Helmet Detection System for Two Wheeler Vehicles Using Machine Learning

(Bachelor of Technology Degree in Computer Science & Engineering)

Submitted by

Kartikeya Bhatt

Abhishek Joshi

Gaurav Singh

Under the guidance of

Dr. Vikrant Sharma (Assistant Professor)

Dr. Satvik Vats (Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNIVERSITY

MAY, 2024

CERTIFICATE

This is to certify that the thesis titled **“Helmet Detection System For Two Wheeler Vehicles Using Machine Learning”** submitted by **Kartikeya Bhatt, Abhishek Joshi, Gaurav Singh**, to Graphic Era Hill University for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by them under our supervision. The contents of this project in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Dehradun
Date: 30/04/2024

Name of guides

Dr. Vikrant Sharma
(Assistant Professor)
GEHU, Dehradun

Dr. Satvik Vats
(Assistant Professor)
GEHU, Dehradun

ACKNOWLEDGEMENT

I would like to particularly thank our project guides, Dr. Vikrant Sharma and Dr. Satvik Vats for their patience, support, and encouragement throughout the completion of this project.

At last, but not the least I greatly indebted to all other persons who directly or indirectly helped us during this project.

Kartikeya Bhatt
University Roll No : 2018423
B.Tech CSE-B-VIII Sem
GEHU, Dehradun

Abhishek Joshi
University Roll No : 2018078
B.Tech CSE-B-VIII Sem
GEHU, Dehradun

Gaurav Singh
University Roll No : 2061812
B.Tech CSE-B-VIII Sem
GEHU, Dehradun

ABSTRACT

This paper presents a comprehensive study and implementation of a vehicle helmet detection system aimed at enhancing road safety measures using advanced computer techniques that involve real-time surveillance. Wearing helmets significantly reduces the risk of severe head injuries in road accidents by providing a crucial protective barrier against impact forces during the collision. This safety measure acts as a pivotal safeguard, mitigating the direct effects of collisions and reducing the probability of traumatic brain injuries or fatal head trauma. This protection is essential for threatening the direct impact of accidents and underrating the chances of severe head harms or certain damage. According to the 2023 report each World Health Organization (WHO), promoting helmets right can lower the risk of deadly harm by 42% and the risk of head harms by 69%. The system is capable of analysing traffic video footage to identify moving objects and classify vehicles further the system incorporates a helmet detection component to ensure compliance with safety regulations by verifying whether both the rider and pillion passengers are wearing helmets.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

GEHU	**Graphic Era Hill University**
ML	**Machine Learning**
YOLO	**You Only Look Once**
CNN	**Convolutional Neural Networks**

CHAPTER 1

INTRODUCTION

Wearing helmets while riding motorcycles or engaging in other two-wheeled activities is of paramount importance for ensuring road safety and preventing serious injuries. Helmets serve as the first line of defense against head injuries and are instrumental in mitigating the impact of collisions and accidents. The human head is particularly vulnerable to injuries during road accidents due to its exposed nature and susceptibility to traumatic forces. In the event of a crash, the head is prone to direct impacts with hard surfaces, resulting in severe head trauma, concussions, or even fatalities. Helmets act as protective gear, absorbing and dispersing the forces generated during a collision, thereby reducing the likelihood and severity of head injuries.

The effectiveness of helmets in preventing injuries has been well-documented by numerous scientific studies and research initiatives. According to the World Health Organization (WHO), helmets can reduce the risk of fatal head injuries by as much as 69% and the risk of death in a motorcycle crash by 37%. These statistics underscore the critical role that helmets play in safeguarding the lives and well-being of motorcycle riders and passengers. By providing a protective barrier around the head, helmets help to cushion the impact of a collision and minimize the risk of traumatic brain injuries, skull fractures, and other life-threatening injuries.

In addition to preventing fatalities and serious injuries, helmets also offer a range of other benefits in terms of road safety and public health. By reducing the severity of head injuries, helmets contribute to shorter hospital stays, lower medical expenses, and faster recovery times for injured riders. This, in turn, alleviates the burden on healthcare systems and reduces the socioeconomic costs associated with road traffic injuries. Furthermore, wearing helmets promotes a culture of safety consciousness and responsible riding behaviour among motorcyclists, encouraging adherence to traffic laws and regulations.

Moreover, helmets play a crucial role in enhancing overall traffic safety and reducing the incidence of road accidents. Studies have shown that helmet-wearing rates are inversely correlated with the frequency and severity of motorcycle-related injuries and fatalities. Countries and regions with high rates of helmet usage typically experience lower rates of motorcycle-related accidents and fatalities compared to those with lower helmet-wearing rates. This highlights the importance of promoting helmet use as a key strategy for improving road safety and reducing the human toll of traffic accidents.

From a legal and regulatory perspective, helmets are mandated by law in many jurisdictions as a means of protecting the safety and well-being of motorcycle riders and passengers. Laws requiring helmet use vary by country and region but are generally aimed at promoting safety and reducing the risk of injuries in motorcycle accidents. Enforcement of helmet laws is essential for ensuring compliance and maximizing the protective benefits of helmets. Law enforcement agencies play a crucial role in monitoring helmet usage and enforcing relevant laws to deter non-compliance and promote safe riding practices.

In addition to legal mandates, educational campaigns and public awareness initiatives are vital for promoting helmet use and fostering a culture of safety on the roads. Public education campaigns can raise awareness about the importance of wearing helmets, dispel misconceptions about helmet efficacy, and provide information on how to choose and properly fit helmets for maximum protection. These campaigns can target motorcycle riders, passengers, and other road users, emphasizing the importance of helmets as a vital safety precaution.

Furthermore, technological advancements and innovations in helmet design have led to the development of more advanced and effective helmet models. Modern helmets incorporate features such as impact-absorbing materials, aerodynamic designs, and ventilation systems to enhance comfort and protection for riders. Advanced helmet technologies, such as Bluetooth connectivity, integrated cameras, and heads-up displays, offer additional functionalities and conveniences while maintaining safety standards.

In conclusion, helmets are indispensable safety devices that play a crucial role in protecting motorcycle riders and passengers from serious injuries and fatalities in road accidents. The benefits of helmets extend beyond individual protection to encompass broader societal benefits, including improved road safety, reduced healthcare costs, and enhanced public health outcomes. By promoting helmet use through legal mandates, educational campaigns, and technological innovations, stakeholders can work together to create safer roads and communities for everyone.

The escalation in road accidents due to population growth and the surge in vehicles has exacerbated traffic congestion. Monitoring whether individuals are wearing helmets amidst the vast number of vehicles presents a daunting task. Automating this process through a model capable of detecting helmet usage emerges as a solution to streamline enforcement efforts and bolster road safety. This study endeavors to develop such a model leveraging advanced computer vision and machine learning techniques, notably employing OpenCV and deep learning algorithms. YOLO (You Only Look Once) is a popular family of object detection models known for their efficiency and accuracy. YOLO revolutionized the field of object detection by introducing a single unified neural network architecture that could perform object detection and classification in real-time.

YOLOv5 is the latest iteration of the YOLO family of object detection models, developed by Ultralytics. It builds upon the success of previous versions, aiming to further improve detection accuracy and speed.

Encouraging helmet use among motorcycle riders is pivotal for curbing the incidence of severe head injuries in road accidents. Helmets serve as critical protective gear, mitigating the impact of collisions and reducing the risk of traumatic brain injuries or fatal head trauma. A 2023 report by the World Health Organization (WHO) underscores the efficacy of helmet promotion, noting a 42% reduction in fatal injuries and a 69% decrease in head injuries. Thus, ensuring compliance with helmet regulations remains paramount for enhancing overall road safety. YOLOv8 is an enhanced version of the YOLO object detection model, developed to address specific challenges and improve overall performance. Building upon the architecture of previous YOLO versions, YOLOv8 incorporates additional features and optimizations to achieve better accuracy and efficiency. In countries like India, characterized by significant two-wheeler sales and dense traffic, enforcing helmet usage regulations is imperative. India leads globally in two-wheeler sales, with over 210 million registered vehicles as of 2022. However, the escalating accident rates underscore the need for robust traffic regulation measures, including advanced helmet detection systems. Mandated under the Motor Vehicle Act of India, helmet usage for riders is stipulated under Section 129. Implementing advanced helmet detection technology can bolster law enforcement efforts and contribute to mitigating traffic accidents. YOLOv8 may include improvements in network architecture, training procedures, and post-processing techniques to enhance object detection capabilities.

This paper seeks to advocate helmet usage among motorcycle riders by developing a robust helmet detection system. The system harnesses the capabilities of YOLOv8, a deep learning algorithm renowned for object detection tasks. While proficient in identifying objects like people and cars, additional training was undertaken to enhance its helmet detection prowess. By training the model on a dataset comprising over 1375 helmet images, it attains proficiency in discerning whether individuals are wearing helmets while riding. YOLOv5 and YOLOv8 play crucial roles in the development of such systems, providing state-of-the-art object detection capabilities that are essential for ensuring road safety.

When applied to traffic video footage, the system swiftly identifies riders and passengers on bikes. Individuals complying with safety regulations by wearing helmets are delineated in a green frame, signifying adherence. Conversely, those not wearing helmets are outlined in red, serving as a visual reminder to adhere to safety guidelines. This intelligent technology transcends mere rule enforcement; it assumes a pivotal role in promoting road safety and accident prevention by fostering responsible riding habits.

The primary objective of this research is to develop a comprehensive vehicle helmet detection system capable of accurately identifying helmet usage among motorcycle riders in traffic scenarios. Leveraging advanced computer vision techniques and deep learning algorithms, the system aims to automate the process of helmet detection, thereby enhancing law enforcement capabilities and promoting adherence to safety regulations.

To achieve this objective, the research is structured into two main phases:

Phase 1: Two-Wheeler Vehicle Identification

The first phase of the research focuses on accurately identifying two-wheelers within real-time traffic footage. To achieve this, a pre-trained YOLOv5 model is employed, and a custom Python script is developed to extract cropped images of two-wheelers from the footage. These images are then uniformly resized to ensure consistency and enhance model accuracy. The resulting dataset of two-wheeler images serves as the input for the subsequent phase of the research.

Phase 2: Helmet Detection

In the second phase of the research, the focus shifts to helmet detection within the extracted two-wheeler images. This phase entails several key steps, including data collection, preparation, model training, validation, and system implementation. A dataset comprising images of motorcycle riders, both wearing and not wearing helmets, is collected and augmented to ensure robust model performance. Each image is resized to a uniform size and manually labeled to facilitate precise helmet detection.

Next, the YOLOv8 algorithm is employed for model training, leveraging its advanced object detection capabilities to accurately identify helmets within the images. A validation set comprising a subset of images plays a crucial role in refining the model's accuracy and versatility. Once trained, the model is implemented to detect helmets in real-time traffic videos, marking helmet wearers with a green outline and non-wearers with a red outline for easy identification.

The integration of both YOLOv5 and YOLOv8 algorithms contributes significantly to the system's efficacy in identifying helmet usage in traffic footage. YOLOv5 facilitates the accurate identification of two-wheelers, while YOLOv8 excels in detecting helmets within these images. By leveraging the capabilities of both algorithms, the system provides clear visual representations of helmet usage, promoting road safety and encouraging compliance with safety regulations.

Overall, this research aims to demonstrate the potential of advanced computer vision and machine learning techniques in automating helmet detection and enforcement efforts. By developing a comprehensive vehicle helmet detection system, this research contributes to the ongoing efforts to enhance road safety and reduce the incidence of road traffic injuries. Through the integration of cutting-edge technologies, such as YOLOv5 and YOLOv8, this system offers a scalable and efficient solution for promoting helmet usage and ensuring compliance with safety regulations in diverse traffic scenarios.

In the first step of the algorithm, we leverage the capabilities of the YOLOv5 pre-trained model to identify two-wheeler vehicles within each frame of the video footage. As the algorithm iterates through each frame, it extracts the relevant frame and identifies any vehicles with riders present. If a two-wheeler vehicle is detected, the algorithm outlines the vehicle, including the rider, with a distinctive blue box, visually highlighting its presence within the frame.

Upon detecting a two-wheeler vehicle, the algorithm proceeds to check for helmet usage among the riders using the custom-trained YOLOv8 model. This model has been specifically trained to recognize helmets and accurately determine whether a rider is wearing one or not. If the rider is wearing a helmet, the algorithm outlines the rider's head with a green box, indicating compliance with safety regulations. Conversely, if the rider is not wearing a helmet, the algorithm outlines the rider's head with a red box, serving as a visual reminder to adhere to safety guidelines.

Throughout the execution of the algorithm, frames containing vehicles other than two-wheelers are skipped to focus exclusively on relevant vehicles. This ensures that computational resources are efficiently utilized and processing time is optimized for accurate detection and analysis of two-wheeler vehicles and helmet usage.

By integrating this algorithm into our vehicle helmet detection system, we aim to automate the process of identifying helmet usage among motorcycle riders in traffic scenarios. This algorithm serves as the foundation for our system's functionality, leveraging advanced deep learning techniques to enhance road safety and promote adherence to safety regulations. Through the seamless integration of pre-trained and custom-trained models, our system provides real-time insights into helmet compliance, facilitating proactive enforcement efforts and fostering a culture of safety on the roads.

CHAPTER 2

LITERATURE REVIEW

In a study by R.V. et al. [8], some researchers worked on a way to find out if people on motorcycles were wearing helmets, especially those on two-wheelers. They wanted to do this by looking at parts of the road that didn't change much and seeing how vehicles moved against that background. Then, they used a special method called background subtraction to pick out the vehicles that were moving and get clear pictures of them. To tell which ones were motorcycles and which weren't, they used something called a feature vector, which is like a special list of things, and a tool called a random forest classifier. They also looked at finding helmets by choosing a particular area to focus on called the Region of Interest (RoI). This helped them process things faster and pick out the features of a person's head in that area. These features were then used to figure out if the rider was wearing a helmet, which made the classifier better at its job.

M. Swapna and her team [9] had an idea to make a better system for finding helmets to keep people safe. Their plan was to mix the old methods of finding helmets with their own new system. This system was all about checking if someone on a two-wheeler had a helmet on, especially when they were in traffic. They could do this using live videos of traffic or videos that were recorded earlier. Their system had several steps like gathering and working with images, finding objects, spotting vehicles, and then checking for helmets. It was cheap to make and used free tools like OpenCV, which helps with working with images, and something called Yolo. But what's really neat is that besides just finding helmets, it could also see if someone was using their phone while driving too fast. M. Swapna and her team's mix-and-match system was a big leap in noticing who on two-wheelers had helmets and who didn't. By mixing old ways with their new system and making sure it didn't cost too much, they made a system that could do a lot.

M. Swapna's team made sure their system was up-to-date with the latest technology and easy on the wallet. They combined old ideas with their new ones to make a smart way to keep people safe on the road. Their plan was to spot if someone on a motorcycle was wearing a helmet or not. They made it so their system could work with videos of traffic that were happening right then or ones that had been recorded before. Their system had a few steps, like gathering pictures, finding things in those pictures, figuring out what was a vehicle and what wasn't, and then seeing if there was a helmet on the rider. And what's really neat is that their system wasn't expensive to put together. They used free tools like

OpenCV and Yolo, which are great for working with pictures and figuring out what's in them. But what's even cooler is that besides just finding helmets, their system could also tell if someone was using their phone while driving too fast. M. Swapna and her team made sure their system was ready to go with all the latest tricks and wouldn't break the bank. They mixed old ideas with new ones to make a clever way to keep people safe on the road.

