# SIMULATION OF OBJECT DETECTION AND CLASSIFICATION USING MACHINE LEARNING

by

S. M. IRFAN

MUHAMMAD IFTEKHAR CHOWDHURY

**BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING**

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DECEMBER 2021

# SIMULATION OF OBJECT DETECTION AND CLASSIFICATION USING MACHINE LEARNING

by

S. M. IRFAN

MUHAMMAD IFTEKHAR CHOWDHURY

A thesis/project

submitted as partial fulfillment of the requirement for the degree of

**BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING**

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DECEMBER 2021

# CERTIFICATE OF APPROVAL

The thesis entitled as "**Simulation of Object Detection and Classification Using Machine Learning**" submitted by **S. M. Irfan**, bearing Matric ID. **ET163077** and **Muhammad Iftekhar Chowdhury**, bearing Matric ID. **ET163083** of session **Spring 2021**, to the Department of Electrical and Electronic Engineering, International Islamic University Chittagong, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering and approved for the examination held on **24th December, 2021**.

_____

Supervisor

Engr. Muhammad Jalal Uddin

Assistant Professor

Department of Electrical and Electronic Engineering

International Islamic University Chittagong.

# DECLARATION

It is hereby declared that this work has been done by us and no portion of the work contained in this thesis/project has been submitted elsewhere for the award of any degree or diploma.

_____

S. M. Irfan

_____

Muhammad Iftekhar Chowdhury

# ACKNOWLEDGMENT

All praises and thanks to Allah, the Lord of the world, the Most Beneficent, the Most Merciful for helping us to accomplish this work.

Secondly, we are very much thankful to our supervisor, **Engr. Muhammad Jalal Uddin**, Assistant Professor, Department of Electrical and Electronic Engineering for his excellent advice, motivation, inspiration, and corrections throughout the thesis work, which could not have been accomplished without his cooperation.

We would also want to express our gratitude to the faculty members of the department of EEE for their assistance in this respect. Also, we express thanks to all of the writers and researchers whose work we used to organize the study and write this thesis.

We would like to express our deep gratitude to our parents and friends, who have not only supported us to study but also inspired us with the motivation and support that has helped us to complete this journey of education, particularly this thesis work.

_____

S. M. Irfan

_____

Muhammad Iftekhar Chowdhury

# ABSTRACT

Risk factors of road accidents are increasing around the modern world. It is also applicable to Bangladesh. To reduce road accidents we proposed a system that detects objects on the road. We can implement this system using the available pre-trained model. Unfortunately, most available pre-trained models cannot detect Bangladeshi vehicles shapes and patterns. Thus, Our approach is to make a custom dataset and model from scratch. Based on this dataset and model irregular-shaped vehicles that are available in Bangladesh can be detected. To achieve our proposed method, we have collected Bangladeshi Vehicles Dataset containing 248 images. Based on this dataset a custom model has been built. We tested another 82 images which are distinct from the dataset to verify the accuracy of the custom model. We got a roughly 65 percent accuracy rate to detect these native vehicles varying the range from 91 - 37 percent. Thus, this system can be implemented as a road collision avoidance system in developing countries like Bangladesh.

# TABLE OF CONTENTS


**CERTIFICATE OF APPROVAL** ii

**DECLARATION** iii

**ACKNOWLEDGEMENT** iv

**ABSTRACT** v

**TABLE OF CONTENTS** vi

**LIST OF FIGURES** x

**LIST OF TABLES** xii

**LIST OF ABBREVIATIONS** xiii

**CHAPTER 1 INTRODUCTION** 1

1.1 Introduction 1

1.2 Background 2

1.3 Motivation 3

1.4 Why do We need this system? 3

1.5 Objective of the Thesis 4

1.6 Advantages of object detection 4

1.7 Thesis outline 4

**CHAPTER 2** 6

2.1 Introduction 6

2.2 Review of previous works 6

2.2.1 Previous works 7

2.2.2 The Haar cascade technique is used to identify vehicles 7

2.2.3 FPGA-based Automatic Reverse Braking System 8

2.2.4 Obstacle detection using the infrared sensor 8
2.2.5 Several sophisticated technologies facilitate braking 8
2.2.5.1 Autonomous Emergency Braking (AEB) system 9
2.2.5.2 Anti-lock braking system (ABS) 9
2.2.5.3 Electronic Brake Force Distribution (EBD) 10
2.3 Computer Vision 11
2.3.1 Machine Learning 12
2.3.2 OpenCV 13
2.4 Related Works 13
2.5 Worldwide research on Vision-based intelligent vehicle 14
2.6 Human Detection 14
2.7 Avoiding Obstacles 15
2.8 HOG Detection 16
2.9 YOLO as a Real-Time Object Detection System 17
2.9.1 YOLO vs. other detectors 18
2.9.2 YOLO constraints 19
2.9.3 How does YOLO perform? 20
2.10 Convolution Neural Network (CNN) 21
2.11 Global status report on road safety issued by World Health Organization (WHO) 21
2.12 In Bangladesh, a lack of qualified drivers contributes to fatal traffic accidents 22

**CHAPTER 3 METHODOLOGY** **26**

3.1 Introduction 26

3.2 Soft Detection 26

3.2.1 Soft Detection Diagram 28

3.3 Data Pre-processing 28

3.3.1 Data augmentation 29

3.3.2 Data annotation 29

3.4 Dataset handling 29

3.5 Model Training 31

3.5.1 Model training hyperparameter 32

**CHAPTER 4 SIMULATION** **34**

4.1 Simulation environment 34

4.2 Simulation of detection of different objects 38

4.2.1 Pedestrian detection 38

4.2.2 Bus 39

4.2.3 CNG Taxi 39

4.2.4 Rickshaw 40

4.2.5 Bicycle 41

4.2.6 Car 42

4.2.7 Truck 42

**CHAPTER 5 RESULT and DISCUSSION** **43**

5.1 Class ID detection 43

5.2 Custom Model accuracy 44

5.3 Dataset comparison 45

5.4 Advantage of this system 46

5.5 Limitations of this system 46

**CHAPTER 6 CONCLUSION 48**

6.1 Conclusion 48

6.2 Future scopes of work 48

**REFERENCES 49**

**APPENDIX 56**

# LIST OF FIGURES


| | | |
|---|---|---|
| Fig. 2.1 | Anti-Lock Braking System (ABS) | 10 |
| Fig. 2.2 | EBD applies extra braking force to front wheels | 11 |
| Fig. 2.3 | Original HOG algorithm flowchart | 16 |
| Fig. 2.4 | Previous work of HOG feature extraction processor's | 17 |
| Fig. 2.5 | Detectors with one or two stages | 18 |
| Fig. 2.6 | Single object detection | 19 |
| Fig. 2.7 | Multiple object detection | 20 |
| Fig. 2.8 | Global status report on road safety issued by WHO, 2018 | 22 |
| Fig. 3.1 | Detection Method | 26 |
| Fig. 3.2 | Soft Detection | 28 |
| Fig. 3.3 | Dataset handling | 30 |
| Fig. 4.1 | Computer Vision as Object Detection algorithm | 34 |
| Fig. 4.2 | Simulation of model training | 35 |
| Fig. 4.3 | Simulation of model training | 35 |
| Fig. 4.4 | Simulation of image processing | 36 |
| Fig. 4.5 | Simulation of real-time video processing | 36 |
| Fig. 4.6 | Simulation of real-time video processing | 37 |
| Fig. 4.7 | Simulation output | 37 |
| Fig. 4.8 | Pedestrian Detection as a person | 38 |
| Fig. 4.9 | Bus with bounding boxes | 39 |

| | | |
|---|---|---|
| Fig. 4.10 | CNG Taxi with bounding boxes | 40 |
| Fig. 4.11 | Rickshaw with bounding boxes | 41 |
| Fig. 4.12 | Bicycle detection with a person | 41 |
| Fig. 4.13 | Car with bounding boxes | 42 |
| Fig. 4.14 | Truck with bounding boxes | 42 |
| Fig. 5.1 | Separately detected objects on YOLO and custom model | 43 |
| Fig. 5.2 | Detection accuracy of the custom model | 45 |

# LIST OF TABLES


| | | |
|---|---|---|
| Table 2.1 | Road accidents rate in Bangladesh from 2004 to 2008 | 23 |
| Table 3.1 | Image allocation of different classes in the dataset | 31 |
| Table 3.2 | Model training hyperparameters | 33 |
| Table 5.1 | Detection accuracy of Custom Model | 44 |
| Table 5.2 | Dataset comparison | 46 |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| WHO | World Health Organization |
| UBC | University of British Colombia |
| RADAR | Radio Detection and Ranging |
| LiDAR | Light Detection and Ranging |
| IR | Infrared Radiation |
| CV | Computer Vision |
| IPP | Integrated Performance Primitives |
| MLL | Machine Learning Library |
| EMI | Electromagnetic Interface |
| FPGA | Field Programmable Gate Array |
| AEB | Autonomous Emergency Braking |
| ABS | Anti-lock Braking System |
| EBD | Electronic Brake Force Distribution |
| AI | Artificial Intelligence |
| NURBS | Non-Uniform Rational B-Splines |
| IP | Internet Protocol |
| USB | Universal Serial Bus |
| PROMETHEUS | Program for European Traffic with Highest Efficiency and Unprecedent Safety |
| BRTA | Bangladesh Road Transport Authority |
| BLAST | Bangladesh Legal Aid and Services Trust |
| RGBD | Red Green Blue Depth |
| BGR | Blue Green Red |
| YOLO | You Only Look Once |
| COCO | Common Objects in Context |

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Human existence has gotten more pleasant and effortless as new technologies emerge in several fields of research. The advancement of embedded technology in the automobile sector makes human life safer and more convenient. Over the years, the general public, governments, and the automobile industry have all been interested in automotive safety. Vehicle accidents are one of the biggest disasters that can strike at any moment and in any location. According to statistics from the World Health Organization (WHO), road deaths are the leading cause of death worldwide. Annually, over 1.2 million people die on the world's roadways, 20% of them are killed or injured as a result of aggressive driving or poor road conditions [1]. Almost everyone owns a car nowadays, raising the number of vehicles on the road and raising the number of accidents and road collisions each year [2]. Accidents are typically caused by the following factors:

1) A complicated and unpredictable road environment.
2) Excessive speeding of vehicles.
3) Driving while intoxicated.
4) Driver distractions.
5) Avoiding safety equipment such as Seat Belts and Helmets.

Numerous technologies are being created to assist drivers, and these technologies should provide drivers with a safer environment with a monitoring system, however owing to infrastructure costs, almost all of these technologies are only accessible in luxury automobiles. Ileri, K, et al. developed In 2016, a set of lane identification algorithms and a distance evaluation method were created to recognize lanes and measure distances for a driving automobile [2]. Alpar et al. proposed the Corona method in 2016 [3]. To avoid a forward accident, this approach might differentiate the vehicle's brake light from those other light sources. Prior, in 2015, Garethiya et al proposed a predictive vehicle collision avoidance system using Raspberry Pi and ultra-sonic sensors [4]. A team of researchers at the University of British Columbia (UBC) in Canada has developed a new technique for

detecting vehicles and estimating inter-vehicle distances. In 2017, Huang et al published a paper on how to use cameras to detect vehicles and their distance from each other using a single lens [5]. To reduce traffic accidents and the loss of human life, an accident alarm system is proposed. This is an automatic system that does not require any human involvement to fulfill its tasks. The system should be secure, accurate, simple, and inexpensive. This type of technology may be installed in any vehicle to assure the driver's driving safety as well as the rider's walking safety. This thesis describes the objects detection such as vehicles and pedestrians using Machine Learning Library (MLL) and OpenCV, by following this process vehicles can detect obstacles and it reduces accidents.

### 1.2 Background

In computer vision systems, object recognition is essential. Video surveillance [6], medical image processing [7], and robotic control [8] are just a few of the uses. Background removal, temporal differentiation, support vector machine, optical flow, contour fitting [9], and Kalman filtering are some of the approaches that may be used for this. Besides the approaches mentioned above, Convolutional Neural Networks (CNN) is the most modern development to object identification. Breakthroughs in image categorization began when Alex [10] Deep convolutional neural networks (DNN) were used to win the 2012 ImageNet competition. In the ImageNet [11] competition, they trained a Deep CNN to recognize 1.2 million high-resolution pictures with over 1,000 classifications. They outperformed earlier state-of-the-art models in terms of accuracy. As a result, several academics became interested in developing a unique method for developing efficient deep convolutional neural networks. In publication [12], To tackle object detection with minimum training examples, the authors developed a low-shot transmission detection method employing a configurable deep learning model and a regulated transfer learning framework. For object detection, another study [13] presented a gating network and a range decision-making network. The regional identification network tells you where to look for locations to learn attributes from. The gatting network, on the other hand, functions as a feature generator that modifies feature maps locally. For visual detection and tracking [14], convolutional neural networks were utilized. They constructed an ad hoc huge dataset comprising positive and negative instances

of framed objects from the ImageNet database and used Alex Net [10] architecture to identify the most promising patch. Active learning is another way of detecting objects [15]. This is a category of image classification techniques that search for the most useful examples to include in a training set. Finally, based on form and color sequence comparison, artificial neural networks were employed to detect things.

