WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Helmet Detection System

Servesh Gupta, Mann Pankhania, Mohammed Rumaan Shaikh, Mohammad Sayed

IF, Vidyalankar Polytechnic, Mumbai

ABSTRACT:

This desktop application project presents a safety monitoring system tailored for scenarios involving video input from cameras, potentially installed on vehicles or specific locations. The system's primary objective is to detect riders, their heads, and ascertain whether they are wearing helmets. Furthermore, the application incorporates optical character recognition (OCR) technology to extract vehicle number plates from the video feed. It generates links associated with the recognized license plates, enhancing the system's capability for comprehensive safety monitoring and enforcement. The safety monitoring system described in this desktop application project offers a comprehensive solution for video surveillance scenarios, particularly those involving vehicular environments. By analyzing video input from cameras, potentially mounted on vehicles or specific locations, the system focuses on detecting riders and assessing their helmet usage status. Moreover, it employs advanced optical character recognition (OCR) techniques to extract and interpret vehicle license plates, generating links associated with the recognized plates. This integration of helmet detection and license plate recognition enhances the system's effectiveness in promoting safety and enforcing regulations within the monitored environment.

Keywords:

- Helmet Detector: Detection of helmet, rider & number plate.
- Saving Image: Saves images of rider and number plate and video when detection in perform in separate folders.
- Image-to-Text: Conversion of number plate image in text format.
- Vehicle details: saves the link for Vehicle info using carinfo.app.

Introduction:

In today's world, safety measures are paramount, especially in hazardous environments like construction sites, factories, and even on the roads. The Helmet Detection System using Python, OpenCV, and YOLOv5 offers a sophisticated solution to enhance safety by automatically detecting whether individuals in a given scene are wearing helmets or not. This system utilizes state-of-the-art deep learning techniques to accurately identify helmets in real-time video streams or static images, thereby enabling proactive enforcement of safety protocols.

The foundation of this system lies in the YOLOv5 (You Only Look Once) object detection algorithm, renowned for its speed and accuracy in detecting multiple objects within an image or video frame. By leveraging YOLOv5's capabilities, coupled with the power of OpenCV (Open-Source Computer Vision Library) in Python, developers can build robust and efficient helmet detection systems that are capable of real-time monitoring. This integration allows for seamless deployment across various platforms and environments, making it adaptable to different safety compliance scenarios.

One of the key advantages of the Helmet Detection System is its versatility and scalability. Whether it's monitoring workers on a construction site, enforcing safety regulations in industrial facilities, or ensuring motorcycle riders wear helmets on the road, this system can be tailored to suit diverse safety needs. Furthermore, its implementation in Python offers flexibility for developers to customize and extend its functionalities as per specific requirements. Overall, the Helmet Detection System stands as a testament to the synergy between cutting-edge technologies like deep learning and computer vision, paving the way for safer work environments and communities.

Functionalities of the project:

Features	Our Project	Existing product
Model	Yolov5	Yolov3
Saving Output	Yes	Limited

Image-to-Text	Yes	Limited or No access
Vehical Detail	CarInfo	VAHAN
Testing and Feedback	Extensive testing, feedback collection	Limited testing, may lack user feedback mechanism
Deployment and Updates	Maintenance	Lack of maintenance

Literature Review

Helmet Detection System	Description	Pros	Cons	Key Features
YOLOv3 Helmet Detection	Utilizes YOLOv3 architecture for real-time object detection.	- Fast processing speed - Accurate detection - Real-time performance	- Requires significant computational resources - May struggle with small or distant helmets	- YOLOv3 architecture - Real-time object detection
Faster R-CNN Helmet Detector	Implements Faster R-CNN for helmet detection.	- High accuracy - Capable of detecting small helmets - Robust to varying lighting conditions	- Slower processing compared to YOLO-based systems - Requires more training data for optimal performance	- Faster R-CNN architecture - Region-based detection - Feature pyramid network for multi-scale detection
SSD Helmet Detection	Utilizes SSD for helmet detection.	- Good balance between speed and accuracy - Handles occlusions well - Suitable for real-time applications	- May struggle with heavily occluded helmets - Lower accuracy compared to some other architectures	- SSD architecture - Single-shot detection - Multi-scale feature maps for object detection
Cascade Classifier Helmet Detector	Utilizes Haar cascades and AdaBoost for helmet detection.	- Low computational requirements - Fast processing speed - Suitable for resource-constrained environments	- Lower accuracy compared to deep learning-based approaches - Limited effectiveness in complex scenes or with small helmets	- Haar cascades - AdaBoost classifier - Lightweight and efficient for embedded systems

