International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Enhanced Helmet Detection in Complex Industrial Environments Using an Improved YOLO-BASED Model

Author 1: Aliveli Shivapriya
Email: alivelishivapriya07@gmail.com

Author 2: Alugonda Abilash Reddy
Email: abilashreddyalugonda@gmail.com

Author 3: Anchuri Sai Vardhan Reddy
Email: saivardhan612@gmail.com

Guide: Dr.Ch.Narasimha Chary
Organization: Guru Nanak Institutions, India

keywords: Safety helmet detection, YOLOv8n

ABSTRACT:

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 YOLOv8, named YOLOv8n. For data augmentation, the mosaic data augmentation method is employed, which generates many tiny targets. In the backbone network, a coordinate attention (CA) mechanism is added to enhance the focus on safety helmet regions in complex backgrounds, suppress irrelevant feature interference, and improve detection accuracy. In the neck network, a slim-neck structure fuses features of different sizes extracted by the backbone network, reducing model complexity while maintaining accuracy. In the detection layer, a small target detection layer is added to enhance the algorithm's learning ability for crowded small targets.

CHAPTER-1

INTRODUCTION

In work environments such as construction sites, tunnels, and coal mines, wearing a safety helmet is one of the fundamental requirements to ensure personnel safety. It effectively reduces the risk of head injuries when construction workers fall from heights, providing crucial protection [1], [2], [3]. Monitoring whether individuals are wearing safety helmets as per regulations relies on video data collected by cameras and assessed through manual supervision. However, traditional monitoring methods face increased labor costs, surveillance fatigue, and subjective judgments. Therefore, the development of high-performance safety helmet detection algorithms holds significant importance. Safety helmet detection methods have been enhanced with the continuous advancement of algorithms in computer vision and improvements in computational capabilities. As a highly regarded technology, deep learning has found widespread application in safety helmet recognition. Compared to traditional methods, deep learning algorithms, especially the

YOLO series, have achieved a remarkable balance between accuracy and speed [4]. However, Yolo-based safety helmet detection methods still encounter challenges in achieving high accuracy for small targets in complex backgrounds.

Complex environments may feature numerous interfering objects, such as buildings and trees, making it difficult for the algorithm to locate and identify safety helmets accurately Additionally, safety helmets' small and monochromatic features make them susceptible to interference from other objects in complex backgrounds, leading to misjudgments. Complex backgrounds may also involve occlusion phenomena, such as overlapping crowds and passing vehicles, causing the safety helmet's shape to be incomplete or partially obscured, making it challenging for the algorithm to identify. Since the introduction of the YOLO single-stage object detection algorithm, it has garnered widespread attention from industry scholars. In recent years, the YOLO algorithm has undergone continuous optimization. In 2023, the Ultra analytics team proposed the YOLOv8 version, which, while meeting real-time requirements, exhibits high detection accuracy and a lightweight network structure suitable for object detection. The progress in object detection has inspired the development of safety helmet detection methods using deep learning. Numerous researchers assert that deep learning technology represents a crucial avenue for tackling construction security management challenges.

1.2 SCOPE OF THE PROJECT

The scope of this project focuses on improving safety in high-risk environments, such as construction sites, by accurately detecting the wearing of safety helmets. The project aims to address challenges in detecting small targets, like safety helmets, in complex backgrounds or at long distances. By incorporating enhancements like the Coordinate Attention (CA) mechanism, slim-neck structure, and small target detection layer, the project aims to improve the algorithm's ability to detect helmets in real-world scenarios.

1.3 OBJECTIVE

The objective of this project is to develop an enhanced safety helmet detection system using an improved YOLOv8 model, named YOLOv8n, to overcome the challenges of detecting small targets in complex environments. The primary goals include improving the accuracy of detecting safety helmets, particularly in crowded or distant settings, by incorporating a small target detection layer. Additionally, the project aims to optimize the model's performance through the integration of the Coordinate Attention (CA) mechanism and a slim-neck structure, which helps focus on relevant features while reducing computational load and model complexity. The system is designed for real-time helmet detection with high accuracy and efficiency, making it suitable for deployment in high-risk environments like construction sites. Lastly, the proposed model's performance is validated and compared with other mainstream object detection algorithms, using metrics such as precision, recall, and map to demonstrate its superiority.