### 1.3 Motivation

Image/vision innovation is also implemented for detection, tracking and safety [16]. It is the most difficult duty since quick processing is required to inform the driver as soon as feasible. Every frame in which a pedestrian appears must be detected [17]. However, the system based on image/vision technology has certain disadvantages. The system breaks down in several unfavorable weather conditions, such as smog, severe and intense wet conditions. When attempting to distinguish between shadows and pedestrians, the system occasionally generates an error. The system necessitates high-resolution cameras, and implementing such a system is a challenging undertaking since it creates errors owing to vehicle dampening and vibrations. Ultrasonic sensors and radar sensors have recently been employed in vehicle obstacle detection systems [18],[19]. Because of its extended detectable range and greater reliability, the driver monitoring assistance system employs RADAR [20]. LiDAR sensors [21],[22] are used to scan road borders. Also, they are used for detecting any types of obstructions, and a safe vehicle path is created by them. Obstructions and road limits are detected using downward-facing LiDAR sensors. Moreover, such a system consumes more energy and is, therefore, more costly to set up. Thus, Radar systems are the best solution for detecting obstacles.

### 1.4 Why do We need this system?

As previously said, this automatic braking technology is in high demand to boost life-saving efforts.

- A significant benefit is the cost problem. It is rather inexpensive. So many of these may be used anywhere they are needed.
- It protects those in need of rescue and can detect impediments in its path.

### 1.5 The Objective of the Thesis

Our work has various objectives, which are as follows:

- To make a custom model for Bangladeshi roads & vehicles using RCNN.
- Training the custom model using Tensorflow machine learning library.
- To verify the accuracy of our trained custom model.

### 1.6 Advantages of the object and pedestrian identification

- Object detection enables us to identify and locate the objects in an image or video.
- It can be used to determine and monitor the object and pedestrian's precise positions.
- Also, this detection allows us to precisely classify and label them using this type of identification and localization.
- It may be used to count objects and pedestrians in an environment such as a road also.
- Object and pedestrian detection can be used to reduce road collisions.
- It can be also used for managing the road traffic system.

### 1.7 Thesis Outline

We discussed and divided our thesis in six chapters, and the following are the outlines of our report:

- The first chapter is the introduction. This introduction part of this system has been covered in this chapter.
- The second chapter provides a survey of the literature, and some past work those are relating to this system is reviewed.

- The third chapter described the methodology and the technique of the approach, which includes the way of developing the entire system as well as the essential flow charts, block diagrams and programming techniques.
- The fourth chapter is all about simulation techniques. How this system can be simulated is discussed in this chapter.
- The fifth chapter is titled as result and discussion. and it discusses the outcome and objective verification of our system.
- The sixth Chapter contains a conclusion and recommendations for further study.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

Road accidents have been one of the most severe problems in today's world, and the majority of them are the result of carelessness and stupidity on the part of drivers. Drivers must be alert of a number of elements, such as vehicle direction and velocity, passing automobiles, vehicle placement, and potentially harmful or unusual objects ahead. People have been exploring and investigating ways to construct a driving-assistance system in order to increase driving safety. If a driver assistance system could collect information about the surrounding environment in advance, it would significantly lessen the stress of driving for drivers while also making driving safer and easier. Advanced driver assistance systems (ADAS) are technologies designed to automate car systems for improved safety and driving performance. Safety precautions are intended to prevent crashes and accidents by providing technologies that warn the driver of possible hazards or by installing safeguards and taking control of the vehicle. Object tracking has a direct influence on driver safety, and damage may be quickly caused owing to their ignorance. Vehicle detection has been examined using a variety of sensors including RADAR, Li-DAR, sensor fusion, and monochromatic vision. Vision-based vehicle detection approaches have become a trendy topic in the field of research because of their availability, low-cost, high-performance cameras and practical computer vision technology.

## 2.2 Review of previous work

We looked at various existing works involving human detection, object detection, Model training and machine learning related works, automated braking vehicles, rescue vehicles, various monitoring vehicles and so on. The following sections will provide an in-depth analysis of the related works.

### *2.2.1 Previous works*

Collision Avoidance System (CAS) is a vehicle safety system that is meant to minimize the likelihood and severity of an accident. These approaches have progressed from simple systems to cognitive computing that is now being utilized and researched. To detect an impending accident, they employ electrical circuitry in conjunction with distance sensors and, occasionally, video sensors. Once detected, these systems can alert the driver of an impending accident or act autonomously by applying the brakes without any human input [23].

The method employs two ranged sensors, one for the forward end and another one for the backward end. When a minimum distance separation is achieved, it detects obstructions. However, the system is restricted to detecting obstructions and emitting warning signals; no countermeasures are automatically activated to prevent crashes.

In addition, [4] makes use of the Raspberry Pi a widely used microcomputer and an ultrasonic sensor detecting potential to estimate the distance between moving and fixed objects. The sensors are installed to detect obstacles in front of the vehicle as well as obstacles in the vehicle's blind area. There is no anti-collision mechanism in the system.

### *2.2.2 The Haar cascade technique is used to identify vehicles*

The road detection method is a difficult system to implement since there are several types of road incidents. The method is used to determine the best drivable area of the roadways. This method is used to find the road surface and display the vehicle's drivable zone. The HSV algorithm is used to recognize roads. A digital picture is divided into numerous parts known as super-pixels. The provided image is transformed into a super-pixel image then partitioned. Each pixel is colored to the user's preference. The partitioned pixels are separated, and the required pixel is used as an input to the HSV color space. The appropriate pixel color is retrieved, and the road in the image is identified [24].

### *2.2.3 FPGA-based Automatic Reverse Braking System*

The suggested auto-braking system is made up of obstacle sensors installed in the automobile that identify the obstruction's reversing route and transmit a signal to the FPGA. Its output of the object sensor is analog, but the FPGA wants data in digital form, therefore we need a circuit called the Sensing circuit. The sensing circuit will transform analog information into a digital format. That is, the analog output of the sensor will be converted to digital and sent to the FPGA. As a sensing circuit, ADC is utilized. The behavior of the FPGA is defined by VHDL, which works as a controller logic and is created with the help of FSM, which will detect the object based on the digital input and take appropriate action. Switching circuits, such as tail lights or LEDs, supply the output. In practice, the final output is linked to a mechanical device, which is linked to the vehicle's braking system, which is in charge of applying brakes, giving rise to the idea of automated breaking [25].

### *2.2.4 Obstacle detection using the infrared sensor*

Earlier research on obstacle detection involved the use of infrared sensors [26], which have been widely utilized as sensing devices for avoiding obstacles. Because an infrared sensor has nonlinear behavior and the main notion is based on reflection from a moving object, it causes inaccuracy in the estimated distance. As a result, these sensors were untrustworthy for accurate readings. As a result, these sensors are only appropriate for measuring small distances up to 25 cm.

### *2.2.5 Several sophisticated technologies facilitate braking*

The first law of motion, by Sir Isaac Newton, gave birth to the creation of an automobile's braking system because an automotive vehicle requires not only a power source but also an effective braking system, since the more the horsepower, the greater the brake force necessary to stop or de-accelerate that vehicle. This concept sparked several studies in the subject of braking, resulting in its evolution, which allows us to choose an appropriate brake system based on our needs today. A brake system is a collection of numerous couplings and

equipment designed in such a way that they convert the vehicle's kinetic energy into thermal energy, which then slows or stops the vehicle. The frictional pressure formed by the frictional contact between the brake shoe and the moving drum or disc of the braking system converts kinetic strength to thermal strength.

*2.2.5.1 Autonomous Emergency Braking (AEB) system*

It is also an automated streetcar safety process that enables sensors for monitoring the closeness of cars in front of it and recognizes instances in which the velocity vector and range between both the host and intended vehicles indicate that an accident is imminent. In such a case, emergency brakes can be automatically deployed to avoid or at least reduce the impact of a collision. According to recent research, if all automobiles were equipped with the technology, it could have reduced crashes by up to 27 percent and saved up to 8,000 lives each year [27].

*2.2.5.2 Anti-lock braking system (ABS)*

ABS is an automobile safety system that enables a vehicle's wheels to retain torque multiplication contact with the ground in response to driver inputs when braking, keeping control of the vehicle and avoiding uncontrolled sliding. It's a computerized system that employs the cutoff and frequency braking techniques utilized by advanced users with older brake systems. It does it significantly more quickly and with effective coordination than a person could. ABS [27] improves vehicle handling and shortens stopping distances on dry and slick roads. Anti-lock braking performance is one of the most essential factors influencing vehicle safety. **Fig. 2.1** depicts the Anti-Lock Braking System (ABS). Stopping distance is a critical measure for assessing a road vehicle's overall braking ability. The following are the primary parameters that influence brake stopping distance:

1. The vehicle's weight.

2. The vehicle's speed.

3. Road adhesion coefficient.

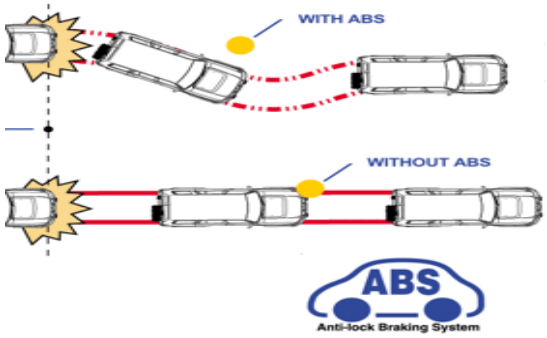


**Fig. 2.1:** Anti-Lock Braking System (ABS).

*2.2.5.3 Electronic Brake Force Distribution (EBD)*

EBD helps cars to stop at short ranges by dispersing braking force based on the vehicle's weight distribution [28]. Each tire of an automobile has a distinct load due to the uneven distribution of weight. Vehicle dynamics also influence the strain that a tire must handle. When braking on a horizontal plane, the vehicle's weight changes from the back to the front. When braking into a turn, on the other hand, the pressure transfers outside in accordance to the turning. The more pressure a wheel can support, the better its grip will be and the greater braking power it will have. The EBD makes use of this physical reality. The EBD changes the proportion of front brake forces on the left and right to enhance the stopping effect. In **Fig. 2.2** we can see that the front wheels take a majority portion of force than the back wheels. Wheel velocities are monitored closely, and if tire slide owing to low pressures is observed, deceleration power on heavier laden tires is intensified. EBD is an upgrade over ABS in that it employs the same elements as ABS and hence requires just a change in the algorithm to generate an EBD braking system.

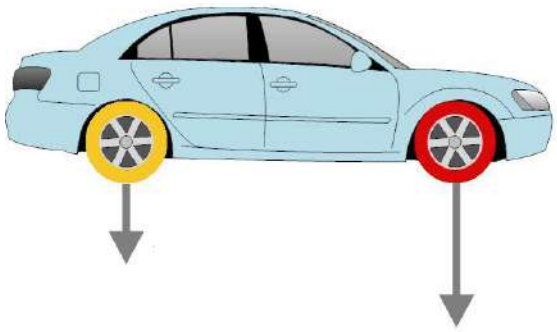

**Fig. 2.2:** EBD applies extra braking force to front wheels.

### 2.3 Computer Vision

Computer vision is a subject of investigation that focuses on assisting computers in decoding high-level data from electronic images and movies [29]. It discusses methods for gathering, organizing, and analyzing high-level data in order to synthesize symbolic data. Pattem recognition and reinforcement learning are aided by artificial intelligence, which is a part of computer vision. In the medical sciences, industry, and the military, computer vision has a wide range of uses. Computer vision is being used in autonomous automobiles, which is a recent application field. In the independent sections of tasks, vision is provided in a variety of ways. Some systems employ web cameras, but others prefer IP cameras, which are ready to use and require just placement. It is unusual to see a Raspberry Pi being used to provide computer vision given its processing capabilities. We've gone through some of the work that combines computer vision and image processing. In general, real-time image processing is a complicated operation that necessitates a significant amount of computing power. Making the Raspberry Pi competent at processing real-time pictures and videos and delivering computer vision, on the other hand, make the system cost-effective [30]. The Raspberry Pi is utilized for educational reasons, with students being required to create a system that includes a raspberry pi and image processing technology. The unique elements of this work that make it stand out are its energy-saving and low-cost configuration. The version they utilized is not disclosed, even though Model B is referenced in the report. Image processing languages include Octave and Python. For the pupils, they created a raspberry pi-centric microcomputer and a digital signal processing study. Various methods are used to detect the

driver's sleepiness [31]. Several of them are steering pattern monitoring, which mainly uses input from an electric operated steering system, vehicle position in lane observing, which employs a lane monitoring camera, and driver face monitoring [32], which employs computer vision to detect the driver's face using an external USB camera or a built-in camera. Body sensors are employed in physiological measuring systems to monitor the activity of body components [33]. This one demonstrates the usage of RPI in surveillance and monitoring systems in conjunction with computer vision [34]. A tracking system and motion detection are part of the surveillance progression. With the help of the Raspberry Pi, Simple CV is utilized to offer computer vision. It aids in the detection of motion. This causes lights to turn on to guarantee that the movement is captured, as well as the camera's footage to be streamed online. There is a mention of the MPJG streamer to help with feed streaming. There is no mention of the type of camera utilized.

