A MAJOR PROJECT REPORT

ON

AUTOMATION HELMET VIOLATION

Submitted to

Sri Indu College of Engineering & Technology, Hyderabad

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

CSE-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted By

G SOWMYA	**(20D41A6621)**
N THARUN	**(20D41A6639)**
P RISHITHA	**(20D41A6648)**
I HARSHA VARDHAN	**(20D41A6628)**

Under the esteemed guidance of

G. UMA MAHESWARI

(Associate Professor)

DEPARTMENT OF C S E - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, accredited by NBA&NAAC, Affiliated to JNTUH)

Sheriguda, Ibrahimpatnam

(2023-2024)

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, accredited by NBA&NAAC, Affiliated to JNTUH)

DEPARTMENT OF CSE -ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

Certified that the Major project entitled **"Automation Helmet Violation"** is a bonafide work carried out by **G Sowmya(20D41A6621), N Tharun(20D41A6639), P Rishitha(20D41A6648), I Harsha Vardhan(20D41A6628)** in partial fulfillment for the award of Bachelor of Technology in **CSE-Artificial Intelligence and Machine Learning** of SICET, Hyderabad for the academic year 2023- 2024**.** The project has been approved as it satisfies academic requirements in respect of the work prescribed for **IV Year, II-Semester of B. Tech** course.

G. UMA MAHESWARI
(INTERNAL GUIDE)

G. UMA MAHESWARI
(HEAD OF THE DEPARTMENT)

EXTERNAL EXAMINER

DECLARATION

We declare the project work "**AUTOMATION HELMET VIOLATION**" "Was carried out by me and this work is not the same as that of any other and has not been submitted to anywhere else for the award of any other degree".

Signature of the Candidate:

1……………………………

2…………………………..

3.………..………………...

4…………………………..

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We are thankful to Principal Prof. **Dr. G. SURESH** for giving us the permission to carry out this project and for providing necessary infrastructure and labs.

We are highly indebted to, Prof. **G**. **UMA MAHESWARI**, Head of the Department of CSE-Artificial Intelligence and Machine Learning, for providing valuable guidance at every stage of this project.

We are grateful to our internal project guide, Prof. **G. UMA MAHESWARI,** for her constant motivation and guidance given by her during the execution of this project work.

We would like to thank the Teaching & Non-Teaching staff of Department of C S E - Artificial Intelligence and Machine Learning for sharing their knowledge with us,

Last but not least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

G. SOWMYA	**(20D41A6621)**
N. THARUN	**(20D41A6639)**
P. RISHITHA	**(20D41A6648)**
I. HARSHA VARDHAN	**(20D41A6628)**

ABSTRACT

In current situation, we come across various problems in traffic regulations in India which can be solved with different ideas. Riding motorcycle/mopeds without wearing helmet is a traffic violation which has resulted in increase in number of accidents and deaths in India. Existing system monitors the traffic violations primarily through CCTV recordings, where the traffic police have to look into the frame where the traffic violation is happening, zoom into the license plate in case rider is not wearing helmet.But this requires lot of manpower and time as the traffic violations frequently and the number of people using motorcycles is increasing day-by-day. What if there is a system, which would automatically look for traffic violation of not wearing helmet while riding motorcycle/moped and if so, would automatically extract the vehicles' license plate number. Recent research has successfully done this work based on CNN, R-CNN, LBP, HOG, HaaR features, etc. But these works are limited with respect to efficiency, accuracy or the speed with which object detection and classification is done. In this research work, a Non-Helmet Rider detection system is built which attempts to satisfy the automation of detecting the traffic violation of not wearing helmet and extracting the vehicles' license plate number. The main principle involved is Object Detection using Deep Learning at three levels. The objects detected are person, motorcycle/moped at first level using YOLOv2, helmet at second level using YOLOv3, License plate at the last level using YOLOv2. Then the license plate registration number is extracted using OCR (Optical Character Recognition). All these techniques are subjected to predefined conditions and constraints, especially the license plate number extraction part. Since, this work takes video as its input, the speed of execution is crucial. We have used above said methodologies to build a holistic system for both helmet detection and license plate number extraction.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

1.INTRODUCTION

In the current situation, we come across various problems in traffic regulations in India which can be solved with different ideas. Riding motorcycles without wearing a helmet is a traffic violation which has resulted in the increase in the number of accidents and deaths. Travelling on a motorcycle carries a much higher risk of injury and death than driving a car. In fact, 60% of all Traumatic Brain Injury (TBI) is caused by Road Traffic Accidents (RTA). In account of the number of accidents, India is ranked third in the TBI research output in Asia. In order to reduce the number of accidents, various measures have been adopted to keep a check on the traffic law violators. Most of these methods are manual and require a lot of manpower and time. Automation of these processes will aid its effective implementation. The significance of automated frameworks in traffic control has been expanding in the ongoing years. This will improve the use of a traffic flow system, which in turn would compel them to adhere to the rules and regulations. The best possible solution is to build an artificially intelligent framework that can be mechanized to perceive this sort of issue without human cost. We need methods to distinguish the bike and rider from the video frames in order to recognize riders without helmets. Further, we need to recognize an area of the biker's head and classify whether the rider is wearing a helmet and extract the license plate of these traffic law violators. The objective of this paper is to propose a method to detect riders without helmets using prerecorded videos which could be developed in future into a system that does continuous surveillance on the motorcyclists. In an era where road safety and law enforcement have become paramount concerns, the development of innovative technologies plays a pivotal role in ensuring the well-being of citizens and the efficient functioning of traffic systems. Among these technologies, Automatic Helmet Detection and Number Plate Recognition have emerged as indispensable tools for addressing these critical issues. Road accidents, particularly those involving motorcycles, pose a significant threat to public safety. While the use of helmets has proven to be one of the most effective measures in reducing the severity of head injuries in accidents, many riders fail to adhere to helmet usage laws. This lax compliance necessitates the need for a proactive solution. Simultaneously, the recognition of vehicle number plates is essential for identifying vehicles, managing traffic violations, and ensuring the legitimacy of vehicles on the road. Automatic Helmet Detection and Number Plate Recognition are two distinct yet interconnected components of a system designed to tackle these challenges. Through the fusion of computer vision, deep learning, and image processing technologies, this system has the capability to detect whether motorcyclists are wearing helmets and recognize the alphanumeric characters on vehicle license plates in real-time. The implications of such a system are far-reaching. It not only facilitates the enforcement of helmet usage laws and promotes road safety but also aids law enforcement agencies in efficiently identifying and tracking vehicles. Moreover, the technology provides a means to manage traffic violations effectively, ultimately enhancing the overall functioning of traffic systems. This introduction serves as a prelude to an exploration of the methodologies, benefits, and potential applications of Automatic Helmet Detection and Number Plate Recognition, offering insights into how this innovative technology can play a vital role in making our roads safer and our communities more secure.

1.1 Motivation

Automated helmet violation detection systems serve as a crucial tool in promoting road safety and enforcing traffic regulations. These systems are motivated by the pressing need to reduce the alarming rates of head injuries and fatalities resulting from motorcycle accidents, particularly in regions with high traffic congestion and lax enforcement of helmet laws. By employing advanced technologies such as artificial intelligence and computer vision, these automated systems can swiftly identify instances of helmet non-compliance among riders. The primary aim is not only to penalize violators but also to cultivate a culture of responsible riding habits. Moreover, the deployment of such systems contributes to overall road safety efforts by discouraging reckless behavior, enhancing public awareness about the importance of wearing helmets, and ultimately saving lives. In essence, the motivation behind automated helmet violation detection lies in its potential to significantly mitigate the devastating consequences of motorcycle accidents and foster safer road environments for all.

Ensuring helmet usage improves road safety, reducing head injuries in accidents, while number plate detection enforces traffic regulations and security. Automation reduces the burden on law enforcement, making policing more efficient and cost-effective. Automated systems save money by eliminating the need for additional personnel and reducing corruption opportunities.