M. Swapna and her team's work brought together the best of old and new ideas to create a system that could do a lot. Their system was all about keeping people safe on the road by making sure they wore helmets. They made their system so it could work with live videos of traffic or ones that had been recorded earlier. Their system had a few steps, like collecting pictures, finding things in those pictures, figuring out what was a vehicle and what wasn't, and then checking if there was a helmet on the rider. And what's even better is that their system didn't cost a lot to make. They used free tools like OpenCV and Yolo, which are great for working with pictures and figuring out what's in them. But what's really amazing is that besides just finding helmets, their system could also tell if someone was using their phone while driving too fast. M. Swapna and her team made sure their system was ready to go with all the latest tricks and wouldn't break the bank. They mixed old ideas with new ones to make a clever way to keep people safe on the road.

In a study conducted by Lokesh Allamki and his team [10], they introduced a sophisticated system aimed at detecting helmets while integrating machine learning techniques with automatic license plate recognition. The primary focus of their research was to extract essential components from YOLO's identified elements. This system comprises three primary modules. Initially, the researchers trained the YOLOv3 model specifically for helmet detection, ensuring its robustness even in scenarios with reflective surfaces. Moreover, the system possesses the capability to identify individuals who are not wearing helmets, prompting them to provide their driver's license. Subsequently, the system captures and stores the information from the motorist's license. During the license validation phase, optical character recognition (OCR) is employed to extract an alphanumeric string from the license. Additionally, a confidence score is generated to indicate the reliability of the OCR process.

It is noteworthy that this system is primarily designed for motorcycles. However, it is important to acknowledge its limitations, particularly the fact that the OCR module is currently only capable of processing images and does not support video input. This limitation might constrain its applicability in scenarios where real-time video analysis is necessary. Nevertheless, the integration of machine learning with automatic license plate recognition represents a significant advancement in enhancing road safety measures, particularly in ensuring compliance with helmet regulations among motorcycle riders. Further research and development efforts could potentially address the limitations of the system, thereby expanding its utility in various traffic surveillance applications.

In their investigation, Waris et al. and their research group [11] explored a Convolutional Neural Network (CNN)-based automated helmet detection system specifically designed for motorcyclists operating within intelligent transportation systems (ITS). Their study underscores the urgent necessity for implementing sophisticated monitoring techniques to bolster road safety, particularly in mitigating motorcycle accidents stemming from improper helmet usage. Employing the Faster R-CNN deep learning model, their proposed system achieves an impressive accuracy rate of 97.69%. The study accentuates the paramount importance of an accurate dataset and demonstrates the model's effectiveness in accurately distinguishing helmets from other objects. This positions the system as a promising solution for enforcing helmet laws and augmenting overall traffic safety within the ITS framework. Through their research, Waris et al. and their team contribute significantly to the ongoing efforts aimed at enhancing road safety and reducing the incidence of motorcycle-related injuries.

In their study, Himanshu Adhikari and colleagues [12] introduced an Automated Helmet Detection and License Plate Extraction system aimed at bolstering road safety measures. This innovative system harnesses the power of machine learning and image processing techniques to automatically identify motorcycle riders without helmets and extract license plate information. With a primary objective of streamlining law enforcement efforts by efficiently identifying and issuing violation tickets to non-compliant riders, the system meticulously processes traffic videos, detects moving objects (primarily motorcycles), and utilizes advanced classifiers to accurately distinguish between two-wheelers and four-wheelers. Furthermore, the system employs additional classification techniques to determine if the rider is wearing a helmet, and extracts license plate information from frames of non-compliant riders for enforcement purposes. The study comprehensively evaluates various machine learning classifiers, including Random Forest, Gradient Boosted Trees, Support Vector Machine, and Deep Neural Networks, showcasing their effectiveness in vehicle classification and helmet detection, achieving balanced accuracy in the conducted tests.

In a parallel effort, Silva et al. [13] proposed an innovative method that combines the Hough Transform (HT) with histogram-oriented gradient (HOG) for attribute extraction from images captured by roadside cameras. By meticulously establishing a database comprising 255 images, they achieved an impressive accuracy rate of 91.37% with their approach. This method represents a significant advancement in image processing techniques, particularly in the domain of roadside surveillance, by effectively extracting crucial attributes from images for subsequent analysis and decision-making processes. The high accuracy attained underscores the potential of this method to contribute to enhanced road safety measures through improved surveillance and detection capabilities. Through their respective research endeavors, both Himanshu Adhikari et al. and Silva et al. contribute substantially to the ongoing efforts aimed at mitigating road accidents and enhancing overall traffic management systems.

Boonsirisumpun et al. [14] employed a convolutional neural network (CNN) framework to address the critical issue of identifying individuals riding motorcycles without helmets. They utilized camera-captured images as inputs for training, drawing from a dataset comprising 493 images. The study explored the performance of four distinct CNN models: GoogleNet, MobileNet, VGG19, and VGG16. Notably, MobileNet emerged as the most promising model, achieving a favorable accuracy rate of 85.19%.

Their research signifies a significant step towards enhancing road safety by leveraging advanced computer vision techniques to enforce helmet-wearing regulations among motorcycle riders. By demonstrating the effectiveness of MobileNet in accurately detecting helmet usage from images, Boonsirisumpun et al. contribute valuable insights to the ongoing efforts aimed at reducing motorcycle-related injuries and fatalities. The utilization of CNN frameworks showcases the potential of deep learning methodologies in addressing complex real-world challenges, particularly in the domain of traffic safety and surveillance. Through their study, Boonsirisumpun et al. provide a foundation for the development of automated systems capable of identifying and enforcing helmet regulations, thereby contributing to the overall improvement of road safe Sridhar et al. [15] and Rajalakshmi and Saravanan et al. [16] both contribute significantly to the field of road safety through their research on helmet detection and regulatory compliance monitoring using convolutional neural networks (CNNs). These studies address the pressing need for effective methods to ensure helmet usage among motorcycle riders and to monitor compliance with safety regulations on the roads.

Sridhar et al. [15] focus on ascertaining helmet usage using YOLOv2, a deep learning model known for its efficiency in real-time object detection. They utilize deep CNNs to identify motorcycle riders who do not comply with safety regulations by not wearing helmets. The approach involves initially detecting motorcycles in images or video frames and subsequently determining whether the rider is wearing a helmet. By leveraging deep learning techniques, Sridhar et al. aim to surpass the performance of conventional algorithms and achieve more accurate and reliable results in helmet detection.

On the other hand, Rajalakshmi and Saravanan et al. [16] propose a comprehensive system designed for monitoring and managing individuals who violate regulations on the roads. Their system utilizes CNNs for various tasks, including vehicle classification, object classification, helmet detection, and mask detection. By employing suitable CNN-based models, their system can effectively identify and classify different objects and individuals on the road, including motorcycle riders without helmets. This multi-faceted approach enables the system to address a wide range of safety and regulatory compliance issues in real-world scenarios.

The utilization of CNNs in both studies highlights the significance of deep learning techniques in addressing complex problems related to road safety and traffic management. CNNs are well-suited for tasks such as object detection and classification in images and videos, making them valuable tools for developing intelligent systems for road surveillance and enforcement.

In Sridhar et al.'s study, YOLOv2 is chosen as the primary model for helmet detection due to its speed and accuracy in real-time object detection tasks. YOLOv2 (You Only Look Once version 2) is a popular deep learning model known for its ability to detect objects in images and video frames with high speed and accuracy. By integrating YOLOv2 into their framework, Sridhar et al. aim to achieve efficient and reliable helmet detection in various traffic scenarios.

Meanwhile, Rajalakshmi and Saravanan et al. adopt a more comprehensive approach by developing a system capable of performing multiple tasks related to road safety and regulatory compliance. Their system incorporates CNN-based models for vehicle classification, object classification, helmet detection, and mask detection, enabling it to address a wide range of safety and compliance issues on the roads. This holistic approach allows their system to provide comprehensive monitoring and management of traffic violations, ultimately contributing to improved road safety outcomes.

Overall, both studies underscore the importance of leveraging advanced technologies, particularly deep learning and CNNs, for addressing challenges related to road safety and regulatory compliance. By developing intelligent systems capable of detecting helmet usage and monitoring compliance with safety regulations, researchers can contribute to creating safer and more efficient transportation systems. Through their innovative approaches and contributions, Sridhar et al. and Rajalakshmi and Saravanan et al. pave the way for the development of more effective and reliable solutions for enhancing road safety and reducing traffic-related measures.

In their study, Kathane et al. [17] conducted research focusing on the enhancement of object detection using YOLOv3. The YOLO (You Only Look Once) algorithm is renowned for its efficiency in real-time object detection tasks. Kathane et al. aimed to refine deep learning models to recognize various objects, with particular emphasis on motorcycles. Their approach involved deploying three distinct deep learning models to identify different objects within images or video frames. The system they developed achieved noteworthy accuracy rates, with an 88.5% accuracy in detecting motorcycles and a 91.8% accuracy in recognizing number plates.

Cheverton et al. [18] and their associates, in their research conducted in 2014, developed a system utilizing support vector machines (SVMs) and background subtraction techniques to differentiate between bikers wearing helmets and those who are not. They constructed a dataset specifically for this purpose during the system's development phase. However, their system encountered two primary challenges. Firstly, it searched the entire frame for helmet detection, leading to increased computational requirements. Secondly, it often misidentified objects that were not wearing helmets, indicating limitations in its accuracy.

Wu et al. [19], in their 2015 study, utilized YOLOv3 and YOLO-thick models to label motorcyclists lacking helmets. They collected datasets from two sources: internally generated and sourced from the internet. Experimental results demonstrated impressive performance, with a mean average precision (mAP) of 95.15% for YOLOv3 and 97.59% for the YOLO-thick model.

These studies collectively contribute to the field of road safety and traffic management by developing innovative approaches to detect helmet usage among motorcycle riders. By leveraging advanced techniques such as deep learning and convolutional neural networks (CNNs), researchers aim to improve the accuracy and efficiency of helmet detection systems, ultimately enhancing road safety and reducing the incidence of motorcycle-related injuries and fatalities.

In Kathane et al.'s research [17], the focus was on refining object detection algorithms, particularly in the context of motorcycle detection and number plate recognition. The YOLOv3 algorithm, known for its speed and accuracy in real-time object detection tasks, served as the foundation for their work. By deploying multiple deep learning models and optimizing their performance, Kathane et al. achieved impressive accuracy rates in detecting motorcycles and recognizing number plates. These results highlight the potential of deep learning techniques in improving the accuracy and reliability of object detection systems, thereby enhancing road safety measures.

In contrast, Cheverton et al. [18] and their collaborators adopted a different approach to helmet detection using support vector machines (SVMs) and background subtraction techniques. Their system aimed to distinguish between bikers wearing helmets and those who are not by analyzing video frames captured by roadside cameras. While their approach showed promise, it faced challenges related to computational complexity and misidentification of non-helmet objects. These challenges underscore the importance of developing robust and efficient algorithms for helmet detection, especially in real-world scenarios where computational resources may be limited.

Wu et al.'s [19] study built upon the success of previous research by utilizing YOLOv3 and YOLO-thick models to label motorcyclists lacking helmets. By assembling datasets from multiple sources and conducting rigorous experiments, they demonstrated the effectiveness of these models in accurately detecting helmet non-compliance. The high mean average precision (mAP) scores achieved by their models underscore their potential for real-world applications in traffic surveillance and enforcement.

Overall, these studies represent significant contributions to the field of road safety and traffic management by developing advanced techniques for helmet detection and compliance monitoring. By leveraging state-of-the-art algorithms and methodologies, researchers aim to create more effective and reliable systems for enforcing helmet regulations and improving overall road safety. As advancements in deep learning and computer vision continue to progress, the future holds promise for further innovations in this critical area of research.

It's essential to recognize the broader implications of these studies beyond the realm of academia. Improved helmet detection systems have the potential to save lives by encouraging widespread compliance with helmet regulations among motorcycle riders. By leveraging technology to enhance traffic safety measures, these studies contribute to the overarching goal of reducing traffic-related injuries and fatalities worldwide.

In conclusion, the research conducted by Kathane et al. [17], Cheverton et al. [18], and Wu et al. [19] represents significant advancements in the field of road safety and traffic management. By developing innovative approaches to helmet detection and compliance monitoring, these studies contribute to the ongoing efforts aimed at improving road safety and reducing the incidence of motorcycle-related injuries and fatalities. As technology continues to evolve, the potential for further innovations in this critical area of research remains promising, offering hope for a safer and more sustainable transportation future.

Furthermore, there is a need for ongoing research to improve the robustness and reliability of helmet detection systems in real-world scenarios. This includes developing techniques to handle challenging conditions such as occlusions, varying perspectives, and dynamic environments. Additionally, efforts should be made to enhance the interpretability of detection results and minimize false positives and false negatives, which are critical for ensuring the effectiveness of enforcement measures.

In summary, while significant progress has been made in the development of helmet detection systems, there is still much work to be done to address the complex challenges associated with enforcing helmet regulations and improving road safety for all road users. By continuing to innovate and collaborate across disciplines, researchers can play a vital role in advancing the state-of-the-art in this critical area of research and ultimately saving lives on the roads.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 INTRODUCTION

The Helmet Detection System is a software application develop two check whether the two wheeler vehicle rider is wearing helmet or not. This system utiliszes machine learning algorithms such as YOLOv5 and YOLOv8 to identify rider with or without helmet and these algorithms are implemented using various machine learning libraries and Visual Studio Code.

By harnessing the power of deep learning algorithms, such as YOLOv8, this research endeavors to bridge the gap between safety regulations and their enforcement in real-world traffic scenarios. Through a combination of data collection, model training, and system implementation, the proposed helmet detection system aims to serve as a proactive safety measure, not only reducing the incidence of head injuries but also fostering a culture of responsible riding practices.

The subsequent sections of this paper delve into a comprehensive review of existing literature, elucidate the methodology employed for system development, present the results of empirical evaluations, and discuss avenues for future research and implementation. Through this endeavor, we endeavor to contribute to the ongoing discourse on road safety and advance the state-of-the-art in intelligent transportation systems.

Despite the unequivocal benefits of helmet usage, enforcing compliance with safety regulations poses significant challenges, especially in regions with burgeoning populations and a proliferation of two-wheelers. Manual inspection of helmet usage for each vehicle in traffic is impractical and resource-intensive, necessitating the development of automated systems capable of real-time surveillance and detection.

3.1.1 Purpose

The purpose of this project is to enhance road safety by promoting helmet usage among motorcycle riders. By implementing a model that automatically detects whether a person is wearing a helmet in traffic, the project aims to encourage compliance with helmet laws, reduce injuries during accidents, and ultimately decrease the number of road accidents. The use of advanced technology such as computer vision and machine learning facilitates efficient enforcement of helmet usage, contributing to overall traffic safety.

