sustainability

Article

Construction Site Hazards Identification Using Deep Learning and Computer Vision

Muneerah M. Alateeq [1], Fathimathul Rajeena P. P. [1,*] and Mona A. S. Ali [1,2,*]

[1] Computer Science Department, College of Computer Science and Information Technology, King Faisal University, Al Ahsa 36291, Saudi Arabia
[2] Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt
* Correspondence: fatimah.rajeena@kfu.edu.sa (F.R.P.P.); m.ali@kfu.edu.sa (M.A.S.A.)

Abstract: Workers on construction sites face numerous health and safety risks. Authorities have made numerous attempts to enhance safety management; yet incidents continue to occur, impacting both worker health and the project's forward momentum. To that end, developing strategies to improve construction site safety management is crucial. The goal of this project is to employ computer vision and deep learning methods to create a model that can recognize construction workers, their PPE and the surrounding heavy equipment from CCTV footage. Then, the hazards can be discovered and identified based on an analysis of the imagery data and other criteria including weather conditions, and the on-site safety officer can be contacted. Our own dataset was used to train the You Only Look Once model, version 5 (YOLO-v5), which was put to use as an object detection model. The detection model's performance in tests showed promise for fast and accurate object recognition in the field.

Keywords: object detection; PPE; heavy equipment; YOLO-5

check for updates

Citation: Alateeq, M.M.; P. P., F.R.; Ali, M.A.S. Construction Site Hazards Identification Using Deep Learning and Computer Vision. *Sustainability* **2023**, *15*, 2358. https://doi.org/10.3390/su15032358

Academic Editors: Wei Tong Chen, Ying-Hua Huang and Han-Hsiang Wang

Received: 17 December 2022
Revised: 17 January 2023
Accepted: 21 January 2023
Published: 28 January 2023

1. Introduction

In the past decade, the worldwide construction sector has experienced explosive growth, and safety management on building sites has become a top priority. However, accidents continue to occur in these locations and are frequently undetected or noticed extremely late. According to the Saudi Arabian Health and Safety Association, commercial and public sector workplace injuries in 2016 totaled over 67,000 and related medical treatment cost over SAR 370 million [1]. Minor and significant injuries on the job have ramifications for the workers, their families and the project's schedule and budget. Therefore, there have been many efforts in recent years to make construction sites safe, productive and smart.

Construction sites are known to be highly hazardous environments due to their dynamic and temporary nature. Some studies showed that accidents at construction sites can be caused by many factors, as follows [2,3]: lack of awareness and experience among workers, lack of safety training among workers, workers not wearing personal protective equipment (PPE), no safety officer located in danger zones and machinery defects and errors. Construction workers are involved in many activities that might expose them to risks and accidents. Some of the hazards on construction sites may involve falling from heights, being electrocuted, being struck by heavy equipment or falling materials, being caught in equipment and many more, depending on the nature of the site [3]. Those accidents can be prevented by following safety policies and rules such as providing appropriate safety training for the workers, monitoring PPE compliance, regularly inspecting machines for defects and errors, identifying danger zones and assign a safety officer to each danger zone [2]. Despite all the efforts that have been conducted by the authorities to reduce accidents and manage safety, it is still a complex task that needs to be accomplished manually.

The safety and wellbeing of employees are essential targets for a successful project [4]. Employees' overall health is crucial for authorities when approving and monitoring a project. On the other hand, accidents can lead to complications or failures related to the employees or the project. In general, construction sites pose a high risk to employees' health and wellbeing, which is mainly related to accidents. The procedures and factors involved in predicting such a risk are quite diverse and complex, to be observed and analyzed manually. In the future, computer vision and deep learning algorithms can be used to analyze, identify and predict the hazards to be avoided, eliminated or reduced.

The study of construction safety algorithms and approaches began decades ago. Regarding the subject of construction safety management, researchers have employed numerous computer vision and machine learning approaches, including YOLO, Fast-RNN and many more. The most significant component of this process is obtaining accurate resultsn a fair amount of time, which pushes researchers to focus their efforts on the development of procedures that will lead to greater accuracy. This section is devoted to providing a concise summary of the most recent attempts and strategies that have been presented in the past few years.