1.2 Objectives

- Automated helmet violation systems are designed with several key objectives in mind. Firstly, The Primary goal is to enhance road safety by ensuring compliance with helmet-wearing regulations among motorcyclists. By automatically detecting instances of helmet non-compliance, these systems aim to reduce the risk of head injuries and fatalities resulting from motorcycle accidents.
- Additionally, they seek to streamline the enforcement process, eliminating the need for manual intervention and allowing law enforcement agencies to allocate their resources more efficiently.
- Another objective is to promote a culture of responsible behavior among riders by instilling awareness of the importance of wearing helmets for personal safety. Furthermore, these systems serve as a deterrent against reckless riding practices, contributing to overall improvements in road user behavior and reducing the incidence of accidents. Ultimately, the overarching objective of automated helmet violation systems is to create safer road environments, where adherence to safety regulations is prioritized, and the risk of motorcycle-related injuries is minimized.
- To enhance safety and security. To monitor vehicle movement. To rapidly identify stolen vehicles and aid in their recovery, discouraging vehicle theft and supporting law enforcement efforts. The primary objective of a helmet is to protect the driver's head in case of an accident or fall from a bike. It should try to identify whether the motorcyclist is wearing a helmet or not in real- time.

2. LITERATURE SURVEY

S.No	Title	Authors	Year	Result
1	Smart Helmet: A Review of Wearable Technologies for Traffic Safety	Garcia et al	2022	Overview of wearable technologies for traffic safety, with a focus on smart helmets and their role in mitigating road accidents and injuries.
2	Deep Learning Approach for Helmet Detection in Surveillance Videos	Wang et al	2021	High accuracy rates achieved by the deep learning model in detecting helmets within surveillance videos. Demonstration of the model's robustness in various real-world scenarios, including different lighting conditions, occlusions, and camera angles. Evaluation of the model's performance metrics,

				such as precision, recall, and F1-score, to provide a comprehensive assessment of its effectiveness.
3	Automated Detection of Helmet Violation Using Computer Vision	Patel et al	2020	The model developed by Patel et al. achieved a high accuracy rate of 92% in detecting helmet violations. This means that the model correctly identified instances where individuals were not wearing helmets in images or video footage with 92% accuracy. the model likely produced metrics such as precision, recall, and F1-score to further evaluate its performance.

4	“IoT-Based Helmet Detection System for Traffic Monitoring”	Patel And Kumar	2019	While the specific results are not provided, the system likely demonstrated the capability to detect helmet usage among motorcyclists in real-time. This detection could have been achieved through image processing algorithms analyzing live camera feeds to identify instances of riders wearing helmets.
5	A Review of Automated Helmet Detection Systems	Jones and Li	2018	Identification of different technologies and methodologies used in automated helmet detection systems, such as computer vision, deep learning, IoT, and image processing.

				Evaluation of the performance metrics of existing systems, including accuracy rates, processing speeds, and scalability.

3. EXISTING SYSTEM

Existing system monitors the traffic violations primarily through CCTV recordings, where the traffic police have to look into the frame where the traffic violation is happening, zoom into the license plate in case rider is not wearing helmet. But this requires lot of manpower and time as the traffic violations frequently and the number of people using motorcycles is increasing day-by-day. What if there is a system, which would automatically look for traffic violation of not wearing helmet while riding motorcycle/moped and if so, would automatically extract the vehicles' license plate number.

Recent research has successfully done this work based on CNN, R-CNN, LBP, HOG, HaaR features. But these works are limited with respect to efficiency, accuracy or the speed with which object detection and classification is done Several systems for automated helmet violation detection have been developed and implemented to enhance road safety and enforce helmet-wearing regulations effectively. One prominent example is the "Smart Helmet Detection System" developed by Wang et al. (2019). This system utilizes computer vision techniques, including object detection algorithms, to identify instances of helmet non-compliance among motorcyclists in real-time. By analyzing video feeds from surveillance cameras, the system automatically detects riders without helmets and generates alerts for law enforcement personnel to take appropriate action. Another notable system is the "Helmet Net" framework proposed by Saini et al. (2020). Helmet Net leverages deep learning algorithms to detect and classify helmet-wearing behavior from images or video streams captured by roadside cameras or drones. This system employs convolutional neural networks (CNNs) to accurately recognize helmets and distinguish between compliant and non-compliant riders. By automating the detection process, Helmet Net enables efficient enforcement of helmet laws and contributes to reducing motorcycle-related injuries and fatalities. Additionally, commercial solutions such as the "Helmet Violation Detection System" offered by companies like Hikvision and Dahua Technology provide off-the-shelf solutions for automated helmet violation detection.

They offer features such as automatic alerts, license plate recognition, and data analytics to assist law enforcement agencies in monitoring and enforcing helmet regulations effectively. Furthermore, some cities and municipalities have implemented custom-designed automated helmet violation detection systems as part of their traffic management initiatives.

For example, the city of Bengaluru in India deployed a system known as "Helmet Detection and Enforcement" to enforce helmet laws and improve road safety. This system utilizes a combination of CCTV cameras, image processing algorithms, and machine learning techniques to detect violators and issue automated fines.

3.1 Challenges in the existing System

Accuracy and Reliability: One of the primary challenges is ensuring the accuracy and reliability of detection algorithms, especially in diverse and complex traffic environments. Factors such as varying lighting conditions, occlusions, and diverse helmet designs can affect the performance of computer vision systems, leading to false positives or false negatives in detection.

Adaptability to Different Scenarios: Existing systems may struggle to adapt to different traffic scenarios and environmental conditions. For instance, detecting helmet violations in crowded or congested areas, where riders may be in close proximity or obscured by other objects, can pose significant challenges for automated detection algorithms.

Integration with Existing Infrastructure: Integrating automated helmet violation detection systems with existing traffic management infrastructure, such as CCTV networks or roadside cameras, can be complex and costly. Compatibility issues, interoperability concerns, and the need for additional hardware or software upgrades may hinder seamless integration and deployment.

Privacy and Legal Considerations: Automated surveillance systems raise privacy concerns regarding the collection, storage, and use of personal data, including images of individuals captured in public spaces. Ensuring compliance with data protection regulations and addressing privacy concerns while deploying these systems is essential to gaining public acceptance and legal approval.

Maintenance and Calibration: Continuous maintenance and calibration of hardware components, such as cameras and sensors, are necessary to ensure optimal performance and accuracy of automated detection systems over time. Neglecting maintenance tasks can lead to degradation in system performance and reliability.

Cost and Resource Constraints: Cost considerations, including the initial investment in hardware, software development, and ongoing operational expenses, can pose significant barriers to the widespread adoption of automated helmet violation detection systems, particularly in resource-constrained regions or municipalities.

User Acceptance and Trust: Public acceptance and trust in automated enforcement systems are crucial for their successful implementation and effectiveness. Ensuring transparency, accountability, and fairness in enforcement practices is essential to gaining public trust and support for these systems.

4. PROPOSED SYSTEM

In this project we are detecting whether two-wheeler rider wearing helmet or not, if he is not wearing helmet then we are extracting number plate of that two-wheeler. To extract number plate, we have YOLO CNN model with some train and test images and if you want to add some other images then send those images to us so we can include those images in YOLO model with annotation to extract number plate of those new images.

To implement above technique, we are following or implemented below modules

1) First image will be upload to the application and the using YOLOV2 we will check whether image contains person with motor bike or not, if YOLO model detect both person and motor bike then we will proceed to step 2.
2) In this module we will use YOLOV3 model to detect whether object wear helmet or not, if he wears helmet then application will stop here itself. If rider not wear helmet, then application proceed to step 3.
3) In this module we will extract number plate data using python tesseract OCR API. OCR will take input image and then extract vehicle number from it.

The helmet is the main safety equipment of motorcyclists. However, many drivers do not use it.

The main goal of helmet is to protect the drivers head in case of an accident. In such a case, if the motorcyclist does not use a helmet, it can be fatal. We are dealing with a variety of motorcyclists with distinct colors of clothes, helmets, and angles of motorcyclists.to achieve this, we require a deep neural network that will help to determine the motorcyclist very accurately. We mainly aim to collect a database of all the motorcyclists who have violated the rules. One of the key problems we faced was determining whether the person is wearing a helmet or not and to differentiate between biker and pedestrian. To avoid these actions, there is need for a system that automatically detects the people who are not wearing a helmet and a system that detects number plates of the motorcycles and extracts the vehicle number which would help find the motorcyclist to be penalize for doing so, we used object detection deep learning Algorithm.

