International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

Safety Helmet Detection And License Plate Detection Using Advanced Yolov10

[1]Mohd Abdul Razzak, [2]Abdul Adnan, [3]Syed Awez Ali , [4]SMs.Imreena Ali
[1,2,3]B.E Students, Department of CSE, ISLEC, India.
[4]Assistant Professor, Department of CSE, ISLEC, India.
abdulrazzaknomaan@gmail.com, mohdadnan265@gamil.com, syedalibhai1234@gmail.com,

***ABSTRACT**:*
This project presents an advanced computer vision system for realtime Safety Helmet Detection and License Plate Recognition using the latest YOLOv10 object detection architecture. The primary objective is to enhance workplace safety and vehicle monitoring by automatically identifying individuals without safety helmets in industrial zones and capturing vehicle license plates for surveillance and regulation purposes. YOLOv10, known for its superior accuracy and speed, enables efficient multi-object detection in complex environments. The system is trained on annotated datasets containing diverse helmet types and vehicle plates under varying conditions, ensuring robust performance. Wearing safety helmets can effectively reduce the risk of head injuries for construction workers in high-altitude falls. In order to address the low detection accuracy of existing safety helmet detection algorithms for small targets and complex environments in various scenes, this study proposes an improved safety helmet detection algorithm based on YOLOv10.

INTRODUCTION

In the modern industrial and transportation sectors, ensuring the safety of workers and maintaining effective vehicle monitoring systems are of paramount importance. This project introduces a cutting-edge computer vision system that leverages
the power of the YOLOv10 object detection architecture for real-time Safety Helmet Detection and License Plate Recognition. The integration of these two critical applications addresses the growing need for intelligent surveillance systems that can operate accurately and efficiently in complex and dynamic environments. The primary goal of this system is to automatically identify individuals who are not wearing safety helmets in industrial and construction zones, thereby promoting workplace safety, and simultaneously recognize vehicle license plates for enhanced monitoring and regulatory compliance. The use of YOLOv10, renowned for its state-of-the-art performance in terms of detection accuracy and inference speed, enables the proposed system to perform multi-object detection with high precision, even in challenging conditions involving small targets and cluttered backgrounds. To achieve this, the model is trained on richly annotated datasets that include a wide range of helmet types and vehicle license plates captured under diverse lighting and weather conditions. Furthermore, the study introduces improvements to the standard YOLOv10 architecture to specifically address the challenges associated with detecting safety helmets in complex scenarios, where traditional algorithms often struggle with low accuracy. By incorporating advanced techniques and optimization strategies, the enhanced model demonstrates superior robustness and reliability in detecting both safety helmets and vehicle plates. This innovative approach not only contributes to reducing the risk of head injuries caused by falls in high-altitude construction work but also supports law enforcement and industrial management in maintaining safety standards and operational efficiency. Q

1.2 SCOPE OF THE PROJECT

The scope of this project encompasses the design, development, and implementation of a real-time computer vision-based system for dual-purpose detection—safety helmet detection and license plate recognition—using the YOLOv10 deep learning architecture. The system is intended for deployment in industrial environments, construction zones, and secured entry/exit points of facilities where monitoring both personnel safety compliance and vehicle movement is essential. It aims to accurately detect whether individuals are wearing safety helmets and identify vehicle license plates under varying environmental conditions, including different lighting, occlusions, and background complexities. The project includes the collection and annotation of diverse datasets, model training and optimization, system validation, and real-time deployment testing. Additionally, enhancements to the YOLOv10 model are within scope, specifically targeting improved detection accuracy for small and partially occluded objects like safety helmets in crowded scenes. While the primary application is focused on occupational safety and vehicle surveillance, the scalable and modular design allows for future integration with broader smart surveillance systems, IoT frameworks, and automated access control mechanisms.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

1.3 OBJECTIVE

The primary objective of this project is to develop an intelligent and efficient real-time computer vision system capable of detecting safety helmets on individuals and recognizing vehicle license plates using the advanced YOLOv10 object detection algorithm. This dual-function system aims to enhance safety compliance in industrial and construction environments by automatically identifying workers who are not wearing protective helmets, thereby helping to prevent potential head injuries and promote a culture of safety. Simultaneously, the system is designed to capture and recognize vehicle license plates with high accuracy, facilitating effective monitoring and regulation of vehicular access within secured areas. A key focus is placed on improving detection performance in challenging scenarios involving small targets, occlusions, variable lighting conditions. By training the YOLOv10 model on comprehensive and diverse datasets, the project seeks to ensure robust and reliable performance in real-world applications. Ultimately, the objective is to contribute to safer workplaces and more secure environments through the integration of cutting-edge AI and deep learning technologies.