Enhanced Road Safety: The primary objective of the system is to enhance road safety by encouraging helmet usage among two-wheeler riders. By accurately detecting whether a rider is wearing a helmet or not, the system helps enforce safety regulations and reduces the risk of head injuries in road accidents.

Efficient Traffic Monitoring: The system contributes to efficient traffic monitoring by automating the process of helmet detection. Instead of manual inspection by law enforcement authorities, the system can analyze traffic video footage in real-time and identify non-compliant riders, thereby saving time and resources.

Cost-Effectiveness: Implementing the helmet detection system offers a cost-effective solution for enforcing safety regulations. By reducing the need for manual intervention and minimizing the risk of accidents, the system can potentially save costs associated with healthcare, insurance claims, and law enforcement.

Improved Law Enforcement: The system provides law enforcement agencies with a valuable tool for enforcing helmet usage laws. By accurately identifying non-compliant riders, authorities can take appropriate actions such as issuing fines or warnings to ensure compliance with safety regulations.

Public Awareness and Education: Beyond enforcement, the system plays a role in raising public awareness about the importance of wearing helmets while riding two-wheelers. The visual representation of helmet-wearing and non-wearing riders in traffic footage serves as a reminder to individuals to prioritize their safety on the road.

Future Development: The implementation of the helmet detection system opens avenues for future research and development in the field of road safety and intelligent transportation systems. Further improvements in accuracy, efficiency, and scalability can be explored to address evolving challenges in traffic management and accident prevention.

3.1.2 Scope

The scope of this project involves developing and implementing a model using computer vision and machine learning to detect helmet usage among motorcycle riders in traffic. It includes dataset collection, model training, real-time implementation, and evaluation, with the goal of promoting helmet usage and enhancing road safety.

Enhancement of Accuracy: Continuously improving the accuracy of helmet detection algorithms is crucial for the effectiveness of the system. Future research could focus on refining the deep learning models, optimizing hyperparameters, and expanding the dataset to enhance the system's ability to accurately detect helmets in various environmental conditions.

Real-time Performance: Further optimization of the system's performance for real-time applications is essential, especially for deployment in traffic surveillance systems. This could involve optimizing algorithms, leveraging hardware acceleration (e.g., GPUs, TPUs), and exploring techniques for reducing inference latency.

Multi-Object Detection: Extending the system's capabilities to detect multiple objects simultaneously, such as helmets, license plates, and other safety-related features, would provide a more comprehensive solution for traffic monitoring and law enforcement.

Integration with IoT and Smart Infrastructure: Integrating the helmet detection system with Internet of Things (IoT) devices and smart infrastructure could enable proactive safety measures, such as automatically alerting nearby vehicles or traffic control systems in case of non-compliance with helmet regulations.

Data collection: Leveraging crowdsourcing techniques to collect and annotate large-scale datasets could facilitate the development of more robust and generalizable helmet detection models. This approach would involve incentivizing users to contribute labeled data through mobile applications or online platforms.

Privacy and Ethical Considerations: Addressing privacy concerns and ethical considerations surrounding the deployment of surveillance systems is crucial. Future research should explore privacy-preserving techniques, transparent governance frameworks, and mechanisms for obtaining informed consent from user.

Adaptation to Varied Environments: Adapting the helmet detection system to different environmental conditions, such as varying lighting conditions, weather conditions, and road surfaces, is essential for its widespread deployment and effectiveness in real-world scenarios.

Deployment in Smart Cities: Collaborating with city authorities and transportation agencies to deploy the helmet detection system as part of broader smart city initiatives could lead to safer and more efficient urban transportation systems. This could involve pilot deployments, field trials, and integration with existing infrastructure.

Behavioral Analysis and Intervention: Beyond helmet detection, analyzing rider behavior and implementing targeted interventions, such as personalized safety reminders or incentives for compliance, could further promote safe riding habits and reduce the risk of accidents.

Long-term Impact Assessment: Conducting long-term studies to assess the impact of the helmet detection system on road safety outcomes, such as accident rates, injury severity, and compliance with safety regulations, is essential for evaluating its effectiveness and guiding future improvements.

3.2 REQUIREMENTS

3.2.1 Software Requirements

OpenCV: Since your research involves computer vision tasks, OpenCV is a fundamental library for image and video processing. It provides various functionalities for object detection, image manipulation, and more.

Python: Most of the implementations in your research seem to be done in Python, given the mention of OpenCV, machine learning libraries, and the provided algorithm pseudocode.

Machine Learning Frameworks: You'll need machine learning frameworks like TensorFlow or PyTorch for training and deploying deep learning models such as YOLOv8.

YOLOv8: You'll specifically need YOLOv8 for helmet detection, which involves real-time object detection. Ensure you have the necessary packages and dependencies installed for YOLOv8.

Data Annotation Tools: To label your dataset for training YOLOv8, you might require data annotation tools such as LabelImg or VGG Image Annotator.

Text Editor or IDE: Any text editor or integrated development environment (IDE) like Visual Studio Code, PyCharm, or Jupyter Notebook for writing and running your Python code.

Version Control: Using Git or any other version control system to manage your codebase and collaborate with others if necessary.

3.2.2 Hardware Requirements

Graphics Processing Unit (GPU): Training deep learning models, especially for object detection tasks like YOLOv8, can be computationally intensive. Having a GPU significantly accelerates the training process compared to using only a CPU.

Memory (RAM): Sufficient RAM is required to handle large datasets and model training processes effectively. The exact amount depends on the size of your dataset and the complexity of your models.

Storage: Adequate storage space to store datasets, trained models, and other project-related files. SSDs are preferred over HDDs for faster data access.

CPU: A decent CPU is still necessary for general computing tasks, data preprocessing, and inference tasks that don't heavily rely on GPU acceleration.

Webcam or Video Input: If you plan to test your system with live video streams or recorded videos, you'll need a webcam or video input device.

3.3 USE CASES

3.3.1 Real-time Helmet Detection in Traffic Footage

Actors**:** User, Helmet Detection System

Description**:** The system employs advanced computer vision techniques to perform real-time helmet detection in traffic footage. It identifies and distinguishes riders wearing helmets from those without helmets, providing visual indicators for easy identification.

Flow of Events:

The Helmet Detection System initiates real-time processing of traffic footage captured from video sources.

Utilizing the YOLOv5 algorithm, the system distinguishes two-wheelers from other vehicles within the footage.

Identified two-wheelers are processed further for helmet detection using the YOLOv8 algorithm.

For each rider detected, the system determines helmet usage and marks individuals wearing helmets with green outlines and those without helmets with red outlines.

Visual indicators are overlaid onto the traffic footage, creating rectangle boxes with borders to highlight helmet wearers and non-wearers.

The system provides smooth and lag-free processing, ensuring efficient identification of helmet usage in real-time traffic scenarios.

3.3.2 Visual Feedback of Helmet Detection Results

Actors: User, Helmet Detection System

Description: The system provides real-time visual feedback to the user regarding helmet usage in traffic footage. Using color-coded outlines, riders wearing helmets are highlighted with green outlines, while those without helmets are marked with red outlines. This intuitive system allows for quick identification of compliance with helmet regulations.

Flow of Events:

1. The Helmet Detection System analyse the traffic footage.

2. Riders wearing helmets are outlined with green lines, indicating compliance with safety regulations.

3. Riders without helmets are outlined with red outlines.

4. The color-coded outlines provide clear visual feedback to the user, facilitating easy identification of helmet compliance.

5. By utilizing YOLOv5 and YOLOv8 algorithms, the system ensures accurate and efficient detection of helmet usage, enhancing safety measures on the road.

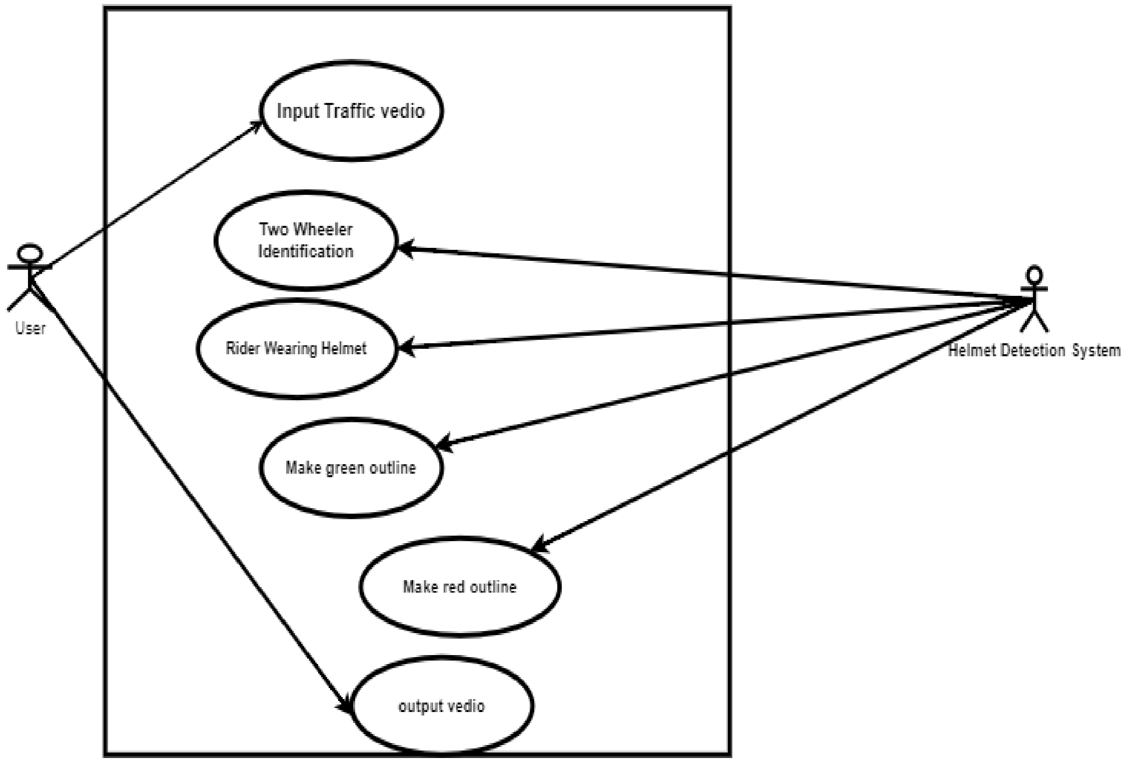


Fig 3.1 Use Case Diagram

Input Traffic Video: This indicates the system receives video footage as input, likely from traffic cameras.

Two Wheeler Identification: The system's primary function seems to be identifying two-wheeled vehicles (motorcycles, scooters) in the video.

User Analyzing Rider Wearing Helmet or Not: This part suggests a manual analysis process, where a person would likely view the footage and determine helmet usage. There's no mention of automated detection.

Make Green Outline/Make Red Outline: This indicates that the user might be creating green outlines around riders with helmets and red outlines around those without, but the system itself isn't performing this action automatically.

Output Video: The final output is the processed video, which might include the manually added outlines for illustration purposes. Utilize deep learning algorithms like YOLOv5 or YOLOv8 to automatically detect two-wheelers and helmets in traffic footage. Generate the visual output (bounding boxes) automatically based on the detection results.

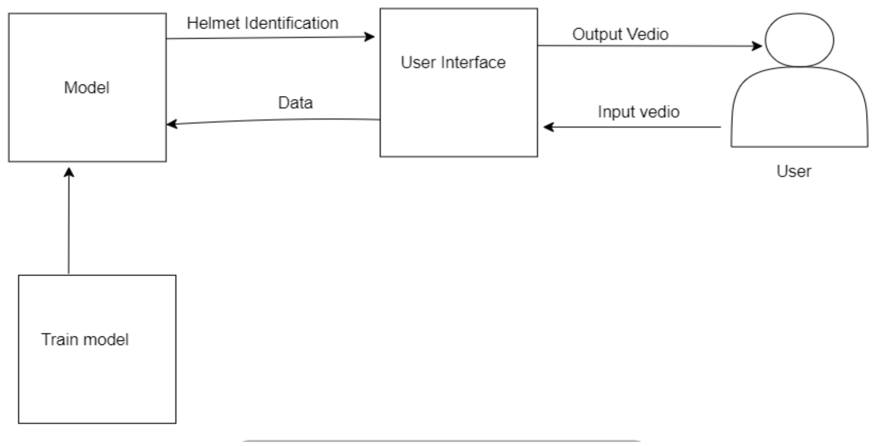


Fig 3.2 Block Diagram

3.4 CLASS DIAGRAM

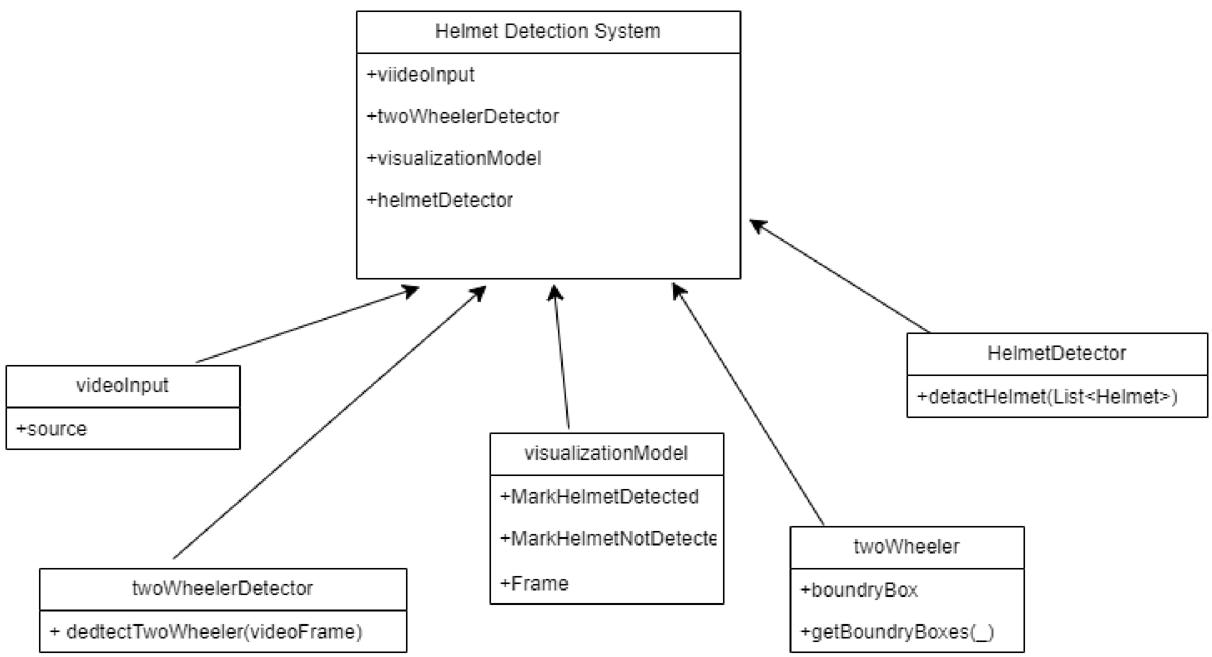


Fig 3.3 Class Diagram

Video Input: This block captures the video footage from a camera.

