CAMARINES SUR POLYTECHNIC COLLEGES
ISO 9001:2015 Certified

CAMARINES SUR POLYTECHNIC COLLEGES
COLLEGE *of* COMPUTER STUDIES

HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS

A Thesis Project
presented to the Faculty of
College of Computer Studies

In Partial Fulfillment of the Requirements
for the degree Bachelor of Science in Computer Science

By
Dela Justa, Aina Mae F.
Epres, Caren Joy L.
Matubis, Maria Angela N.

May 2025

APPROVAL PAGE

In partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science, this research entitled **HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS** prepared and submitted by **Dela Justa, Aina Mae F.**, **Epres, Caren Joy L.**, **Matubis, Maria Angela N.**, , has been examined and is recommended for approval and acceptance.

KAELA MARIE FORTUNO, MIT

Adviser

This research project entitled, **HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS**, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance and approval for ORAL EXAMINATION.

RESEARCH COMMITTEE:

TIFFANY LYN O. PANDES, MSc

Member

ROSEL O. ONESA, MIT

Member

IAN P. BENITEZ, DIT

Chairman

PANEL OF EXAMINERS

APPROVED by the Committee on Oral Examination with a grade of **PASSED** on December 5, 2025.

IAN P. BENITEZ, DIT
Chairman

TIFFANY LYN O. PANDES, MSc
Member

ROSEL O. ONESA, MIT
Member

ACCEPTED and **APPROVED** in partial fulfillment of the requirements in Bachelor of Science in Computer Sciencewith a grade of 90.

ROSEL O. ONESA, MIT
OIC Dean, College of Computer Studies
Date:______________________

DEDICATION

This project is wholeheartedly dedicated to our parents, who have unwaveringly provided us with financial and moral support. Thank you for meeting all our needs throughout the development of our prototype and for teaching us that even the biggest tasks can be achieved when done one step at a time.

We also extend this dedication to our friends and classmates who generously shared their ideas, answered our questions, and supported us whenever guidance was needed. Your insights, encouragement, and patience have played a significant role in our progress, and we are truly grateful for your presence throughout this journey.

Lastly, we dedicate this study to future researchers who may build upon this work. May it serve as a guide and inspiration as you pursue your own innovations.

ACKNOWLEDGMENTS

The researchers would like to express their deepest gratitude to the individuals whose guidance, support, and encouragement made this thesis possible.

First and foremost, we extend our sincere appreciation to Rosel O. Onesa, our Dean and Thesis 1 instructor, for her unwavering guidance, insightful feedback, and dedication in strengthening our foundation as researchers. Her mentorship during the initial stages of this study greatly contributed to the direction and structure of our work.

Our profound thanks also go to Ms. Tiffany Lyn O. Pandes, our Thesis 2 instructor and respected panel member, for her continuous support, constructive suggestions, and patience throughout the development and completion of this project.

We would like to express our heartfelt gratitude to Ms. Kaela Marie Fortuno, our Adviser, whose expertise, encouragement, and professional guidance were vital to the success of this study. Her consistent support and commitment inspired us to pursue excellence.

Special thanks are also extended to Sir Allan Ibo, our Consultant, for sharing his technical knowledge, especially in areas requiring advanced expertise. His contributions significantly enhanced the technical reliability of our system.

We gratefully recognize Ms. Ma. Allaine Agna our Grammarian, for meticulously reviewing our manuscript and ensuring clarity, coherence, and adherence to academic standards.

Our sincere appreciation is likewise extended to the Thesis Panel, chaired by Ian P. Benitez, with members Ms. Tiffany Lyn O. Pandes, and Rosel O. Onesa, for their professional evaluation, constructive recommendations, and guidance that improved the overall quality of this research.

We would also like to acknowledge the riders who willingly participated in our data collection process. Their cooperation in allowing us to capture images and videos was essential for building and training our dataset, and without their assistance, the development of our thesis project would not have been possible.

Deep appreciation is given to the College of Computer Studies, along with all faculty members who have imparted knowledge and skills throughout our academic journey.

We extend heartfelt thanks to our families, friends, and loved ones for their continuous support, patience, and motivation throughout the challenging phases of this study. Their encouragement gave us strength to persevere.

Above all, we express our gratitude to God, whose guidance, wisdom, and blessings sustained us from the beginning until the completion of this thesis. To everyone who contributed—whether directly or indirectly—to the success of this study, thank you very much.

ABSTRACT

Title:	HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS
Authors:	Dela Justa, Aina Mae F. Epres, Caren Joy L. Matubis, Maria Angela N.
Number of Pages:	110
School:	Camarines Sur Polytechnic Colleges
Degree Conferred:	Bachelor of Science in Computer Science
Keywords:	Helmet Detection; Computer Vision; YOLOv8; Deep Learning; Road Safety;

This study developed an automated Helmet Compliance Detection system using Computer Vision and YOLOv8 to address increasing motorcycle-related violations. A total of 1,242 images were collected, annotated, and used to train the model for 150 epochs, resulting in an accuracy of 83.3%. The system effectively detected motorcycles, proper and improper helmet use, and overloading, and it generated automatic video evidence of violations. Field deployment along Nabua Highway showed reliable performance under varying lighting conditions. The results demonstrate that the prototype is a practical solution for real-time helmet compliance monitoring.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Background of the Problem

Motorbike accidents had been steadily increasing worldwide, leading to severe injuries and fatalities. One major contributing factor was the lack of helmet compliance and the dangerous practice of triple riding. In India alone, over 37 million individuals own and operate two-wheelers, making it critical to implement an effective monitoring system to enforce safety regulations and reduce accidents. A webcam was used for real-time video input, capturing and processing images to detect violations. The trained neural network then analyzed the webcam input, providing output based on the learned data. The system achieved an estimated 70% accuracy, with future improvements aimed at enhancing detection precision and real-time performance.[12]. Many motorcyclists frequently violated traffic rules by not wearing helmets, and enforcement by traffic police was often limited due to the demanding nature of manual monitoring. This automated helmet detection prototype had the potential to enhance traffic law enforcement and reduce human intervention, leading to safer road environments [5].

The requirement for ongoing surveillance, particularly in busy locations or along lengthy stretches of road, exacerbated this problem. The safety of motorcycle riders was directly put at risk by the ineffectiveness of the enforcement procedures. The creation of an automated, vision-based safety identification and monitoring that could precisely identify the presence or absence of helmets in real-time was required to solve this issue [11]. Given the significant portion of traffic-related fatalities attributed to motorcycle accidents resulting from non-compliance with helmet regulations. Acknowledging the critical role of helmets in rider protection, this paper presented an innovative approach to helmet violation detection using deep learning methodologies [19].

Deep learning, a subset of machine learning that uses artificial neural networks to learn from large amounts of data, was applied in automatic helmet detection. Deep learning models were trained using large datasets of helmet-wearing and non-helmet-wearing people. The neural networks learned to recognize the features that distinguished helmet-wearers from non-helmet-wearers. Once trained, the deep learning model was used to automatically detect whether a rider was wearing a helmet or not [21].

Motorcycle-related accidents had become a growing concern worldwide, significantly contributing to road injuries and fatalities. According to the World Health Organization (WHO), more than 1.35 million people died annually due to road crashes, with motorcycle riders being among the most vulnerable. One of the leading causes of these accidents was the failure to wear helmets, which served as a critical protective measure against head injuries. Despite laws mandating helmet use, non-compliance remained a widespread issue, exacerbated by weak enforcement and inadequate monitoring. In the Philippines, motorcycle accidents significantly increased over the years, making them one of the leading causes of road fatalities. According to the Metropolitan Manila Development Authority (MMDA), in 2022, motorcycle-related accidents accounted for more than 30% of road crash incidents in Metro Manila alone, resulting in severe injuries and fatalities. MMDA also reported a 17.3 percent increase in motorcycle-related road crashes in 2023. Based on the data from its Road Safety Unit, the MMDA said that a total of 26,599 motorcycle-related crashes were recorded in 2022 [14].

Republic Act No. 10054, also known as the Motorcycle Helmet Act of 2009, mandated that all motorcycle riders and their passengers wear standard protective helmets while on the road. This law aimed to reduce head injuries and fatalities by ensuring that helmets meet specific safety standards. Despite the implementation of Republic Act No. 10054, also known as the Motorcycle Helmet Act of 2009, which mandates all motorcycle riders to wear standard protective helmets, many riders continued to violate this law, leading to preventable deaths [17]. The so-called "nutshell helmet" is classified as an unsafe and non-standard protective gear for motorcycle riders. It does not comply with the regulations outlined in Republic Act No. 10054, or the Motorcycle Helmet Act of 2009, which mandates that all motorcycle riders and passengers must wear protective helmets bearing a Philippine Standard (PS) or Import Commodity Clearance (ICC) mark certified by the Bureau of Product Standards (BPS). Typically manufactured from inferior materials, nutshell helmets provide insufficient protection to the head and face and are often devoid of visors, reducing visibility during adverse weather conditions such as rain or strong winds. These helmets are originally intended for bicycles or skateboards, and therefore, lack the durability and impact resistance required for motorcycle use. Consequently, the Land Transportation Office (LTO) and the Inter-Agency Council for Traffic (I-ACT) have strongly discouraged their use. Some local government units (LGUs), including Dagupan City, have even imposed penalties on individuals found using such substandard helmets. To ensure both safety and regulatory compliance, motorcycle riders are advised to wear full-face or J-type helmets with PS or ICC certification, as these provide greater protection and effectively minimize the risk of head injuries in road accidents.

A major challenge in enforcing helmet compliance was the reliance on manual monitoring by law enforcement officers, which was often inconsistent and inefficient. Traditional methods such as road checkpoints and manual inspections required significant resources and were prone to human error. Moreover, with the increasing number of motorcyclists on the road, it became nearly impossible for authorities to monitor helmet compliance effectively. The absence of a scalable and automated monitoring system contributed to the ongoing problem, creating a need for technological solutions that ensured stricter enforcement of traffic laws. With advancements in artificial intelligence (AI) and computer vision, deep learning technologies emerged as powerful tools for automating helmet compliance detection. Deep learning, a subset of AI, enabled machines to process vast amounts of visual data, recognize patterns, and make accurate classifications. Technologies such as

YOLO (You Only Look Once) and OpenCV allowed real-time helmet detection with high precision, making them ideal for traffic monitoring applications. These technologies had been implemented in smart surveillance systems for vehicle detection, passenger counting, and helmet compliance monitoring.

To address the limitations of manual enforcement, this research proposed the development of a Helmet Compliance Detection prototype using computer vision and deep learning algorithms to automatically detect whether motorcycle riders were wearing helmets correctly. The prototype focused on enhancing helmet compliance monitoring through several key features. It accurately determined if a rider was properly wearing a helmet on their head and not just carrying it. It also verified if the helmet was securely fastened and correctly positioned. Helmets, generally classified into several categories based on their structure and intended use, were considered in the study. The prototype concentrated on the standard motorcycle helmet, which covered the entire head and included a chin strap and often a visor. This type of helmet offered the most protection and was typically required by law in many regions. The prototype also included vehicle filtering, which identified if a vehicle was a motorcycle or not. If yes, it proceeded to helmet detection. In helmet detection, there were three classes: person with no helmet, person with proper helmet, and person with wrong helmet use. In the person with wrong helmet use category, it included improper use of a helmet such as not fastened correctly, just holding the helmet, or wearing the wrong helmet (e.g., bicycle helmets, construction helmets). This helped prevent the use of improper or mismatched helmets, which were flagged as violations to promote stricter adherence to safety standards. Moreover, the prototype enforced passenger limits by counting riders to ensure no more than two people were on a motorcycle at any time. Any overloading was automatically flagged as a violation. Upon detecting any violations, a red warning was displayed on the system monitor, and the prototype automatically saved short video clips as evidence, supporting authorities in tracking and penalizing repeat offenders. By integrating these features, the prototype aimed to improve road safety, assist law enforcement in effectively implementing helmet laws, and ultimately reduce motorcycle-related accidents and fatalities.

Statement of the Problem

Many motorcycle riders did not follow helmet laws, which led to a high risk of accidents, serious injuries, or even death. Traffic officers faced challenges in manually checking whether motorcycle riders were wearing helmets, as the process was time-consuming and required significant effort. Since officers could not monitor every rider, many violations went unnoticed, making the enforcement of helmet laws difficult. Identifying helmet usage under various conditions was a challenge for the proposed prototype. In poor lighting, such as at night or in dark areas, the prototype struggled to clearly identify the rider's head. Similarly, in adverse weather conditions like fog or heavy rain, recognizing helmets was difficult. When there were large numbers of motorcycles, it was hard to check if each rider was wearing a helmet. Because of these challenges, the prototype needed to be tested to ensure it accurately detected helmets and provided reliable results. It was evaluated under different conditions such as varied weather and lighting. Its speed and real-time detection performance had to be assessed to ensure it was reliable in supporting road safety efforts.

Objectives of the Study

This section outlined the study's objectives in developing an AI-based helmet detection prototype to improve road safety.

General Objective

The main objective of this study was to design and develop a Helmet Compliance Detection Using Computer Vision for Safer Roads that effectively monitored and detected helmet violations among motorcycle riders using Artificial Intelligence (AI), Deep Learning, and

Computer Vision. This prototype aimed to provide an accurate and automated solution for identifying non-compliance with helmet regulations, reduced reliance on manual monitoring, and enhanced the enforcement of road safety laws.

Specific Objectives

The specific objectives of this study were as follows:

1. Implemented deep learning models using YOLO for object detection, and OpenCV for image and video processing.

2. Developed an Artificial Intelligence-based prototype integrating the implemented models for helmet detection.

3. Evaluated the performance of the designed helmet detection prototype under different conditions such as lighting variations, weather changes, and multiple riders using accuracy, precision, recall, F1-score, mean average precision (mAP), and detection speed measured in frames per second (FPS).

Significance of the Study

This study focused on applying Artificial Intelligence in traffic law enforcement, particularly in monitoring motorcycle helmet compliance. It benefited the following stakeholders:

- **Students.** Particularly those studying Computer Science gained valuable insights into the practical applications of AI in traffic law enforcement. This study served as a reference for developing intelligent transportation systems and encouraged innovative approaches to road safety.

- **Motorcycle Riders.** By ensuring helmet compliance, the prototype promoted rider safety, reduced the risk of severe injuries or fatalities, and encouraged responsible riding behavior.

- **Law Enforcement.** The prototype automated helmet compliance monitoring, reduced manual inspections, and improved accuracy. It enhanced efficiency, minimized human error, and provided valuable data for road safety policies.

- **Camarines Sur.** The implementation of this prototype benefited Camarines Sur by improving road safety and reducing motorcycle-related accidents. Local authorities used this technology to enhance traffic enforcement and ensure compliance with helmet laws.

- **Researcher.** This research established a foundation for AI-driven traffic monitoring, enabling further studies in deep learning, object detection, and real-time surveillance.

- **Future Researchers.** The study laid the foundation for further research on AI-driven law enforcement systems, enabling advancements such as database integration and expanded traffic violation detection.

Scope and Limitation

This study aimed to develop and implement an AI-based prototype that used YOLOv8 for detecting motorcycles, e-bikes, and bicycles, as well as helmet usage. OpenCV was used for real-time video and image processing, and the prototype identified whether riders were wearing helmets properly. In addition, it counted the number of passengers on each vehicle to ensure compliance with road regulations, particularly limiting motorcycle passengers to two. The prototype was deployed along Nabua Highway. The implementation involved capturing real-time video through strategically placed surveillance cameras.

The prototype's outputs, including flagged violations such as no helmet, improper helmet use, or overloading, were stored as video clips for review by authorities. However, the prototype had limitations. It functioned only in areas covered by surveillance cameras. Its accuracy declined in low-light or adverse weather conditions such as rain or fog. Differentiating among similar vehicle types (e.g., tricycle, e-bikes, and motorcycles) sometimes introduced errors. The prototype was not connected directly to enforcement systems during the pilot phase and initially functioned as a standalone system.

Project Dictionary

To avoid problems in understanding the terms used, the following technical terms are conceptually and operationally defined to provide better understanding.

- **AI (Artificial Intelligence).** The simulation of human intelligence in machines that enabled them to perform tasks such as learning, reasoning, and visual recognition [18]. In this study, the prototype integrated AI-powered computer vision models to automatically analyze video data, detect helmets and count passengers without human intervention.

- **Algorithm.** A set of well-defined instructions or rules used to solve a specific problem or perform a computation [1]. In this study, the prototype used machine learning and image processing algorithms to detect helmets, count passengers, and recognize plate numbers from camera feeds.

- **Accident Prevention.** Encompasses strategies and measures aimed at reducing the occurrence of unintended events that resulted in injury, death, or property damage. It involves identifying potential hazards, assessing risks, and implementing interventions to mitigate these risks [8]. In this study, accident prevention referred to the deployment of artificial intelligence (AI) and computer vision technologies to monitor and analyze real-time data from surveillance prototypes. The goal was to detect and alert authorities about potential accidents or safety violations, thereby enabling timely interventions to prevent incidents.

- **Computer Vision.** A field of artificial intelligence that enabled computers and systems to derive meaningful information from digital images, videos, and other visual inputs [20].In this study, the prototype processed video feeds from cameras to automatically detect helmets and count passengers without manual intervention.

- **Dataset.** A structured collection of data used to train or evaluate machine learning models. In computer vision, datasets consisted of labeled images or videos [2]. In this study, the prototype utilized a dataset containing images of motorcycle riders with and without helmets, plate numbers, and various riding conditions to train the object detection model. These datasets were sourced from public datasets or collected manually for model training and validation.

- **Deep Learning.** A subset of machine learning involving neural networks with multiple layers that learned patterns and representations from large datasets [7]. In this study, AI models were used to detect helmets in video using deep neural networks.

- **Helmet.** A protective covering for the head, typically made of a hard material, used as part of safety gear to prevent head injuries [13]. In this study, it was what the prototype identified in real-time using computer vision techniques, ensuring that riders wore it properly.

- **Helmet Compliance.** Wearing of a helmet the right way and following the law when riding a motorcycle [23]. It helped prevent injuries and deaths in road accidents. In this study, helmet compliance was the main focus. The system used YOLOv8 to check if riders were wearing helmets properly and to spot those who were not, to help make roads safer.

- **Helmet Detection.** A computer vision task that involved identifying and verifying the presence of a helmet on a person in images or videos [9]. In this study, the prototype detected helmets in real-time using computer vision algorithms and determines if helmets were worn on the head and not held or carried by the riders.

- **Image Processing.** The manipulation of images through computational algorithms to enhance quality or extract useful information [6]. In this study, image processing techniques analyzed video footage to detect helmets and passengers, ensuring compliance with safety regulations and identifying violations.

- **Law Enforcement.** Refers to the system and practices used by government agencies to ensure public order, uphold laws, and prevent or investigate criminal activities [3]. In this study, AI-driven surveillance aided authorities by detecting violations, gathering evidence, and enhancing enforcement efficiency through real-time monitoring.

- **Object Detection.** A computer vision technique that identified and located objects within an image or video [22]. In this study, object detection was used to recognize motorcyclists and determine whether they were wearing helmets by analyzing real-time footage from surveillance cameras.

- **Passenger Counting.** The process of counting the number of passengers in a vehicle using sensors or computer vision techniques [10]. In this study, the prototype employed image processing and object detection to count passengers and compared this number with detected helmets to ensure compliance.

- **Road Safety.** The methods and measures used to prevent road users from being killed or seriously injured, including regulations, infrastructure, and education [15]. In this study, road safety included enforcing helmet laws, using an AI-powered monitoring prototype, improving traffic management, and promoting awareness campaigns.

- **Traffic Monitoring.** The systematic observation and recording of vehicular movement and flow on roads, often used to manage congestion and improve traffic systems [4]. In this study, traffic monitoring involved using AI-powered cameras and sensors to detect motorcyclists, assess helmet usage, and identify violations in real time.

- **YOLO (You Only Look Once).** A deep learning-based object detection model that processed an image in a single pass to detect multiple objects in real time [16]. In this study, the prototype employed YOLOv8 to efficiently detect helmets on motorcycle riders and identify passengers within video footage.

Notes

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms*. (3rd ed.). MIT Press. https://archive.org/details/introduction-to-algorithms-third-edition-2009.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2020. Imagenet: a large-scale hierarchical image database. In *2020 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255. DOI: 10.1109/CVPR.2009.5206848.

[3] Encyclopaedia Britannica. n.d. Law enforcement. (n.d.). https://www.britannica.com/topic/law-enforcement.

[4] Encyclopaedia Britannica. n.d. Traffic control. (n.d.). https://www.britannica.com/technology/traffic-control.

[5] Tanmayee Godbole, Purva Sapkal, Sujal Malapure, and Shrihari Ganbavale. 2024. Safe rider: helmet detection using machine learning and open cv. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. https://www.ijraset.com/best-journal/safe-rider-helmet-detection-using-machine-learning-and-open-cv.

[6] Rafael C. Gonzalez and Richard E. Woods. 2018. *Digital Image Processing*. (4th ed.). Pearson. https://www.pearson.com/store/p/digital-image-processing/P100000006584.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. https://www.deeplearningbook.org/.

[8] Lennart Harms-Ringdahl. 2013. *Guide to Safety Analysis for Accident Prevention*. IRS Riskhantering AB. https://irisk.se/sabook/SA-book1.pdf.

[9] Ahatsham Hayat and Fernando Morgado-Dias. 2022. Deep learning-based automatic safety helmet detection system for construction safety. *Applied Sciences*, 12, 16, 8268. DOI: 10.3390/app12168268.

[10] Hyunduk Kim, Myoung-Kyu Sohn, and Sang-Heon Lee. 2022. Development of a real-time automatic passenger counting system using head detection based on deep learning. *Journal of Information Processing Systems*, 18, 3, 428–442. DOI: 10.3745/JIPS.04.0246.

[11] Raman Kumar, Yogendra Kumar, Ahmed Alkhayyat, Lovneesh Sharma, Ankita Joshi, and Ayush Dogra. 2023. Vision-based safety identification and monitoring system using tensorflow. In *International Conference on Intelligent Systems Design and Applications*. Springer Nature Switzerland, (Dec. 2023), 369–375. DOI: 10.1007/978-3-031-64650-8_36.