EXISTING SYSTEM:

YOLOv8 algorithm is to accurately detect safety helmets in realtime, particularly in challenging environments like construction sites. YOLOv8 serves as the backbone for the detection process by identifying helmets, even when they are small or located at a distance. The algorithm is particularly effective due to its highspeed object detection capabilities, enabling the system to process video frames in real-time. Key features such as the Coordinate Attention (CA) mechanism and small target detection layer enhance
YOLOv8's performance, allowing it to focus on relevant helmet features while suppressing background noise. This makes YOLOv8 ideal for detecting safety helmets in diverse and complex settings, ensuring worker safety by providing immediate alerts when helmets are not detected.

1.4.1 EXISTINGSYSTEM DISADVANTAGES:

- YOLOv8 includes a specialized layer for small target detection.
- Improved Detection in Complex Environments. YOLOv8 can handle large datasets and work efficiently in different scenarios.
- YOLOv8 is optimized for performance, reducingcomputational load without sacrificing accuracy.

Title: Safety Helmet Detection in Electrical Power Scenes based on Improved Lightweight YOLOv5.
Author: Zuhe Li, Zhenwei Huang, Hongyang Chen, Lujuan Deng and Fengqin Wang.
Year: 20.
Description: To tackle the challenges of diverse targets, complex scenes, and partial occlusion in safety management during electrical field operations, the YOLO series algorithm, recognized for its exceptional accuracy and swift processing capabilities, has been applied to various scene detection tasks. To ascertain if workers have donned safety helmets and ensure the safety of electrical field operations, we propose a lightweight algorithm based on the improved YOLOv8 for constructing a digital safety helmet detection system. By incorporating the VoV-GSCSP module, we reduced model complexity, decreased computational load, and improved detection accuracy. Simultaneously, by combining the GSConv module, we enhanced the network's feature extraction capability, enabling the network to adapt more rapidly and accurately to various complex electrical scenes, thereby strengthening the network's robustness in safety helmet detection. Finally, we validated the effectiveness of the proposed model using the pre-existing dataset for safety helmet detection.
Title: Real-time detection of coal mine safety helmet based on improved YOLOv8
Author: S Jie Li, Shuhua Xie, Xinyi Zhou, Lei Zhang
,
Year: 20.
Description: The existing coal mine safety helmet detection method has problems such as low detection accuracy, susceptibility to environmental impact, poor real-time performance, and a large number of parameters. So, this paper proposes a Miner Helmet detection algorithm based on YOLO, abbreviated as MH-YOLO. First, the convolutional block attention mechanism (CBAM) is applied to improve the CSPDarkNet53 to 2-Stage FPN (C2f) module of the backbone network and enhance feature-extraction capability. Second, the MaxPooling (MP) module is used to replace the partial subsampling convolution of YOLOv8 to reduce the impact of unbalanced sample categories and improve the recall rate. In addition, a

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

small target detection layer is added to further improve the small target characteristics by fusing shallow network features with deep network features. Finally, the ZoomCat and Scalseq Module (ZAS) feature-extraction module is used to improve the detection accuracy of small and overlapping targets. Training and testing were conducted on the public dataset CUMTHelmet from China University of Mining and Technology and DsLMF + helmet from Xi'an University of Science and Technology. The proposed MHYOLO achieves mAP50 values of 92.4% and 97.8%, respectively, surpassing the comparative networks. The detection time is 10.1 ms, enabling accurate and realtime detection of whether coal miners are wearing safety helmets.

