OBSTACLE DETECTION AND ALARM SYSTEM FOR AUTONOMOUS DRIVING CAR

A PROJECT REPORT

Submitted by

AKASH KAMAL.R (810015205003)

MUTHU RAMAN.R (810015205046)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING, BIT CAMPUS, TIRUCHIRAPPALLI

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2019

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"OBSTACLE DETECTION AND ALARM SYSTEM FOR AUTONOMOUS DRIVING CAR"** is the bonafide work of **"AKASH KAMAL R (810015205003)** and **MUTHU RAMAN R (810015205046)"** who carried out the project work under my supervision.

Mr. D. VENKATESAN

Head Department of CSE

Department of Information Technology

Anna University, BIT Campus
University College of Engineering
Tiruchirappalli - 620 024

Dr. R. KRISHNAMOORTHY

SUPERVISOR

Department of Information Technology

Anna University, BIT Campus
University College of Engineering
Tiruchirappalli - 620 024

Submitted for the ANNA UNIVERSITY Viva-voce Examination held on

……………….

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that the work entitled "**OBSTACLE DETECTION AND ALARM SYSTEM FOR AUTONOMOUS DRIVING CAR**" is submitted in partial fulfillment of the requirement for the award of the degree in B. TECH, in University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. It is the record of our own work carried out during the academic year 2018 – 2019 under the supervision and guidance of **Dr. R. KRISHNAMOORTHY**, Professor, Department of Information Technology, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places.

R. AKASH KAMAL
(810015205003)

R. MUTHU RAMAN
(810015205046)

I certify that the declaration made above by the candidates is true

Signature of the Guide,

DR. R. KRISHNAMOORTHY
PROFESSOR
Department of Information Technology
University College of Engineering,
Anna University – BIT Campus,
Tiruchirappalli – 620 024.

ACKNOWLEDGEMENT

We would like to thank our honorable Dean **Dr. T. SENTHIL KUMAR**, Professor for having provided us with all required facilities to complete our project without hurdles.

We would also like to express our sincere thanks to **Mr.D.VENKATESAN**, Head of the Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

We would like to thank our Project Coordinator **Mr. M. PRASANNA KUMAR**, Teaching Fellow, Department of Information Technology for her kind support.

We would like to thank and express our deep sense of gratitude to our project guide **Dr. R. KRISHNAMOORTHY,** Professor, Department of Computer Science and Engineering, for his valuable guidance throughout the project. We also extend our thanks to all other teaching and non-teaching staff for their encouragement and support.

We thank our beloved parents and friends for their full support in the moral development of this project.

ABSTRACT

The rate of death caused due to road accidents is increasing at an alarming way. Many road rules and regulations are framed in order to reduce it. Most of the road accidents are caused by human errors. This can only be nullified when the manual driving process is replaced by automating driving. There are many factors to be considered while automating the driving process. Deep learning concepts help us making a near human replacement for the driving process.

In this work, a novel methodology is proposed for the purpose of detection of high priority obstacles, followed by design of an alarm system. This proposed obstacle detection is completely based on computer vision. In contrast to the existing systems that use sensors and other related components, the proposed system makes use of a single camera with image processing techniques. The proposed system utilizes the images captured from the single camera and identifies all the obstacles with Convolutional Neural Network. In this proposed technique, low priority obstacles are neglected, as they may not be the cause for any sort of damage. After identifying only higher priority obstacles that comes close to the self-driving vehicle, the proposed system computes the distance between the obstacle and the driverless car by calculating the area of the bounding box associated with that obstacle. The proposed system also gives an alarm on the monitor, once the distance exceeds the threshold distance. The proposed obstacle detection and alarm system for autonomous driving car has been experimented in python platform and the results are encouraging.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
CNN	-	Convolutional Neural Network
IoU	-	Intersection over Union
GPU	-	Graphics Processing Unit

CHAPTER – 1

INTRODUCTION

1.1 OVERVIEW

The most common problem we face while driving is accidents. As cities are developed, the complexity of driving gets higher. This results in high accidents. In order to avoid this, Autonomous self-driving car comes into the play. Human has the intelligence to avoid and handle road obstacles. But there are many factors and situations where humans fail to utilize it correctly and it leads to accidents and life loss. Due to technological advancement it is now possible to provide intelligence to machine using Machine learning process. Thus the self-driving autonomous car has the capability to get enriched with intelligence of obstacle detection, handling and avoidance. Vision based approach is now emerging with a lot of advantages over the previous methodologies where sensors played a vital role. Deep learning concept is also implemented to provide intelligence to the vehicle. While going through vision based approach Convolutional Neural Network (CNN) acts as the key in accomplishing intelligence providence. It is proved that the accident rate can be reduced at a large scale when manual driving is completely replaced by autonomous driving car. The major background knowledge of the techniques and methodologies used in this proposed work is discussed below.