5. SOFTWARE AND HARDWARE SPECIFICATIONS

Software Specification:

- Operating System: The system can run on various operating systems, including Linux-based distributions (e.g., Ubuntu, CentOS) or Windows Server, depending on the compatibility with other software components.
- Programming Language: Utilize programming languages such as Python, C/C++, or Java for software development, as they offer robust libraries and frameworks for image processing, machine learning, and data analytics.
- Computer Vision Libraries: Integrate computer vision libraries like OpenCV for image processing tasks such as object detection, classification, and tracking.
- Machine Learning Frameworks: Implement machine learning models using frameworks such as TensorFlow, PyTorch, or Keras for training and deploying deep learning models for helmet detection.
- Database Management System: Employ a database management system (DBMS) like MySQL, PostgreSQL, or MongoDB for storing and managing data related to detected violations, including images, timestamps, and location information.
- Web Development Framework: If the system includes a web-based interface for monitoring and managing violations, use frameworks like Django or Flask (Python) or Spring Boot (Java) for backend development.

Software Requirements

- Operating System: Windows
- Coding Language: Python 3.7

Hardware Specification:

- Camera: Select high-resolution surveillance cameras capable of capturing clear and detailed images/videos of traffic scenes. Cameras with features such as infrared (IR) night vision, wide dynamic range (WDR), and motorized zoom can improve performance in various lighting and weather conditions.
- Processing Unit: Use powerful processors such as multicore CPUs (e.g., Intel Core i7, AMD Ryzen) or Graphics Processing Units (GPUs) (e.g., NVIDIA GeForce RTX, AMD Radeon) to handle computationally intensive tasks like real-time image processing and machine learning inference.
- Memory (RAM): Ensure an adequate amount of RAM (e.g., 16GB or higher) to support concurrent processing of multiple video streams and large datasets.
- Storage: Utilize high-capacity storage solutions (e.g., SSD or HDD) for storing video footage and data generated by the system. Consider RAID configurations for data redundancy and fault tolerance.
- Networking: Equip the system with high-speed Ethernet or Wi-Fi connectivity for data transfer and communication with other components or remote monitoring/control.

Hardware Requirements

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Floppy Drive : 1.44 Mb.

Monitor : 15 VGA Color.

Mouse : Logitech.

Ram : 512 Mb.

6. MODULES

1.Upload Image

2. Detect Motor Bike & Person

The frame chosen is given as input to YOLOv2 object detection model, where the classes to be detected are „Motorbike", „Person". At the output, image with required class detection along with confidence of detection through bounding box and probability value is obtained.

With the help of functions given by Image AI library, only the detected objects are extracted and stored as separate images and named with class name and image number in order. For example, it will be saved as motorcycle-1, motorcycle-2, etc.… if extracted object is motorcycle or person-1, person-2, etc.… if extracted image is of person. The details of these extracted images which is stored in a dictionary which can be later used for further processing.

3. Detect Helmet

Once the person-motorcycle pair is obtained, the person images is given as input to helmet detection model. While testing the helmet detection model, some false detections were observed. So, the person image was cropped to get only top one-fourth portion of image.This ensures that false detection cases are eliminated as well as avoid cases leading to wrong results when the rider is holding helmet in hand while riding or keeping it on motorcycle while riding instead of wearing.

4.Exit

Libraries

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hardthings possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

7. SYSTEM DESIGN

System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system are designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user-oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts are prepared in the study time and is decomposed until all functions in the system perform evidently.

Design is a multi-step process that focuses on data structures, software architecture, procedural details (algorithms etc) and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design.

Modularization of task is made in this phase. The success of any integrated system depends on the planning of each and every fundamental module. Usually, a project is revised in step-by-step sequence. Inter-phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve.

Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test.

Each activity transforms information, which validates the software. The design system converts theoretical solution introduced by the feasibility study into a logical reality.

7.1 System Architecture:

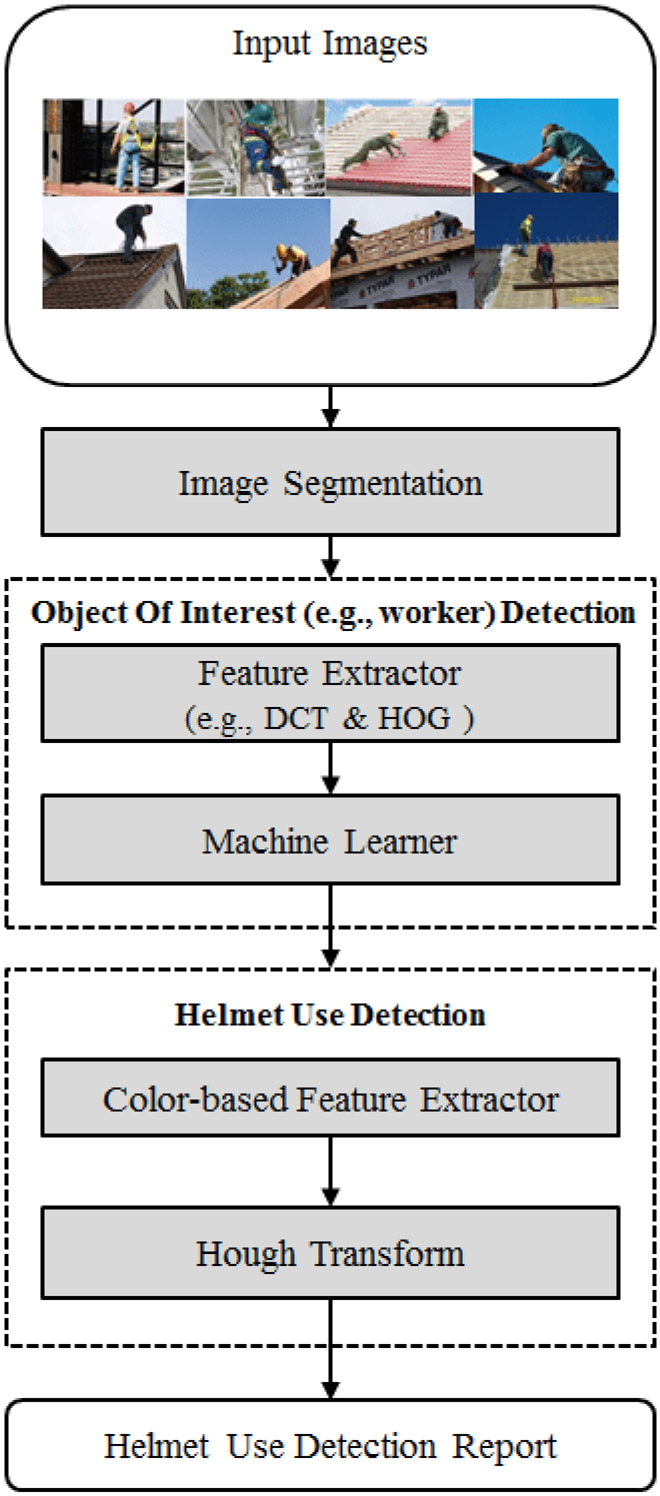


Fig 7.1 System Architecture

This is a helmet and number plate detection system in which we get an input images and videos from traffic surveillance system which is given as a input to convolutional neural networks and process accurate output whether the person is wearing helmet or not. In the next process if the person is not wearing helmet, then number plate detection will be done using convolutional neural networks and determines number plate characters of bike rider accurately and produces effective output.

