WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Vehicle Collision Detection using Bounding Box Methodology

Dr.H Balaji[1], N Shiva Sai[2], K Abhinav[3], D Ganesh[4]

[1] Associate Professor, Dept. of Computer Science and Engineering, SNIST, Hyderabad, 501301, India
[2,3],4B. Tech Scholars, Dept. of Computer Science and Engineering, SNIST, Hyderabad, 501301, India

ABSTRACT

The safety of vehicles on the road is of paramount importance, and one of the major concerns is preventing collisions between vehicles. Real-time vehicle collision detection using bounding box methodology is a project that aims to use computer vision and machine learning techniques to detect collisions between vehicles on the road. In this project, we use a convolutional neural network (CNN) to detect vehicles and their positions on the road. The CNN uses YOLOv3 for object detection and bounding box methodology to detect vehicles in the images. Once the model is trained, it can be used for real-time detection of collisions between vehicles on the road. The real-time detection system uses a camera to capture images of vehicles on the road. The images are then processed by the CNN to detect vehicles and their positions. If a collision is detected, an alert is sent to the driver or autonomous vehicle to avoid the collision. Keywords: — Tensor flow, YOLO, Yolov3, Deep Learning

Key Words: — Tensor flow, YOLO, Yolov3, Deep Learning

INTRODUCTION

Real-time Vehicle Collision Detection using Bounding Box Methodology is a major project that aims to use computer vision and machine learning techniques to detect collisions between vehicles on the road. This project uses a convolutional neural network (CNN) to detect vehicles and their positions on the road. The CNN uses YOLOv3 for object detection and bounding box methodology to detect vehicles in the images. Once the model is trained, it can be used for real-time detection of collisions between vehicles on the road. The real-time detection system uses a camera to capture images of vehicles on the road. The images are then processed by the CNN to detect vehicles and their positions. If a collision is detected, an alert is sent to the driver or autonomous vehicle to avoid the collision.

1.1 Problem Statement

In this study, a new bounding-box-based vehicle tracking algorithm is presented to extract statistical information on highway traffic. A novel shaking filter and a new voting approach are employed in the vehicle detection and tracking phases to reduce camera shaking effects that cause misdetection, misclassification, and mis-tracking.

1.2 Existing system

The existing systems are simple and effective but are only able to analyze the rear portion of the vehicle. Increased risk of collisions: Without a collision detection system, drivers may not be aware of potential collisions, which could increase the risk of accidents Delayed response, and high insurance costs. A variety of driving safety assessments have been explored. Hence, not very effective and have low accuracy. Some Have Evaluated the safety issue mainly based on real-time vehicle kinematics. Others have comprehensively tried to monitor the D-V-E (driver, vehicle, and environment) statutes.

1.3 Proposed system

We propose an automated, real-time system for the beforehand detection of vehicle collisions during high traffic and intimate the concerned people using the application. Our approach will work on still images, recorded- videos, and real-time live videos and will detect, classify, track, and compute moving object velocity and direction using a convolution neural network. Using YOLO, it will be able to detect the front as well as the rear view of the vehicle and alert us beforehand. The Advantages of the proposed system are Secured, Interpretability, High accuracy, Lightweight model & fast processing. Moreover, this system can be used in the cases of Self-driving cars. Where it would analyze the collision possibility automatically and drive accordingly. It could be used in self-driving cars, traffic surveillance systems, traffic management, and automated driving applications

1.4 Objective

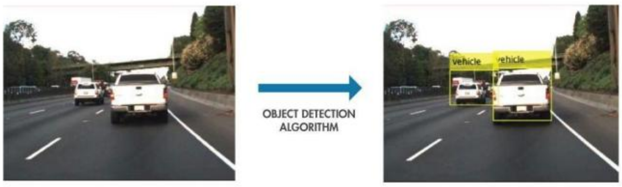
A vehicle collision detection system is a technological device that helps to identify and prevent collisions between vehicles. The system works by using a combination of sensors, cameras, and software algorithms to detect the presence of other vehicles, objects, and pedestrians on the road. This is a user-friendly project. The project is designed in such a way that the user can easily use the model. There is no harm to the environment due to this project. As the application is user-friendly the time taken to learn and use the application is very less but inside the project follows its layers of security. The information in this project has been secured. No one can manipulate the information present in the database or CSV files. There have been given respective privileges such as read, write permissions, etc.) to respective users and even the sensitive data is encrypted. The project is not harmful to any human. As mentioned earlier there have been taken a few measures where security has been provided. During the execution of the project, nowhere sensitive data is leaked. The cost taken to implement this project in the real world is very low. As the project is developed totally in Python. The Python programming language is open source.