In 2019, Zhang et al. [5] proposed a framework to manage safety on construction sites based on computer vision and a real-time location system. Imagery data collected from on-site cameras were analyzed using Fast R-CNN to detect and classify objects and determine the danger zones. The location of workers could be tracked using Bluetooth Low Energy devices attached to their safety equipment. If a dangerous situation was detected, the workers were notified through a loud sound and vibration by their mobile, which was paired with their Bluetooth devices. This framework was useful for proactive safety management and it was cost efficient. However, due to the fact that construction sites are noisy environments, the warning sound might not be heard. One way to solve this issue is to use a light strip attached to the safety vest instead.

In 2019, Wang et al. [6] proposed a methodology to predict safety hazards on construction sites based on deep learning and computer vision. They used 2410 images from construction site surveillance cameras as a training and testing dataset. These images contained construction workers and five types of heavy equipment. The first step of the methodology was to extract and detect workers and equipment from images using faster R-CNN. Then, the danger zone was specified for the equipment, and the trajectory of the workers was predicted using the DeepSORT framework and Kalman filter. Based on the result of the second step, the spatial–temporal relation between the workers and the equipment was analyzed, and the hazards were predicted. The proposed method obtained a high accuracy, 95%, for detecting the workers and equipment while the accuracy of assigning the safety status to the workers was 87.45%.

In 2019, Zhao et al. [7] proposed a method to detect safety officers and track pedestrians on construction sites based on deep learning. They combined multiple datasets of humans, vests and helmets. YOLO-v3 was used to detect safety officers using their helmets and reflective vest, while the Kalman filter and Hungarian algorithm were used for pedestrian tracking. The precision of detecting pedestrians, helmets and vests was 89%, 84% and 94%, respectively. Moreover, it maintained a high detection speed, close to the real-time requirement, with 18 frames/second.

In 2020, Nipun et al. [8] proposed three models based on YOLO-v3 and machine learning classifiers to check whether workers were wearing their PPE. They used Pictor-v3 as a training dataset for their three models. In this study, the authors focused on detecting hats and vests, but the model could be scaled to detect other PPE, such as gloves and glasses. The first model detected the workers and the PPE first, and then NN and DT were used to check if the workers wore the detected PPE or not. The second model localized the workers and directly classified them based on their PPE into one of four classes: workers not wearing a hard hat or vest (W), workers wearing just a hard hat (WH), workers wearing just a vest (WV), and workers wearing both a hard hat and vest (WHV). In the third model, all the workers were detected, then a CNN classifier was applied to classify the workers into the W, WH, WV and WHV classes. Among these three models, the second one gave the best performance with 72.3% mAP, followed by the third model with 67.93%. On the other hand, the first model gave the fastest performance with 13 FPS.

In 2020, Delhi et al. [9] proposed a framework to check the PPE compliance in real time to ensure the safety of construction workers. The authors trained the model using manually collected images from construction sites and the Internet. The framework was based on CNN and YOLO-v3 deep learning networks and focused on detecting hard hats and safety jackets. It classified the detected workers into four categories: not safe, safe, no hard hat and no jacket. When the detected worker fell into the not safe category, an alarm and time-stamped report were generated. The accuracy of the proposed model was 96.92%, and the average precision was 0.98.

The above methods provided a solution to construction site safety management, but with some limitations. Some systems focused solely on detecting PPE compliance without providing a method to notify safety officers, so countermeasures could be taken. Furthermore, weather conditions play an important role in safety management, particularly in harsh environments such as deserts. Working around heavy equipment in high winds and gusts is extremely dangerous. No method took the weather into account as a risk factor.

The related work put a spotlight on developing an approach based on computer vision and deep learning, by considering how to improve some of the limitations of the approaches mentioned above.

The purpose of this research can be summarized as follows:

1. Building our own dataset: The performance of any deep learning model relies significantly on the quality of the training data. Construction sites are dynamic, so we built a dataset that combines images from existing datasets (Pictor-v3 TTM, Construction-YOLOv5 and ACID) and self-captured images from local construction sites to include different weather conditions.
2. Image preprocessing: Image augmentation techniques are used in this project to produce different scales of the images to enlarge the dataset size. This helps deal with different CCTV cameras positions later in the deployment.
3. Object detection model: In this project, we do not propose a new architecture for YOLO, but, instead, we investigate the latest version of YOLO, version 5, which has not been tested with construction site data.
4. Weather conditions: In construction sites, weather conditions play an important role in the project's progress. Wind speed and gusts are important factors in determining when to stop lifting activities, and temperature determines when to stop construction activities in general. In this project, we connect a weather API to our system to identify hazardous situations.