### *2.3.1 Machine learning*

Machine learning is a growing field of computational algorithms that aim to simulate intelligence by learning from its surroundings. They are regarded as the workhorses in the new era of so-called big data. Recognition systems, computer vision, aerospace engineering, finance, entertainment, and computational biology, as well as biological and medicinal applications, have all successfully employed machine learning techniques. In advanced stages of local illness, electromagnetic radiation (radiation therapy) is utilized to treat more than half of cancer patients and is the major treatment approach. Radiation therapy is a complicated set of actions that not only cover the time between consultation and treatment, but also go above and beyond to ensure that patients have received the specified amount of radiation and are reacting appropriately. The sophistication of these procedures can vary and may involve multiple phases of sophisticated human-machine interactions and decision-making, which would naturally welcome the use of machine learning algorithms to enhance and digitalize these processes, such as radioactivity physical quality control, contouring and therapeutic interventions planning, image-guided radiation therapy, breathing motion management, treatment response model construction, and outcome prediction. Machine learning algorithms' capacity to learn from the present environment and adapt to already

experienced tasks should lead to improvements in radiation safety and efficacy, as well as improved outcomes [35].

### *2.3.2 OpenCV*

OpenCV is a free and open-source library for visual recognition tasks such as video and image processing. OpenCV was created with a significant emphasis on real-time applications and computational performance in mind. If the proper IPP library runtime is installed, OpenCV will utilize it automatically. OpenCV's main goal is to provide straightforward visual recognition, classification, labeling, identification approach that helps engineer to build sophisticated systems. Machine Learning Library (MLL) is highly useful and integrated for computer vision (CV) related works.

## 2.4 Related Works

Vehicle detection and vehicle light detection research may be separated into two categories: sensor-based detection methods and vision-based detection approaches. Vehicle light detection may be seen as a vehicle detection method to some extent.

A common way for vehicle detection is to employ sensors such as laser beams, millimeter-wave radar systems or optoelectronic devices. The first two are referred to as active sensors, whereas optical sensors are referred to as passive sensors [36]. While active sensors have yielded encouraging results, there are significant disadvantages, including poor scanning speed and low spatial resolution. A significant issue is the expensive expense of a laser or radar. When compared to active sensors, the optical sensor provides a cost advantage of 70%. Optical sensors may be used to monitor cars, detect lanes, and identify objects. Certain complex circumstances are difficult to manage with optical sensors, such as lighting changes, complex outside settings, and crowded backgrounds [37]. Vehicle detection using visual techniques is a difficult challenge. Many studies on vehicle detection have been conducted. In article [38], for example, HOG characteristics were employed for vehicle detection. To accomplish vehicle detection, several characteristics including rectangle

features and a Gabor filter, were provided [39]. To detect automobiles, researchers exploited shadow qualities of a specific region of interest, such as Histogram equalization and entropy. Because the car in front does not always appear in the default zone of interest in a dynamic environment, this approach has the problem of not explaining how the zones of interest are set in the input frame.

### 2.5 Worldwide research on Vision-based intelligent vehicle

Many public-funded and private institutions throughout the world have launched numerous initiatives with the eventual objective of creating autonomous cars, including a large number of research units working together. These efforts have resulted in several prototypes and solutions based on a variety of methods [40]. The PROMETHEUS program was a forerunner in this field in Europe (Program for European Traffic with Highest Efficiency and Unprecedented Safety). More than 13 automakers and many research institutes from 19 European countries were involved in the project. Several prototype vehicles and systems (VaM0Rs, VITA, VaMP, MOB-LAB, and GOLD) were developed as a result of this research. Despite the fact that the first attempts to develop intelligent automobiles were made in Japan in the 1970s, significant research activity in Europe began in the late 1980s and early 1990s. By working on the "Personal Vehicle System" project, MITI, Nissan, and Fujitsu became pioneers in this industry [41]. The Advanced Cruise-Assist Highway System Research Association (AHSRA) was formed in 1996 by vehicle manufacturers and a significant number of research institutions [40].

### 2.6 Human Detection

There is an abundance of human-detection methods. We studied a few research papers to get ideas. [42] It is a system in which they develop a wireless sensor network capable of detecting human falls. This procedure is complicated. Because we intend to detect a human in an area where a human cannot go, As a result, the human detection mechanism must be compact and simple to connect with the Raspberry Pi. [43] Some people use the radar sensor to detect humans. The working principle and procedure are excellent. However, we are

unable to locate this radar sensor in Bangladesh. As a result, we must skip the radar sensor section. [44] Another method of person detection is the use of an RGBD camera. However, adopting the strategy necessitates the implementation of CNNs (Convolutional Neural Networks). As a result, the working procedures will become more complex, and managing a camera will be extremely difficult for us [45]. The usage of passive infrared (PIR) sensors is quite beneficial. Because the sensor is widely available, inexpensive, and simple to locate. The functioning procedure of passive infrared (PIR) sensors is particularly useful since they sense human temperature. As a result, locating the human is much simpler.

### 2.7 Avoiding Obstacles

Many people avoid barriers in a variety of ways. [46] They are employing a touch sensor capable of detecting motion as well as a few other things. They are utilizing a six-dimensional force/torque sensor to detect the collision, which can sense the torque of the robot. Researchers are employing a fuzzy logic-based system with sensors [47]. Where they use the sensor to determine the position of the thing. Another method for detecting avoidance is using non-uniform rational B-splines (NURBS). [48] By employing this, they also find the correct path from source to destination. [49] Ultrasonic sensors are extremely handy for avoiding obstacles. This sensor sends a wave on one side and receives it on the other. A hardware platform chapter describes the ultrasonic sensor's operation. [50] It employs a Kinect distance sensor, which costs approximately 10,000 Tk, for obstacle avoidance. Robo avoids the same problem by employing ultrasonic sensors, which cost 400 Tk for three sensors. In comparison, our system is significantly more practical. Furthermore, it is incapable of detecting humans at all. Furthermore, it may be used to detect vehicles, animals, and other objects. On the contrary, ours can be used in any outdoor rescue effort to detect impediments.

## 2.8 HOG Detection

**Fig. 2.3** depicts an object detection flow diagram using the original HOG technique [51].

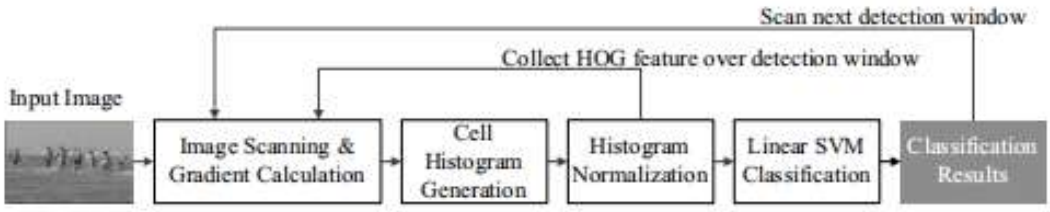


**Fig. 2.3:** Original HOG algorithm flowchart.

Real-time object detection is becoming increasingly important in a range of fields, including monitoring, transportation systems, and robots. Because it is immune to light changes and achieves great analytical accuracy in recognizing objects with varied textures, the Histograms of Oriented Gradients (HOG) [51] approach is frequently employed in object identification systems.

With a lot of computing work, newer high-performance general-purpose computers can detect things in real time. The processor, however, uses a great deal of power and is hence inappropriate for mobile devices with short battery life. As a result, to broaden the range of applications, a high-performance, low-power HOG extracting features machine is expected.

**Fig. 2.4** depicts the image resolution vs frame rate for multiple published HOG hardware descriptions. Zhang et al. [52] proposed employing GPGPU for efficient object detection. For real-time applications, some FPGA implementations [53]–[57], and an FPGA-GPU architecture [58] have been proposed. Cao et al. [59] achieved the best performance of any FPGA implementation when compared to other implementations. This study, on the other hand, focuses on stop-sign identification in particular. HOG features can be applied to a wide range of applications. As a result, next-generation HOG feature extraction processors must be more expandable and perform better.

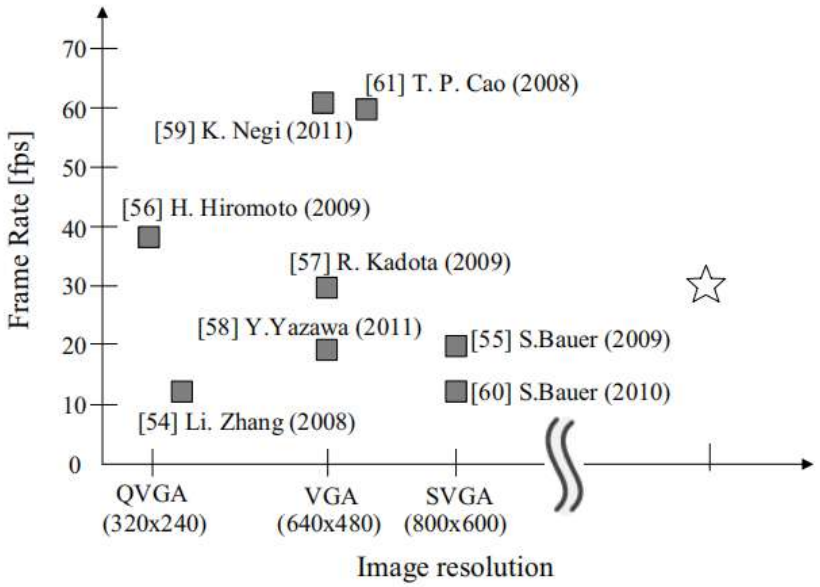


**Fig. 2.4:** Previous work of HOG feature extraction processor's [52].

## 2.9 YOLO as a Real-Time Object Detection System

Object detection is a more sophisticated kind of image classification in which a neural network predicts items in an image and highlights them with bounding boxes. Object recognition is the process of recognizing and localizing items in an image that belong to predefined classes. In **Fig. 2.5** shows different stages detectors.

Object detection in two stages,

The usage of techniques that divide the object detection problem statement into two phases is referred to as two-stage object detection.

1. Recognizing potential object zones.

2. Diverging the image into object classes in certain places.

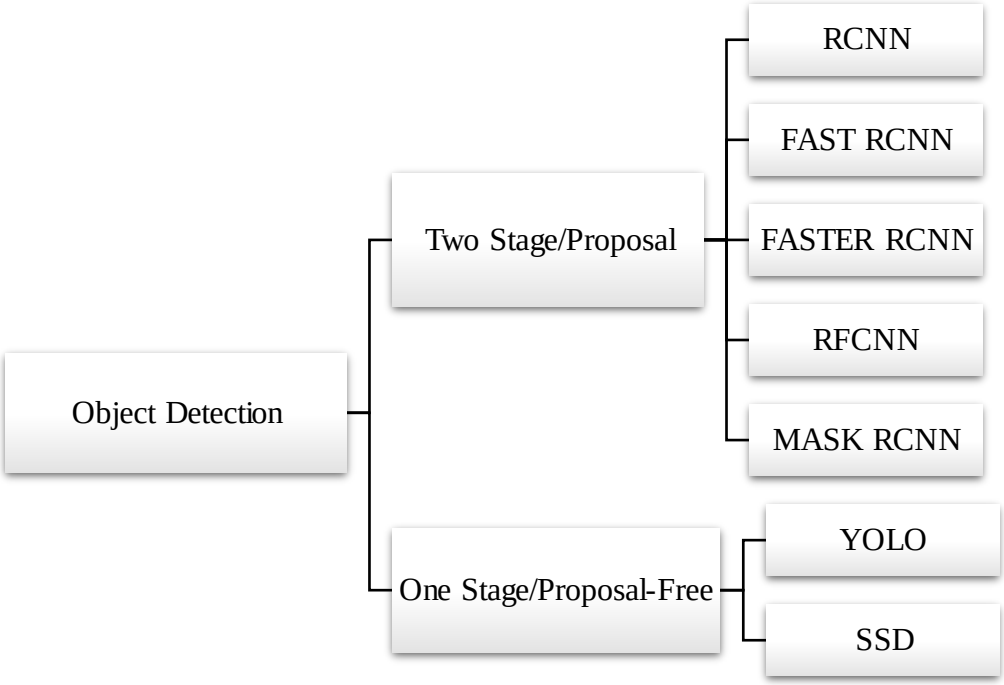


**Fig. 2.5:** Detectors with one or two stages.

Popular two-step algorithms, such as Fast-RCNN and Faster-RCNN, often employ a Region Proposal Network, which suggests regions of interest that may contain objects.

### *2.9.1 YOLO vs. other detectors*

In addition to increased prediction accuracy and a better intersection of the union (when compared to real-time object detectors) in bounding boxes, YOLO has the apparent advantage of speed. YOLO is a much quicker algorithm than its competitors, reaching speeds of up to 45 frames per second [60].

Here's how YOLO actually works in practice in **Fig. 2.6**.


**Fig. 2.6:** Single object detection.

YOLO executes all of its predictions using a single fully connected layer, whereas Faster RCNN uses the Neural Network Approach to uncover prospective feature points and then separately detect those sections.

As a result, methods that employ Region Proposal Networks must run several iterations for the same picture, whereas YOLO simply needs one.

### *2.9.2 YOLO constraints*

Although YOLO (You Only Look Once) appears to be the greatest technique for solving object detection problems, it has significant limitations. Because each grid can detect only one object, YOLO fails to recognize and separate small objects in photos that appear in

groups. Small objects that occur in groups, such as a line of ants, are so difficult for YOLO to identify and localize. YOLO also has lesser accuracy when compared to significantly slower object identification techniques such as Fast RCNN.

