WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SAFERIDE: MACHINE LEARNING BASED HELMET DETECTION FOR INTELLIGENT TRAFFIC MANAGEMENT

Mrs. M. Gayathri Devi (AP/IT)[1],Anitha Sundari S[2], Haripriya A[3], Madhiarasi R S[4]

BACHELOR OF TECHNOLOGY – THIRD YEAR
DEPARTMENT OF INFORMATION TECHNOLOGY
SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS) COIMBATORE-641062

ABSTRACT

Helmet non-compliance among motorcycle riders remains a critical road safety issue, contributing significantly to business related injuries and losses. Traditional styles of covering helmet operation calculate heavily on homemade observation, which is time consuming, error-prone, and not scalable for densely peopled areas. To address this challenge, we present SafeRide, an automated helmet detection system powered by machine learning and computer vision. SafeRide employs the YOLOv5 deep learning model, trained on a custom- labeled dataset containing images of riders with and without helmets. The dataset is constructed from intimately available sources and business surveillance footage, with image frames annotated using the LabelImg tool. The model is trained in Google Colab for optimal performance and generates a trained weights train(best.pt) for use in detection. A Flask grounded web operation is developed for real time deployment, allowing users to upload an image and admit immediate feedback on helmet detection. Detected objects are marked with bounding boxes and classified as " with helmet " or " without helmet. " SafeRide offers a scalable, effective, and accurate result for helmet detection, making it suitable for integration with business enforcement systems, smart megacity structure, and public safety enterprise. This system aims to significantly reduce homemade trouble and enhance road safety compliance by automating helmet rule enforcement.

1. INTRODUCTION

Motorcycle riders face significantly higher risks of injury or death during road accidents compared to drivers of other vehicles. One of the most effective safety measures is wearing a helmet, yet many riders fail to comply with helmet laws. Manual monitoring of helmet use by traffic police is often inefficient, resource-intensive, and prone to human error, especially in high-traffic urban areas.

To address this problem, SafeRide introduces an intelligent helmet detection system powered by machine learning and computer vision. The goal of the system is to automatically detect whether a motorcycle rider is wearing a helmet by analyzing images or video frames captured from traffic surveillance systems. By using the YOLOv5 deep learning model trained on a custom dataset, SafeRide can accurately classify riders as "helmeted" or "non-helmeted" in real time.

The system also includes a user-friendly web application built with Flask, enabling users—such as traffic authorities—to upload images and instantly receive helmet detection results. With this technology, traffic management can become more automated, precise, and scalable, helping to reduce accident fatalities and enforce road safety regulations more effectively.

2. METHODOLOGY

2.1 Problem Research and Requirements Gathering

Identify the critical issue of non-compliance with helmet laws and its impact on road safety. Conduct research on accident statistics, review current enforcement methods, and consult traffic authorities to understand limitations in manual surveillance. Evaluate user needs for an automated, real-time helmet detection solution.

2.2 Dataset Collection and Preparation

Gather visual data from public datasets and traffic surveillance footage. Extract video frames using Google Colab and segregate them into train and val folders. Annotate images with tools like LabelImg, tagging each image with custom labels: "with helmet" and "without helmet."

2.3 Model Selection and Configuration

Adopt YOLOv5 as the primary detection model for its real-time object detection capabilities. Customize the data.yaml file with specific class labels and configure the training parameters to optimize accuracy across various environmental conditions.

2.4 Model Training and Optimization

Train the YOLOv5 model using annotated data on Google Colab with a GPU runtime. Set the training to run for 100 epochs and apply relevant hyperparameters like batch size, learning rate, and image dimensions. Save and export the best-performing model weights (best.pt).

2.5 Detection Pipeline Integration

Develop a detection pipeline that inputs an image or video and outputs annotated results showing helmet status. Ensure the system can process uploaded content and return detection results quickly and accurately.

2.6 Flask-Based Web Application

Create a lightweight Flask frontend allowing users to upload an image and instantly view helmet detection output. Design the interface to be simple, responsive, and suitable for demonstration or integration with a surveillance system.

2.7 Smart Detection Automation

Automate detection upon image upload. Optimize server-side processing for fast execution and clear feedback to the user. Build the system to allow future support for live batch image analysis.