2. Materials and Methods

The primary goal of this project is to improve hazard identification by combining existing models with data preprocessing techniques to reach high accuracy. The methodology of the proposed project consists of the following main steps, where each step is responsible for a specific task. Figure 1 shows an overview of the proposed system.

1. Imagery data are collected to build the training and validation datasets.
2. YOLO-v5 algorithm is used for object detection.
3. The model is trained to recognize PPE and heavy equipment using our datasets.
4. Weather conditions obtained from the API is used to predict the wind speed and temperature hazards.
5. Hazards are identified based on the status of the workers (if they are wearing the appropriate PPE or not), the type of equipment around them and the weather conditions.
6. When a hazard is identified, the safety officer is notified to prevent the accident.

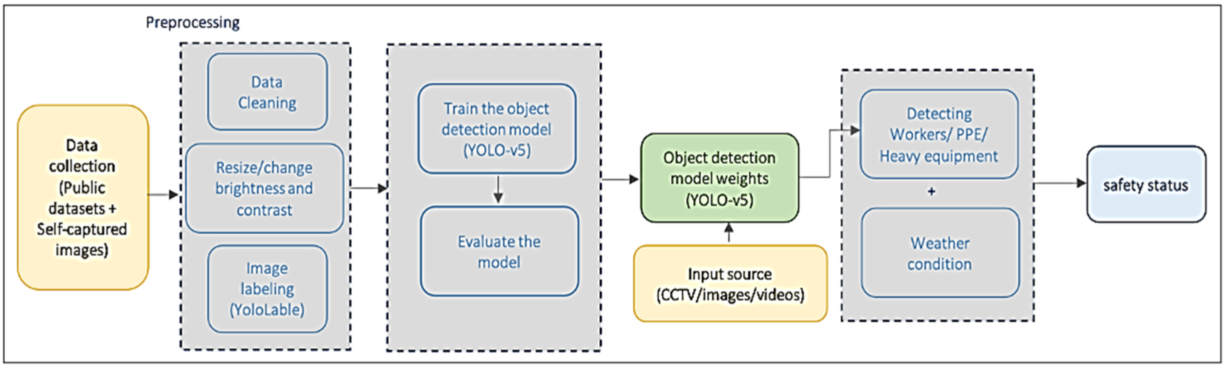

Preprocessing
Data collection (Public datasets + Self-captured images)
Data Cleaning
Resize/change brightness and contrast
Image labeling (YoloLable)
Train the object detection model (YOLO-v5)
Evaluate the model
Object detection model weights (YOLO-v5)
Input source (CCTV/images/videos)
Detecting Workers/ PPE/ Heavy equipment
+
Weather condition
safety status

Figure 1. Proposed framework of the system.

2.1. Dataset of the Study

Training an object detector is a supervised learning problem. For that, we need to specify a dataset to train our model. The following describes the process of building the dataset.

2.2. Data Collection

The choice of the dataset is an important factor in the accuracy and reliability of the model. To conduct our experiment, two datasets were used, one for detecting workers and PPE and the other for detecting heavy equipment. Worker/PPE dataset images were collected from Pictor-v3 dataset (from [10]) and self-captured images. Figure 2 shows sample of Pictor-v3 dataset images.

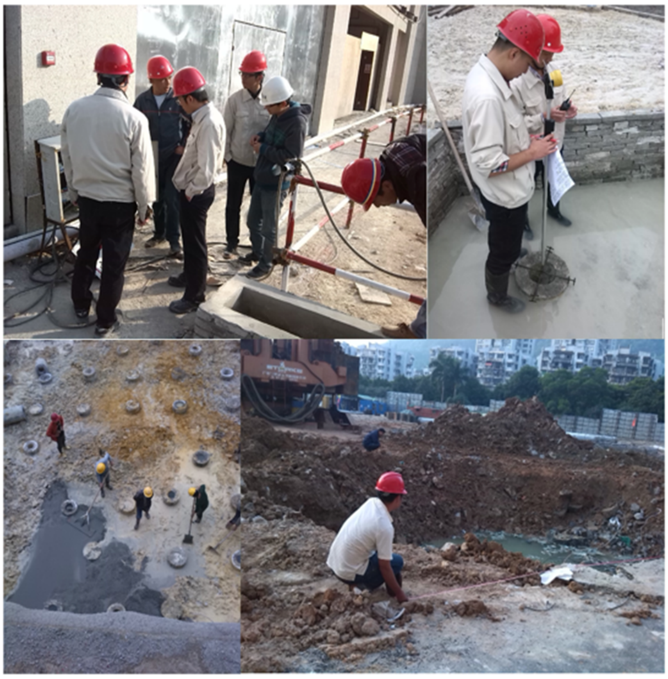


Figure 2. Samples of images of Pictor-v3 dataset.