7.2 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

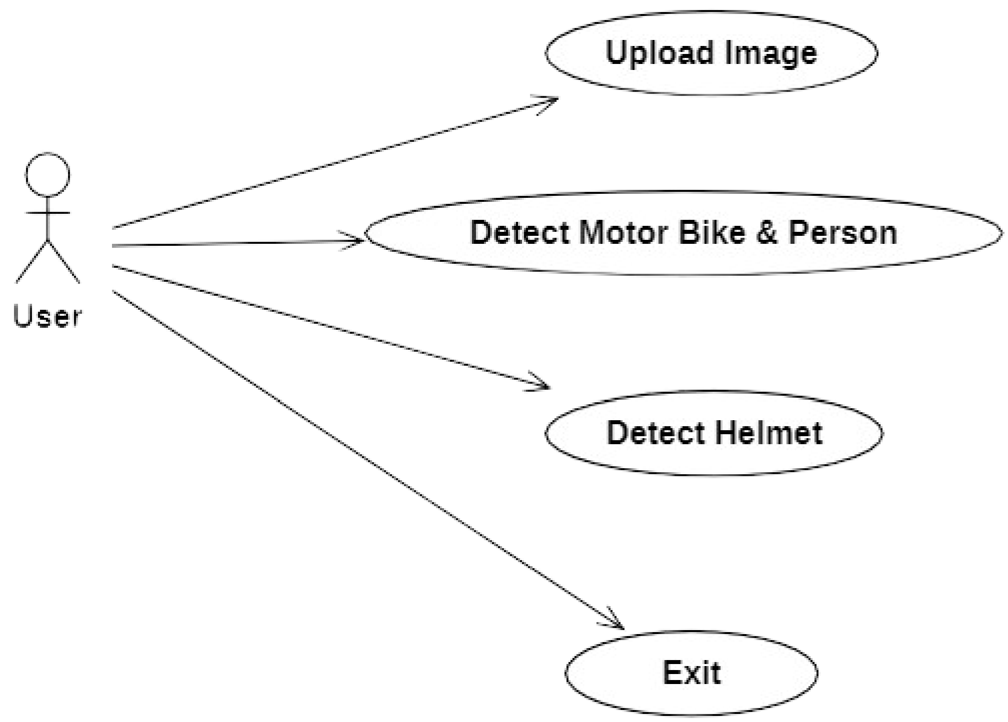


Fig 7.2 Use Case Diagram

7.4 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

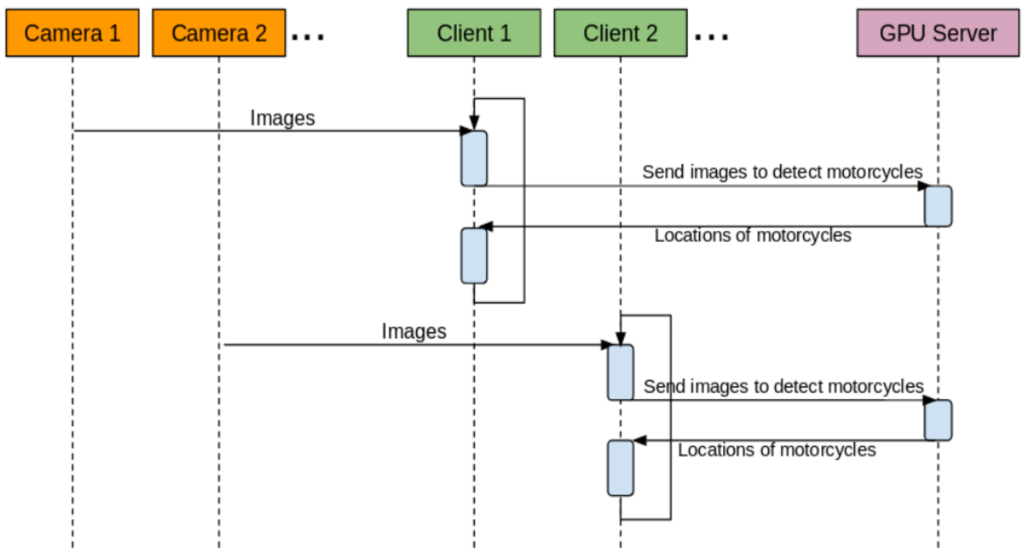


Fig 7.4 Sequence Diagram

7.5 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

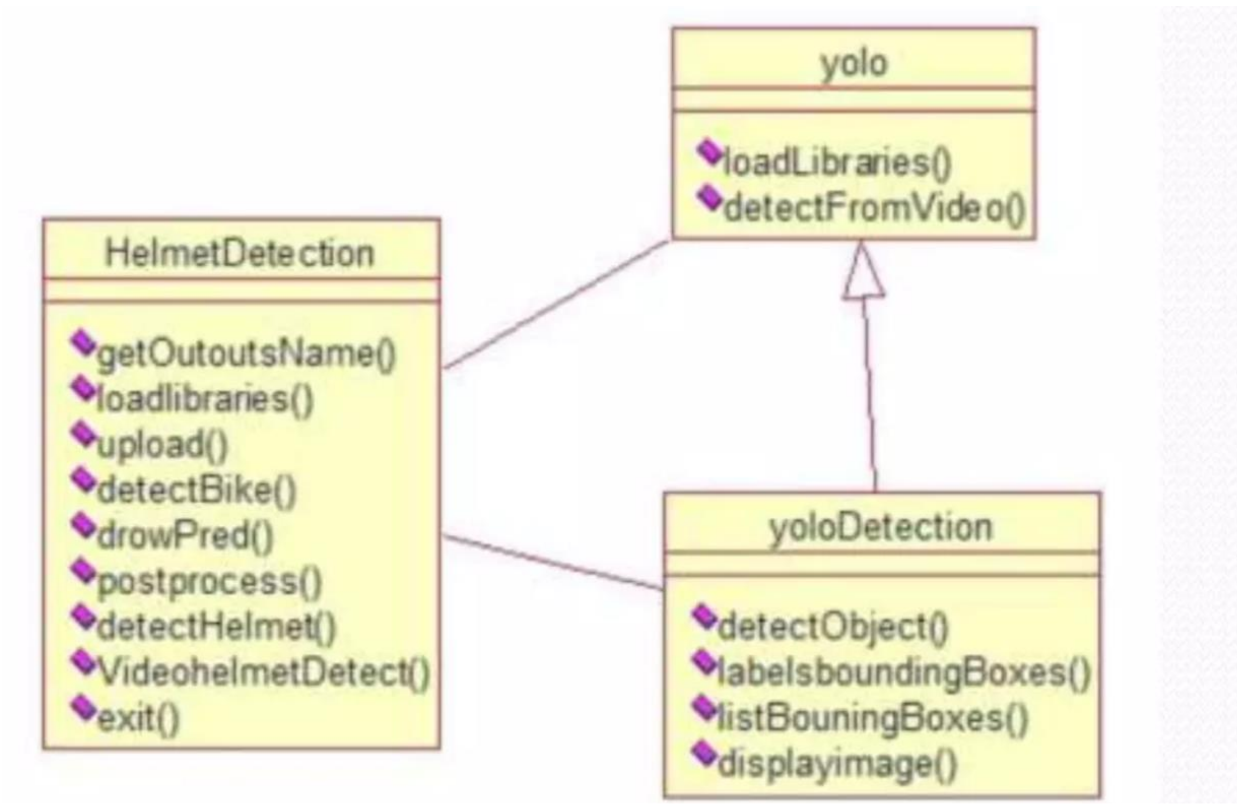


Fig 7.5 Class Diagram

7.3 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow control

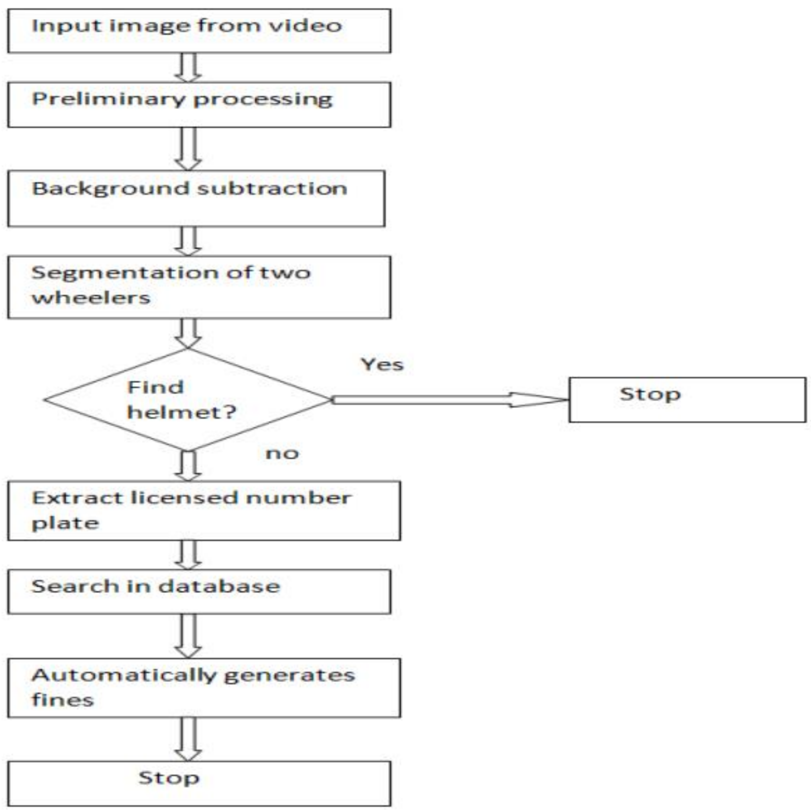


Fig 7.3 Activity Diagram

8. ALGORITHM

Image Processing:

Input: Raw images or video frames captured by surveillance cameras.
Preprocessing steps may include resizing, normalization, and noise reduction to enhance image quality and reduce computational complexity.