2. LITERATURE SURVEY

Vehicle Collision Detection and Avoidance System based on YOLO and Deep Learning" proposes a collision detection system that uses YOLOv4 and deep learning algorithms to detect vehicles and obstacles on the road. The system provides visual and audio warnings to the driver and can automatically apply the brakes to avoid collisions.

The method we applied in this project is YOLO:

The system uses YOLO to detect objects in the camera feed from the vehicle's surroundings, including other vehicles, pedestrians, and cyclists. The YOLO algorithm returns the bounding boxes and probabilities of the objects detected in the image. For car detection, we used the YOLO net, which was proposed by, improved in, and more recently, the last version was presented in. Despite other convolutional networks, it detects objects with a single pass creating a grid of S x S boxes, where each box has a logistic regression and a classification method, the regression method predicts each box with five values: x, y, w, h, and the confidence of the object being there. The classifier predicts C conditional class probabilities. At the final stage, multiple bounding boxes appear around a single object, so non-maximum suppression is applied in order to keep the strongest detection around a single object. The architecture of YOLO for this research project is shown in Table I, obtained from the Darknet library. The result in this step is the bounding boxes for each car, the YOLO performance is shown, here, this network has very high accuracy and more important has a very low time processing against the rest.



YOLO model has been trained on large datasets of images and videos and are capable of detecting various types of objects, including vehicles, pedestrians, and obstacles.

YOLO algorithm achieves its result by applying a neural network to an image. The image is divided into an S x S grid and comes up with a bounding box [13]. This algorithm has 24 convolutional layers which in turn have two fully connected layers. The reduction in feature space is done by Alternating 1x1 convolutional layers from preceding layers. The object identification problem is considered to be a regression problem with the objective of spatially bounding box separation along with the probability of associated classes in the bounding boxes. A single neural network can predict the bounded boxes and class probabilities directly from the input images in just one evaluation which can be optimized end-to-end.

The bounding box is described using these four descriptions

Centre of bounding box (bx, by)

Width (bw)

Height (bh)

C: class name of the identified object Pc is the probability of objects in the bounding box.

3. SOFTWAREREQUIREMENT ANALYSIS

Following are the libraries used in the code:

- Numpy
- Cv2
- Time
- Librosa

NumPy-

NumPy is a fundamental library in Python that provides efficient and convenient tools for working with arrays and performing numerical computations. Its extensive functionality and performance optimizations make it a popular choice for scientific computing, data analysis, and machine learning applications. NumPy is widely used in various fields such as scientific research, data analysis, machine learning, and computational mathematics. Its efficient array operations, broadcasting capabilities, and extensive mathematical functions make it a key library for working with large datasets and performing complex computations in Python

Cv2 -

The "cv2" module in Python provides a wide range of functions and classes for tasks related to computer vision, such as image and video processing, object detection, feature extraction, camera calibration, and more. It allows developers to manipulate and analyze images and videos, as well as perform various operations like filtering, resizing, cropping, and drawing on images.

Time-

The "time" module in Python provides functions and classes for working with time-related operations. It allows you to measure time, manipulate time values, and perform various time-related calculations AND provides many other functions and capabilities for working with time in Python.

Librosa-

Librosa is a Python library for audio and music signal analysis. It provides a wide range of functions and tools for tasks such as audio feature extraction, spectral analysis, beat tracking, tempo estimation, and more. Librosa aims to simplify the process of working with audio data in Python and provides an intuitive interface for performing complex audio analysis tasks.

3.1 HARDWARE REQUIREMENTS:

- Display
- A system with all standard accessories.
- Cameras

3.2Functional Requirements:

- Maintenance of intake images and dividing them into grids.
- Maintenance of data intake objects as input.
- Maintenance of CSV (Comma Separated Values).
- Creating the CSV file and applying data pre-processing and predicting the
- collision of vehicles

4 System Design

4.1 System Architecture

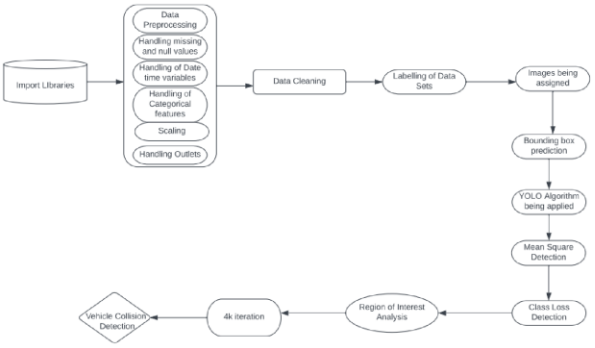


Fig 4.1 System Architecture

4.2 UML DIAGRAMS

UML is a graphical language for visualizing, specifying, constructing, and documenting information about software-intensive systems. UML gives a standard way to write a system model, covering conceptual ideas. UML can be used to develop diagrams and provide users (programmers) with ready-to-use, expressive modeling examples.