1.2 ARTIFICIAL INTELLIGENCE

Artificial Intelligence, commonly referred to as AI, is basically any intelligence demonstrated by a machine that leads it to an optimal or sub optimal solution. Artificial Intelligence is the reason for many tasks getting self-done easier.

This forms the backbone of many high peak of development in technological field. AI is most commonly used in Robotics, Autonomous driving car, etc.

1.3 DEEP LEARNING

Deep learning is motivated by the function of human brain. Neural Network forms the core concept, which mimics the activities and function of our body's Neural Network system. Deep learning has the capability of studying in unsupervised manner from data which is in unstructured form. Deep learning is a subset of Machine Learning (ML) which uses layers of Neural network to undergo the process of Machine Learning. The major areas where Deep learning is used are fraud detection, Image processing, autonomous self-driving car, medicine etc.

1.4 NEURAL NETWORK

Neural networks are a set of algorithms that are framed after the human brains. They are designed to recognize patterns. They interpret the data of any form by converting them into binary format and extract feature automatically (Unsupervised) that is further used for many applications. Neural network helps in clustering and classifying data. Deep neural network is a component of larger machine learning application involving algorithms for reinforcement learning, regression and classification. A neural network consists of many layers. Each layer consists of many nodes. Nodes are the places where the computation happens. Few types of neural networks are

- Feedforward neural network
- Recurrent neural network

- Convolutional neural network

1.5 CONVOLUTIONAL NEURAL NETWORK

Out of many neural network, convolutional neural network model forms the major part in vision based approach. CNNs are powerful image processing, artificial intelligence that use deep learning to perform both generative and descriptive tasks, often using machine vision that includes image and video recognition. The main advantage of CNN is that it requires little preprocessing compared to other image classification technique. As it learns the features by itself, human efforts in trying to preprocess the image and to extract the features is significantly reduced.

1.6 USAGE IN AUTONOMOUS DRIVING CAR

Autonomous driving car is emerging and many machine learning algorithms are applied to make it a perfect mimic of manual driving. In this project work we are using CNN to identify multiple objects and classification of those objects, which are considered to be far too complex while compared to binary image classification. Deep learning helps in obtaining all the information required to provide intelligence about obstacle handling on autonomous driving car. In addition to this, specific additional features are added to make the alarm system work perfectly which are briefly discussed in the preceding chapters.

1.7 BENEFITS

The main advantage of our proposed Alarm system for autonomous self-driving car is that the obstacles are well handled. While considering sensors it detects both a cardboard box and a pedestrian under the same category as ‘obstacles’. It doesn’t give priority to the objects as the objects are unclassified. Using our proposed methodology, we can distinguish object which results in handling in a perfect way. For example, a high speed car should stop only for high priority objects like pedestrians, car, motorbike etc. and not for objects like cardboard box, trash etc.

1.8 MOTIVATION

It is stated that about 821 people die every day in India just because of road accidents. This can be completely avoided with Autonomous driving car. But this can be only accomplished if all the cars are provided with intelligence to handle speed, obstacles, rules simultaneously while driving. Implementing all these in real time is a much more difficult task. Hence we reduced our scope to Obstacle handling for autonomous driving car.

1.9 OBJECTIVES

The major objectives of this project work are listed as follows:

- Reduce the rate of accident
- Provide intelligence to the autonomous driving car
- Develop an alarm system to warn about the road obstacle

1.10 ORGANIZATION OF THE REPORT

The rest of the report is framed as follows. In chapter 2, numerous related works that are related to our proposed systems are presented. In chapter 3, the architecture diagram and all of its components are briefly explained. In addition to it, the hardware and the software components required are also discussed. In chapter 4, the Convolutional Neural Network model is elaborately explained and the sub components of the model are listed. The detailed working and explanation of obstacle detection in autonomous self-driving car are presented in Chapter 4. Chapter 4 also deals about the implementation of alarm system. The performance of the proposed CNN model is experimented and the results are presented in chapter 5. Conclusion and further works are drawn in chapter 6.