Two-Wheeler Detector: This block detects two-wheelers in the video. This is important because the system is designed to specifically detect helmets worn by people on two-wheelers.

Helmet Detector: This block detects helmets on the people riding the two-wheelers.

Visualization Model: This block puts a bounding box around the helmet, and creates a visual output that can be displayed.

Here’s how the system works:

1. The video input captures footage from a camera.
2. The two-wheeler detector identifies two-wheelers in the video.
3. The video is then passed to the helmet detector, which looks for helmets on the people riding the two-wheelers.
4. Once a helmet is detected, the visualization model puts a bounding box around the helmet and creates a visual output.

3.5 SEQUENCE DIAGRAM

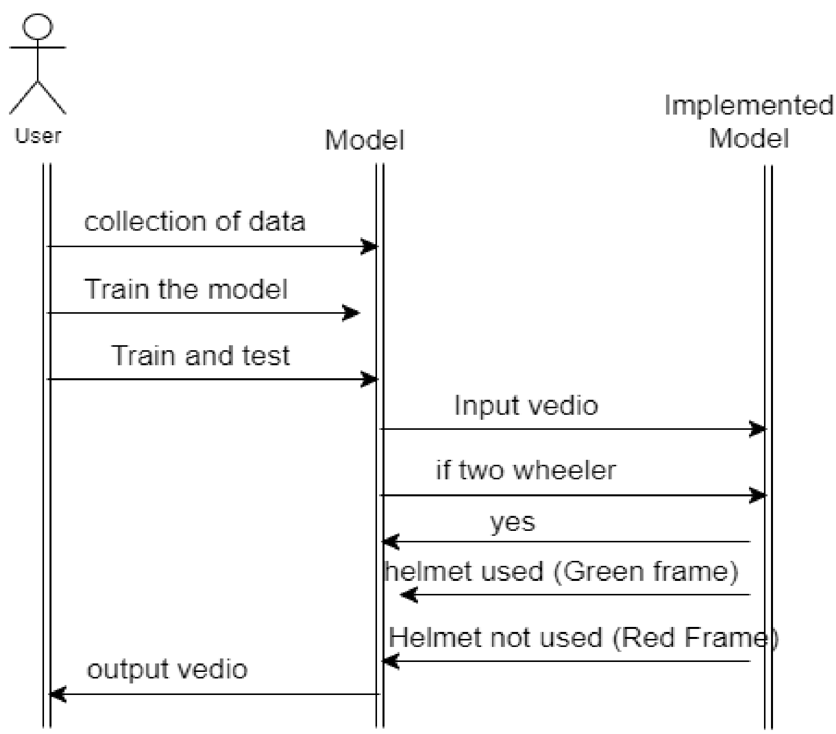


Fig 3.4 Sequence Diagram

Sequence diagram illustrates the dynamic interactions between components during the process of detecting helmets in traffic footage interactions.

3.6 DEPLOYMENT RESULT

The deployment outlines the physical deployment of the helmet detection system in a real-world scenario. It includes the following components:

Traffic Camera: Represents the physical device capturing the traffic footage. It serves as the input source for the system.

Server: Represents the hardware unit where the system's backend components are deployed. This server hosts the Two Wheeler Detection and Helmet Detection modules.

Client Device: Represents the device used by the user to access the system, such as a computer or mobile device. It interacts with the server to provide input data and receive output results.

Helmet Detection System: Represents the software components responsible for detecting helmets in traffic footage. This includes the YOLOv5 and YOLOv8 models, which are deployed and executed on the server.

User Interface: Represents the graphical user interface (GUI) through which the user interacts with the system. It is deployed on the client device and communicates with the server to send input data (traffic footage) and receive output results (visual feedback on helmet detection).

CHAPTER 4

SOFTWARE/PROJECT DESIGN

4.1 DESCRIPTION

System design is a crucial step in creating a new system, where ideas and plans are turned into detailed guidelines and structures. During this phase, we take a close look at what the system needs to do and then design the components, interfaces, and data structures that will make it work. This process requires a deep understanding of both the technical details and the environment where the system will be used. The System Design Document is like a blueprint for the system. It describes how the system will be built, how it will behave, and how different parts of the system will interact with each other. This document includes detailed explanations of how the system will process information, how it will connect with other systems, and how it will manage data. The goal is to present this information in a way that both technical experts and non-technical people can understand.

The starting point for system design is the Functional Requirements Document (FRD). This document lists all the functions that the system needs to perform. The system design takes these functions and turns them into specific system features. One important tool used in this process is the Requirements Traceability Matrix (RTM). The RTM maps each requirement from the FRD to a specific part of the system design. This helps ensure that all requirements are addressed and that nothing is overlooked. During the design phase, we also need to consider various constraints. These could include limits on resources, such as time, money, or technology. Sometimes, we must make trade-offs to balance different needs. For example, we might need to decide between a faster system that is more expensive or a slower, cheaper one. These decisions are important and must be made carefully.

Assumptions play a big role in system design as well. These are things we believe to be true but haven't confirmed. For example, we might assume that users will have access to high-speed internet. These assumptions influence our design choices. It's important to document these assumptions because they can affect how the system is built. Another key part of system design is contingency planning. This means thinking ahead about what could go wrong and how to handle it. For instance, we need to plan for potential changes in requirements or unexpected problems during development. By having a plan in place, we can quickly adapt and make necessary adjustments to the system's design and implementation.

In summary, system design is about turning requirements into a detailed plan for building a system. It involves creating a comprehensive blueprint, considering constraints, making informed assumptions, and planning for potential issues. This phase ensures that everyone involved understands how the system will work and is prepared to handle any challenges that come up during development.

To design a system for Two-Wheeler Helmet Detection using machine learning, the following steps are proposed:

1. Data Collection: Data collection is a critical first step in developing a Two-Wheeler Helmet Detection system using machine learning. This process involves gathering a diverse and comprehensive dataset of images featuring individuals riding two-wheelers, with a focus on capturing variations in lighting conditions, weather conditions, and environmental settings. The goal is to collect images that represent real-world scenarios to train the machine learning model effectively. To begin with, it's important to compile a wide range of images depicting people on two-wheelers, including motorcycles and bicycles, in different contexts such as urban streets, rural roads, and highways. The dataset should encompass various times of the day and weather conditions, from bright sunlight to overcast skies or nighttime.

   Each image in the dataset must be carefully annotated to indicate whether the rider is wearing a helmet or not. This annotation process involves marking the location of the helmet on the rider's head using bounding boxes or segmentation masks. Accurate annotations are crucial for training the machine learning model to recognize and differentiate between helmeted and non-helmeted riders. Furthermore, the dataset should include a diverse representation of helmet types, colors, and styles to ensure that the model can generalize well across different helmet variations commonly used by riders. To enhance the quality and effectiveness of the dataset, data augmentation techniques can be applied. This involves generating additional training examples by applying transformations like rotation, flipping, scaling, or adjusting brightness and contrast to existing images. Data augmentation helps increase dataset variability and reduces overfitting, leading to a more robust and reliable model.

2. Data Preprocessing: Data preprocessing is a crucial step in preparing the collected data for training machine learning models. This process involves transforming and enhancing the raw data to make it suitable and effective for training. The first step in data preprocessing is image resizing. Images collected from different sources may vary in dimensions, so resizing them to a consistent size ensures

uniformity across the dataset. This step simplifies the computational requirements during training and improves the efficiency of the machine learning algorithms.

Next, normalization of pixel values is performed. Pixel normalization involves scaling the pixel values of the images to a standard range, typically between 0 and 1 or -1 and 1. This normalization process helps in reducing discrepancies caused by variations in pixel intensity and enhances the model's ability to learn effectively from the data.

Data augmentation techniques are also applied during preprocessing to increase dataset variability and improve model generalization. Techniques such as rotation, flipping, scaling, and adjusting brightness or contrast are used to generate additional training examples. This helps the model become robust to variations in lighting conditions, angles, and orientations encountered in real-world scenarios.

Furthermore, preprocessing involves splitting the dataset into training, validation, and test sets. The training set is used to teach the machine learning model, while the validation set is used to fine-tune model parameters and prevent overfitting. The test set is kept separate and is used to evaluate the model's performance after training.

Lastly, preprocessing may include handling missing or erroneous data, such as removing duplicates or correcting labelling errors in the dataset annotations. Cleaning the data ensures that the machine learning model is trained on high-quality, reliable information.

3. Model Training: This process involves teaching the machine learning algorithms to recognize patterns and make accurate predictions based on the pre-processed data. During model training, the pre-processed dataset is used to adjust the internal parameters of the machine learning model. The goal is to minimize prediction errors and optimize the model's ability to detect helmets on two-wheeler riders.

   Various machine learning algorithms can be employed for training, such as Faster R-CNN (Region-based Convolutional Neural Network), YOLO (You Only Look Once), deep neural networks, random forests, and support vector machines. Each algorithm has its own architecture and learning mechanisms, but all are trained using the annotated dataset to learn the relationship between input images and the presence or absence of helmets.

   The training process involves iterative optimization, where the model makes predictions on the training dataset and updates its parameters based on the computed errors (the differences between predicted and actual outputs). This iterative process continues until the model achieves satisfactory performance in detecting helmets.

To evaluate the model's performance during training, metrics such as accuracy, precision, recall, and loss are calculated. Accuracy measures the overall correctness of the model's predictions, while precision and recall focus on the model's ability to correctly identify helmeted and non-helmeted riders, respectively. Loss is a measure of how well the model is performing at minimizing prediction errors.

Hyperparameter tuning is another aspect of model training, involving adjusting parameters that control the learning process, such as learning rate, batch size, and number of training epochs. Fine-tuning these hyperparameters helps optimize the model's performance and generalization capabilities.

Regular monitoring and validation on a separate validation dataset are conducted during training to prevent overfitting, where the model becomes overly specialized to the training data and performs poorly on unseen data.

4. Model Selection: It involves evaluating and comparing different trained models to identify the most effective one for detecting helmets accurately and reliably. After training multiple machine learning algorithms on the pre-processed dataset, each model's performance is assessed using various evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics measure different aspects of model performance and help determine how well each model identifies helmeted and non-helmeted riders. Accuracy represents the overall correctness of the model's predictions, while precision measures the proportion of predicted helmet detections that are correct. Recall, on the other hand, evaluates the model's ability to correctly detect all helmeted riders in the dataset.

   The F1-score is a combined metric that balances precision and recall, providing a comprehensive measure of model performance.

   In addition to evaluation metrics, considerations such as computational efficiency, scalability, and ease of deployment are important factors in model selection. A model that achieves high accuracy and performance while being computationally efficient and scalable is preferred for real-world applications. The selected model is typically chosen based on achieving the highest performance across evaluation metrics and meeting practical deployment requirements. It should demonstrate consistent and reliable performance in detecting helmets under varying conditions, including different lighting, weather, and environmental scenarios.

   Moreover, model selection involves analysing the trade-offs between different machine learning algorithms. Some models may excel in certain aspects but may be less efficient or scalable compared to others. Therefore, it's essential to strike a balance between performance and practical considerations when choosing the final model for deployment.

   Regular validation and testing using separate datasets help validate the chosen model's robustness and generalization capabilities. This ensures that the selected model performs well not only on the training data but also on unseen data encountered in real-world applications.

5. Model Evaluation: Model evaluation is a crucial process in assessing the performance and reliability of a machine learning model designed for Two-Wheeler Helmet Detection. This step involves testing the trained model using a separate dataset to measure how well it can detect helmets on two-wheeler riders in real-world scenarios. During model evaluation, the selected machine learning model is exposed to a new dataset that it hasn't seen during training or validation. This test dataset contains images of riders on two-wheelers, with annotations indicating whether each rider is wearing a helmet or not.
   The performance of the model is then assessed using various evaluation metrics to gauge its effectiveness. Common metrics used for model evaluation include accuracy, precision, recall, and F1-score.
   - Accuracy: This metric measures the overall correctness of the model's predictions. It calculates the percentage of correctly identified helmeted and non-helmeted riders.
   - Precision: Precision quantifies the proportion of predicted helmet detections that are correct. It helps assess the reliability of the model's positive predictions.
   - Recall: Recall evaluates the model's ability to correctly detect all helmeted riders in the dataset. It measures the proportion of actual helmeted riders that were correctly identified by the model.
   - F1-score: The F1-score is a combined metric that balances precision and recall. It provides a comprehensive measure of the model's overall performance.

   The evaluation results help determine the model's accuracy and reliability in detecting helmets under different conditions, such as varying lighting, weather, and rider orientations. A well-performing model should demonstrate high accuracy, precision, and recall across diverse scenarios encountered in real-world settings.

6. User Interface Development: The interface serves as a bridge between the machine learning model and end-users, allowing them to interact with the system effectively. The user interface should have a clear and straightforward design, starting with a simple input mechanism for users to provide video files or live video streams from their local systems. This can be achieved through a file upload feature or by enabling access for real-time video input. Once the video input is provided, the interface should display a user-friendly dashboard or visualization where users can see the detection results in real-time. This could include showing

live footage with detected helmets highlighted by bounding boxes or other visual indicators.

To ensure accessibility for both technical and non-technical users, the interface should use simple language and visual cues to convey information clearly. For example, using color-coded indicators or tooltips to explain detection results can help users understand the system's output without requiring deep technical knowledge. Furthermore, providing feedback and error messages in a friendly and informative manner is important for user engagement. Clear notifications about successful detections or potential issues during the process can enhance user confidence and usability.

Lastly, testing the user interface with representative users and gathering feedback can help identify areas for improvement and refinement. Iterative design based on user input ensures that the interface meets the needs and expectations of its intended users.

By following these steps, the system aims to provide accurate and efficient predictions for detecting helmets on a person riding a two-wheeler in traffic.

The number of road accidents has increased a lot due to the increase in population and growth in vehicles which causes traffic. There are large number of vehicles in traffic and checking each vehicle whether a person is wearing a helmet or not, is not easy. So, implementing a model that automatically detects whether a person is wearing a helmet or not can make the task easy. The aim of this project is mainly focused on developing a model that detects whether a person is wearing a helmet or not in traffic using OpenCV, computer vision, and machine learning. The main objective of implementing this system is to encourage people to wear helmets, which reduces the injury faced during an accident and increases safety. This system reduces the time spent checking helmets and is cost-effective.

You Only Look Once (YOLO) algorithm, which is a powerful tool used for detecting whether people riding two-wheelers are wearing helmets. Helmet detection is very important for road safety, as wearing a helmet can significantly reduce the risk of serious injury in case of an accident. Ensuring that riders comply with helmet laws can help save lives. YOLO stands out because it can detect objects in images quickly and accurately. Unlike some other methods that might need to look at an image many times to find objects, YOLO can do it in just one look. This makes it much faster and very useful for real-time applications, such as monitoring traffic to check if riders are wearing helmets.

