YOLO V8: An improved real-time detection of safety equipment in different lighting scenarios on construction sites

Lakshmi Thara R
lakshmir13@srmist.edu.in

SRM Institute of Science and Technology College of Engineering, SRM Nagar
Bhavya Upadhyay
SRM Institute of Science and Technology College of Engineering, SRM Nagar
Ananya Sankrityayan
SRM Institute of Science and Technology College of Engineering, SRM Nagar

Research Article

Keywords: Computer vision (CV), Image & Video processing, Safety equipment, YOLOv8, Accuracy

Posted Date: April 2nd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4179998/v1

Additional Declarations: No competing interests reported.

Abstract

In this work we aimed to detect the safety equipment worn by the workers on construction site using the YOLOv8 model. Its a state-of-the-art deep learning model recognized for its speed and accuracy, in detecting objects within dynamic construction environments. Focusing on classes such as Helmet, Vest, Gloves, Human, and Boots, we assess YOLOv8's efficacy in real-time safety hazard detection. The classes have been labelled using the labelImg software for training the model, with that the testing of different images and videos have been carried out. After deploying the trained model, it shows an impressive accuracy rate of approximately 98.017% with the YOLOv8 model, surpassing previous iterations. Additionally, our Recall and Precision values achieve high levels at 94.9% and 94.36% respectively, while the F1 score and mean Average Precision (mAP) values approximate 91% and 91.9% respectively. These robust performance metrics underscore the reliability and effectiveness of YOLOv8 compared to other existing YOLO models, marking a significant advancement in object detection for construction site management.

1. Introduction

Construction projects are multifaceted environments characterized by constant activity, diverse equipment, materials, and changing conditions. Traditional methods for monitoring progress and ensuring safety rely on manual inspections, which are time-consuming, labor-intensive, prone to errors, and may expose workers to additional risks. The emergence of computer vision (CV) techniques, particularly object detection, offers a promising solution for automating various aspects of construction site management. Object detection enables computers to identify and localize specific objects within images or videos, facilitating automated progress monitoring, resource tracking, and real-time safety hazard detection. This leads to improved efficiency, accuracy, and worker safety. Among various object detection models, YOLOv8 is notable for its real-time performance and high accuracy. With its single-stage detection architecture, YOLOv8 enables quick predictions without compromising precision, making it particularly suitable for construction sites where real-time monitoring and immediate response to safety hazards are imperative. This research aims to explore YOLOv8's potential for object detection in construction environments, evaluating its effectiveness in identifying a wide range of objects relevant to construction workflows, including workers, equipment, materials, and potential safety hazards. By comparing YOLOv8's performance to existing solutions, this study aims to provide valuable insights for the development of automated construction site monitoring systems. The construction industry faces challenges related to safety and operational efficiency, where traditional monitoring methods often fall short. This project investigates the integration of YOLO (You Only Look Once), a machine learning model, to revolutionize construction site safety and object detection. In light of the limitations of manual safety checks, YOLO's real-time object detection capabilities offer a promising solution. Its swift processing and single-pass approach enable YOLO to identify and categorize objects seamlessly within construction environments. YOLO facilitates real-time object detection, addressing limitations of manual safety checks, identifying safety gear compliance, construction equipment, and materials. Furthermore, it enhances safety protocols, streamlines operations, and tackles challenges such as privacy concerns and model accuracy. YOLO-based solutions promise improved safety, reduced incidents, optimized resource allocation, and informed decision-making, ultimately aiming to transform the construction industry. Construction sites represent dynamic ecosystems, constantly evolving with diverse activities, equipment, and materials. Manual monitoring and safety management on these sites are time-consuming, labor-intensive, and error-prone, often exposing workers to additional risks. The emergence of computer vision techniques, particularly object detection, offers a promising avenue for automating various aspects of construction site management. Object detection, a subfield of CV, enables computers to identify and locate specific objects within images or videos. This technology holds immense potential for construction sites, where real-time monitoring and immediate response to safety hazards are paramount.

In 2022, Duan et al [1] introduced a project providing a pre-trained YOLOv8 model and a dedicated dataset specifically designed for construction site object detection tasks, termed the Site Object Detection Dataset (SODA), comprising 15 object classes, over 20,000 diverse images, and 286,201 annotated objects. While showcasing a YOLOv5-based system effectively monitoring hard hat usage, limitations include dataset bias, reliance on specific object classes, potential generalization issues, and limited applicability to diverse construction environments. In the year 2020, Jin et al. [2] utilized the Faster R-CNN model for automated progress monitoring, demonstrating its capability to track specific construction phases and elements, yet highlighting dependencies on high-quality image data, computational complexity, challenges with occlusion, and limited scalability to large sites. In 2022, Han et al. [3] conducted a review of vision-based techniques for safety monitoring, including earlier versions of YOLO, emphasizing limitations such as dataset size and diversity, algorithm scalability, challenges in detecting small or occluded objects, and computational requirements. In our study, we referenced the research conducted by [4] Sary et al. compared the performance of YOLOv5 and YOLOv8 architectures in the year of 2023 for human detection using aerial images in that they utilized the anchor boxes and grid cells to facilitate accurate detection across varying scales within the images. In the year 2020, D Benyang et al. [5] presented a safety helmet detection method using YOLOv4, noting potential limitations in detecting helmets under diverse conditions, scalability to large sites, and real-time processing speed challenges. In 2023, Zhang et al. [6] investigated YOLOv8 for real-time object detection in construction progress monitoring, highlighting dependencies on image quality, limited accuracy in complex environments, and challenges with occluded objects. In 2022, Docto et al. [7] developed a system for image capture and object detection with limitations in scalability for outdoor use and varying lighting conditions. In the year 2019, Zhao et al. [8] proposed enhancements for object perception using a TurtleBot2 robot, noting challenges in real-world adaptability and scaling. In 2022, Zhu et al. [9] presented a UAV tracking method based on gaze prediction, with constraints in accuracy and real-time implementation. In the year 2023, Abrar et al. [10] introduced a novel approach by combining computer vision techniques with YOLOv8 for the purpose of accurately counting people. However, their study highlighted certain limitations concerning the representativeness and generalization of the dataset used, indicating avenues for further research and improvement in this domain. In the year 2023, Lu et al.'s [11] paper provided valuable insights into deep learning-based object detection algorithms and aerial imagery analysis. Incorporating these techniques into our construction site object detection project proved instrumental, enhancing accuracy and efficiency while aligning with industry advancements to improve safety protocols at construction sites. In 2021, Wang et al. [12] on fast personal protective equipment detection for Real Construction Sites Using Deep Learning Approaches, in Sensors. Their research focuses on implementing deep learning approaches for rapidly detecting personal protective equipment (PPE) in real construction sites, contributing to improved safety measures. In 2021, Liu et al. [13] introduced a helmet-wearing detection system utilizing YOLOv4-MT. Their effective identification of individuals without helmets, crucial for accident prevention in various settings, significantly enhances safety precautions. This innovation underscores their commitment to advancing safety measures through cutting-edge technology. In 2020, Chen et al. [14] presented a study on automatically identifying and analyzing the productivity of excavator operations using surveillance footage on building sites, the study provided insights into optimizing construction processes through effective activity monitoring and analysis, showcasing advancements in construction site management. In the year 2024, Gao et al. [15] applied an enhanced YOLOv8 version to improve item recognition in Jiangnan traditional private gardens, aiming to boost detection accuracy, their work significantly advances detection techniques, crucial for preserving cultural assets in historical contexts. In the year 2022, Song [16] introduces a multi-scale safety helmet detecting system in Sensors. Built on RSSE-YOLOv3, this system enhances safety precautions by reliably identifying safety helmets across various scales, particularly benefiting sectors reliant on helmet compliance for worker protection. In the year 2021, Elghaish et al. [17] provided a comprehensive examination of deep learning's application in building site management, the study explored the scient metric, thematic, and critical domains, offering insights into deep learning approaches' role in enhancing construction site management and operations. In 2021, Shao et al. [18] on a machine vision-based intelligent wearable detection approach. The study explores machine vision improvements for enhancing wearable device detection capabilities, showing potential applications across industries such as healthcare and logistics. In 2020, Yousif et al. [19] introduced an optimized neutrosophic K-means with genetic algorithm for automatic vehicle license plate recognition. This methodology could offer insights for improving construction site object detection systems. In 2018, Kulkarni et al. [20] implemented automatic number plate recognition for helmetless motorcyclists. Inspired by their work, we aimed to utilize their deep learning techniques for enhancing object detection in construction site monitoring. In the year 2020, Pustokhina et al. [21] proposed an autonomous vehicle license plate identification system for intelligent transportation systems. Combining convolutional neural networks and optimal K-means, the study aimed to enhance security and efficiency in transportation infrastructure, it underscored advancements in vehicle surveillance technology. In the year 2015, Aalsalem et al. [22] introduced an automated vehicle parking monitoring and management system using ANPR cameras. This innovation offers insights into ANPR technology for efficient parking management, contributing to transportation research and technology advancements. In the year 2020, Shashirangana et al. [23] conducted an extensive survey on automated license plate recognition systems. The study utilized a variety of approaches and shedding light on developments in this area, enhancing transportation and security applications required a deeper understanding of license plate recognition systems.