### *2.9.3 How does YOLO perform?*

In YOLO, the image is divided into N grids, some of which contain an nxn equivalent segment, using the YOLO approach. Each of the N grids is in charge of identifying and finding the element it holds. The parameters of the bounding box in respect to its cell parameters, as well as the component name and the likelihood that the object is present in the cells, are predicted by these grids. This approach saves time and effort by handling both cell detection and recognition from an image, but it results in a huge number of duplicate predictions since numerous cells forecast the same item with various bounding box forecasts [60]. To address this issue, YOLO employs non-maximum possible suppression.

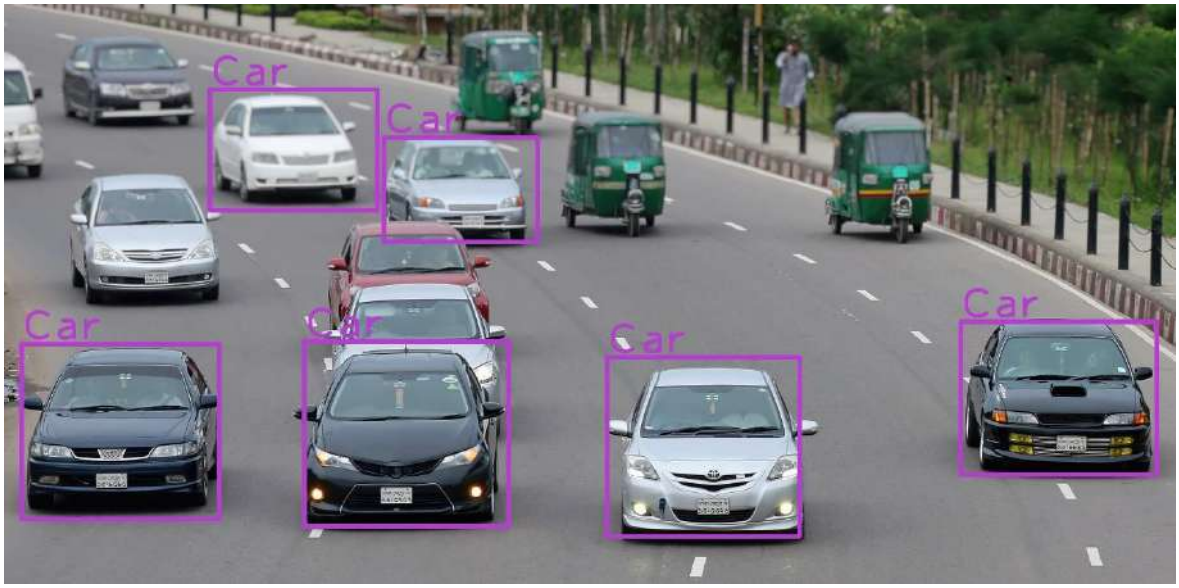


**Fig. 2.7:** Multiple object detection.

When not maximally suppressed, YOLO suppresses all connected components with decreasing confidence intervals. YOLO does this by assessing the posterior distribution associated with each option and selecting the one with the best chance. After that, the bounding boxes with the largest intersection with the high probability structuring element are suppressed. This technique is done until the boundary boxes that are wanted are achieved.

### 2.10 Convolution Neural Network (CNN)

A CNN, or Convolutional Neural Network, is a deep learning neural network that analyzes ordered arrays of data like photos. Convolutional neural networks are widely employed in computer vision and have advanced to the state-of-the-art in a variety of visual imaging applications, including localization and natural language processing for text categorization. Lines, gradients, circles, and even eyes and faces are among the patterns that convolutional neural networks excel at identifying in input pictures. Convolutional neural networks are extremely useful for computer vision because of this. Convolutional neural networks, unlike previous computer vision algorithms, can operate immediately on a raw image and do not require any preparation. A convolutional neural network is a feed-forward neural network with up to 20 or 30 layers. The strength of a convolutional neural network comes from a unique sort of layer called the convolutional layer. Multiple convolutional layers are placed on top of one another in convolutional neural networks, each capable of recognizing increasingly complex structures. Handwritten digits can be recognized with three or four convolutional layers, while human faces can be distinguished with 25 layers [61].

A convolutional neural network's use of convolutional layers mimics the organization of the human visual cortex, in which a series of layers processes an input image and identifies ever more complicated features.

### 2.11 Global status report on road safety issued by World Health Organization (WHO)

The rate of road deaths has climbed to 1.35 million yearly, according to the World Health Organization's (WHO) 2018 Global Status Report on Road Safety, published in December

2018. Vehicle accidents are the biggest cause of morbidity and mortality among those aged 5 to 29. Walkers, bicyclists, and motorbikes endure a disproportionately large amount of the cost, especially in underdeveloped nations. The cost of movement, according to the analysis, is too expensive, especially considering the availability of established options. To fulfill any future global target and preserve lives, drastic action is required to put these safeguards in place [62]. In **Fig. 2.8** we can see the death factor distribution according to the continents.

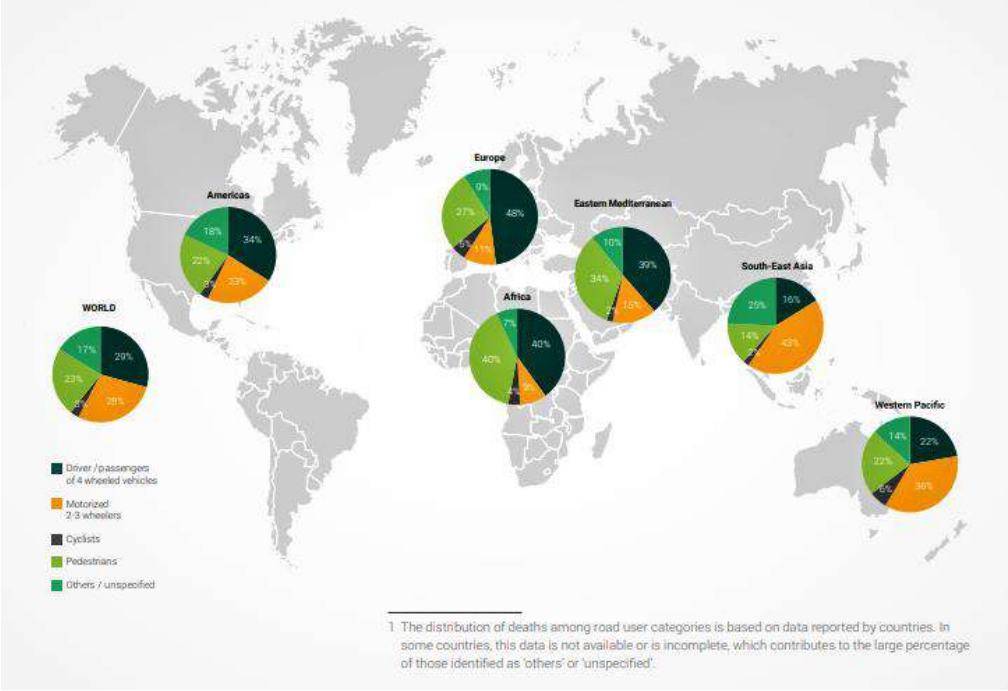


**Fig. 2.8:** Global status report on road safety issued by WHO, 2018 [62].

## 2.12 In Bangladesh, a lack of qualified drivers contributes to fatal traffic accidents

More than half of drivers, according to one road safety organization, do not even have licenses. Widespread demonstrations erupted following the deaths of director Tareque Masud and ATN News CEO Mishuk Munier in 2011, and college students Abdul Karim Rajib and Dia Khanom Mim in 2018. In the month immediately prior to Tareque and Mishuk's deaths, 43 students and another person were murdered in Chattograms Mirsarai when a pickup truck bringing them back from a football game drove into a wayside pond

and collapsed in one of Bangladesh's worst road traffic accidents. Drivers implicated in these instances were penalized, and the administration altered regulations to raise punishments for drivers engaged in traffic accidents, but public outcry and demonstrations resulted in almost no change. After the death of his wife, Jahanara Kanchan, on October 22, 1993, film star Ilias Kanchan founded the Nirapad Sarak Chai, or We Want Safe Roads Movement. In 2017 [63], the government declared it National Road Safety Day. Since then, many programs have been organized by government and private organizations to commemorate the day.

However, despite attempts to improve consciousness, the incidence of traffic deaths has risen in recent years. In 2016, 2,463 individuals died in 2,566 road traffic accidents, according to the police. The yearly death toll increased to 2,513 in 2017 and 2,635 in 2018. The toll nearly quadrupled to 4,138 in 2019. Abrar Hossain, a university graduate, was one of the casualties that year, and his murder provoked widespread demonstrations in Dhaka.

The majority of the collisions happened on highways, and the drivers in most instances escaped. Due to prolonged closures caused by the coronavirus outbreak, the mortality toll was lowered to 3,918 in 2020, but it has since climbed to 3,502, or 14 people per day, in the first eight months of this year. Trucks and closed vans caused more fatalities than buses in 2020. In 2021, there were 975 bus accidents recorded, 1,315 car cars covered van incidents and 981 motorbike incidents. In **Table 2.1** we can see the road accidents rate in Bangladesh from the year 2004 to 2008 [64]. Despite the fact that the number of fatalities and injuries has steadily increased, no noticeable improvement in terms of transportation security has indeed been made.

**Table 2.1:** Road accidents rate in Bangladesh from 2004 to 2008 [64]

| Year | Accidents | Fatalities | Injured | Causalities |
|---|---|---|---|---|
| 2004 | 3917 | 2968 | 2752 | 5720 |
| 2005 | 4949 | 3187 | 2754 | 5941 |
| 2006 | 3794 | 3193 | 2409 | 5602 |
| 2007 | 4869 | 3749 | 3273 | 7022 |
| 2008 | 4427 | 3765 | 3284 | 7049 |

According to the Bangladesh Road Transport Authority (BRTA), there were 4.5 million vehicles registered through September of last year, with 51,668 of them being buses. In comparison, the amount of people with authentic driver's licenses is about 50% of that. According to Nirapad Sarak Chai, there are around 24 lakhs of amateur and unauthorized drivers on the roadways, the majority of whom operate buses, lorries, and covered vans. According to BRTA figures, slightly more than 30 lakhs of motorcycles will be cataloged until 2020, compared to 17 lakhs million motorcycles. Also, around 120,000 motorcycle drivers were operating without a license. According to statistics, motorcycles were involved in 1,127 traffic accidents in 2020, a 6% rise from the earlier year. While operators have been amnestied for the majority of road traffic collisions, very little have been prosecuted. There is scarcely anyone who does not want to reduce accidents, yet there is a significant trend among vehicle operators to violate road traffic management laws, according to Liton Ershad, a Nirapad Sarak Chai official. However, he claims that things are progressively developing in this segment. Sufferers of car causalities frequently do not seek judicial action, and even when they do, the vast majority of cases are resolved out of court. As evidenced in the cases of journalist Mozammel Hossain Montu, who was died in a road collision, and college student Rajib Hossain, who lost his hand in a road chase between two vehicles and died, the courts frequently grant compensation to the victims. However, retrieving the funds is frequently challenging. In cases involving vehicles, for example, the proprietors are sometimes slow to compensate, and the matter might drag on for years as a result. According to the Road Transport Act (RTA) of 2018 allows accident sufferer or their next relative to seek allowance from the "financial aid fund."

Bangladesh Legal Aid and Services Trust (BLAST) strives to get a legal allowance for road accident sufferers. It is a private NGO (Non-Governmental Organization) currently operating in nineteen districts of Bangladesh. Sharmin Akhtar, a prominent lawyer, is in charge of the organization's legal affairs. She claims that the majority of road accident lawsuits are settled out of court because an arrangement is made outside of the legal action and procedure between two parties. Thus, people are demotivated to seek judicial steps if any road collision incident happens. Again, the delays in getting compensation to annoy many litigants, and many sooner or later depart. “As a result, extremely few cases are filed following the accidents. High Court hears the majority of compensation cases. This is

because the applications are based on a breach of the constitutional right to life [which falls under the jurisdiction of the High Court]. As a result, the court evaluates the case and renders a decision," she elaborated [63]. According to Sharmin Akhtar, while allowance could be desired under the Road Transport Act (RTA), there is very little chance of acquiring it through that channel.

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

The proposed detection system can be achieved by two-way. Each approach has its own capability for detecting objects. When both methods are used in conjunction, the detection accuracy is improved. In the unlikely event that one fails, the other will step in to fill the void. The first one is accomplished via distance measurement hardware, such as an ultrasonic sensor or a laser-based distance measurement sensor. We are calling the term "Hard Detection" to refer to this technique. Because it is hardware-based, utilizing many types of distance measurement sensors. And the second detection approach, on the other hand, is entirely based on software technology such as computer vision, machine learning, convolutional neural network algorithm. As always, we are calling the term “Soft Detection” to refer to the second technique. The latter mentioned technique will be discussed in this report.