1.4 EXISTING SYSTEM:

In a similar project previously done using OpenCV, safety helmet detection was achieved through image processing and object detection techniques. OpenCV was used to preprocess input images by resizing, converting color spaces, and filtering to improve image quality. Object detection methods, such as Haar cascades, helped identify helmets in the images. Features like edges and contours were extracted to locate the helmets, and bounding boxes were drawn around detected helmets, with labels indicating whether a helmet was worn or not. This approach allowed for real-time video processing, providing immediate feedback. However, it may face challenges in complex environments or with small targets, which is where YOLOv8 can offer improved accuracy and efficiency.

1.4.1 EXISTINGSYSTEM DISADVANTAGES:

- Object detection methods like Haar cascades may struggle to accurately detect helmets in complex or cluttered environments.
- Less Performance with Small Targets Detection.

- OpenCV may require additional processing time for feature extraction.
- OpenCV-based methods may not generalize well across diverse datasets.

1.5 LITERATURE SURVEY

Title: An improved safety helmet detection algorithm based on YOLOv8

Authors: X. Wu, D. Hong, Z. Huang, and J. Chanussot

Year: 2023

DESCRIPTION: Safety helmets are very important for the life safety of operators in construction sites, mines and other high-risk operating environments. Therefore, based on the YOLOv8 framework, this study proposes the CC-YOLOv8 safety helmet detection algorithm, optimizing the issues of low accuracy and high computational resource consumption faced by traditional detection methods. By introducing the C2fcc module, the backbone network feature extraction capability of the algorithm is significantly enhanced. Meanwhile, the EMA attention mechanism is added to the algorithm, which effectively improves the object localization accuracy. The experimental results show that the algorithm demonstrates superior performance in a variety of scenarios and conditions, and its mAP0.5 reaches 92.6%, which is improved by 0.5% compared with the original algorithm. This research result provides an efficient and accurate new method for safety helmet detection**.**

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: 2023

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: 2023.

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 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 CUMT-Helmet from China University of Mining and Technology and DsLMF + helmet from Xi'an University of Science and Technology. The proposed MH-YOLO achieves mAP50 values of 92.4% and 97.8%, respectively, surpassing the comparative networks. The detection time is 10.1 ms, enabling accurate and real-time 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, YOLO-PL, 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. We design the Enhanced PAN (E-PAN) structure to merge the higher-layer, low-noise information with the lower-layer information based on the Path Aggregation Network (PAN). The YOLO-P algorithm improves detection accuracy by using the E-PAN structure.

1.6 PROPOSED SYSTEM

YOLOv8 algorithm is to accurately detect safety helmets in real-time, 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 high-speed 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.

1.6.1 PROPOSED SYSTEM ADVANTAGES:

- 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, reducing computational load without sacrificing accuracy.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

The working of OpenCV in the context of safety helmet detection involves several key steps. First, OpenCV captures real-time video frames or images from a camera or a stored dataset. The images are then pre-processed to enhance quality and remove noise, involving operations like resizing, grayscale conversion, histogram equalization, and filtering (such as Gaussian blur) to make object detection more accurate. Next, object detection algorithms like Haar cascades or HOG (Histogram of Oriented Gradients) are applied to identify potential regions where helmets might be present. OpenCV then analyses various features in the image, such as edges, corners, and contours, to detect helmet-like shapes, identifying distinctive patterns common to helmets. Once potential helmet regions are detected, OpenCV draws bounding boxes around them to highlight the helmet's location and labels the object as a "helmet." Finally, additional post-processing steps are applied, such as filtering false positives based on size, shape, or confidence thresholds. OpenCV's real-time processing capabilities allow the system to continuously analyses video frames, detect helmets, and provide immediate feedback, such as alerts if a helmet is not detected.

2.2 METHODOLOGIES

The proposed system for safety helmet detection using YOLOv8n is implemented through a modular approach. Each module is designed to handle a specific task, and together they form a complete pipeline for efficient and accurate detection. The following modules describe the methodology adopted in this project

2.2.1MODULES NAME:

- Input Image/Video
- Object Detection Using YOLOv8
- 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 high-risk environments, capturing images or video frames.

2. Object Detection Using YOLOv8:

YOLOv8, a state-of-the-art object detection model, scans the input images to detect various objects. In this case, it is specifically trained to detect safety helmets. YOLOv8 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:

YOLOv8 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

5. Small Target Detection:

A small target detection layer is added in the detection layer to improve the model's ability to detect helmets that may be small or located far away in the frame. This addition enables better performance in real-world environments where helmets may be partially obscured or seen from a distance.

6. Output Detection:

Once YOLOv8 identifies the helmet, the system outputs the result with high accuracy, showing whether the person is wearing a safety helmet. If the helmet is not detected, an alert is triggered, warning the relevant personnel.