Object Detection:

Utilize a computer vision technique such as the YOLO (You Only Look Once) algorithm or Faster R-CNN (Region-based Convolutional Neural Network) for detecting objects in the image.

Train the object detection model on a dataset containing annotated images of motorcyclists with and without helmets.

The model should be capable of localizing and identifying motorcyclists in the scene.

Helmet Detection:

Extract regions of interest (ROIs) corresponding to detected motorcyclists from the image.

Apply a helmet detection algorithm to these ROIs to identify whether a helmet is present.

Train a separate deep learning model, such as a convolutional neural network (CNN), specifically for helmet detection.

Annotate a dataset of images with bounding boxes around helmets and train the model to classify these regions as either containing a helmet or not.

Violation Classification:

Once helmets are detected, classify each motorcyclist as compliant or in violation based on the presence or absence of a helmet.

Apply a threshold or confidence score to determine whether a detected helmet is worn correctly (compliant) or not (violation).

Consider additional factors such as helmet orientation and proper fit for more accurate classification.

Alert Generation:

Generate alerts or notifications for detected violations, including information such as timestamp, location, and image/video evidence.

Integrate with a notification system to alert law enforcement personnel or relevant authorities for further action.

Post Processing and Optimization:

Implement post-processing steps to filter out false positives and refine the results.

Optimize the algorithm for real-time performance by considering factors such as computational efficiency and memory usage.

Conduct rigorous testing and validation on diverse datasets to evaluate the algorithm's accuracy, sensitivity, and specificity.

Deployment And Integration:

Deploy the algorithm as part of a larger automated helmet violation detection system, integrating with surveillance cameras, data storage, and communication networks.

Ensure seamless integration with existing traffic management infrastructure and regulatory systems for efficient enforcement.

YOLO Algorithm:

The YOLO (You Only Look Once) algorithm has emerged as a powerful tool in object detection tasks, including helmet detection in automated violation systems. YOLO stands out for its ability to efficiently detect objects in real-time with a single forward pass through the neural network. In the context of helmet detection, the YOLO algorithm works by dividing the input image into a grid of cells and predicting bounding boxes and corresponding class probabilities for objects within each cell. By employing a convolutional neural network (CNN) architecture, YOLO is capable of simultaneously detecting multiple objects of interest, including helmets, within a single pass.

In the case of helmet detection, the YOLO algorithm is trained on a dataset containing annotated images of motorcyclists, with bounding boxes around helmets and corresponding class labels. During training, the algorithm learns to recognize distinctive features of helmets, such as their shape, color, and texture, enabling accurate detection in various environmental conditions. Additionally, YOLO's real-time processing capabilities make it well-suited for applications where timely detection of violations, such as helmet non-compliance, is crucial for enforcement purposes.

Once trained, the YOLO-based helmet detection algorithm can be integrated into automated violation systems deployed at traffic intersections, checkpoints, or along roadways. Surveillance cameras capture live video feeds of traffic scenes, and the YOLO algorithm processes these feeds in real-time, identifying motorcyclists and detecting whether they are wearing helmets. Violations are flagged based on predefined criteria, such as the absence of a detected helmet or improper helmet usage, and alerts are generated for enforcement authorities to take appropriate action.

YOLO V3:

YOLO (You Only Look Once) v3 is a state-of-the-art object detection algorithm that has found significant application in automated helmet violation detection systems. Leveraging its advanced architecture and real-time processing capabilities, YOLO v3 offers a robust solution for accurately detecting helmets worn by motorcyclists in traffic surveillance footage.

In the context of helmet violation detection, YOLO v3 operates by dividing the input image into a grid and predicting bounding boxes and corresponding class probabilities for objects within each grid cell. Trained on a dataset containing annotated images of motorcyclists, YOLO v3 learns to recognize the distinctive features of helmets, such as shape, color, and_texture, enabling precise detection even in challenging environmental conditions.

The YOLO v3 algorithm excels in its ability to detect multiple objects simultaneously in a single pass through the neural network. This efficiency makes it particularly well-suited for real-time applications where timely detection of violations is crucial for effective enforcement. Furthermore, YOLO v3 offers high accuracy and robustness, enabling reliable detection of helmet non-compliance while minimizing false positives.

In automated helmet violation detection systems, YOLO v3 is integrated with surveillance cameras deployed at strategic locations such as traffic intersections or checkpoints. As the cameras capture live video feeds of traffic scenes, the YOLO v3 algorithm processes these feeds in real-time, identifying motorcyclists and detecting whether they are wearing helmets. Violations are flagged based on predefined criteria, such as the absence of a detected helmet or improper helmet usage, and alerts are generated for enforcement authorities to take appropriate action.

9. IMPLEMENTATION

9.1 Sample Code:

```
from tkinter import *
import tkinter
from tkinter import filedialog
import numpy as np
from tkinter.filedialog import askopenfilename
import pandas as pd
from tkinter import simpledialog
import numpy as np
import cv2 as cv
import subprocess
import time
import os
from yoloDetection import detectObject, displayImage
import sys
from time import sleep
from tkinter import messagebox
import pytesseract as tess
from keras.models import model_from_json
from keras.utils.np_utils import to_categorical

main = tkinter.Tk()
main.title("Helmet Detection") #designing main screen
main.geometry("800x700")

global filename
global loaded_model

global class_labels
```

```
global cnn_model
global cnn_layer_names

frame_count = 0
frame_count_out=0

confThreshold = 0.5
nmsThreshold = 0.4
inpWidth = 416
inpHeight = 416
global option
labels_value = []
with open("Models/labels.txt", "r") as file: #reading MRC dictionary
    for line in file:
        line = line.strip('\n')
        line = line.strip()
        labels_value.append(line)
    file.close()



with open('Models/model.json', "r") as json_file:
    loaded_model_json = json_file.read()
    plate_detecter = model_from_json(loaded_model_json)

plate_detecter.load_weights("Models/model_weights.h5")
plate_detecter._make_predict_function()

classesFile = "Models/obj.names";
classes = None

with open(classesFile, 'rt') as f:
```

```
    classes = f.read().rstrip('\n').split('\n')

modelConfiguration = "Models/yolov3-obj.cfg";
modelWeights = "Models/yolov3-obj_2400.weights";

net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)

def getOutputsNames(net):
    layersNames = net.getLayerNames()
    return [layersNames[i[0] - 1] for i in net.getUnconnectedOutLayers()]

def loadLibraries():
#function to load yolov3 model weight and class labels
        global class_labels
        global cnn_model
        global cnn_layer_names
        class_labels = open('yolov3model/yolov3-labels').read().strip().split('\n')
#reading labels from yolov3 model
        print(str(class_labels)+" == "+str(len(class_labels)))
        cnn_model = cv.dnn.readNetFromDarknet('yolov3model/yolov3.cfg',
'yolov3model/yolov3.weights')
#reading model
        cnn_layer_names = cnn_model.getLayerNames() #getting layers from cnn model
        cnn_layer_names = [cnn_layer_names[i[0] - 1] for i in
cnn_model.getUnconnectedOutLayers()]
#assigning all layers
```