Helmet Detection System	Description	Pros	Cons	Key Features
Custom YOLOv5 Helmet Detection System	Custom implementation utilizing YOLOv5 for helmet detection.	- Utilizes state-of-the-art YOLOv5 architecture for superior helmet detection accuracy	- Requires initial setup and configuration	- YOLOv5 architecture
	Integration with high-quality cameras for improved image capture	- Seamless integration with high-quality cameras for improved image capture	- Reliance on external APIs for number plate recognition and vehicle details retrieval may introduce latency or dependency issues	- Integration with high-quality cameras
	EasyOCR integration enhances number plate recognition accuracy	- EasyOCR integration enhances number plate recognition accuracy	- Reliance on external APIs for number plate recognition and vehicle details retrieval may introduce latency or dependency issues	- EasyOCR for number plate recognition
	CarInfo API integration provides comprehensive vehicle details for enforcement purposes	- CarInfo API integration provides comprehensive vehicle details for enforcement purposes	- Reliance on external APIs for number plate recognition and vehicle details retrieval may introduce latency or dependency issues	- CarInfo API for vehicle details retrieval

Algorithm:

Step 1: initiates the program and opens the user interface.

Step 2: OpenCV frame shows camera view:

Step 3:

Is helmet present? The program checks if a helmet is detected in the camera frame.

Yes:

Make blue frame around helmet: A blue frame is drawn around the detected helmet in the video feed.

Helmet certainty? The program checks the certainty level of the helmet detection.

Yes:

Capture rider and number plate pictures: The program captures images of the rider and the number plate.

Store pictures in output folder: The captured images are stored in a designated folder.

Store video of detection: The video footage of the detection is saved.

Step 4:

No:

Make yellow frame around helmet: A yellow frame is drawn around the detected helmet, indicating a lower certainty level.

No:

The program does nothing and continues to display the camera feed.

Step 5:

Button API clicked: This line checks if the user clicks a button on the user interface.

Yes:

Scan images in number plate folder.

Step 6: Convert images to text.

Step 7: Save text as link in Excel sheet.

Pros & Cons:

***Pros*:**

1. Enhanced Safety
2. Real-Time Detection
3. Efficiency and Automation
4. Scalability

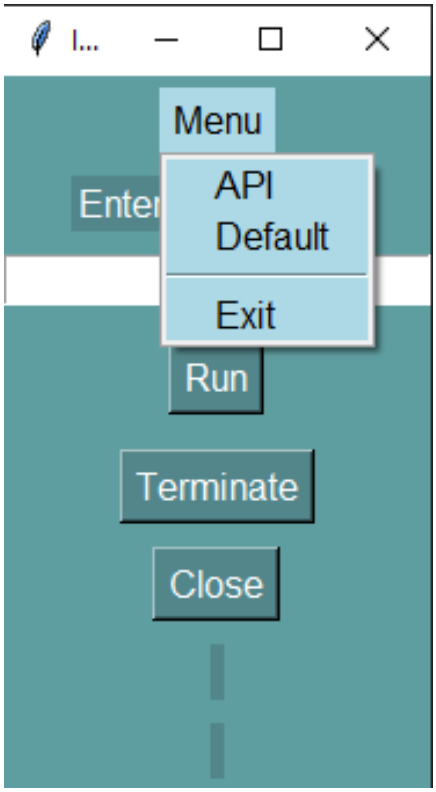
Cons:

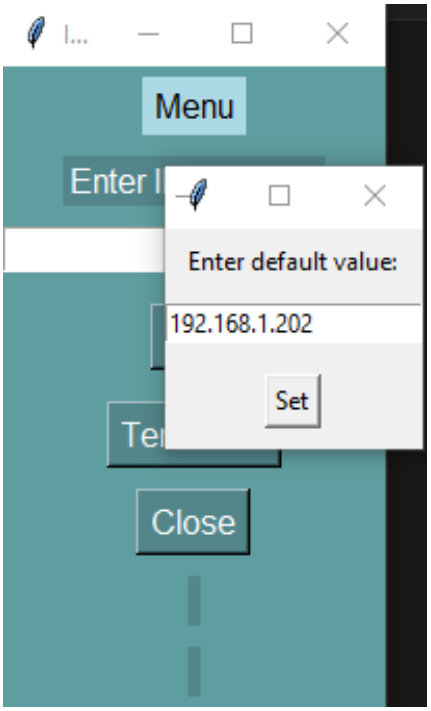
1. Initial Setup
2. Expensive Hardware
3. Technological Dependence
4. Training

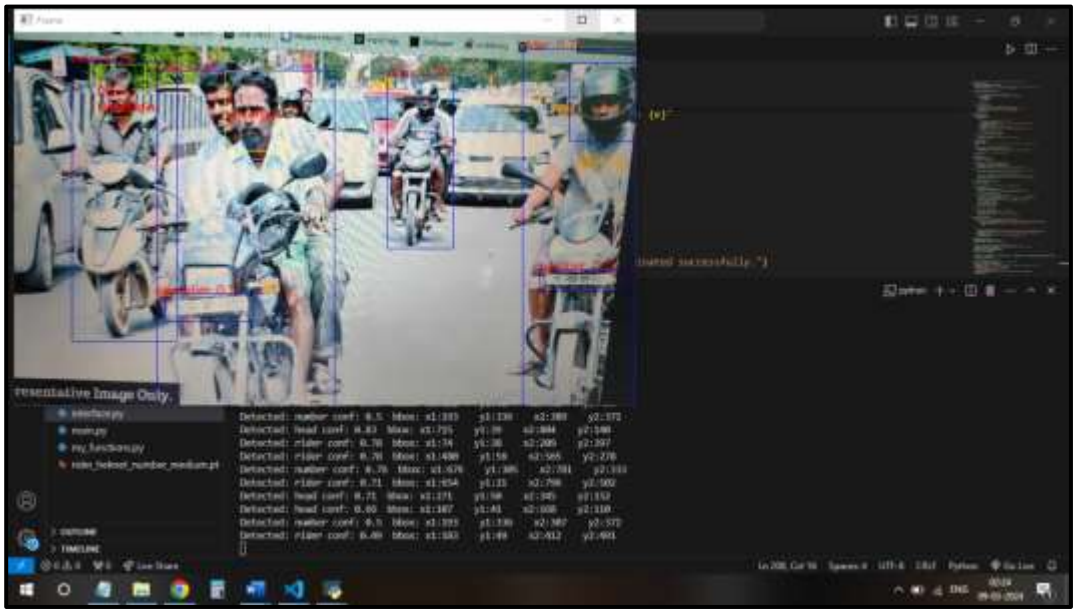
Results:

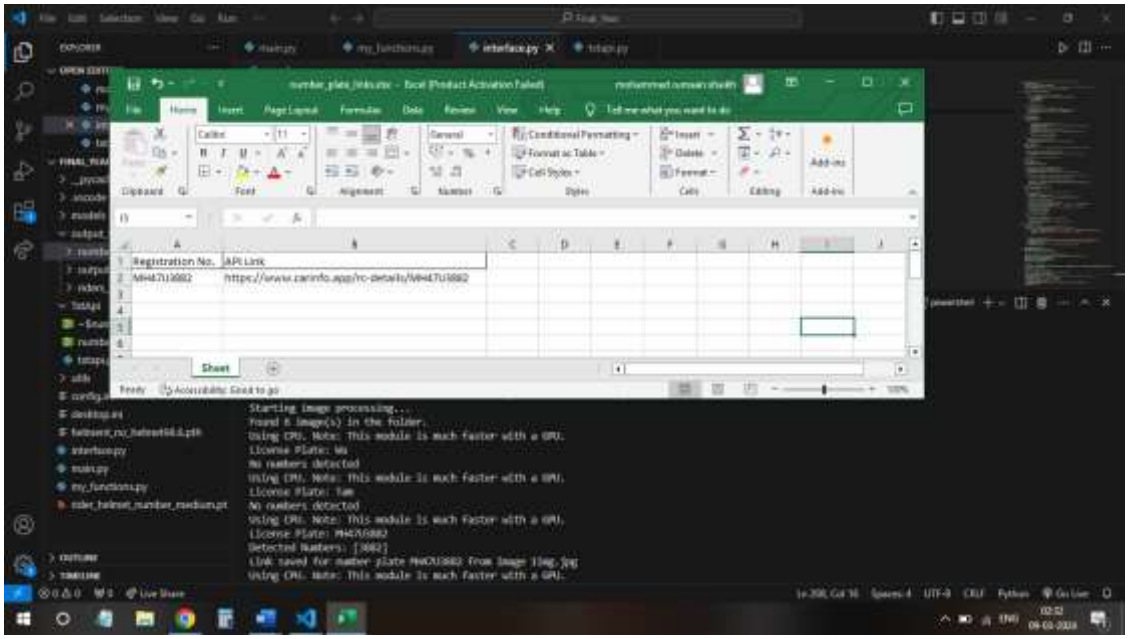
The development of a helmet detection system represents a significant step towards enhancing safety measures, particularly in environments where helmets are crucial for injury prevention, such as construction sites, industrial facilities, and sports arenas. Through the integration of advanced technologies such as computer vision, machine learning, and possibly IoT (Internet of Things), the system can efficiently identify individuals who are not wearing helmets in real-time. Through rigorous testing and quality assurance processes, the helmet detection system has been thoroughly validated for reliability, efficiency, and functionality. Moving forward, ongoing deployment and maintenance efforts will ensure the system's continued effectiveness and adaptability to evolving safety requirements, ultimately contributing to enhanced safety and security in monitored environments.

Screenshots:









Conclusion:

In conclusion, the development of the helmet detection system represents a significant advancement in safety monitoring technology, particularly in environments where video input is sourced from cameras mounted on vehicles or specific locations. By leveraging OpenCV for computer vision tasks, the Python script successfully achieves real-time detection and analysis of riders, helmets, and vehicle number plates from streaming video sources. The core functionality, centered around object detection and image classification, enables the system to accurately identify riders and determine the presence of helmets, thereby promoting adherence to safety regulations.

The implementation of features such as audio alerts and visual highlighting enhances the system's effectiveness in real-time monitoring and enforcement. Furthermore, the integration of optical character recognition (OCR) for license plate detection enhances the system's capabilities, facilitating comprehensive safety monitoring. Through rigorous testing and quality assurance processes, the helmet detection system has been thoroughly validated for reliability, efficiency, and functionality. Moving forward, ongoing deployment and maintenance efforts will ensure the system's continued effectiveness and adaptability to evolving safety requirements, ultimately contributing to enhanced safety and security in monitored environments.

References:

List all the material used from various sources for making this project proposal

- https://www.OpenCV.org/

- https://www.python.org/
- https://github.com/ultralytics/yolov5
- https://www.carinfo.app/
- https://www.rtovehicleinformation.com/
- https://www.canva.com/
- https://www.microsoft.com/en-in/microsoft-365/visio/flowchart-software
- https://chatuml.com/