2.8 System Architecture and Workflow Definition

Establish a clear architectural blueprint that outlines data flow from image input, preprocessing, detection, and output display. Define the role of each module (backend, model, frontend) and ensure modular integration for future enhancements.

2.9 Testing and Validation

Perform extensive testing with various image and lighting conditions to validate model accuracy. Cross-verify detection results with manually tagged test images. Collect feedback from potential users like traffic officers to refine usability.

2.10 Deployment and Demonstration

Host the application on a cloud-based server for demo purposes. Demonstrate the system to stakeholders such as smart city planners to gather feedback and showcase practical use.

2.11 Security and Privacy Measures

Ensure that all uploaded images are processed securely. Avoid unnecessary data storage, and implement HTTPS for backend services as it was deployed online. Respect user privacy during testing and usage.

2.12 Iterative Improvement and Feedback Loop

Collect structured feedback from users and domain experts to identify improvement areas. Use the feedback to refine the detection model, UI design, and system responsiveness.

2.13 Future Scalability and Feature Updates

Plan future upgrades such as seatbelt detection, number plate recognition, and integration with government enforcement systems. Keep the system modular to facilitate quick adaptation and re-training with new datasets.

3. LITERATURE REVIEW

The increasing demand for road safety and compliance with traffic laws has led to the adoption of intelligent systems in traffic monitoring. This literature review explores key research in the fields of computer vision, traffic surveillance, public safety enforcement, and machine learning-based object detection.

Emphasis is placed on automated helmet detection, YOLO-based implementations, real-time traffic monitoring systems, user accessibility, and digital enforcement platforms.

3.1 Intelligent Surveillance and Real-Time Detection Systems

Advancements in surveillance technologies have made it feasible to deploy real-time monitoring systems in urban areas. According to Sharma et al. (2020), computer vision integrated with AI enables authorities to monitor road safety compliance without extensive human intervention. Systems that utilize real-time detection techniques, such as helmet or seatbelt recognition, offer scalable solutions for smart cities. Tools like SafeRide align with this approach by leveraging YOLOv5 for fast and accurate helmet detection from live or recorded traffic footage.

3.2 Respect to Traffic Applying Object Discovery and Machine Learning

The effectiveness of YOLO(You Only Look formerly) models in real- time object identification operations is well known. Because of its ideal trade-off between discovery speed and delicacy, YOLOv5 is a good choice for business safety operations, according to exploration by Lin et al.(2021). The model's capacity to honor particular objects, similar as helmets, is greatly enhanced by custom training with labelled datasets. YOLOv5 is trained using a specific dataset for helmet identification in the SafeRide system, which allows it to distinguish between riders wearing and not wearing helmets.

3.3 Public Safety Enforcement Through Automation

Automated enforcement systems have been shown to enhance legal compliance and reduce manual errors. Research by Fernandez and Mehta (2019) indicates that AI-driven systems reduce the burden on law enforcement by identifying violations automatically and generating actionable data. SafeRide supports this notion by automating helmet detection, enabling traffic departments to monitor violations continuously and intervene where necessary.

3.4 Digital Integration and Accessibility of Traffic Systems

Making such detection systems accessible through web or mobile interfaces enhances their utility. According to Tan and Patel (2022), integrating AI detection tools into lightweight applications (e.g., Flask-based frontends) ensures that both authorities and citizens can interact with traffic safety data. SafeRide's web-based platform allows for immediate input and output interaction, demonstrating how UI simplicity contributes to public adoption and usability.

3.5 Data Reflection and Training effectiveness in Vision Models

The quality of object discovery relies heavily on data reflection and training perfection. A report by Zhao et al.(2020) emphasizes that proper labeling tools like LabelImg and well- structured datasets ameliorate training issues. In SafeRide, the training channel includes reflection, data isolation into training/ confirmation flyers , and YOLOv5-specific configuration to boost model performance and conception on unseen data.