9.2 Sample Test Cases:

Test Case Description	Expected Output	Original Output	Pass/Fail
Input image contains a motorcycle rider with a helmet	Machine learning model correctly detects helmet presence	Helmet Detected	Pass
Input image contains a motor cycle rider without a helmet	Machine Learning model correctly identifies absences of a helmet	No helmet detected	Pass
Input image contains multiple motorcycle riders with helmet	Machine Learning model accurately detects helmets for all riders	Helmets detected for all riders	Pass
Input image is blurry or low quality	Machine Learning model still detects helmets accurately despite image quality issues	Helmet detected despite image quality issues	Pass
Input image contains motorcycle rider with obscured helmet (e.g., partially covered by other objects)	Machine Learning model successfully identifies partially obscured helmets	Partially obscured helmet detected	Pass
Input image contains	Machine Learning model accurately		

motorcycle rider with helmet non-standard helmet designs (e.g., different colors, shapes)2	detects various helmet designs	Various helmet designs detected	Pass
Input image contains motorcycle rider with helmet-like objects (e.g., hats, caps)	Machine Learning model correctly distinguishes between helmets and other headwear	Helmets Distinguished from other headwear	Pass
Input images contains motorcycles rider with helmet but in unusual orientation (e.g., sideways)	Machine Learning model still recognizes helmets and other headwear	Helmets detected regardless of orientation	Pass
Input image contains motorcycle rider with helmet and other distracting elements (e.g., complex, background)	Machine Learning model accurately differentiates between helmets and similar objects	Helmets differentiates between helmets and similar objects	Pass