2.3 TECHNIQUE USED OR ALGORITHM USED

2.3.1 EXISTING TECHNIQUE: -

- **OpenCV**

OpenCV in the context of safety helmet detection involves several key steps. First, OpenCV captures real-time video frames or images from a camera or a stored dataset. The images are then preprocessed to enhance quality and remove noise, involving operations like resizing, grayscale conversion, histogram equalization, and filtering (such as Gaussian blur) to make object detection more accurate. Next, object detection algorithms like Haar cascades or HOG (Histogram of Oriented Gradients) are applied to identify potential regions where helmets might be present. OpenCV then analyzes various features in the image, such as edges, corners, and contours, to detect helmet-like shapes, identifying distinctive patterns common to helmets. Once potential helmet regions are detected, OpenCV draws bounding boxes around them to highlight the helmet's location and labels the object as a "helmet." Finally, additional post-processing steps are applied, such as filtering false positives based on size, shape, or confidence thresholds. OpenCV's real-time processing capabilities allow the system to continuously analyze video frames, detect helmets, and provide immediate feedback, such as alerts if a helmet is not detected.

2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:

- **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 real-time, with detected helmets highlighted by bounding boxes, and the system can trigger alerts if no helmet is detected. YOLOv8's ability to operate in real-time allows for continuous and efficient monitoring, making it highly effective for safety helmet detection in various environments.

CHAPTER 3

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

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 attribute-based 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 intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

CHAPTER 4

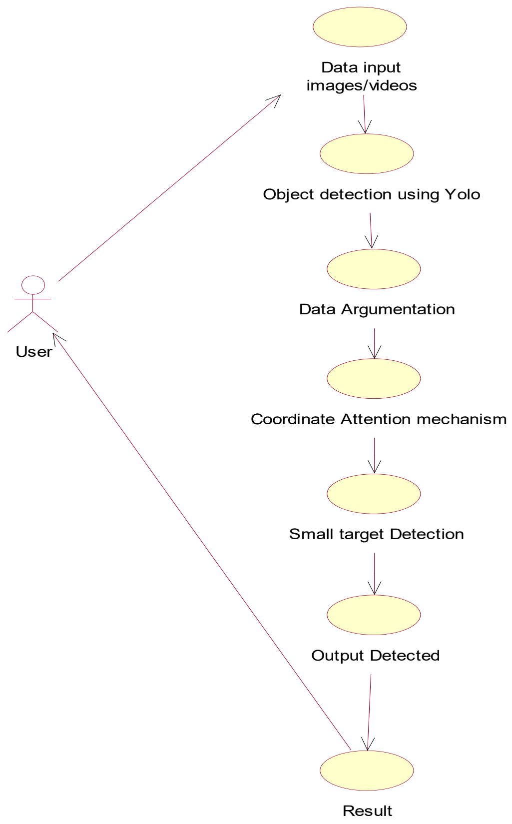
DESIGN ENGINEERING

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

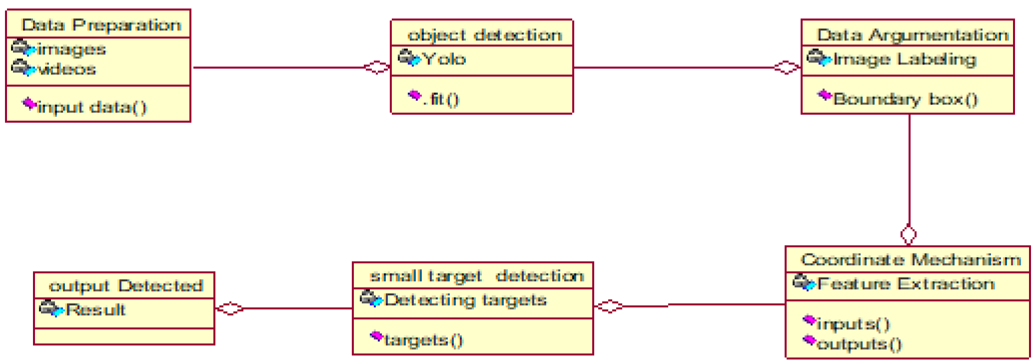
4.2.1 USE CASE DIAGRAM



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.