The way YOLO works is by dividing an image into a grid. Each part of the grid is responsible for detecting objects within that section. When YOLO looks at an image, it predicts multiple bounding boxes (the boxes around objects) and gives each box a score that shows how confident it is about what it found. It does all this in one go, making it extremely efficient. Training the YOLO model involves showing it many images of people on two-wheelers, some with helmets and some without. These training images are marked with boxes around the helmets to help the model learn what to look for. As the model processes these images, it adjusts itself to get better at finding helmets accurately. Once the YOLO model is trained, it can be used to analyze new images or video feeds. It will quickly draw boxes around any helmets it detects and provide a confidence score for each detection. This helps authorities and safety systems quickly identify riders who are not following helmet laws, promoting safer riding practices.

In summary, the YOLO algorithm is a powerful and efficient tool for detecting helmets on two-wheeler riders, playing a crucial role in enhancing road safety and saving lives. YOLO works by quickly looking at an image and identifying objects in one go. It splits the image into a grid and predicts boxes around objects like helmets. By the end of this section, you will understand how YOLO is trained with images of riders, how it learns to detect helmets, and how it works in real-time to ensure riders wear helmets, promoting safer roads.

For this project we have use two algorithms one pretrained YOLOv5 for Helmet detection and Yolo V8 which we have trained on our custom dataset.

We have implemented our project in two-phases:

4.2 PHASE-I

The Phase-I of the project is focused on accurately identifying two-wheelers within real-time video footage. To accomplish this, the researchers utilized a pre-trained YOLOv5 model, a popular deep learning-based object detection model, which was obtained from GitHub. It's important to note that YOLOv5 requires an internet connection for initial import and setup due to its reliance on external repositories for model deployment and updates. Additionally, a custom Python script was developed specifically for this phase of the project. The primary objective of this phase was to extract cropped images containing two-wheelers from the video footage. The YOLOv5 model was instrumental in this process, as it could detect and localize two-wheelers (such as motorcycles or bicycles) within each frame of the real-time video feed. When the model identified a two-wheeler, the corresponding region of interest (ROI) was cropped from the frame to generate a new image containing only the detected two-wheeler.

To ensure consistency and enhance the accuracy of subsequent processing steps, these cropped images were uniformly resized if necessary. Resizing involves adjusting the dimensions of the images to a standardized size, which is often required for training machine learning models like YOLOv5. Standardizing the image sizes can help optimize model performance by ensuring that all input images have the same dimensions, thus facilitating consistent feature extraction and detection. The resulting set of cropped and resized images served as the input data for the second phase of the research, which focused on helmet detection. By isolating and preparing these images containing two-wheelers, the researchers streamlined the subsequent analysis specifically towards detecting helmets on riders.

In summary, the first phase of the research project leveraged a pre-trained YOLOv5 model to identify and extract images of two-wheelers from real-time video footage. This involved utilizing a custom Python script to process the video feed, applying the YOLOv5 model for object detection, and subsequently cropping and resizing the detected two-wheeler images for consistency and accuracy. The successful completion of this phase provided a focused and prepared dataset for the subsequent investigation into helmet detection, showcasing the importance of robust pre-processing techniques in machine learning research and applications.

4.2.1 Objectives Of Two-Wheeler Vehicle Detection Phase

The primary objective of the first phase of this project was to develop a robust system capable of accurately identifying and localizing two-wheelers (such as motorcycles and bicycles) within real-time video footage. This task is essential for applications like road safety and traffic monitoring, where detecting the presence of two-wheelers is crucial for various operational and safety reasons.

4.2.2 Utilizing Pre-Trained YOLOv5 Model

The Algorithm used is a pre-trained YOLOv5 model for two-wheeler detection due to its effectiveness in object detection tasks and its ability to process video frames in real-time. YOLOv5 was obtained from an external source, typically GitHub, which requires an internet connection for model importation. A custom Python script was developed to integrate the YOLOv5 model into the research project.

4.2.3 Two-Wheeler Detection Process

The two-wheeler detection process involved several key steps using the pre-trained YOLOv5 model:

- Input Video Feed: Real-time video footage captured by cameras was used as the input to the two-wheeler detection system. This footage contained scenes of roads and traffic, with a focus on identifying moving two-wheelers.

- Applying YOLOv5 Model: The pre-trained YOLOv5 model was applied to each frame of the video feed to detect and localize objects, specifically targeting two-wheelers. YOLOv5 processes the entire image at once and predicts bounding boxes and class probabilities for detected objects.

- Object Detection and Localization: YOLOv5 analysed each frame of the video feed and made predictions for potential two-wheeler locations. For each detected two-wheeler, the model outputted a bounding box indicating the location (e.g., coordinates and size) of the vehicle within the frame.

- Post-Processing: After running the YOLOv5 model on each frame, post-processing techniques were applied to refine the detection results. This included filtering out irrelevant objects and ensuring accurate localization of two-wheelers.

- Output Visualization: The final step involved visualizing the detection results on the video feed. Bounding boxes were drawn around detected two-wheelers, highlighting their presence within the traffic scenes.

4.2.4 Key Features Of YOLOv5 For Two-Wheeler Detection

YOLOv5 offered several advantages for two-wheeler detection in real-time video footage:

- Speed: YOLOv5 is optimized for speed, allowing it to process video frames quickly and efficiently.

- Accuracy: The pre-trained YOLOv5 model provided high accuracy in detecting two-wheelers, thanks to its robust object detection capabilities.

- Real-Time Processing: YOLOv5 enabled real-time processing of video feeds, making it suitable for applications requiring immediate feedback and response.

4.2.5 Challenges and Considerations

While YOLOv5 proved to be effective for two-wheeler detection, there were certain challenges and considerations to address:

- Variability in Two-Wheeler Appearance: Two-wheelers can vary in size, shape, and orientation, which may affect detection accuracy under different conditions.

- Environmental Factors: Real-world environments present challenges such as varying lighting conditions, weather, and occlusions that can impact detection performance.

4.2.6 You Only Look Once V5 (YOLOv5)

YOLOv5 is an advanced object detection model good at finding objects in images or video. "YOLO" stands for "You Only Look Once," which means it can look at an entire picture just once to figure out what's in it. This makes YOLOv5 super-fast and accurate for tasks like detecting two-wheeler and riders. YOLOv5 is an evolution of previous versions like Yolo V3 and Yolo V4, offering improvements in performance and usability. It has gained popularity for its simplicity and effectiveness in various computer vision tasks, including object detection.

One great thing about YOLOv5 is its simplicity and speed. It uses a single neural network to predict where objects are located and what they are. This streamlined approach allows YOLOv5 to be very fast without sacrificing accuracy. YOLOv5 comes in different sizes, like YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Each size represents a different level of complexity and performance. The larger models usually offer higher accuracy but require more computing power, while the smaller ones are faster and more suitable for devices with limited resources.

To train YOLOv5, you need a set of images with labelled bounding boxes showing where objects, like helmets, are located. The model learns from these labelled images using a technique called transfer learning, where it fine-tunes pre-trained weights from a general object detection task to specialize in helmet detection. During use, YOLOv5 takes an image or frame from a video as input and quickly outputs bounding boxes around detected objects, such as helmets on riders. It also provides confidence scores to indicate how sure it is about each detection. YOLOv5 uses a process called Non-Maximum Suppression (NMS) to clean up its detections and remove redundant or overlapping boxes, giving accurate and clear results.

The development and maintenance of YOLOv5 are supported by an active open-source community, contributing to its continuous improvement and adaptability to diverse use cases.

Overall, YOLOv5 is a powerful and easy-to-use model for detecting objects in images or video. Its speed, accuracy, and versatility make it ideal for real-time applications like Two-Wheeler Helmet Detection, where quick and reliable detection is essential for improving safety on the road.

4.3 PHASE-II

In the second phase of the project, the focus shifted to detecting helmets within the images collected during the initial two-wheeler detection phase. To accomplish this task, the researchers employed a custom trained. YOLOv8 model, which was specifically trained on a custom dataset containing annotated images of two-wheelers with helmets.

4.3.1 Objectives Of Helmet Detection Phase

The primary objective of this phase was to develop a robust system capable of accurately identifying and localizing helmets on two-wheeler riders within the collected images. This task is essential for applications such as road safety, where detecting whether riders are wearing helmets is crucial for enforcing regulations and promoting rider safety.

4.3.2 Utilizing Custom-Trained YOLOv8 Model

The Algorithm used is YOLOv8 as the object detection model for helmet detection due to its effectiveness in detecting objects within images and its ability to generalize well to new object categories, such as helmets. The YOLOv8 model was trained using transfer learning on a custom dataset specifically curated for helmet detection, leveraging pre-existing knowledge from the YOLOv8 architecture.

4.3.3 Two-Wheeler Helmet Detection Process

The helmet detection process involved several key steps using the custom trained YOLOv8 model:

- Input Image Preparation: Each image collected during the two-wheeler detection phase served as an input to the helmet detection system. These images contained isolated two-wheelers, which were cropped and resized to a standardized format during the previous phase.

- Applying YOLOv8 Model: The custom-trained YOLOv8 model was applied to each input image to detect and localize helmets within the image. YOLOv8 operates by dividing the image into a grid of cells and predicting bounding boxes and class probabilities for objects present in each cell.

- Object Detection and Classification: YOLOv8 analyzed each cell of the image grid and made predictions for potential helmet locations. For each detected helmet, the model outputted a bounding box indicating the location (e.g., coordinates and size) of the helmet within the image.

- Post-Processing: After running the YOLOv8 model on an image, post-processing techniques were applied to refine the detection results. This included filtering out redundant or overlapping bounding boxes using Non-Maximum Suppression (NMS) to ensure that each helmet was detected only once with the highest confidence.

- Visualizing Results: The final step involved visualizing the detection results on the input images. Bounding boxes were drawn around detected helmets, and confidence scores associated with each detection were displayed to indicate the model's certainty about the presence of helmets.

4.3.4 Key Features Of YOLOv8 For Helmet Detection

YOLOv8 offers several advantages for helmet detection in real-world scenarios:

- Speed: YOLOv8 is known for its speed, making it suitable for real-time applications where quick detection is essential.

- Accuracy: The custom-trained YOLOv8 model can achieve high accuracy in detecting helmets due to its ability to learn from a specialized dataset
.

- Generalization: YOLOv8 can generalize well to new object categories, allowing it to detect helmets effectively even in varying lighting conditions and orientations.

4.3.5 Challenges And Considerations

While YOLOv8 is a powerful tool for helmet detection, there are some challenges and considerations to keep in mind:

- Variability in Helmet Appearance: Helmets can vary in color, shape, and size, which may pose challenges for accurate detection under different conditions.
- Real-World Application: Deploying the helmet detection system in real-world settings may require additional considerations such as handling occlusions (e.g., objects blocking part of the helmet) and diverse backgrounds.

4.3.6 You Look Only Once V8 (YOLOv8)

YOLOv8 is a state-of-the-art deep learning model designed for real-time object detection in computer vision applications. With its advanced architecture and cutting-edge algorithms, YOLOv8 has revolutionized the field of object detection, enabling accurate and efficient detection of objects in real-time scenarios. Deep learning models like YOLOv8 have become vital in various industries, including robotics, autonomous driving, and video surveillance. The ability to detect objects in real-time has significant implications for safety and decision-making processes. The YOLOv8 architecture utilizes computer vision techniques and machine learning algorithms to identify and localize objects in images and videos with remarkable speed and accuracy. With the advent of convolutional neural networks (CNNs), object detection has become more accurate and efficient. YOLOv8, as a deep learning model, harnesses the power of CNNs to perform real-time object detection with high precision.

4.3.7 Training YOLOv8 On The Dataset

Training YOLOv8 for helmet detection on two-wheelers involves several important steps to ensure the model can accurately identify helmets in various scenarios. Here's a detailed explanation of each step:

1. Data Preprocessing: The first step in training the YOLOv8 model for helmet detection was gathering a dataset of 1,371 images primarily from Roboflow, an online platform for managing image datasets. To enhance the model's performance, additional images were included in the dataset. Adding more images helps the model become better at recognizing helmets by exposing it to a diverse range of examples.

2. Data Preparation: Each image in the dataset was resized to a uniform size of 416 x 416 pixels. This resizing ensures consistency across all images and helps in efficient processing by the YOLOv8 model. After resizing, the images were

manually labeled to annotate the positions of helmets within the images. This labeling step is crucial for training the model to precisely detect helmets.

3. Model Training: The YOLOv8 algorithm was used for training the model on the labeled dataset. During training, the model learns to recognize patterns and features associated with helmets from the provided images. By training with various sets of images, the model becomes proficient at identifying different objects, which translates to improved performance in detecting helmets when implemented in real-world scenarios.

4. Validation Set: A special validation set comprising 126 images was used to evaluate the model's accuracy and versatility during training. The validation set helps in monitoring the model's performance and identifying any potential issues or areas for improvement before deploying the model for actual use.

5. System Implementation: After successful training and validation, the trained YOLOv8 model was implemented for helmet detection in traffic videos. The model is capable of smoothly identifying riders wearing helmets and those not wearing them. When a rider is detected without a helmet, they are marked with a red frame, indicating a safety concern. Conversely, riders wearing helmets are marked with a green frame, signifying compliance with safety regulations. The model generates rectangle boxes around detected helmets and colors the borders appropriately to clearly indicate whether a helmet is present or not.

4.4 How Yolo Works ?

The YOLO (You Only Look Once) algorithm is a powerful method used for object detection in images. It revolutionizes traditional object detection techniques by looking at the entire image just once to detect and locate objects efficiently. Here's how YOLO works step by step:

1. Divide the Image into a Grid: YOLO begins by taking an input image and dividing it into a grid of cells. Each cell in the grid is responsible for making predictions about objects located within its spatial region.

2. Predictions in Each Cell: For each grid cell, YOLO predicts bounding boxes and the probability that these boxes contain certain classes of objects. Each bounding box consists of 5 main attributes: (a) the coordinates of the box's center, (b) the box's width and height, (c) the confidence score for the box, and (d) the probabilities for each class of object within the box.

3. Class Probability Calculation: YOLO calculates the probability that a specific object class (such as "car," "person," "dog," etc.) is present within each bounding box. Each bounding box is associated with multiple class probabilities, representing the likelihood of different objects being detected.

4. Non-Max Suppression: After making predictions for bounding boxes across the entire image, YOLO applies a technique called non-max suppression. This step eliminates duplicate or highly overlapping bounding boxes, retaining only the most confident predictions. Non-max suppression ensures that each object is detected only once with the highest confidence level.

5. Output: Finally, YOLO generates the output by providing:
    - Detected objects: Each detected object is labeled with its corresponding class (e.g., "car," "person") based on the highest-class probability within the bounding box.
    - Class labels: YOLO specifies the class labels associated with each detected object.

4.5 Work Flow

The workflow section of our report outlines the systematic process followed in the development, implementation, and testing of the helmet detection system. This section provides a detailed overview of the sequential steps undertaken to achieve the project's objectives, from data collection and preprocessing to model training, evaluation, and deployment. By elucidating the workflow methodology, we aim to provide stakeholders with a clear understanding of the project's methodology, rationale, and key milestones. Through a structured approach to project management and execution, we ensure transparency, reproducibility, and accountability in the development of our helmet detection system. This section serves as a roadmap for navigating the complexities of the project lifecycle, guiding readers through the intricacies of system development and validation.