Our collected dataset contains 826 images with 5241 instances of three categories: workers, safety helmets and reflective vests. Table 1 shows the number of cases across different classes. Figure 3 shows an example of self-captured images.

Table 1. Number of instances across the worker/PPE dataset classes.

Class	No. of Annotations
Worker	2784
Safety helmet	2027
Reflective vest	430

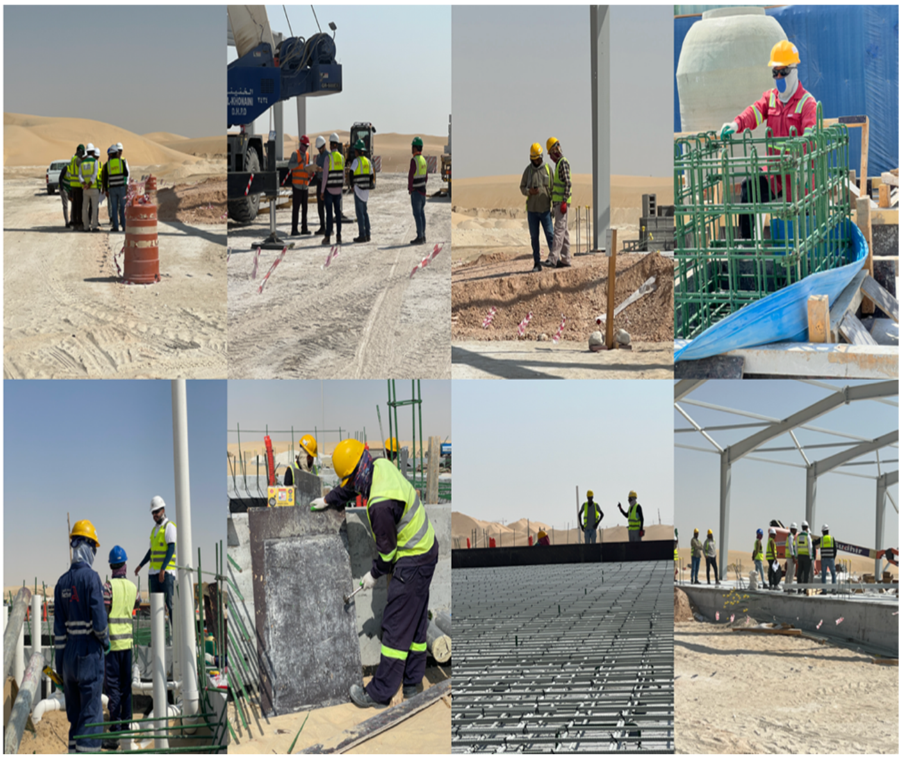


Figure 3. Examples from the self-captured worker/PPE dataset.

The heavy equipment dataset images were collected from publicly available datasets: ACID [11], TTM [12] and Construction-YOLOv5 [13]. Each dataset contains different classes of heavy equipment, and seven classes of the most commonly used heavy equipment were chosen. Figure 4 shows samples of images of the datasets.

In addition to these datasets, self-captured images from local construction sites were collected. Our dataset contains 6338 photos with 9701 cases of seven classes of the most commonly used heavy equipment in construction sites: bulldozer, dump truck, excavator, grader, loader, mobile crane and roller. Figure 5 shows example of self-captured images. Table 2 shows the number of instances across the seven classes.

Table 2. Number of instances across the heavy equipment dataset classes.

Class	No. of Annotations
Bulldozer	1339
Dump truck	2626
Excavator	2046
Grader	1145
Loader	641
Mobile crane	766
Roller	1138

Figure 4. Samples of images of heavy equipment datasets.

Figure 5. Example of the self-captured heavy equipment dataset.

