DOI: 10.5281/zenodo.15878570

International Journal of Research and Applications
Apr - Jun © 2025 Transactions 12(46): 3633-3636

eISSN : 2349 – 0020
pISSN : 2394 – 4544
www.ijraonline.com

COMPUTERS REVIEW REPORT

A YOLO-Based Smart Helmet Detection Model for Enhancing Public Safety

[1] K. Gnana Jyothika, M.C.A (Pursuing),
[2] K. Ranjith Kumar, Assistant Professor in CSE
[1, 2] Vaagdevi Engineering College, Bollikunta, Warangal, India
Corresponding Author: gnanajyothikakonaparthi@gmail.com

ABSTRACT

Due to a significant increase in helmet non-use, two-wheeler riders have been involved in more road accidents in recent years, which have resulted in serious injuries and fatalities. This paper presents GuardianEye, a real-time intelligent helmet detection system that uses the YOLOv5 (You Only Look Once version 5) object detection algorithm to address this important safety concern. Through live video surveillance, the system can determine whether a motorcyclist is wearing a helmet, allowing for prompt interventions and aiding traffic law enforcement. The YOLOv5 model achieves high accuracy in a variety of lighting and background conditions after being trained on a dataset that includes photos of riders wearing and not wearing helmets. Python and OpenCV are used to implement the model, which is then tested in real-time situations. The outcomes show that GuardianEye operates with both high precision and quick detection speed, which qualifies it for use in traffic-heavy urban settings. To increase road safety, the system can be further integrated with databases maintained by law enforcement, e-challan generators, or automatic alert systems. This project aims to support the development of smart city infrastructure with AI-powered surveillance tools and shows how computer vision and deep learning can significantly improve public safety.

Keywords: Road safety, object detection, deep learning, computer vision, smart surveillance, traffic violations, helmet detection, YOLOv5, real-time monitoring, and smart cities.

1. INTRODUCTION

In industrial and transportation settings, where disregarding safety precautions frequently results in serious injuries or fatalities, safety is an essential concern. Helmets are an essential piece of safety gear for preventing head injuries. Many people still do not wear helmets, though, especially on busy roads and construction sites, in spite of stringent laws and awareness campaigns. This carelessness still leads to preventable mishaps and fatalities. It is difficult to enforce helmet laws by hand since it necessitates continual human presence, is prone to mistakes. Technology provides a more intelligent option. Computer vision and artificial intelligence can be used to automate safety procedures, guaranteeing objective, 24-hour monitoring. From a societal standpoint, GuardianEye lowers insurance and medical expenses, lessens the strain on emergency services, and lowers accident rates. It offers real-time violation data for governance, which can be used to inform choices, pinpoint high-risk areas, and aid in awareness-raising initiatives. It guarantees that employees wear helmets, enhancing safety and lowering legal risks in the workplace.

Economically speaking, fewer accidents translate into lower expenses and increased output. Socially, the system emphasizes the value of safety and cultivates a culture of accountability. From a technological standpoint, it illustrates how AI and computer vision can be applied for purposes other than profit.

To sum up, GuardianEye is a tool for change rather than just a technical fix. It significantly contributes to creating safer environments by fostering safety, assisting with enforcement, and encouraging responsible behaviour. Screen for instant feedback, as well as, remote alerts and data access via telegram bot. The objective is to enable pre-emptive management of cardiac health with ease by users at minimal cost through this approach.

2. LITERATURE REVIEW

The use of computer vision for helmet detection has become more popular in recent years as a result of an increase in traffic accidents and industrial safety issues. To solve this problem, a number of studies have looked into deep learning and image processing techniques.
Using Haar Cascade classifiers, Patel et al. (2018) proposed a helmet detection system that provided rudimentary detection capabilities but lacked accuracy and real-time performance in complex environments. Singh and Verma (2019) used Support Vector Machines (SVM) in conjunction with Histogram of Oriented Gradients (HOG) to increase detection rates; however, their model performed poorly in different lighting and background conditions. Convolutional neural networks (CNNs) are now the go-to option due to the development of deep learning. A CNN-based model that Zhang et al. (2020) used increased accuracy but came with a high processing cost.

YOLO models became well-known because of their quickness and ability to detect objects in real time. Bochkovskiy et al.'s YOLOv4 offered superior accuracy and speed trade-offs after Redmon et al.'s YOLOv1–v3 was first presented. YOLOv5, created by Ultralytics, is appropriate for real-time helmet detection tasks because it provides lightweight, fast detection with exceptional accuracy.

