HELMET DETECTION IN TWO-WHEELER

MINOR PROJECT REPORT

By

VARUN PRAKASH(RA2211003010632)
JHANVI SINGH(RA2211003010625)
NAVYA MUDGAL(RA2211003010644)

Under the guidance of

Dr. Aswathy K Cherian

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in **DEPARTMENT OF COMPUTING TECHNOLOGIES**

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Helmet Detection in Two-Wheeler**" is the bonafide work of **Varun Prakash (RA2211003010632), Jhanvi Singh (RA2211003010625)** and **Navya Mudgal (RA2211003010644)** who carried out the work under my supervision.

SIGNATURE

DR. ASWATHY K CHERIAN
Assistant Professor
Department of Computing Technologies
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

This project focuses on the implementation of a helmet detection system using the YOLOv3 pre-trained model in Python for backend processing and HTML/CSS for frontend visualization. The system aims to accurately detect the presence or absence of safety helmets in real-time scenarios, emphasizing the importance of safety in high-risk environments such as construction sites and sports arenas. Leveraging OpenCV and Flask, the project provides a user-friendly interface for prompt decision-making and interventions. Thorough testing and deployment on a web server ensure accessibility across diverse platforms, contributing to a safer working and recreational environment for individuals in various sectors.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We extend my gratitude to our **HoD Dr. M. Pushpalatha, Professor and Head, Department of Computing Technologies** and the Departmental colleagues for their Support.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G, Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our Course project Faculty **Dr. Aswathy K Cherian, Assistant Professor, CTECH,** for her assistance, timely suggestion and guidance throughout the duration of this course project.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

1. INTRODUCTION

In various high-risk settings, the adherence to safety helmet usage stands as a crucial factor in averting severe head injuries and accidents. The utilization of Python's robust image processing libraries, particularly OpenCV, serves as a fundamental tool in ensuring efficient image analysis and precise helmet detection.

Motivation:

This project is motivated by the critical necessity to bolster safety measures across diverse sectors. Preventing avoidable accidents and injuries remains a top priority, making the development of an advanced yet user-friendly helmet detection system imperative.

Objective:

The primary objective of this project is to establish a streamlined system that seamlessly integrates Python's backend capabilities with an intuitive HTML/CSS frontend for real-time helmet detection. The aim is to create an inclusive solution that accurately identifies the presence or absence of helmets, thereby fostering a culture of safety and accountability in various domains.

Problem Statement:

The project addresses the challenge of designing an integrated and responsive solution for detecting safety helmets within dynamic environments. Emphasizing the swift and effective presentation of critical information is key to reducing the likelihood of accidents and injuries.

Challenges:

Overcoming variations in lighting conditions, accommodating diverse helmet designs, and ensuring real-time processing present significant challenges. Simultaneously, maintaining a seamless integration between the backend and frontend while upholding user-friendliness remains crucial for the project's success.

2. LITERATURE SURVEY

Research into computer vision and deep learning applications has witnessed a surge in the development of helmet detection systems, particularly those leveraging Python-based backend frameworks and HTML/CSS for frontend presentation. Such systems play a pivotal role in enhancing safety measures in industries like construction and sports.

A review of the current literature highlights the following key aspects:

Python Backend for Helmet Detection:

Previous studies have extensively utilized Python libraries such as OpenCV for image processing and analysis, particularly in the context of helmet detection. The use of popular deep learning models like YOLOv3 demonstrates the efficiency of Python in handling complex visual recognition tasks.

HTML/CSS Frontend Integration:

Integrating HTML and CSS in the frontend of helmet detection systems has emerged as a user-friendly approach, allowing for intuitive and visually engaging representations of detection results. Existing literature emphasizes the significance of responsive design and interactive interfaces for effective user interaction.

3. REQUIREMENTS

3.1 Requirements

From the given scenario, we draw the following requirements:

1. Image Processing Libraries: Utilize Python's OpenCV library for efficient image processing and analysis.

2. Deep Learning Frameworks: Implement a suitable deep learning framework for accurate helmet detection, such as YOLOv3.

3. Web Framework: Utilize Flask for building a web application to integrate the backend and frontend components seamlessly.