Title: Helmet Detection Based On Improved YOLO V8

Author: Sahir Suma, ANOOP G L, MITHUN B N,

Year: 2022

Description: This paper presents an automated Helmet Detection system for two-wheeler riders in India, using the advanced YOLO v8 algorithm for improved road safety. The system employs the Ultralytic YOLO algorithm and is trained on a carefully curated dataset generated via Robo Flow. It incorporates Convolutional Neural Network (CNN) and Neural Network (NN) architectures, demonstrating superior accuracy and efficiency compared to previous models. Ongoing refinements aim to enhance accuracy further and bounding box precision, highlighting the system's potential to significantly improve road safety in India.

Title: YOLO-PL: Helmet wearing detection algorithm based on improved YOLOv4.

Author: Haibin Li [a b], Dengchao Wu [a b], Wenming Zhang

Year: 2019.

Description: Workplace safety accidents are a pervasive issue worldwide. According to the National Work Safety Supervision Administration, a striking 67.95 % of construction accidents occur due to workers not wearing helmets. Existing helmet-wearing detection algorithms, however, tend to underperform in real-world scenarios where challenges such as smaller helmet areas in images, complex backgrounds, and object occlusions are present. Additionally, these models have a considerable amount of parameters, which impedes their practical deployment. This study proposes a novel, lightweight helmet detection algorithm, YOLOPL, based on YOLOv4, to address these challenges. Initially, we designed the YOLO-P algorithms. YOLO-P algorithms optimize the network structure by refining its ability to detect small objects and improving the anchor assignment in the detection head.

1.6 PROPOSED SYSTEM

YOLOv10 (You Only Look Once version 10) is the latest evolution in the YOLO family of realtime object detection models, offering significant improvements in both performance and efficiency. Building upon the foundational principles of previous versions, YOLOv10 integrates cutting-edge enhancements to achieve higher accuracy and faster inference times while maintaining a lightweight architecture suitable for deployment on edge devices. One of the key innovations in YOLOv10 is its **improved backbone and head architecture**, designed to extract rich and robust features from input images. This results in better object localization and classification, even in challenging environments with occlusions or low lighting. YOLOv10 also incorporates advanced attention mechanisms and optimized anchorfree detection techniques, reducing computation overhead and enhancing detection precision across a wide range of object sizes.

1.6.1 PROPOSED SYSTEM ADVANTAGES:

- No Need for Non-Maximum Suppression (NMS)
- Faster Inference (Real-Time Performance)
- Fewer Parameters and Computations Higher Accuracy
- Better Multi-Scale Object Detection.

CHAPTER 2: PROJECT DESCRIPTION

2.1 GENERAL: This project focuses on the development of an advanced real-time computer vision system that combines safety helmet detection and license plate recognition using the state-ofthe-art YOLOv10 object detection algorithm. The system is designed to enhance safety measures in industrial and construction environments by ensuring that all personnel are equipped with protective helmets, while also providing an automated solution for monitoring and identifying vehicle license plates for surveillance and access control purposes. YOLOv10 is chosen for its remarkable speed and accuracy, making it ideal for real-time applications involving multiple object detection in complex scenes. The system utilizes a well-annotated and diverse dataset, including various helmet types and license plates captured under different environmental conditions. It is further optimized to address the challenges of detecting small or partially obscured objects in cluttered backgrounds. The solution is intended to be scalable, efficient, and easily integrable with existing surveillance infrastructures, providing a significant contribution to workplace safety and vehicle monitoring automation.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

2.2 METHODOLOGIES

2.2.1MODULES NAME: Modules Name:

* Input Image/Video Object Detection Using YOLOv10 Data Augmentation *Coordinate Attention (CA) Mechanism
 Small Target Detection Output Detection

2.2.2 MODULES EXPLANATION:

1) **Input Image/Video:** The system processes real-time input from cameras, typically installed on construction sites or highrisk environments, capturing images or video frames.
2) **Object Detection Using YOLOv10:** YOLOv10, a state-ofthe-art object detection model, scans the input images to detect various objects. In this case, it is specifically trained to detect safety helmets. YOLOv10 performs this in a single pass, making it highly efficient for real-time detection.
3) **Data Augmentation:** To improve the model's ability to detect small and distant helmets, the system uses mosaic data augmentation. This technique helps generate tiny targets, ensuring better model generalization and enhancing its performance in crowded and complex scenes. **4) Coordinate Attention (CA) Mechanism:** YOLOv10 is enhanced with a Coordinate Attention (CA) mechanism in the backbone network. This mechanism allows the model to focus on safety helmet regions in complex backgrounds, effectively suppressing irrelevant features and improving detection accuracy.