CHAPTER – 2

RELATED WORKS

In the design of autonomous driving car, many researches are progressing to optimize the driving process. Vision based approach is mainly used in autonomous driving cars. In vision based approach, Convolutional Neural Network plays a vital role in Image processing.

Deep learning mechanism eases the complexity of many real time problems. Our ultimate motivation is to reduce the rate of road accidents by making the driving process automatic. This can be achieved by providing intelligence to machine. I. Arel et.al (2010) explained the application of deep learning used in various real time application, which paved the way for implementing deep learning in our proposed methodology. Deep learning provides near human like interpretation, decision making experience. Deep learning has shown great promise in recent years in the field of object detection and recognition. Convolutional Neural Networks (CNN) are dedicated to vision-based approaches and they are quite feasible for Graphics Processing Unit (GPU) acceleration in real-time applications. The GPUs, originally designed for 3D modeling and rendering, are now solving classic image processing problems and provide tremendous improvement in speed over CPU-only implementations. GPUs when deployed in the perception system of autonomous vehicles could process video frames at a sufficiently high frame rate and facilitate high-speed driving by detecting the obstacles well before for motion planning to avoid collision.

In the method proposed by Tianmei Guo et.al (2017) a simple Convolutional Neural Network is used to classify images. This forms the basis for all the upcoming developments made in CNN to make it work faster. This method works for detecting single object in an image. But in real time multiple objects tend to appear in a single frame extracted from our input source. M.D. Zeiler et.al (2014) briefly explained that CNNs are some special multi-layer neural networks designed specifically for 2D data, like video and images. The CNNs are motivated by minimal data preprocessing requirements, and they largely receive raw input image and extract features on its own. This process is carried out using R-CNN method which is briefly presented in Bin Liu et.al (2017). This uses sliding window method to find out objects located at different areas in an image. This sliding window method is applied on Region Of Interest (ROI). But this methodology has high computation time and also requires premier hardware components to make the processing speed faster. This is because a single image is scanned a lot of time which depends on the size of the sliding window used. This work is utilized in the application oriented way by Gowdham Prabhakar et.al (2017) where vision based approach is used to identify object in roads using fast R-CNN. Here PASCAL VOC dataset is used for detecting and classification of objects. Since it uses sliding window mechanism the computation power to process the input feed is high. It requires high level GPU and computational power to process.

There are multiple objects present in a single image. The R-CNN mechanism reported earlier is high computational. J. Redmon et.al (2016) designed a methodology to predict the class of multiple objects present in a given image. The image is equally divided and each image slice is processed to find the objects present in it. This approach scans the image only once. Thus the computation time of the image is very less compared with the method discussed earlier.

Our proposed technique reduces the computation power by using J. Redmon et.al methodology in which the image is scanned only once and detects all the objects present in the image with the help of Convolutional Neural Network model (CNN).

A Convolutional Neural Network model is designed to identify the all the objects present in the image. This also includes the co-ordinates of the bounding box associated with each object, the class of the object and the conditional probability of the object. As a result of our implementation the output should be filtered. Our work uses deep learning concepts to design the neural network model and the name is briefly explained in the upcoming chapters given below. Zhang Shan-xin et.al (2010) explain the evaluation metric named Intersection over Union which is used in our proposed work to filter the unnecessary boxes or objects detected. The authors explain the methodology to compute the union, intersection and difference of two polygons (which is square in our case). This filtering system as a whole is called as Non-max suppression mechanism.

The distance between the object and the camera is calculated from the area of the bounding box associated with the object. The warning system is also based on the distance of the object which is briefly explained in chapter - 5

CHAPTER – 3

ARCHITECTURAL DESIGN

3.1 ARCHITECTURE

In this section the architecture diagram of our proposed alarm system for autonomous self-driving car is given. Each component in explained briefly below.