4.2.1 CLASS DIAGRAM

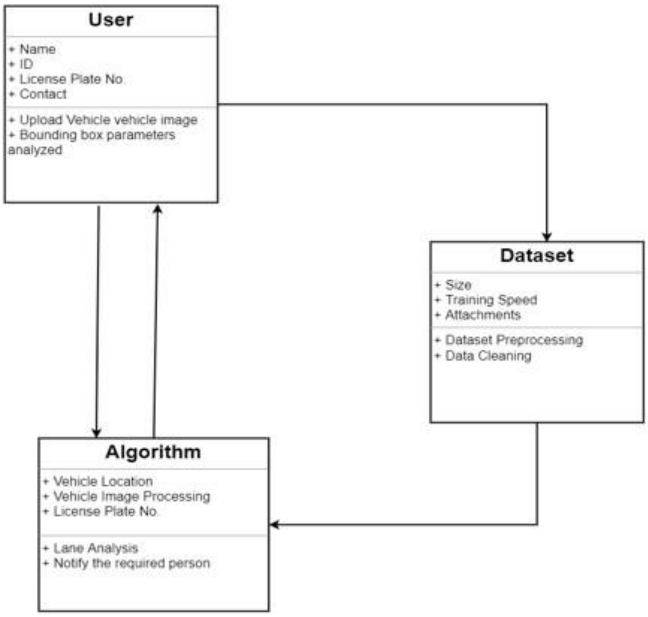


Fig 4.2.1Class Diagram for VCD

The above class diagram represents the total flow of the project and can able to visualize all the fields and methods that are used in the project. There consists of 7 dependent levels to execute the project. Every level is dependent on any other level because the project undergoes a top-down approach**.**

4.2.2 USE CASE DIAGRAM

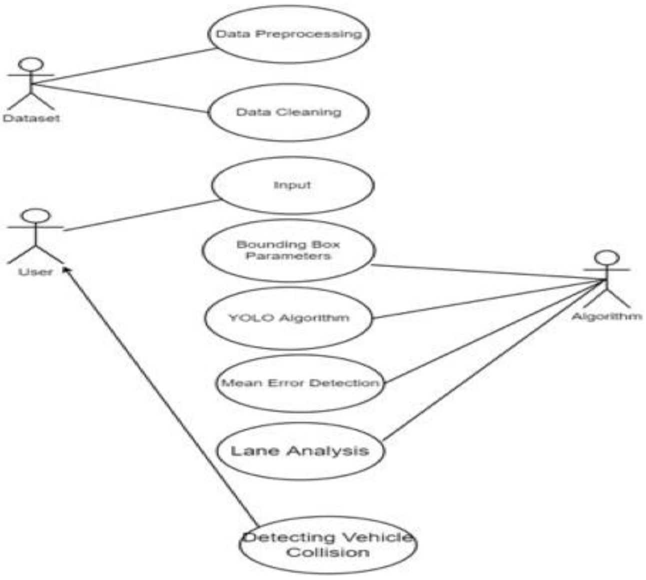


Fig 4.2.2USE CASE Diagram for VCD

4.2.3 SEQUENCE DIAGRAM

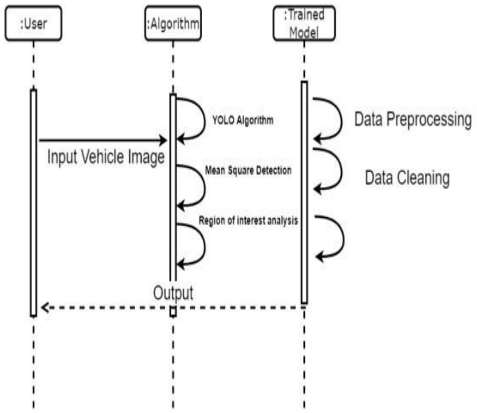


Fig 4.2.3 SEQUENCE Diagram for VCD

The sequence diagrams follow similar steps that are followed by a class diagram.The sequence diagram explains the object interactions arranged in a time sequence. It depicts the objects and classes involved in the project and the sequence of messages exchanged between the objects needed to carry out the functionality of the project**.**