The success of this research project relies on achieving a minimum accuracy threshold of 80–82% with the YOLOv8 model, surpassing previous iterations. By enhancing object detection accuracy in dynamic construction environments, we aim to improve progress monitoring, resource allocation, and real-time safety hazard detection. YOLOv8 shows promise as a robust solution for construction site management, contributing to safer and more efficient work environments. This research expands the knowledge base on utilizing computer vision for construction site management, aiming to provide a safer and more efficient environment for workers and stakeholders. For our specific requirements, we selected the YOLOv8-M model, tailored for detecting small and densely-packed objects in images and videos. Through adjustments aligned with contemporary neural network design principles, we refined the detection network structure to offer greater comprehensiveness and detail. This adapted version is referred to as YOLOv8. The overall model architecture is depicted in Fig. 1.

In Fig. 1 is the core architecture of an object detection model, encompassing a backbone, neck, and head. The backbone, a pre-trained Convolutional Neural Network (CNN) which extracts multi-level features from input images, while the neck utilizes path aggregation blocks like the Feature Pyramid Network (FPN) which then merges these features to enhance object perception across various scales. Finally, the head component is responsible for object classification and bounding box prediction, offering flexibility with one-stage models such as YOLO or SSD for rapid predictions or two-stage models like the R-CNN series for precise localization. Together, these components constitute a robust framework for accurate and efficient object detection in computer vision applications.

2. Proposed System & its Methodology

The study introduces a methodology for gathering a diverse range of construction images via web mining. These images undergo preprocessing and organization before being split for training and validation. Precision performance is evaluated using different analysis methods and test scenarios for the YOLO model(18). Leveraging its capability to handle varying sizes, the YOLOv8 model enhances detection accuracy and performance.

In Fig. 2, the initial stage of the object detection process involves a classifier generating numerous region proposals, commonly referred to as bounding boxes, across the image. These proposals delineate areas likely to contain objects of interest. Subsequently, a secondary classifier refines these proposals by assigning precise class labels to each, thereby identifying specific objects within the bounding boxes, such as a fire hydrant or a stack of bricks. This two-stage methodology aims to enhance both the speed and accuracy of object detection. By initially generating a broad array of proposals, the subsequent refinement stage can concentrate computational resources on the most promising regions of interest within the image. The accompanying text within the image pertains to Non-Maximum Suppression (NMS), a pivotal technique employed to eliminate redundant bounding boxes. Given that the initial classifier may generate multiple bounding boxes for the same object, NMS functions to prune these redundancies, ensuring the output comprises only the most pertinent and high-quality bounding boxes.

In Fig. 3A flowchart outlining a process for training an object detection model. The process starts with data collection, where a diverse set of construction site images are acquired through crowdsourcing and web-mining. Next, data preprocessing cleans and organizes the collected data to ensure consistency and relevance. The data is then split into training and validation sets. The training set is used to train the model, while the validation set is used to evaluate the model's performance. After training, the model's performance is assessed using metrics like precision, recall, and F1-score. These metrics measure how well the model can identify objects in the images. The flowchart then shows a step for analyzing the model across different scenarios. It involves testing the model's ability to detect objects of varying sizes, in different lighting conditions, or in crowded environments. Finally, based on the results of the analysis, the process moves to performance enhancement. Here, researchers would explore methods to improve the accuracy of the object detection model. The parameters considered include:

After extensive training, our model has been refined using 5 crucial parameters as shown in Fig. 4 are essential for construction site safety monitoring. These parameters encompass the detection of various objects and scenarios. The project development process encompasses several key stages, notably Data Collection, Data Preprocessing, Training & Validation, Performance Evaluation, and Performance Enhancement. Specifically focusing on safety measures. In this study, we construct a construction site workers image dataset which contains 5 classes targets. We generated various images using internet and also visited some construction sites and collected approximately 2092 images using different angles at different times of the day,

A. Data Collection and Preparation: Traditional approaches to dataset construction often involve retrieving images from web search engines, initially attempted through the utilization of web crawlers and similar tools. However, this method yielded limited results due to the disorderly nature of construction sites, where objects of interest were frequently entangled with unrelated items. Consequently, our dataset comprised solely of images directly collected from the internet and real construction sites [1]. Two distinct methodologies were explored for dataset acquisition: leveraging publicly available datasets and creating a tailored custom dataset to align with research specifications. Public datasets underwent scrutiny for relevance, while custom dataset creation involved the meticulous capture of high-quality images and videos at construction sites to ensure representation across various project phases, lighting conditions, and object categories. Throughout the data collection process, several noteworthy challenges were encountered, offering valuable insights for similar endeavors: In the context of construction sites, where machinery is relatively sparse compared to workers and materials, it is imperative to meticulously capture a diverse array of shots focusing on specific targets from varying angles during photography. The complexity and clutter inherent in construction sites, coupled with potential visual obstructions and occlusions, make obtaining positive samples via a single shooting method arduous, necessitating the integration of diverse shooting techniques. Furthermore, while data obtained from cameras positioned at elevated vantage points on-site can offer panoramic views and capture large machinery such as tower cranes, the clarity of video data for labeling smaller objects may be compromised.