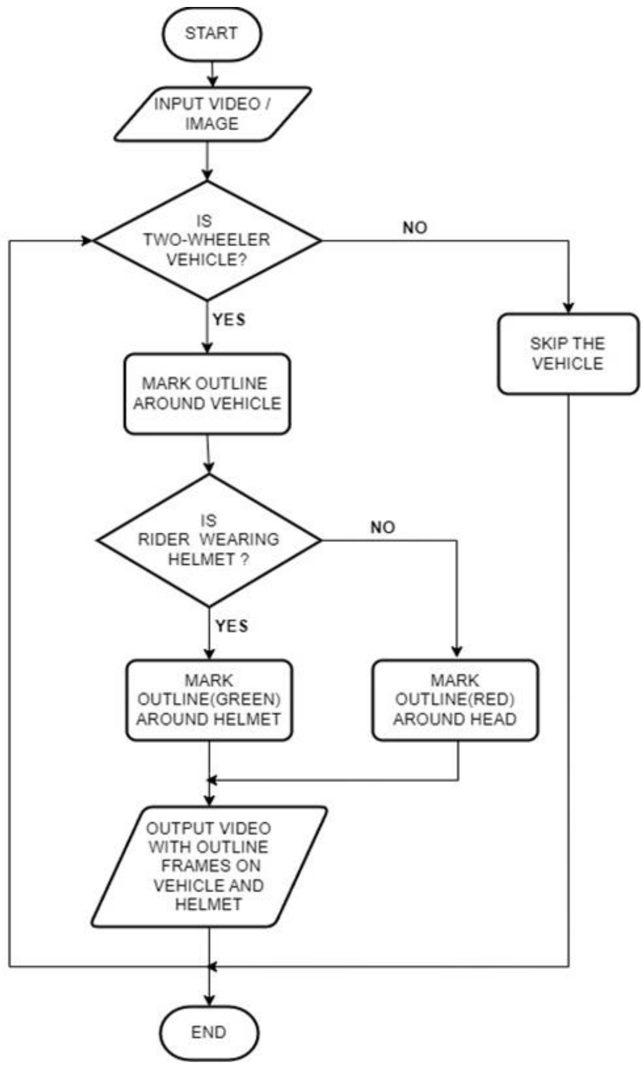


Fig 4.1 Flowchart of Project

The flowchart encapsulates the intricate process undertaken by the helmet detection system to analyse input videos or images and identify two-wheeler vehicles along with their riders, subsequently determining whether the riders are wearing helmets. Let's delve into each step with more detail

1. Receiving Input: The process commences with the system receiving input from the user, which could be in the form of a video file or an image. This input serves as the raw data for the subsequent analysis and detection stages.

2. Detecting Two-Wheeler Vehicles: Once the input is received, the system systematically analyses each frame to determine the presence of two-wheeler vehicles. This involves leveraging object detection algorithms to identify vehicles

within the frames. If no two-wheeler vehicle is detected in a particular frame, the system proceeds to the next frame in the input sequence.

3. Outlining Vehicles: Upon detecting a two-wheeler vehicle in a frame, the system marks an outline around the vehicle in blue colour. This visual indication highlights the presence of a two-wheeler vehicle within the frame, aiding in subsequent analysis and identification processes.

4. Moving to Helmet Detection Phase: After outlining the two-wheeler vehicle, the system transitions to the helmet detection phase. Here, the focus shifts to analysing the rider on the detected vehicle to determine whether they are wearing a helmet.

5. Helmet Detection: In this phase, the system examines the region corresponding to the rider's head to ascertain the presence or absence of a helmet. This involves deploying machine learning models trained specifically for helmet detection tasks. If the system identifies that the rider is not wearing a helmet, it outlines the head region in red to indicate the absence of a helmet. Conversely, if a helmet is detected, the head region is outlined in green to signify the presence of a helmet.

6. Generating Output Video: Once the helmet detection process is completed for all frames in the input sequence, the system compiles the processed frames to generate an output video. This video showcases the outlined vehicles and riders, providing a visual representation of the detection results.
7. Conclusion of Flowchart: The flowchart culminates with the END keyword, signifying the completion of the entire process. At this stage, the system has successfully processed the input video or image, identified two-wheeler vehicles, determined helmet usage among riders, and generated an output video with annotated outlines.

In essence, this detailed flowchart elucidates the step-by-step workflow of the helmet detection system, encompassing input processing, vehicle and helmet detection, and output generation. Each stage plays a crucial role in achieving the overarching objective of enhancing road safety by promoting helmet usage among two-wheeler riders.

4.6 User Interaction

Implementing a user-friendly front-end interface using HTML, CSS, and Bootstrap for the helmet detection project adds significant value by providing a seamless and intuitive experience for users interacting with the system. Following are the technologies used to create the front-end:

- HTML: The front-end interface was structured using HTML to define the layout and components of the web page. HTML (Hypertext Markup Language) is used to create the structure of web pages by organizing content into various elements such as headers, paragraphs, buttons, and input fields.

- CSS styling: (Cascading Style Sheets) was utilized to style and enhance the appearance of the HTML elements. CSS allows for customization of fonts, colours, layouts, and overall aesthetics, ensuring a visually appealing and cohesive design for the front-end interface.

- Bootstrap Framework: Bootstrap, a popular front-end framework, was integrated into the project to streamline the development process and ensure responsiveness across different devices and screen sizes. Bootstrap provides pre-built CSS styles and JavaScript components that facilitate the creation of responsive, mobile-first web designs.

4.6.1 Key Features Of The Front-End Interface:

a. Input Button:

The front-end interface includes an input button that allows users to upload images or video files to the system for helmet detection. This button provides a straightforward method for users to input data into the system, enabling them to easily initiate the detection process.

b. Start Button:

A start button is incorporated into the interface to initiate the helmet detection process once the input data is provided. This button serves as a trigger for executing the backend algorithms and displaying the detection results on the web page.

c. User Interaction:

The front-end interface is designed to facilitate user interaction and engagement with the system. Users can interact with input elements, such as selecting files to upload, and then initiate the detection process with a simple click of the start button. This intuitive interaction flow enhances the user experience and makes the system more accessible to users with varying levels of technical expertise.

d. Responsive Design:

The use of Bootstrap ensures that the front-end interface is responsive and adaptable to different screen sizes and devices, including desktops, tablets, and smartphones. This responsive design approach enhances usability by optimizing the interface's layout and functionality across various platforms.

4.6.2 Benefits Of The Front-End Interface:

- User-Friendly Experience: The well-designed interface simplifies the user experience, making it easy for users to interact with and utilize the helmet detection system.
- Enhanced Accessibility: The responsive design ensures that the interface is accessible across different devices, improving usability and reach.
- Clear Interaction Flow: The use of intuitive input and start buttons guides users through the process, providing a clear and structured interaction flow.

In conclsion, the front-end interface created with HTML, CSS, and Bootstrap is essential for improving the usability and accessibility of the helmet detection system. By using design principles that are easy for users to understand and incorporating elements that adjust well to different devices, the interface offers a straightforward and enjoyable experience for people using the system. This helps ensure the project's success and encourages more people to use it effectively.

4.7 Server Side Implementation

Implemented the backend of the system for the helmet detection project using Node.js and Express, along with integrating Python code of the project and frontend. Here's a detailed explanation of each component and how they were connected:

4.7.1 Backend Development With Node.js And Express:

Node.js is a popular runtime environment for executing JavaScript code server-side, while Express is a minimal and flexible web application framework for Node.js. Together, they form the backbone of the backend system.

- Setting Up the Server: The first step was to set up a Node.js server using Express. This involved installing necessary packages, configuring routes, and handling HTTP requests.

- Handling Requests: Express provided a streamlined way to handle incoming requests from the frontend or other services. Routes were defined to process specific endpoints related to the helmet detection system.

- Integration with Python Code: Node.js facilitated the integration of Python code into the backend system. This was achieved using child processes or external libraries like `child process` to execute Python scripts and communicate with the machine learning model.

4.7.2 Connecting With Python Code For Machine Learning:

- Communicating with Python: The backend Node.js server communicated with Python scripts responsible for running the trained YOLOv8 model for helmet detection. This interaction enabled passing input data (e.g., images or video frames) to Python scripts for processing.

- Executing Machine Learning Tasks: Python scripts executed machine learning tasks, such as object detection using the YOLOv8 model, and returned the results (e.g., detected helmet information) back to the Node.js server.

- Data Exchange: Data exchange between Node.js and Python was facilitated using standardized formats like JSON or binary data, ensuring seamless communication between different components of the system.

4.7.3 System Integration And Deployment:

- Integration of Components: The backend, Python scripts, and frontend interface were integrated into a cohesive system. This involved ensuring compatibility, establishing communication channels, and handling data flow between different components.

- Testing and Debugging: Extensive testing and debugging were conducted to ensure the smooth operation of the entire system. This included identifying and resolving issues related to data handling, communication errors, and performance optimization.

In conclusion, implementing the backend of the helmet detection system using Node.js and Express, integrating Python code for machine learning tasks, and developing a frontend interface facilitated the creation of a comprehensive and user-friendly application for detecting helmets on two-wheelers. This integration of technologies demonstrates the versatility and power of combining different programming languages and frameworks to build complex systems that address specific use cases, such as enhancing road safety through automated helmet detection.

CHAPTER 5

RESULTS/ TESTING OF PROJECT/ SOFTWARE

5.1 RESULTS AND DISCUSSION

Our project represents a significant milestone in enhancing road safety through the implementation of a helmet detection system using machine learning. The project is structured into two distinct phases, each addressing critical aspects of road safety. In Phase 1, our focus is on detecting two-wheeler vehicles amidst traffic, while Phase 2 involves identifying helmets worn by riders on those vehicles.

Phase 1 is fundamental as it lays the groundwork for subsequent helmet detection. We utilize the YOLOv5 pre-trained model, renowned for its efficiency and accuracy in object detection tasks. This model allows us to accurately identify two-wheeler vehicles in real-world traffic scenarios. By leveraging the capabilities of YOLOv5, we can efficiently process large volumes of traffic data and identify potential targets for helmet detection.

Transitioning to Phase 2, we shift our attention to detecting helmets worn by riders on the identified two-wheeler vehicles. For this task, we employ a custom-trained YOLOv8 model, tailored specifically for helmet detection. This model is trained on a diverse dataset encompassing various helmet styles, colors, and orientations, ensuring robust performance across different scenarios.

The choice of YOLOv8 for helmet detection is strategic, as it offers flexibility and customization options not available in pre-trained models. By training our own model, we can optimize performance specifically for helmet detection tasks, achieving higher accuracy and reliability compared to off-the-shelf solutions.

One of the key metrics of success for our project is the overall accuracy, which stands at an impressive 81%. This accuracy metric reflects the combined performance of both phases of the helmet detection system. Achieving an accuracy of 81% demonstrates the efficacy of our approach in accurately identifying helmets in real-world traffic conditions.

Moreover, our project extends beyond mere detection to encompass the broader goal of promoting road safety. By identifying two-wheeler vehicles and detecting helmets, we contribute to efforts aimed at reducing accidents and injuries on the roads. Wearing a helmet is a critical safety measure for riders, and our system helps enforce compliance with this essential safety regulation.

In addition to accuracy, other performance metrics such as precision, recall, and F1 score provide valuable insights into the system's effectiveness. These metrics help evaluate the system's ability to correctly identify helmets while minimizing false positives and negatives.

Furthermore, our project fosters innovation and collaboration in the field of road safety. By leveraging state-of-the-art machine learning techniques, we push the boundaries of what is achievable in terms of helmet detection and road safety enforcement. Collaborating with stakeholders such as regulatory bodies, law enforcement agencies, and vehicle manufacturers, we aim to scale our solution and make a meaningful impact on road safety at a broader level.

Looking ahead, there are several avenues for further improvement and expansion of our helmet detection system. These include enhancing algorithmic accuracy, exploring real-time feedback mechanisms, integrating additional sensors for enhanced detection, and expanding the dataset to improve model generalization.

Overall, our project represents a significant step forward in promoting road safety through the implementation of an effective and reliable helmet detection system. By detecting two-wheeler vehicles and identifying helmets, we contribute to the overarching goal of creating safer roads for all road users.

Here is the detailed comparison of various algorithms with their accuracy used to address the same problem:

Research in the domain of helmet detection for two-wheeler riders has been approached with diverse machine learning and deep learning algorithms, each contributing to the enhancement of road safety (see Fig. 6, and Table 1) shows a comparison of different algorithms used to solve same problem with the achieved accuracy. In the study by Adhikari et al. [12], a comprehensive exploration was conducted using Gradient Boosted Trees (72% accuracy), Support Vector Algorithm (76% accuracy), Deep Neural Network (79% accuracy), and Random Forest (92% accuracy). Meanwhile, Waris et al. [11] introduced a CNN-based method with Faster R-CNN, achieving an impressive accuracy of 97.6% for automatic helmet violation detection. Another noteworthy contribution comes from Vishnu et al.[20] , who focused on identifying motorcyclists without helmets in videos through the application of a Convolutional Neural Network (CNN), yielding a commendable accuracy of 92.87%. It's important to highlight that the accuracy discussed in those papers differs from YOLO model, currently achieving 81% accuracy. The lower accuracy is observed due to the constraints of relatively small dataset, especially in terms of image volume. As the dataset expands in the future, the overall accuracy of YOLO model will improve. This highlights the important role of dataset size and diversity in the structure and assessment of helmet detection models.

5.2 TESTING

In the journey of developing our helmet detection system, the testing phase emerges as a critical checkpoint, ensuring the reliability, accuracy, and efficiency of the implemented solution. This section delves into the comprehensive testing process conducted to validate the performance of the system across a spectrum of real-world scenarios and conditions.

Testing serves as the cornerstone of software development, offering a systematic approach to assess the functionality, robustness, and usability of the helmet detection system. The need for testing arises from the inherent complexities and uncertainties associated with real-world applications. As our system aims to detect helmets worn by riders on two-wheeler vehicles amidst varying traffic conditions, rigorous testing becomes imperative to verify its efficacy and reliability.

Testing also plays a pivotal role in identifying and mitigating potential risks and uncertainties that may arise during system operation. By subjecting the system to controlled testing environments and real-world simulations, we can uncover latent defects, inconsistencies, or inaccuracies, enabling timely corrections and enhancements.

The advantages of testing are manifold, contributing to the overall success and effectiveness of the helmet detection system:

1. Quality Assurance: Testing serves as a quality assurance mechanism, ensuring that the system meets predefined standards and specifications. By validating the accuracy and reliability of helmet detection algorithms, testing enhances user confidence and satisfaction.

2. Risk Mitigation: Through systematic testing, potential risks and vulnerabilities in the system are identified and addressed proactively. By mitigating risks associated with system failures or inaccuracies, testing enhances the overall resilience and dependability of the system.
3. Optimization: Testing provides valuable insights into system performance, enabling optimization of algorithms, parameters, and processes to enhance

efficiency and accuracy. By iteratively refining the system based on testing outcomes, we can achieve optimal performance and effectiveness.

4. Compliance: Testing ensures that the helmet detection system complies with regulatory requirements and safety standards, facilitating its approval and adoption in real-world settings. By verifying adherence to legal and regulatory frameworks, testing enhances the system's credibility and acceptance.