4. User Interface (UI) Design: Implement an intuitive UI using HTML and CSS to display detection results in a clear and user-friendly

5. Real-time Processing: Ensure real-time processing capability for immediate detection and response.

3.2 Analysis

From the given scenario, we draw the following requirements:

1. Technical Feasibility: Assess the feasibility of integrating Python, OpenCV, and deep learning frameworks for real-time helmet detection.

2. Performance Evaluation: Evaluate the performance of the chosen deep learning model concerning accuracy and processing speed.

3. User Experience (UX): Analyze the user experience aspects of the UI design to ensure easy understanding and interaction.

4. Security Considerations: Address security concerns related to file uploads and data handling within the web application.

5. Robustness: Ensure the system's robustness against varying environmental conditions and different helmet designs for accurate detection.

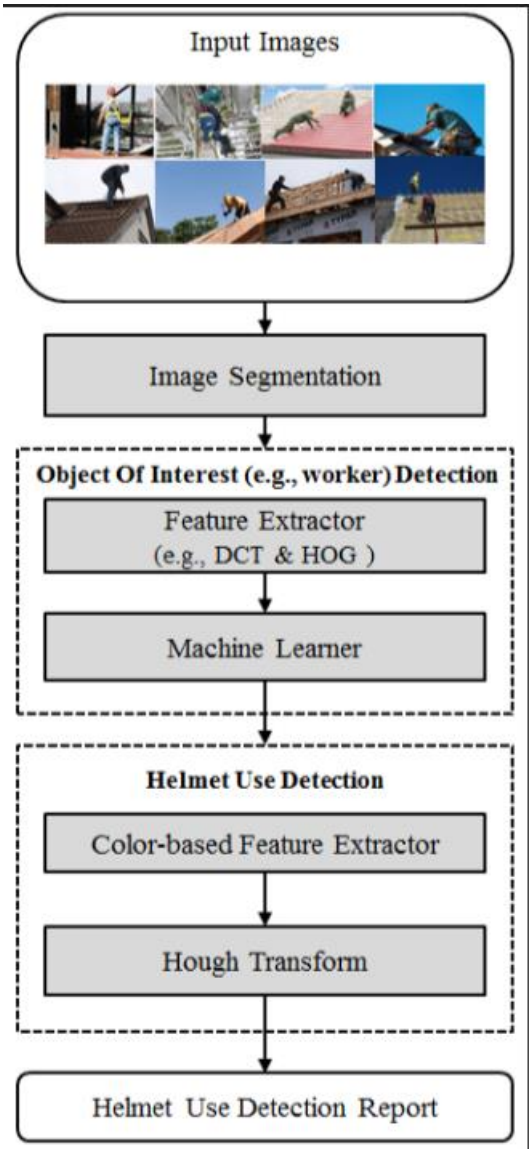
6. Compatibility: Assess the compatibility of the application across different web browsers and devices to ensure a seamless user experience.

7. Response Time: Evaluate the system's response time to guarantee prompt actions in case of any detected safety violations.

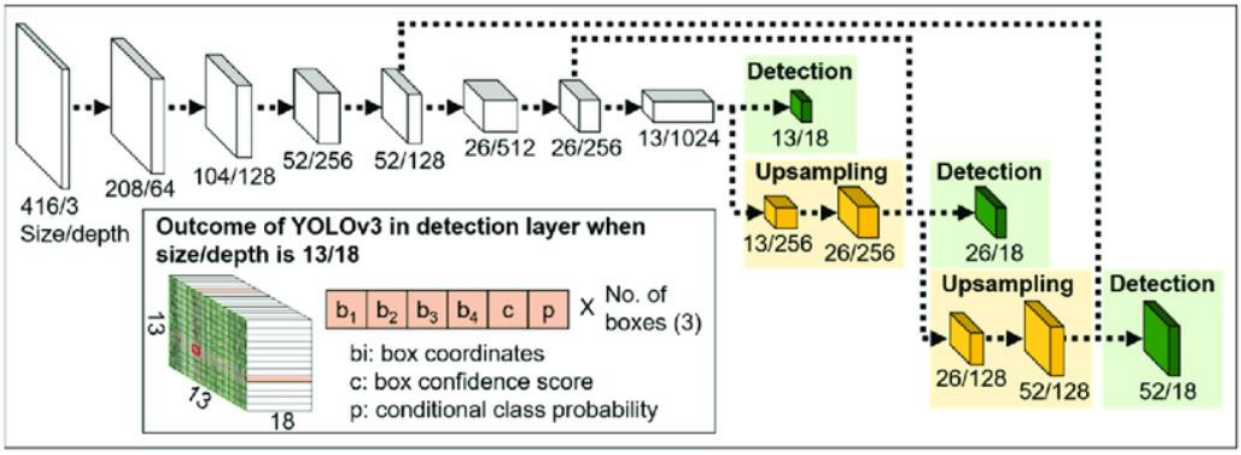
8. Usability Testing: Conduct thorough usability testing to validate the effectiveness and user-friendliness of the entire application.