B. Data Cleaning: Data cleaning is crucial in preparing datasets for object detection using YOLOv8, especially in construction site monitoring. Several essential factors require attention to ensure the model's efficacy and precision:

- Noise Reduction: Employ filtering methods like Gaussian blur to remove irrelevant elements such as debris and vehicles from images, enhancing the dataset's clarity and focus.
- Handling Occlusions: Utilize advanced annotation techniques, such as instance segmentation, to accurately label obscured objects, ensuring the model's robustness against obstructed views.
- Accounting for Variability: Augment the dataset with variations in lighting conditions and scene complexities using techniques like data augmentation and histogram equalization, improving the model's adaptability to diverse environmental factors.
- Consistent Annotation: Adhere strictly to labeling standards and conduct regular quality assessments to maintain consistency and precision across annotations, ensuring the reliability of the dataset for training.
- Balancing Class Distribution: . Address any disparities in class distributions by either gathering additional samples for underrepresented classes or utilizing resampling methods such as oversampling or under sampling to achieve a more balanced representation of different classes in the dataset.

C. Data Annotation: Accurate dataset annotation is crucial for ensuring the effectiveness of object detection models like YOLOv8. In this model, objects are annotated using rectangular bounding boxes, with corresponding class labels assigned to each box. To annotate images for YOLOv8, labeling tools such as LabelImg or YOLO_mark are typically employed. This involves drawing bounding boxes around objects of interest and assigning class labels to them, indicating the type of object represented. Coordinate annotation entails recording the coordinates of the bounding boxes' top-left corner, width, and height, usually normalized relative to the image dimensions. These annotations are saved in formats compatible with YOLOv8's requirements, such as YOLO text format or XML format, and organized alongside the images in a directory structure suitable for training. Following this annotation process ensures datasets are effectively prepared for training YOLOv8, thereby facilitating accurate object detection. The LabelImg tool as shown in **Fig 5** has been employed for manual annotation to enhance efficiency and accuracy. Particular attention has given to ensuring tight bounding boxes for each object, addressing issues such as the misidentification of hats and vests. For each image, corresponding annotation files were generated, containing class and bounding box coordinate information for each target in YOLO format label files.

3. Results & Discussion

A study has been conducted on construction site object detection utilizing YOLOv8 model, employing a comprehensive methodology, when comparing YOLOv5 and YOLOv8, both renowned for image and video processing, our analysis centers on crucial factors: architecture, performance, and practicality, guiding us in discerning their respective strengths and limitations. Initially, a dataset is being collected, consisting of images featuring construction site workers and five target classes: Helmet, Vest, Gloves, Boots, and Humans. Around 2092 images were gathered from internet databases and actual construction sites, ensuring diversity in angles, lighting conditions, and contexts. Challenges encountered during data collection included capturing a comprehensive range of shots, managing occlusion and clutter, and addressing issues with data clarity from high vantage points. Following data collection, data cleaning procedures were rigorously implemented to ensure the integrity and quality of the dataset. This encompassed comprehensive measures to reduce noise, handle occlusions, and account for variability in lighting conditions and scene complexities. To reduce noise, advanced image processing techniques such as filtering and segmentation were applied, effectively eliminating irrelevant elements such as debris and vehicles. Additionally, sophisticated annotation methods were employed to accurately annotate obscured objects and incorporate diverse occlusion scenarios, thereby enhancing the model's robustness. Furthermore, augmentation techniques were utilized to address variability in lighting conditions and scene complexities, ensuring the dataset's adaptability. Consistent annotations were maintained through strict adherence to labeling conventions and regular quality checks. Moreover, efforts were made to balance class distribution by collecting additional samples for underrepresented classes and employing resampling techniques where necessary. Overall, these rigorous data cleaning procedures were instrumental in preparing a high-quality dataset for subsequent analysis and model training, ensuring accurate and reliable results without compromising on integrity or quality. Accurate annotation of objects using rectangular bounding boxes and class labels is crucial, facilitated by manual annotation using the LabelImg tool. The YOLOv8 model underwent training and optimization, with evaluation of various model variants to strike a balance between precision and computational efficiency. Hyperparameter tuning techniques were applied, including adjustments to learning rate, batch size, and optimizer configurations. The efficacy of the trained YOLOv8 model has been assessed using standard metrics such as precision, recall, and mean average precision. Overall, the thorough approach ensured that the YOLOv8 model has been trained and optimized to effectively detect objects of interest on construction sites, thereby enhancing safety measures and monitoring capabilities in real-world scenarios.

Table 1
The identified TP, TN, FP, FN values from YOLOv8 detection model

Parameters	TP (True Positive)	TN (True Negative)	FP (False Positive)	FN (False Negative)
Boots	560	1265	33	30
Gloves	30	1837	11	30
Helmets	307	1553	13	15
Human	360	1483	16	20
Vest	450	1399	107	22

Table 2
Performance metrics calculated for YOLOv8 model

Parameters	Accuracy	Recall	Precision
Boots	96.6%	94.9%	94.4%
Gloves	98.9%	75%	73.1%
Helmets	98.5%	95.3%	95.9%
Human	97.6%	94.7%	95.7%
Vest	97.9%	95.3%	96.3%

The Table 1 presents the metrics for evaluating the performance of the model, including True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) that has been obtained from the trained dataset of 2092 images for the epoch of 20. Using the values obtained from Table 1, the performance metrics such as Accuracy, Recall and Precision have been calculated and illustrated in Table 2. The formula for calculating the Precision, Accuracy and Recall are given in equations (1), (2), and (3) respectively. Such metrics offer significant insights into the model's efficacy in accurately identifying and categorizing objects, thereby contributing to informed decision-making across diverse applications.

$$Precision = \frac{TruePositives\,(Tp)}{TruePosivites\,(Tp) + FalsePosivitves\,(Fp)}$$

1

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn}$$

2

$$Recall = \frac{TruePositives\,(Tp)}{TruePositives\,(Tp) + FalseNegatives\,(Fn)}$$

3

From Fig. 7 we can depict a model's performance measures throughout several training epochs. Here epochs indicate both the individual epoch number and the total epoch count. For instance, "17/20" indicates that this era is the 17th of 20th total epochs. With 20 epochs, the YOLOv8 object detection training procedure started. Several indicators were tracked during the training phase in order to evaluate the model's development and performance. The particulars of each epoch, which includes processing speed, instance counts, GPU memory utilization, and loss values, were noted and analyzed.