[12] Hema Latha Maddi. 2023. Advanced computer vision-based surveillance system for helmet detection and triple rider identification on motorcycles using machine learning. *Unpublished Manuscript*. https://Advanced_Computer_Vision-Based_Surveillance_System_for_Helmet_Detection_and_Triple_Rider_Identification_on_Motorcycles_using_Machine_Learning.

[13] Merriam-Webster. [n. d.] Helmet. Merriam-Webster.com Dictionary. (). https://www.merriam-webster.com/dictionary/helmet.

[14] Metropolitan Manila Development Authority. 2023. Mmda logs 17.3% increase in motorcycle-related crashes in 2023. Inquirer.net News. (2023). https://newsinfo.inquirer.net/1927991/mmda-logs-17-3-increase-in-motorcycle-related-crashes-in-2023.

[15] Oxford Reference. [n. d.] Road safety. In *A Dictionary of Public Health*. (). https://www.oxfordreference.com/display/10.1093/acref/9780199218707.001.0001/acref-9780199218707-e-3052.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2020. You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://arxiv.org/abs/1506.02640.

[17] Republic of the Philippines. 2009. Republic act no. 10054 - motorcycle helmet act of 2009. Official Law. (2009). https://ltoportal.ph/ra-10054-motorcycle-helmet-act-philippines/.

[18] Stuart Russell and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*. (4th ed.). Pearson. https://aima.cs.berkeley.edu/.

[19] Yahia Said, Yahya Alassaf, Refka Ghodhbani, Yazan Ahmad Alsariera, Taoufik Saidani, Olfa Ben Rhaiem, Mohamad Khaled Makhdoum, and Manel Hleili. 2024. Ai-based helmet violation detection for traffic management system. *CMES-Computer Modeling in Engineering & Sciences*, 141, 1. DOI: 10.32604/cmes.2024.052369.

[20] Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications*. Springer. DOI: 10.1007/978-1-84882-935-0.

[21] Vikash Kumar Thakur. 2024. Enhancing safety: a computer vision approach to helmet detection. Medium. (2024). https://medium.com/@vkt08/enhancing-safety-a-computer-vision-approach-to-helmet-detection-c5a48a7275d2.

[22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9627–9636. DOI: 10.1109/ICCV.2019.00972.

[23] World Health Organization. 2018. Helmet use saves lives. Fact sheet. (2018). https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

CHAPTER 2

RELATED LITERATURE AND STUDIES

This chapter presented a review of both international and local literature relevant to the research topic. The researchers collected information using the college library, the internet, and various other references that assisted them in their study.

Helmet Detection using Computer Vision

Helmet detection using computer vision involved automatically identifying helmet use among motorcyclists through AI and image processing. Using deep learning models like YOLO and tools like OpenCV, this prototype detected violations in real-time. It enhanced road safety, supported law enforcement, and reduced manual monitoring in traffic surveillance applications.

According to Afzal et al. [2021] developed a deep learning-based automatic helmet detection system for real-time videos. The researchers used various models, with Faster R-CNN and Region Proposal Network (RPN) addressing challenges like low resolution, weather, occlusion, and illumination. The model was retrained on a self-generated dataset from three locations in Lahore, Pakistan. The system achieved an impressive accuracy of 97.26%, demonstrating its potential to improve authorities' ability to monitor motorcyclists violating traffic laws [1]. Another review from Singh [2024] emphasizes how computer vision improves helmet compliance in dangerous industries by using AI-powered detection to monitor safety in real-time, enforce compliance, and reduce workplace dangers [25]. Furthermore, Siebert et al. [2024] developed a low-cost and affordable computer vision method to check if helmets are used by motorcycle riders in five cities in Southeast Asia using crowdsourced images from Mapillary. They trained their algorithm on over 800,000 images and it achieved high accuracy and detected over 1.3 million motorcycles. The results show that drivers are more often to wear helmets than passengers and people wear helmets more on big roads than small roads. This approach is helpful because it is accurate and useful without the need for people to go out and collect data [24].

According to Giron et al. [2020] the no helmet no ride law that was implemented in the Philippines is still not working because many motorcycle riders are not following it. The government has partnered with De La Salle University to fix this issue by using artificial intelligence. The reseachers used Computer Vision to automatically check if riders are wearing helmets or not. The system used deep learning, especially Convolutional neural networks to improve detection accuracy [7]. According to Soltanikazemi et al. [2023] , the helmet violation detection system was YOLOv5-based and it is developed using genetic algorithm optimization to monitor in real-time. The model achieved a high accuracy, and it was ranked 4th in the AI City Challenge in 2023. The study shows the advantages of deep learning and discusses how it is more effective than the traditional methods in ensuring helmet compliance and enhancing motorcycle safety in roads [26].

The work of Mutyala et al. [2023], introduced a real-time helmet detection warning system that is powered by Detection Transformers (DETR) for improving detection precision and operational effectiveness in improving motorcycle safety. The system detects motorcycle riders that are not using and wearing helmets in real time, and it will generate alerts to improve the road safety. DETR uses a self-attention mechanism to capture complex relationships in image sequences, allowing accurate helmet detection even in difficult conditions like poor lighting. The system combines video feed analysis with DETR's object detection features, ensuring minimal processing delay. Testing results show the system's high precision and recall rates in different situations. This solution can be customized to send alerts to authorities or directly notify and inform riders, this might decrease violations and promote safety [15]. Tomas and Doma [2023], used the YOLOv5 algorithm in their study, YOLOv5 is used to detect helmets, and it classifies their usage among motorcycle riders in the Philippines. The researchers processed video footage from Makati City and optimized the model hyperparameters for better accuracy. The study suggested enhancing data consistency and using separate models for detection and classification tasks. The findings showed the best results in helmet detection, which could contribute more to road safety [28].

Object Detection Models and Image Processing Techniques

Object detection models like YOLO are widely used for real-time detection tasks. YOLO, combined with OpenCV for image processing and TensorFlow for model training and deployment, offers efficient and accurate object recognition. These tools work together in applications such as traffic monitoring, helmet detection, and safety compliance systems. Image processing techniques are fundamental in preparing and enhancing visual data for analysis by machine learning models. In this study, several image processing methods were applied to ensure that the system accurately detects whether an individual is wearing a helmet. These techniques help in improving the quality of the input data, extracting relevant features, and enabling more accurate and efficient model training.

The YOLO (You Only Look Once) is a deep learning-based object detection model that can detect multiple objects in real-time with high speed and accuracy. According to the study of Jiang et al. [2022] , the YOLO algorithm was improving and evolving from time to time, and it makes object detection faster and more accurate. It compares different YOLO versions and explains its performance compared to traditional methods like Convolutional Neural Networks (CNNs). They highlight that YOLO is still improving and evolving and it is very useful in areas like in security, finance, and other applications [9]. Similarly Terven et al. [2022] discuss the evolution of YOLO from its first version to YOLOv8, YOLO-NAS, and YOLO with transformers. It highlights its improvements in architecture, accuracy, and speed. They compare YOLO with different models like R-CNN and SSD and they explore its applications in fields of robotics, healthcare, security, and traffic monitoring. Future research aims to improve YOLO's real-time detection and its efficiency [27]. YOLO (You Only Look Once) was introduced by Joseph Redmon and his team in 2015 to address the limitations of earlier object detection models like Fast R-CNN. While Fast R-CNN was accurate, it was too slow for real-time applications, taking 2–3 seconds to process a single image. In contrast, YOLO performs detection with just one forward pass through the network, enabling much faster and real-time predictions [6]. With the increasing number of motorcycle users and the issue of helmet non-compliance, Kumar et al.[2024] developed a Real-Time Helmet Detection System to improve road safety by detecting helmet violations and capturing vehicle license plates. The system uses YOLO for object detection and a mechanism for license plate recognition, consisting of three steps: identifying motorcyclists, verifying helmet usage, and capturing the license plate. It achieved 64% accuracy in vehicle identification, 78% in helmet detection, and 92% in license plate recognition [12]. In Addition, Muhammad et al. [2024] designed a real-time helmet detection system using YOLOv8, deployed on edge devices to enhance the safety of motorcyclists in Indonesia. During testing, the model demonstrated strong performance in detecting helmets (91.1% F1 score), riders (81.7%), and non-helmeted riders (33.0%). They also evaluate the system's CPU usage (78%), RAM (77.4%), temperature (33°C–65°C), and power consumption (6.5 W). This system shows potential for integration into smart city infrastructure, improving the efficiency of traffic law enforcement [14]. Furthermore,Choubey et al. [2025] introduces a YOLOv3-oriented model created for identifying license plates and helmets within images. They improved data quality and variety by pre-processing and created a tailored annotated dataset for helmets and license plates. The model underwent training through a multi-phase approach [4].

OpenCV is an open-source library that provides a vast collection of tools for computer vision tasks, including image processing, feature extraction, object recognition, and real-time video analysis. According to Satheesh et al. [2024] using OpenCV, a publicly available computer vision library. For the goal of observing objects, attributes for extraction and image preprocessing, OpenCV offers a broad array of tools and functionalities. By utilizing OpenCV, we will guarantee consistency and reliability when managing various real-world situations, encompassing various lighting situations, obstructions, and vehicle angles [22].

Vehicle Classification

According to Chandrika et al. [2020], the growing number of vehicles exceeding 1 billion globally makes it difficult for authorities to manage traffic and provide sufficient infrastructure. Their study introduces a vehicle detection and classification system using image processing, broken into six stages: image acquisition, analysis, object detection, counting, classification, and result display. The proposed system helps monitor traffic flow, detect rule violations, and classify vehicles into categories such as motorcycles, cars, vans, and trucks, thereby supporting better traffic planning and management [3].

Similarly, Ong et al. [2022], vehicle classification plays a key role in enhancing security, managing traffic congestion, and preventing road accidents. One challenge in this process is the poor image quality from video sources, which makes object recognition difficult. To address this, their study implemented and compared YOLOv5 and Faster R-CNN algorithms for classifying vehicles into five categories: motorcycle, car, van, bus, and lorry. The results showed that YOLOv5 outperformed Faster R-CNN, achieving a mean average precision (mAP) of 0.91, precision of 0.81, and recall of 0.86, making it more suitable for accurate vehicle classification using video-based image data [16].

In line with this,Sanjana et al. [2021], vehicle detection and classification have become increasingly important due to the growing number of vehicles, traffic violations, and road accidents. Their review explores various methodologies that have evolved over the years, shifting from basic image processing to machine learning approaches. This progression has led to the integration of helmet detection and license plate recognition, using object detection and text recognition models that are now easier to implement through built-in frameworks or customizable tools [21].

Moreover, Espinosa et al. [2021] , motorcycles are classified as Vulnerable Road Users (VRUs), alongside bicycles and pedestrians, and are among the most frequently involved in urban traffic accidents. To address this issue, their study reviews the use of automatic video processing techniques particularly leveraging CCTV surveillance systems for the detection and tracking of motorcycles. The authors emphasize the effectiveness of deep learning algorithms within the field of computer vision for these tasks. Additionally, they discuss the use of standard performance metrics, introduce the Urban Motorbike Dataset (UMD) for evaluation purposes, and outline current challenges and potential future research directions in this emerging field [5].

Passenger Counting

Passenger counting systems utilize sensors and computer vision models to automatically count individuals boarding or exiting vehicles. These prototypes often use YOLO for real-time detection and OpenCV for image processing. They help optimize public transport operations, monitor capacity, and improve service efficiency in buses, trains, and other mass transit systems.

The study of Rendon et al. [2023], which introduced a computer vision method using deep learning to detect, count and estimate the number of passengers in Bogota's Trans-Milenio stations, this study shows how accurate passenger counting in public transport systems is important. They analyzed images with nearly 900,000 labeled heads and achieved a very accurate result, with an error of only one person per image. This is better than counting them by hand. This method is scalable and low-cost, and it is useful for improving the planning and running of public transport systems [19]. The paper by Radovan et al. [2024] discusses different passenger counting systems, comparing traditional technologies like RFID and infrared sensors with newer methods using image processing and machine learning. It explores the advantages and disadvantages of each system and how to improve these. It also discusses concerns under GDPR. The authors propose some improvements for passenger counting solutions and suggest ways to enhance public transport operations to make it more effective [18].

According to the study by Bhatt et al. [2024] , wearing a helmet when motorcycling is important because it helps reduce the likelihood of serious head injuries in accidents. With the help of modern technology such as real-time surveillance and computer vision, it is now possible to automatically determine whether riders are wearing a helmet using video footage on the road. The aim of this system is to strengthen the implementation of road safety laws by detecting not only the driver but also the passenger if they are wearing a helmet. Based on a report by the World Health Organization (WHO) in 2023, the correct use of a helmet reduces the risk of death by 42% and the risk of head injury by 69% [WHO, 2023] [2].

Evaluation Metrics of YOLOv8

YOLO (You Only Look Once) is a real-time object detection algorithm that offers a faster and more efficient alternative to traditional detection methods. Specifically, as a single-stage detector, YOLO employs a convolutional neural network (CNN) to predict both bounding boxes and object classes directly from input images. It achieves this by dividing the image into a grid, which enables the detection of multiple objects in a single pass [11].

In this context, several studies have evaluated the performance of YOLO and its variants in terms of speed, accuracy, and adaptability. According to Prakash and Palanivelan [2024], YOLO revolutionized object detection by enabling real-time performance through its single-pass, grid-based prediction approach [17]. Moreover,Karthika et al. [2024] assessed YOLOv8 for its high precision and speed, highlighting its effectiveness across static images, video streams, and live feeds [10]. Furthermore,Varghese and Sambath [2024] demonstrated that YOLOv8 outperforms earlier versions by integrating attention mechanisms, dynamic convolution, and voice recognition, which results in improved accuracy and computational efficiency [29]. Similarly, Safaldin et al. [2024] proposed an enhanced

YOLOv8 model tailored for detecting moving objects in dynamic environments. Through architectural and preprocessing modifications, their model improved motion sensitivity and achieved strong results on datasets such as KITTI, LASIESTA, PESMOD, and MOCS recording 90% accuracy, 90% mAP, 30 FPS, and 80% IoU [20]. However,Hussain [2024] conducted a comparative analysis of YOLO architectures, noting that YOLOv8 features enhanced feature extraction and anchor-free detection, while YOLOv10 achieves even greater real-time performance by incorporating large-kernel convolutions and eliminating non-maximum suppression [8].

Synthesis of the State-of-the-Art

The reviewed literature highlighted the growing importance and effectiveness of computer vision and deep learning techniques in addressing road safety concerns, particularly in enforcing helmet compliance, and counting passengers in real-time.

International and local studies consistently emphasize the role of helmet detection systems using deep learning models such as YOLO (You Only Look Once), Faster R-CNN, and Detection Transformers (DETR). Afzal et al. [2021] demonstrated a highly accurate system using Faster R-CNN, achieving 97.26% accuracy despite challenges like occlusion and weather conditions [1]. Complementing this, Singh [2024] and Giron et al. [2020] recognized the potential of AI in improving helmet compliance in both traffic and industrial settings [25], [7]. Further innovations were noted by Siebert et al. [2024] , who employed crowdsourced images for a low-cost helmet detection approach, and by Soltanikazemi et al. [2023] , whose YOLOv5-based system earned top ranks in the AI City Challenge. Mutyala et al. [2023] introduced DETR-powered real-time systems with alert features, while Tomas and Doma [2023] highlighted the importance of model optimization in improving helmet detection in the Philippines [24], [26], [15], [28].

The integration of object detection and image processing tools YOLO, OpenCV, TensorFlow has been fundamental in enabling real-time, accurate, and resource-efficient systems. Jiang et al. [Jiang et al.] and Terven et al. [2022] chronicled the evolution of YOLO from its initial versions to YOLOv8 and YOLO-NAS, highlighting architectural improvements and broader applications across various fields [9], [27].

YOLO (You Only Look Once) is a real-time object detection algorithm that efficiently detects multiple objects in a single pass using a grid-based CNN approach. Studies highlight its speed, accuracy, and adaptability, especially in its latest version, YOLOv8.Prakash and Palanivelan [2024] emphasized its real-time performance, while Karthika et al. [2024] noted its high precision in various visual inputs [17], [10].Varghese and Sambath [2024] showed improvements in YOLOv8 through added attention mechanisms and voice recognition, and Safaldin et al. [2024] reported strong performance in dynamic environments [29], [20].Hussain [2024] compared YOLO versions, noting YOLOv8's enhanced detection and YOLOv10's improved performance using large-kernel convolutions and anchor-free techniques [8].Kumar et al. [2023] and Muhammad et al. [2024] demonstrated real-world implementations that integrate YOLO yielding high recognition rates and strong performance on edge devices [13], [14]. Similarly, Choubey et al. [2025] emphasized dataset preparation and multi-phase training in developing a YOLOv3 model for detecting helmets and plates [4].

The utility of OpenCV was established by Satheesh et al. [2024] , who showed how it aids in preprocessing, feature extraction, and robustness in varied real-world conditions [22]. In tandem to emerged as a powerful framework for training and deploying models, with Kumar et al. [2024] and Sharma [2024] showcasing its flexibility in creating cost-effective and scalable safety monitoring systems. [12] [23]

Several studies highlight the importance of vehicle classification in improving traffic management, security, and accident prevention. Espinosa et al. [2021] focus on tracking motorcycles using deep learning, while Sanjana et al. [2021] highlight the shift from traditional image processing to integrated helmet and plate detection using modern frameworks

[5], [21]. Ong et al. [2022] show YOLOv5's superior accuracy over Faster R-CNN, and Chandrika et al. [2020] propose a full system for detecting, counting, and monitoring vehicles [16], [3]. Despite varying approaches, all studies support the effectiveness of computer vision in traffic-related applications.

The literature also covers passenger counting systems, crucial for optimizing transport services. Rendon et al. [2023] developed a deep learning method for head counting in Bogotá, yielding minimal errors and proving useful for transit planning [19]. Radovan et al. [2024] compared traditional methods like RFID with modern image processing approaches, offering improvements under regulatory frameworks such as GDPR [18]. Bhatt et al. [2024] expanded on this by developing a real-time system that also includes helmet detection for both drivers and passengers, echoing the WHO's findings on the life-saving importance of helmets [2].

In summary, the reviewed studies underscore a significant trend: AI-powered computer vision prototypes are revolutionizing public safety enforcement. Tools like YOLO, TensorFlow and OpenCV when integrated with real-time video analysis form the backbone of intelligent traffic monitoring systems. These systems not only automate compliance checks but also promise scalability, cost-efficiency, and broad applicability in smart city infrastructures.

Gap Bridged of the Study

The existing helmet detection systems mostly focused on identifying if the motorcycle driver was wearing a helmet, often ignoring the passenger. These systems were usually made for controlled or international settings and did not consider the real traffic conditions in the Philippines, such as poor lighting, blurry movements, and blocked views in live road situations. Many of these systems also checked only if a helmet was present, without checking if it was worn properly or securely fastened. In some cases, riders just carried the helmet instead of wearing it, and these systems could not tell the difference. Also, most existing systems did not check if the rider was using the correct helmet type for the motorcycle. Another issue was that these systems did not detect overloading, where more than two people were riding a motorcycle, something that was common but often ignored.

To address these problems, this study presented a real-time helmet compliance monitoring prototype designed specifically for Philippine roads like the Nabua Highway. The prototype used the YOLOv8 object detection model to detect if helmets were being worn correctly by both drivers and passengers, checked if the helmet matched the type of vehicle, counted the number of riders to spot overloading, and saved short video clips of violations as evidence. This offered a more complete, localized, and practical way to support traffic law enforcement and improve road safety.

Notes

[1] Adil Afzal, Hafiz Siddiq Umer, Zeeshan Khan, and Muhammad Usman Ghani Khan. 2021. Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision. In *2021 International Conference on Artificial Intelligence (ICAI)*. DOI: 10.1109/ICAI52203.2021.9445206.

[2] Kartikeya Bhatt, Abhishek Joshi, Gaurav Singh, Vikrant Sharma, and Satvik Vats. 2024. Ai guardian on the road: harnessing machine learning for two-wheeler helmet detection. In *Proceedings of the 2024 International Conference on Computational Intelligence and Computer Applications (ICCICA)*. DOI: 10.1109/iccica60014.2024.10585188.

[3] Roopa R. Chandrika, N. Gowri Ganesh, A. Mummoorthy, and K. Karthick Raghunath. 2020. Vehicle detection and classification using image processing. In *Proceedings of the 2019 International Conference on Emerging Trends in Science and Engineering (ICESE)*. Vol. 1. IEEE, 1–6. DOI: 10.1109/ICESE46178.2019.9194678.

[4] Rajnandani Choubey, Saloni Sharma, Saloni Mourya, Aditya Narayan, and Dhruv Bhargav. 2025. Helmet and number plate detection by using yolo and opencv. https://www.researchgate.net/profile/Ijeasm-Journal/publication/388218838_Helmet_and_Number_plate_detection_by_using_yolo_and_open_cv/links/678f7986ec3ae3435a72778d/Helmet-and-Number-plate-detection-by-using-yolo-and-open-cv.pdf. (2025).

[5] Jorge E. Espinosa, Sergio A. Velastín, and John W. Branch. 2021. Detection of motorcycles in urban traffic using video analysis: a review. *IEEE Transactions on Intelligent Transportation Systems*, 22, 10, 6115–6130. DOI: 10.1109/TITS.2020.2997084.

[6] GeeksforGeeks. 2021. Yolo – you only look once (real-time object detection). Accessed April 22, 2025. (2021). https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/?ref=ml_lbp.

[7] Nemuel Norman F. Giron, Robert Kerwin C. Billones, Alexis M. Fillone, Jay Robert Del Rosario, Melvin K. Cabatuan, Argel A. Bandala, and Elmer P. Dadios. 2020. Motorcycle rider helmet detection for riding safety and compliance using convolutional neural networks. In *Proceedings of the IEEE Conference*. https://ieeexplore.ieee.org/document/9400149.