2.3. Data Cleaning

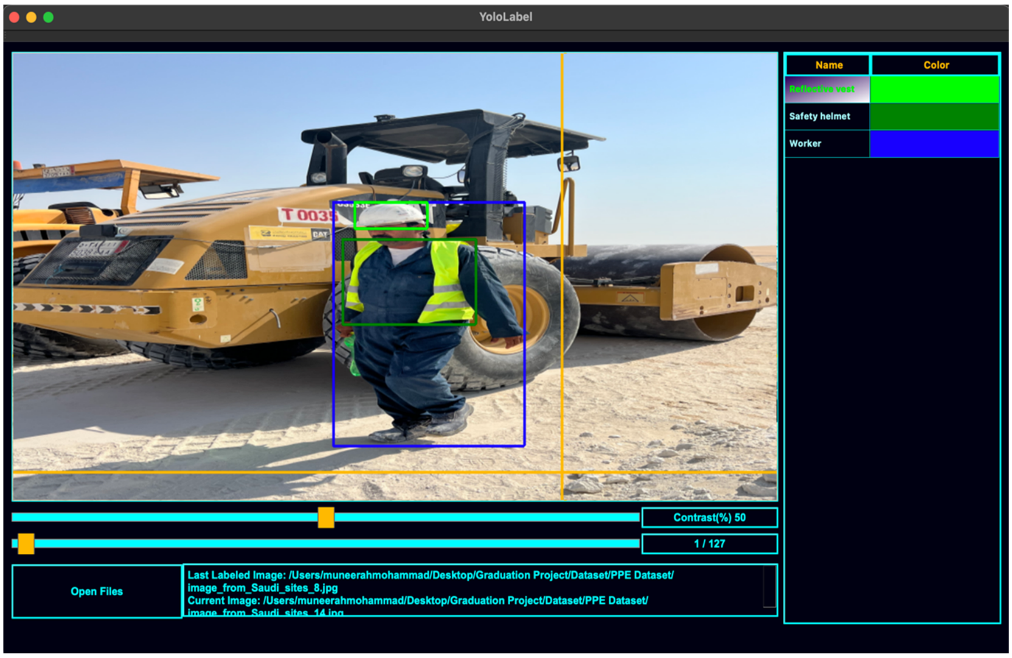
After data collection, the data were cleaned to remove invalid data. This step involved the elimination of duplicate images and images that violate the privacy of the construction company. The two datasets were cleaned manually by removing duplicate and low-quality photos.

2.4. Image Preprocessing

Before providing the object detection model with images as inputs, these input images have to be preprocessed. All the photos in the dataset were resized to the shape 416 × 416. Another preprocessing step in the dataset was changing the brightness and contrast of some images to enhance them.

2.5. Image Labeling

Different object detectors have other labeling formats. YOLO family uses two file extensions for labeling objects: the .jpeg image file and the .txt text file. The image file is just a simple image file, while the text file is used to store the labels, the types of objects present in the image and the coordinates of their bounding boxes. The number of rows within the text file indicates the number of objects within the image. Many labeling tools can be used to label the objects, such as YOLO_mark, BBox-Label-Tool, labelImg, etc. In this project, images were labeled using the YOLOLabel tool, which is simple and provides a good GUI. This tool is publicly available in [14]. To locate the objects we wanted to detect, we drew the bounding box and chose the class from the list of predefined classes. Figure 6 shows the YoloLabel interface and an example of the image labeling process.



Name	Color
Reflective vest	🟩
Safety helmet	🟩
Worker	🟦

Last Labeled Image: /Users/muneerahmohammad/Desktop/Graduation Project/Dataset/PPE Dataset/
image_from_Saudi_sites_8.jpg
Current Image: /Users/muneerahmohammad/Desktop/Graduation Project/Dataset/PPE Dataset/
image_from_Saudi_sites_14.jpg

Figure 6. YOLO label interface.

2.6. Splitting Data

In our experiments, the entire dataset was randomly split into 70% training (578 images of workers and PPE instances and 4400 images of heavy equipment instances), 20% validation (165 images of workers and PPE instances and 1300 images of heavy equipment instances) and 10% test (83 images of workers and PPE instances and 636 images of heavy equipment instances).

2.7. Object Detection Algorithms

The YOLO algorithm was first introduced by Redmon et al. in 2015 [15], which was followed by many different versions. YOLO implements a single-stage object detector. In general, YOLO algorithms divide the image into s×s grid, and, if the object center is within one of the grids, this grid will detect the object [16].

YOLO-v5, proposed by Glenn Jocher et al. in 2020, is the updated version of the YOLO family. It is more flexible, faster and more accessible than the previous versions; however, it is slightly less accurate than YOLO-v4 [16]. The advantages of being faster and easier outweigh the difference in accuracy, making it an attractive choice to be used in object detection for our project.