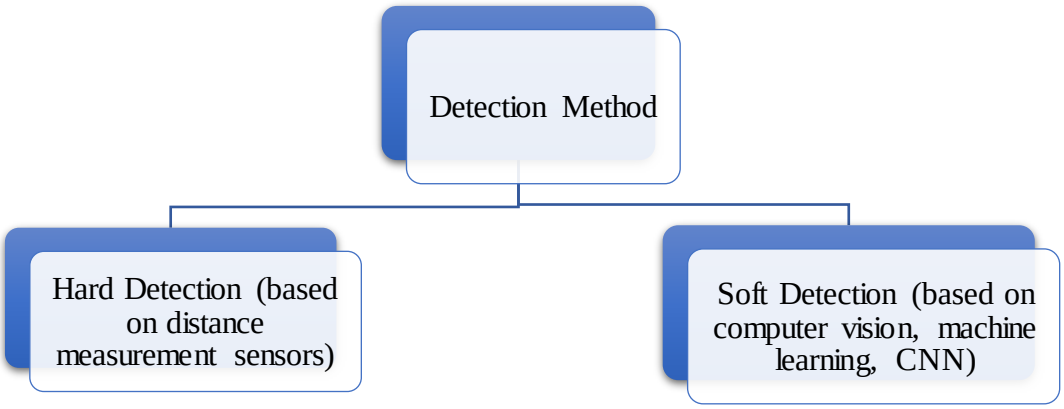


**Fig. 3.1:** Detection Method

## 3.2 Soft Detection

A dataset and a model are required in order to detect an object. It doesn't matter if it's pre-trained or constructed from scratch. By using a pre-trained model, it is hard to detect some custom datasets and objects which has not been modeled yet. Since Bangladesh has a variety

of vehicle patterns, the pre-trained model that is now available is not entirely capable of classifying and detecting all of them. Thus, our approach is to be made a custom dataset and model for Bangladeshi vehicles. The convolutional neural network technique R-CNN has been employed for model training, while Google's machine learning package "Tensorflow" [65] was employed as a data mining tool. Also, we used the OpenCV computer vision library as the simulation technique.

Dataset can be a collection of connected sets of information that is formed of discrete pieces yet may be controlled by a computer as a whole. The purpose of this methodology is to collect data about Bangladeshi roads and vehicles and to transform it into a usable dataset. Here, images and real-time video feed serve as the key components of our customized dataset. In order to do this, we compiled images and videos from locations like Muradpur and Bahaddarhat in order to get native Bangladeshi automobile shapes and patterns. This obtained dataset may be used to train models for vehicle detection, tracking categorization, and segmentation using the deep learning technique. The dataset features 248 images of various sorts of indigenous Bangladeshi vehicles. The qualities of the dataset have a compelling effect on the accuracy of the detection.

### *3.2.1 Soft Detection diagram*

In **Fig. 3.2** a detailed diagram is introduced about how this "Soft Detection" technique functions.

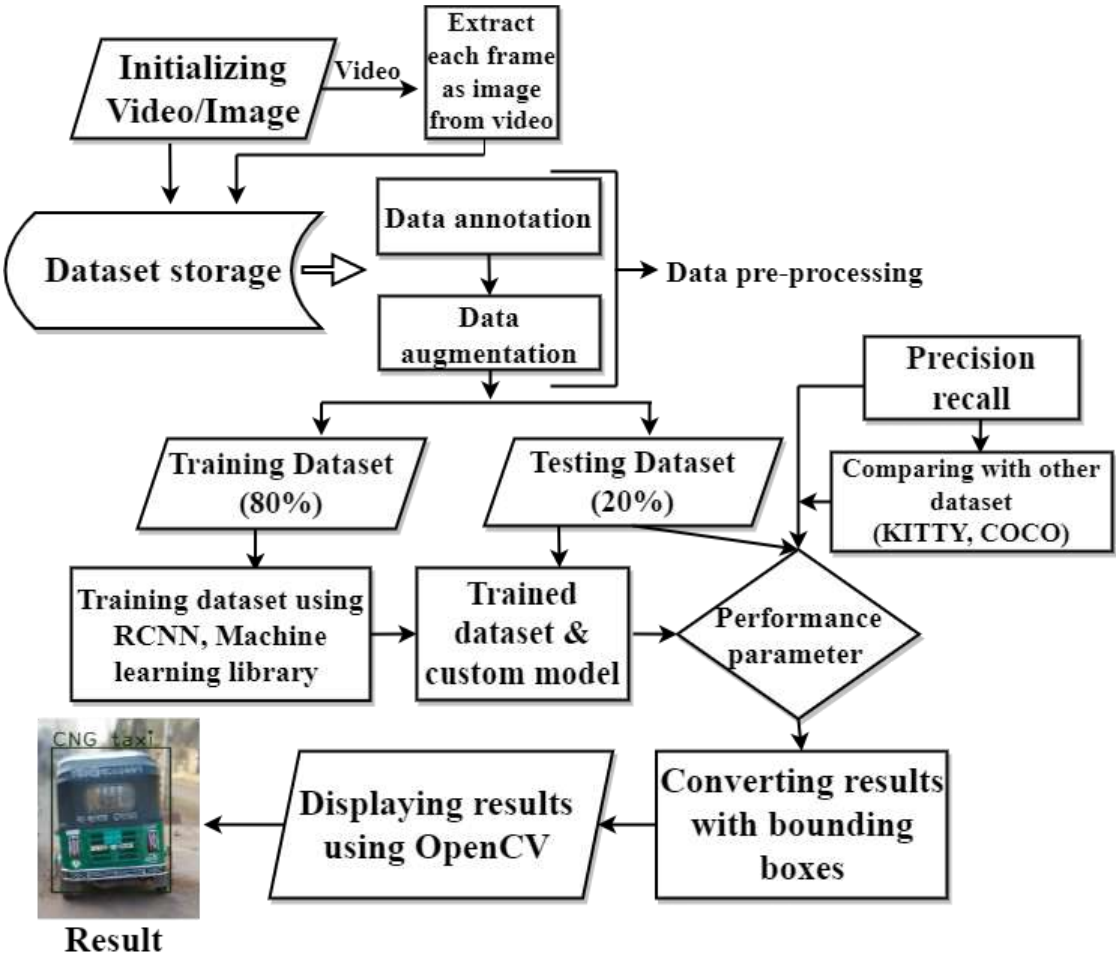


**Fig. 3.2:** Soft Detection.

## 3.3 Data Pre-processing

Data pre-processing has two steps to do. To create a custom dataset, we must first isolate the region of interest from the real image. These two steps are Data augmentation and Data annotation. Data Pre-processing has a significant role in model training. The detection rate of the output model has very much relied on how carefully we processed our data [66]. We

primarily collected 248 different types of images from the street of Chittagong, Bangladesh. So, we have to generate corresponding config files which are known as XML files. Thus, our dataset contains 248 XML files for corresponding images. These XML files contain properties of the dataset.

### *3.3.1 Data augmentation*

In data analysis, it refers to approaches for enhancing the quantity of data by incorporating slightly altered copies of previously collected data. As a result, we preserved the original image but significantly adjusted it. It creates additional samples for the same image. We cropped the same image 3-4 times and stored it for better gaining better accuracy.

### *3.3.2 Data annotation*

It is the process of categorizing data that is available in a variety of different media, such as text, video, or image. Labeled data sets are essential for machine learning in order for machines to comprehend the input patterns quickly and clearly. We used the "LabelImg" python script to do this. Using this script we separated the region of interest [67] of each image which contains at least one object to detect. Also, this python script helps us to generate a corresponding XML file for each image. Combining both images and corresponding XML files we are calling it a working dataset for model training.

## 3.4 Dataset handling

After making a workable dataset we separated it into two types. The first one is the Training Dataset and the second one is the Testing Dataset. In a workable dataset, a training dataset is used to build the model. That's why it has a larger amount of data. On the other hand, the test dataset is to validate the model which has been trained using the training dataset. From **Fig. 3.3** we will have an idea about how a dataset is processed in model training. We can say the test dataset is a subgroup of the training dataset. To organize the dataset there are two conditions that test dataset need to be fulfilled.

1. It is sufficiently large enough to produce statistically significant findings.
2. It is a representative sample of the entire data set. In other words, avoid selecting a test set that is dissimilar or conflicting to the training set.

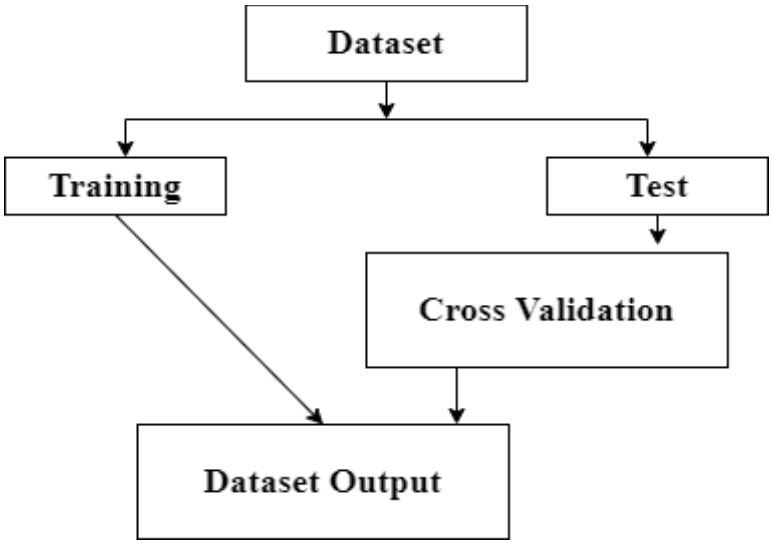


**Fig. 3.3:** Dataset handling

For this proposed model we took a ratio of (8:2) of total data [68]. By this proportion we have,

$$\text{Training dataset} = 248 * 80\%$$

$$= 199 \text{ images}$$

$$\text{Testing dataset} = 248 * 20\%$$

$$= 49 \text{ images}$$

It is obvious that a larger amount of dataset is required to make a custom model. Otherwise, detection accuracy is hampered due to poorly handled datasets. We used 248 images of different vehicles which we collected from different locations of Chittagong, Bangladesh. In **Table 3.1** we have an idea about how many images are allocated for each class of objects.

**Table 3.1:** Image allocation of different classes in the dataset

| Class ID | Image Qty. |
|---|---|
| Person | 31 |
| BUS | 53 |
| CNG Taxi | 37 |
| Bicycle | 26 |
| Truck | 22 |
| Car | 47 |
| Rickshaw | 32 |

### 3.5 Model Training

Model training is the elementary part of our system. We cannot detect any object without training the model before. Model training means learning a machine a dataset that will give the desired output. Because machines cannot think or watch like humans. Suppose, an infant has no ability to think like an adult. But day by day that infant has been gained the ability to think. For this, that infant needs to be guided or educated. The same case is considered for machine learning. The machine needs to be guided or educated to get desired output. We can say that this so-called guidance is the "**Model**".

For model training, we used **Google Colab** as the training environment, **OpenCV** as the computer vision library, **RCNN** as the convolutional neural network algorithm, and Google's machine learning library **Tensorflow** as machine learning library. After the model training which took roughly 8 hours to complete by using GPU hardware acceleration, we got the interference graph file. It is the output file of model training. By using this interference graph file we can detect and classify objects. It is worthy to mention that the model file cannot detect and classify objects which have not been trained on.

We test our model on a precision level of 70%. That means if the input object is 70% matched with the model file then it will detect the object. Since no model is 100% accurate it cannot

be said that our trained model also gets that level of accuracy. We briefly discussed about our model accuracy in the result chapter.

### *3.5.1 Model training hyperparameter*

Machine learning techniques rely heavily on hyperparameters. A hyperparameter is a variable whose standard is resolved prior to the start of the machine learning process. The standard of other variables, on the other hand, is determined by the model learning process. The efficiency and precision of the learning process are affected by algorithm hyperparameters.

Hyperparameters are crucial since they can have a direct influence on the training algorithm's performance as well as the performance of the model being trained. Proper hyperparameter selection is crucial to the effectiveness of neural network design and has a significant effect on the learned model. For example, if the learning rate is very low, the model will miss important trends and patterns. If it is set too high, it is probable that collisions will happen.

So what are the properties and use of hyperparameters in Machine Learning? The answer is following:

- They are used to frequently employed in procedures to improve in the determination of model parameters.
- They are used to frequently fine-tuned for a specific predictive modeling application.
- They are commonly used to define the Heuristic algorithms.
- Model cannot be learned directly from the raw data in a traditional model training approach and must be specified.
- They are used to define higher-level model components such as complexities and learning capabilities.
- The developer or engineer needs to define the hyperparameter in order to obtain the best outcome.

In **Table 3.2** we can see the hyperparameter and its values that we used to build and train this model.

**Table 3.2:** Model training hyperparameters

| **Hyperparameter Name** | **Values** |
|---|---|
| Batch Size | 8 |
| Number of Steps | 31 |
| Iteration per epoch | 115 |
| Decay | 0.0006 |
| Momentum | 0.8 |
| Learning Rate | $10^{-3}$ |

# CHAPTER 4

# SIMULATION

## 4.1 Simulation environment

Google Colab is our primary simulation library. For model training and image processing, it needs a higher amount of computational power. Google Colab offers us a free GPU hardware acceleration technique. In **Fig. 4.1** we can see how the object detection algorithm works. It is such a simple algorithm to implement and get the desired output. From the trained model as a graph file, we can detect objects according to this algorithm.