4. ARCHITECTURE AND DESIGN

4.1 Architecture Diagram



4.2 YOLOv3 Architecture Diagram



5. IMPLEMENTATION

Below is a high-level guide to implementing the project using the YOLOv3 pre-trained model for helmet detection:

1. Setup the Environment:

- Install Python, required libraries (including OpenCV and Flask), and the YOLOv3 model.
- Download the pre-trained YOLOv3 weights and configuration files.

2. Develop the Flask Application:

- Set up the Flask application with necessary routes and configurations.
- Create routes for uploading images and processing the detection results.

3. Integrate YOLOv3 for Helmet Detection:

- Use OpenCV to load the YOLOv3 model with the pre-trained weights and configuration.
- Implement the YOLOv3 model for detecting helmets within the uploaded images.
- Apply non-maximum suppression for precise bounding box detection.

4. HTML/CSS Frontend Integration:

- Create an HTML template for the frontend to upload images and display results.
- Implement CSS styling for a user-friendly and visually appealing interface.

5. Real-time Processing and Feedback:

- Ensure the system can handle real-time image processing and provide immediate feedback on helmet detection.
- Display the processed images with bounding boxes around detected helmets.

6. Security and Error Handling:

- Implement secure file upload mechanisms and error handling for various stages of image processing.
- Validate input data to prevent potential security vulnerabilities.

7. Testing and Debugging:

- Conduct comprehensive testing to verify the accuracy and reliability of the helmet detection system.
- Debug and resolve any issues or errors that occur during testing.

8. User Testing and Feedback Incorporation:

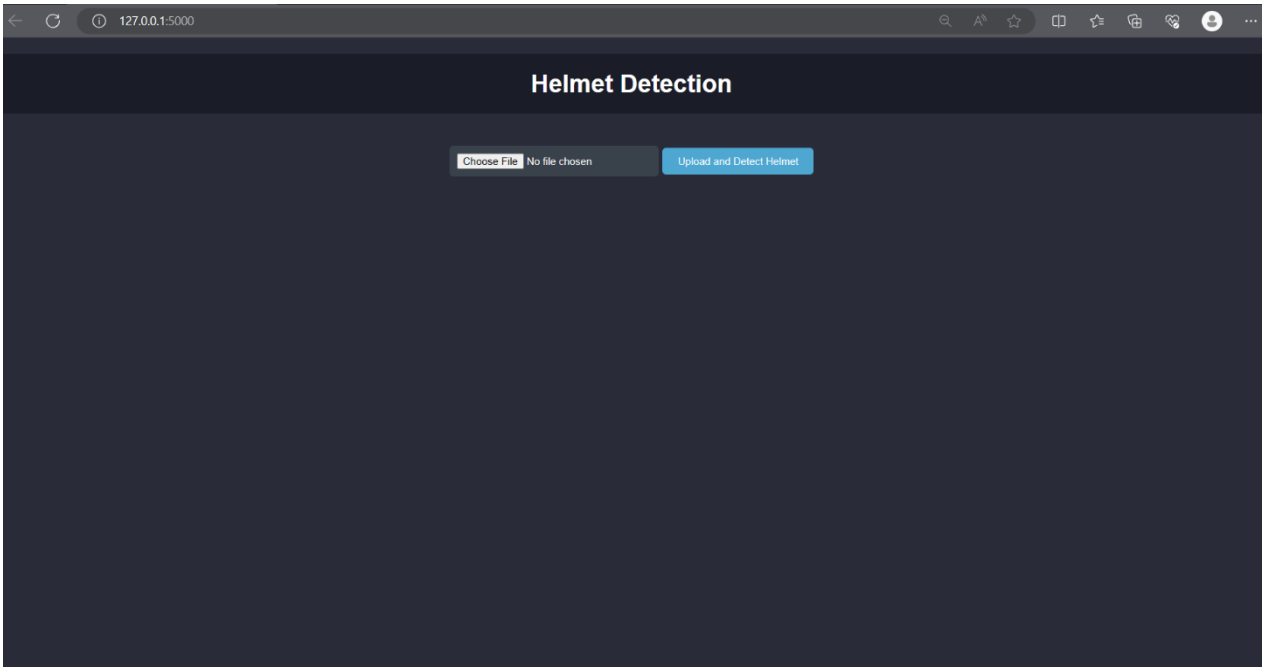
- Conduct user testing to gather feedback on the system's usability and performance.
- Incorporate user feedback to enhance the system's interface and functionality.