The architecture of YOLO-v5, like any object detector, is composed of three main parts: backbone, neck and head. YOLO-v5 uses Cross Stage Partial Network (CSPNet) as a model backbone [17], which helps to extract the essential features from a given image. For the model neck, YOLO-v5 uses Path Aggregation Network (PANet) [18] to generate the feature pyramids, which helps to generalize unseen data. The final detection is performed by the model head, which applies the anchor boxes in the features and produces the final vector along with the bounding boxes and class probabilities [19]. Figure 7 shows the network architecture of YOLO-v5.

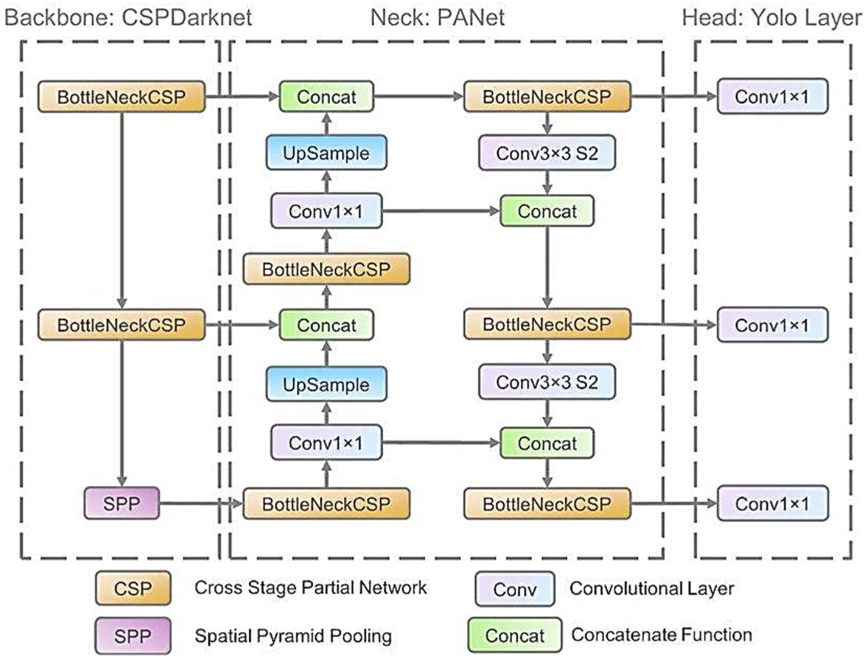


Figure 7. Architecture of YOLO-v5 [20].

2.8. Testing and Evaluation

To test the proposed model, imagery data collected from a local construction site using CCTV cameras were used. Then, the accuracy of the proposed methodology was measured using Intersection over Union (IoU) and a confusion matrix. IoU is a metric that calculates the intersection area between the actual bounding box and the predicted bounding box to check whether the detected object is valid, as shown in Equation (1). The value of IoU ranges from 0 to 1, where 0 indicates no overlap, and 1 indicates perfect overlap [21]. The confusion matrix consists of true prediction indicated by true positive TP and true negative TN, while false prediction is indicated by false positive FP and false negative FN. Many metrics can be driven from the confusion matrix, such as precision, recall and mAP. The precision of the model measures whether the model is reliable or not. It tells us how many TP observations the model could detect out of all the positive observations; see Equation (2). To measure the sensitivity of a model, we use recall. Recall tells us how many

TP observations the model detected correctly; see Equation (3). The mean average precision (mAP) is used to measure the accuracy of the object detection model across all classes in a given dataset, as shown in Equation (4), where AP is the average of all precisions (see Equation (5)), and n is the total number of classes [22].

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of intersection}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detection}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \tag{4}$$

$$\text{AP} = \sum_{k=0}^{k=n-1} [\text{Recalls}(k) - \text{Recalls}(K+1)] * \text{Precisions}(k) \tag{5}$$