[8] Muhammad Hussain. 2024. Yolov5, yolov8 and yolov10: the go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988*. DOI: 10.48550/arXiv.2407.02988.

[9] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. 2022. A review of yolo algorithm developments. *Procedia Computer Science*, 199, 1066–1073. https://www.sciencedirect.com/science/article/pii/S1877050922001363.

[10] B. Karthika, M. Dharssinee, V. Reshma, R. Venkatesan, and R. Sujarani. 2024. Automated object detection and recognition using yolov8. In *Proceedings of the International Conference on Communication, Computing, and Networks Technologies (ICCCNT)*. IEEE. DOI: 10.1109/ICCCNT61001.2024.10724411.

[11] Kili Technology. 2023. Yolo algorithm: real-time object detection from a to z. Accessed April 29, 2025. (2023). https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z.

[12] Abhash Kumar, M. S. Guru Prasad, Dheeraj Soni, Vibha Rani, Kanika Arora, and D. R. Ganesh. 2024. Real-time helmet detection system with vehicle number extraction for two-wheeler vehicles using yolov8. In *Proceedings of the 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*. IEEE, 672–676. https://ieeexplore.ieee.org/document/10551102.

[13] Raman Kumar, Yogendra Kumar, Ahmed Alkhayyat, Lovneesh Sharma, Ankita Joshi, and Ayush Dogra. 2023. Vision-based safety identification and monitoring system using tensorflow. In *International Conference on Intelligent Systems Design and Applications*. Springer Nature Switzerland, 369–375. https://link.springer.com/chapter/10.1007/978-3-031-64650-8_36.

[14] Fadil Muhammad, Ismail Bintang, Rian Fahrizal, Ceri Ahendyarti, Romi Wiryadinata, and Imamul Muttakin. 2024. Real-time motorcyclist helmet detection using yolov8 on edge device. In *2024 International Conference on Informatics Electrical and Electronics (ICIEE)*. IEEE, 1–7. DOI: 10.1109/ICIEE63403.2024.10920426.

[15] Pavan Mutyala, Chandrika Gali, Jaidev Chivakula, Venkatesh Duggirala, and P. Padmaja. 2023. Automated helmet detection system using rt-detr for real-time monitoring of motorcyclist safety. *International Journal of Scientific Research in Engineering and Management*. DOI: 10.55041/IJSREM40150.

[16] K.W. Ong, S.L. Loh, and T.H. Cheong. 2022. Vehicle classification using neural networks and image processing. *International Journal of Electrical Engineering and Applied Sciences (IJEEAS)*, 5, 2. DOI: 10.54554/ijeeas.2022.5.02.006.

[17] Immidisetty V. Prakash and M. Palanivelan. 2024. A study of yolo (you only look once) to yolov8. In *Algorithms in Advanced Artificial Intelligence*. CRC Press, 257–266. DOI: 10.1201/9781003529231-40.

[18] Aleksander Radovan, Leo Mršić, Goran Đambić, and Branko Mihaljević. 2024. A review of passenger counting systems based on image processing and machine learning. *Proceedings of the International Conference on Transportation Engineering and Systems*. https://www.mdpi.com/2673-4117/5/4/172.

[19] William David Moreno Rendon, Carolina Burgos Anillo, Daniel Jaramillo-Ramirez, and Henry Carrillo. 2023. Passenger counting in mass public transport systems using computer vision and deep learning. *IEEE Latin America Transactions*, 21, 4, 537–545. DOI: 10.1109/TLA.2023.10128885.

[20] Mukaram Safaldin, Nizar Zaghden, and Mahmoud Mejdoub. 2024. An improved yolov8 to detect moving objects. *IEEE Access*, 12, 59782–59806. DOI: 10.1109/ACCESS.2024.3393835.

[21] S. Sanjana, S. Sanjana, V. R. Shriya, M. Sandeep, and R. Vaibhav. 2021. A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. *Evolutionary Intelligence*, 14, 979–987. DOI: 10.1007/s12065-020-00493-7.

[22] Kavuri K. S. V. A. Satheesh, Nandam Sai Akhila, Dondapati Amarnadh, Paruchuri Sagar Swetha, Avula Venkata Sohan, and Vasireddy Pardhiv. 2024. Automated helmet monitoring system using deep learning. *EPRA International Journal*. DOI: 10.36713/epra16328.

[23] L. Sharma. 2024. Vision-based safety identification and monitoring system using tensorflow. In *Intelligent Systems Design and Applications: Information and Network Security*. Vol. 3, 369. DOI: 10.1007/978-3-031-64650-8_36.

[24] Felix Wilhelm Siebert, Christoffer Riis, and Frederik Boe Hüttel. 2024. 365: urban motorcycle helmet use surveillance in southeast asia – a cost-efficient computer vision approach. *Injury Prevention*. DOI: 10.1136/injuryprev-2024-SAFETY.181.

[25] Surendra Singh. 2024. Visual vigilance: enhancing helmet recognition through computer vision. https://www.researchgate.net/publication/382495810. (2024).

[26] Elham Soltanikazemi, Ashwin Dhakal, Bijaya Hatuwal, and Imad Eddine Toubal. 2023. Real-time helmet violation detection in ai city challenge 2023 with genetic algorithm-enhanced yolov5. https://www.researchgate.net/publication/377236120. (2023).

[27] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. 2022. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *AI*, 5, 4, 83. https://www.mdpi.com/2504-4990/5/4/83.

[28] John Paul Q. Tomas and Bonifacio T. Doma. 2023. Motorcycle helmet detection and usage classification in the philippines using yolov5 algorithm. In *Proceedings of the ACM International Conference on Computing and Data Science (ICCDS 2023)*. DOI: 10.1145/3581792.3581796.

[29] Rejin Varghese and M. Sambath. 2024. Yolov8: a novel object detection algorithm with enhanced performance and robustness. In *Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE, (Apr. 2024), 1–6. DOI: 10.1109/ADICS58448.2024.10533619.

CHAPTER 3

METHODOLOGY

This chapter explained the methods used to develop the Helmet Compliance Detection Using Computer Vision for Safer Roads. It included data collection, prototype development using the YOLO algorithm, integration of added features, and testing to ensure the prototype performed well in real-time detection of traffic violations.

Research Design

Constructive research design involved the development of new prototypes or solutions based on existing knowledge and theories. This methodology enabled researchers to create practical, functional solutions that could be implemented in real-world scenarios. It focused on addressing real-world challenges by combining theoretical insights with practical applications. In fields such as technology and computer science, constructive research typically involved the creation of software or systems that enhanced or refined existing solutions, all while building on established principles to improve functionality and effectiveness [1].

This study used a constructive research design to develop a real-time helmet compliance detection prototype that enhanced road safety monitoring through the use of AI-powered technologies. This research design was appropriate for the study because it focused on building a functional and innovative prototype that integrated computer vision components to address the identified gaps in traffic law enforcement. This study developed an intelligent detection prototype that was capable of identifying motorcycle riders without helmets, improper helmet usage, overloading of passengers, and missing. The prototype used the YOLOv8 object detection algorithm for real-time identification and OpenCV for visual processing. It also triggered alerts and automatically recorded violations for documentation and enforcement purposes. The prototype was designed for use along Nabua Highway, Camarines Sur, and it aimed to function effectively even in varying lighting and weather conditions. By constructing and evaluating this prototype, the study contributed a practical and scalable solution to improve road safety compliance using modern AI techniques. Adopting constructive research allowed this study to develop a practical solution for improving road safety. In everyday life, many accidents happened because riders did not wear helmets. To address this issue, this study built a prototype that automatically checked if riders were wearing helmets using computer vision. The prototype used a detection algorithm to identify helmets in real time. Through testing and collecting more data, the prototype was improved to make it more accurate. The goal was to build a prototype that traffic authorities could use to check helmet compliance and improve road safety. This study helped to make the roads safer and could help prevent accidents and save lives.

Theorems, Algorithm and Mathematical Framework

In the field of computer vision, algorithms and mathematical models were important in developing systems for real-time object detection. This study used a YOLOv8-based approach to detect helmet usage, count motorcycle passengers, and recognize license plates. YOLO (You Only Look Once) was a single-stage object detection algorithm known for its speed and accuracy, making it suitable for deployment in real-time environments.

YOLOv8 Object Detection Algorithm

YOLOv8 was the latest version of the YOLO family of algorithms, designed for fast and accurate object detection. Unlike previous versions, YOLOv8 introduced an anchor-free architecture, improved feature extraction, and decoupled detection heads for classification and localization, making it more flexible and precise. According to Muhammad et al. [2024] YOLOv8 was used for real-time helmet detection in Indonesia, achieving a 91.1%

F1 score for helmet detection and 81.7 % accuracy for rider detection [2]. This study highlighted YOLOv8's effectiveness in real-world applications, emphasizing its potential for smart city integration and law enforcement, particularly in monitoring motorcyclist safety.

YOLOv8 worked by predicting bounding boxes and class probabilities directly from full images in one evaluation, treating detection as a regression problem. As illustrated in Figure 1, the algorithm followed a streamlined architecture composed of an input layer, backbone, neck, and prediction head, resulting in accurate and real-time object detection. It employed advanced loss functions, such as Complete Intersection over Union (CIoU), to improve bounding box accuracy. The algorithm also utilized Non-Maximum Suppression (NMS), which filtered overlapping bounding boxes and retained only the most confident predictions. Furthermore, YOLOv8 outputs were detected only when the confidence score exceeded a predefined threshold, reducing false positives.

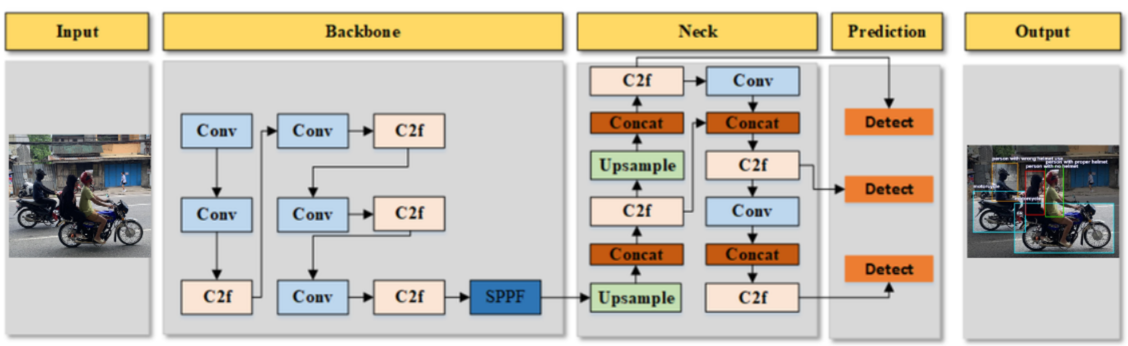


Figure 1: **Yolov8 Object Detection Architecture**

Figure 1 illustrated the YOLOv8 architecture, showing how the model processes images for object detection. The input image is first resized and normalized before passing through the backbone, which extracts features using convolutional layers and C2f blocks. Next, the neck refines and fuses multi-scale features through concatenation and upsampling, and finally, the decoupled prediction head produces separate outputs for classification and localization, generating bounding boxes and labels. This process demonstrates how YOLOv8 efficiently extracts detailed spatial information and distinguishes between object classes and locations. By separating classification from localization, the architecture improves prediction accuracy and reduces errors, making the system more reliable for real-time applications. The clear flow from input to output highlights the model's structured approach, ensuring that object detection is precise and systematic. Overall, the YOLOv8 architecture provides a robust framework for real-time detection, combining feature extraction, multi-scale fusion, and decoupled prediction to achieve accurate and efficient performance.

Detection Mechanism of Yolov8

Bounding Box Prediction

YOLOv8 predicted the center coordinates (x_{pred}, y_{pred}), width (w_{pred}), and height (h_{pred}) for each object within a grid cell.

The confidence score, used for evaluating bounding box accuracy, was given by the formula:

$$Confidence = P_{object} \times IOU_{pred,truth} \tag{3.1}$$

Class Probability Prediction

YOLOv8 outputs a probability distribution across multiple object classes. For each bounding box, the network predicted the likelihood that it belonged to a particular class (e.g., helmet, rider, license plate).

Complete Intersection over Union (CIoU)

To optimize bounding box predictions, YOLOv8 utilizes the Complete Intersection over Union (CIoU) loss function, which improved upon the standard IoU by considering not only the overlap area but also the distance between the center points of the predicted and ground truth boxes, as well as the consistency of their aspect ratios. This enhancement leads to more precise and reliable bounding box regression, resulting in improved overall performance for object detection tasks.

Intersection over Union (IoU), on the other hand, served as a fundamental evaluation metric for object detection models, as it measured the degree of overlap between predicted and ground truth bounding boxes. A higher IoU indicated more accurate localization, while lower values reflect poor alignment. The following figure illustrated how IoU is computed and highlighted its role in assessing detection accuracy, with emphasis on YOLOv8's refined approach.

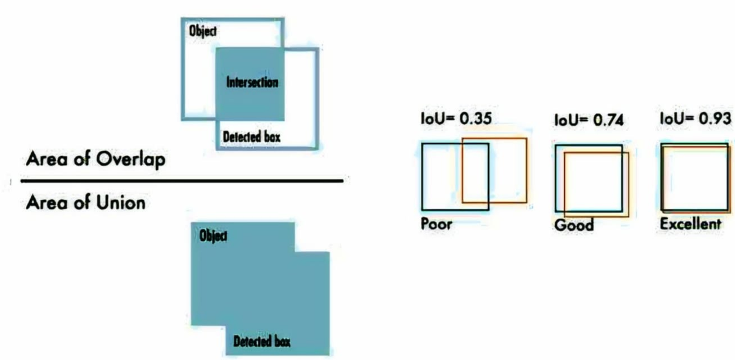


Figure 2: **Intersection over Union (IoU)**

As illustrated in Figure 2, IoU is computed by dividing the overlapping area of two boxes by their total combined area, indicating how closely the predicted box aligns with the ground truth. A higher IoU value signifies a better overlap between the predicted bounding box and the actual object. The figure further demonstrates this concept through three examples showing IoU values of 0.35, 0.74, and 0.93, corresponding to poor, good, and excellent box alignment, respectively. An IoU closer to 1 means the model has predicted the object's location with high precision, while lower values indicate weaker performance. [3]

Non-Maximum Suppression (NMS)

YOLOv8 employed Non-Maximum Suppression (NMS) to efficiently eliminate redundant bounding boxes that predicted the same object. After the model generated multiple bounding boxes, NMS ranked them according to their confidence scores, identifying how likely each box contained an object. The algorithm then selected the highest-scoring box and suppressed any overlapping boxes whose Intersection over Union (IoU) with the selected box exceeded a predefined threshold. This process ensured that each detected object was represented by only one bounding box, reducing clutter and improving the clarity of detection results. The following figure demonstrated how NMS improved detection clarity and reduced overlap.

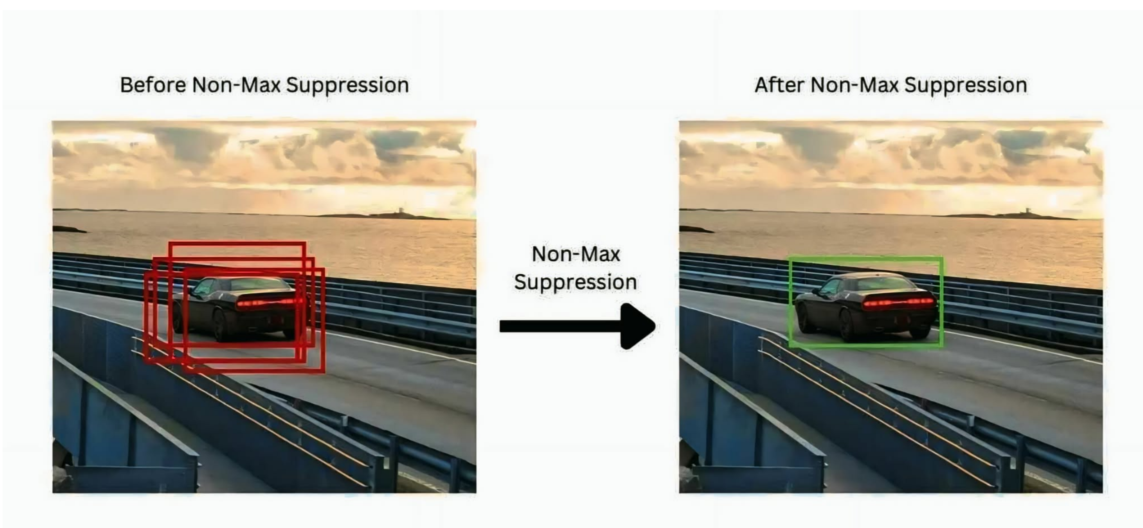


Figure 3: **Example of Non-Maximum Suppression**

Figure 3 effectively highlighted the importance of Non-Maximum Suppression (NMS) in enhancing the quality of object detection results produced by YOLOv8. Without NMS, as shown in the left image, the model outputted numerous overlapping bounding boxes for a single object, which could compromise the interpretability of the results and the accuracy of object localization. The right image, after NMS was applied, displayed a single, high-confidence bounding box, illustrating the algorithm's ability to reduce redundancy and improve detection clarity. This reduction in noise not only improved precision but also lowered computational load during post-processing, making the system more efficient.

In our study, the use of NMS was particularly important in refining predictions in complex scenes involving multiple or closely spaced objects such as identifying several riders or helmets in traffic scenarios. The figure provided clear visual evidence of how NMS strengthened both the robustness and operational efficiency of the YOLOv8-based detection pipeline [4].

This process improves precision and lowers computational load during post-processing, increasing the system's efficiency. NMS is particularly useful in complex scenarios with multiple or closely spaced objects, such as detecting several riders or helmets in traffic. By filtering overlapping boxes, the system produces cleaner outputs and more reliable detections, enhancing both robustness and operational performance. Overall, NMS is a critical component of the YOLOv8 pipeline, ensuring accurate, efficient, and interpretable detection results in real-world traffic monitoring applications.

Materials and Statistical Tools/Evaluation Methods

This section provided detailed information on the materials, statistical tools, and evaluation methods employed in the development and assessment of the Helmet Compliance Detection Using Computer Vision for Safer Roads. It included the hardware and software components, the process followed to implement the prototype, the sampling technique, the statistical tests used for performance evaluation, and the methods employed for evaluating the prototype's effectiveness.

The section covered the hardware, software, and tools used in developing the prototype, along with the workflow for implementation, data sampling methods, and evaluation. It highlighted the use of statistical techniques such as accuracy, confusion matrices, and error rate analysis, and explained how effectiveness was assessed through detection accuracy, response time, and reliability under real-world conditions.

Instrument

The research tool used by the researchers to carry out the study was described in this section, including its design, purpose, and role in collecting and analyzing data for the evaluation of the Helmet Compliance Detection prototype. It also explained how the tool was applied during testing, the parameters it measured, and the way it supported the assessment of the system's accuracy, reliability, and effectiveness in real-world conditions.

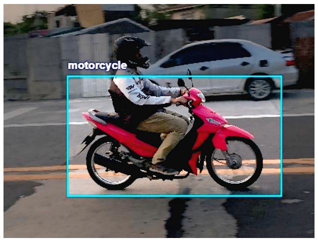
Dataset

The dataset used in this study was custom-created and annotated by the researchers to support helmet compliance monitoring and accurate motorcycle detection using YOLOv8. It contains five classes: Motorcycle, Not Motorcycle (for vehicle filtering), Person with No Helmet, Person with Proper Helmet, and Person with Wrong Helmet Use. Passenger counting was handled directly in the system's code by detecting the number of riders per motorcycle and flagging violations if more than two were present. A total of 1,242 images were collected from diverse real-world scenarios to ensure the trained model could reliably differentiate between helmet compliance cases while maintaining accurate motorcycle and non-motorcycle classification. The dataset was split into training (993 images), validation (125 images), and testing (124 images) to optimize model performance and evaluation.

Table 1
Distribution of Images and Instances per Class

Class	Images	Instances
Motorcycle	172	205
Not Motorcycle	55	93
Person with No Helmet	52	73
Person with Proper Helmet	119	168
Person with Wrong Helmet Use	61	85
Total	**459**	**624**

The table summarizes the distribution of images and instances across the dataset used for training the helmet compliance detection model. While the dataset contains 459 unique images, the number of instances is higher at 624 because a single image may include multiple objects and multiple classes. Classes such as Motorcycle and Person with Proper Helmet have higher instance counts, indicating the presence of crowded traffic scenes where multiple riders and vehicles appear simultaneously. In contrast, classes associated with helmet violations contain fewer images and instances, reflecting the relative scarcity of these events in real-world data collection. This imbalance can influence the model's learning process, potentially leading to variations in detection performance across classes. As a result, evaluation metrics such as precision, recall, and F1-score are more appropriate than overall accuracy for assessing model effectiveness. These dataset characteristics provide context for interpreting the model's performance trends discussed in the subsequent sections.



(a) **Motorcycle**

(b) **Person with no helmet**

(c) **Person with wrong helmet use**

(d) **Person with proper helmet**

(e) **Not Motorcycle**

Figure 4: **Samples of Dataset: Motorcycle, person with no helmet, person with wrong helmet use, person with proper helmet and not motorcycle.**

The image above presents the Helmet Compliance Detection Using Computer Vision prototype, which was trained on a unified dataset containing four major classes that support its detection capabilities. These classes include Motorcycle, Person with No Helmet, Person with Wrong Helmet Use, and Person with Proper Helmet, allowing the system to accurately recognize motorcycles and evaluate helmet compliance among riders. Together, these classes enable the prototype to detect motorcycles, identify helmet violations, and assess overloading behavior by analyzing multiple riders in a single scene. This structure ensures that the system can analyze several safety-related factors simultaneously, improving its effectiveness in real-world monitoring. To further enhance detection accuracy, the prototype also uses a separate Vehicle Filtering Dataset containing a Not Motorcycle class, ensuring that non-motorcycle vehicles are filtered out before performing helmet and passenger analysis. Overall, the integration of these datasets strengthens the model's reliability and supports a more efficient and precise detection process for road safety enforcement.