2.3 TECHNIQUE USED OR ALGORITHM USED

2.3.1 EXISTING TECHNIQUE: -

 Yolov8

YOLOv8 algorithm works by first receiving real-time video or image frames from cameras installed in the environment. The model then processes these frames to detect and classify objects, specifically safety helmets. YOLOv8’s backbone network extracts feature from the input images, identifying patterns and structures that are relevant to the appearance of helmets. The Coordinate Attention (CA) mechanism is used to enhance the model's focus on helmet regions, helping to suppress irrelevant background features and improving detection accuracy in complex environments. Additionally, YOLOv8 employs a small target detection layer to better handle smaller helmets, ensuring accurate detection even for helmets that are distant or partially obscured. The algorithm then places bounding boxes around the detected helmets and assigns class labels (helmet or no helmet). The output is provided in realtime, with detected helmets highlighted by bounding boxes, and the system can trigger alerts if no helmet is detected.

2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:

YOLOv10 (You Only Look Once version 10) is a powerful and fast object detection model used to identify objects in images and videos in real time. It is the newest and most advanced version in the YOLO series. One of its biggest improvements is that it no longer needs a process called Non-Maximum Suppression (NMS), which earlier versions used to remove extra overlapping boxes. This makes YOLOv10 faster and more accurate. It also uses a better way to understand and combine information from different parts of an image, helping it detect both large and small objects more effectively. YOLOv10 is designed to work well even with fewer computing resources, making it ideal for real-time applications like detecting safety helmets on workers or reading vehicle license plates. With different model sizes available, it can be used in a variety of situations—from small devices to large systems—while still maintaining high performance. YOLOv10 models are up to **37% to 70% faster** in processing images than YOLOv8 models. This makes YOLOv10 ideal for real-time applications like surveillance and live video analysis. With improved feature fusion and adaptive techniques, YOLOv10 handles **objects of all sizes** better than YOLOv8, especially small or overlapping objects.

REQUIREMENTS ENGINEERING

3.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attributebased deduplication systems by resorting to the hybrid cloud architecture.

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

*Usability : The system is designed with completely automated process hence there is no or less user intervention.

Reliability: The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance: This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability: The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

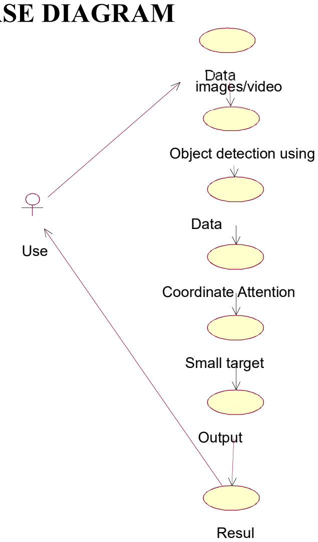
Implementation :The system is implemented in web environment using Jupyter notebook software. The server is used as the intelligence server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

DESIGN ENGINEERING

4.1 GENERAL : Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering**.**

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM

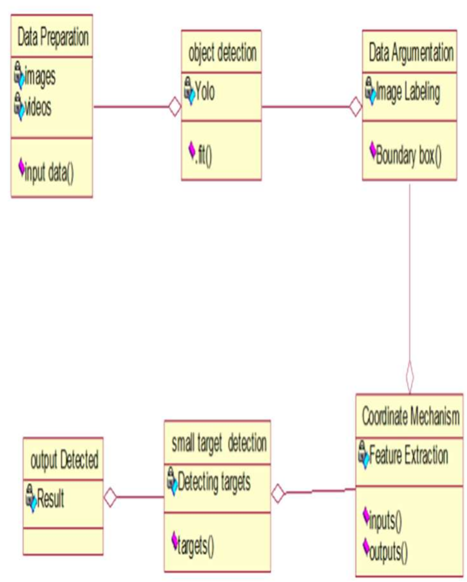


EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor.

Each will play a certain role to achieve the concept.

4.2.2 CLASS DIAGRAM



EXPLANATION : In this class diagram represents how the classes with attributes and methods are linked

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

together to perform the verification with security. From the above diagram shown the various classes involved in our project.