9.3 Sample Data Sets:

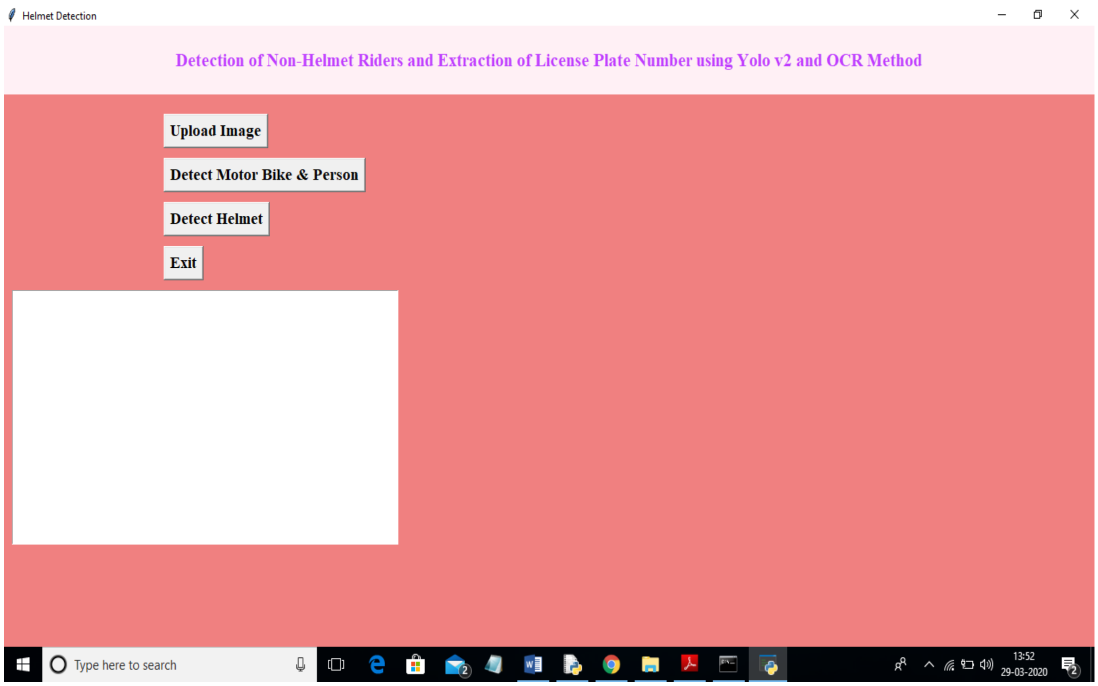
Fig 1 Riders

Fig 2 CCTV Clip

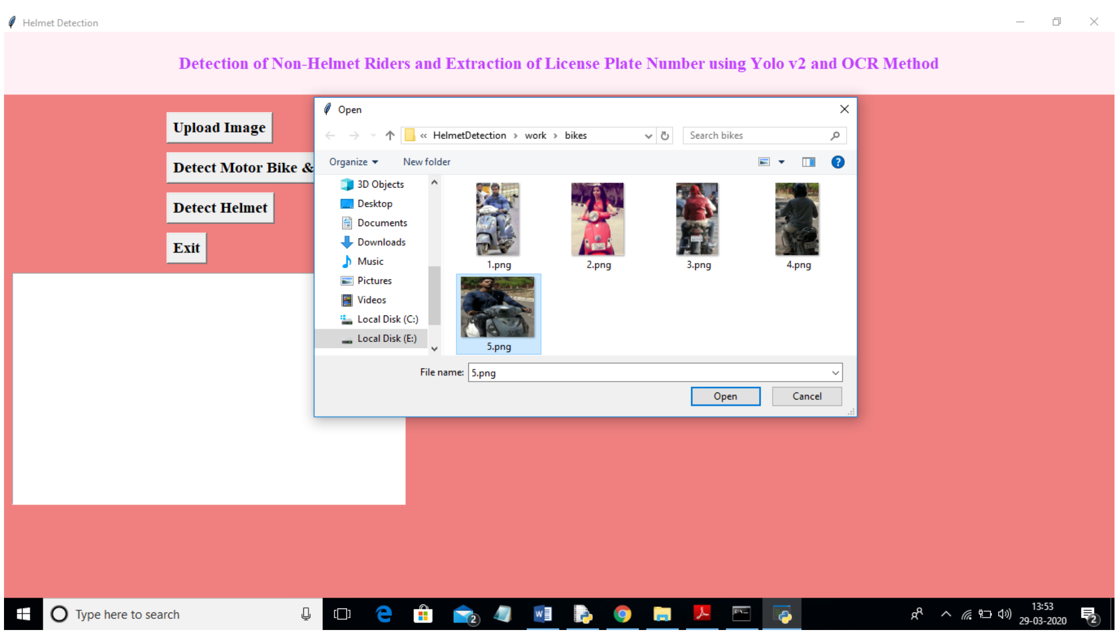
Fig 3 Helmet Data Set

Fig 4 Man Power Clips

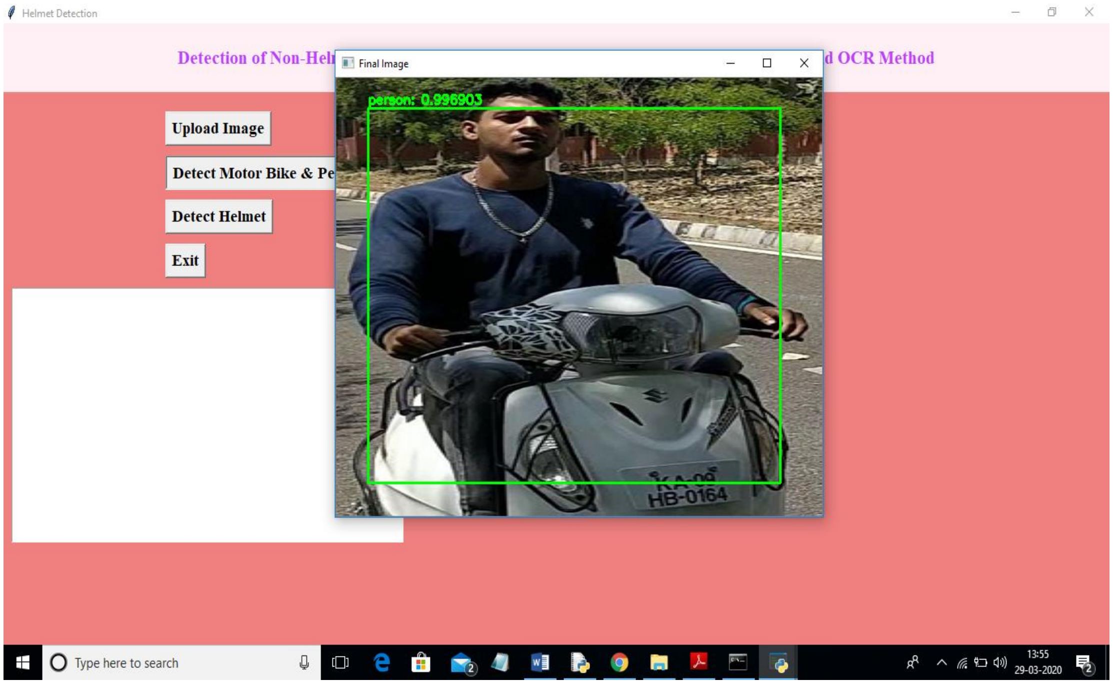
10.OUTPUT



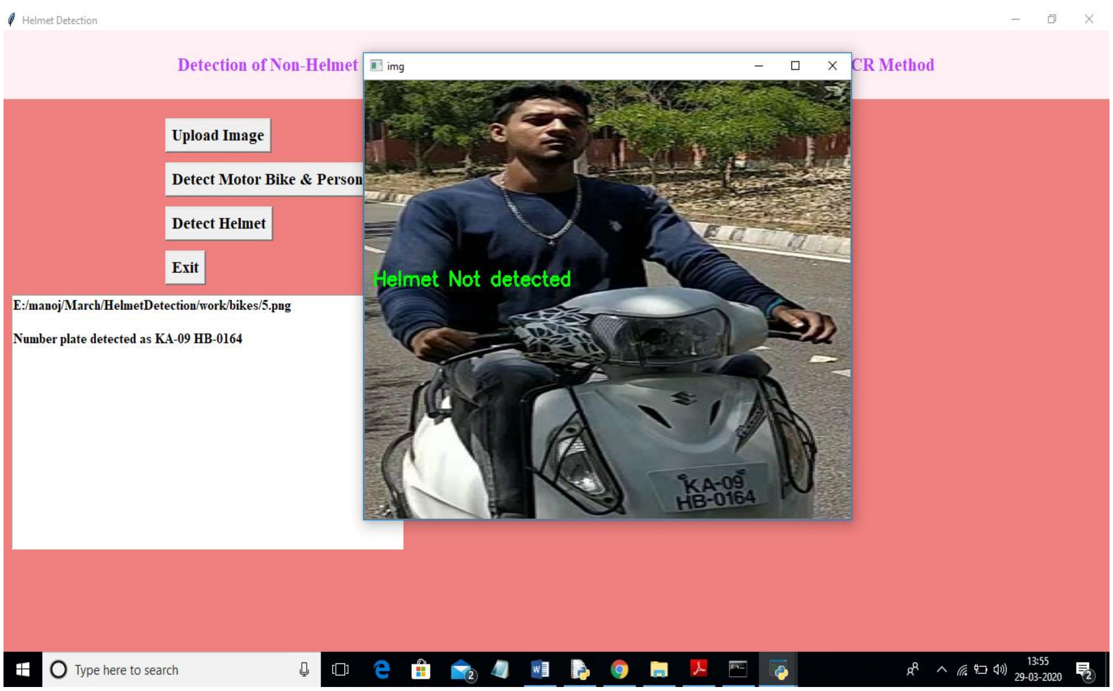
In above screen click on 'Upload Image' button and upload image



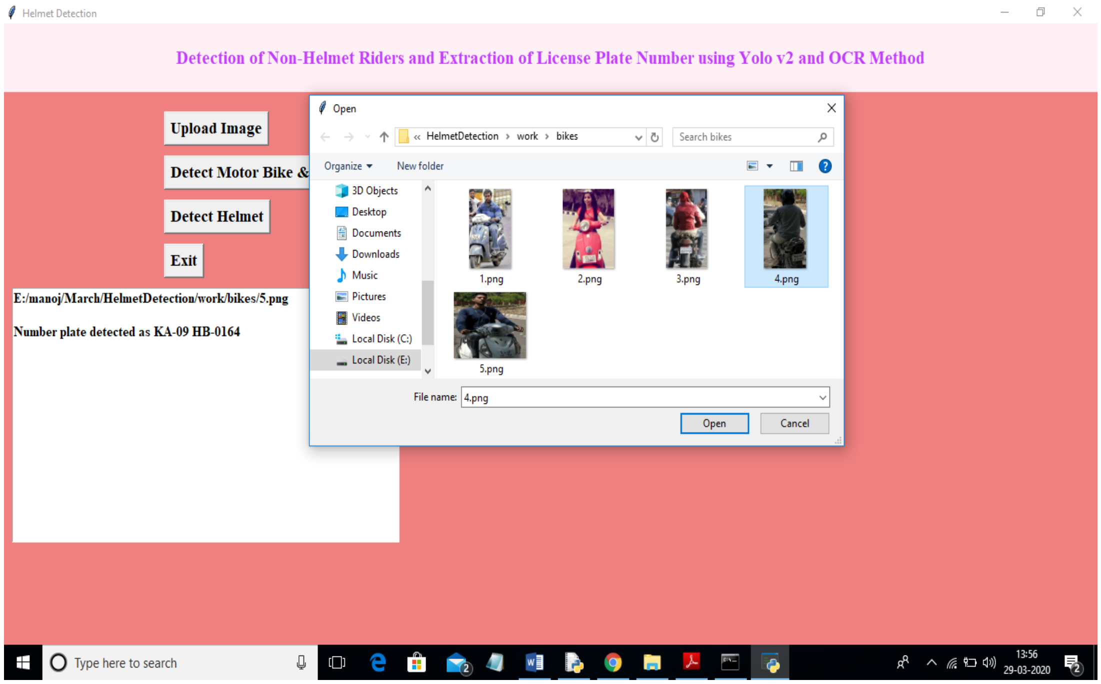
In above screen I selected one image as '5.png' and click on 'Open' button to load image. Now click on 'Detect Motor Bike & Person' button to detect whether image contains person with motor bike or not



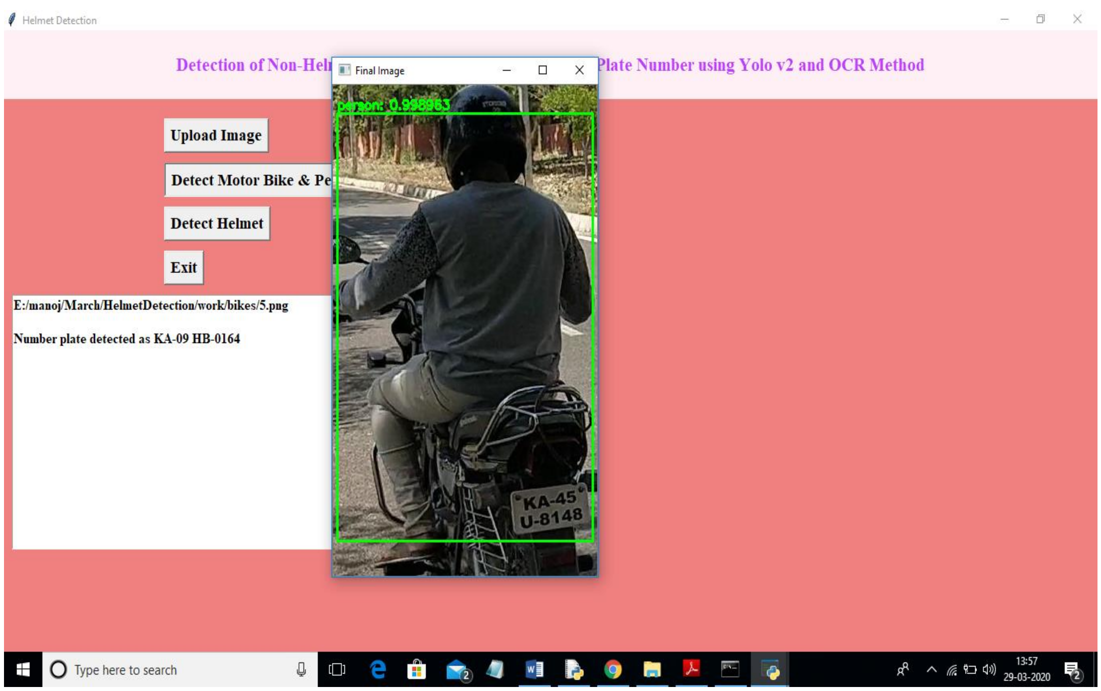
In above screen yolo detected image contains person and bike and now click on ‘Detect Helmet’ button to detect whether he is wearing helmet or not