During the initial epoch, GPU memory utilization peaked at 7.16 GB, accompanied by box, cls, and dfl losses of 1.017, 0.079, and 1.239, respectively, with the model exhibiting a size of 640 and detecting 107 instances at a processing speed of 1.88 iterations per second. Promising results across evaluation metrics were observed, indicating potential performance enhancements with continued training. Subsequent epochs displayed consistent trends, albeit with variations in processing speeds, instance counts, and loss values. Notably, metrics such as precision, recall scores, and loss values demonstrated positive trends, culminating in a significant decrease in box, cls and dfl losses to 0.5322, 0.2684, and 0.9351 respectively, by the 20th epoch, with the GPU memory usage stabilizing at 7.42GB. Despite identifying fewer instances, the model showcased notable performance improvements achieving a processing speed of 1.97 iterations per second and a remarkable increase in mean average precision at 50% IoU (mAP50) from 71.9–91.9% over the course of training. The efficiency of the training process is being further underscored by completing all 20 epochs within a mere 0.850 hours or 51 minutes, highlighting both the effectiveness of the training setup and the model's rapid learning capabilities. This thorough analysis highlights how model training progresses iteratively, stressing the importance of steady performance improvements and increased confidence in the model's abilities, which are essential for practical applications across different domains.

In Fig. 8, a confusion matrix is presented, a fundamental tool frequently utilized to assess the performance of classification models on a designated test dataset where true values are known. This particular matrix originates from a model designed to classify images into five distinct categories: boots, gloves, helmets, humans, and vests. The matrix is structured with the true labels on the y-axis (vertical axis) and the predicted labels on the x-axis (horizontal axis). Each cell within the matrix denotes the count of predictions made by the model, with the rows representing the true labels and the columns representing the predicted labels. For instance, the cell located in the 'boots' row and 'boots' column indicates 560 true positive predictions for boots.

The primary diagonal of the matrix, spanning from the top left to the bottom right, showcases the number of accurate predictions for each category, where the predicted label aligns with the true label. Ideally, in a perfectly performing classifier, all values would lie exclusively on this diagonal, with off-diagonal cells registering zero, signifying no misclassifications. However, upon inspection of this matrix, it becomes evident that the model has excelled in certain classes (such as 'human' and 'vest') but has also exhibited misclassifications, including instances of confusing 'gloves' with 'background' or 'boots' with 'vest'. The colour scale provided on the right side of the matrix denotes intensity corresponding to the number of observations, with darker shades typically indicating higher counts. This visual representation facilitates rapid identification of accurately classified categories and those frequently confused, thereby offering valuable insights into the model's performance nuances and areas for potential improvement.

In Fig. 9 Frame 103/322 shows a construction site, and it has been annotated that an object detection algorithm has been used to identify various elements. The object detection algorithm's study shows that there is a good level of confidence in detecting important components in the image, especially when it comes to safety and the presence of people at a work site. The algorithm shows a high degree of assurance, with most assurance scores being higher than 80%. It assigns assurance levels ranging from 0.71 to 0.94 and recognizes individuals, helmets, vests, and boots with proficiency. On the other hand, there are certain cases of label overlap and redundancy, particularly when people are identified repeatedly in particular image regions. The algorithm's segmented picture processing, which produces independent human feature detection, could be the source of this redundancy. Although this kind of duplication is typical in object detection, it emphasizes the need for sophisticated algorithms to maximize label assignments and improve overall efficiency. Despite these minor challenges, the algorithm's robust performance underscores its potential applicability in safety-critical environments, showcasing promising implications for real-world deployment and utilization.

After testing the trained model on both day and night lighting conditions, we concluded that the trained YOLOv8 model shows maximum accuracy possible even in dark lighting conditions as shown in Fig. 10, though it is observed that there is a slight drop in performance metrics of the night settings when compared to day settings. It is important to mention that though both images show different lighting conditions but they both are 2 different images as it wasn’t possible to capture the same scenario in different lighting conditions. Below is a comparison of the performance metrics of both day and night lighting condition.

Table 3
Performance metrics calculated for day and night lighting conditions

Lighting conditions	Accuracy	Recall	Precision
Day light setting	98.01%	94.36%	94.9%
Night light setting	94.12%	93.04%	91.1%

While the model's performance remains impressive in nighttime conditions, with an accuracy rate of 94.12% as shown in Table 3, there is a slight decrease compared to daytime conditions. However, this marginal drop is indicative of the model's consistent and commendable performance across varying lighting environments. The precision rate of 91.1% further validates the model's capability to effectively identify night lighting instances, showcasing its reliability in low-light scenarios. Additionally, the recall rate of 93.04% highlights the model's adeptness in accurately capturing the presence of night lighting conditions. Overall, the YOLOv8 model demonstrates exemplary performance across both daylight and nighttime settings, with its slight superiority in daylight conditions emphasizing its robustness and efficacy. These findings underscore the model's suitability and effectiveness in tasks requiring precise lighting condition detection, thereby affirming its value and applicability in practical scenarios.

Figure 11 presents a comprehensive depiction of diverse metrics tracked across 20 epochs during the training of a machine learning model, indicative of its deep learning nature, as inferred from the contextual relevance of the metric terminology. The horizontal axis spans from 0 to 20 epochs, representing the progression of training iterations, while the vertical axis denotes metric values. Each graph encapsulates performance metrics and loss values discerned during both the training (train) and validation (val) phases, providing an insightful visualization of the model's learning trajectory and performance evolution throughout the training process.

In the provided graphs (a, b, c), representing "train/box_loss", "train/cls_loss", and "train/dfl_loss" respectively, the consistent decrease in training box loss, classification loss, and directional feature loss with increasing epochs suggests an improvement in accuracy across box prediction, classification, and directional feature prediction. Similarly, graphs (d, e), depicting metrics for Precision and Recall, exhibit an increasing trend with the number of epochs, indicating enhanced precision and accuracy over time. This is attributed to the iterative learning process of the model, where each epoch iteration refines the model's understanding of the data. On the contrary, graphs (f, g, h), displaying validation box loss, classification loss, and directional feature loss, demonstrate plateauing trends, signaling potential model convergence or overfitting during the validation phase. The validation loss plateau may be attributed to the model's tendency to memorize the training data rather than generalize to new data, highlighting the need for regularization techniques to prevent overfitting. Moving to graphs (i, j), representing mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds, the increasing trend over 20 epochs indicates improved precision in object detection as training progresses. This is due to the model's ability to better discern objects from the background and accurately localize them within the image. Overall, the increase in epochs allows the model to iteratively adjust its parameters based on the training data, leading to improved performance metrics such as precision and recall. However, the plateauing of validation loss suggests a need for caution to avoid overfitting and ensure the model's generalization to unseen data.