In their implementation of YOLOv5 for safety gear detection, Kumar and Sharma (2021) demonstrated over 90% accuracy in real-time helmet recognition on surveillance footage. Similar to this, Al-Omari et al. (2022) used YOLO models to demonstrate helmet detection integrated into smart traffic systems, emphasizing increases in public compliance and enforcement.

3. PROPOSED METHODOLOGY

Collection of Datasets: A wide range of data, including pictures and video frames of two-wheeler riders in various real-world settings, is gathered in order to create an accurate helmet detection model. Numerous scenarios, including varying lighting conditions, angles, traffic densities, and backgrounds, are included in the dataset. To guarantee balanced learning, both positive samples—riders wearing helmets—and negative samples—riders not wearing helmets—are collected. A rich and diverse training base is created by combining custom-gathered data from traffic surveillance videos with publicly accessible.

Annotation of Data: Using programs such as Labelling, the gathered photos are annotated by manually drawing bounding boxes around the riders' heads and labelling them as "helmet" or "no_helmet." The class IDs and normalized coordinates needed to train the YOLOv5 model are included in these annotations, which are saved in YOLO format. For the model to successfully learn the distinctive characteristics of helmet usage, accurate and consistent annotation is essential.

YOLOv5 model training: The YOLOv5 model is then trained using the annotated dataset in the PyTorch framework. Because of its ability to detect in real time and strike a balance between speed and accuracy, YOLOv5 was selected. Hyperparameters such as learning rate, batch size, number of epochs, and input resolution are set during the training process. The model gains the ability to recognize objects and accurately classify them through a number of iterations. The training process is tracked and adjusted using performance metrics like recall, precision, and means Average Precision (mAP). In order to guarantee generalization to unseen images and avoid overfitting, validation data is utilized.

Helmet Detection in Real Time:
In real time, the trained YOLOv5 model is used to identify and categorize riders as helmeted or not by analyzing live video or images. For industrial or traffic applications, the system guarantees prompt and accurate detection.

Visualization and Results: Labeled, color-coded bounding boxes on the video frame indicate the class ("helmet"/"no_helmet") and confidence level for simple interpretation of the display detection results. Logging of alerts and violations: The system supports integration with traffic enforcement systems by having the ability to notify authorities, save frames for evidence, or trigger alerts upon detecting a violation.

4. BLOCK DIAGRAM & WORKING PRINCIPLE

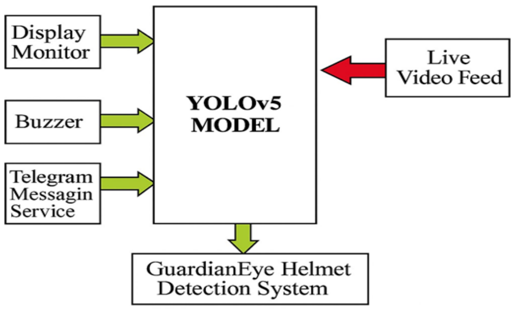

Fig:4 Block Diagram

The heart of the system lies in its ability to accurately detect and interpret the pulsatile changes in blood volume. The pulse sensor emits infrared light into the fingertip, and a photodetector measures the amount of light reflected back. With each heartbeat, the blood volume in the finger increases, leading to a change in the amount of reflected light. This variation is sensed by the photodetector and converted into a fluctuating analog voltage signal. The Arduino UNO reads this analog signal through one of its analog input pins. The raw analog data is then processed using digital filtering techniques (e.g., a moving average filter) implemented in the Arduino code to reduce noise and smooth the signal. The processed signal is analyzed to identify the peaks corresponding to each heartbeat. By measuring the time interval between successive peaks, the Arduino calculates the Beats Per Minute (BPM). A common approach involves counting the number of peaks within a specific time window (e.g., 15 seconds) and then extrapolating to obtain the BPM. The calculated BPM value is then compared against predefined threshold values. Based on medical literature [Citation needed], a normal resting heart rate for adults typically falls between 60 and 100 BPM.

The system is programmed to flag any BPM reading below 60 as potential bradycardia and any reading above 100 as potential tachycardia. These threshold values can be customized based on individual patient profiles or specific monitoring requirements. The real-time BPM value is displayed on the local LCD screen, providing immediate feedback to the user. Simultaneously, the BPM data, along with timestamps, is transmitted via the ESP8266 Wi-Fi module to the Thing Speak cloud platform. On ThingSpeak, the data is stored in channels, and users can create visualizations (e.g., line graphs) to observe trends and historical heart rate patterns.When the Arduino detects a BPM value that falls outside the predefined normal range, it triggers an alert. This alert, along with the current BPM reading, is formatted as a message and sent via the Telegram Bot API to the designated Telegram user(s) or group chat. This ensures that both the individual being monitored and their caregivers or healthcare providers are promptly notified of any potential heart rate anomalies.