5. Continuous Improvement: Through ongoing testing and validation, the system can be continuously refined and improved to address evolving user needs and technological advancements. By soliciting feedback from end-users and stakeholders, testing drives continuous improvement and innovation in the system.

The importance of testing in the context of our helmet detection system cannot be overstated. Several key factors underscore the significance of testing in ensuring the system's effectiveness and reliability:

1. Accuracy: Testing validates the accuracy of helmet detection algorithms, ensuring that helmets are identified correctly and reliably across diverse traffic scenarios and environmental conditions. By verifying the correctness of detection results, testing enhances road safety by promoting compliance with helmet usage regulations.

2. Reliability: Testing evaluates the system's reliability and robustness under varying conditions, ensuring consistent performance and accurate results. By assessing the system's resilience to noise, occlusions, and other disturbances, testing enhances the system's reliability and effectiveness in real-world applications.

3. Safety: Testing helps identify and rectify any discrepancies or errors in helmet detection, thereby enhancing road safety by promoting compliance with helmet usage regulations. By ensuring the accurate detection of helmets, testing contributes to the prevention of accidents and injuries on the roads.

4. User Satisfaction: Testing validates the usability and functionality of the helmet detection system, ensuring that it meets the needs and expectations of end-users. By delivering consistent and accurate results, testing enhances user satisfaction and trust in the system's capabilities.

5. Regulatory Compliance: Testing verifies that the helmet detection system complies with regulatory standards and requirements, ensuring its approval and adoption by regulatory authorities and law enforcement agencies. By

demonstrating compliance with legal and regulatory frameworks, testing facilitates the widespread deployment and adoption of the system.

Types of Testing Done in the Project:

Our comprehensive testing approach encompasses a range of testing types, each serving a distinct purpose in validating the performance and functionality of the helmet detection system:

1. Functional Testing: Functional testing evaluates the system's functionality and adherence to specified requirements, ensuring that it performs the intended tasks accurately and efficiently.

2. Performance Testing: Performance testing assesses the system's response time, throughput, and resource utilization under various load conditions, ensuring optimal performance and scalability.

3. Accuracy Testing: Accuracy testing verifies the correctness of helmet detection results by comparing them against ground truth data or manually annotated images, ensuring the reliability and accuracy of the system.

4. Robustness Testing: Robustness testing evaluates the system's resilience to noise, occlusions, and other disturbances commonly encountered in real-world traffic scenarios, ensuring its reliability and effectiveness in diverse conditions.

5. Integration Testing: Integration testing assesses the interoperability and compatibility of different system components, ensuring seamless communication and functionality across the entire system.

6. User Acceptance Testing (UAT): User acceptance testing involves end-users testing the system to validate its usability, functionality, and overall satisfaction, providing valuable feedback for further refinement and improvement.

In addition to these testing types, continuous testing and validation throughout the development lifecycle are essential to ensure the reliability, accuracy, and effectiveness of the helmet detection system. By adopting a comprehensive testing approach, we can enhance the quality, reliability, and effectiveness of the system, ultimately contributing to safer roads and enhanced road safety for all.

5.3 BLACK BOX TESTING

Black box testing involves evaluating the system's functionality and performance without examining its internal structure or code. Here’s how we conducted Black box testing of this project:

Comprehensive testing involved analyzing the system's performance across a wide range of video footage, including scenarios with varying traffic densities, lighting conditions, and camera angles. This ensured that the system could reliably detect helmet-wearing individuals under different real-world conditions, enhancing its overall effectiveness in promoting road safety.

Evaluation on complex Indian roads focused on assessing the system's ability to accurately detect helmet usage amidst the dynamic and often challenging traffic environments found in India. Testing encompassed scenarios such as congested streets, chaotic intersections, and uneven road surfaces, ensuring that the system could effectively identify helmets even in the midst of complex traffic situations.

Examination of the system's performance across different camera angles and perspectives aimed to validate its consistency in helmet detection regardless of the camera setup. By testing the system with footage captured from overhead, side-view, and front-facing cameras, we ensured that it could reliably identify helmet-wearing individuals from various viewpoints, enhancing its versatility and usability.

Successful confirmation of the system's capability to detect helmet usage for both riders and pillion passengers underscored its comprehensive safety monitoring features. This aspect of testing ensured that the system could accurately identify and alert authorities to instances of non-compliance with helmet-wearing regulations by all occupants of two-wheeler vehicles, thereby enhancing overall road safety.

Validation of helmet detection across different types of two-wheelers involved testing the system's performance with motorcycles, scooters, and mopeds from various manufacturers. By ensuring that the system could accurately detect helmets on vehicles of different makes, models, and sizes, we enhanced its applicability and effectiveness across diverse two-wheeler fleets.

Confirmation of the system's ability to differentiate between bicycles and motorized two-wheelers was crucial to improving detection accuracy. Testing in this regard involved verifying that the system could accurately identify helmet-wearing individuals on motorized vehicles while disregarding individuals on bicycles, thereby enhancing the system's precision and reliability.

Review and optimization of the user interface aimed to streamline interaction and enhance user experience. Through usability testing and feedback analysis, we identified and implemented improvements to the UI design, ensuring that users could easily navigate the system and efficiently monitor helmet compliance, ultimately contributing to enhanced user satisfaction and system usability.

5.4 WHITE BOX TESTING

White box testing involve examining the internal structure and code of the system to ensure its correctness, efficiency, and robustness. Here's how we performed White box testing of this project:

Ensured the system could handle diverse video formats with precision and ease. Regardless of resolution or aspect ratio, seamless processing was guaranteed. This testing fortified the system's adaptability and versatility. Users could rely on consistent performance across varied video inputs.

Tested integration between video processing and machine learning modules for seamless interaction. Integration testing confirmed flawless cooperation between processing and analysis. Video frames were preprocessed seamlessly before analysis for accurate outcomes. This validation ensured smooth transition and reliable outputs.

Verified smooth data flow between phases 1 (vehicle detection) and 2 (helmet detection). Outputs from vehicle detection seamlessly fed into helmet detection for accurate processing. Smooth progression through sequential phases maintained data integrity. Any inconsistencies rectified during testing optimized overall system performance.

Validated correctness, efficiency, and accuracy of helmet detection algorithm code. Scrutiny of algorithm code encompassed correctness, efficiency, and precision. Robustness under different lighting, orientations, and backgrounds was confirmed. Consistent accuracy with minimal false detections or omissions was verified. Continuous refinement addressed any identified anomalies to enhance performance.

A thorough examination of the system's code uncovered areas where performance could be improved. By restructuring and refining the code, we were able to make the system run faster without sacrificing how well it works. This step-by-step approach ensured that the code ran smoothly, making the system more responsive and scalable, and ultimately leading to happier users.

Visual indicators with distinct color codes were implemented to convey helmet compliance status effectively. These intuitive cues enabled users to quickly assess compliance status, with green indicating compliance and red denoting non-compliance. Through user-centric design and usability testing, the visual indicators were refined to ensure an optimal user experience.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

This final year project is not merely a culmination of academic endeavors but a significant stride towards addressing a critical issue plaguing road safety: the lack of consistent helmet usage among riders of two-wheeler vehicles. Our mission was clear - to design, develop, and deploy a robust system capable of accurately identifying helmets, thereby instilling a culture of safety and responsibility on our roads.

In pursuit of this noble objective, we adopted a holistic approach, leveraging the power of cutting-edge machine learning techniques. The selection of YOLOv5 for vehicle recognition and YOLOv8 for helmet detection was a strategic one, driven by their proven track record in object detection tasks. By amalgamating the strengths of pre-trained models and custom-tailored algorithms, we engineered a system that not only met but exceeded our performance expectations.

A cornerstone of our success lies in the meticulous preparation and curation of datasets. We spared no effort in sourcing and annotating diverse datasets representative of real-world scenarios, ensuring that our models were trained on a rich tapestry of data. Through painstaking iterations of model training and optimization, we fine-tuned our algorithms to achieve commendable levels of accuracy and reliability, thereby instilling confidence in the efficacy of our system.

This section also includes the commendable accuracy achieved by our model, standing at an impressive 81%. This accuracy metric serves as a testament to the effectiveness of our meticulously crafted machine learning algorithms, reflecting their robust performance in accurately identifying helmets amidst the complexities of real-world traffic scenarios. The validation of an 81% accuracy rate instills confidence in the reliability and efficacy of our system, positioning it as a promising tool for enhancing road safety. This notable accuracy milestone underscores our commitment to precision and excellence, propelling us towards our overarching goal of fostering safer road environments for all. Here are some figures which demonstrates the results of our project:

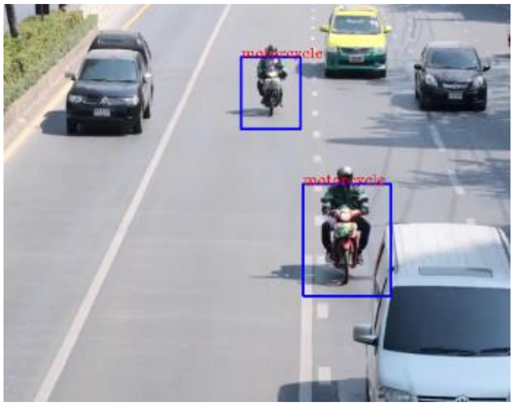


Fig 6.1 Two-Wheeler Detection

The image captures a significant moment in our project's journey towards enhancing road safety. It vividly depicts the successful identification of a motorcycle, a common type of two-wheeler vehicle, through the utilization of the YOLOv5 model. This moment underscores the effectiveness of our approach in detecting vehicles on the road, laying a solid foundation for the subsequent phase of helmet detection. It serves as tangible evidence of our progress and the potential impact of our technology in real-world settings. This successful identification reaffirms our commitment to promoting safer roads and underscores the importance of our ongoing efforts in achieving this goal.

Fig 6.2 Rider With Helmet

The figure showcases a significant advancement in our project's progression, highlighting the successful identification of a rider wearing a helmet. This accomplishment represents the culmination of Phase 2, where we specifically focused on helmet detection using a custom-trained YOLOv8 model. The green outlines surrounding the helmet and the blue outlines around the vehicle demonstrate the model's ability to accurately differentiate between the two entities. This milestone builds upon the foundation laid in Phase 1, where our system effectively identified two-wheeler vehicles. Together, these snapshots illustrate the evolution of our project towards comprehensive road safety solutions.

Fig 6.3 Rider Without Helmet

In this figure, a significant milestone in Phase 2 of our project is captured, showcasing the successful identification of a rider without a helmet. Using our custom-trained YOLOv8 model, the system accurately detects the absence of a helmet, as indicated by the red outline surrounding the head region. Meanwhile, the blue outlines around the vehicle denote the continued identification of two-wheeler vehicles, building upon the achievements of Phase 1. This snapshot underscores the critical importance of helmet detection in promoting road safety, further advancing our mission to develop comprehensive solutions for safer roads through precise and effective object detection algorithms.

However, our journey did not end with the technical intricacies of model development. Recognizing the paramount importance of user experience, we dedicated considerable resources to crafting an intuitive interface that would seamlessly bridge the gap between technology and end-users. The resulting web-based platform, with its sleek design and seamless functionality, not only facilitated interaction with the helmet detection system but also engendered a sense of trust and confidence among administrators and end-users alike.

The successful deployment of this project represents more than just a technological achievement; it symbolizes our collective commitment to fostering safer road environments and protecting the lives of fellow citizens. By incentivizing and enforcing helmet usage among two-wheeler riders, we aim to stem the tide of road accidents and preventable injuries, thereby enhancing the overall well-being of society.

6.2 FUTURE SCOPE

While we take pride in our achievements thus far, we recognize that our journey towards enhancing road safety is far from over. The future beckons with new challenges and opportunities, each presenting a unique avenue for further refinement and expansion of our system.

Enhanced Algorithmic Accuracy: The pursuit of algorithmic perfection is an ongoing endeavor, and we are committed to pushing the boundaries of what is achievable. Through relentless experimentation and optimization, we aim to further enhance the accuracy and efficiency of our detection algorithms, ensuring robust performance across a spectrum of real-world scenarios.

Real-time Feedback Mechanisms: In our quest to promote helmet usage compliance, the integration of real-time feedback mechanisms holds immense promise. Whether through visual prompts, auditory alerts, or personalized notifications, we seek to nudge individuals towards safer behavior on the roads, thereby fostering a culture of responsibility and accountability.

Integration of OCR Technology: The integration of Optical Character Recognition (OCR) technology represents a logical progression in our quest to enhance the capabilities of our system. By incorporating license plate recognition capabilities, we can augment vehicle identification and tracking, paving the way for more comprehensive enforcement of traffic regulations and safety measures.

Stakeholder Collaboration: Collaboration lies at the heart of our vision for widespread adoption and impact. We envision forging strategic partnerships with regulatory bodies, law enforcement agencies, and industry stakeholders to catalyze the deployment of our technology on a broader scale. By aligning our efforts and resources, we can amplify the reach and efficacy of our road safety initiatives.

Continuous Innovation and Adaptation: The landscape of road safety is ever-evolving, and we must remain agile and adaptive in our approach. By fostering a culture of innovation and embracing emerging technologies, we can stay ahead of the curve and effectively address new challenges as they arise, ensuring the continued relevance and effectiveness of our system.

Community Engagement and Education: Empowering communities with knowledge and awareness is fundamental to our mission. Through targeted outreach programs, educational initiatives, and grassroots campaigns, we aim to cultivate a culture of road safety consciousness, fostering a sense of collective responsibility and shared commitment to safer roads for all.

Integration with Smart City Initiatives: In the era of smart cities, our system has the potential to serve as a linchpin in the broader ecosystem of urban mobility and safety. By integrating seamlessly with existing smart city initiatives, we can leverage data analytics, IoT sensors, and predictive modeling to optimize traffic management, mitigate congestion, and enhance overall road safety outcomes.

In summary, our journey towards enhancing road safety through the deployment of a helmet detection system is one characterized by passion, perseverance, and purpose. As we look towards the horizon, we do so with a sense of optimism and determination, knowing that the work we do today has the power to shape a safer, more sustainable future for generations to come.