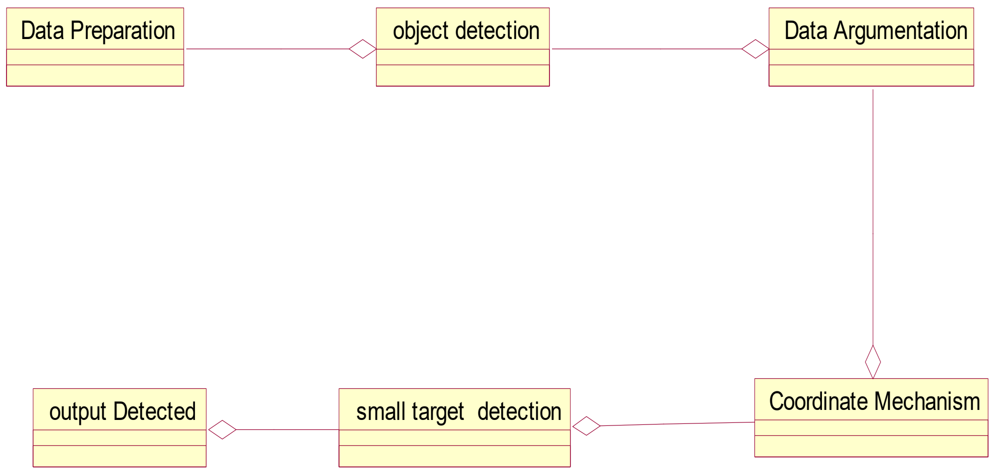
4.2.2 CLASS DIAGRAM



EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

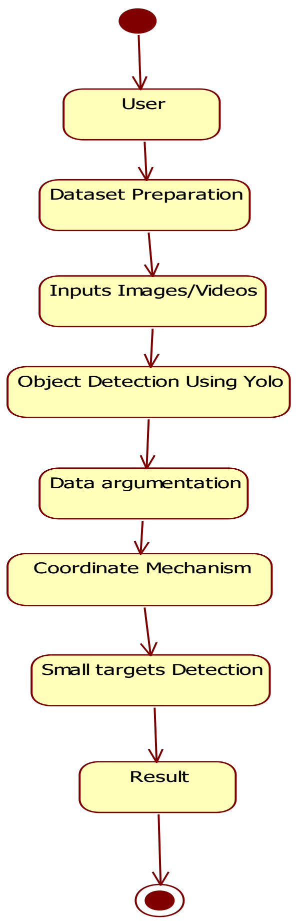
4.2.3 OBJECT DIAGRAM



EXPLANATION:

In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

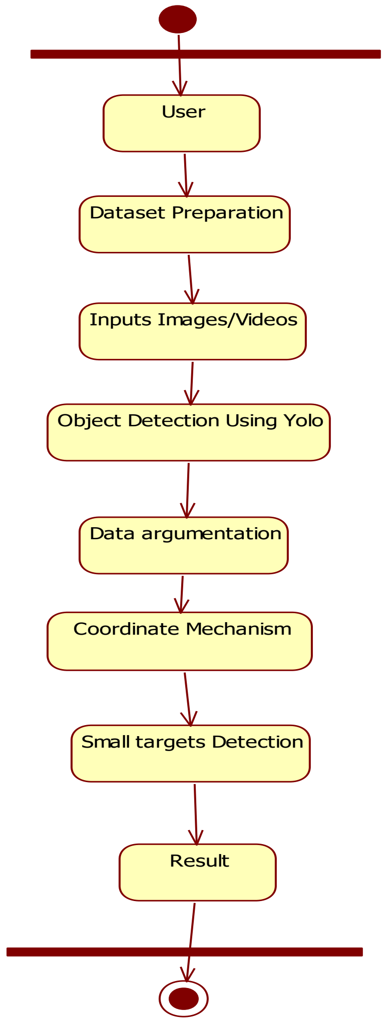
4.2.4 STATE DIAGRAM



EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

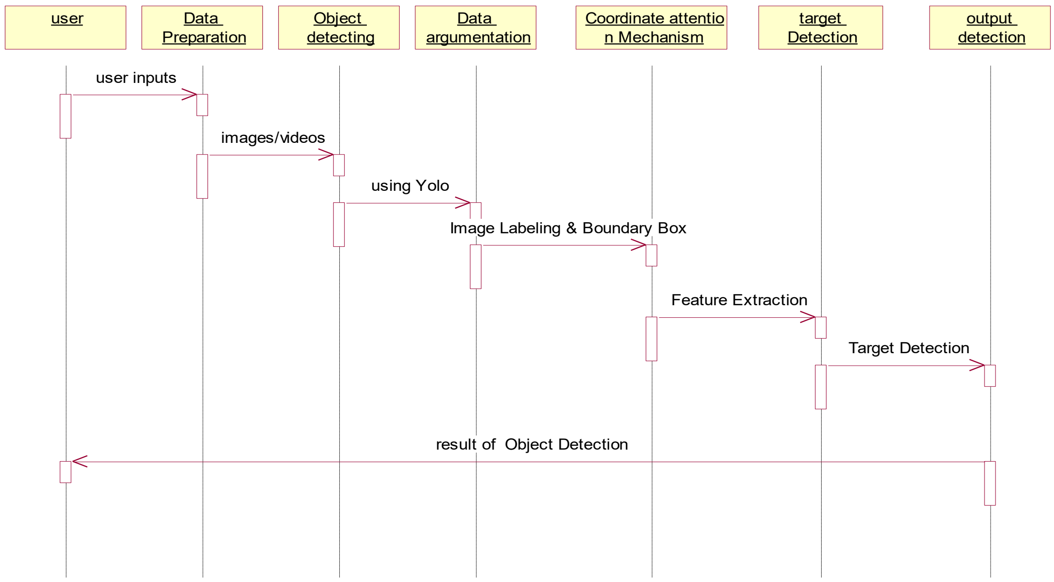
4.2.5 ACTIVITY DIAGRAM



EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

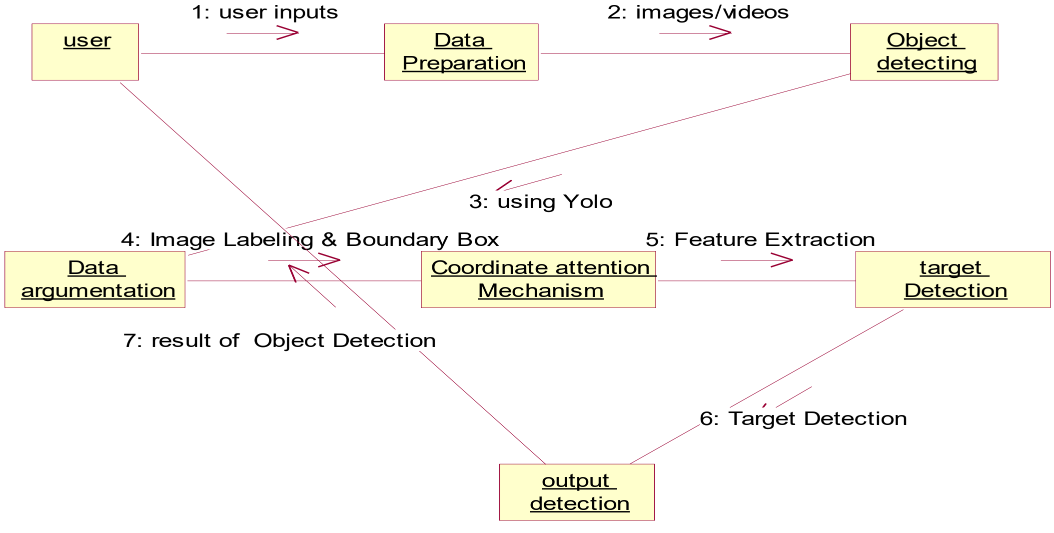
4.2.6 SEQUENCE DIAGRAM



EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

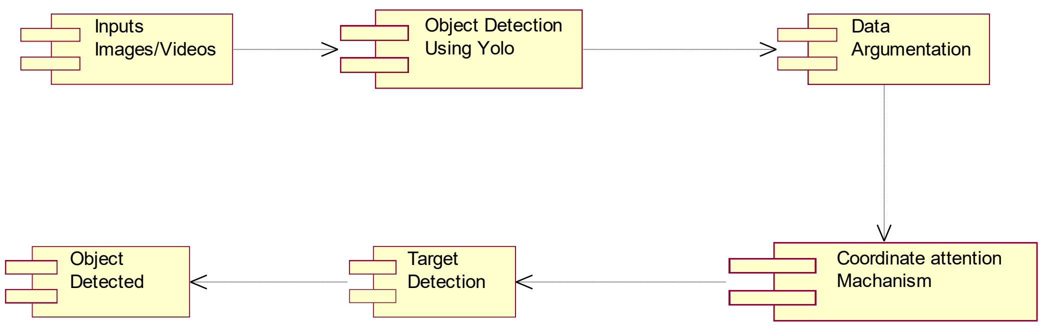
4.2.7 COLLABORATION DIAGRAM



EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

4.2.8 COMPONENT DIAGRAM

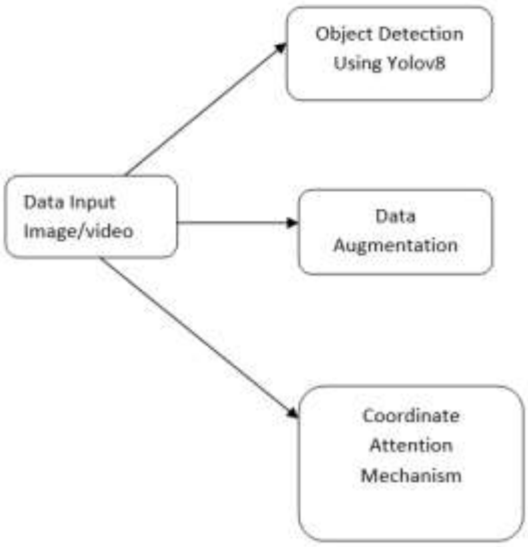


EXPLANATION

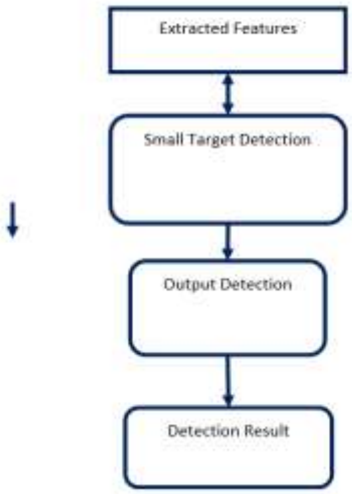
In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

4.2.9DATA FLOW DIAGRAM

Level 0



Level 1

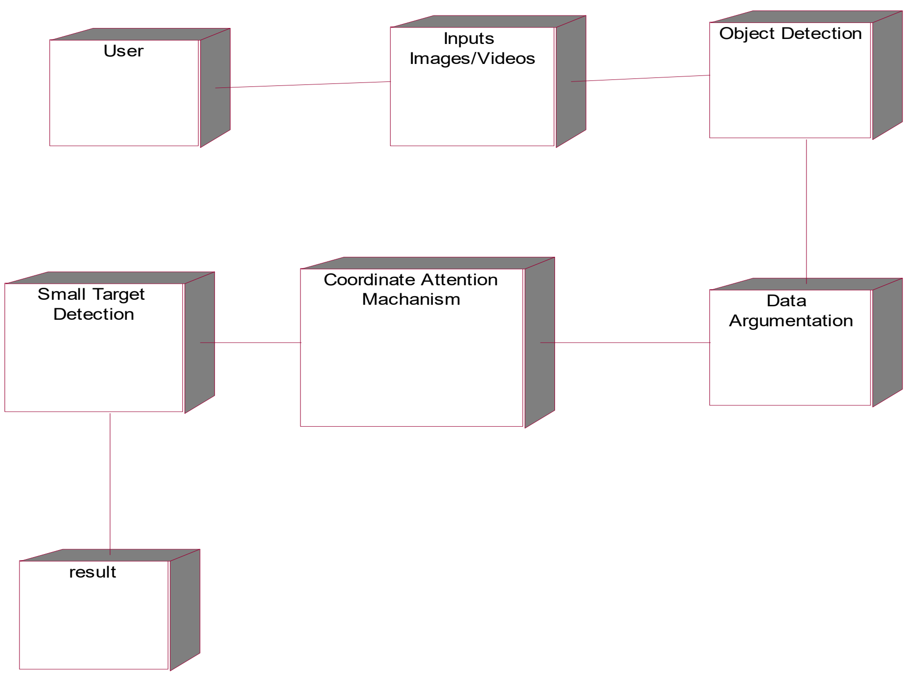


EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

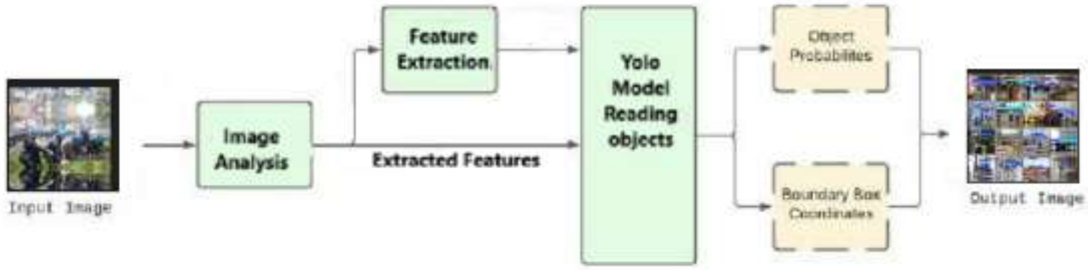
4.2.10 DEPLOYMENT DIAGRAM



EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

SYSTEM ARCHITECTURE:



CHAPTER 5

DEVELOPMENT TOOLS

5.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

5.3 Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.4 Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Spyder3

- Spyder3 is an open-source Python IDE (Integrated Development Environment) designed mainly for scientists, engineers, and data analysts who work with Python. It provides a complete environment to write, run, debug, and analyze Python programs.
- You can imagine it as a digital laboratory for Python experiments , where code, output, and data live together on one screen.