Conclusion

This study conducted an in-depth exploration into training images and videos using a YOLOv8 model specifically designed for object detection, in construction sites. The groundwork involved the careful planning of a diverse dataset comprising of 2092 images collected from both online repositories and real-world construction environments. To ensure data quality, rigorous data cleaning procedures were implemented, focusing on noise reduction techniques using Gaussian blur to eliminate irrelevant elements such as debris and vehicles that enhanced the dataset clarity. Advanced annotation methods, including instance segmentation, are employed to accurately label obscured objects, ensuring model robustness against occluded views. Augmenting the dataset with variations in lighting conditions and scene complexities through techniques like data augmentation and histogram equalization improves model adaptability. The model's performance during training has been heavily reliant on accurate object annotations, which were individually labeled using the LabelImg tool. Learning rate, batch size, and optimizer settings were among the hyperparameters that were repeatedly experimented with in the training phase in order to improve the YOLOv8 model for construction site object detection. In order to evaluate the efficiency of the YOLOv8 model the accuracy, recall, and precision have been calculated by the trained and tested model across a variety of classes, including boots, gloves, helmets, humans, and vests. Periodic analysis of the model's outputs over multiple epochs revealed variations in processing speed i.e. 1.97 images/second, instance counts of 274 images, and by the 20th epoch a box loss of

0.5322 followed by a classification (cls) loss of 0.2684 and a directional feature (dfl) loss of 0.9351. These loss values are crucial indicators of the model's performance, with lower values generally reflecting better accuracy in object detection and classification. The box loss pertains to the accuracy of bounding box predictions, while the cls loss measures the accuracy of class predictions. Additionally, the dfl loss evaluates another aspect of prediction accuracy, possibly related to specific features within the detected objects. These loss metrics play a significant role in optimizing the model's training process and enhancing its overall performance. Certainly, apparent enhancements in performance indicators indicated how effectively the model adapted to and benefited from the training dataset. The YOLOv8 model showed substantial improvements in detection capabilities and a decrease in loss values by the 20th epoch. The model's effective memory management has been shown by the constant GPU memory utilization throughout the training period. These results demonstrate how the YOLOv8 model helps safety procedures and surveillance systems on construction sites by offering real-time object detection capabilities to reduce accidents and enhance monitoring procedures. The development of computer vision solutions tailored to the industrial safety and surveillance sectors is greatly helped by such insights.

Declarations

Author Contribution

Lakshmi Thara R: Define Problem, Editing, Supervision, Review the write-up, • Bhavay Upadhayay: Implementation of the problem, Software, Validation, Original draft preparation• Ananya: Implementation of the problem, Software, Validation, Original draft preparation

References

1. Duan, Rui et al. "SODA: Site Object Detection Dataset for Deep Learning in Construction." *arXiv* abs/2202.09554 (2022). https://doi.org/10.48550/arXiv.2202.09554
2. Jin, Ruoyu et al. “Detection of Personal Protective Equipment (PPE) Compliance on Construction Site Using Computer Vision Based Deep Learning Techniques.” Frontiers in Built Environment (2020). https://doi.org/10.3389/fbuil.2020.00136
3. Han, Kun and Xiangdong Zeng. “Deep Learning-Based Workers Safety Helmet Wearing Detection on Construction Sites Using Multi-Scale Features.” IEEE Access 10 (2022): 718-729. https://doi.org/10.1109/AIMV53313.2021.9670937
4. Sary, Indri & Andromeda, Safrian & Armin, Edmund. “Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images. Ultima Computing : Jurnal Sistem Komputer.” (2023). https://doi.org/https://doi.org/10.31937/sk.v15i1.3204
5. D. Benyang, L. Xiaochun and Y. Miao, "Safety helmet detection method based on YOLO v4," 16th International Conference on Computational Intelligence and Security (CIS), Guangxi, China, (2020) https://doi.org/10.3390/s22176702

6. Z. Zhang, Y. Tang, Y. Yang and C. Yan, "Safety Helmet and Mask Detection at Construction Site Based on Deep Learning," IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, (2023) http://dx.doi.org/10.17798/bitlisfen.1297952
7. J. P. Docto, A. I. Labininay and J. F. Villaverde, "Third Eye Hand Glove Object Detection for Visually Impaired using You Only Look Once (YOLO)v4-Tiny Algorithm," 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 2022, pp. 1-6, doi: 10.1109/IICAIET55139.2022.9936740.
8. Zhao, Yu & Huang, Ran & Hu, Biao. (2019). A Multi-Sensor Fusion System for Improving Indoor Mobility of the Visually Impaired. 2950-2955. http://dx.doi.org/10.1109/CAC48633.2019.8996578
9. A. Zhu, J. Yang and W. Yu, "A novel target tracking method of unmanned drones by gaze prediction combined with YOLO algorithm," 2021 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2021, pp. 792-797,doi: 10.1109/ICUS52573.2021.961499.
10. Abrar Elaoua; Mohamed Nadour; Lakhmissi Cherroun; Abdelfattah Elasri, “Real-Time People Counting System using YOLOv8 Object Detection, 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)”, 28-29 November 2023, http://dx.doi.org/10.1109/IC2EM59347.2023.10419684
11. Lu, Dunlu & Ye, Jianxiong & Wang, Yangxu & Yu, Zhenghong. “Plant Detection and Counting: Enhancing Precision Agriculture in UAV and General Scenes”. IEEE Access. (2023). PP. 1-1. 10.1109/ACCESS.2023.3325747.
12. Wang, Zijian, Yimin Wu, Lichao Yang, Arjun Thirunavukarasu, Colin Evison, and Yifan Zhao. 2021. "Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches" *Sensors* 21, no. 10: 3478. https://doi.org/10.3390/s21103478
13. J. Liu and L. Liu, "Helmet Wearing Detection Based on YOLOv4-MT," 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE), Wuhan, China, 2021, pp. 1-5, doi: 10.1109/RCAE53607.2021.9638914.
14. Chen, Chen, et al. “Automated Excavators Activity Recognition and Productivity Analysis from Construction Site Surveillance Videos.” Automation in Construction, vol. 110, Feb. 2020, p. 103045. Crossref, https://doi.org/10.1016/j.autcon.2019.103045.
15. Gao, C., Zhang, Q., Tan, Z. et al. Applying optimized YOLOv8 for heritage conservation: enhanced object detection in Jiangnan traditional private gardens. Herit Sci 12, 31 (2024). https://doi.org/10.1186/s40494-024-01144-1
16. Song H. “Multi-Scale Safety Helmet Detection Based on RSSE-YOLOv3. Sensors (Basel)”. 2022 Aug 13. doi: 10.3390/s22166061.
17. Elghaish F, Matarneh ST, Alhusban M “The application of “deep learning” in construction site management: scientometric, thematic and critical analysis”. (2021), https://doi.org/10.1108/CI-10-2021- 0195
18. X. Shao, H. Li and S. Liu, "Research on Intelligent Wearable Detection Method Based on Machine Vision", 2021 IEEE 3rd International Conference on Civil Aviation Safety and

InformationTechnology(ICCASIT),2021.http://dx.doi.org/10.1109/ICCASIT53235.2021.9633505