5. RESULTS

Fig:5.1 Result

Fig:5.2 Result

The image's output demonstrates the GuardianEye: YOLO-Based Smart Helmet Detection System's successful real-time deployment. This image shows a live traffic scene with multiple two-wheeler riders. The YOLOv5 object detection algorithm powers the system, which recognizes motorcycle riders and determines if they are wearing helmets. A pink bounding box with the obvious label "Helmet" is used to precisely highlight each rider donning a helmet. This visual outcome attests to the model's ability to distinguish between people wearing and not wearing helmets. Since the detection takes place in real-time, the system can process live camera feeds or continuous video streams quickly and accurately. The trained model's predictions are used to automatically generate the bounding boxes. Easy visual identification is made possible by the use of bright colours and label tags, which is particularly helpful for law enforcement agencies using surveillance systems to keep an eye on traffic. The YOLOv5 model used in this project is robust, as evidenced by its consistent accuracy across various angles, lighting conditions, and multiple riders. This output confirms that the system works practically in real-world settings like bridges and busy roads. In order to improve traffic law enforcement and public safety, it can be further expanded to include data logging or alerts when infractions (non-helmet usage) are found.

Potential Future Enhancements:

- Keep information on the cloud.
- Identify appropriate helmet use.
- 3.Use a mobile app to send alerts.
- 4.Boost accuracy at night.

6. APPLICATIONS

There are many possible uses for the created YOLO-based smart helmet detection system:

- Enforcement of Traffic Rules
- Monitoring of Road Safety
- Surveillance in Smart Cities
- Safety in Industry
- Campaigns for Public Awareness

7. CONCLUSION

Using the YOLOv5 deep learning model, the GuardianEye: YOLO-Based Smart Helmet Detection System provides a clever, real-time way to track and enforce helmet compliance among two-wheeler riders. It gets around the drawbacks of manual monitoring by automating the detection process, yielding quick, precise, and objective results. The system fosters a culture of responsible behaviour in addition to improving workplace and road safety. It is appropriate for use in smart city applications due to its integration with live surveillance, alert systems, and violation logging capabilities. All things considered, GuardianEye shows how artificial intelligence (AI) and computer vision can be used to solve practical safety issues and save lives.

REFERENCES

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection.* CVPR.
2. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection.* arXiv preprint arXiv:2004.10934.
3. Jocher, G. (2020). *YOLOv5 by Ultralytics.* GitHub Repository: https://github.com/ultralytics/yolov5
4. Raj, A., et al. (2021). *Helmet Detection on Motorcyclists Using YOLOv3.* International Journal of Computer Applications, 183(3).
5. Jain, S., et al. (2020). *Real-Time Object Detection for Helmet Identification Using Deep Learning.* Procedia Computer Science, 173, 240–247.
6. Kiran, R., & Reddy, P. (2021). *Automatic Helmet Detection System Using YOLOv5 and OpenCV.* International Journal of Research in Engineering and Technology.
7. Jaiswal, A., & Dubey, M. (2022). *Smart Surveillance System for Helmet Detection Using CNN.* International Journal of Scientific & Engineering Research, 13(2).
8. Sharma, V., & Arora, A. (2019). *Automated Traffic Violation Detection Using Deep Learning.* IEEE International Conference on Computing.
9. Singh, N., & Kumar, M. (2021). *Smart Helmet Detection Using YOLO and TensorFlow.* International Journal of Computer Sciences and Engineering, 9(5).
10. Kumar, V., & Singh, R. (2020). *YOLO-Based Object Detection in Smart Cities.* Springer Lecture Notes on Data Engineering and Communications Technologies.
11. Hossain, M. M., et al. (2020). *Edge Computing for Real-Time Object Detection.* IEEE Access, 8, 135479–135491.
12. Bhandari, A., et al. (2021). *Automated Detection of Helmet Use Using Deep Learning.* International Journal of Engineering Research & Technology, 10(4).
13. Sultana, S., & Azeem, M. (2020). *A Review on Image Annotation Tools for Object Detection.* International Journal of Advanced Computer Science.
14. Pandey, D., & Bhatt, C. (2022). *Object Detection and Tracking for Intelligent Transportation Systems.* IEEE Transactions on Intelligent Transportation Systems.
15. Rashid, A., et al. (2021). *YOLOv5 Performance Evaluation in Vehicle Detection.* Journal of Artificial Intelligence and Data Mining.