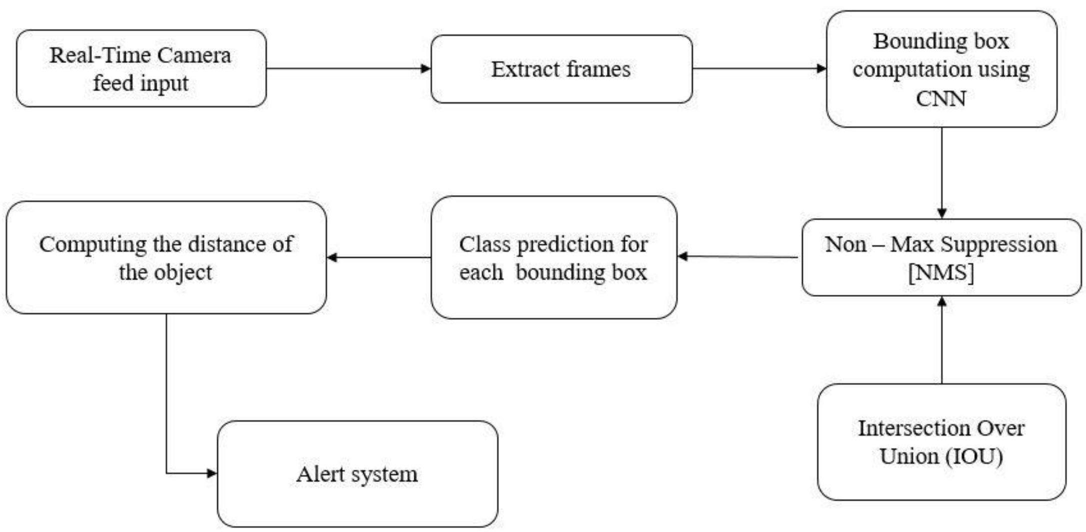

Figure 3.1 Architecture of proposed obstacle detection and alarm system for autonomous self-driving car

3.2 SYSTEM ANALYSIS

3.2.1 INPUT FEED

In this proposed obstacle detection and alarm system for autonomous self-driving car, the input is fed to the system in the form of video. The video consists of many frames. Each frame is separated and processed using this novel method.

3.2.2 COMPUTATION OF BOUNDING BOX

There are many methods to compute the bounding box associated with each object. In this proposed method we use deep learning methodology to compute the co-ordinates of the bounding box. Convolutional Neural Network model is used to compute the bounding box co-ordinates. The learning process for our CNN model is described below.

3.2.3 LEARNING PROCESS

Our Convolutional Neural Network needs training to identify the co-ordinates of the position of the object and also the class of the object. The dataset used for training is COCO dataset. Our model is trained with totally 80 number of classes which includes all the types of vehicles and etc. Our model is specialized for identifying different types of objects that are commonly found in roads and pathways.

As we don't have enough resource locally for computation, we used Google Colab to train our CNN model. Google colab provides free shared GPU and CPU which helps in finishing the training task much faster. Google colab is powered with Nvidia Quadro GPU.

3.2.4 FRAMEWORKS AND LIBRARIES USED

In our work we have used Tensorflow open source framework which is developed by Google to implement Machine learning algorithms easier. We have also used Keras which is an open source library capable of running on top of Tensorflow. Keras is mainly used to design convolutional neural network and it enables us to experiment our design much faster. Matplot library is used to plot the bounding boxes in our image.

3.2.5 NON-MAX SUPPRESSION

After each frame is being processed by our CNN model, there are many boxes associated with a single object. These are nullified by using non-max suppression method. One of the most popular technique viz Intersection Over Union is used.

3.2.6 DISTANCE COMPUTATION

This proposed methodology uses the area of the object to calculate the distance between the object and the car (specifically camera). The area of the object is normalized to a certain limit.

3.2.7 ALARM SYSTEM

This alarm system is designed to notify the autonomous self-driving car to ensure safety when an object comes closer to the car. As area for different object varies for a constant distance, it is necessary to imply specific threshold value for each object class.

3.3 SOFTWARE AND HARDWARE ENVIRONMENT

3.3.1 HARDWARE ENVIRONMENT

Processor	: Any processor above 2.5 GHz
RAM	: 4 GB and above
Hard Disk	: 10 GB
GPU	: AMD Radeon R5 M420 or above
Input Device	: Standard Keyboard, Mouse
Output Device	: VGA Monitor

3.3.2 SOFTWARE ENVIRONMENT

- Jupyter Notebook
- Google colab
- Anaconda prompt
- Google chrome
- Python 3.6

CHAPTER 4

PROPOSED OBSTACLE AND ALARM SYSTEM FOR AUTONOMOUS DRIVING CAR

This chapter briefly explains the complete working procedure of our proposed methodology. The procedure and implementation of all the steps involved in our method are described on this chapter.