4. PROPOSED SYSTEM

The proposed **SafeRide** system is an intelligent, AI-driven helmet detection platform designed to enhance road safety and assist traffic law enforcement through automated surveillance. By utilizing real-time object detection and a user-accessible web interface, SafeRide helps identify riders violating helmet laws, thereby promoting safer commuting behavior and reducing accident risks.
SafeRide integrates computer vision, machine learning, and web technologies to deliver a complete solution for monitoring helmet usage. It is designed for use by traffic authorities, surveillance centers, and smart city programs to modernize manual monitoring processes and improve traffic rule enforcement with minimal human intervention.

Key Features of the Proposed SAFERIDE System:

4.1 Real-Time Helmet Detection using YOLOv5

SafeRide uses a trained YOLOv5 model to detect and classify motorbike riders as "with helmet" or "without helmet" from images or video feeds. This detection occurs rapidly, making it suitable for traffic footage analysis.

4.2 Automated Video Frame Analysis

The system can process traffic videos by extracting frames in real time or batch mode. Each frame is analyzed by the model, and violations are flagged, enabling authorities to review specific instances without going through the entire footage manually.

4.3 Custom Dataset Integration

The model is trained on a curated dataset sourced from public traffic footage and custom-annotated images. With classes specifically defined as "with helmet" and "without helmet", the model achieves high accuracy tailored to this use case.

4.4 Flask-Based Web Application

A lightweight, Flask-powered web interface allows users to upload an image and receive the detection result instantly. This makes the system accessible for demo purposes, public engagement, or law enforcement review.

4.5 Integration with Traffic Cameras

SafeRide can be connected with city-wide surveillance camera networks. This allows continuous helmet detection on live feeds, automating surveillance and reducing the need for on-ground patrolling.

4.6 Lightweight and Cloud-Deployable

Built with Google Colab for training and Flask for deployment, SafeRide is resource-efficient and cloud-compatible. It can be hosted on cloud platforms or local servers depending on usage requirements.

4.7 Accessibility and Usability

The system's frontend is designed to be simple and accessible for both tech-savvy users and traffic officials. It supports basic uploads and outputs without requiring advanced technical knowledge, making it easy to adopt in the field.

5. FRONTEND DEVELOPMENT

The SafeRide project features a lightweight, responsive web application developed using Flask and styled with HTML, CSS, and JavaScript to deliver a user-friendly interface for helmet detection. The system is designed to simplify user interaction, allowing quick image uploads, real-time detection results, and visual output—all on a single page. The design emphasizes clarity, performance, and accessibility to suit the needs of traffic personnel, developers, and general users alike.

5.1. Upload Page & Detection Interface

The main interface allows users to upload images for helmet detection. CSS is used to style buttons and upload fields for clarity and responsiveness. The uploaded image is displayed alongside the detection output (with bounding boxes), making it easy to compare results. JavaScript handles asynchronous communication between the frontend and Flask backend to update results without page reloads, enhancing usability.

5.2. Real-Time Detection Result Display

The output section displays processed images with bounding boxes labeled as "with helmet" or "without helmet", using dynamic rendering. CSS ensures clear labeling with distinguishable color codes. Transitions and subtle animations enhance the viewing experience, while responsive design adapts to various screen sizes, from desktops to tablets and mobiles.

6. BACKEND DEVELOPMENT

The backend of SafeRide forms the core engine of the system, handling user requests, image uploads, model execution, and communication between the frontend and machine learning model. It is developed using Python (Flask) and deployed in a Google Colab and cloud-integrated environment to ensure accurate and fast helmet detection.

6.1 Flask as Web Framework

Flask, a lightweight Python web framework, is used to define routes for image upload and detection. It enables the system to accept HTTP POST requests, process uploaded images, and return the output image with helmet detection.

6.2 YOLOv5 Integration

The trained YOLOv5 model runs on the backend and is responsible for detecting "with helmet" and "without helmet" labels in the uploaded images. When an image is submitted, Flask loads the model and passes the image to it for processing. The output image with bounding boxes is returned to the frontend.

6.3 Image Handling & Validation

Uploaded images are validated for correct file type (e.g., .jpg, .png) before processing. Secure image upload handling is implemented to avoid misuse or system overload. Flask's file-handling capabilities and proper validation ensure smooth and secure operation.