19. B. B. Yousif, M. M. Ata, N. Fawzy and M. Obaya, "Toward an optimized neutrosophic K-means with genetic algorithm for automatic vehicle license plate recognition (ONKM-AVLPR)", IEEE Access, 2020. https://doi.org/10.1109/ACCESS.2020.2979185.
20. Y. Kulkarni, S. Bodkhe, A. Kamthe and A. Patil, "Automatic number plate recognition for motorcyclists riding without helmet", 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), pp. 1-6, 2018.https://doi.org/10.53730/ijhs.v6nS1.7537
21. I. V. Pustokhina, D. A. Pustokhin, J. J. Rodrigues, D. Gupta, A. Khanna, K. Shankar et al., "Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems", Ieee Access, vol. 8, pp. 92907-92917, 2020.http://dx.doi.org/10.1109/ACCESS.2020.299300823
22. M. Y. Aalsalem, W. Z. Khan and K. M. Dhabbah, "An automated vehicle parking monitoring and management system using ANPR cameras", 2015 17th International Conference on Advanced Communication Technology (ICACT), 2015.https://doi.org/10.1016/j.trpro.2016.05.372
23. J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, "Automated license plate recognition: a survey on methods and techniques", IEEE Access, vol. 9, pp. 11203-11225, 2020.http://dx.doi.org/10.24425/ijet.2023.144361

Figures

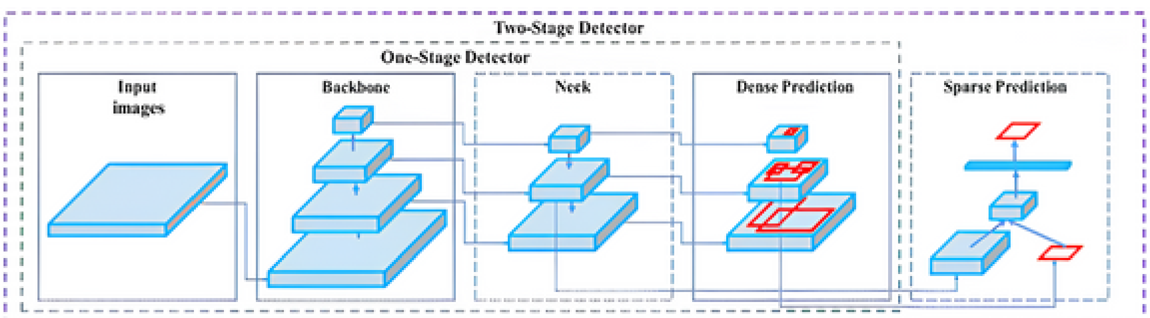

Figure 1

Object Detection Model Architecture

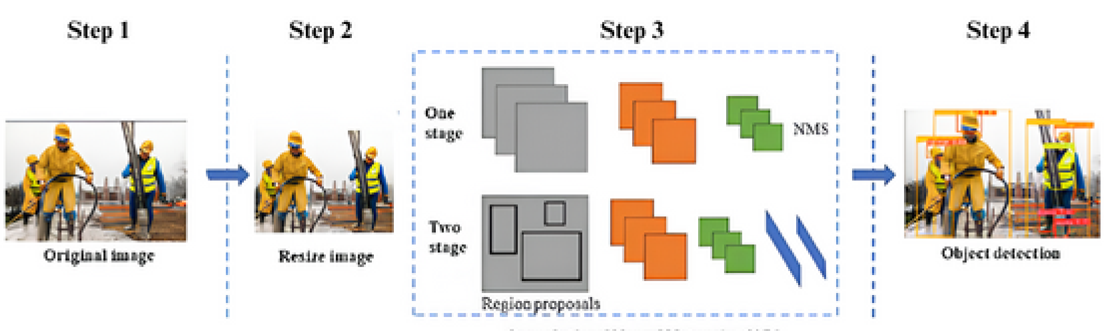

Figure 2

Visual representation of the object detection algorithm

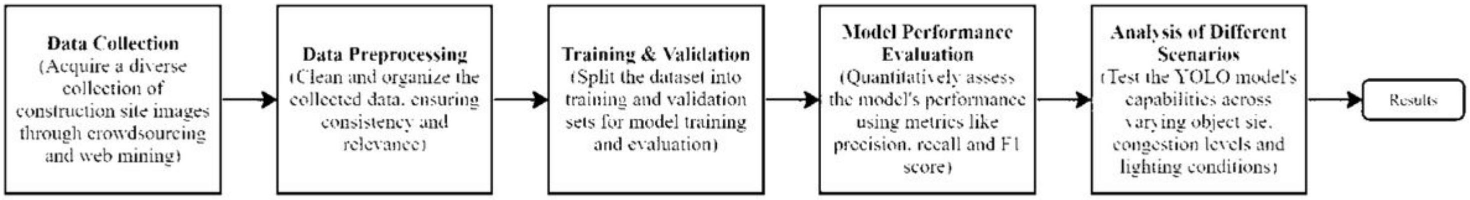

Figure 3

Block diagram of construction data collection

```
yolo_yaml = {
  'path': dataset_root,
  'train': os.path.join(dataset_root, 'train/images'),
  'val': os.path.join(dataset_root, 'val/images'),
  'test': os.path.join(dataset_root, 'test/images'),
  'names': {
    0: 'Helmet',
    1: 'Vest',
    2: 'Boots',
    3: 'Gloves',
    4: 'Human'
  }
}
```