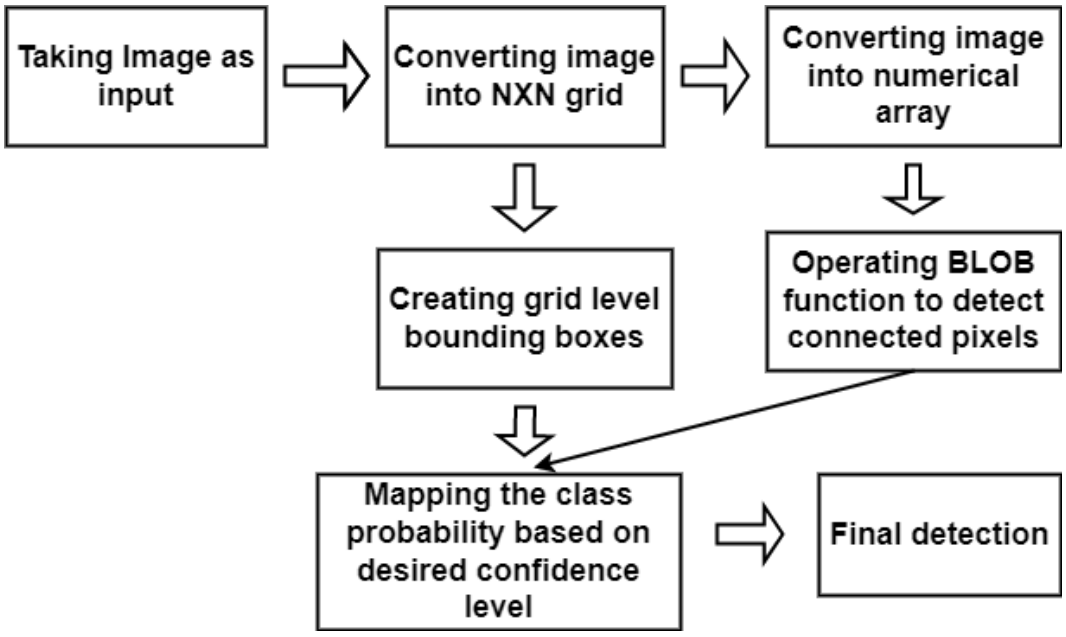


**Fig. 4.1:** Computer Vision as Object Detection algorithm.

**Fig. 4.2** and **Fig. 4.3** illustrate the simulation of model training.

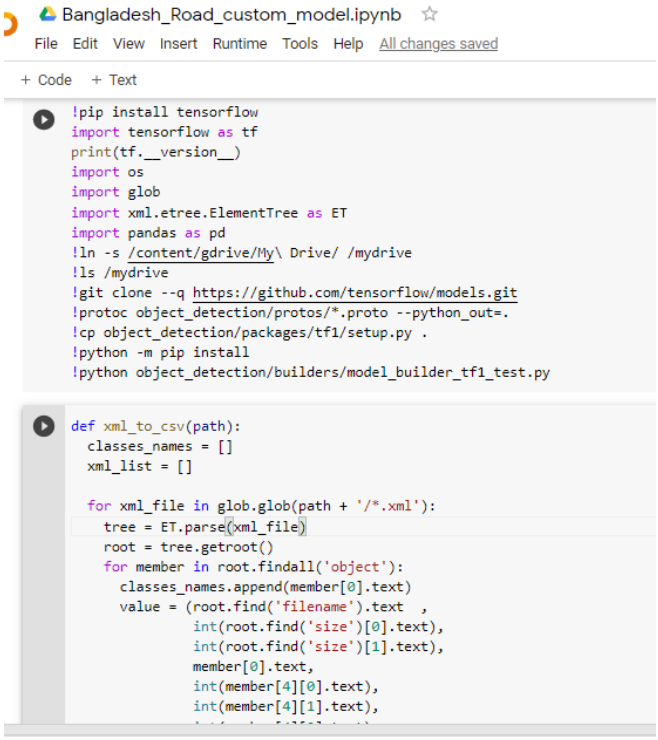


**Fig. 4.2:** Simulation of model training

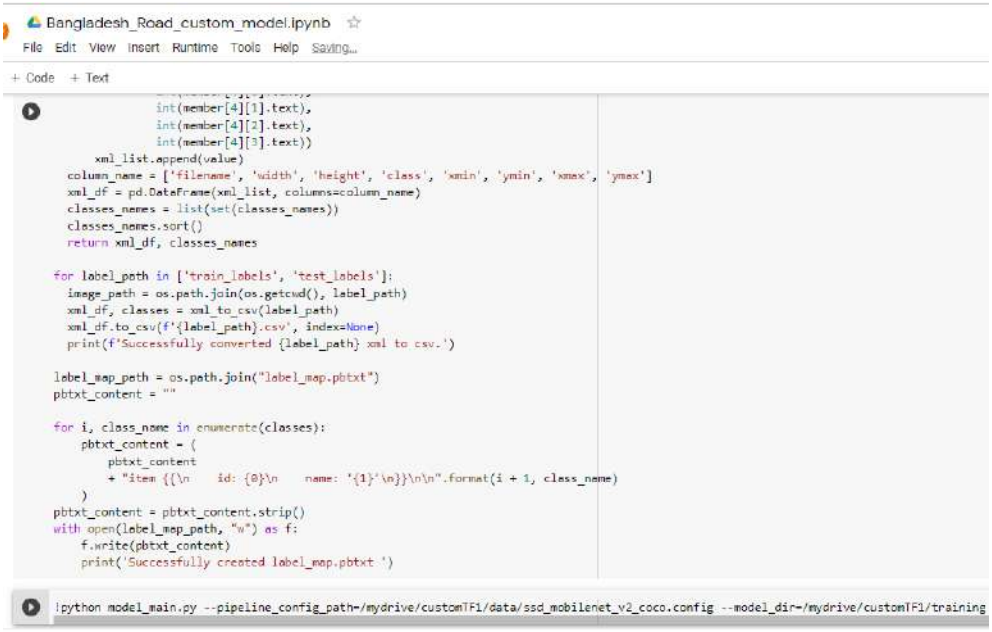


**Fig. 4.3:** Simulation of model training

**Fig. 4.4** illustrates the simulation of single image processing.

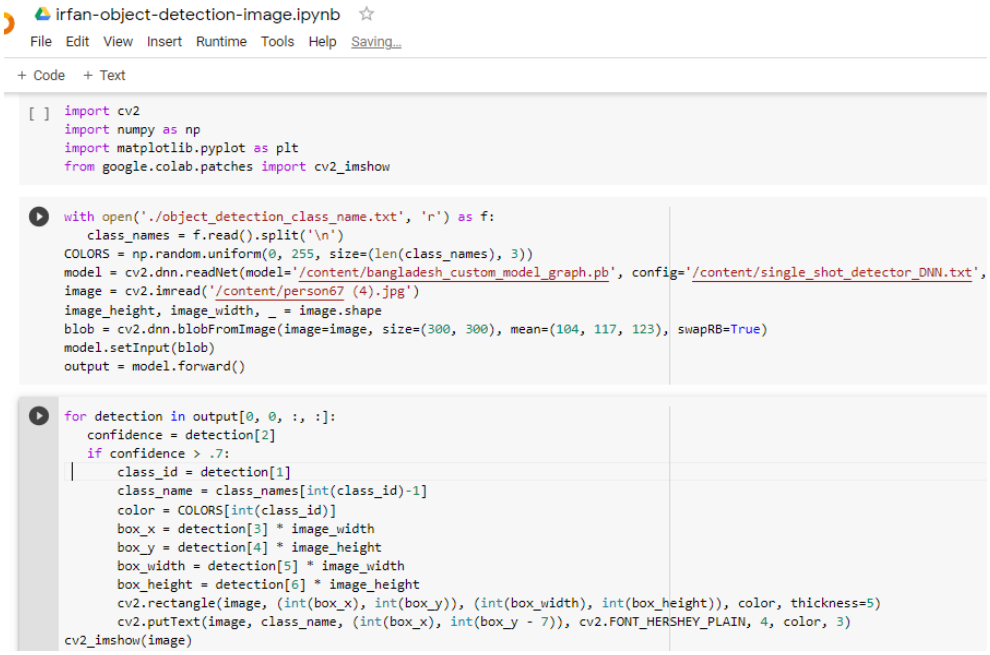


**Fig. 4.4:** Simulation of image processing

**Fig. 4.5** illustrates the simulation of real-time video processing.


**Fig. 4.5:** Simulation of real-time video processing

**Fig. 4.6** also shows us the simulation of real-time video processing.

```
while cap.isOpened():
    ret, frame = cap.read()
    if ret:
        image = frame
        image_height, image_width, _ = image.shape
        blob = cv2.dnn.blobFromImage(image=image, size=(300, 300), mean=(104, 117, 123), swapRB=True)
        start = time.time()
        model.setInput(blob)
        output = model.forward()
        end = time.time()
        fps = 1 / (end-start)
        for detection in output[0, 0, :, :]:
            confidence = detection[2]
            if confidence > .4:
                class_id = detection[1]
                class_name = class_names[int(class_id)-1]
                color = COLORS[int(class_id)]
                box_x = detection[3] * image_width
                box_y = detection[4] * image_height
                box_width = detection[5] * image_width
                box_height = detection[6] * image_height
                cv2.rectangle(image, (int(box_x), int(box_y)), (int(box_width), int(box_height)), color, thickness=2)
                cv2.putText(image, class_name, (int(box_x), int(box_y - 5)), cv2.FONT_HERSHEY_SIMPLEX, 1, color, 2)
                cv2.putText(image, f"{fps:.2f} FPS", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2_imshow(image)
        out.write(image)
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    else:
        break
cap.release()
cv2.destroyAllWindows()
```

**Fig. 4.6:** Simulation of real-time video processing

**Fig. 4.7** illustrates the output of the simulation.


**Fig. 4.7:** Simulation output

## 4.2 Simulation of different objects detection

Now we can see some different objects which have been taken into account. Here is the list of objects on which we tested our simulation.

1. Pedestrian of road.
2. Bus.
3. CNG Taxi.
4. Rickshaw.
5. Bicycle.
6. Car.
7. Truck.

### *4.2.1 Pedestrian detection*

The following images of **Fig. 4.8** are from simulation output which detects pedestrians and classify its class id as a person.


**Fig. 4.8:** Pedestrian Detection as a person.

### *4.2.2 Bus*

The following images of **Fig. 4.9** are from simulation output which detects Bus and classifies its class id as also Bus.


**Fig. 4.9:** Bus with bounding boxes.

### *4.2.3 CNG Taxi*

The following images of **Fig. 4.10** are from simulation output which detects Taxi and classifies its class id as a CNG Taxi.


**Fig. 4.10:** CNG Taxi with bounding boxes

### *4.2.4 Rickshaw*

The following images of **Fig. 4.11** are from simulation output which detects Rickshaw and classifies its class id as Rickshaw.


**Fig. 4.11:** Rickshaw with bounding boxes.

### *4.2.5 Bicycle*

The following images of **Fig. 4.12** are from simulation output which detects Bicycle and classify its class id as Bicycle.


**Fig. 4.12:** Bicycle detection with a person.

### *4.2.6 Car*

The following images of **Fig. 4.13** are from simulation output which detects Car and classifies its class id as Car.


**Fig. 4.13:** Car with bounding boxes.

### *4.2.7 Truck*

The following images of **Fig. 4.14** are from the simulation output which detects Bangladeshi Truck and classifies its class id as a truck.


**Fig. 4.14:** Truck with bounding boxes.

# CHAPTER 5

# RESULT and DISCUSSION

## 5.1 Class ID detection

To verify the accuracy of this proposed system we need to compare it with other available pre-trained models. For this purpose, we selected the **YOLO** model as benchmarking model. Now our method is that, If the YOLO model cannot detect the object then surely our custom model has to detect it. Also, it has to classify according to the custom dataset. From **Fig. 5.1** we can see two respective images. The first one was simulated using the YOLO model which was detected Bus as Truck. And the second image was simulated using our custom dataset and model which detected it as Bus. Thus, we can reach a conclusion that the YOLO model either cannot detect an object or failed to classify that object accurately Since it is not trained on Bangladeshi vehicle datasets. On the other hand, our custom model detects the object and classifies it accurately.


**Fig. 5.1:** Separately detected objects on YOLO and custom model.

### 5.2 Custom Model accuracy

To evaluate the accuracy of our trained custom model we tested 82 images of different Bangladeshi vehicles. These 82 images are distinct from the images which we used to build the model. From **Table 5.1** we can observe the detection accuracy of the different classes of vehicles and objects.

**Table 5.1:** Detection accuracy of Custom Model.

| Class ID | Input Image Qty. | Detected Image Qty. | Undetected Image Qty. | Class ID Error | Detection accuracy w/o class error (%) | Detection accuracy with class error (%) | Overall accuracy (%) |
|---|---|---|---|---|---|---|---|
| BUS | 19 | 16 | 3 | 2 | 84% | 73% | 78% |
| CNG Taxi | 13 | 9 | 4 | 1 | 69% | 61% | 65% |
| Car | 14 | 10 | 4 | 0 | 71% | 71% | 71% |
| Rickshaw | 16 | 7 | 9 | 2 | 43% | 31% | 37% |
| Truck | 8 | 6 | 2 | 4 | 75% | 25% | 50% |
| Person | 12 | 11 | 1 | 0 | 91% | 91% | 91% |

Here we can find some interesting results regarding the detection accuracy of our custom model. Road pedestrians have the highest detection accuracy which is roughly 91%. So why it has a higher amount of accuracy than others? The human body has a distinct shape and pattern which can easily differentiate among other vehicles and objects. That's why it is much easier to detect on the road.