SYSTEM ARCHITECTURE:

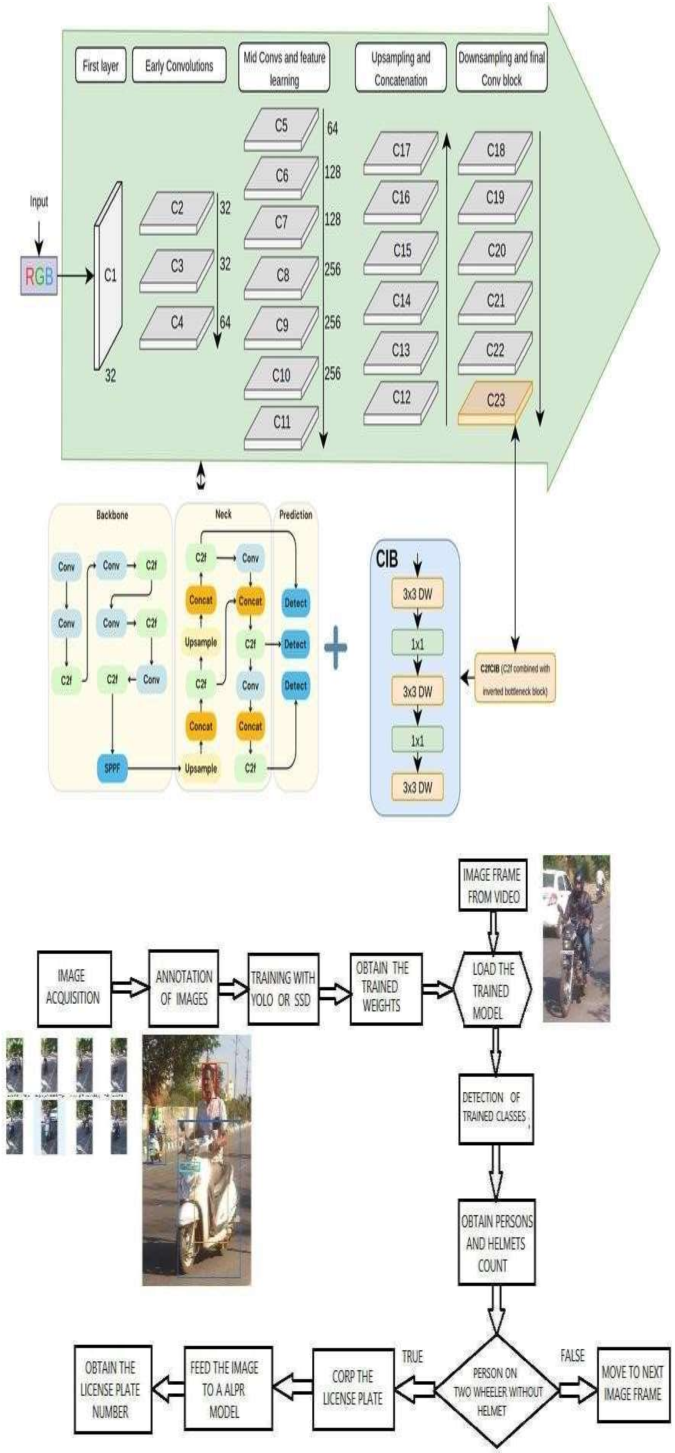


Fig 4.11: System Architecture

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

ISSN 2347–3657

Volume 13, Issue 2s, 2025

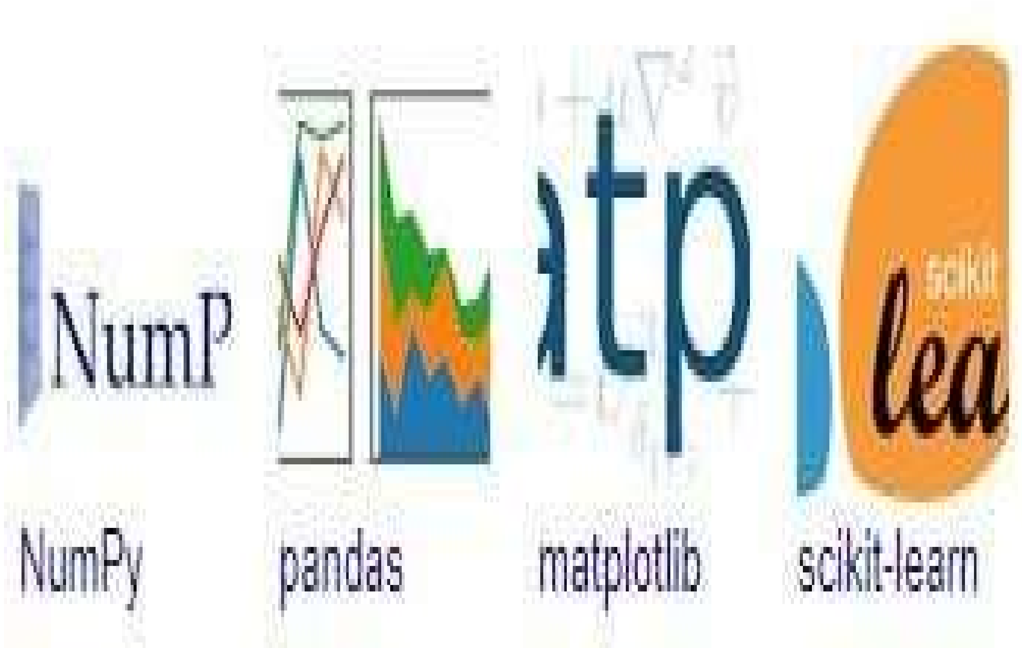


Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 5 : IMPLEMETATION

5.1 GENERAL

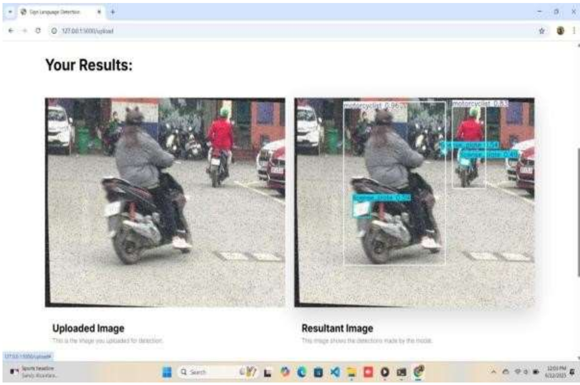
```
from flask import Flask, render_template, request, redirect, url_for, session, send_from_directory, Response
import os import sqlite3 import cv2 import subprocess from werkzeug.utils import secure_filename from ultralytics import
YOLO   app = Flask(__name__) app.secret_key =