Procedure / Process

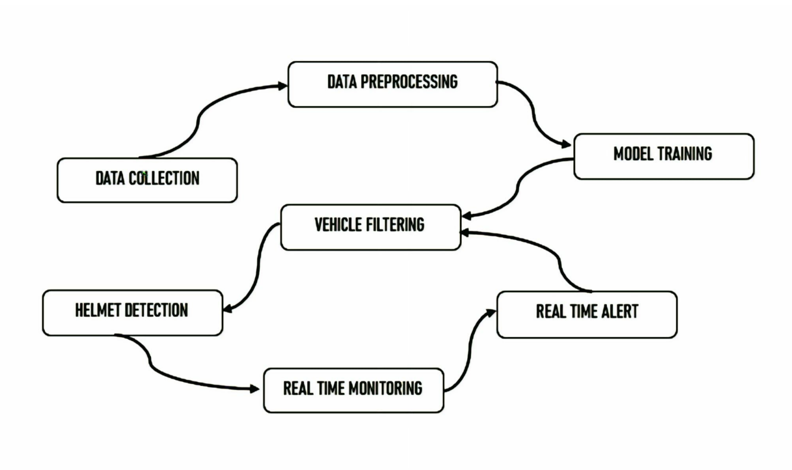


Figure 5: **Prototype of Flowchart**

The diagram illustrated the real-time workflow of the Helmet Compliance Detection System, showing how each stage contributes to continuous monitoring and violation detection. The process began with video data collection and preprocessing, where incoming frames were prepared for analysis, followed by the application of a trained YOLOv8 model designed to detect helmets worn by motorcycle riders and passengers. This setup demonstrates how the system transitions from raw video input to intelligent detection, highlighting the model's role in identifying compliance and violations accurately. This workflow is important because it helps the system focus only on motorcycles, making the detection faster and more accurate. After filtering, the system evaluated helmet compliance, triggered alerts when violations were detected, and automatically saved short video clips as evidence, enabling clear documentation for enforcement and review. Overall, the continuous looping of the detection process ensures uninterrupted monitoring, making the system effective for supporting road safety initiatives and strengthening helmet law enforcement through automated, real-time analysis.

Data Collection

Video footage is live along the Nabua Highway under various traffic and lighting conditions to capture real-world motorcycle scenarios. These videos serve as the primary input for detecting helmet usage.

Data Preprocessing

The collected data were processed using OpenCV to resize frames, enhance image quality, and normalize the input. This step ensures that the data is clean and ready for analysis by the detection model.

Model Training

The YOLOv8 model was trained using labeled data to identify whether the rider and passenger are wearing helmets. The model was tested with a separate dataset to ensure its accuracy in helmet detection.

Vehicle Filtering

As vehicles passed through the camera, the system first filtered and detected whether the vehicle was a motorcycle. Only motorcycles were analyzed for helmet compliance.

Helmet Detection

Once a target vehicle was identified as a motorcycle, the system proceeded to detect whether the rider was wearing a helmet. If a passenger was also detected, the system checked whether the passenger was wearing a helmet, holding one, or using it incorrectly.

Real-time Monitoring

All detection processes ran in real time, ensuring continuous monitoring and immediate feedback on helmet law compliance.

Real-time Violation Alert

If any person on the motorcycle (rider or passenger) was detected without a helmet or using a helmet incorrectly, a real-time violation alert popped up on the monitoring screen. This allowed for immediate awareness and potential enforcement.

Normalized Value

The normalized value is used to standardize the raw user feedback scores, transforming them into a range between 0 and 1. The formula for normalization is:

$$\text{Normalized Value} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{3.2}$$

Where:

- X is the raw score obtained from user feedback.
- $\min(X)$ is the minimum possible value (typically 0.0).
- $\max(X)$ is the maximum possible value (typically 1.0).

After applying the formula, the responses were converted into normalized values, which were then mapped to the ranges presented in Table 3, allowing each response to be categorized according to defined satisfaction levels. This process ensures that the feedback is organized systematically and can be interpreted clearly. Normalization makes the responses consistent and comparable across different components of the prototype, preventing discrepancies that may arise from varying input values. The categorized results provide a clear understanding of satisfaction, highlighting which aspects of the prototype perform well and which areas may require improvement or refinement. This structured approach supports informed evaluation, allowing developers and researchers to identify priorities for enhancement. Overall, the normalization and categorization process not only allows for accurate and reliable assessment of the prototype's effectiveness but also serves as a valuable guide for future improvements, ensuring that refinements are focused on areas that need the most attention.

Evaluation Method

To evaluate the performance of the proposed Helmet Compliance Detection Using Computer Vision for Safer Roads, several standard evaluation metrics were utilized. These metrics assess the system's accuracy in detecting helmet usage, counting passengers, and identifying violations in real time. The evaluation was based on comparing the model's predictions against manually annotated ground truth data using test video segments.

Accuracy

Accuracy measured the overall effectiveness of the prototype by calculating the percentage of correctly identified objects (motorcycles, helmets and overloading violations) as well as correctly filtered non-motorcycle vehicles. It provides a general view of the system's performance across all detection tasks.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

Where:

- **TP (True Positives):** Instances where motorcycles, helmets, passengers, overloading violations, or non-motorcycles were correctly detected or filtered.

- **TN (True Negatives):** Instances where non-violations or non-motorcycle objects were correctly ignored.
- **FP (False Positives):** Instances where incorrect objects were detected, such as misclassifying a non-motorcycle as a motorcycle or falsely detecting a helmet violation.
- **FN (False Negatives):** Instances where motorcycles, helmets, passengers, overloading violations, or non-motorcycles were missed.
- **Precision:** Precision measures the ability of the system to correctly identify positive cases out of all instances that the system marked as positive.

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

- **Recall:** Recall evaluates the ability of the system to detect all actual positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

- **F1-Score:** The F1-Score is the harmonic mean of precision and recall.

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.6}$$

Mean Average Precision (mAP)

Mean Average Precision (mAP) evaluates both the precision and localization accuracy of the predicted bounding boxes across all object classes.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3.7}$$

Where:

- N = Number of object classes
- AP_i = Average Precision for the i-th class

Frames Per Second (FPS)

FPS measures how fast the system processes video frames.

$$FPS = \frac{\text{Number of Processed Frames}}{\text{Time Taken (in seconds)}} \tag{3.8}$$

Table 2 presents the Confusion Matrix used to evaluate the performance of the Helmet Compliance Detection Model. The matrix summarizes the relationship between actual and predicted classifications using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. High TP and TN counts indicate that the model accurately detects helmet use and non-use, while lower FP and FN values demonstrate strong reliability and precision. These results align with the model's training accuracy, confirming the YOLOv8's effectiveness in distinguishing helmet compliance among riders. However, minor misclassifications may occur under challenging conditions such as motion blur or low lighting. Overall, Table 2 validates that the model performs efficiently and reliably in classifying helmet compliance for real-time monitoring applications.

$$FPS = \frac{\text{Number of Processed Frames}}{\text{Time Taken (in seconds)}} \tag{3.9}$$

Equation 3.8 defines how Frames Per Second (FPS) is computed to measure the real-time processing speed of the detection system. The FPS value is determined by dividing the number of processed frames by the total processing time, reflecting the system's capability to handle live video feeds efficiently. A higher FPS value signifies smoother and faster detection, ensuring prompt alert generation for helmet violations. This metric supports the model's suitability for continuous monitoring and live deployment. However, performance may slightly decrease on devices lacking GPU acceleration, resulting in reduced FPS. In summary, Equation 3.8 confirms the system's ability to operate in real time while maintaining accurate and consistent detection performance.

Theoretical Framework

This section outlined the theoretical underpinnings that guided the development of the Helmet Compliance Detection Prototype using computer vision. The framework integrated four key theories: Computer Vision Theory, Automated Law Enforcement Theory, Surveillance Theory, and Real-Time Embedded Systems Theory.

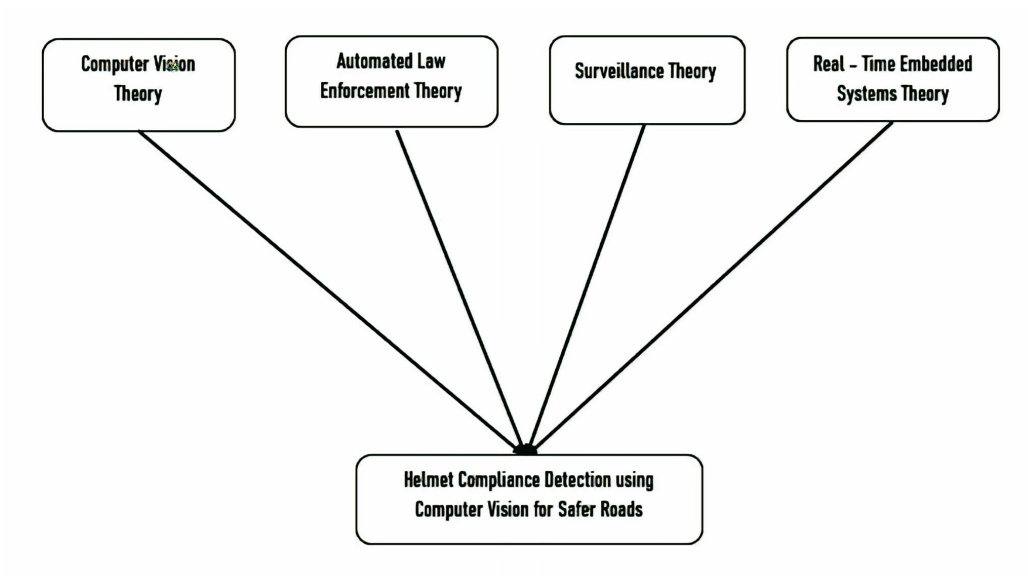


Figure 6: **Theoretical Framework of the Helmet Compliance Detection**

Figure 6 presented a simplified visual representation of the theoretical framework that guided the development of the Helmet Compliance Detection Prototype. The framework was built upon four foundational theories: Computer Vision Theory, Automated Law Enforcement Theory, Surveillance Theory, and Real-Time Embedded Systems Theory, each of which contributed specific principles to the system's design. These theories collectively show how technical and sociological aspects were integrated to enhance detection accuracy, operational efficiency, and real-time performance. By grounding the system in established theoretical concepts, the framework ensures that the prototype is not only technically effective but also socially relevant, supporting lawful and responsible enforcement. The connection between the theories and the framework highlights the importance of a multidisciplinary approach, combining engineering, computer science, and social science to guide the system's development. Overall, the theoretical framework provides a strong foundation for the prototype, demonstrating how combining multiple perspectives can inform design decisions, improve functionality, and ensure that the system meets both technical and societal objectives.

Computer Vision Theory

This theory provided the foundation for interpreting and processing visual inputs (video or image data) to extract meaningful patterns. In this prototype, YOLOv8 was applied to enable real-time detection of motorcycles, riders' helmet compliance, and overloading violations.

Automated Law Enforcement Theory

This theory emphasized the role of intelligent systems in supporting or replacing human roles in enforcing regulations. In the context of traffic compliance, the integration of technologies such as OpenCV and YOLOv8 aligned with the principles of automation for more accurate, consistent, and scalable monitoring.

Surveillance Theory

Surveillance Theory explained the sociotechnical importance of systematically observing and recording behaviors to ensure safety and rule compliance. This theory justified the deployment of camera-based monitoring systems in public spaces to detect and deter traffic violations, promoting accountability and public safety.

Real-Time Embedded Systems Theory

This theory supported the technical design of prototypes that processed data and responded within strict time constraints. It underpinned the implementation of real-time detection features in the system, enabling low-latency processing of live video feeds through optimized algorithms and embedded computing environments.

Notes

[1] Casper Lassenius, Timo Soininen, and Jari Vanhanen. 2001. Constructive research. In *Proceedings of the 26th Information Systems Research Seminar in Scandinavia (IRIS 26)*. Accessed: 2025-04-22. https://www.pm.lth.se/fileadmin/_migrated/content_uploads/3._constructive_research.pdf.

[2] Fadil Muhammad, Ismail Bintang, Rian Fahrizal, Ceri Ahendyarti, Romi Wiryadinata, and Imamul Muttakin. 2024. Real-time motorcyclist helmet detection using yolov8 on edge device. In *Proceedings of the 2024 International Conference on Informatics Electrical and Electronics (ICIEE)*. IEEE, (Dec. 2024). DOI: 10.1109/ICIEE63403.2024.10920426.

[3] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. 2023. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5, 4, (Nov. 2023), 1680–1716. DOI: 10.3390/make5040083.

[4] ThePythonCode. 2021. Non-maximum suppression using opencv in python. https://thepythoncode.com/article/non-maximum-suppression-using-opencv-in-python. Accessed: 2025-04-22. (2021).

CHAPTER 4
RESULTS AND DISCUSSION

The data gathered during the study are presented and evaluated in this chapter. A discussion and thorough analysis of helmet compliance, wrong helmet use, and motorcycle overloading are also included in this chapter.

Data Collection

The first process in developing the Helmet Compliance Detection System was data gathering, which served as the foundation for all subsequent stages of model development. This phase involved systematically collecting, preparing, and organizing image data required for training the deep learning model. The researchers aimed to assemble a comprehensive, diverse, and well-balanced dataset that accurately reflected real-world motorcycle riding conditions. To achieve this, images were sourced from various environments, including highways, urban streets, and low-light areas, ensuring that the dataset captured a wide range of scenarios such as different weather conditions, traffic densities, rider positions, and helmet types.

In addition to collecting raw images, the data gathering stage also included filtering out low-quality or irrelevant images, labeling each sample according to predefined categories, and ensuring that each category contained a sufficient number of representative examples. This careful preparation was essential for improving model performance and reducing the risk of biased or inconsistent predictions. Figure 7 below presents sample images of the five identified categories used in the study, illustrating the diversity of motorcycle riding scenarios incorporated into the dataset.

(a) **motorcycle**

(b) **not motorcycle**

(c) **person with no helmet**

(d) **person with proper helmet**

(e) **person with wrong helmet use**

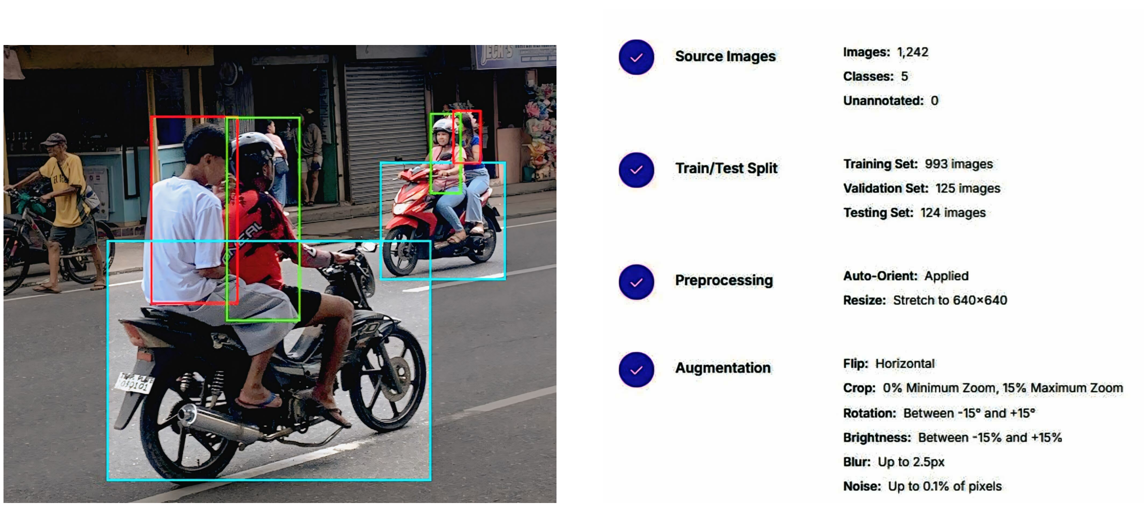
Figure 7: **Examples of Dataset Classes for Helmet Compliance Detection.**

As shown in Figure 7, the researchers successfully compiled a dataset that consisted of 1,242 labeled images, categorized into five classes: motorcycle, non-motorcycle, person with no helmet, person with proper helmet, and person with wrong helmet use. The dataset covered different environmental settings to ensure varied representation. Since images of "wrong helmet use" were difficult to obtain, the researchers captured original photos and supplemented them with online sources to achieve balance among all categories.

The findings indicated that dataset quality and diversity were vital for developing an accurate helmet compliance detection model. Images captured under different lighting, angles, and scenarios improved generalization and reduced bias. Careful annotation and image processing ensured consistency and reliability prior to model training. The comprehensive dataset allowed the model to adapt to realistic traffic conditions, supporting effective detection in real-world applications. Data augmentation techniques such as flipping, rotation, and brightness adjustment enhanced robustness against environmental variations. Despite its adequacy for training, the dataset's limited size and lack of nighttime images presented minor constraints. Future work is recommended to expand image collection under diverse lighting and weather conditions to further improve model adaptability and performance. Overall, the dataset played a critical role in building a reliable Helmet Compliance Detection Prototype, providing sufficient quality, balance, and realism for accurate model training and evaluation.

Data Preprocessing

Following the data-gathering phase, the next procedure involved data preprocessing, which ensured that all images were properly formatted, annotated, and enhanced for model training. This process was carried out using Roboflow, a platform that provided tools for annotation, preprocessing, and augmentation. The preprocessing workflow performed on the dataset, including image organization, data splitting, and transformation steps.



(a) **Annotating the datasets** (b) **Generating the datasets**

Figure 8: **Data Pre-processing**

As illustrated in Figure 8, the preparation of the dataset in Roboflow involved two main stages. Image (a) displays the annotation process, where each image was manually labeled and categorized into five classes: motorcycle, non-motorcycle, person with no helmet, person with proper helmet, and person with wrong helmet use. This ensured that all images were properly classified and ready for model training. Meanwhile, Image (b) illustrates the generation of the dataset, which included resizing all images to 640×640 pixels and applying several enhancement techniques such as rotation, flipping, brightness adjustment, and cropping. These steps improved image consistency and diversity, allowing the YOLOv8 model to perform effectively across different lighting conditions, camera angles, and motion scenarios.

A total of 1,242 images were collected and processed in Roboflow. Each image was uploaded, labeled, and organized into the five aforementioned categories, ensuring data completeness and proper structure. To support accurate model training and evaluation, the dataset was divided into 993 images for training, 125 for validation, and 124 for testing. This distribution allowed the model to learn from the majority of the images while its accuracy was assessed on separate, unseen subsets. This stage highlighted the importance of maintaining consistency in image preparation to achieve reliable deep learning results. Standardizing image dimensions and applying controlled augmentations enhanced the model's adaptability to various real-world motorcycle riding conditions. Although the dataset remained smaller than large public collections, these measures ensured that the data were well-organized, sufficiently varied, and suitable for effective model training.

Model Training

This section discusses the training and evaluation of the YOLOv8 model developed for helmet compliance detection. It presents the model's learning process and performance based on various evaluation metrics and visual results. Collectively, the figures and tables comprehensively summarize the model's development, learning behavior, and overall detection performance across multiple real-world testing conditions, ensuring robust, accurate, and efficient helmet detection outcomes.

Table 2
YOLOv8 Training Results

Epoch	Time	Train/Box_Loss	Train/Cls_Loss	Train/Dfl_Loss	Precision(B)	Recall(B)	mAP50(B)	mAP50-95(B)	Val/Box_Loss	Val/Cls_Loss	Val/Dfl_Loss	Lr/Pg0	Lr/Pg1	Lr/Pg2
126	6360.51	0.75791	0.45379	1.05759	0.80359	0.80185	0.81238	0.45752	1.49237	0.71563	1.65138	0.000194425	0.000194425	0.000194425
127	6412.57	0.74665	0.4505	1.05552	0.82939	0.77792	0.81233	0.45731	1.48609	0.73031	1.6396	0.000187092	0.000187092	0.000187092
128	6464.33	0.75166	0.45257	1.05661	0.79837	0.80707	0.81184	0.45776	1.49981	0.72916	1.64966	0.00017976	0.00017976	0.00017976
129	6515.65	0.74331	0.4458	1.05266	0.78773	0.80445	0.80437	0.4522	1.50162	0.71369	1.66705	0.000172427	0.000172427	0.000172427
130	6567.16	0.74332	0.4428	1.04925	0.83522	0.75742	0.80926	0.45308	1.5077	0.7282	1.66317	0.000165095	0.000165095	0.000165095
131	6617.98	0.73888	0.4422	1.047	0.79833	0.80173	0.80099	0.45012	1.50439	0.73274	1.66988	0.000157762	0.000157762	0.000157762
132	6669.68	0.74697	0.4498	1.05269	0.78536	0.81475	0.79961	0.45236	1.50791	0.73883	1.68036	0.000150429	0.000150429	0.000150429
133	6724.6	0.73601	0.43509	1.04564	0.80231	0.805	0.80724	0.45385	1.49944	0.72958	1.67922	0.000143097	0.000143097	0.000143097
134	6776.04	0.73314	0.43905	1.04535	0.827	0.79017	0.80382	0.45241	1.49828	0.73196	1.6699	0.000135764	0.000135764	0.000135764
135	6827.46	0.73802	0.44291	1.0466	0.83094	0.77603	0.80317	0.44906	1.5107	0.73402	1.69538	0.000128432	0.000128432	0.000128432
136	6879.04	0.74193	0.44178	1.05257	0.83278	0.77339	0.79924	0.44867	1.50778	0.72286	1.69688	0.000121099	0.000121099	0.000121099
137	6930.26	0.73002	0.43664	1.04419	0.79711	0.78305	0.80078	0.45237	1.50962	0.7192	1.685	0.000113767	0.000113767	0.000113767
138	6984.09	0.72548	0.43668	1.04603	0.78357	0.7936	0.79602	0.44868	1.50591	0.72574	1.67841	9.91012e-05	9.91012e-05	9.91012e-05
139	7036.32	0.72771	0.43275	1.04649	0.82369	0.78635	0.80699	0.45486	1.49677	0.72329	1.67684	9.17686e-05	9.17686e-05	9.17686e-05
140	7087.92	0.72609	0.43721	1.04344	0.82096	0.79007	0.80784	0.45777	1.50444	0.72084	1.67684	8.4436e-05	8.4436e-05	8.4436e-05
141	7141.32	0.68102	0.35303	1.00733	0.82935	0.77311	0.803	0.44489	1.5151	0.7147	1.70039	6.24382e-05	6.24382e-05	6.24382e-05
142	7191.72	0.66473	0.33836	0.99444	0.81125	0.79118	0.80295	0.44949	1.5046	0.71617	1.70118	5.51056e-05	5.51056e-05	5.51056e-05
143	7241.58	0.64573	0.33418	0.9918	0.8248	0.78388	0.80395	0.45306	1.50615	0.72032	1.71166	4.77732e-05	4.77732e-05	4.77732e-05
144	7292.55	0.64122	0.33231	0.98752	0.79349	0.78769	0.79914	0.45306	1.50419	0.71693	1.70198	4.04408e-05	4.04408e-05	4.04408e-05
145	7340.88	0.63183	0.3316	0.98424	0.82966	0.76569	0.79915	0.45252	1.49896	0.72323	1.70198	3.31078e-05	3.31078e-05	3.31078e-05
146	7391.35	0.63669	0.3314	0.98866	0.84181	0.77041	0.80284	0.45321	1.50939	0.72793	1.70039	2.57752e-05	2.57752e-05	2.57752e-05
147	7439.61	0.63068	0.32582	0.98083	0.81884	0.80224	0.80492	0.45482	1.50247	0.72536	1.71493	1.84426e-05	1.84426e-05	1.84426e-05
148	7490.37	0.62258	0.32381	0.98083	0.80132	0.80224	0.80492	0.45482	1.50247	0.72536	1.71493	1.84426e-05	1.84426e-05	1.84426e-05
149	7538.48	0.61945	0.32406	0.97695	0.82539	0.77995	0.80342	0.45361	1.50333	0.7281	1.7064	1.84426e-05	1.84426e-05	1.84426e-05
150	7587.08	0.62007	0.32603	0.97975	0.80859	0.78988	0.80107	0.45456	1.49947	0.72524	1.7064	1.84426e-05	1.84426e-05	1.84426e-05

Table 3 presents the YOLOv8 training results showing various performance metrics across multiple epochs. The parameters include loss components such as *Box Loss*, *Classification Loss*, and *Distribution Focal Loss*, together with precision, recall, and mean Average Precision (mAP) values. The table also displays validation losses and learning rate parameters, reflecting the model's optimization behavior during the training process. The table summarizes the learning progress and convergence performance of the YOLOv8 model during training. By observing these metrics, the researchers were able to determine how effectively the model minimized losses while improving detection accuracy over time.