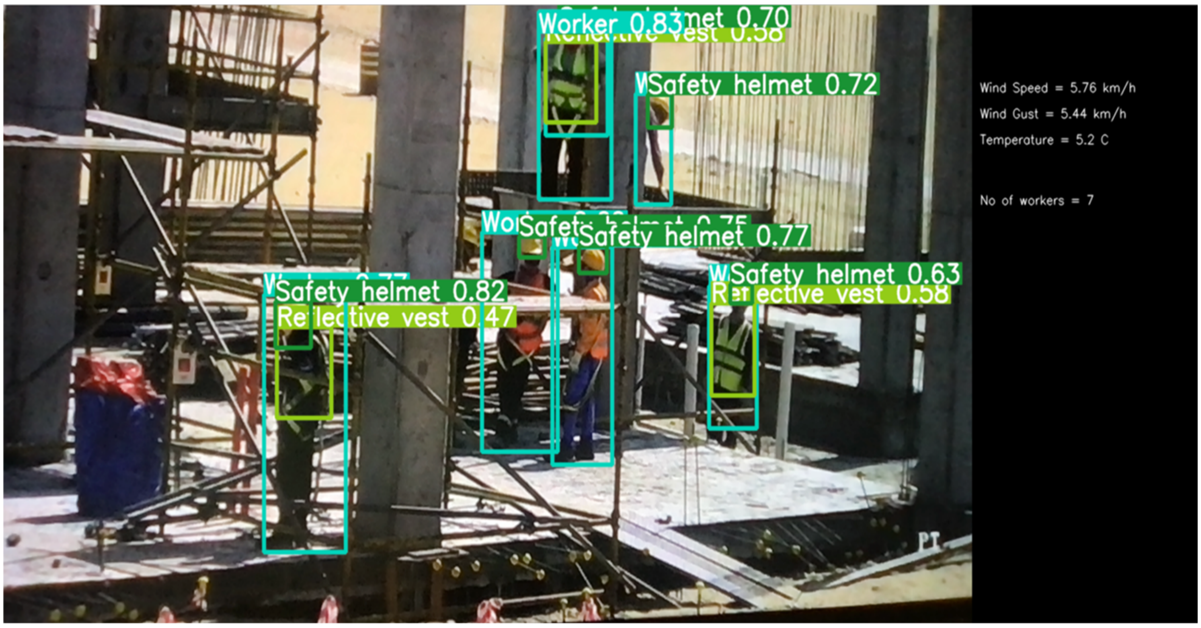
2.9. Experiment Setup

Python programming language was used to conduct this experiment on Google ColaboratoryPro (Google Colab Pro) on Mac operating system using the online cloud service with graphics processing unit (GPU) hardware. YOLO-v5 model source code was taken from the original author of the model [23–32], and then it was altered based on our needs. The model was pre-trained on COCO dataset weights.

2.10. System GUI

The GUI of the system was built using the TKinter package, which is the most commonly used package for GUI programming in Python. It provides a fast and easy object-oriented interface [33].

The GUI of the system includes a video frame to show the detection results, weather data and the number of workers detected by the model, as shown in Figure 8.



Wind Speed = 5.76 km/h
Wind Gust = 5.44 km/h
Temperature = 5.2 C

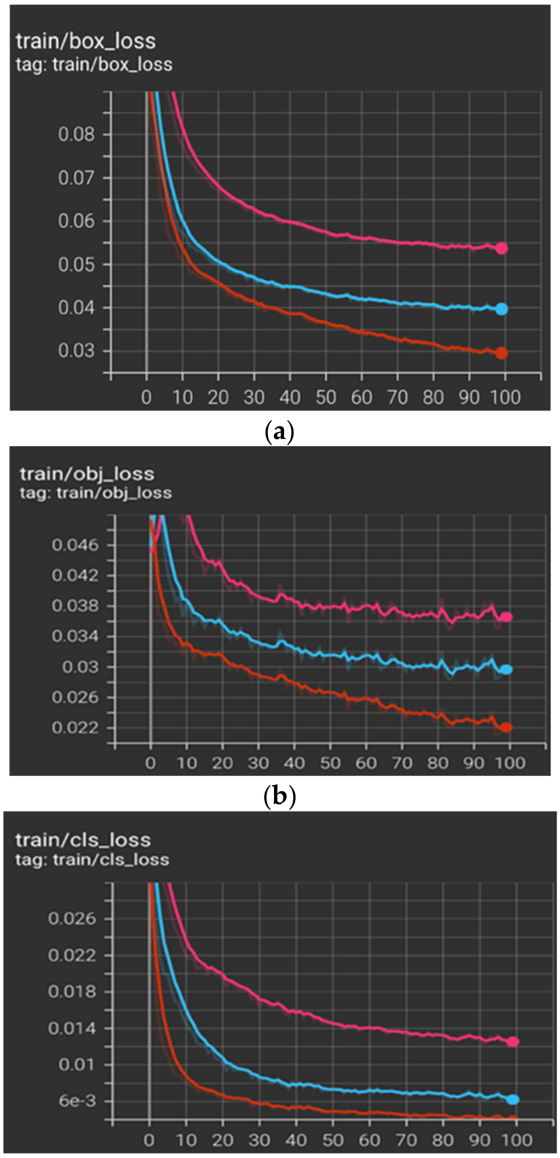
No of workers = 7

Figure 8. GUI of the system.