4.1 BOUNDING BOX PREDICTION

For our autonomous self-driving car we need a method to identify the objects. A convolutional neural network model is designed using deep learning concept for bounding box prediction. This model is used to predict the bounding box co-ordinates and the Conditional class probabilities ie., The likelihood of the objet class. The detailed processing of the model is discussed below.

4.1.1 NETWORK ARCHITECTURE

The main aim is to build a Convolutional Neural Network model which gives a (19 x 19 x 425) tensor as our output. The model decomposes the dimension of the image to (19 x 19) ie, it splits the model into (19 x 19) cells. The input image has a shape of (608 x 608). Input image of any size and dimension is preprocessed into the spatial dimension of (608 x 608) which is the standard input format for our model.

The model comprises of a total of 23 Convolution layers. Larger the number of layers the more accurate results will be obtained. 80 different types of objects can be identified using our model. Here each bounding box predicts exactly 5 bounding box. There will be a total of 85 parameters for each bounding boxes. We are computing 5 box for each grid cell. Hence a total of about (5 x 85) values are obtained for each grid cell. This explains the output tensor which is of the form (19 x 19 x 425).

The output for each bounding box is of the form

$$Y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

where,

Pc = Conditional Probability

bx= x coordinate, the center of the object corresponding to the upper left corner of the grid cell

by = y coordinate, the center of the object corresponding to the upper left corner of the grid cell

bh = height of the bounding box

bw = width of the bounding box

c1,c2,c3…. c80 = class labels

4.1.2 ACTIVATION FUNCTION

LeakyReLU is the activation function used in every layer of the CNN. LeakyReLU is a variant of ReLU. This helps in resolving the "dying ReLU" problem by having a small negative scope of around 0.01. The function is represented as

$$R(z) = \begin{Bmatrix} z & z > 0 \\ \alpha z & z \leq 0 \end{Bmatrix}$$

where α = 0.01. This helps in obtaining the negative scope.

4.1.3 LOSS FUNCTION

Our model predicts multiple bounding boxes for each grid cell. Hence we need to reduce multiple bounding boxes predicted for a single object. Our model uses sum squared error between the predicted value and the ground truth. The loss function is applied for each parameter in the output tensor of our model. This includes the class, object localization values, confidence score.

4.1.3.1 CLASSIFICATION LOSS

The classification loss ($L_{classification}$) at each cell is the squared error of the class conditional probabilities for each class. It is given by the formula

$$L_{classification} = \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \widehat{p_i}(c))^2$$

where,

$p_i(c)$ = Calculated class probability value

$\widehat{p_i}(c)$ = Ground truth class probability value

4.1.3.2 LOCALIZATION LOSS

The localization loss ($L_{localization}$) calculates the error present in the location and the size between the predicted and ground truth value of the bounding box.

$$L_{localization} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2]$$
$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\widehat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\widehat{h_i}})^2]$$

where,

λ_{coord} = constant

$x_i, y_{i,}$ = calculated x and y axis co-ordinates

$\widehat{x_i}, \widehat{y_i}$ = ground truth x and y axis co-ordinates

w_i, h_i = calculated width and height of the bounding box

$\widehat{w_i}, \widehat{h_i}$ = ground truth width and height of the bounding box

4.1.3.3 CONFIDENCE LOSS

The confidence loss ($L_{confidence\ obj}$) measures the error rate in the likelihood of the object presence in the bounding box. There are two cases. A bounding box may or may not have an object present in it. Thus there are two loss functions for the confidence score of the object.

If the object is present in the bounding box, the formula for calculating the confidence loss is given by

$$L_{confidence\ obj} = \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} 1_{ij}^{obj} \left(C_i - \widehat{C}_i\right)^2$$

where,

C_i = calculated confidence value

$\widehat{C}_i$ = ground truth confidence value

If the object is not present in the bounding box, the confidence loss ($L_{confidence\ noobj}$) can be given as

$$L_{confidence\ noobj} = \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} 1_{ij}^{obj} \left(C_i - \widehat{C}_i\right)^2$$