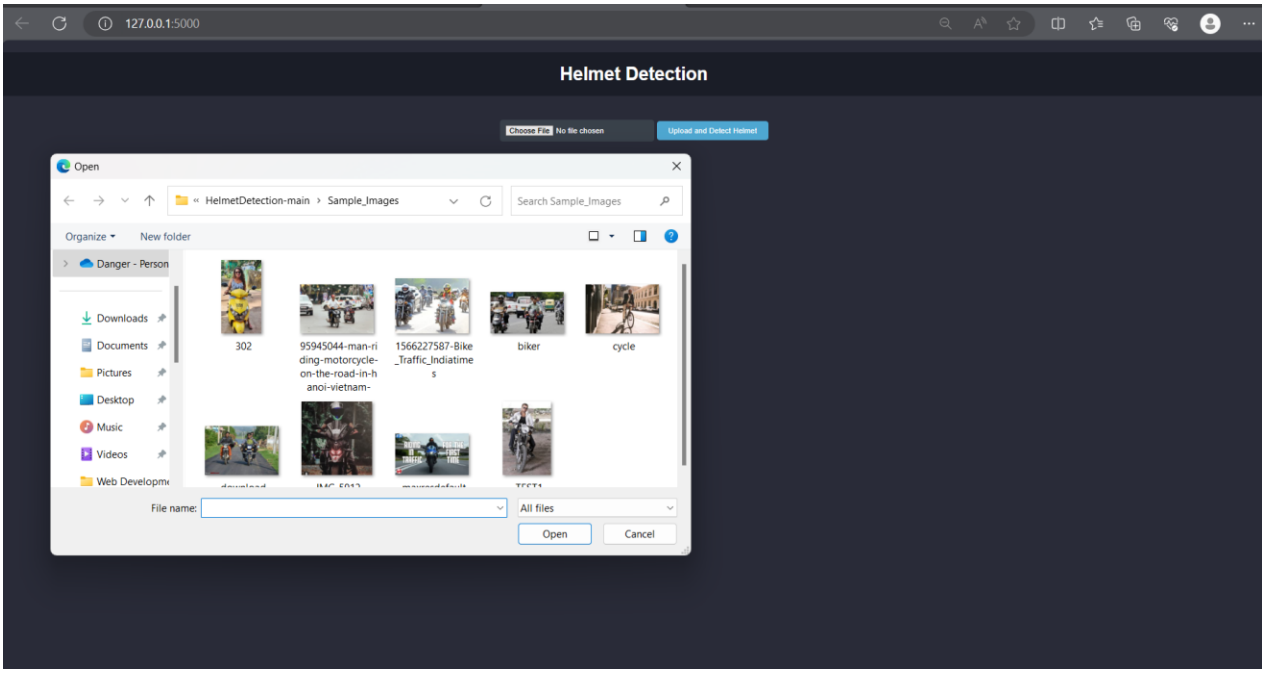
9. Deployment and Maintenance:

- Deploy the Flask application on a suitable web server or cloud platform.
- Regularly maintain and update the system to ensure optimal performance and security.