Key Features:

- **Code editor**: write python programs easily and Supports syntax highlighting and auto-completion.
- **IPython Consloe**: Allowes you to run Python commands interactively and see results instantly.
- **Variable Explorer:** Displays variables created in the program and helps in analyzing data and debugging code.
- **Debugger:** Helps find and fix errors step by step in a program
- **Integration with scientific Libraries:** works well with libraries like Numpy, Pandas, Matplotlib, and SciPy

CHAPTER 6

SOFTWARE TESTING

6.1 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 test. Each test type addresses a specific testing requirement.

6.2 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

6.3Types of Tests

6.3.1 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.

6.3.2 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**.**

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

6.3.4 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.

6.3.5 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.

6.3.6 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

6.2.7 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.

CHAPTER 7

FUTURE ENHANCEMENT

7.1 FUTURE ENHANCEMENTS:

The feature enhancements of YOLOv8 for safety helmet detection in this project include several key improvements. The Coordinate Attention (CA) mechanism helps the model focus on helmet regions, improving detection accuracy by reducing background noise. Additionally, the small target detection layer ensures that helmets, even when distant or partially obscured, are accurately detected. Mosaic data augmentation boosts the model’s ability to handle complex scenes by creating synthetic training data, enhancing its performance in varied conditions. YOLOv8 also features a slim-neck structure that reduces model complexity while maintaining detection accuracy. It is optimized for computational efficiency, allowing for faster processing and making it suitable for deployment on devices with limited resources. These enhancements collectively improve the model’s real-time processing capabilities and overall detection performance, making YOLOv8 highly effective for safety helmet detection in real-world, dynamic environments.

CHAPTER 8

CONCLUSION AND REFERENCES

8.1 CONCLUSION

its positive impact on helmet wearing detection for enhanced workplace safety is evident. However, existing helmet detection models face challenges in recognizing small targets and complex backgrounds. This study proposes and implements an improved algorithm named YOLOv8n-SLIMCA to address these issues. Through a series of comparative and ablation experiments, the following conclusions are drawn: Adopting the Slim-Neck structure for feature fusion in the backbone network significantly reduces the model’s size and computational load. Specifically, FLOPs decreased by 9.76%, parameters decreased by 6.98%, and speed improved by 9.52%, with minimal compromise on accuracy. Hence, the Slim-Neck structure proves to be an excellent lightweight module. Secondly, introducing Mosaic data augmentation, a small target detection layer, and the CA module effectively improves accuracy. Mosaic data augmentation enriches the dataset with small scale helmet samples; the small target detection layer aids the model in focusing on multiscale features, especially for small sized targets, thereby enhancing the accuracy of small target helmet detection. The CA attention module outperforms SE and CBAM attention mechanisms, allowing more focused attention on crucial regions and reducing interference from complex backgrounds. In summary, the proposed YOLOv8n-SLIM-CA algorithm, compared to the YOLOv8n algorithm, achieves a 2.151% improvement in mAP@0.5, reaching 94.361%. Its detection performance surpasses other algorithms in scenarios involving small targets, dense targets, and complex environments. This algorithm meets real-time and accuracy requirements for helmet detection and has low computational demands, with 11.3GB FLOPs, 2.74MB parameters, and 2.3 ms inference speed. It is suitable for deployment on mobile and edge devices, making it applicable for monitoring construction site videos and having broad applications in the industrial sector.

8.2 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.

[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 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.

[12] H. Liu, X. Wang, W. Zhang, Z. Zhang, and Y.-F. Li, ''Infrared head pose estimation with multi-scales feature fusion on the IRHP database for human attention recognition,'' Neurocomputing, vol. 411, pp. 510–520, Oct. 2020.

[13] W. Yang, G. L. Zhou, Z. W. Gu, X. D. Jiang, and Z. M. Lu, ''Safety helmet wearing detection based on an improved YOLOv3 scheme,'' Int. J. Innov. Comput., Inf. Control, vol. 18, no. 3, pp. 973–988, 2022.