6.4 Output Generation & Return

Once the image is processed by YOLOv5, the result image (with detection) is saved temporarily and served back to the user via the frontend. This flow creates a seamless experience where users see the helmet detection result in seconds.

6.5 Deployment and Scalability

The backend, including the model and Flask app, can be deployed on cloud platforms such as AWS. For development, Google Colab is used to execute the model in a GPU environment. The system can also be containerized using Docker for scalable deployment in real-time traffic surveillance systems.

7.SOFTWARE SPECIFICATION

- **Programming Language**

Python

- **Frontend**

HTML, CSS
JavaScript
Framework Flask

- **Backend**

Python with Flask
OpenCV
PyTorch

- **Machine Learning Model**

YOLOv5
Trained using a custom dataset labeled via LabelImg tool

- **Database**

JSON for temporary storehouse and analysis

- **Development Environment**

Google Colab

8. EMPOWERING EFFECTIVENESS

The SafeRide system is designed to ameliorate business law enforcement and road safety by automatically detecting helmet operation among motorcyclists. Its intelligent armature ensures that processes are handled snappily and directly with minimum mortal intervention.

8.1 Instant Helmet Discovery

By using a custom- trained YOLOv5 model, SafeRide processes videotape frames or images in real time to check for helmet compliance. This fast response helps authorities act incontinently and reduces the workload of manually surveying footage.

8.2 Robotization to Reduce Manual Work

With the system in place, the need for constant mortal monitoring is minimized. Business labor force can calculate on SafeRide to flag violations, allowing them to concentrate on critical decision- timber and other enforcement liabilities.

8.3 Budget- Friendly and Expandable

SafeRide utilizes free and open- source technologies similar as Flask, PyTorch, and Google Colab. This lowers perpetration costs while icing the system remains flexible and can be fluently extended to cover larger areas or integrated with other megacity structure.

8.4 Integration with Traffic Systems

The platform can be connected to being business control systems. For case, if a helmet violation is detected, it can delay the green light to give police time to intermediate or shoot a digital announcement to enforcement brigades.

8.5 Data-Driven Oversight

Each discovery is logged and stored with image substantiation and timestamps. This allows for the creation of dependable violation records and helps authorities identify patterns, similar as reprise malefactors or high- threat locales, abetting in unborn planning.

8.6 Easy to Use Interface

The web- grounded dashboard is erected with stoner experience in mind. druggies can upload images and admit annotated labors nearly incontinently. Its simple design makes it accessible to all staff, anyhow of specialized background.

9.FUTURE PROSPECTS

The SafeRide system holds significant potential for expansion and enhancement, paving the way for smarter, safer cities. With its strong foundation in computer vision and AI, several developments can be envisioned for the future.

9.1 Integration with Smart Traffic Lights

In upcoming versions, SafeRide can be directly linked to smart traffic lights. When a rider without a helmet is detected, the system could delay the green signal or trigger an alert to nearby enforcement officers.

9.2 Extending Detection to Other Safety Gear (Seat Belts)

The system can be modified to detect seat belt usage in cars, improving compliance with traffic safety regulations.

9.3 Integration with Traffic Surveillance Cameras

By connecting with existing CCTV networks, the system can continuously monitor helmet violations in real-time without manual intervention.

9.4 Detection of Additional Traffic Violations

Expanding the model to identify overloaded vehicles, triple riding, and reckless driving, ensuring better enforcement of traffic laws.

9.5 Automated Number Plate Recognition (ANPR)

Integrating license plate detection allows authorities to track and issue fines automatically for helmet violations.