As the training progressed, the loss values (*Train/Box*, *Train/Cls*, and *Train/Dfl*) gradually decreased, while performance indicators such as precision and recall stabilized at higher levels. The improvement in *mAP@50* and *mAP@50–95* shows that the model achieved high detection accuracy, while the reduction in validation losses confirms effective generalization and minimal overfitting. The results imply that the YOLOv8 model efficiently learned the distinctive features of helmets and riders, making it reliable for real-time detection. This demonstrates that the trained model can contribute to accurate helmet compliance monitoring and promote road safety initiatives.

Despite the positive outcomes, slight fluctuations in some loss values suggest that further fine-tuning or extended epochs may still enhance model stability. Variations in lighting or image quality may have also influenced certain performance measures. Overall, the YOLOv8 training results demonstrate a stable and effective learning process, achieving high accuracy and reliable detection performance. These findings confirm that the model is well-trained and suitable for implementation in helmet compliance detection systems.

Table 3

YOLOv8 Validation Results Showing Detection Accuracy per Class

Class	Images	Instances	Precision (P)	Recall (R)	mAP50	mAP50–95
All	227	624	0.804	0.795	0.831	0.459
Motorcycle	172	205	0.970	0.956	0.976	0.669
Not Motorcycle	55	93	0.911	0.978	0.958	0.778
Person with No Helmet	52	73	0.699	0.671	0.671	0.260
Person with Proper Helmet	119	168	0.662	0.676	0.653	0.251
Person with Wrong Helmet Use	61	85	0.776	0.691	0.740	0.337
Mean Precision:		0.8037				
Mean Recall:		0.7945				
mAP50:		0.8095				
mAP[0.5–0.95]:		0.4589				

After completing the training phase, the model was evaluated to determine its accuracy in detecting motorcycles and assessing helmet usage. This evaluation provided valuable insights into the YOLOv8 model's ability to correctly identify each class within the dataset and measure its effectiveness in real-world detection scenarios. As shown in Figure 10, the YOLOv8 model was trained using a dataset consisting of five classes: motorcycle, not motorcycle, person with no helmet, person with proper helmet, and person with wrong helmet use. The dataset was divided into training, validation, and testing sets to ensure fair and balanced performance assessment. Before training, the images were resized and normalized to match the input requirements of the model. The training process ran for 150 epochs using batch processing, which enabled the model to efficiently learn from multiple images simultaneously. During evaluation, the model achieved high precision and recall in detecting motorcycles, while helmet-related classes, specifically person with no helmet, person with proper helmet, and person with wrong helmet use demonstrated moderate but consistent performance. The overall mean precision and recall reached approximately 80

These results indicated that the YOLOv8 model learned effectively during training. The steady decrease in box, classification, and DFL losses reflected continuous improvement throughout the learning process. Precision and recall values stabilizing near 0.8 showed that the model accurately detected most target objects while minimizing false positives. However, the moderate scores for helmet-related categories suggested that detecting helmet violations remained challenging due to visual similarities, partial occlusions, and varying rider positions. The findings demonstrated that the developed model could support real-time helmet compliance monitoring. With accuracy levels around 80%, the system showed potential for deployment in local traffic enforcement and safety management. The model's consistency across multiple categories also suggested strong adaptability for future integrations into surveillance systems. Further optimization such as expanding the dataset or fine-tuning the model parameters could improve its ability to recognize subtle helmet violations more precisely. Although the model achieved strong performance overall, certain challenges were observed. The dataset contained fewer samples for specific classes, particularly wrong helmet use, which limited the model's learning depth. Additionally, lower precision under stricter evaluation thresholds, reflected by the mAP@[0.5:0.95] score of 0.46, indicated the need for enhanced dataset diversity and improved feature extraction to capture complex helmet patterns.

In conclusion, the YOLOv8 model performed effectively in detecting motorcycles and assessing helmet compliance. Its high precision and recall confirmed its reliability as a detection framework, while the moderate mAP scores highlighted opportunities for refinement in identifying specific helmet violations. The evaluation verified that the model successfully met its intended objectives and provided a strong basis for developing an automated helmet compliance detection system.

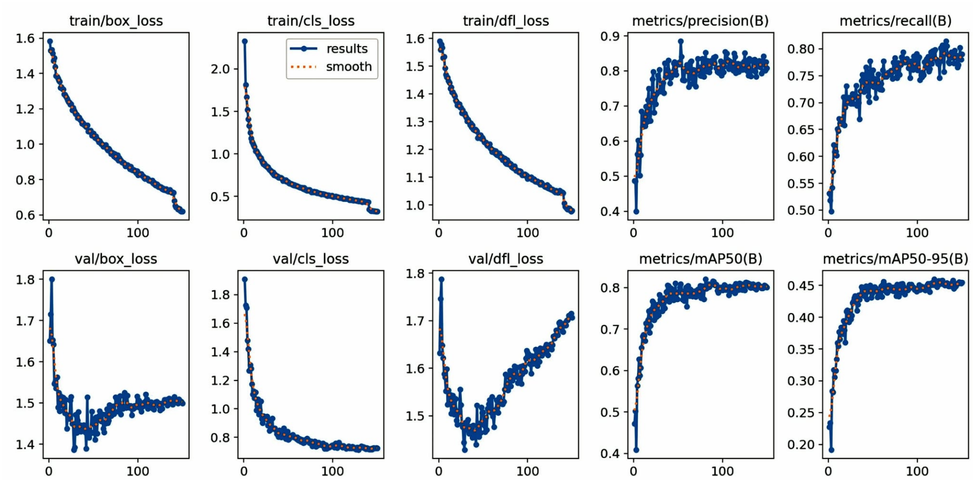


Figure 9: **Training Results**

As shown in Figure 9, the YOLOv8 model effectively learned throughout the training process. The box, classification, and DFL losses decreased consistently, indicating proper learning and convergence. Both precision and recall stabilized around 0.8, signifying reliable detection and classification accuracy. The model achieved approximately 0.8 mAP@50 and 0.45 mAP@[0.5:0.95], reflecting strong accuracy under standard evaluation but lower performance under stricter thresholds. Minor signs of overfitting were observed in validation, though the model remained stable and accurate overall. The steady reduction in loss values confirmed that the model successfully adapted to the dataset during training. The balance between precision and recall suggested that the YOLOv8 model was neither over-detecting nor missing significant objects. The slight overfitting may have resulted from the limited dataset size, where the model learned specific visual patterns too precisely.

These findings demonstrated that the YOLOv8 model possessed the learning capability required for real-world traffic applications. The results suggested its potential for robust and consistent helmet compliance detection once deployed in an operational setting. Although the model achieved stable accuracy, performance decreased under stricter thresholds (mAP@[0.5:0.95] = 0.45). This suggested that additional tuning or larger datasets could help enhance performance for more complex scenes. In summary, the training outcomes confirmed effective model learning, consistent accuracy, and readiness for further evaluation and deployment.

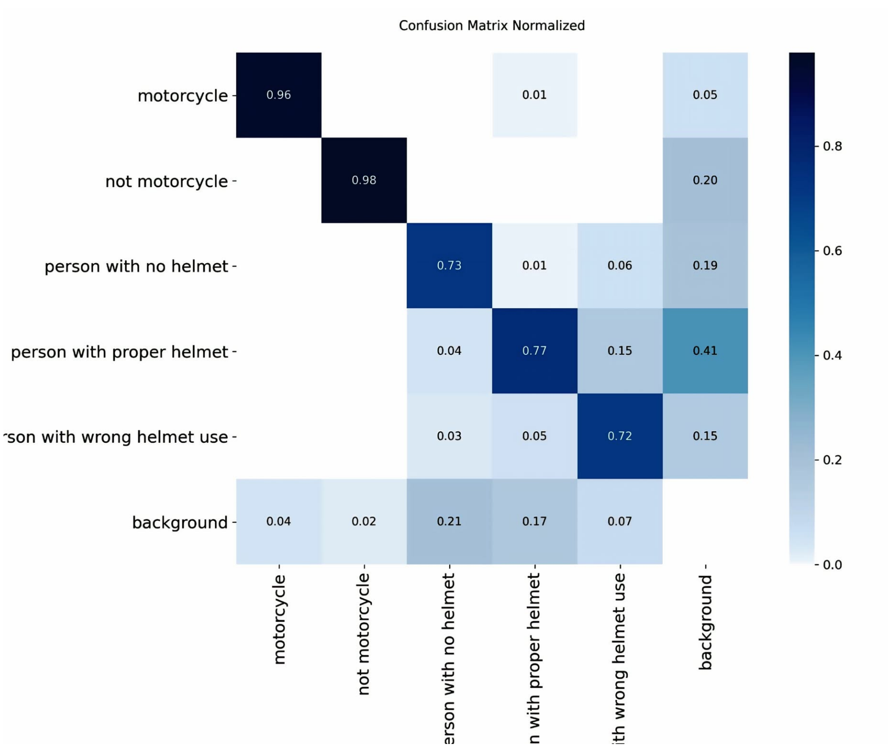
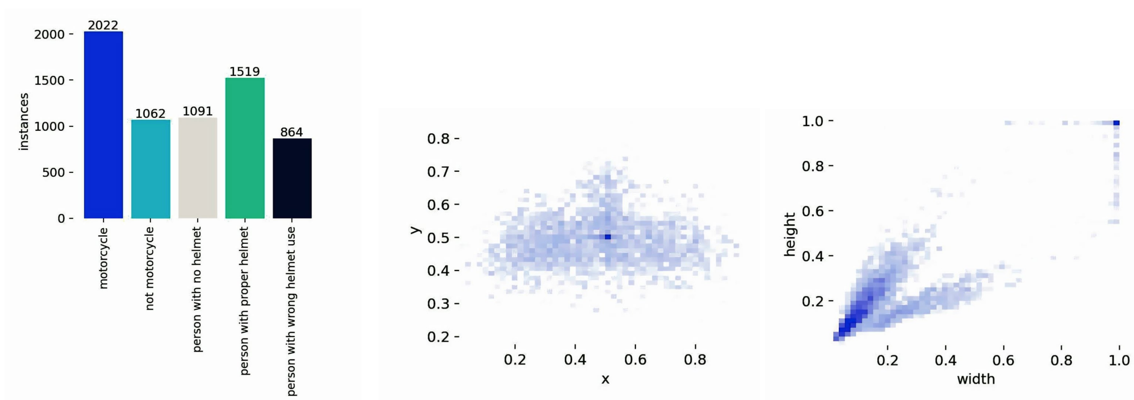


Figure 10: **Confusion Matrix**

As presented in Figure 10, the confusion matrix revealed high accuracy in identifying motorcycles (96%) and non-motorcycles (98%), while helmet-related categories achieved moderate accuracy: 73% for no helmet, 77% for proper helmet, and 72% for wrong helmet use. Most misclassifications occurred between proper and wrong helmet use or between no helmet and the background. The results indicated that the model was highly effective in motorcycle classification but faced difficulty with finer distinctions among helmet-related categories. This challenge may be attributed to overlapping visual features and limited instances for certain categories.

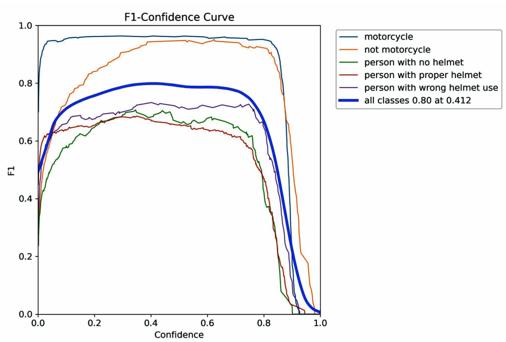
The confusion matrix results confirmed that the model could reliably distinguish motorcycles but needed further refinement for more complex helmet classifications. Despite these gaps, the model still demonstrated practical utility for road monitoring. Misclassifications suggested the need for more balanced class representation in future datasets, particularly for wrong helmet use and no helmet images. Overall, the confusion matrix showed strong classification performance, highlighting the model's reliability for motorcycle detection and moderate success in differentiating helmet compliance types.



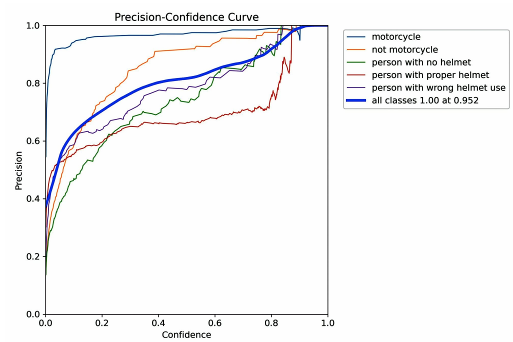
a.) Class Distribution **b.) Box Center Distribution** **c).Box Size Distrinution**

Figure 11: **Dataset Visualization Results**

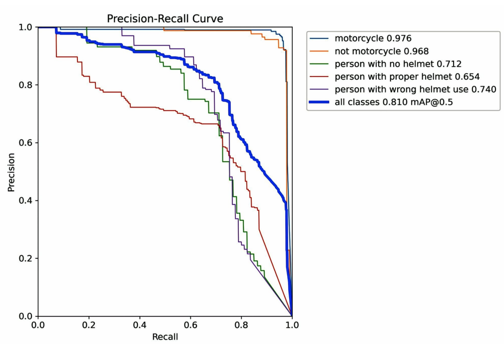
As shown in Figure 11a, most bounding boxes are concentrated near the center of the image frame, indicating that the dataset was captured with consistent camera positioning and frontal viewpoints. Figure 11b shows that the class distribution is generally balanced across motorcycles, proper helmet use, no helmet, and incorrect helmet use, allowing the model to learn from a wide range of scenarios. Figure 11c further reveals that most annotated objects have small width–height ratios, meaning that the dataset mainly contains distant or moderately sized subjects such as riders and helmets. These findings indicate that the dataset provides stable framing, balanced class exposure, and detailed small-object samples, all of which support effective YOLOv8 training and reliable detection performance. However, the limited number of off-centered objects and the dominance of small object sizes may reduce the model's adaptability in situations where riders appear at the edges of the frame or very close to the camera. Overall, the dataset offers a strong foundation for accurate helmet compliance detection but would benefit from added spatial diversity and varied object scales to further improve generalization.



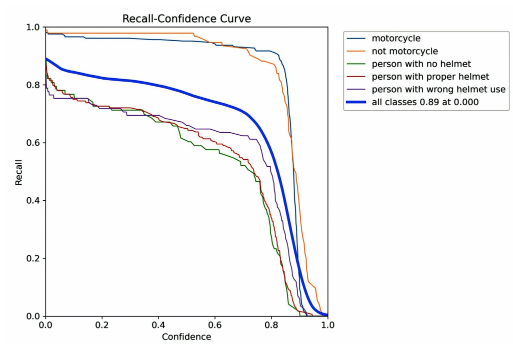
(a) **F1**



(b) **Precision**



(c) **Precision–Recall**

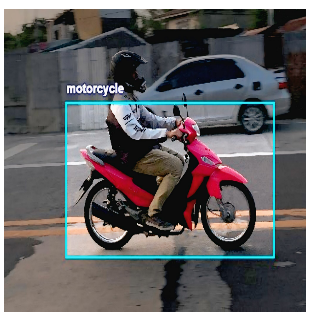


(d) **Recall–Confidence**

Figure 12: **Box Curve Results**

As shown in Figure 12, the F1-Confidence Curve (Graph A) demonstrated the model's best trade-off between precision and recall at a confidence threshold of 0.41, achieving an F1 score of 0.80. The Precision-Confidence Curve (Graph B) showed precision increasing with higher confidence thresholds, while helmet-related classes achieved slightly lower precision. The Precision-Recall Curve (Graph C) showed 0.976 precision for motorcycles and 0.981 for non-motorcycles, compared to 0.712, 0.654, and 0.740 for helmet-related classes. The Recall-Confidence Curve (Graph D) revealed that recall peaked at 0.89 under lower thresholds.

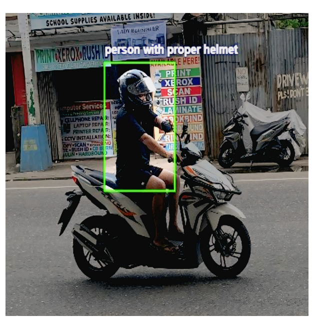
The graphs confirmed that the model balanced sensitivity and precision effectively but found it more challenging to maintain this balance for helmet-related detections. The lower F1 scores for those classes reflected the complexity of helmet visibility and differentiation. The results reinforced that YOLOv8 performed well overall and that its detection capability was strong for general object classes. The consistent precision and recall for motorcycles indicated real-world deployment potential. The drop in precision for helmet classes showed the need for dataset enhancement and improved feature differentiation during model training. To conclude, the performance graphs validated the model's effectiveness in detecting motorcycles and highlighted the areas requiring improvement in recognizing subtle helmet variations.



(a) **motorcycle**

(b) **not motorcycle**

(c) **person with no helmet**



(d) **person with proper helmet**

(e) **person with wrong helmet use**

Figure 13: **Samples of Dataset: (a) Motorcycle, (b) Not motorcycle, (c) Person with no helmet, (d) Person with proper helmet, (e) Person with wrong helmet use.**

This section discusses the training and evaluation of the YOLOv8 model developed for helmet compliance detection. It presents the model's learning process and performance based on various evaluation metrics and visual results. Collectively, the figures and tables comprehensively summarize the model's development, learning behavior, and overall detection performance across multiple real-world testing conditions, ensuring robust, accurate, and efficient helmet detection outcomes.

Vehicle Filtering

The vehicle filtering stage of the Helmet Compliance Detection Prototype yielded significant findings, as it successfully identified motorcycles from mixed traffic and ensured that helmet detection was applied only to the appropriate vehicle type. This selective filtering reduced false detections, minimized unnecessary computations, and improved the system's accuracy during real-time operation. These results indicated that isolating motorcycles before performing helmet detection greatly enhanced the model's reliability, particularly in complex scenarios involving dense traffic, varying speeds, and overlapping objects. When compared with existing literature, the findings aligned with prior studies that emphasized the importance of pre-filtering and object prioritization in computer vision workflows. Research using YOLO-based frameworks similarly demonstrated that motorcycle isolation improved detection precision and reduced classification errors, supporting the effectiveness of the approach used in this study. The implications of these results suggested that the prototype could be adapted to diverse environments, as the filtering mechanism allowed the system to maintain consistent performance across different lighting conditions, camera angles, and urban settings. Furthermore, the improved efficiency and reduced processing load strengthened the system's potential for long-term deployment and integration into larger intelligent transportation infrastructures. However, certain limitations were observed, especially when motorcycles appeared partially occluded, affected by motion blur, or exposed to extreme lighting conditions, which occasionally reduced detection accuracy. In summary, the vehicle filtering stage played a vital role in strengthening the detection pipeline, enabling more focused, efficient, and reliable helmet compliance monitoring while establishing a solid foundation for future system enhancements and broader real-world implementation.

Figure 14: **Vehicle Filtering**

As presented in the figure 14, the vehicle filtering process focused on detecting motorcycles while ignoring tricycles and other non-target vehicles. The YOLOv8 model correctly classified tricycles under the “not motorcycle” class, excluding them from further analysis. The prototype then concentrated on motorcycles, accurately identifying helmets worn by riders and passengers, ensuring that only relevant detections were considered for compliance evaluation. The results demonstrated that the filtering mechanism performed effectively, enabling the model to concentrate solely on motorcycles. This selective detection minimized false helmet assessments involving tricycles or other vehicle types. The model’s ability to identify motorcycles with riders and passengers confirmed the reliability of its vehicle classification and filtering function.