4.2.4 ACTIVITY DIAGRAM

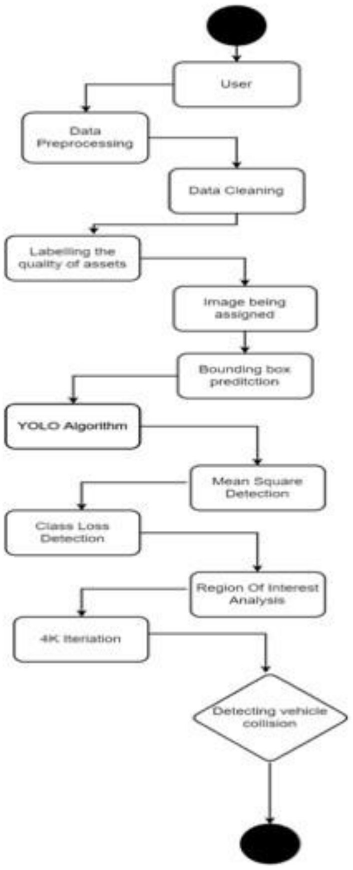
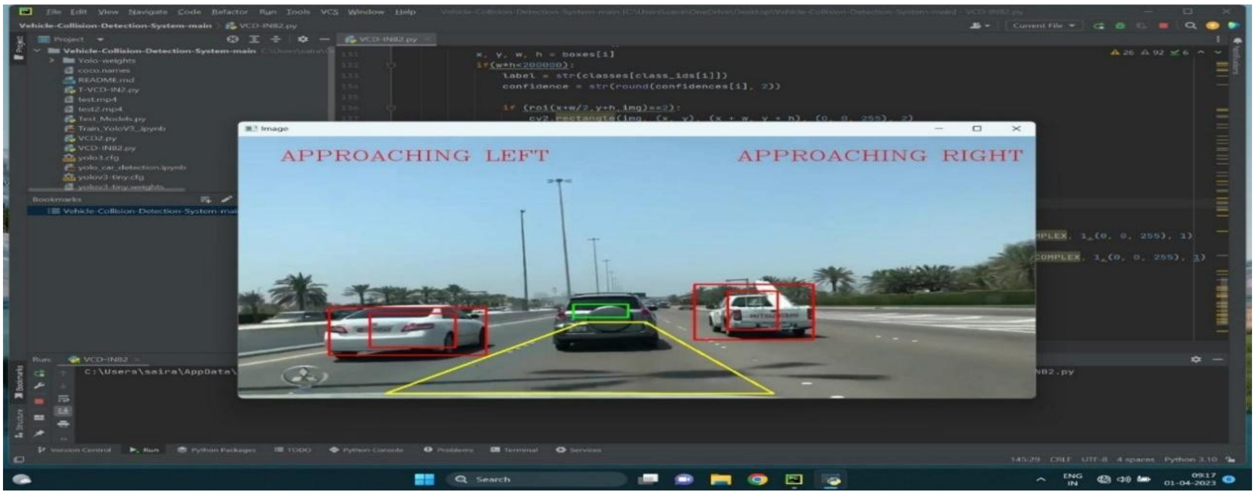


Fig 4.2.4ACTIVITY Diagram for VCD

The activity diagram would provide a detailed overview of the system's functionality and the steps involved in detecting and avoiding collisions. Each step would be represented by a box or oval, with arrows indicating the flow of data or control between steps. The diagram could be used as a reference for the development team and as a tool for explaining the system's functionality to stakeholders