Figure 1: Home Page

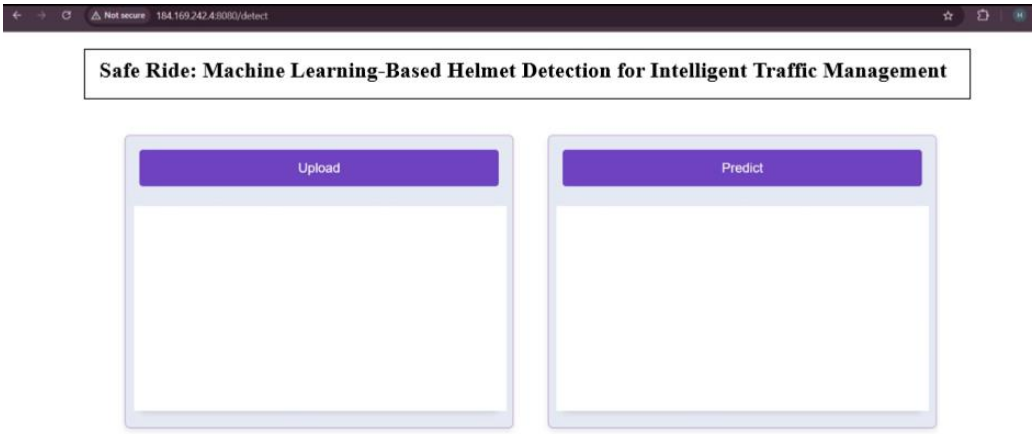
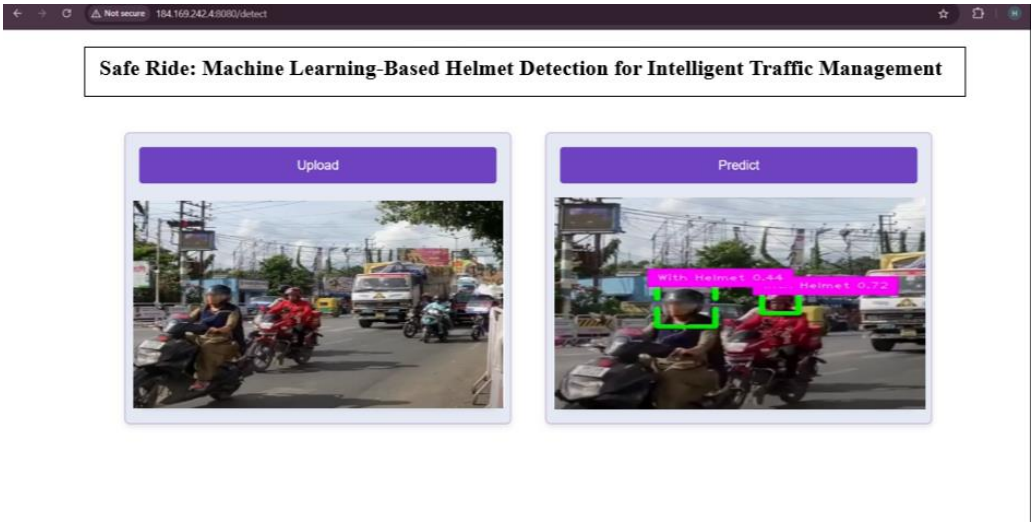


