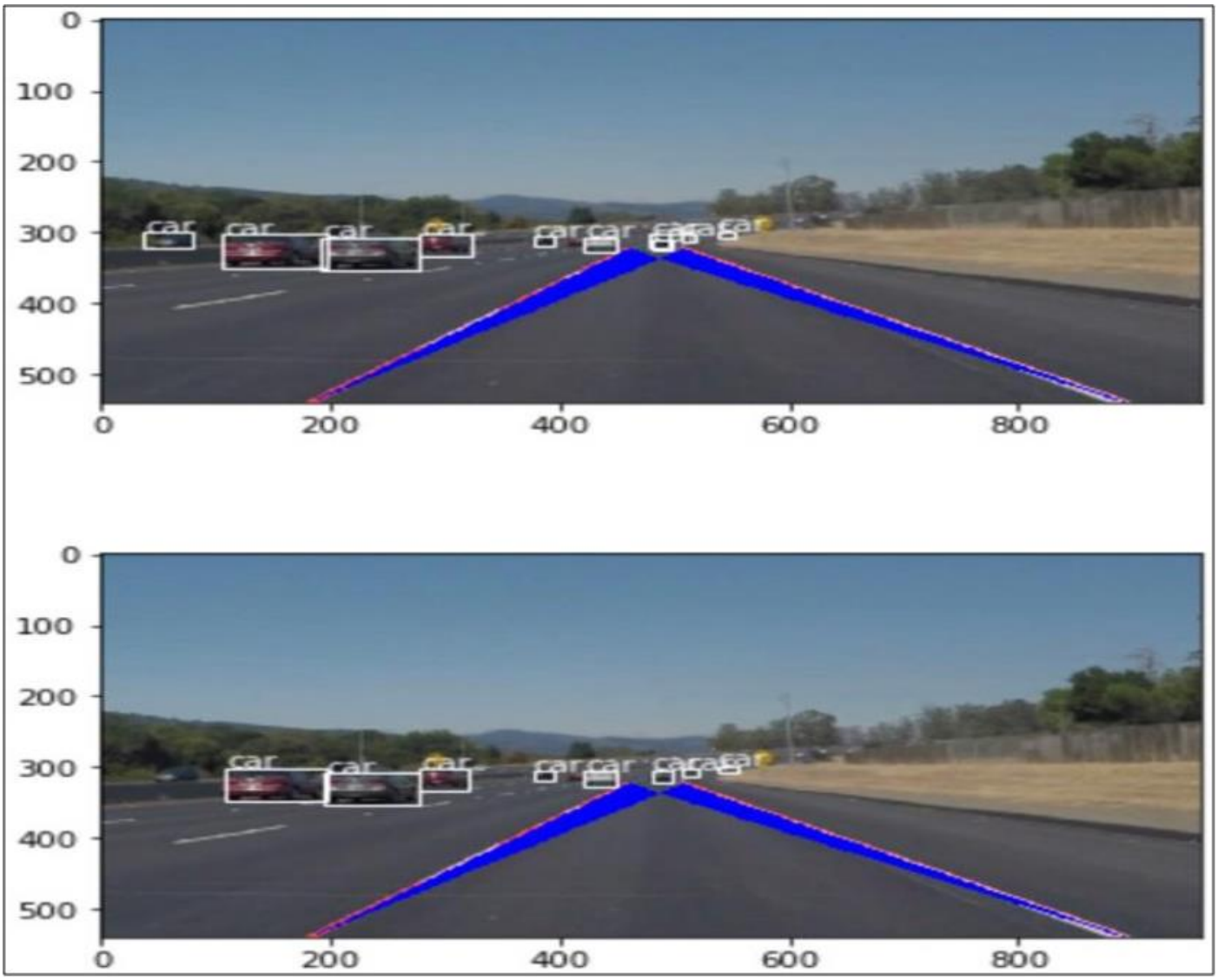

Fig 2 Output

V. CONCLUSION

In summary, computer vision techniques have been utilized to develop two-lane detection algorithms capable of accurately identifying and tracking vehicles on two-lane roads. The system employs a pre-trained YOLO model for the task of object recognition. Images captured by the vehicle's front camera present certain challenges. Research indicates that lane detection achieved nearly 80% accuracy, while obstacle detection had an average accuracy of 74.1%, as measured by the mean average precision (mAP). Car classification accuracy typically exceeded 80%. However, the system requires testing on additional datasets and further algorithm enhancements for better on-road performance with various adjustments. The lux meter was utilized to determine the level of intensity. The light emitted by the

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1657

bulb, playing a role in the formation of a system that can identify and overcome obstacles.

The algorithm must be able to adjust to different circum- stances, including changes in the environment. Lighting and other obstacles, possibly by hiring additional staff. When the algorithm was applied to the same basic dataset and compared with others, it was found that a wide range of obstacles could be identified using the 80-layer pre-trained YOLO model. The accuracy for individual layers varied, with some layers achieving up to 98.20%. Analysis showed that 75.06% of accidents involved cars, while 42.49% were due to stop signs, traffic lights, and other factors. When compared to a similar system utilizing Histogram of Oriented Gradients (HOG) and linear Support Vector Machines (SVM), it was found that the latter was only capable of detecting the presence of vehicles.Additionally, sharing the entire image and extracting features was more cost-effective than using a network vector via SVM. While the network requires modification only once for YOLO, SVM and random forest need approximately 150 modifications. Consequently, YOLO is about ten times faster than SVM and HOG. The confidence level was adjusted to 50%.

REFERENCES

[1]. Real Time Lane Detection in Autonomous Vehicles Using Image Processing Authors: Jasmine Wadhwa, G.S. Kalra and B.V. Kranthi Published: October 05, 2015

[2]. CNN based lane detection with instance segmentation in edge- cloud computing. Author: Wei Wang, Hui Lin, Junshu Wang Published: 19 May 2020 (CNN based lane detection with instance segmentation in edge-cloud computing — Jour- nal of Cloud Computing — Full Text (springeropen.com))

[3]. Real time lane detection for autonomous vehicles Au- thors: Abdulhakam. AM. A ssidiq, Othman O. Khalifa, Md. Rafiqul Islam, Sheroz Khan. Published: December 2021. (Real time lane detection for autonomous vehicles — IEEE Confer- ence Publication — IEEE Xplore)

[4]. A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios. Authors: Qingquan Li, Long Chen, Ming Li, Shih- Lung Shaw and Andreas Nuchter. Joi Published: December 2021. (A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios — IEEE Journals Magazine — IEEE Xplore)