[14] R. Cheng, X. He, Z. Zheng, and Z. Wang, ''Multi-scale safety helmet detection based on SAS-YOLOv3-tiny,'' Appl. Sci., vol. 11, no. 8, p. 3652, Apr. 2021.

[15] J. Chen, S. Deng, P. Wang, X. Huang, and Y. Liu, ''Lightweight helmet detection algorithm using an improved YOLOv4,'' Sensors, vol. 23, no. 3, p. 1256, Jan. 2023.

[16] Z. Li, W. Xie, L. Zhang, S. Lu, L. Xie, H. Su, W. Du, and W. Hou, ''Toward efficient safety helmet detection based on YoloV5 with hierarchical positive sample selection and box density filtering,'' IEEE Trans. Instrum. Meas., vol. 71, pp. 1–14, 2022.

[17] L. Wang, X. Zhang, and H. Yang, ''Safety helmet wearing detection model based on improved YOLO-M,'' IEEE Access, vol. 11, pp. 26247–26257, 2023.

[18] J. Bao, S. Li, G. Wang, J. Xiong, and S. Li, ''Improved YOLOV8 network and application in safety helmet detection,'' J. Phys., Conf. Ser., vol. 2632, no. 1, Nov. 2023, Art. no. 012012. [19] F. M. Talaat and H. ZainEldin, ''An improved fire detection approach based on YOLO-v8 for smart cities,'' Neural Comput. Appl., vol. 35, no. 28, pp. 20939–20954, 2023.

[20] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, ''A modified YOLOv8 detection network for UAV aerial image recognition,'' Drones, vol. 7, no. 5, p. 304, May 2023.
[21] D. Wu, S. Jiang, E. Zhao, Y. Liu, H. Zhu, W. Wang, and R. Wang, ''Detection of camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation,'' Appl. Sci., vol. 12, no. 22, p. 11318, Nov. 2022.
[22] Z. Cao, M. Shao, L. Xu, S. Mu, and H. Qu, ''MaskHunter: Real-time object detection of face masks during the COVID-19 pandemic,'' IET Image Process., vol. 14, no. 16, pp. 4359–4367, Dec. 2020.
[23] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, ''Real-time face mask detection method based on YOLOv3,'' Electronics, vol. 10, no. 7, p. 837, Apr. 2021.
[24] G. Zhang, J. Yang, Y. Zheng, Y. Wang, Y. Wu, and S. Chen, ''Hybridattention guided network with multiple resolution features for person reidentification,'' Inf. Sci., vol. 578, pp. 525–538, Nov. 2021.
[25] J. Hu, L. Shen, and G. Sun, ''Squeeze-and-excitation networks,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.
[26] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, ''CBAM: Convolutional block attention module,'' in Proc. Eur. Conf. Comput. Vis., 2018.
[27] Q. Hou, D. Zhou, and J. Feng, ''Coordinate attention for efficient mobile network design,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 13708–13717. [28] F. Chollet, ''Xception: Deep learning with depthwise separable convolutions,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1800–1807.
[29] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, ''ShuffleNet v2: Practical guidelines for efficient CNN architecture design,'' in Proc. Eur. Conf. Computer Vis., 2018.
[30] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, ''Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles,'' 2022, arXiv:2206.02424.
[31] K. Zhang, Y. Wu, J. Wang, Y. Wang, and Q. Wang, ''Semantic contextaware network for multiscale object detection in remote sensing images,'' IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2022.
[32] X. Li, C. Fu, X. Li, and Z. Wang, ''Improved faster R-CNN for multi-scale object detection,'' J. Comput.-Aided Design Comput. Graph., vol. 31, no. 7, p. 1095, 2019. [33] J. Wang, G. Zhu, S. Wu, and C. Luo, ''Worker's helmet recognition and identity recognition based on deep learning,'' Open J. Model. Simul., vol. 9, no. 2, pp. 135–145, 2021.
[34] S. Yue, Q. Zhang, D. Shao, Y. Fan, and J. Bai, ''Safety helmet wearing status detection based on improved boosted random ferns,'' Multimedia Tools Appl., vol. 81, no. 12, pp. 16783–16796, May 2022.
[35] H. Sun and P. Gong, ''A safety-helmet detection algorithm based on attention mechanism,'' in Proc. 7th IEEE Int. Conf. Netw. Intell. Digit. Content (IC-NIDC), Nov. 2021, pp. 11–15.
[36] Y. Liu, B. H. Lu, J. Peng, and Z. Zhang, ''Research on the use of YOLOv5 object detection algorithm in mask wearing recognition,'' World Sci. Res. J., vol. 6, no. 11, pp. 276–284, 2020.