'supersecretkey'
# Database Connection Function def get_db_connection():
conn = sqlite3.connect('Python.db')     conn.row_factory = sqlite3.Row  # Dictionary-like rows     return conn

# Create Users Table def create_users_table():     conn = get_db_connection()     cursor = conn.cursor()     cursor.execute('''
CREATE TABLE IF NOT EXISTS Users(            id INTEGER
PRIMARY KEY AUTOINCREMENT,            username TEXT NOT NULL,            email TEXT NOT NULL UNIQUE,
password TEXT NOT NULL         )
''')
```

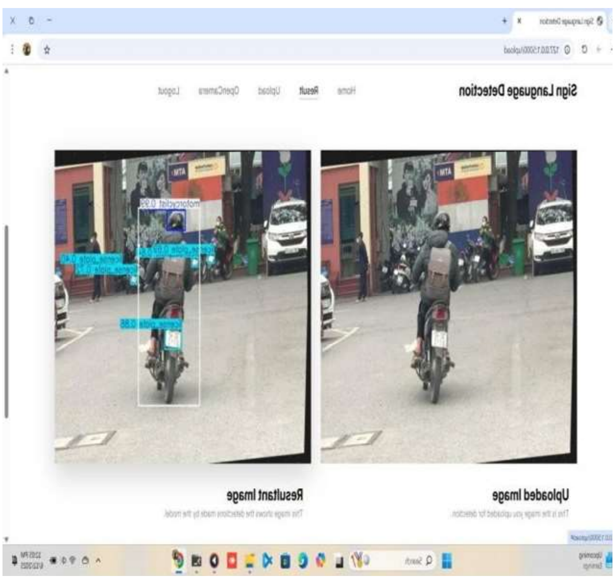

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

6-SNAPSHOTS

General:





CHAPTER 7 SOFTWARE TESTING

7.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

7.3 Types of Tests

7.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

7.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

ISSN 2347–3657

Volume 13, Issue 2s, 2025

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

7.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

7.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination
  Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

7.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

.FUTURE ENHANCEMENT

8.1 FUTURE ENHANCEMENTS:

In the future, this project can be enhanced by incorporating additional safety features and expanding its capabilities to create a more comprehensive intelligent surveillance system. One potential enhancement is the integration of facial recognition technology to identify and track individual workers, ensuring personalized safety compliance and attendance monitoring. The system can also be extended to detect other personal protective equipment (PPE) such as safety vests, gloves, and goggles, making it suitable for broader occupational safety applications. Another promising direction is the integration with Internet of Things (IoT) devices and cloud-based platforms for real-time alerts, remote monitoring, and centralized data storage and analysis. Moreover, employing edge computing devices can improve processing efficiency and reduce latency, allowing for seamless real-time performance in low-bandwidth environments. The system could also benefit from multilingual optical character recognition (OCR) for reading license plates from different regions and countries. Continuous model refinement using transfer learning and real-time feedback loops would further enhance the accuracy and adaptability of the system to evolving environments and requirements.

CHAPTER 9 : CONCLUSIONAND REFERENCES

9.1 CONCLUSION

In conclusion, this project successfully demonstrates the potential of leveraging advanced deep learning techniques, particularly the YOLOv10 object detection algorithm, to develop an efficient and accurate real-time system for Safety Helmet Detection and License Plate Recognition. By addressing critical challenges such as small object detection, environmental variability, and multiobject recognition, the proposed system significantly contributes to enhancing workplace safety and vehicle monitoring in industrial and construction settings. The integration of these functionalities into a single, unified framework ensures improved compliance with safety regulations and streamlined surveillance operations. With its high performance, scalability, and potential for further enhancements, this system represents a forward-thinking solution that aligns with the growing demand for intelligent automation in safetycritical environments. The outcomes of this project pave the way for future advancements in smart surveillance systems, contributing to safer and more secure operational spaces.

1.2 REFERENCES

[1] F. Zhou, H. Zhao, and Z. Nie, “Safety helmet detection based on YOLOv5,” in Proc. IEEE Int. Conf. Power Electron., Comput. Appl. (ICPECA), Jan. 2021, pp. 6–11.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

[2] L. Huang, Q. Fu, M. He, D. Jiang, and Z. Hao, "Detection algorithm of safety helmet wearing based on deep learning," Concurrency Comput., Pract. Exper., vol. 33, no. 13, p. e6234, Jul. 2021. [3] S. Sanjana, S. Sanjana, V. R. Shriya, G. Vaishnavi, and K. Ashwini, "A review on various methodologies used for vehicle classification, helmet detection and number plate recognition," Evol. Intell., vol. 14, no. 2, pp. 979–987, Jun. 2021.

[4] M. U. Farooq, M. A. Bhutto, and A. K. Kazi, "Real-time safety helmet detection using YOLOv5 at construction sites," Intell. Autom. Soft Comput., vol. 36, no. 1, pp. 911–927, 2023.

[5] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike-riders without helmet using surveillance videos in real-time," in Proc. Int. Joint Conf. Neural Netw., Jul. 2016, pp. 3046– 3051.

[6] A. H. M. Rubaiyat, T. T. Toma, M. Kalantari-Khandani, S. A. Rahman, L. Chen, Y. Ye, and C. S. Pan, "Automatic detection of helmet uses for construction safety," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Workshops (WIW), Oct. 2016, pp. 135–142.

[7] M.-W. Park, N. Elsafty, and Z. Zhu, "Hardhat-wearing detection for enhancing on-site safety of construction workers," J. Construct. Eng. Manage., vol. 141, no. 9, Sep. 2015, Art. no. 04015024.

[8] B. E. Mneymneh, M. Abbas, and H. Khoury, "Automated hardhat detection for construction safety applications," Proc. Eng., vol. 196, pp. 895–902, Jan. 2017.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic 10 segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014.

[10] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP343-352