5. RESULT ANALYSIS

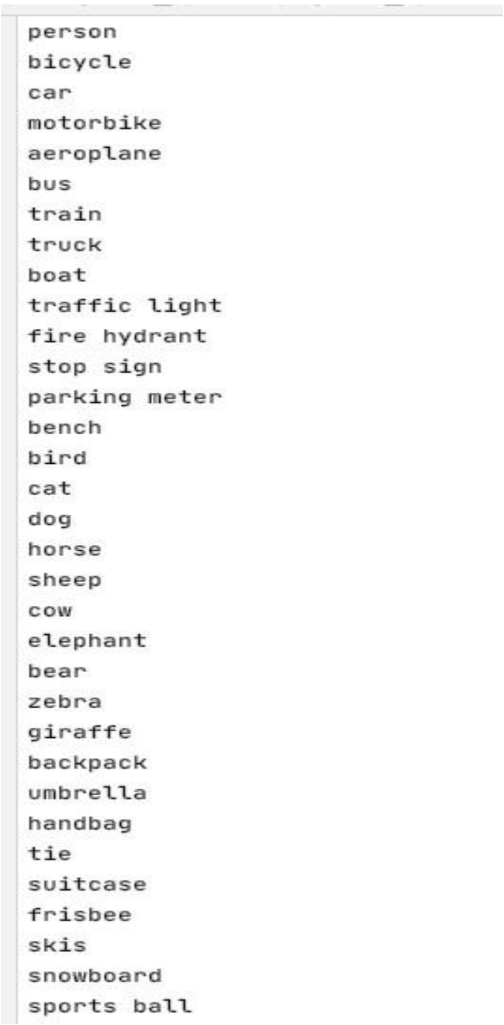
When a vehicle is approaching, the distance is calculated between the vehicle and our vehicle. Ifthe distance is too close a warning "Vehicle approaching left" or "Vehicle approaching right" is printed with respect to the vehicle position.



```
x, y, w, h = boxes[i]
if(w*h<200000):
    label = str(classes[class_ids[i]])
    confidence = str(round(confidences[i], 2))

    if (roi(x+w/2,y+h,img)==2):
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

```
C:\Users\saira\AppData\
```



```
person
bicycle
car
motorbike
aeroplane
bus
train
truck
boat
traffic light
fire hydrant
stop sign
parking meter
bench
bird
cat
dog
horse
sheep
cow
elephant
bear
zebra
giraffe
backpack
umbrella
handbag
tie
suitcase
frisbee
skis
snowboard
sports ball
```

6.CONCLUSION

In this study, the proposed accident detection system can be trained by using a regression-based algorithm called YOLO(you only look once) algorithm on the sample vehicle datasets and the vehicle detection process has been successfully performed by the trained model vehicle detector being tested on the test data set with the live video feeds from the webcam. The proposed system is faster than other object detection methods and predicts the object better than other object detection algorithms such as Faster-CNN or Fast CNN. The input can also be optimized and give better results. Further, the system alerts via wireless communication devices to nearby emergency vehicles

.REFERENCES

1. Mousa, “Canny Edge-Detection based Vehicle Plate Recognition”, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 5, No. 3, pp. 1-8, 2012.
2. S. Du, M. Ibrahim, M. Shehata and W. Badawy, “Automatic License Plate Recognition (ALPR): A State-of-the-Art
3. Review”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, No. 2, pp. 311-325, 2013.
4. S. Ghofrani and M. Rasouli, “Farsi License Plate Detection and Recognition based on Characters Features”, Majlesi Journal of Electrical Engineering, Vol. 5, No. 2, pp. 44-51, 2011.
5. M.S. Sarfraz, A. Shahzad, M.A. Elahi, M. Fraz, I. Zafar and E.A. Edirisinghe, “Real-Time Automatic License Plate
6. Recognition for CCTV Forensic Applications”, Journal of Real-Time Image Processing, Vol. 8, No. 3, pp. 285-295, 2013.
7. R. Azad, F. Davami and B. Azad, “A Novel and Robust Method for Automatic License Plate Recognition System based on Pattern
8. Recognition”, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 3, pp. 64-70, 2013.
9. H. Caner, H.S. Gecim and A.Z. Alkar, “Efficient Embedded Neural Network-based License Plate Recognition System”, IEEE Transactions on
10. Vehicular Technology, Vol. 57, No. 5, pp. 2675-2683, 2008.
11. S.D. Palmer and O.N. Aharoni, “System for Collision Prediction and Traffic Violation Detection,” U.S. Patent 20 130 093 895, Apr. 18, 2013.
12. G. Liu, Z. Ma, Z. Du and C. Wen, “The Calculation Method of Road Travel Time based on License Plate Recognition Technology”, Proceedings ofInternational Conference on Computer Science and Education, pp. 385-389, 2011,
13. J.M. Guo and Y.F. Liu, “License Plate Localization and Character Segmentation with Feedback Self-Learning and
14. Hybrid Binarization Techniques”, IEEE Transactions on Vehicular Technology, Vol. 57, No. 3, pp. 1417-1424, 2008.
15. P.H. Torr and D.W. Murray, “The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix”, International Journal of Computer Vision, Vol. 24, No. 3, pp. 271-300, 1997.
16. Y.W. Chen and C.J. Lin, “Combining SVMs with Various Feature Selection Strategies”, Available at: https://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf. [12]J. Weston, S. Barnhill and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines,” Machine Learning, Vol. 46, No. 1-3, pp. 389-422, 200