Figure 4

Parameters used to train the model

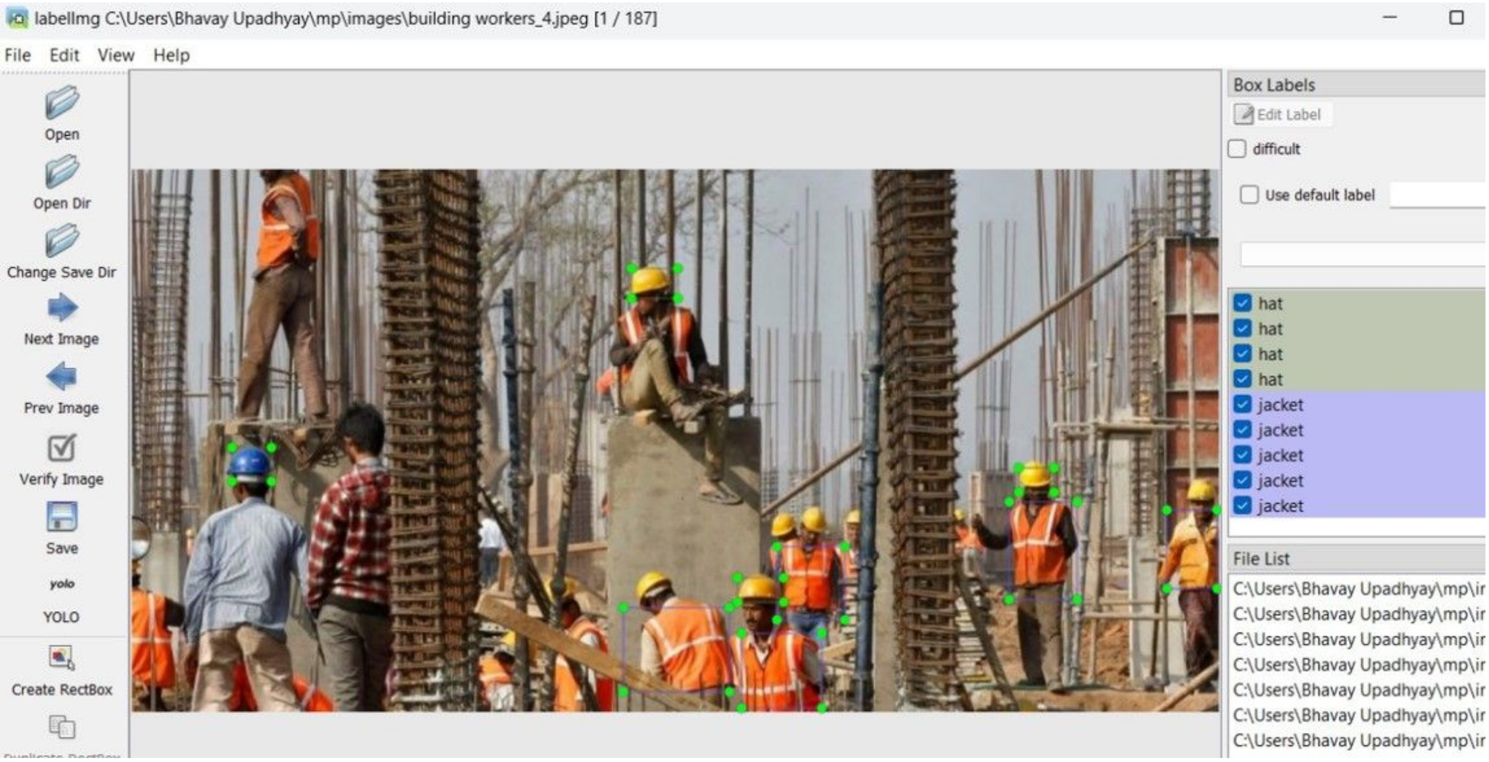

Figure 5

Class label assigned to each box using labelImg

```
Starting training for 20 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       1/20      7.16G      1.017      1.079      1.239        107        640: 100% 274/274 [02:25<00:00,  1.88it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:07<00:00,  1.75it/s]
                   all        420       1797      0.732       0.66      0.719      0.519

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       2/20      7.39G     0.9842     0.8084      1.208         89        640: 100% 274/274 [02:21<00:00,  1.93it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.12it/s]
                   all        420       1797      0.746      0.658      0.716      0.512

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       3/20      7.43G     0.9638     0.7587      1.191         78        640: 100% 274/274 [02:21<00:00,  1.94it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.24it/s]
                   all        420       1797      0.753      0.673      0.738      0.541

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/20      7.41G     0.9228     0.7084       1.17        109        640: 100% 274/274 [02:20<00:00,  1.95it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.05it/s]
                   all        420       1797      0.738      0.769      0.802      0.584

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/20      7.42G     0.8491     0.6095      1.123         62        640: 100% 274/274 [02:20<00:00,  1.95it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.11it/s]
                   all        420       1797      0.735      0.797      0.813      0.617

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/20       7.4G      0.829     0.5771      1.111        102        640: 100% 274/274 [02:19<00:00,  1.96it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.13it/s]
                   all        420       1797      0.812      0.776       0.82      0.629

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       8/20      7.39G     0.7944     0.5447      1.092        116        640: 100% 274/274 [02:20<00:00,  1.95it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.24it/s]
                   all        420       1797      0.782       0.78      0.834      0.646

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       9/20      7.42G     0.7802       0.52       1.08         98        640: 100% 274/274 [02:20<00:00,  1.95it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.28it/s]
                   all        420       1797      0.809      0.826      0.859      0.668

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      10/20      7.39G      0.752      0.491      1.064        103        640: 100% 274/274 [02:19<00:00,  1.97it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.29it/s]
                   all        420       1797      0.856      0.841      0.875      0.684
...
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      11/20      7.37G      0.735     0.4316       1.07         53        640: 100% 274/274 [02:19<00:00,  1.96it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.14it/s]
                   all        420       1797      0.852      0.836      0.878      0.686

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      12/20      7.42G     0.7125      0.411      1.047         52        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.15it/s]
                   all        420       1797      0.873       0.83       0.87      0.685

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      13/20      7.38G      0.693     0.3847      1.032         61        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.18it/s]
                   all        420       1797      0.928      0.816      0.893      0.703

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      14/20      7.38G     0.6617     0.3657      1.014         37        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.11it/s]
                   all        420       1797      0.885      0.845      0.895      0.715

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      15/20      7.35G     0.6345     0.3418     0.9934         45        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.08it/s]
                   all        420       1797      0.864      0.868      0.873      0.711

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      16/20      7.38G     0.6178     0.3311     0.9828         50        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:05<00:00,  2.34it/s]
                   all        420       1797      0.904      0.842      0.882      0.716

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      17/20      7.34G     0.5948      0.312     0.9715         46        640: 100% 274/274 [02:18<00:00,  1.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.11it/s]
                   all        420       1797      0.907      0.858      0.894       0.73

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      18/20      7.41G     0.5714     0.2943     0.9569         58        640: 100% 274/274 [02:19<00:00,  1.97it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.23it/s]
                   all        420       1797      0.917      0.893      0.903      0.744

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/20       7.4G     0.5508     0.2824     0.9441         60        640: 100% 274/274 [02:19<00:00,  1.97it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.27it/s]
                   all        420       1797      0.898      0.895      0.904      0.744

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/20      7.42G     0.5322     0.2684     0.9351         44        640: 100% 274/274 [02:18<00:00,  1.97it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 14/14 [00:06<00:00,  2.13it/s]
                   all        420       1797      0.924      0.908      0.919      0.761

20 epochs completed in 0.850 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/train/weights/best.pt, 52.0MB
```

Figure 6

Fig7 Number of epochs and their corresponding mAP, instances, GPU memory, box loss, cls loss and dfl loss for trained images