The total loss function ($L_{totalloss}$) can be expressed completely as

$$
\begin{aligned}
L_{totalloss} = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\widehat{w_i}}\right)^2 + \left(\sqrt{h_i} - \sqrt{\widehat{h_i}}\right)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} 1_{ij}^{obj} \left(C_i - \widehat{C_i}\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} 1_{ij}^{obj} \left(C_i - \widehat{C_i}\right)^2 \\
& + \sum_{i=0}^{S^2} 1_{i}^{obj} \sum_{c \in classes} (p_i(c) - \widehat{p_i}(c))^2
\end{aligned}
$$

where,

λ_{coord} = constant

$p_i(c)$ = Calculated class probability value

$\widehat{p_i}(c)$ = Ground truth class probability value

$x_i, y_{i,}$ = calculated x and y axis co-ordinates

$\widehat{x_i}, \widehat{y_i}$ = ground truth x and y axis co-ordinates

w_i, h_i = calculated width and height of the bounding box

$\widehat{w_i}, \widehat{h_i}$ = ground truth width and height of the bounding box

C_i = calculated confidence value

$\widehat{C_i}$ = ground truth confidence value

4.2 FILTERING OF BOUNDING BOXES

Our Convolutional Neural Network model predicts many number of bounding boxes associated with a single object and bounding box where no object present are also predicted. This leads to confusion and less accuracy of alert timing and has a possibility of leading to false alarm. This can be reduced by using our proposed methodologies as explained in the following subsections.

4.2.1 CLASS CONFIDENCE FILTERING

The scenario where absence of object inside a bounding box may occur while running an image through our model. This leads to false alarm at a high scale. This can be neglected by filtering the boxes based on their confidence value. The confidence value is normalized between 0 and 1. We have applied a threshold value of 0.6. Thus boxes with high probability of object occurrence will be taken.

4.2.2 NON-MAX SUPPRESSION

Even after filtering all the boxes using class confidence filtering there will be lot of overlapping boxes associated with a single object. This can be reduced by using the Non-max suppression methodology. It is a method to evaluate the relativeness of our output with the ground truth. It uses a very efficient function called **Intersection over Union (IoU)**.

4.2.2.1 INTERSECTION OVER UNION (IOU)

Intersection over Union is a function used to compute the relativeness of the predicted output value with the ground truth. It is an evaluation metric used to find out the redundant bounding boxes for the same object.

In this case we are using the co-ordinate value of the top right corner and bottom left corner of the bounding box. The new intersection can be formed by taking the maximum co-ordinate values of the top right corner and the minimum co-ordinate value of the bottom left corner. If the any of the intersection box co-ordinate value falls below zero, it is set to zero as it results no intersection. The area of the intersection area can be calculated by computing the width and height of the intersection area. The union area can be calculated be calculated by summing the area of both bounding boxes and subtracting with the intersection area.

In this project work the IoU area is calculated by taking the ratio of area of Intersection with the area of Union. This gives the level of overlapping of two boxes. Thus if IoU is high then that two bounding boxes represent the same object at a higher scale.

4.2.2.2 IMPLEMENTATION OF NON-MAX SUPPRESSION

Out of many bounding box detected, the one with highest conditional probability is chosen. Then the chosen one is compared with all the remaining bounding boxes and IoU which is previously discussed is used as the evaluation metric here. Thus the IoU value which is higher than the threshold value is neglected as it mostly represents the same block. We have set an threshold value of 0.6. This process is repeated for all the boxes with higher Conditional Probability value. Thus after this, every unnecessary boxes gets filtered.

The algorithm for computing bounding box parameters is presented here under.

Algorithm:

Input: *Camera video frame.*

Output: *Bounding box parameters.*

Begin

Step 1: *Feed the input image into our trained CNN model.*

Step 2: *Compute the bounding box co-ordinates, class of the object and conditional probability using the CNN model.*

Step 3: *Calculate the Intersection over Union (IoU) value for multiple box associated with the same object.*

Step 4: *Apply Non-max suppression to filter multiple bounding boxes.*

End

Next section deals with the warning system implementation and the methodologies proposed in our proposed work

4.3 ALARM SYSTEM

In the previous section we discussed about the processing steps to filter all the redundant bounding boxes. This section deals about the process of producing alarm when the high priority obstacle approaches the autonomous driving car using our proposed methodology. There are few methods used which are discussed clearly below

4.3.1 LANE DETECTION

There are certain scenarios where a false alarm can be produced. Let us consider a scenario where a car is approaching us in the opposite direction in the opposite lane. This rises an alarm which is considered to be false, as a fast moving car should not care about the car driving in the neighboring lane. There are many scenario, where specific objects should be focused less. It is unnecessary to focus on vehicles parked on the roadside or the pedestrians walking on the platform. This can be avoided by detecting the lane in which our autonomous vehicle travel by.