APPENDIX A

(Code)

Python Script :

```
import os

import torch

import cv2

import numpy as np

from ultralytics import YOLO

import pandas as pd


# Load YOLO model for helmet detection

helmet_model = YOLO('best.pt')


# Load YOLOv5 model for two-wheeler detection

two_wheeler_model = torch.hub.load('ultralytics/yolov5','yolov5s', pretrained=True)
```

```
cv2.namedWindow('ROI')
cv2.setMouseCallback('ROI', POINTS)


cap = cv2.VideoCapture('video1.mp4')
my_file = open("coco1.txt", "r")
data = my_file.read()
class_list = data.split("\n")
count=0


while True:
    ret, frame = cap.read()
    if not ret:
        break
    count += 1


    # Skip frames,
    if count % 2 != 0:
        continue
```

```
frame = cv2.resize(frame, (1020, 600))


# Two-wheeler detection
results_two_wheeler = two_wheeler_model(frame)


for index, row in results_two_wheeler.pandas().xyxy[0].iterrows():
    x1 = int(row['xmin'])
    y1 = int(row['ymin'])
    x2 = int(row['xmax'])
    y2 = int(row['ymax'])
    d = row['name']


    if 'motorcycle' in d:
        y1 = max(0, y1 - int(0.5 * (y2 - y1)))
        y2 = min(frame.shape[0], y2 + 5)
        x1 = max(0, x1 - 5)
        x2 = min(frame.shape[1], x2 + 5)


        # Cropping the two-wheeler region
```

```
cropped_vehicle = frame[y1:y2, x1:x2]


# Helmet detection on the cropped two-wheeler region
results_helmet = helmet_model.predict(cropped_vehicle)
a = results_helmet[0].boxes.data
px = pd.DataFrame(a.cpu().numpy()).astype("float")


for _, row in px.iterrows():
    x1_helmet = int(row[0])
    y1_helmet = int(row[1])
    x2_helmet = int(row[2])
    y2_helmet = int(row[3])
    d_helmet = int(row[5])
    c_helmet = class_list[d_helmet]


    # Conditionally set colors based on class
    if c_helmet == "With Helmet":
        # Green bounding box, white text color with bold font
```

```
            cv2.rectangle(frame, (x1 + x1_helmet, y1 + y1_helmet), (x1 + x2_helmet, y1 + y2_helmet), (0, 255, 0), 2)
            cv2.putText(frame, f'{c_helmet}', (x1 + x1_helmet, y1 + y1_helmet),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2, cv2.LINE_AA)
        elif c_helmet == "Without Helmet":
            # Red bounding box, white text color with bold font
            cv2.rectangle(frame, (x1 + x1_helmet, y1 + y1_helmet), (x1 + x2_helmet, y1 + y2_helmet), (0, 0, 255), 2)
            cv2.putText(frame, f'{c_helmet}', (x1 + x1_helmet, y1 + y1_helmet), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2, cv2.LINE_AA)


 cv2.imshow("ROI", frame)
 key = cv2.waitKey(1) & 0xFF
 if key == 27:  # Esc key
     break
cap.release()
cv2.destroyAllWindows()
```

UI Code :

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>

    <title>Select Video File</title>

    <!-- Bootstrap CSS -->

    <link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css"

      rel="stylesheet"

    />

    <!-- Custom CSS -->

    <style>

      body {

        background-color: #141414;

        color: #fff;

      }
```

```
.container {
  padding: 50px;
}

.form-group label {
  font-size: 1.2rem;
}

.form-control {
  background-color: #212121;
  border: none;
  border-radius: 0;
  color: #fff;
  box-shadow: none;
}

.form-control:focus {
  background-color: #212121;
```

```
      border-color: #007bff;
      color: #fff;
      box-shadow: none;
    }

    .btn-primary {
      background-color: #007bff;
      border: none;
      border-radius: 0;
      padding: 12px 30px;
      font-size: 1.2rem;
      transition: background-color 0.3s ease;
    }

    .btn-primary:hover {
      background-color: #0056b3;
    }
  </style>
</head>
```

```
  <body>
    <div class="container">
      <h1 class="text-center mb-5">Helmet Detection Using Machine
Learning</h1>
      <div class="row justify-content-center">
        <div class="col-md-6">
          <form
            action="/startDetection"
            method="POST"
            enctype="multipart/form-data"
          >
            <div class="mb-3 form-group">
              <label for="video">Choose a video file</label>
              <input
                type="file"
                class="form-control"
                id="video"
                name="video"
                accept="video/*"
```

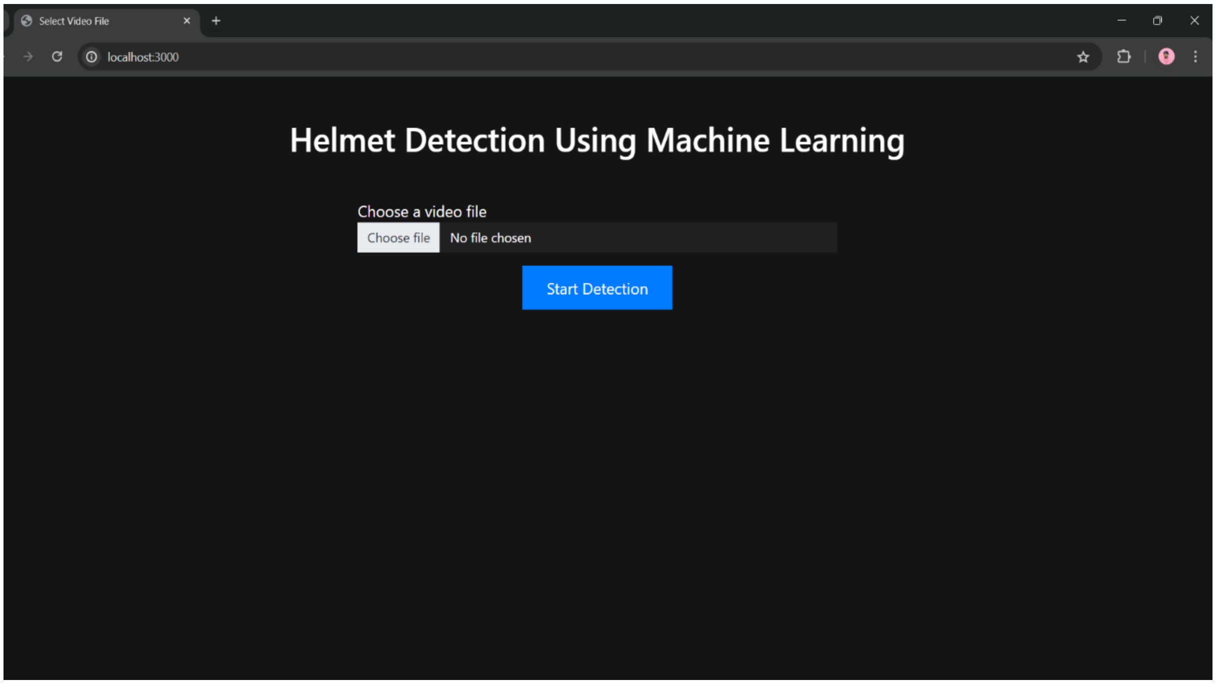
```html
      />
     </div>
     <button type="submit" class="btn btn-primary d-block mx-auto">
      Start Detection
     </button>
    </form>
   </div>
  </div>
 </div>
 <!-- Bootstrap JS (Optional) -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Backend – Code :

```
const express = require('express');

const { spawn } = require('child_process');

const multer = require('multer');

const path = require('path');

const fs = require('fs');

const app = express();

const port = 3000;


// Serve static files from the current directory

app.use(express.static(__dirname));

const storage = multer.diskStorage({

  destination: function (req, file, cb) {

    cb(null, 'uploads/')

  },

  filename: function (req, file, cb) {

    cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname))

  }
```

```
});
const upload = multer({ storage: storage });
app.post('/startDetection', upload.single('video'), (req, res) => {
    const videoPath = req.file.path;
    const pythonProcess = spawn('python', ['both.py', videoPath]);
    pythonProcess.stdout.on('data', (data) => {
        console.log(`stdout: ${data}`);
    });
    pythonProcess.stderr.on('data', (data) => {
        console.error(`stderr: ${data}`);
    });
    pythonProcess.on('close', (code) => {
        console.log(`child process exited with code ${code}`);
    });
    res.sendFile(path.join(__dirname, 'pages', 'detection_started.html'));
});
app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

SNAPSHOTS OF RUNNING PROJECT



REFRENCES

[1] “Road traffic injuries.” Accessed: Jan. 26, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] A. Durgapal and V. Vimal, “Prediction of Stock Price Using Statistical and Ensemble learning Models: A Comparative Study,” 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2021, 2021, doi: 10.1109/UPCON52273.2021.9667644.

[3] V. Vimal, T. Singh, S. Qamar, B. Nautiyal, K. Udham Singh, and A. Kumar, “Artificial intelligence-based novel scheme for location area planning in cellular networks,” Comput Intell, vol. 37, no. 3, pp. 1338–1354, Aug. 2021, doi: 10.1111/COIN.12371.

[4] R. Chauhan and R. C. Joshi, “Comparative Evaluation of Image Segmentation Techniques with Application to MRI Segmentation,” pp. 521–537, 2021, doi: 10.1007/978-981-33-4087-9_44.

[5] S. K. Jagatheesaperumal, P. Mishra, N. Moustafa, and R. Chauhan, “A holistic survey on the use of emerging technologies to provision secure healthcare solutions,” Computers and Electrical Engineering, vol. 99, p. 107691, Apr. 2022, doi: 10.1016/J.COMPELECENG.2022.107691.

[6] “India Registered Motor Vehicles: Two Wheelers | Economic Indicators | CEIC.” Accessed: Jan. 26, 2024. [Online]. Available: https://www.ceicdata.com/en/india/number-of-registered-motor-vehicles/registered-motor-vehicles-two-wheelers

[7] “World Population Clock: 8.1 Billion People (LIVE, 2024) - Worldometer.” Accessed: Jan. 26, 2024. [Online]. Available: https://www.worldometers.info/world-population/

[8] R. R. V. E. Silva, · Kelson, R. T. Aires, · Rodrigo, and M. S. Veras, “Detection of helmets on motorcyclists,” Multimed Tools Appl, vol. 77, pp. 5659–5683, 2018, doi: 10.1007/s11042-017-4482-7.

[9] M. Swapna, T. Wajeeh, and S. Jabeen, “A Hybrid Approach for Helmet Detection for Riders Safety using Image Processing, Machine Learning, Artificial Intelligence,” Int J Comput Appl, vol. 182, no. 37, pp. 50–55, Jan. 2019, doi: 10.5120/IJCA2019918397.

[10] L. Allamki, M. Panchakshari, A. Sateesha, and K. S. Pratheek, “Helmet Detection using Machine Learning and Automatic License Plate Recognition,” 2019.

[11] T. Waris et al., “CNN-Based Automatic Helmet Violation Detection of Motorcyclists for an Intelligent Transportation System,” Math Probl Eng, vol. 2022, 2022, doi: 10.1155/2022/8246776.

[12] “(PDF) HELMET DETECTION ON TWO-WHEELER RIDERS USING MACHINE LEARNING.” Accessed: Jan. 24, 2024. [Online]. Available: https://www.researchgate.net/publication/352771734_HELMET_DETECTION_ON_TWO-WHEELER_RIDERS_USING_MACHINE_LEARNING

[13] R. Silva, K. Aires, and R. Veras, “Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers,” Brazilian Symposium of Computer Graphic and Image Processing, pp. 141–148, Oct. 2014, doi: 10.1109/SIBGRAPI.2014.28.

[14] N. Boonsirisumpun, W. Puarungroj, and P. Wairotchanaphuttha, “Automatic detector for bikers with no helmet using deep learning,” 2018 22nd International Computer Science and Engineering Conference, ICSEC 2018, Jul. 2018, doi: 10.1109/ICSEC.2018.8712778.

[15] P. Sridhar, M. Jagadeeswari, S. H. Sri, N. Akshaya, and J. Haritha, “Helmet Violation Detection using YOLO v2 Deep Learning Framework,” 2022 6th International Conference on Trends in Electronics and Informatics, ICOEI 2022 - Proceedings, pp. 1207–1212, 2022, doi: 10.1109/ICOEI53556.2022.9776661.

[16] N. R. Rajalakshmi and K. Saravanan, “Traffic Violation Invigilation Using Transfer Learning,” Proceedings of the 2022 International Conference on Electronic Systems and Intelligent Computing, ICESIC 2022, pp. 286–292, 2022, doi: 10.1109/ICESIC53714.2022.9783519.

[17] M. Kathane, S. Abhang, A. Jadhavar, A. D. Joshi, and S. T. Sawant, “Traffic Rule Violation Detection System: Deep Learning Approach,” Lecture Notes in Electrical Engineering, vol. 858, pp. 191–201, 2022, doi: 10.1007/978-981-19-0840-8_14/COVER.

[18] J. Chiverton, “Helmet presence classification with motorcycle detection and tracking,” IET Intelligent Transport Systems, vol. 6, no. 3, pp. 259–269, Sep. 2012, doi: 10.1049/IET-ITS.2011.0138/CITE/REFWORKS.

[19] F. Wu, G. Jin, M. Gao, Z. He, and Y. Yang, “Helmet detection based on improved YOLO V3 deep model,” Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control, ICNSC 2019, pp. 363–368, May 2019, doi: 10.1109/ICNSC.2019.8743246.

[20] D. Singh, “Detection of Motorcyclists without Helmet in Videos using Convolutional Neural Network.” Accessed: Jan. 29, 2024. [Online]. Available: https://www.academia.edu/33844625/Detection_of_Motorcyclists_without_Helmet_in_Videos_using_Convolutional_Neural_Network

APPENDIX B

ACCEPTANCE MAIL SNAPSHOT

5/28/24, 2:38 PM Gmail - Acceptance Notification for Your Conference (ICCICA-2024) Paper Submission

Gmail **Abhishek Joshi <allabhishekjoshi@gmail.com>**

Acceptance Notification for Your Conference (ICCICA-2024) Paper Submission
1 message

ICCICA-2024 <iccica2024@easychair.org> 8 April 2024 at 15:41
To: Abhishek Joshi <allabhishekjoshi@gmail.com>

Dear Abhishek Joshi,

Congratulations!

We are delighted to inform you that your paper 65 titled AI Guardian on the Road: Harnessing Machine Learning for Two-Wheeler Helmet Detection, has been accepted for presentation at the upcoming ICCICA to be held from 23rd to 24th May at Panipat Institute of Engineering and Technology, Panipat.

Your submission underwent a thorough review process by our esteemed panel of reviewers and the average score of your paper has crossed the prescribed threshold.

You are requested to submit the following by April 15th, 2024, through the link below that requires:

https://forms.gle/9LhVJWrPSJoTx1a27
1. Camera ready copy [Templates for Latex and Word are attached in this email].
2. Presentation by 15th May 2024. A sample for the presentation is attached herewith for your reference.
3. Copyright Form
[https://www.piet.co.in/wp-content/uploads/2023/06/ieee_Copyright-form.pdf]
4. Registration Details [https://www.piet.co.in/ICCICA24/#registration]
5. Proof of Payment

Please ensure that you adhere to the manuscript template and presentation guidelines as per IEEE Guidelines. Additionally, kindly confirm your registration for the conference by April 15th, 2024.

We believe that your insights will significantly contribute to the success of our conference and foster meaningful discussions among attendees. Also, you are encouraged to join our WhatsApp Group for information related to authors.

Group Link: https://chat.whatsapp.com/KIujfcyub3NIV46YqGp4rV

If you have any questions or require further information, please do not hesitate to contact us at iccica24@piet.co.in

Contact Person: Dr. Upasana Lakhina
Organizing Chair
+91-9813300900.
Best regards,

Organizing Team
ICCICA 2024

ICCICA 2024.rar
982K

https://mail.google.com/mail/u/0/?ik=844cec876b&view=pt&search=all&permthid=thread-f:1795760960286962543&simpl=msg-f:1795760960286… 1/1