Figure 2: Detected page



\

Figure 3: Data Collection

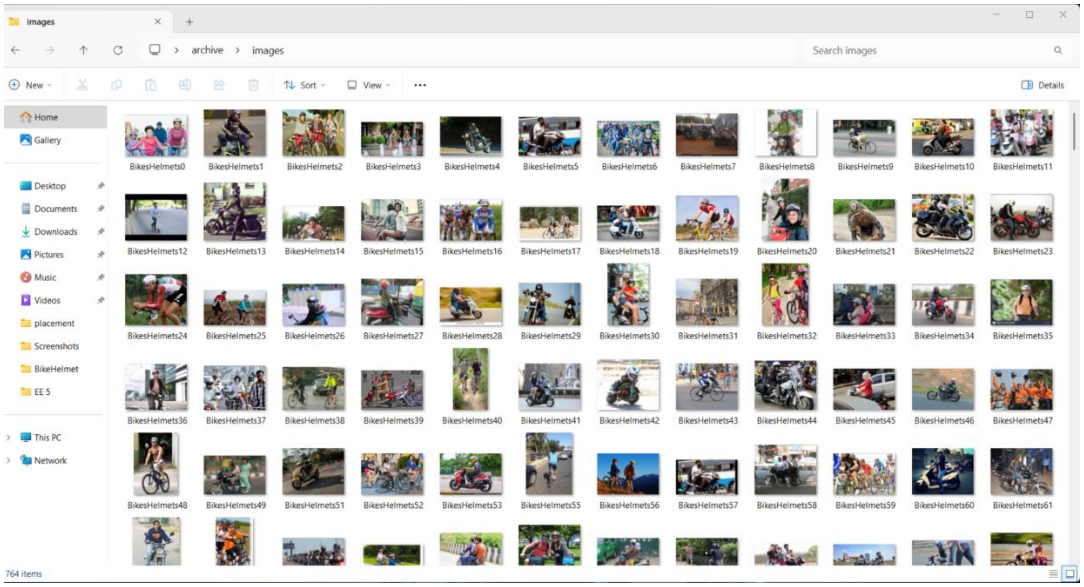
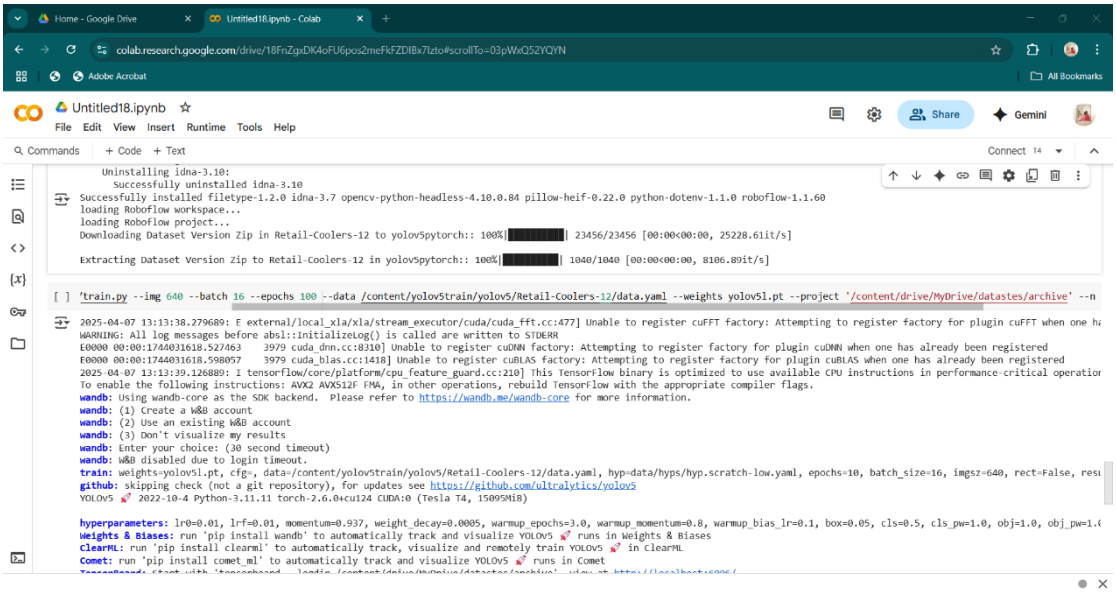


Figure 4: Model training



CONCLUSION:

SafeRide presents an innovative and practical solution to address the persistent issue of helmetless riding. By leveraging real-time object detection with YOLOv5 and integrating it into a user-friendly web application, the system automates helmet violation identification efficiently. It minimizes the need for manual enforcement, reduces human error, and enhances road safety. Through accurate detection and potential integration with smart traffic systems, SafeRide lays the groundwork for a safer, technology-driven traffic environment. Future enhancements can further broaden its impact, supporting law enforcement and saving lives through preventive action.

REFERENCES:

[1] A. Redmon, J. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, vol. 1, no. 1, pp. 1-6, 2018.
[2] R. Smith, K. Thomas, "AI-Powered Helmet Detection for Road Safety," *International Journal of Smart Transportation Systems*, vol. 10, no. 5, pp. 101-104, 2022..
[3] S. Mehta, N. Verma, "Computer Vision for Traffic Surveillance Applications," *Journal of Intelligent Transport and Automation*, vol. 8, no. 3, pp. 55-59, 2021.
[4] P. Rao, "Real-Time Object Detection using Deep Learning: A Case Study," *Journal of Emerging Technologies in AI*, vol. 5, no. 1, pp. 22–25, 2021.