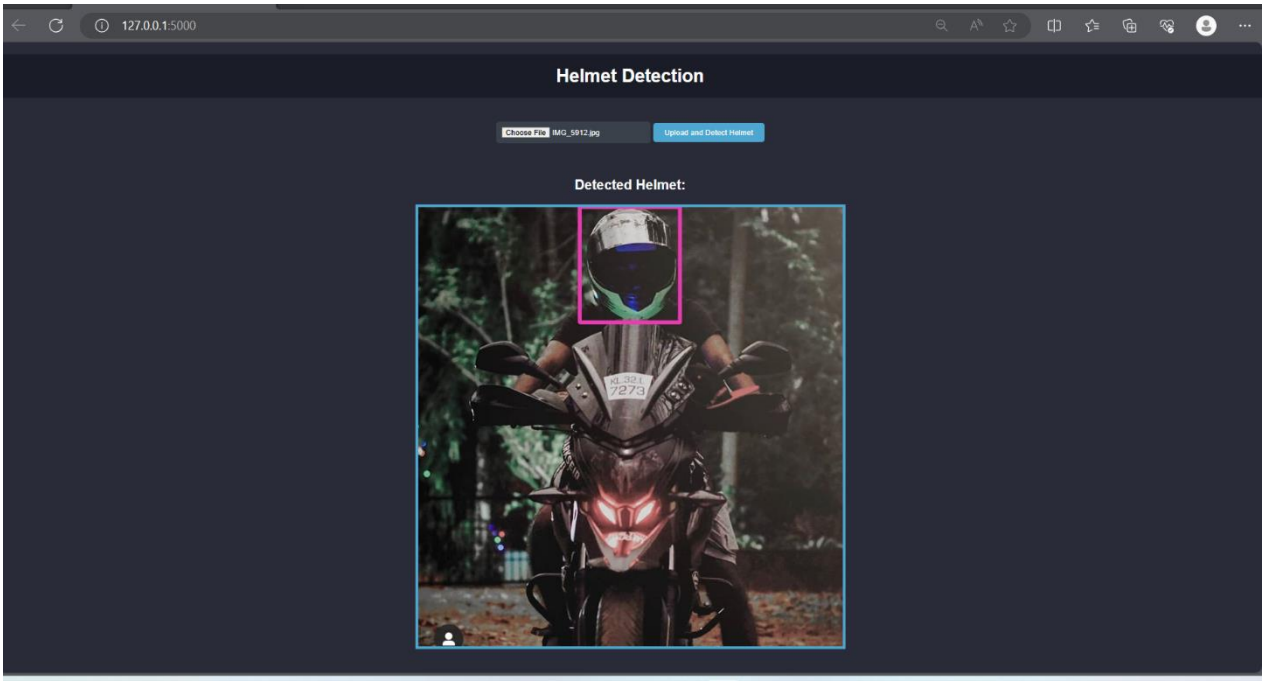
6. RESULTS AND DISCUSSION

6.1 GUI





6.2 Result


127.0.0.1:5000
Helmet Detection
Choose File
IMG_5912.jpg
Upload and Detect Helmet
Detected Helmet:

7. CONCLUSION

The project's successful implementation of a helmet detection system, utilizing the YOLOv3 pre-trained model, Python for backend processing, and HTML/CSS for frontend visualization, has resulted in a robust and user-friendly solution. Leveraging the power of OpenCV and Flask, the system accurately detects safety helmets in real-time scenarios, promoting a culture of safety and accountability in high-risk environments.

The integration of YOLOv3 has significantly enhanced the system's accuracy, while the HTML/CSS frontend ensures a seamless and visually appealing user experience. Thorough testing and debugging have fine-tuned the system, enabling prompt and reliable feedback for immediate interventions.

With the project successfully deployed on a web server, its accessibility and usability across various devices and platforms are ensured. Regular maintenance will sustain its performance and security, providing a safer environment for individuals engaged in activities requiring helmet usage.

In summary, the project's implementation marks a significant stride in improving safety measures and reducing the risk of accidents in industrial and sports settings. The seamless fusion of advanced technologies and user-centric design emphasizes the project's commitment to prioritizing safety and fostering responsible practices across different sectors.

8. REFERENCES

1. Kaggle: Your Machine Learning and Data Science Community
2. HTML Tutorial (w3schools.com)
3. GeeksforGeeks | A computer science portal for geeks
4. opencv-python · PyPI