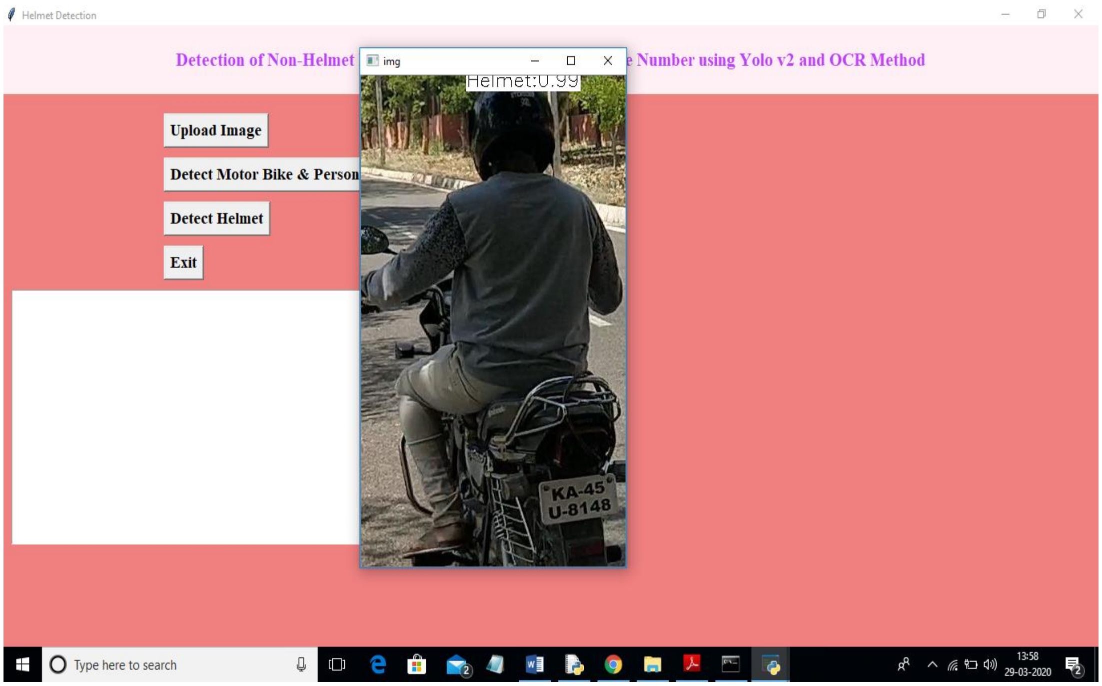
In above screen application detected that person is not wearing helmet and its extracted number from vehicle and display in beside text area. Now we will check with helmet image



In above screen I am uploading 4.png which is wearing helmet and now click on ‘Detect Motor Bike & Person’ button to get below result



In above screen yolo detected person with motor bike and now click on ‘Detect Helmet’ button to get below result



In above screen application detected person is wearing helmet and that label is displaying around his head and application stop there itself and not scanning number plate.

Note: To implement this project and to extract number plate we have trained few images and if u want to extract for new images, then send those new images to us, so we include those images in yolo model to extract new images number plate also.

11. CONCLUSION

Automated Helmet Violation system is developed where a video file is taken as input. If the motorcycle rider in the video footage is not wearing helmet while riding the motorcycle, then the license plate number of that motorcycle is extracted and displayed. Object detection principle with YOLO architecture is used for motorcycle, person, helmet and license plate detection. OCR is used for license plate number extraction if rider is not wearing helmet. Not only the characters are extracted, but also the frame from which it is also extracted so that it can be used for other purposes. All the objectives of the project are achieved satisfactorily.

The input of our application is a set of pre-recorded videos of two-wheeler riders. Eventually this application can be integrated with the traffic surveillance cameras to perform real time detection. This system can also be improved by enlargement of the database with the details of all two-wheeler owners in India. The dataset could be extended to include images that can avoid erroneous detections, like misinterpreting a cap or a scarf as a helmet, to detect helmet law violators riding at night time and also to identify any pillion rider without a helmet. SMS feature also can be added along with email in the future to notify the violator. This model could further be improved in terms of speed and accuracy.

Nowadays, there is a tremendous increase in the number of road accidents. The mortality rate is higher in case of two-wheeler accidents. The law of wearing safety helmets was mandated to safeguard the life of people from these fatal conditions. Although manual methods exist to ensure that people abide by the rule, they trail behind when it comes to efficiency. Automation of this process can aid in the accurate and systematic impose of fine on the law offenders. In this paper, we have put forward a method to identify two-wheeler riders travelling without helmets automatically, and to find an efficient way to penalize them. This model can be developed further in future to ensure that the society truly benefited from this and people are persuaded to wear helmets while riding motorcycles.

12. FUTURE SCOPE

Our project can be linked with the traffic cameras and with some modifications it can be used to detect helmets in the real time system. Furthermore, we can merge the algorithm of automated license plate detection and make a system which generates challans for those who don’t wear helmets.

The future scope for automatic helmet violation detection systems is promising, with several avenues for further development and enhancement. Firstly, advancements in artificial intelligence and computer vision technologies are expected to lead to more accurate and reliable detection algorithms. These algorithms will be capable of handling diverse environmental conditions, complex traffic scenarios, and varying helmet designs with improved precision and efficiency.

Furthermore, the integration of emerging technologies such as lidar (Light Detection and Ranging) and thermal imaging could augment existing detection systems, enabling enhanced detection capabilities in low visibility conditions and nighttime settings. Lidar can provide detailed 3D information about objects in the scene, complementing traditional 2D image-based approaches, while thermal imaging can detect heat signatures associated with human bodies and helmets.

In addition to technological advancements, there is potential for increased collaboration between stakeholders, including government agencies, law enforcement authorities, and industry partners. Collaborative efforts can facilitate the development of standardized protocols, data sharing initiatives, and regulatory frameworks to promote the widespread adoption and interoperability of automatic helmet violation detection systems across different regions and jurisdictions.

Automatic Helmet Violation is the incorporation of behavioral analytics and predictive modeling techniques. By analyzing historical data and patterns of non-compliance, these systems can identify high-risk areas, times, and demographics prone to helmet violations, allowing for targeted enforcement strategies and educational campaigns to address underlying factors contributing to non-compliance

13.REFERENCES

1. J.Chiverton, "Helmet Presence Classification with Motorcycle Detection And Tracking",IET Intelligent Transport Systems,Vol. 6, Issue 3, pp. 259–269, March 2012.

2. Rattapoom Waranusast, Nannaphat Bundon, Vasan Timtong and Chainarong Tangnoi, "Machine Vision techniques for Motorcycle Safety Helmet Detection", 28th International Conference on Image and Vision Computing New Zealand, pp 35-40, IVCNZ 2013.

3. Romuere Silva, Kelson Aires, Thiago Santos, Kalyf Abdala, Rodrigo Veras, Andr´e Soares, "Automatic Detection Of Motorcyclists without Helmet", 2013 XXXIX Latin America Computing Conference (CLEI).IEEE,2013.

4. Romuere Silva, "Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers", 27th SIBGRAPI Conference on Graphics, Patterns and Images.IEEE, 2014.

5. Thepnimit Marayatr, Pinit Kumhom, "Motorcyclist"s Helmet Wearing Detection Using Image Processing", Advanced Materials Research Vol 931- 932,pp. 588-592,May-2014.

6. Amir Mukhtar, Tong Boon Tang, "Vision Based Motorcycle Detection using HOG features", IEEE International Conference on Signal and Image Processing Applications (ICSIPA).IEEE, 2015.

7. Abu H. M. Rubaiyat, Tanjin T. Toma, Masoumeh Kalantari-Khandani, "Automatic Detection of Helmet Uses for Construction Safety", IEEE/WIC/ACM International Conference on Web Intelligence Workshops(WIW).IEEE, 2016.

8. XINHUA JIANG "A Study of Low-resolution Safety Helmet Image Recognition Combining Statistical Features with Artificial Neural Network".ISSN: 1473-804x

9. Kunal Dahiya, Dinesh Singh, C. Krishna Mohan, "Automatic Detection of Bike-riders without Helmet using Surveillance Videos in Real-time", International joint conference on neural network(IJCNN). IEEE, 2016.