Our live feed video is of the resolution (1280 x 720). In this first few left portion and the last few right portion are considered to be less important. Thus they can be neglected by avoiding those portion while considering the alarm system.

The lane can be detected by reducing the scale of the video to the center area. In our input video feed the x axis portion must be reduced. The x axis co-ordinate scale is reduced from 1 - 1280 to 420 - 850. Thus if the object falls under this portion of the x axis scale, it is further proceeded with the alarm system procedure. Rest of the objects which falls outside the scaled x axis are not considered

4.3.2 MID-POINT DETECTION

The midpoint of the filtered bounding boxes is calculated with the help of co-ordinates of the boxes. The mid-point of X-axis and Y-axis can be calculated by, Mid x = (x1+x2)/2

Mid y = (y1+y2)/2

where x1, x2, y1 and y2 are the co-ordinates of the bounding boxes.

Now the next step is to find whether the mid-point of the object falls in the scaled x axis value.

4.3.3 DISTANCE METRIC

In our proposed methodology we have calculated the distance by calculating the area of the bounding box and relating it with the class of the object.

For example, considering two different objects like car and pedestrian in our scenario, the area of the bounding box associated with the car varies with the area of the bounding box associated with the pedestrian in which the distance between the car (camera) and the two objects is same. Thus each object has a specific threshold value above which the alarm is raised.

4.3.4 WARNING SYSTEM

The separation of high priority object and low priority objects are done manually. The CNN model used in this proposed work is trained nearly with all the possible road obstacles and subjects. Here the high priority and low priority objects are separated manually. The warning system is implied for the only the high priority objects. The warning is overwritten on the image.

The algorithm of our proposed alarm system is presented under

Algorithm for warning system

Input: *Bounding box parameter values*

Output: *Warning results overlaying on the video frame*

Begin

Step 1: *Calculate the mid-point of the bounding box.*

Step 2: *Check whether the mid-point falls in the middle region of the image.*

Step 3: *Compute the distance of the object from the camera using area metric*

Step 4: *Relate it with the threshold value associated with each object class*

Step 5: *If it falls above the threshold range, create a warning overlay above the Image.*

End

4.4 CONCLUSION

Thus the complete working procedure and algorithm of obstacle detection and alarm system for autonomous self-driving car are explained in this chapter. Implementation is done and the experiment and results are presented in the next chapter.

CHAPTER – 5

EXPERIMENTS AND RESULTS

5.1 INTERMEDIARY RESULTS

Our proposed method has been tested for all the frames extracted in a video. This method is tested for more than 40 videos. All the videos are captured through a game. The resolution of the video is (1280 x 720). Each frame is processed individually. The intermediary output obtained during the process of the steps explained in the previous chapter for a single frame is given below.

5.1.1 FRAME EXTRACTION

The input video is captured through a computer game (GTA 5). The resolution of the video is (1280 x 720). The captured video consists of 50 frames per second. In order to process the video faster, the frames per second is reduced to 15. The input image sample is given in figure 5.1.

Figure 5.1 Sample frame extracted from the video

5.2.2 COMPUTATION OF BOUNDING BOX

The input frame is fed into our trained Convolutional Neural Network model. This CNN model detects all the objects present in the frame. The output of the proposed CNN model corresponding to the frame given in figure 5.1 is given in figure 5.2.

Figure 5.2 Result of CNN model corresponding to the input frame shown in figure 5.1

5.2.3 NON- MAX SUPPRESSION

The unnecessary bounding boxes predicted are removed as per the steps we discussed earlier in section 4.2. The output after applying class score filter corresponding to the frame in figure 5.1 is given in figure 5.3.

Figure 5.3 Result of applying class threshold filtering corresponding to the input shown in figure 5.1

The next step is to apply non-max suppression. The results of applying non-max suppression on preprocessed frame shown in figure 5.3 is given in figure 5.4.