On the other hand, we can see Rickshaw has the lowest amount of detection accuracy among other classes. The answer is lying in the dataset pre-processing. As we discussed early that, Detection accuracy is dependent on the dataset. So, we can say that the dataset of Rickshaw is poorly handled. It is obvious to observe that only the front image of the Rickshaw has been detected by our model. But it cannot detect the rear side image of a Rickshaw. Our

dataset has only two images of the rear side of a rickshaw. Which is considered a poor amount of data given to a model for training. That's why it has a poor detection accuracy among other objects. If we feed enough amount of images from the rear side of the rickshaw in the dataset then it can easily detect the rickshaw from its rear side.

In **Fig. 5.2** we can have a closer and more coherent understanding of the accuracy of our custom model.

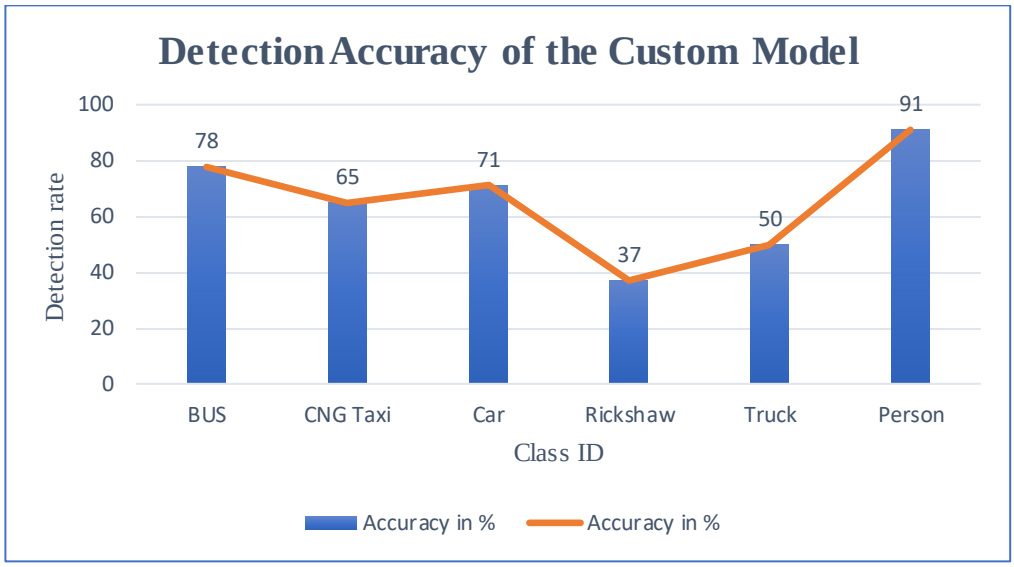


**Fig. 5.2:** Detection accuracy of the custom model.

### 5.3 Dataset comparison

There are many datasets that are developed by programmers and engineers are available worldwide. These datasets are widely used in computer vision, machine learning applications constantly. So, we did a comparative analysis of our dataset among some popular datasets which is practiced worldwide in **Table 5.2.**

**Table 5.2:** Dataset comparison

| Dataset Name | Number of Images | Annotation | Number of classes | Number of vehicle detection classes | Unique Vehicle classes |
|---|---|---|---|---|---|
| KITTI | 7481 | 3D bounding boxes | 8 | 5 | Tram train |
| Waymo | 12 Million | LiDAR-based bounding boxes | 4 | 2 | N/A |
| ApolloScape | 701 | Semantic annotation | 32 | 6 | Motorcycle, boat, Ship. |
| Custom Dataset | 248 | 2D bounding boxes | 7 | 7 | Seven different Bangladeshi vehicles |

## 5.4 Advantage of this system

- It can detect Bangladeshi vehicle pattern which is not accurately recognizable by other available models.
- By using Computer vision and machine learning techniques it has a higher detection rate than a typical hardware-based distance measurement system.
- Custom models can be made for other purposes such as criminal identification, face recognition, intelligence gathering, security surveillance, etc.
- It can classify objects into whether it's human, animal, vehicle, or obstacle.
- Opensource technology. So, any model can be extracted based on the desired output.

## 5.5 Limitations of this system

- Required a large amount of dataset to build a model. Which is sometimes hard to gather.
- Detection accuracy depends on the amount & properties of the dataset. Poor dataset causes poor detection rate.

- Needs a feedback system in the vehicle to initiate the braking. This feedback system is complex to develop.
- Has limited range. It cannot detect until it clearly sees the object.
- Required a large amount of computational power (GPU-based hardware) to train the model.
- Has more complexity than any hardware-based detection system.
- costlier than hardware-based detection system.

# CHAPTER 6

# CONCLUSION

## 6.1 Conclusion

From the above discussion, we can reach the conclusion that emerging technology like computer vision, machine learning can be used for the betterment of road safety and road traffic system. From the simulations, we can say that if this system can implement in real life it will somewhat improve the quality of the road traffic management systems by detecting and classifying road objects of developing countries like Bangladesh.

## 6.2 Future scopes of work

1. Using a Fuzzy logic controller we can develop an autonomous vehicle.
2. A feedback system for transmission and engine can be developed for braking mechanisms in vehicles.
3. Model can be trained for any field such as biometric recognition, security, military etc.
4. Model can be used to manage traffic systems on the road.

# REFERENCES


[1] "(PDF) Trends and Patterns of Fatal Road Accidents in Nigeria (2006-2014)." https://www.researchgate.net/publication/279536918_Trends_and_Patterns_of_Fatal_Road_Accidents_in_Nigeria_2006-2014 (accessed Dec. 23, 2021).

[2] K. Ileri, A. Duru, and S. Gorgunoglu, "Single Board Computer Based Real Time Forward," no. 8, pp. 22–25, 2016.

[3] O. Alpar, "Corona segmentation for nighttime brake light detection," *IET Intelligent Transport Systems*, vol. 10, no. 2, pp. 97–105, Mar. 2016, doi: 10.1049/IET-ITS.2014.0281.

[4] S. Garethiya, L. Ujjainiya, and V. Dudhwadkar, "PREDICTIVE VEHICLE COLLISION AVOIDANCE SYSTEM USING RASPBERRY-PI," 2015.

[5] D. Y. Huang, C. H. Chen, T. Y. Chen, W. C. Hu, and K. W. Feng, "Vehicle detection and inter-vehicle distance estimation using single-lens video camera on urban/suburb roads," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 250–259, Jul. 2017, doi: 10.1016/J.JVCIR.2017.04.006.

[6] V. Gajjar, A. Gurnani, and Y. Khandhediya, "Human Detection and Tracking for Video Surveillance A Cognitive Science Approach," Sep. 2017, Accessed: Dec. 23, 2021. [Online]. Available: https://arxiv.org/abs/1709.00726v1

[7] M. Adel, A. Moussaoui, M. Rasigni, S. Bourennane, and L. Hamami, "Statistical-based tracking technique for linear structures detection: Application to vessel segmentation in medical images," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 555–558, 2010, doi: 10.1109/LSP.2010.2046697.

[8] X. T. Truong, V. N. Yoong, and T. D. Ngo, "RGB-D and laser data fusion-based human detection and tracking for socially aware robot navigation framework," *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, pp. 608–613, 2015, doi: 10.1109/ROBIO.2015.7418835.

[9] P. G. Student, C. Engineering, B. V. M. E. College, and V. v Nagar, "a Survey on Object Detection and Tracking," *International Journal of Advance Engineering and Research Development*, vol. 3, no. 01, pp. 2970–2978, 2016, doi: 10.21090/ijaerd.030144.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," pp. 248–255, Mar. 2010, doi: 10.1109/CVPR.2009.5206848.

[12] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A Low-Shot Transfer Detector for Object Detection," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2836–2843, Mar. 2018.

[13] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa, "Deep Regionlets for Object Detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11215 LNCS, pp. 827–844, Dec. 2017, doi: 10.1007/978-3-030-01252-6_49.

[14] S. Pang, J. J. del Coz, Z. Yu, O. Luaces, and J. Díez, "Deep learning to frame objects for visual target tracking," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 406–420, Oct. 2017, doi: 10.1016/J.ENGAPPAI.2017.08.010.

[15] C. C. Kao, T. Y. Lee, P. Sen, and M. Y. Liu, "Localization-Aware Active Learning for Object Detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11366 LNCS, pp. 506–522, Jan. 2018, doi: 10.1007/978-3-030-20876-9_32.

[16] Y. Xu, D. Xu, S. Lin, T. X. Han, X. Cao, and X. Li, "Detection of sudden pedestrian crossings for driving assistance systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 729–739, 2012, doi: 10.1109/TSMCB.2011.2175726.

[17] S. Hisaka and S. Kamijo, "On-board wireless sensor for collision avoidance: Vehicle and pedestrian detection at intersection," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 198–205, 2011, doi: 10.1109/ITSC.2011.6082853.

[18] B. S. Choi and J. J. Lee, "Localization of a mobile robot based on an ultrasonic sensor using dynamic obstacles," *Artificial Life and Robotics 2008 12:1*, vol. 12, no. 1, pp. 280–283, Apr. 2008, doi: 10.1007/S10015-007-0482-4.

[19] J. Kang, M. Jin, J. Park, and D. Park, "A study on application of sensor fusion to collision avoidance system for ships," *ICCAS 2010 - International Conference on Control, Automation and Systems*, pp. 1741–1744, 2010, doi: 10.1109/ICCAS.2010.5669788.

[20] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 971–985, Mar. 2012, doi: 10.1109/TVT.2012.2182785.

[21] M. Heddebaut *et al.*, "Millimeter-wave communicating-radars for enhanced vehicle-to-vehicle communications," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 3, pp. 440–456, 2010, doi: 10.1016/J.TRC.2009.05.004.

[22] W. Zhang, "LIDAR-based road and road-edge detection," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 845–848, 2010, doi: 10.1109/IVS.2010.5548134.

[23] "Collision Avoidance System Using Laser Beams." https://www.researchgate.net/publication/245689677_Collision_Avoidance_System_Using_Laser_Beams (accessed Dec. 23, 2021).

[24] R. S. Kumar, P. K. Stanley, and A. S. Gandhi, "Raspberry Pi based vehicle collision avoidance system," *Proceedings of IEEE International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology, ICIEEIMT 2017*, vol. 2017-January, pp. 211–215, Nov. 2017, doi: 10.1109/ICIEEIMT.2017.8116838.

[25] D. Thakur and A. Thakare, "A Review on Implementation of FPGA for Automatic Reverse Braking System," 2015.

[26] G. Benet, F. Blanes, J. E. Simó, and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and Autonomous Systems*, vol. 40, no. 4, pp. 255–266, Sep. 2002, doi: 10.1016/S0921-8890(02)00271-3.

[27] G. V. R. N. D. S. S. Kanth, H. Campus, and M. Y. H. Jewel, "Collision Warning With Automatic Braking System For Electric Cars," vol. 5, no. 2, pp. 153–165, 2015.

[28] J. Li and J. Wang, "Research on the automotive EBD system based on fuzzy control," *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, vol. 4, 2010, doi: 10.1109/ICCET.2010.5485444.

[29] D. Barik and M. Mondal, "Object identification for computer vision using image segmentation," *undefined*, vol. 2, 2010, doi: 10.1109/ICETC.2010.5529412.

[30] J. Marot and S. Bourennane, "Raspberry Pi for image processing education," *25th European Signal Processing Conference, EUSIPCO 2017*, vol. 2017-January, pp. 2364–2368, Oct. 2017, doi: 10.23919/EUSIPCO.2017.8081633.

[31] J. W. Baek, B. G. Han, K. J. Kim, Y. S. Chung, and S. I. Lee, "Real-Time Drowsiness Detection Algorithm for Driver State Monitoring Systems," *International Conference on Ubiquitous and Future Networks, ICUFN*, vol. 2018-July, pp. 73–75, Aug. 2018, doi: 10.1109/ICUFN.2018.8436988.

[32] T. Hong, H. Qin, and Q. Sun, "An Improved Real Time Eye State Identification System in Driver Drowsiness Detection," *undefined*, pp. 1449–1453, 2007, doi: 10.1109/ICCA.2007.4376601.

[33] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting Driver Drowsiness Based on Sensors: A Review," *Sensors (Basel, Switzerland)*, vol. 12, no. 12, p. 16937, 2012, doi: 10.3390/S121216937.

[34] V. Menezes, V. Patchava, and M. S. D. Gupta, "Surveillance and monitoring system using Raspberry Pi and SimpleCV," *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015*, pp. 1276–1278, Jan. 2016, doi: 10.1109/ICGCIOT.2015.7380661.

[35] I. El Naqa and M. J. Murphy, "What Is Machine Learning?," *Machine Learning in Radiation Oncology*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3_1.

[36] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, May 2006, doi: 10.1109/TPAMI.2006.104.

[37] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using optical sensors: A review," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 585–590, 2004, doi: 10.1109/ITSC.2004.1398966.

[38] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: Features and classifiers," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2019–2034, Jul. 2006, doi: 10.1109/TIP.2006.877062.

[39] A. Takeuchi, S. Mita, and D. McAllester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," *undefined*, pp. 1014–1021, 2010, doi: 10.1109/IVS.2010.5548067.

[40] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1258–1270, 2002, doi: 10.1109/JPROC.2002.801444.

[41] S. Tsugawa, "Vision-based vehicles in Japan: machine vision systems and driving control systems," *undefined*, vol. 41, no. 4, pp. 398–405, 1994, doi: 10.1109/41.303790.