The correct classification of tricycles as "not motorcycle" shows that the YOLOv8 model successfully distinguished between different vehicle structures. By filtering out tricycles, the prototype maintained a precise detection flow, improving the consistency and accuracy of helmet compliance analysis. This focused detection ensured that irrelevant objects did not interfere with the system's performance. Accurate vehicle filtering enhances both the efficiency and credibility of the helmet compliance system. By processing only motorcycles, the system reduced computational load and avoided false detections, making it more practical for real-time traffic monitoring and road safety enforcement. Some minor misclassifications may occur in cases where tricycles visually resemble motorcycles, especially under poor lighting, occlusion, or motion blur. These scenarios can slightly affect filtering precision and detection consistency.

Future improvements may include expanding the dataset with more examples of tricycles and similar vehicles. Fine-tuning confidence thresholds or integrating shape-based filtering could further strengthen the system's ability to differentiate between vehicle types under diverse environmental conditions. In summary, the vehicle filtering stage effectively separated motorcycles from non-target vehicles, allowing the YOLOv8 model to focus on helmet detection. This process significantly improved the accuracy and efficiency of the system, confirming its readiness for practical use in real-world traffic monitoring applications.

Helmet Detection

The helmet detection stage represented the core functionality of the Helmet Compliance Detection Prototype. After the model successfully filtered motorcycles, it proceeded to analyze the riders and passengers to determine whether they were wearing helmets correctly, incorrectly, or not at all. This step aimed to identify compliance violations in real time by classifying detected persons into three main categories: person with proper helmet, person with wrong helmet use, and person with no helmet. The detection process relied on the YOLOv8 model's object recognition capabilities, using bounding boxes and class confidence scores to ensure accurate classification. Figure 17 presents the visual results of the helmet detection process performed by the trained YOLOv8 model.

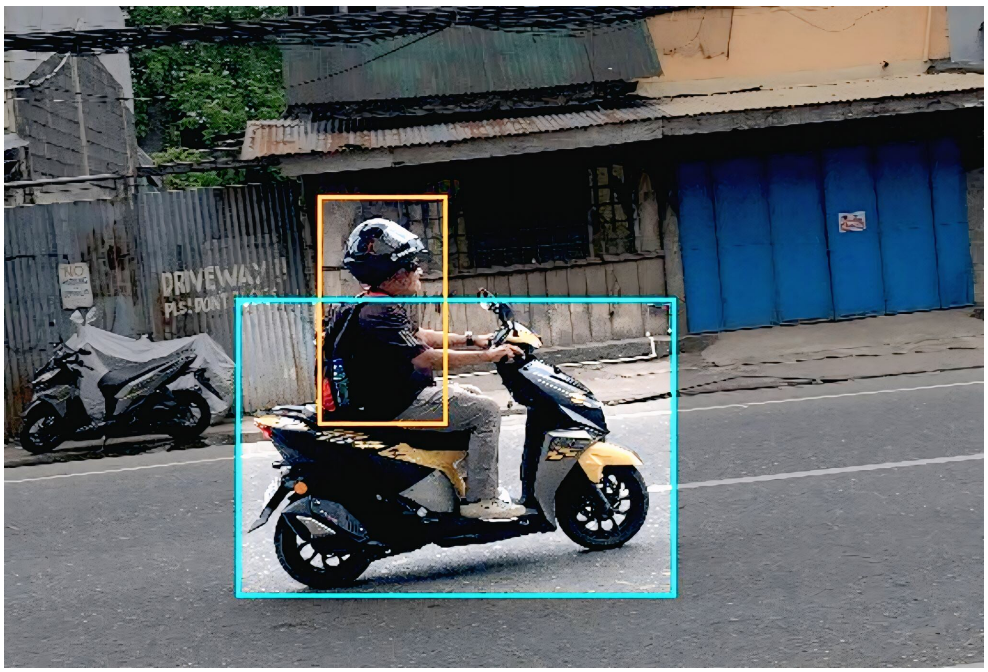


Figure 15: **Helmet Detection**

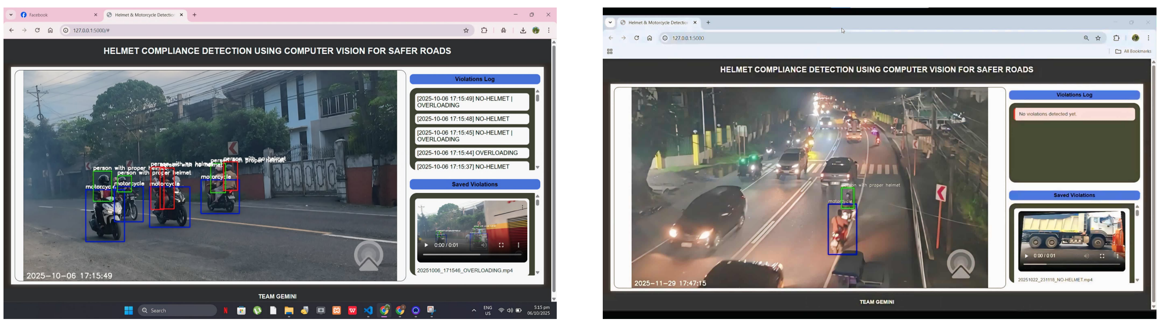
The figure 15 shows that the YOLOv8 model successfully detected a motorcycle rider wearing a helmet incorrectly. The helmet was on the rider's head but lifted or not properly positioned, which the prototype identified as "wrong helmet use." The detection box was accurately placed around the rider's head area, showing that the model could recognize not only the presence of a helmet but also how it was worn. This means the prototype can tell the difference between proper and improper helmet use, proving its ability to detect real-world violations related to helmet compliance. The detection results indicated that the model could distinguish between proper and wrong helmet use with reasonable accuracy. The bounding box and class label correctly represented the identified violation, demonstrating the model's effectiveness in recognizing non-compliant helmet conditions. This confirms that the YOLOv8 model learned the visual differences between correctly and incorrectly worn helmets during training.

The model's correct identification of "wrong helmet use" reflects its capability to analyze detailed visual features, such as helmet position, coverage, and shape. This performance highlights that the prototype does more than simply detect helmets—it can also assess compliance based on how the helmet is worn. Such functionality is crucial for supporting automated enforcement systems that go beyond basic helmet presence detection. Accurate detection of wrong helmet use strengthens the practical application of the helmet compliance prototype in promoting road safety. By identifying riders who wear helmets improperly, the model contributes to more precise violation tracking and data collection. This capability can aid authorities in implementing stricter and smarter enforcement measures to reduce injury risks among motorcycle users.

Although the detection was accurate, challenges may arise in scenarios involving partial occlusions, reflective visors, or riders with non-standard helmets. These visual complexities can affect classification confidence and occasionally lead to misidentification between proper and wrong helmet use. Future improvements could involve expanding the dataset with more examples of riders wearing helmets incorrectly, such as unbuckled or misaligned helmets. Fine-tuning the model's confidence threshold and enhancing image augmentation techniques can further improve recognition under different lighting and motion conditions.

Real-Time Monitoring

The real-time monitoring stage served as the operational phase of the Helmet Compliance Detection Prototype, where all detection processes were executed continuously on live video feeds. After filtering vehicles and identifying motorcycles, the prototype analyzed each rider in real time and classified them as either a person with proper helmet, person with wrong helmet use, or person with no helmet. The YOLOv8 model generated bounding boxes and class labels for each detected individual, clearly displaying their compliance status on the interface. Detected violations, such as missing or improperly worn helmets, were instantly highlighted, allowing for immediate visual feedback. This stage demonstrated the system's capacity for real-time helmet detection, essential for practical use in road surveillance and enforcement applications



a.) Daytime **b.) Night Time**

Figure 16: **Dashboard interface of the Helmet Compliance Detection Prototype**

As illustrated in Figure 16, the dashboard operated smoothly in real time, the YOLOv8 model effectively focused on motorcycles, ignoring other vehicles such as tricycles or cars. Detected riders were classified into categories, and their compliance status was clearly labeled on-screen. The interface highlighted violations immediately, proving the system's capability to perform accurate and fast monitoring of helmet compliance. The system's detection results confirmed that the prototype could track and record helmet violations efficiently. Each recorded event included essential information such as time, video evidence, and violation type. This automated logging process demonstrated the system's potential to reduce manual monitoring tasks and human error in traffic enforcement. Additionally, Image (a) shows the prototype operating under bright daytime conditions, where the model performed with high accuracy due to clear visibility and well-defined object features. In contrast, Image (b) illustrates its performance at night, where artificial lighting and low visibility affected the clarity of incoming frames. Despite these differences, the system still maintained acceptable detection performance, demonstrating its ability to function across varying lighting environments.

Performance, however, still varied under certain conditions. The model encountered difficulties in low-light environments, areas with headlight glare or deep shadows, dense traffic, and situations where helmets were partially obstructed or affected by motion blur. These factors reduced classification accuracy and affected the consistency of recorded events. Identifying these variations was important, as they revealed the prototype's limitations and directed future enhancements. Riders were correctly categorized and their compliance status was displayed on-screen, though rapid motorcycle movement occasionally led to missed or incorrect classifications, especially during sudden accelerations or sharp turns. Frame transitions—such as quick position changes or entering and exiting the camera view—also affected detection stability by limiting clear reference points. Future improvements could include adaptive brightness correction, enhanced nighttime noise reduction, low-light or infrared camera integration, and the use of multiple camera angles for improved visibility. Strengthening performance in challenging weather, such as rain or fog, may also increase reliability. Overall, the Helmet Compliance Detection Dashboard demonstrated effective real-time detection, recording, and management of violations. With strong performance in both daytime (Figure 16a) and nighttime (Figure 16b) conditions, the system proved to be a practical and reliable tool for monitoring road safety, serving as a solid foundation for further advancements in traffic management and public safety.

Real-Time Violation Alert

The prototype provided real-time alerts whenever a rider was detected without a helmet or wearing one incorrectly, displaying a clear “Violation Detected” message on the interface. It also automatically saved short video clips of each violation for documentation and later review. This feature enhanced enforcement efficiency, improved situational awareness, and encouraged safer riding practices. By integrating detection, alerting, and evidence recording, the prototype promoted consistent helmet use, supported data-driven traffic management, strengthened road safety initiatives, and contributed to safer, more disciplined roads across monitored areas, offering valuable insights for authorities to implement targeted safety campaigns and improve compliance through intelligent monitoring systems for sustainable traffic safety.

figures/violation.jpg

Figure 17: **Real Time Violation Alert**

As presented in Figure 17, the prototype accurately detected the motorcycle and its riders, marking them with bounding boxes and assigning confidence-based class labels. In this case, one of the riders was correctly classified under “person with wrong helmet use,” which triggered a red “Violation Detected!!” alert at the top of the frame. A timestamp was also generated automatically, ensuring that each violation was properly documented for later verification and reporting. The system's real-time alert mechanism proved reliable and responsive, showing minimal delay between detection and notification. The recorded video clips provided clear visual evidence, which can be valuable for review and validation by authorities. This feature successfully combined detection accuracy with documentation, supporting both enforcement and analytical applications. The detection and alert process demonstrated the model's ability to function effectively in real-world scenarios. The automatic triggering of alerts and video saving confirmed the system's readiness for deployment in traffic monitoring setups. It also highlighted how intelligent automation could replace manual supervision in identifying and recording safety violations.

This feature helps improve road safety by promoting accountability and consistent helmet use. The alert system not only informs operators of real-time violations but also serves as a deterrent, encouraging riders to follow safety regulations to avoid detection. It provides actionable data that can support government or institutional safety campaigns. Challenges may arise in cases where multiple motorcycles appear close together, or when helmets are partially hidden. These factors can sometimes lead to overlapping bounding boxes or incorrect alert triggering, especially in low-quality or motion-blurred video frames. Future development may include refining multi-object tracking, enhancing video resolution handling, and integrating sound or push notifications for stronger situational awareness. Incorporating cloud-based storage for violations could also improve long-term monitoring and data management. In summary, the Real-Time Violation Alert feature successfully demonstrated the system's ability to detect, notify, and document helmet violations instantly. By combining accurate detection with automated alerts and evidence recording, the prototype supports safer road practices and more efficient traffic enforcement.

CHAPTER 5
CONCLUSION

This chapter provided an overview of the research project "Helmet Compliance Detection using Computer Vision for safer roads" utilizing YOLOv8 model, including its results, conclusions, and recommendations.

Summary

The study "Helmet Compliance Detection Using Computer Vision for Safer Roads" was conducted to provide a practical solution to the increasing number of motorcycle-related accidents in the Philippines. Many of these accidents were caused by riders who did not wear helmets properly and by motorcycles carrying more passengers than allowed. Although the Motorcycle Helmet Act of 2009 required the use of standard protective helmets, enforcement remained weak because manual monitoring was limited and prone to human error. To address this problem, the researchers developed an artificial intelligence–based prototype focused specifically on motorcycles and motorcycle riders. The prototype was designed using the YOLOv8 object detection model together with OpenCV to process video feeds and monitor riders in real time.

A dataset of motorcycle riders with helmets, without helmets, and with improper helmet use was collected, annotated, and used to train the YOLOv8 model. To further improve accuracy, a vehicle filtering feature was added so that the prototype only detected motorcycles and excluded other types of vehicles. The trained model was then integrated into the prototype to identify correct and incorrect helmet usage, detect motorcycles with more than two riders, and record short video clips of violations for evidence. Initial tests conducted on sample videos showed that the prototype could reliably detect different types of violations in real time, particularly under normal lighting conditions. Some limitations were observed in low-light and unfavorable weather simulations, which reduced accuracy. Despite these challenges, the study demonstrated that computer vision and deep learning could effectively support helmet law enforcement. Overall, the prototype showed strong potential to improve road safety by focusing on motorcycle riders and served as a foundation for future enhancements and real-world deployment in AI-based traffic monitoring.

Findings

1. The researchers successfully implemented YOLOv8 for object detection and OpenCV for image and video processing. A total of 1,133 images were collected and categorized into five classes: motorcycle, non-motorcycle, person with no helmet, person with proper helmet, and person with wrong helmet use. Image preprocessing included resizing to 640×640 pixels, correcting orientations, and applying augmentation techniques such as flipping, rotation, cropping, brightness adjustment, blurring, and noise addition. These steps increased dataset diversity and model robustness, allowing the YOLOv8 model to adapt effectively to different lighting and environmental conditions. The implementation confirmed that YOLOv8 and OpenCV were efficient tools for building the foundation of an AI-based helmet compliance prototype.

2. An AI-based real-time prototype was developed, integrating the trained YOLOv8 model to detect motorcycles and assess helmet compliance. The system first filtered motorcycles from the video feed before performing helmet detection to reduce false detections. It featured real-time alerts for violations, automatic video recording, and a web-based dashboard for live monitoring and reviewing saved violation clips. The dashboard displayed timestamps, detection labels, and alert notifications for each recorded event. The prototype functioned effectively in real-time, though occasional

delays were observed when using devices without a dedicated GPU, especially during extended testing sessions.

3. The prototype was tested under various conditions such as different lighting, motion speeds, and multiple riders. It performed best under clear lighting, successfully identifying helmet violations and generating instant alerts. However, performance slightly decreased in low-light or high-motion scenarios, where distinguishing between proper and wrong helmet use became challenging. Despite these limitations, the prototype maintained high reliability in detecting motorcycles and moderate accuracy in helmet classification. These results confirmed the system's potential for real-world use, with improvements needed in dataset diversity, lighting adaptation, and GPU-based optimization.

Conclusions

Based on the findings, the researchers came up with the following conclusions:

1. The study successfully implemented YOLOv8 and OpenCV for object detection and image and video processing. The researchers collected and annotated a dataset of 1,133 images, applying preprocessing and data augmentation techniques to enhance variability and robustness. This provided a solid foundation for training a model capable of detecting motorcycles and classifying helmet usage.

2. A functional monitoring prototype integrating the trained YOLOv8 model was developed, including a web-based dashboard to display live feeds, log violations, and save video clips. The system effectively detected motorcycles and identified helmet compliance, though some misclassifications occurred between proper and wrong helmet use.

3. The prototype demonstrated reliable performance under favorable conditions, accu-

rately detecting motorcycles and issuing violation alerts. Its effectiveness decreased under low-light or fast-moving scenarios, and hardware limitations, such as the lack of a GPU, caused occasional lag. Despite these challenges, the study provides a working prototype with potential for further optimization and real-world deployment for traffic safety monitoring.

Recommendations

Based on the results, the researchers make the following recommendations:

1. The created model can be improved by collecting more datasets, especially for the "wrong helmet use" category, to reduce class imbalance and improve accuracy in detecting different helmet conditions.

2. Future research may also train the model using higher computational resources or a dedicated GPU to shorten training time, reduce lag, and achieve better performance.

3. The prototype may be enhanced by integrating more advanced object detection architectures or combining YOLOv8 with other models to increase accuracy in helmet classification.

4. The dashboard and interface can be further developed to run more smoothly on different devices and networks, allowing real-time monitoring without delays.

5. The model should be fine-tuned with domain-specific datasets to better adapt to real-world conditions, such as varying lighting, weather, and traffic scenarios, while also addressing common misclassifications such as sunglasses being detected as "wrong helmet use," dark helmet types being flagged as violations,or cases where a pedestrian gets misidentified as a passenger because the camera merges them when a motorcycle passes by. Through this refinement, the model's precision can be increased, false positives reduced, and overall detection performance significantly improved.

6. Future researchers may expand the application of the model by adding new features to strengthen road safety enforcement.

BIBLIOGRAPHY

Book Sources

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms*. (3rd ed.). MIT Press. https://archive.org/details/introduction-to-algorithms-third-edition-2009.

Rafael C. Gonzalez and Richard E. Woods. 2018. *Digital Image Processing*. (4th ed.). Pearson. https://www.pearson.com/store/p/digital-image-processing/P100000006584.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. https://www.deeplearningbook.org/.

Lennart Harms-Ringdahl. 2013. *Guide to Safety Analysis for Accident Prevention*. IRS Riskhantering AB. https://irisk.se/sabook/SA-book1.pdf.

Stuart Russell and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*. (4th ed.). Pearson. https://aima.cs.berkeley.edu/.

Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications*. Springer. DOI: 10.1007/978-1-84882-935-0.

Journal Articles

Jorge E. Espinosa, Sergio A. Velastín, and John W. Branch. 2021. Detection of motorcycles in urban traffic using video analysis: a review. *IEEE Transactions on Intelligent Transportation Systems*, 22, 10, 6115–6130. DOI: 10.1109/TITS.2020.2997084.

Tanmayee Godbole, Purva Sapkal, Sujal Malapure, and Shrihari Ganbavale. 2024. Safe rider: helmet detection using machine learning and open cv. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. https://www.ijraset.com/best-journal/safe-rider-helmet-detection-using-machine-learning-and-open-cv.

Ahatsham Hayat and Fernando Morgado-Dias. 2022. Deep learning-based automatic safety helmet detection system for construction safety. *Applied Sciences*, 12, 16, 8268. DOI: 10.3390/app12168268.

Muhammad Hussain. 2024. Yolov5, yolov8 and yolov10: the go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988*. DOI: 10.48550/arXiv.2407.02988.

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. 2022. A review of yolo algorithm developments. *Procedia Computer Science*, 199, 1066–1073. https://www.sciencedirect.com/science/article/pii/S1877050922001363.

Hyunduk Kim, Myoung-Kyu Sohn, and Sang-Heon Lee. 2022. Development of a real-time automatic passenger counting system using head detection based on deep learning. *Journal of Information Processing Systems*, 18, 3, 428–442. DOI: 10.3745/JIPS.04.0246.

Hema Latha Maddi. 2023. Advanced computer vision-based surveillance system for helmet detection and triple rider identification on motorcycles using machine learning. *Unpublished Manuscript*. https://Advanced_Computer_Vision-Based_Surveillance_System_for_Helmet_Detection_and_Triple_Rider_Identification_on_Motorcycles_using_Machine_Learning.

Pavan Mutyala, Chandrika Gali, Jaidev Chivakula, Venkatesh Duggirala, and P. Padmaja. 2023. Automated helmet detection system using rt-detr for real-time monitoring of motorcyclist safety. *International Journal of Scientific Research in Engineering and Management*. DOI: 10.55041/IJSREM40150.

K.W. Ong, S.L. Loh, and T.H. Cheong. 2022. Vehicle classification using neural networks and image processing. *International Journal of Electrical Engineering and Applied Sciences (IJEEAS)*, 5, 2. DOI: 10.54554/ijeeas.2022.5.02.006.

Aleksander Radovan, Leo Mršić, Goran Đambić, and Branko Mihaljević. 2024. A review of passenger counting systems based on image processing and machine learning. *Proceedings of the International Conference on Transportation Engineering and Systems*. https://www.mdpi.com/2673-4117/5/4/172.

William David Moreno Rendon, Carolina Burgos Anillo, Daniel Jaramillo-Ramirez, and Henry Carrillo. 2023. Passenger counting in mass public transport systems using computer vision and deep learning. *IEEE Latin America Transactions*, 21, 4, 537–545. DOI: 10.1109/TLA.2023.10128885.

Mukaram Safaldin, Nizar Zaghden, and Mahmoud Mejdoub. 2024. An improved yolov8 to detect moving objects. *IEEE Access*, 12, 59782–59806. DOI: 10.1109/ACCESS.2024.3393835.

Yahia Said, Yahya Alassaf, Refka Ghodhbani, Yazan Ahmad Alsariera, Taoufik Saidani, Olfa Ben Rhaiem, Mohamad Khaled Makhdoum, and Manel Hleili. 2024. Ai-based helmet violation detection for traffic management system. *CMES-Computer Modeling in Engineering & Sciences*, 141, 1. DOI: 10.32604/cmes.2024.052369.