Figure 5.4 Result of applying Non-max suppression corresponding to the processed frame shown in figure 5.3

5.2.4 DISTANCE COMPUTATION AND ALARM SYSTEM

The distance computation technique is briefly described in the subsection 4.3.3. The result of distance computation of each and every object located in the frame corresponding to the figure 5.1 is given presented in figure 5.5.

Figure 5.5 The distance computation for each filtered bounding boxes associated with an object corresponding to the frame given in figure 5.1

The class name, precision (likeliness of the object) of the object, bounding box co-ordinates, distance value of each and every object linked with the frame given in figure 5.5 are given in table 5.1

Table 5.1 Displays the result obtained for the input frame shown in figure 5.1 and the output presented in figure 5.5

Class name	**Precision of the object**	**Box co-ordinates**	**Distance Value**
Car	0.86	(491, 348) (776, 522)	2.9
Car	0.66	(804, 331) (1052, 420)	2.5

Figure 5.6 3 Result of Warning display when the object comes closer to the car

Table 5.2 Displays the output for the frame corresponding to the frame shown in figure 5.6

Class name	**Precision of the object**	**Box co-ordinates**	**Area Value**	**Warning**
Car	0.83	(424, 367) (769, 635)	3.4	Yes
Car	0.63	(103, 404) (247, 440)	1.4	No

CHAPTER – 6

CONCLUSION

6.1 CONCLUSION

Automating the driving process using deep learning concept is emerging and many research process are going on to overcome random problems. Our proposed methodology implies a novel mechanism to provide the intelligence of identify the object present in the roadway for the car. The main scope of this method is to reduce the accident rate by enriching the car with the intelligence of obstacle handling techniques. Due to manual driving, lot of accidents happens due to human error. But automated driving cars mimics the driving process of humans with the enrichment of intelligence provided by Machine learning approach. There are many scenarios one should look out while designing the automated self-driving car. In this approach, we have implemented a small part of obstacle detection and alarm system mechanism for automated self-driving car. In the previous methods sensors are widely used to accomplish the obstacle detection process. But our novel approach uses a vision based mechanism to locate and identify the type of the objects present. As sensors are cost effective and hard to implement, this vision based approach accompanied with the machine learning mechanism helps in releasing the knot much easier. Using this method an object present under various circumstances (in daytime, in night time, different angles of the object etc.) are all processed effectively. This method clearly outstands the previous generation mechanisms of handling road obstacles.

6.2 FUTURE WORKS

This can be further developed by including the necessary actions to be performed while a warning is indicated by studying the environment of the driving pathway. In our proposed work we have implemented a two dimensional (2D) bounding box which is less accurate. A three dimensional (3D) bounding box explores a lot of details about the object class and position precisely. Exploring the details of the object leads to feeding the automated driving car with more precise instructions on handling the obstacle once it enters in to the warning zone.

REFERENCE:

- I. Arel, D. C. Rose and T. P. Karnowski, “Deep Machine Learning – A New Frontier in Artificial Intelligence Research [Research Frontier],” *IEEE Computational Intelligence Magazine*, Vol. 5, No. 4, pp. 13-18, Nov. 2010.
- Tianmei Guo, Jiwen Dong and Henjian Li Yunxing Gao, “Simple Convolutional Neural Network on Image Classification”, IEEE 2nd international Conference on Big Data Analysis, pp. 721-724, Sep. 2017
- M.D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks”, *Proceedings of the European Conference on Computer Vision* pp. 45-51 2014.
- Bin Liu ; Wencang Zhao and Qiaoqiao Sun, “Study of object detection based on faster R-CNN” , 2017 Chinese Automation Congress (CAC), 2017 pp: 6233 – 6236
- Gowdham Prabhakar and Binsu Kailath, “Obstacle Detection and Classification using Deep Learning for Tracking in High-Speed Autonomous Driving”, 2017 IEEE Region 10 Symposium pp 752 – 757, 2017
- J. Redmon, et al.,”You only look once: Unified, real-time object detection,” [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: pp 779-788.
- Zhang Shan-xin ; Qiang and Rui-lian “An Algorithm for Computing the Union, Intersection and Difference of Two Polygons”, 2010 Second International Conference on Computer Research and Development, 2010 pp: 344 – 348

- R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 1, pp. 142-158, Jan. 1 2016.
- S. Sivaraman and M.M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 4, Dec. 2013.
- COCO Dataset "http://cocodataset.org/#download"