[42] O. O. Khin, Q. M. Ta, and C. C. Cheah, "Development of a wireless sensor network for human fall detection," *2017 IEEE International Conference on Real-Time Computing and Robotics, RCAR 2017*, vol. 2017-July, pp. 273–278, Mar. 2018, doi: 10.1109/RCAR.2017.8311873.

[43] J. Kwon and N. Kwak, "Human detection by Neural Networks using a low-cost short-range Doppler radar sensor," *2017 IEEE Radar Conference, RadarConf 2017*, pp. 0755–0760, Jun. 2017, doi: 10.1109/RADAR.2017.7944304.

[44] P. S. Santoso and H. M. Hang, "Learning-based human detection applied to RGB-D images," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2017-September, pp. 3365–3369, Feb. 2018, doi: 10.1109/ICIP.2017.8296906.

[45] S. Mathur, B. Subramanian, S. Jain, K. Choudhary, and D. R. Prabha, "Human detector and counter using raspberry Pi microcontroller," *2017 Innovations in Power and Advanced Computing Technologies, i-PACT 2017*, vol. 2017-January, pp. 1–7, 2017, doi: 10.1109/IPACT.2017.8244984.

[46] Y. Yuan, J. Gu, F. Chen, Y. Xu, H. Yang, and Y. Miao, "Study of obstacle avoidance navigation robot control based on bland man tracing wall theory," *2015 IEEE International Conference on Information and Automation, ICIA 2015 - In conjunction with 2015 IEEE International Conference on Automation and Logistics*, pp. 398–403, Sep. 2015, doi: 10.1109/ICINFA.2015.7279320.

[47] S. H. Lian, "Fuzzy logic control of an obstacle avoidance robot," *IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 26–30, 1996, doi: 10.1109/FUZZY.1996.551714.

[48] T. C. Lai, S. R. Xiao, H. Aoyama, and C. C. Wong, "Path planning and obstacle avoidance approaches for robot arm," *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2017*, vol. 2017-November, pp. 334–337, Nov. 2017, doi: 10.23919/SICE.2017.8105619.

[49] K. Balasubramanian, R. Arunkumar, J. Jayachandran, V. Jayapal, B. A. Chundatt, and J. D. Freeman, "Object recognition and obstacle avoidance robot," *2009 Chinese Control and Decision Conference, CCDC 2009*, pp. 3002–3006, 2009, doi: 10.1109/CCDC.2009.5192399.

[50] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osório, "Mobile robots navigation in indoor environments using Kinect sensor," *Proceedings - 2012 2nd Brazilian Conference on Critical Embedded Systems, CBSEC 2012*, pp. 36–41, 2012, doi: 10.1109/CBSEC.2012.18.

[51] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.

[52] L. Zhang and R. Nevatia, "Efficient scan-window based object detection using GPGPU," *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2008, doi: 10.1109/CVPRW.2008.4563097.

[53] S. Bauer, U. Brunsmann, and S. Schlotterbeck-Macht, "FPGA Implementation of a HOG-based Pedestrian Recognition System," 2010.

[54] M. Hiromoto and R. Miyamoto, "Hardware architecture for high-accuracy real-time pedestrian detection with CoHOG features," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, pp. 894–899, 2009, doi: 10.1109/ICCVW.2009.5457609.

[55] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, R. Miyamoto, and Y. Nakamura, "Hardware architecture for HOG feature extraction," *IIH-MSP 2009 - 2009 5th International Conference on*

*Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1330–1333, Dec. 2009, doi: 10.1109/IIH-MSP.2009.216.

[56] Y. Yazawa *et al.*, "FPGA Hardware with Target-Reconfigurable Object Detector," *undefined*, vol. E98D, no. 9, pp. 1637–1645, Sep. 2015, doi: 10.1587/TRANSINF.2014OPP0008.

[57] K. Negi, K. Dohi, Y. Shibata, and K. Oguri, "Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm," *2011 International Conference on Field-Programmable Technology, FPT 2011*, 2011, doi: 10.1109/FPT.2011.6132679.

[58] S. Bauer, S. Köhler, K. Doll, and U. Brunsmann, "FPGA-GPU architecture for Kernel SVM pedestrian detection," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pp. 61–68, 2010, doi: 10.1109/CVPRW.2010.5543772.

[59] T. P. Cao and G. Deng, "Real-time vision-based stop sign detection system on FPGA," *Proceedings - Digital Image Computing: Techniques and Applications, DICTA 2008*, pp. 465–471, 2008, doi: 10.1109/DICTA.2008.37.

[60] "YOLO: Real-Time Object Detection Explained." https://www.v7labs.com/blog/yolo-object-detection (accessed Dec. 23, 2021).

[61] "Convolutional Neural Network Definition | DeepAI." https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network (accessed Dec. 23, 2021).

[62] R. Ishrat, "Global Status Report on Road Safety 2018: Summary," *World Health Organization*, no. 1, p. 20, 2018.

[63] "Lack of skilled drivers spurs deadly road crashes in Bangladesh | bdnews24.com." https://bdnews24.com/bangladesh/2021/10/22/lack-of-skilled-drivers-spurs-deadly-road-crashes-in-bangladesh (accessed Dec. 23, 2021).

[64] "(PDF) IMPROVING HIGHWAY SAFETY IN BANGLADESH: ROAD IMPROVEMENT AND THE POTENTIAL APPLICATION OF iRAP." https://www.researchgate.net/publication/346082311_IMPROVING_HIGHWAY_SAFETY_IN_BANGLADESH_ROAD_IMPROVEMENT_AND_THE_POTENTIAL_APPLICATION_OF_iRAP (accessed Dec. 23, 2021).

[65] M. Abadi *et al.*, *{TensorFlow}: A System for {Large-Scale} Machine Learning*, vol. 10, no. July. 2016.

[66] F. Scheidegger, R. Istrate, G. Mariani, L. Benini, C. Bekas, and C. Malossi, "Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy," *Visual Computer*, vol. 37, no. 6, pp. 1593–1610, Jun. 2021, doi: 10.1007/S00371-020-01922-5/TABLES/8.

[67] A. Nieto-Castanon, S. S. Ghosh, J. A. Tourville, and F. H. Guenther, "Region of interest based analysis of functional imaging data," *NeuroImage*, vol. 19, no. 4, pp. 1303–1316, Aug. 2003, doi: 10.1016/S1053-8119(03)00188-5.

[68] "Is there an ideal ratio between a training set and validation set? Which trade-off would you suggest?" https://www.researchgate.net/post/Is-there-an-ideal-ratio-between-a-training-set-and-validation-set-Which-trade-off-would-you-suggest (accessed Dec. 23, 2021).

# APPENDIX

**Source code for model training**

```
!pip install tensorflow
import tensorflow as tf
print(tf.__version__)
import os
import glob
import xml.etree.ElementTree as ET
import pandas as pd
!ln -s /content/gdrive/My\ Drive/ /mydrive
!ls /mydrive
!git clone --q https://github.com/tensorflow/models.git
%cd models/research
!protoc object_detection/protos/*.proto --python_out=.
!cp object_detection/packages/tf1/setup.py .
!python -m pip install
!python object_detection/builders/irfan_model_custom_tf1_test.py
%cd /mydrive/customTF1/data/
!unzip /mydrive/customTF1/images.zip -d .
!unzip /mydrive/customTF1/annotations.zip -d .
!mkdir test_labels train_labels
!ls annotations/* | sort -R | head -274 | xargs -I{} mv {} test_labels/
!ls annotations/* | xargs -I{} mv {} train_labels_1/
def xml_to_csv(path):
  classes_names_1 = []
  xml_list_1 = []

  for xml_file in glob.glob(path + '/*.xml'):
    tree = ET.parse(xml_file)
    root = tree.getroot()
    for member in root.findall('object'):
      classes_names_1.append(member[0].text)
      value = (root.find('filename').text  ,
               int(root.find('size')[0].text),
               int(root.find('size')[1].text),
               member[0].text,
               int(member[4][0].text),
               int(member[4][1].text),
               int(member[4][2].text),
               int(member[4][3].text))
      xml_list.append(value)
  column_name_1 = ['filename', 'width_1', 'height_1', 'class_1', 'xmin', 'ymin', 'xmax', 'ymax']
  xml_df = pd.DataFrame(xml_list, columns=column_name)
  classes_names_object = list(set(classes_names_object))
  classes_names.sort()
  return xml_df, classes_names_object
```

```
for label_path in ['train_labels_1', 'test_labels_1']:
  image_path_1 = os.path.join(os.getcwd(), label_path)
  xml_df, classes = xml_to_csv(label_path)
  xml_df.to_csv(f'{label_path}.csv', index=None)
  print(f'converted {label_path} xml to csv.')

label_map_path_1 = os.path.join("label_map_1.pbtxt")
pbtxt_content_1 = ""

for i, class_name_1 in enumerate(classes):
    pbtxt_content_1 = (
        pbtxt_content_1
        + "item {{\n    id: {0}\n    name: '{1}'\n}}\n\n".format(i + 1, class_name)
    )
pbtxt_content_1 = pbtxt_content_1.strip()
with open(label_map_path_1, "w") as f:
    f.write(pbtxt_content)
    print('created label_map.pbtxt ')
!python /mydrive/customTF1/generate_tfrecord.py train_labels.csv  label_map.pbtxt images/ train.record
!python /mydrive/customTF1/generate_tfrecord.py test_labels.csv  label_map.pbtxt images/ test.record
!python irfan_model_main.py --
pipeline_conFig._path=/mydrive/customTF1/data/ssd_mobilenet_v2_coco.conFig. --
model_dir=/mydrive/customTF1/training --num_train_steps=200000 --sample_1_of_n_eval_examples=1 --
alsologtostderr
```

**Source code for real-time video detection**

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo_1(filename='irfan_photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
```

```
    div.appendChild(video);
    video.srcObject = stream;
    await video.play();


    google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

        await new Promise((resolve)  => capture.onclick = resolve);

    const canvas = document.createElement('canvas');
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;
    canvas.getContext('2d').drawImage(video, 0, 0);
    stream.getVideoTracks()[0].stop();
    div.remove();
    return canvas.toDataURL('image/jpeg', quality);
  }
  ''')
 display(js)
 data = eval_js('takePhoto({})'.format(quality))
 binary = b64decode(data.split(',')[1])
 with open(filename, 'wb') as f:
  f.write(binary)
 return filename
from IPython.display import Image
try:
 filename = take_photo_1()
 print('Saved to {}'.format(filename))
  display(Image(filename))
except Exception as err:
 print(str(err))
with open('./object_detection_class_name.txt', 'r') as f:
  class_names_1 = f.read().split('\n')
COLORS  = np.random.uniform(0, 255, size=(len(class_names), 3))
model_custom = cv2.dnn.readNet(model='./bangladesh_custom_model_graph.pb', conFig.='./single_shot_det
ector_DNN.txt',  framework='TensorFlow')
image_1 = cv2.imread("./photo.jpg")
image_height_1, image_width_1, _ = image.shape
blob = cv2.dnn.blobFromImage_1(image=image, size=(300, 300), mean=(104, 117, 123), swapRB=True)
model.setInput(blob)
output_model = model.forward()
for detection in output_model[0, 0, :, :]:

  confidence_level = detection[2]

  if confidence_level > .7:

    class_id_1 = detection[1]
```

```
        class_name_1 = class_names[int(class_id_1)-1]
        color = COLORS[int(class_id_1)]
        box_x = detection[3] * image_width
        box_y = detection[4] * image_height
        box_width = detection[5] * image_width
        box_height = detection[6] * image_height
        cv2.rectangle(image, (int(box_x), int(box_y)), (int(box_width), int(box_height)), color, thickness=8)
        cv2.putText(image, class_name, (int(box_x), int(box_y - 7)), cv2.FONT_HERSHEY_PLAIN, 4, color, 4
)
cv2_imshow(image_1)
cv2.imwrite('image_result.jpg', image_1)
```

**Source code for Image detection**

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
with open('./object_detection_class_name.txt', 'r') as f:
  class_names_1 = f.read().split('\n')
COLORS = np.random.uniform(0, 255, size=(len(class_names), 3))
model_output = cv2.dnn.readNet(model='./bangladesh_custom_model_graph.pb', conFig.='./single_shot_det
ector_DNN.txt', framework='TensorFlow')
image_1 = cv2.imread('./IMG_20211114_142331.jpg')
image_height_1, image_width_1, _ = image.shape
blob = cv2.dnn.blobFromImage(image_1=image_1, size=(300, 300), mean=(104, 117, 123), swapRB=True)
model.setInput(blob)
output_1 = model.forward()
for detection in output_1[0, 0, :, :]:

  confidence_level = detection[2]

  if confidence_level > .7:

      class_id_1 = detection[1]

      class_name_1 = class_names[int(class_id_1)-1]
      color = COLORS[int(class_id_1)]
      box_x_1 = detection[3] * image_width_1
      box_y_1 = detection[4] * image_height_1
      box_width_1 = detection[5] * image_width_1
      box_height_1 = detection[6] * image_height_1
      cv2.rectangle(image, (int(box_x), int(box_y)), (int(box_width), int(box_height)), color, thickness=9)
      cv2.putText(image, class_name, (int(box_x), int(box_y - 7)), cv2.FONT_HERSHEY_PLAIN, 7, color, 7
)
cv2_imshow(image_output)
cv2.imwrite('image_result.jpg', image_output)
```