S. Sanjana, S. Sanjana, V. R. Shriya, M. Sandeep, and R. Vaibhav. 2021. A review on various methodologies used for vehicle classification, helmet detection and number plate recognition. *Evolutionary Intelligence*, 14, 979–987. DOI: 10.1007/s12065-020-00493-7.

Kavuri K. S. V. A. Satheesh, Nandam Sai Akhila, Dondapati Amarnadh, Paruchuri Sagar Swetha, Avula Venkata Sohan, and Vasireddy Pardhiv. 2024. Automated helmet monitoring system using deep learning. *EPRA International Journal*. DOI: 10.36713/epra16328.

Felix Wilhelm Siebert, Christoffer Riis, and Frederik Boe Hüttel. 2024. 365: urban motorcycle helmet use surveillance in southeast asia – a cost-efficient computer vision approach. *Injury Prevention*. DOI: 10.1136/injuryprev-2024-SAFETY.181.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. 2022. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *AI*, 5, 4, 83. https://www.mdpi.com/2504-4990/5/4/83.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. 2023. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5, 4, (Nov. 2023), 1680–1716. DOI: 10.3390/make5040083.

Other Sources

Adil Afzal, Hafiz Siddiq Umer, Zeeshan Khan, and Muhammad Usman Ghani Khan. 2021. Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision. In *2021 International Conference on Artificial Intelligence (ICAI)*. DOI: 10.1109/ICAI52203.2021.9445206.

Kartikeya Bhatt, Abhishek Joshi, Gaurav Singh, Vikrant Sharma, and Satvik Vats. 2024. Ai guardian on the road: harnessing machine learning for two-wheeler helmet detection. In *Proceedings of the 2024 International Conference on Computational Intelligence and Computer Applications (ICCICA)*. DOI: 10.1109/iccica60014.2024.10585188.

Roopa R. Chandrika, N. Gowri Ganesh, A. Mummoorthy, and K. Karthick Raghunath. 2020. Vehicle detection and classification using image processing. In *Proceedings of the 2019 International Conference on Emerging Trends in Science and Engineering (ICESE)*. Vol. 1. IEEE, 1–6. DOI: 10.1109/ICESE46178.2019.9194678.

Rajnandani Choubey, Saloni Sharma, Saloni Mourya, Aditya Narayan, and Dhruv Bhargav. 2025. Helmet and number plate detection by using yolo and opencv. https://www.researchgate.net/profile/Ijeasm-Journal/publication/388218838_Helmet_and_Number_plate_detection_by_using_yolo_and_open_cv/links/678f7986ec3ae3435a72778d/Helmet-and-Number-plate-detection-by-using-yolo-and-open-cv.pdf. (2025).

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2020. Imagenet: a large-scale hierarchical image database. In *2020 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255. DOI: 10.1109/CVPR.2009.5206848.

Encyclopaedia Britannica. n.d. Law enforcement. (n.d.). https://www.britannica.com/topic/law-enforcement.

Encyclopaedia Britannica. n.d. Traffic control. (n.d.). https://www.britannica.com/technology/traffic-control.

GeeksforGeeks. 2021. Yolo – you only look once (real-time object detection). Accessed April 22, 2025. (2021). https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/?ref=ml_lbp.

Nemuel Norman F. Giron, Robert Kerwin C. Billones, Alexis M. Fillone, Jay Robert Del Rosario, Melvin K. Cabatuan, Argel A. Bandala, and Elmer P. Dadios. 2020. Motorcycle rider helmet detection for riding safety and compliance using convolutional neural networks. In *Proceedings of the IEEE Conference*. https://ieeexplore.ieee.org/document/9400149.

B. Karthika, M. Dharssinee, V. Reshma, R. Venkatesan, and R. Sujarani. 2024. Automated object detection and recognition using yolov8. In *Proceedings of the International Conference on Communication, Computing, and Networks Technologies (ICCCNT)*. IEEE. DOI: 10.1109/ICCCNT61001.2024.10724411.

Kili Technology. 2023. Yolo algorithm: real-time object detection from a to z. Accessed April 29, 2025. (2023). https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z.

Abhash Kumar, M. S. Guru Prasad, Dheeraj Soni, Vibha Rani, Kanika Arora, and D. R. Ganesh. 2024. Real-time helmet detection system with vehicle number extraction for two-wheeler vehicles using yolov8. In *Proceedings of the 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*. IEEE, 672–676. https://ieeexplore.ieee.org/document/10551102.

Raman Kumar, Yogendra Kumar, Ahmed Alkhayyat, Lovneesh Sharma, Ankita Joshi, and Ayush Dogra. 2023. Vision-based safety identification and monitoring system using tensorflow. In *International Conference on Intelligent Systems Design and Applications*. Springer Nature Switzerland, (Dec. 2023), 369–375. DOI: 10.1007/978-3-031-64650-8_36.

Raman Kumar, Yogendra Kumar, Ahmed Alkhayyat, Lovneesh Sharma, Ankita Joshi, and Ayush Dogra. 2023. Vision-based safety identification and monitoring system using tensorflow. In *International Conference on Intelligent Systems Design and Applications*. Springer Nature Switzerland, 369–375. https://link.springer.com/chapter/10.1007/978-3-031-64650-8_36.

Casper Lassenius, Timo Soininen, and Jari Vanhanen. 2001. Constructive research. In *Proceedings of the 26th Information Systems Research Seminar in Scandinavia (IRIS 26)*. Accessed: 2025-04-22. https://www.pm.lth.se/fileadmin/_migrated/content_uploads/3._constructive_research.pdf.

Merriam-Webster. [n. d.] Helmet. Merriam-Webster.com Dictionary. (). https://www.merriam-webster.com/dictionary/helmet.

Metropolitan Manila Development Authority. 2023. Mmda logs 17.3% increase in motorcycle-related crashes in 2023. Inquirer.net News. (2023). https://newsinfo.inquirer.net/1927991/mmda-logs-17-3-increase-in-motorcycle-related-crashes-in-2023.

Fadil Muhammad, Ismail Bintang, Rian Fahrizal, Ceri Ahendyarti, Romi Wiryadinata, and Imamul Muttakin. 2024. Real-time motorcyclist helmet detection using yolov8 on edge device. In *2024 International Conference on Informatics Electrical and Electronics (ICIEE)*. IEEE, 1–7. DOI: 10.1109/ICIEE63403.2024.10920426.

Fadil Muhammad, Ismail Bintang, Rian Fahrizal, Ceri Ahendyarti, Romi Wiryadinata, and Imamul Muttakin. 2024. Real-time motorcyclist helmet detection using yolov8 on edge device. In *Proceedings of the 2024 International Conference on Informatics Electrical and Electronics (ICIEE)*. IEEE, (Dec. 2024). DOI: 10.1109/ICIEE63403.2024.10920426.

Oxford Reference. [n. d.] Road safety. In *A Dictionary of Public Health*. (). https://www.oxfordreference.com/display/10.1093/acref/9780199218707.001.0001/acref-9780199218707-e-3052.

Immidisetty V. Prakash and M. Palanivelan. 2024. A study of yolo (you only look once) to yolov8. In *Algorithms in Advanced Artificial Intelligence*. CRC Press, 257–266. DOI: 10.1201/9781003529231-40.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2020. You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://arxiv.org/abs/1506.02640.

Republic of the Philippines. 2009. Republic act no. 10054 - motorcycle helmet act of 2009. Official Law. (2009). https://ltoportal.ph/ra-10054-motorcycle-helmet-act-philippines/.

L. Sharma. 2024. Vision-based safety identification and monitoring system using tensorflow. In *Intelligent Systems Design and Applications: Information and Network Security*. Vol. 3, 369. DOI: 10.1007/978-3-031-64650-8_36.

Surendra Singh. 2024. Visual vigilance: enhancing helmet recognition through computer vision. https://www.researchgate.net/publication/382495810. (2024).

Elham Soltanikazemi, Ashwin Dhakal, Bijaya Hatuwal, and Imad Eddine Toubal. 2023. Real-time helmet violation detection in ai city challenge 2023 with genetic algorithm-enhanced yolov5. https://www.researchgate.net/publication/377236120. (2023).

Vikash Kumar Thakur. 2024. Enhancing safety: a computer vision approach to helmet detection. Medium. (2024). https://medium.com/@vkt08/enhancing-safety-a-computer-vision-approach-to-helmet-detection-c5a48a7275d2.

ThePythonCode. 2021. Non-maximum suppression using opencv in python. https://thepythoncode.com/article/non-maximum-suppression-using-opencv-in-python. Accessed: 2025-04-22. (2021).

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9627–9636. DOI: 10.1109/ICCV.2019.00972.

John Paul Q. Tomas and Bonifacio T. Doma. 2023. Motorcycle helmet detection and usage classification in the philippines using yolov5 algorithm. In *Proceedings of the ACM International Conference on Computing and Data Science (ICCDS 2023)*. DOI: 10.1145/3581792.3581796.

Rejin Varghese and M. Sambath. 2024. Yolov8: a novel object detection algorithm with enhanced performance and robustness. In *Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE, (Apr. 2024), 1–6. DOI: 10.1109/ADICS58448.2024.10533619.

World Health Organization. 2018. Helmet use saves lives. Fact sheet. (2018). https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

Manikanta Yenugonda and Hema Latha Maddi. 2023. Advanced computer vision-based surveillance system for helmet detection and triple rider identification on motorcycles using machine learning. (2023). https://Advanced_Computer_Vision-Based_Surveillance_System_for_Helmet_Detection_and_Triple_Rider_Identification_on_Motorcycles_using_Machine_Learning.

APPENDICES

APPENDIX A

RELEVANT SOURCE CODE

Listing A.1: Flask-Based Real-Time Helmet Violation Detection System Using YOLO

```
from ultralytics import YOLO
import cv2
from datetime import datetime
import os
import time
import platform
from collections import deque
from flask import Flask, render_template,
     Response, send_from_directory,
    jsonify, request, abort
import threading

app = Flask(__name__)

# === Load YOLO model ===
model = YOLO("best.pt")

# === Setup folders ===
VIOLATION_FOLDER = os.path.join(os.getcwd
    (), "violations")
os.makedirs(VIOLATION_FOLDER, exist_ok=
    True)

# === Classes ===
HELMET_CLASSES = [
"person with proper helmet",
"person with helmet",
"person with wrong helmet use",
"person with no helmet"
]

# Updated violation classes
VIOLATION_CLASSES = [
"person with helmet",
"person with wrong helmet use",
"person with no helmet"
]

# Vehicle classes that should be ignored
IGNORED_VEHICLES = [
"not motorcycle",
"bicycle",
"bike"
]

# === State and constants ===
recording = False
record_start_time = None
recorded_frames = []
recorded_violations = set()
max_duration = 7
last_violations = set()
violation_logs = []

# Variables to prevent duplicate logs/
    videos for the same continuous
    violation event
last_logged_violation = None
violation_active = False
last_detected_riders = {} # Track the
    last detected positions of riders
```

```
violation_cooldown = {} # Track cooldown
    period for each rider
COOLDOWN_PERIOD = 10 # Cooldown period in
     seconds
violation_cooldown = {} # Track cooldown
    for each rider
COOLDOWN_PERIOD = 10 # Cooldown period in
     seconds
last_detected_riders = {} # Track last
    detected positions of riders

camera_index = 0

MIN_FPS = 5
MAX_FPS = 60
DEFAULT_FPS = 15

# === Buffers ===
PRE_SECONDS = 1.5
POST_SECONDS = 1.5

cap = cv2.VideoCapture(camera_index)
latest_frame = None
lock = threading.Lock()

real_fps = cap.get(cv2.CAP_PROP_FPS) if
    cap.isOpened() and cap.get(cv2.
    CAP_PROP_FPS) and cap.get(cv2.
    CAP_PROP_FPS) > 0 else DEFAULT_FPS
pre_violation_buffer = deque(maxlen=max
    (1, int(PRE_SECONDS * real_fps)))
post_violation_buffer = deque(maxlen=max
    (1, int(POST_SECONDS * real_fps)))

# === Helmet stability ===
helmet_status_memory = {}
HELMET_STABILITY_FRAMES = 5 # Number of
    consistent frames required before
    changing helmet label

def camera_thread():
global cap, latest_frame
while True:
if not cap.isOpened():
try:
cap.release()
except Exception:
pass
cap = cv2.VideoCapture(camera_index)
time.sleep(1)
continue

ret, frame = cap.read()
if not ret:
try:
cap.release()
except Exception:
pass
cap = cv2.VideoCapture(camera_index)
time.sleep(1)
continue

with lock:
latest_frame = frame


threading.Thread(target=camera_thread,
    daemon=True).start()


def log_violation(current_violations):
global violation_logs
if not current_violations:
return
label = " | ".join(sorted(
    current_violations))
timestamp = datetime.now().strftime("%Y-%
    m-%d %H:%M:%S")
log_entry = f"[{timestamp}] {label}"
```

```
if len(violation_logs) == 0 or
    violation_logs[0] != log_entry:
violation_logs.insert(0, log_entry)
violation_logs = violation_logs[:50]


def play_beep():
system = platform.system()
try:
if system == "Windows":
import winsound
winsound.Beep(1000, 300)
elif system == "Darwin":
os.system('say "Violation detected"')
else:
print('\a')
except Exception as e:
print(f"[Sound Error] {e}")


def _open_video_writer(filepath, fps,
    size):
codecs_to_try = ["avc1", "mp4v", "H264",
    "XVID"]
for codec in codecs_to_try:
fourcc = cv2.VideoWriter_fourcc(*codec)
out = cv2.VideoWriter(filepath, fourcc,
    fps, size)
if out.isOpened():
print(f"[VideoWriter] Using codec {codec}
    ")
return out
return None


def save_violation_video(frames,
    violations, fps=None):
if not frames:
return None
timestamp = datetime.now().strftime("%Y%m
    %d_%H%M%S")
violation_label = "_".join(sorted(
    violations)).upper() if violations
    else "UNKNOWN"
filename = f"{timestamp}_{violation_label
    }.mp4"
filepath = os.path.join(VIOLATION_FOLDER,
     filename)
fps_to_use = 30

h, w = frames[0].shape[:2]
size = (w, h)

out = _open_video_writer(filepath,
    fps_to_use, size)
if not out:
print("[ERROR] Could not open VideoWriter
    .")
return None

for f in frames:
if (f.shape[1], f.shape[0]) != size:
f = cv2.resize(f, size)
out.write(f.astype("uint8"))

out.release()
print(f"[SAVED] {filename} (fps={
    fps_to_use}, frames={len(frames)})")
return filename


def get_rider_id(person_box,
    motorcycle_box):
px1, py1, px2, py2 = person_box
mx1, my1, mx2, my2 = motorcycle_box
pcx, pcy = (px1 + px2) / 2, (py1 + py2) /
     2
mcx, mcy = (mx1 + mx2) / 2, (my1 + my2) /
     2
```

```
# Create a unique identifier based on
    relative position
return f"{int(pcx-mcx)}_{int(pcy-mcy)}"

def is_rider(person_box, motorcycle_box,
    tolerance=40):
px1, py1, px2, py2 = person_box
mx1, my1, mx2, my2 = motorcycle_box
pcx, pcy = (px1 + px2) / 2, (py1 + py2) /
     2
mx1_t, my1_t = mx1 - tolerance, my1 -
    tolerance
mx2_t, my2_t = mx2 + tolerance, my2 +
    tolerance
return mx1_t <= pcx <= mx2_t and my1_t <=
     pcy <= my2_t


def generate_frames():
global recording, recorded_frames,
    recorded_violations, last_violations
    , latest_frame, pre_violation_buffer
    , post_violation_buffer
global last_logged_violation,
    violation_active,
    last_detected_riders,
    violation_cooldown
violation_buffer = deque(maxlen=3)

while True:
if latest_frame is None:
time.sleep(0.05)
continue

with lock:
frame = latest_frame.copy()

try:
results = model.predict(source=frame,
    conf=0.5, iou=0.45, verbose=False)
except Exception as e:
print(f"[PREDICT ERROR] {e}")
ret, buffer = cv2.imencode('.jpg', frame)
if not ret:
continue
yield (b'--frame\r\nContent-Type: image/
     jpeg\r\n\r\n' + buffer.tobytes() + b
     '\r\n')
continue

motorcycles, persons, valid_boxes = [],
     [], []
boxes = getattr(results[0], "boxes", [])
     if results and results[0] else []

for box in boxes:
try:
xy0 = box.xyxy[0]
xy = xy0.cpu().numpy() if hasattr(xy0, "
     cpu") else xy0
x1, y1, x2, y2 = map(int, xy)
except Exception:
continue

if x2 - x1 <= 40 or y2 - y1 <= 40:
continue

try:
cls_val = box.cls[0] if hasattr(box.cls,
     "__len__") else box.cls
cls_id = int(cls_val)
except Exception:
cls_id = int(getattr(box, "label", -1))

cls_name = model.names[cls_id] if hasattr
     (model, "names") and cls_id in model
     .names else str(cls_id)

# === Classification filter ===
if cls_name == "motorcycle":
```

```
motorcycles.append((x1, y1, x2, y2))
valid_boxes.append((x1, y1, x2, y2,
    cls_name))
elif cls_name.lower() in [v.lower() for v
     in IGNORED_VEHICLES]:
continue # Skip bicycles and other non-
    motorcycle vehicles
elif cls_name in HELMET_CLASSES:
persons.append((x1, y1, x2, y2, cls_name)
    )

# --- Improved Helmet Status Memory
    Tracking ---
person_center = ((x1 + x2) // 2, (y1 + y2
    ) // 2)
person_id = f"{person_center[0]}_{
    person_center[1]}"
prev = helmet_status_memory.get(person_id
    , {"label": cls_name, "streak": 0})

if cls_name == prev["label"]:
prev["streak"] = 0 # same class, reset
    streak
else:
prev["streak"] += 1
if prev["streak"] >=
    HELMET_STABILITY_FRAMES:
prev["label"] = cls_name
prev["streak"] = 0

helmet_status_memory[person_id] = prev
stable_label = prev["label"]

persons[-1] = (x1, y1, x2, y2,
    stable_label)

# Only keep motorcycles that actually
    have a rider
motorcycles = [m for m in motorcycles if
    any(is_rider(p[:4], m) for p in
    persons)]

# Validate persons within motorcycles
if motorcycles and persons:
for px1, py1, px2, py2, cls_name in
    persons:
for mx1, my1, mx2, my2 in motorcycles:
if is_rider((px1, py1, px2, py2), (mx1,
    my1, mx2, my2)):
valid_boxes.append((px1, py1, px2, py2,
    cls_name))
break

# === Violation Detection with Rider
    Tracking ===
current_violations = set()
current_time = time.time()
detected_riders = {}

if motorcycles:
for px1, py1, px2, py2, cls_name in
    persons:
for mx1, my1, mx2, my2 in motorcycles:
if is_rider((px1, py1, px2, py2), (mx1,
    my1, mx2, my2)):
rider_id = get_rider_id((px1, py1, px2,
    py2), (mx1, my1, mx2, my2))

# Check if this rider is in cooldown
if rider_id in violation_cooldown:
if current_time - violation_cooldown[
    rider_id] < COOLDOWN_PERIOD:
continue
else:
del violation_cooldown[rider_id]

detected_riders[rider_id] = (px1, py1,
    px2, py2, cls_name)

if cls_name in VIOLATION_CLASSES:
```

```
if cls_name == "person with helmet" or
    cls_name == "person with no helmet":
current_violations.add("NO-HELMET")
violation_cooldown[rider_id] =
    current_time
elif cls_name == "person with wrong
    helmet use":
current_violations.add("WRONG-HELMET")
violation_cooldown[rider_id] =
    current_time

# Overloading detection (with cooldown)
if len(detected_riders) > 2:
current_violations.add("OVERLOADING")
# Add cooldown for all riders in this
    motorcycle
for rider_id in detected_riders:
violation_cooldown[rider_id] =
    current_time

# Clean up old entries from tracking
    dictionaries
current_rider_ids = set(detected_riders.
    keys())
for rider_id in list(last_detected_riders
    .keys()):
if rider_id not in current_rider_ids:
last_detected_riders.pop(rider_id, None)

last_detected_riders = detected_riders.
    copy()

violation_buffer.append(
    current_violations)
combined_violations = set().union(*
    violation_buffer)
violation_detected = bool(
    combined_violations)

# === Prevent duplicate logs / videos for
    same continuous violation ===
if violation_detected:
# Trigger log/beep only once for a
    continuous violation event,
# or if the violation content changed (
    different violation type)
if (not violation_active) or (
    combined_violations !=
    last_logged_violation):
play_beep()
log_violation(combined_violations)
last_logged_violation =
    combined_violations.copy()
violation_active = True
else:
# No active violation -> reset active
    flag so next event can be logged/
    saved
violation_active = False

# === Drawing annotations ===
annotated_frame = frame.copy()
for x1, y1, x2, y2, cls_name in
    valid_boxes:
if cls_name == "motorcycle":
color = (255, 0, 0)
elif cls_name == "person with proper
    helmet":
color = (0, 255, 0)
elif cls_name in ["person with helmet", "
    person with no helmet"]:
color = (0, 0, 255)
elif cls_name == "person with wrong
    helmet use":
color = (0, 165, 255)
else:
color = (200, 200, 200)

cv2.rectangle(annotated_frame, (x1, y1),
    (x2, y2), color, 2)
```

```
cv2.putText(annotated_frame, cls_name, (
    x1, y1 - 6),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
     255), 1)

# RED WARNING TEXT WHEN VIOLATION IS
    DETECTED
if violation_detected:
cv2.putText(annotated_frame, "VIOLATION
    DETECTED!!",
(50, 50), cv2.FONT_HERSHEY_SIMPLEX,
1.2, (0, 0, 255), 4)

cv2.putText(annotated_frame, datetime.now
    ().strftime("%Y-%m-%d %H:%M:%S"),
(10, annotated_frame.shape[0] - 10), cv2.
    FONT_HERSHEY_SIMPLEX, 0.7, (255,
    255, 255), 2)

# === Video recording logic ===
pre_violation_buffer.append(
    annotated_frame.copy())

if violation_detected:
if not recording:
recording = True
recorded_frames = []
recorded_violations = combined_violations
    .copy()
record_start_time = time.time()

# append frames while recording
recorded_frames.append(annotated_frame.
    copy())
post_violation_buffer.clear()

# save when enough frames accumulated
if len(recorded_frames) >= max(1, int(
    max_duration * real_fps)):
frames_to_save = list(
    pre_violation_buffer) +
    recorded_frames
# save only once per recording session (
    recording flag controls this)
save_violation_video(frames_to_save,
    recorded_violations, fps=30)
recording = False
recorded_frames = []
recorded_violations = set()
last_violations.clear()
pre_violation_buffer.clear()
post_violation_buffer.clear()

else:
if recording:
frames_to_save = list(
    pre_violation_buffer) +
    recorded_frames + list(
    post_violation_buffer)
save_violation_video(frames_to_save,
    recorded_violations, fps=30)
recording = False
recorded_frames = []
recorded_violations = set()
last_violations.clear()
pre_violation_buffer.clear()
post_violation_buffer.clear()
else:
post_violation_buffer.append(
    annotated_frame.copy())

ret, buffer = cv2.imencode('.jpg',
    annotated_frame)
if not ret:
continue
yield (b'--frame\r\nContent-Type: image/
    jpeg\r\n\r\n' + buffer.tobytes() + b
    '\r\n')
```

```
@app.route('/')
def index():
return render_template("dashboard.html")


@app.route('/video_feed')
def video_feed():
return Response(generate_frames(),
    mimetype='multipart/x-mixed-replace;
     boundary=frame')


@app.route('/violations/<path:filename>')
def get_violation(filename):
filepath = os.path.join(VIOLATION_FOLDER,
     filename)
if not os.path.exists(filepath) or not
    filename.lower().endswith(".mp4"):
abort(404)
return send_from_directory(
    VIOLATION_FOLDER, filename, mimetype
    ="video/mp4", as_attachment=False)


@app.route('/saved_videos')
def saved_videos():
try:
files = sorted([f for f in os.listdir(
    VIOLATION_FOLDER) if f.lower().
    endswith(".mp4")], reverse=True)
return jsonify(files)
except Exception as e:
return jsonify({"error": str(e)})

@app.route('/logs')
def get_logs():
return jsonify(violation_logs)


@app.route('/delete_violation/<path:
    filename>', methods=['POST'])
def delete_violation(filename):
global violation_logs
filepath = os.path.join(VIOLATION_FOLDER,
     filename)
if os.path.exists(filepath) and filename.
    lower().endswith(".mp4"):
try:
os.remove(filepath)
except Exception as e:
return jsonify({"success": False, "
    message": str(e)}), 500

time_prefix = filename.split('_')[0]
violation_logs[:] = [log for log in
    violation_logs if time_prefix not in
     log]
return jsonify({"success": True})

return jsonify({"success": False, "
    message": "File not found"}), 404


if __name__ == "__main__":
app.run(host="0.0.0.0", port=5000, debug=
    False)
```