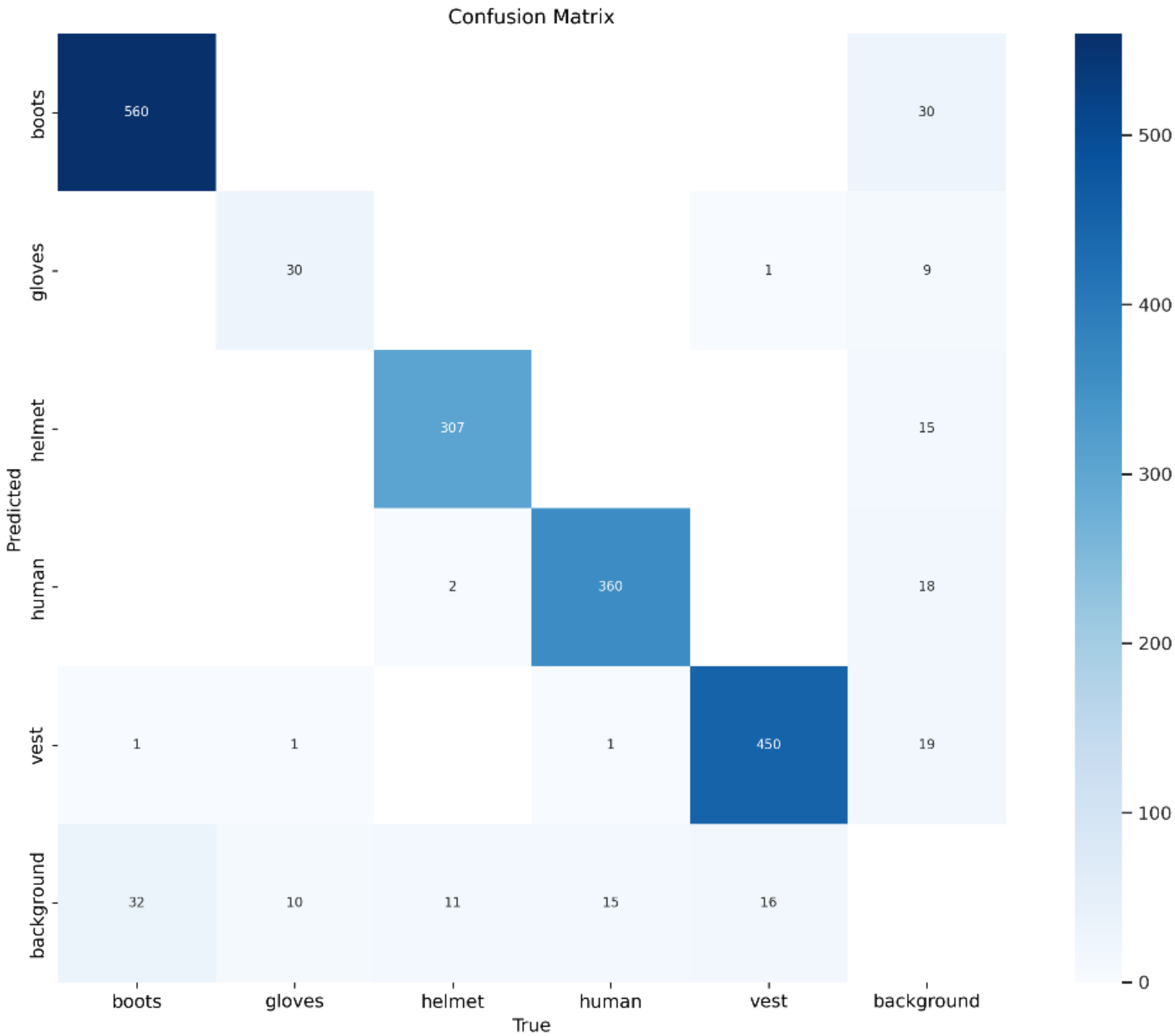

Figure 7

Fig8 Confusion matrix of the trained model

Figure 8

Fig 9 Frame 103/322 of construction Workers at a Worksite video from the results achieved after testing the model

Figure 9

Fig 10 Comparison between day and night lighting obtained from the trained model

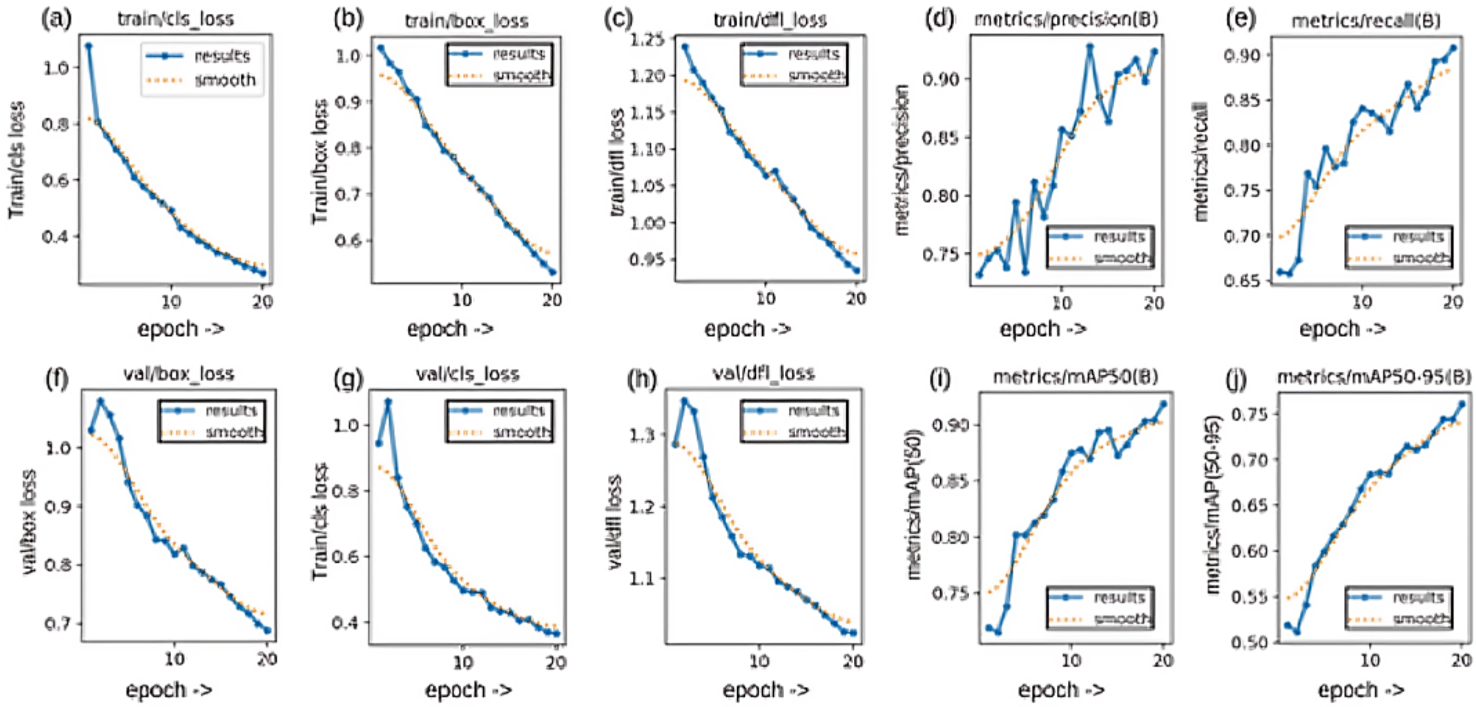

Figure 10

Fig 11 Training and Validation Metrics over 20 Epochs for YoloV8.