APPENDIX B

EVALUATION TOOLS

APPENDIX C

DOCUMENTATIONS

APPENDIX D

JOINT AFFIDAVIT OF UNDERTAKING (PLAGIARISM)

JOINT AFFIDAVIT OF UNDERTAKING

APPENDIX E

PROJECT TEAM ASSIGNMENT FORM

APPENDIX F

ROLE ACCEPTANCE FORM

ROLE ACCEPTANCE FORM
College of Computer Studies

Date:February 26, 2025

To: **Kaela Marie Fortuno, MIT**

We, the third year students of Camarines Sur Polytechnic Colleges pursuing a degree in BACHELOR OF SCIENCE IN COMPUTER SCIENCE, are currently enrolled in Thesis1.

We are writing to humbly request your service and expertise to serve as our Adviser for our thesis. We believe that your knowledge and experience will be essential to greatly enrich our work. Attached are our thesis tentative title proposals for your kind reference.

Thank you and looking forward to your favorable response of our request.

Respectfully,

Dela Justa, Aina Mae
Epres, Caren Joy
Matubis, Maria Angela
Team Gemini

To: **Rosel Onesa, MIT**
Dean, CCS

This formally signifies that I ACCEPT/ REJECT the request to serve as Adviser of the **Team Gemini.**

As Adviser, I agree to perform my duties and responsibilities stipulated in Section 2.6 of the TCP Guidebook from Thesis 1 until Thesis 2.

Furthermore, I agree to set the schedule for advising or consultation to help the students and ensure the success of the thesis/ capstone project.

Conformed:

Kaela Marie Fortuno, MIT
Name and Signature

Role Acceptance Form

ROLE ACCEPTANCE FORM
College of Computer Studies

Date: February 12, 2025

To: **Allan Ibo Jr. MSc.**

We, the third year students of Camarines Sur Polytechnic Colleges pursuing a degree in Bachelor of Science in Computer Science, are currently enrolled in Thesis1.

We are writing to humbly request your service and expertise to serve as our Consultant for our thesis. We believe that your knowledge and experience will be essential to greatly enrich our work. Attached are our thesis tentative title proposals for your kind reference.

Thank you and looking forward to your favorable response of our request.

Respectfully,

Dela Justa, Aina Mae
Epres, Caren Joy
Matubis, Maria Angela
Team Gemini

To: **Rosel Onesa, MIT**
Dean, CCS

This formally signifies that I ACCEPT/ REJECT the request to serve as Consultant of the **Team Gemini**.

As Consultant, I agree to perform my duties and responsibilities stipulated in Section 2.6 of the TCP Guidebook from Thesis 1 until Thesis 2.

Furthermore, I agree to set the schedule for advising or consultation to help the students and ensure the success of the thesis/ capstone project.

Conformed:

Allan Ibo Jr. Msc.
Name and Signature

Role Acceptance Form

ROLE ACCEPTANCE FORM
College of Computer Studies

Date: February 26, 2025

To: **Ma. Allaine C. Agna, LPT**

We, the third year students of Camarines Sur Polytechnic Colleges pursuing a degree in BACHELOR OF SCIENCE IN COMPUTER SCIENCE, are currently enrolled in Thesis1.

We are writing to humbly request your service and expertise to serve as our Grammarian for our thesis. We believe that your knowledge and experience will be essential to greatly enrich our work. Attached are our thesis tentative title proposals for your kind reference.

Thank you and looking forward to your favorable response of our request.

Respectfully,

Dela Justa, Aina Mae
Epres, Caren Joy
Matubis, Maria Angela
Team Gemini

To: **Rosel Onesa, MIT**
Dean, CCS

This formally signifies that I ACCEPT/ REJECT the request to serve as Adviser of the **Team Gemini.**

As Grammarian, I agree to perform my duties and responsibilities stipulated in Section 2.6 of the TCP Guidebook from Thesis 1 until Thesis 2.

Furthermore, I agree to set the schedule for advising or consultation to help the students and ensure the success of the thesis/ capstone project.

Conformed:

Ma. Allaine C. Agna, LPT
Name and Signature

Role Acceptance Form

APPENDIX G

FINAL PROJECT TITLE FORM

FINAL PROJECT TITLE FORM

Team Alias: GEMINI
Proponents/Researchers:

1) Aina Mae Dela Justa
2) Maria Angela Matubis
3) Caren Joy Epres

Proposed Thesis/ Capstone Project Title:

Helmet Compliance Detection Using Computer Vision for Safer Roads

Submitted by: **Aina Mae Dela Justa** ______________________ (Signature of Project Head over printed name) Date: ______________	**Noted:** **ROSEL O. ONESA, MIT** ______________________ (Signature of Subject Adviser over printed name) Date: ______________
Recommending Approval: **KAELA MARIE FORTUNO, MIT** ______________________ (Signature of Thesis Adviser over printed name) Date: ______________	**Approved:** **ROSEL O. ONESA, MIT** ______________________ DEAN, CCS Date: MAY 0 5 2025

APPENDIX H

THESIS PROJECT HEARING FORM (TD, POD, FOD)

APPENDIX I

PANEL RSC (TD,POD,FOD)

APPENDIX J

CONSULTATION LOGS FORM

Thesis/ Capstone Project Title	Computer Vision-Based Helmet Compliance System for Safer Roads		
Proponents:			
Alias:			
Total # of Modules		as approved by the CAPSA / TSA	

PROTOTYPE	DT of Consultation	# of Modules Fully Implemented	# of Modules Partially Implemented	Running Score	Percentage	Project Manager's Signature	TA/TSA/ CAPA/ CAPSA/ Consultant Signature
Consultation on Chapter 1	3/14/25						ff
	Remarks						
	add discussion of [illegible] in the B of P						
	Revise SOs, scope & limitation						
Deadline: ________							
Consultation on Chapter 1	4/1/25						ff
	Remarks						
	Improve all SO+scope						
	cite all meaning in PD						
Deadline:							

CONSULTATION LOGS FORM

*** Must attach with this form your chosen Data and Process Model (IT) or Algorithm Model (CS)

CONSULTATION LOGS FORM

Thesis/ Capstone Project Title		Computer Vision-Based Helmet Compliance System for Safer Roads					
Proponents:		AINA MAE DELA JUSTA			MARIA ANGELA MATUBIS		
Alias:	GEMINI	CAREN JOY EPRES					
Total # of Modules				as approved by the CAPSA / TSA			
PROTOTYPE	**DT of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Manager's Signature**	**TA/TSA/ CAPA/ CAPSA/ Consultant Signature**
CHAPTER 2 CONSULTATION	4/7/25						
	Remarks						
	Revise flow of discussion						
	Add RRL for evaluation of algo						
	Synthesize of all RRL						
	Highlight 'GAP'						
Deadline: ________							
CHAPTER 2 CONSULTATION	5/6/25						
	Remarks						
	Use ACM citation						
Deadline:							

*** Must attach with this form your chosen Data and Process Model (IT) or Algorithm Model (CS)

Thesis/ Capstone Project Title							
Proponents:		Dela Justa, Aina Mae F. Epres, Caren Joy L.			Matubis, Maria Angela N.		
Alias:	Gemini						
Total # of Modules				**as approved by the CAPSA / TSA**			
PROTOTYPE	**DT of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Manager's Signature**	**TA/TSA/ CAPA/ CAPSA/ Consultant Signature**
Consultation of Chapter 3	4/23/25						
	Remarks						
	One figure / table per page						
	Revise Evaluation metrics, add Test case						
	ACM citation on notes						
Deadline: ______							
Consultation of Chapter 3	4/30/25						
	Remarks						
	Use latex						
	Follow Chapter 3 format						
Deadline:							

CONSULTATION LOGS FORM

*** Must attach with this form your chosen Data and Process Model (IT) or Algorithm Model (CS)

Thesis/ Capstone Project Title	Helmet Compliance Detection Using Computer Vision for Safer Roads		
Proponents:		AINA MAE DELA JUSTA	MARIA ANGELA MATUBIS
Alias:	GEMINI	CAREN JOY EPRES	
Total # of Modules		as approved by the CAPSA / TSA	

PROTOTYPE	DT of Consultation	# of Modules Fully Implemented	# of Modules Partially Implemented	Running Score	Percentage	Project Manager's Signature	TA/TSA/ CAPA/ CAPSA/ Consultant Signature
	5/6/25						14
CONSULTATION OF CHAPTER 2	**Remarks**						
	Approved.						
Deadline: ______							
CONSULTATION OF CHAPTER 2	**Remarks**						
Deadline:							

CONSULTATION LOGS FORM

Thesis/ Capstone Project Title	Helmet Compliance Detection Using Computer Vision for Safer Roads		
Proponents:	AINA MAE DELA JUSTA	MARIA ANGELA MATUBIS	
Alias: GEMINI	CAREN JOY EPRES		
Total # of Modules		**as approved by the CAPSA / TSA**	

PROTOTYPE	**DT of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Manager's Signature**	**TA/TSA/ CAPA/ CAPSA/ Consultant Signature**
CONSULTATION OF CHAPTER 3	5/6/25						[signature]
	Remarks						
	Approved.						
Deadline: ________							
CONSULTATION OF CHAPTER 3							
	Remarks						
Deadline:							

CONSULTATION LOGS FORM

CONSULTATION LOGS FORM

Thesis Project Title		HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS					
Proponents:		DELA JUSTA, AINA MAE F.			MATUBIS, MARIA ANGELA N.		
Alias:	GEMINI	EPRES, CAREN JOY L.					
Total # of Modules				**as approved by the TSA**			
PROTOTYPE	**DT of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Manager's Signature**	**TA/TSA/ Consultant Signature**
CONSULTATION OF CHAPTER 1 TO 3	9/10/25						Jf
	Remarks						
	Hanging progress						
	Some of RSC are not yet revised.						
Deadline: ______							
CONSULTATION OF CHAPTER 1 TO 3							
	Remarks						
Deadline:							

*** Must attach with this form your chosen Data and Process Model (IT) or Algorithm Model (CS)

CONSULTATION LOGS FORM

Thesis Title:	**HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS**						
Proponents:	Aina Mae F. Dela Justa Caren Joy L. Epres Maria Angela N. Matubis						
Alias:	GEMINI						
Total # of Modules:				**as approved by the CAPSA /TSA:**			
PROTOTYPE	**Date of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Heads's Signature**	**TA/TSA/ Consultant/ Grammarian Signature**
Consultation of Chapter 1 to 5	10/6/25						H
	Remarks: page # C4: procedure / process (7) - provide proof (in figure / table from each) - multiple figure (check format) C5: Findings based objectives Conclusion based on findings Check format for subsection						
Deadline:							

CONSULTATION LOGS FORM

Thesis Project Title	HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS		
Proponents:		DELA JUSTA, AINA MAE F.	MATUBIS, MARIA ANGELA N.
Alias:	GEMINI	EPRES, CAREN JOY L.	
Total # of Modules		as approved by the TSA	

PROTOTYPE	DT of Consultation	# of Modules Fully Implemented	# of Modules Partially Implemented	Running Score	Percentage	Project Manager's Signature	TA/TSA/ Consultant Signature
CONSULTATION OF CHAPTER 1 TO 3	10/7/25						
	Remarks						
	C4: subsection font, figure label font, PIILS discussion						
	C5: align findings based on objectives						
Deadline: ________							
CONSULTATION OF CHAPTER 1 TO 3							
	Remarks						
Deadline:							

*** Must attach with this form your chosen Data and Process Model (IT) or Algorithm Model (CS)

CONSULTATION LOGS FORM

Thesis Title:	**HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS**						
Proponents:	Aina Mae F. Dela Justa Caren Joy L. Epres Maria Angela N. Matubis						
Alias:	GEMINI						
Total # of Modules:				**as approved by the CAPSA /TSA:**			
PROTOTYPE	**Date of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Heads's Signature**	**TA/TSA/ Consultant/ Grammarian Signature**
Consultation of Prototype	10/9/25						[signature]
	Remarks: Revised done. Good to go!						
Deadline:							

CONSULTATION LOGS FORM

Thesis Title:	**HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS**						
Proponents:	Aina Mae F. Dela Justa Caren Joy L. Epres Maria Angela N. Matubis						
Alias:	GEMINI						
Total # of Modules:				**as approved by the CAPSA /TSA:**			
PROTOTYPE	**Date of Consultation**	**# of Modules Fully Implemented**	**# of Modules Partially Implemented**	**Running Score**	**Percentage**	**Project Heads's Signature**	**TA/TSA/ ~~Consultant/~~ Grammarian Signature**
Consultation of Prototype	10/3/25						[signature]
	Remarks: Good for POD, Good luck!						
Deadline:							

APPENDIX K

LANGUAGE EDITING CERTIFICATION

This is to certify that the undersigned has reviewed and went through all the pages of the Bachelor of Science in Computer Science thesis manuscript titled **"HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS"** of **Dela Justa, Aina Mae F**, **Epres, Caren Joy L Matubis, Maria Angela N**, as against the set of structural rules that govern research writing in accord with the composition of sentences, phrases, and words in the English language.

MA ALLAIGNE C. AGNA

Language Editor

Date:_______________________

APPENDIX L

SECRETARY'S CERTIFICATION

This is to certify that the undersigned has provided accurate recommendations, suggestions, and comments unanimously agreed and approved by the panel of examiners during the oral examination of the thesis titled

"HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS"

prepared and submitted by **Dela Justa, Aina Mae F**, **Epres, Caren Joy L**, **Matubis, Maria Angela N**, and that the same have not been amended, modified or obliterated.

MS. MARRI GRACE MORATA

Secretary

Date:_______________________

APPENDIX M

GRAMMARIAN CERTIFICATE

This is to certify that the undersigned has reviewed and went through all the pages of the Bachelor of Science in Information Technology thesis manuscript titled **"HELMET COMPLIANCE DETECTION USING COMPUTER VISION FOR SAFER ROADS"** of **Aina Mae F. Dela Justa, Caren Joy L. Epres, Maria Angela N. Matubis**, as against the set of structural rules that govern research writing in accord with the composition of sentences, phrases, and words in the English language.

MS. MA. ALLAINE C. AGNA

Grammarian

Date:______________________

APPENDIX N

CERTIFICATE OF TRANSFER

CAMARINES SUR POLYTECHNIC COLLEGES
COLLEGE OF COMPUTER STUDIES

CERTIFICATE
OF TRANSFER

This is to certify that the

Helmet Compliance Detection using Computer Vision for Safer Roads

Developed by Caren Joy L. Epres, Aina Mae F. Dela Justa, Maria Angela N. Matubis, Camarines Sur Polytechnic Colleges (CSPC) - College of Computer Studies (CCS) Students, as their undergraduate thesis project was transferred to the CCS Dean's Office, Camarines Sur Polytecnhic Colleges for further research.

This certificate is issued this ____ day of _____ ______ at CCS Camarines Dean's Office for whatever legitimate purpose it may serve.

ROSEL O. ONESA, MIT
OIC DEAN, COLLEGE OF COMPUTER SCIENCE

APPENDIX O

ACM FORMAT

APPENDIX P

CERTIFICATE OF PLAGIARISM CHECKER

VITA

AINA MAE F. DELA JUSTA

PERSONAL INFORMATION

Date of Birth: July 10, 2003
Age: 22 years old
Gender: Female
Citizenship: Filipino
Religion: Roman Catholic
Civil Status: Single

+639510054702/
+639152023730

aidelajusta@my.cspc.edu.ph

Himaao Norte, Pili, Camarines Sur, 4418

SKILL

Computer Literate
Computer Vision (YOLO, OpenCV)
HTML & CSS

LANGUAGE

Bikol
Tagalog
English

EDUCATION

CAMARINES SUR POLYTECNIC COLLEGES
San Miguel, Nabua, Camarines Sur
Bachelor of Science in Computer Science
2021- PRESENT

PILI NATIONAL HIGH SCHOOL
Pawili, Pili, Camarines Sur
Senior High School
General Academic Strand
2019 - 2021

V. BAGASINA SR. MEMORIAL HIGH SCHOOL
Himaao, Pili, Camarines Sur
Junior High School
2015 - 2019

HIMAAO ELEMENTARY SCHOOL
Himaao, Pili, Camarines Sur
2009-2015

SEMINARS ATTENDED

- 11th Bicol Youth Congress in Information Technology (BYCIT) - 2023
- 12th Bicol Youth Congress in Information Technology (BYCIT) - 2024
- 13th Bicol Youth Congress in Information Technology (BYCIT) - 2025

CAREN JOY L. EPRES

PERSONAL INFORMATION

Date of Birth: June 12, 2003
Age: 22 years old
Gender: Female
Citizenship: Filipino
Religion: Roman Catholic
Civil Status: Single

+639637660444
caepres@my.cspc.edu.ph
Z7, Sta Cruz, Libon, Albay 4507

SKILLS

Computer Literate
Computer Vision
HTML & CSS (Back End)

LANGUANGE

TAGALOG
BICOL
RINCONADA
ENGLISH

EDUCATION

CAMARINES SUR POLYTECNIC COLLEGES
San Miguel, Nabua, Camarines Sur
Bachelor of Science in Computer Science
2022 - PRESENT

NABUA NATIONAL HIGH SCHOOL
San Miguel, Nabua, Camarines Sur
Senior High School
Technical-Vocational-Livelihood (ICT)
2020 - 2022

NABUA NATIONAL HIGH SCHOOL
San Miguel, Nabua, Camarines Sur
Junior High School
2016 - 2020

STA CRUZ ELEMENTARY SCHOOL
Sta Cruz, Libon, Albay
2016

SEMINARS ATTENDED

- 11th Bicol Youth Congress in Information Technology (BYCIT) - 2023
- 12th Bicol Youth Congress in Information Technology (BYCIT) - 2024
- 13th Bicol Youth Congress in Information Technology (BYCIT) - 2025

MARIA ANGELA MATUBIS

PERSONAL INFORMATION

Date of Birth: June 12, 2003
Age: 22 years old
Gender: Female
Citizenship: Filipino
Religion: Roman Catholic
Civil Status: Single

+639109412839
mamatubis@my.cspc.edu.ph
Z7, Sta Cruz, Libon, Albay 4507

SKILLS

Computer Literate
Computer Vision
HTML & CSS (Back End)
Python (Front End)

LANGUANGE

TAGALOG
BICOL
RINCONADA
ENGLISH

EDUCATION

CAMARINES SUR POLYTECNIC COLLEGES
San Miguel, Nabua, Camarines Sur
Bachelor of Science in Computer Science
2022 - PRESENT

NABUA NATIONAL HIGH SCHOOL
San Miguel, Nabua, Camarines Sur
Senior High School
Humanities and Social Sciences
2020 - 2022

NABUA NATIONAL HIGH SCHOOL
San Miguel, Nabua, Camarines Sur
Junior High School
2016 - 2020

STA CRUZ ELEMENTARY SCHOOL
Sta Cruz, Libon, Albay
2016

SEMINARS ATTENDED

- 11th Bicol Youth Congress in Information Technology (BYCIT) - 2023
- 12th Bicol Youth Congress in Information Technology (BYCIT) - 2024
- 13th Bicol Youth Congress in Information Technology (BYCIT) - 2025