In this project, OpenWeatherMap API was used to obtain live weather data. In Python, the request library as first used to obtain the API response. Then, the response was converted into json format, from which the temperature, wind speed and wind gust values were used. The weather data of a specific area was obtained by putting that location's coordinates (longitude and latitude) in the API request URL.

3. Results

The main objective of this experiment is to detect objects with high accuracy and in real-time. In this study, our datasets were trained with the YOLO-v5 model. In the beginning, three different learning rates were experimented with to examine which gave the lowest loss. We started with 0.01, the default value in the YOLO-v5 model, and then exponentially lowered the values to 0.001 and 0.0001. Based on the minimum loss, we found that the model was performing better with a learning rate = 0.01, as shown in red in Figure 9, rather than a learning rate of 0.001 or 0.0001, as shown in blue and pink in Figure 9, respectively.



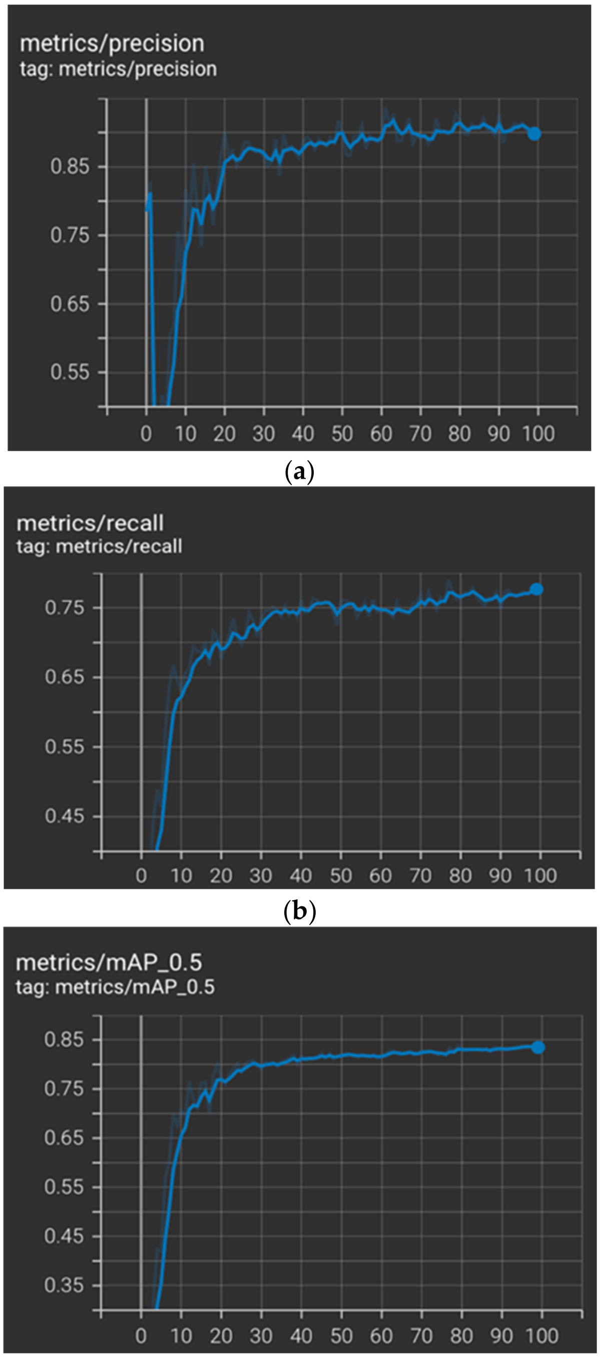
(**a**)

(**b**)

(**c**)

Figure 9. Results of the training loss with different learning rates: (**a**) box loss, (**b**) object loss and (**c**) class loss.

YOLO-v5 was trained with two different datasets, worker/PPE training datasets and heavy equipment training datasets. For the worker/PPE dataset, the model was trained for 100 epochs, and the batch size was 16. The training time took approximately 15 minutes on a Google Colab GPU. Figure 10 shows the model's performance with the worker/PPE datasets in terms of precision, recall and mAP at the 50 IoU threshold.



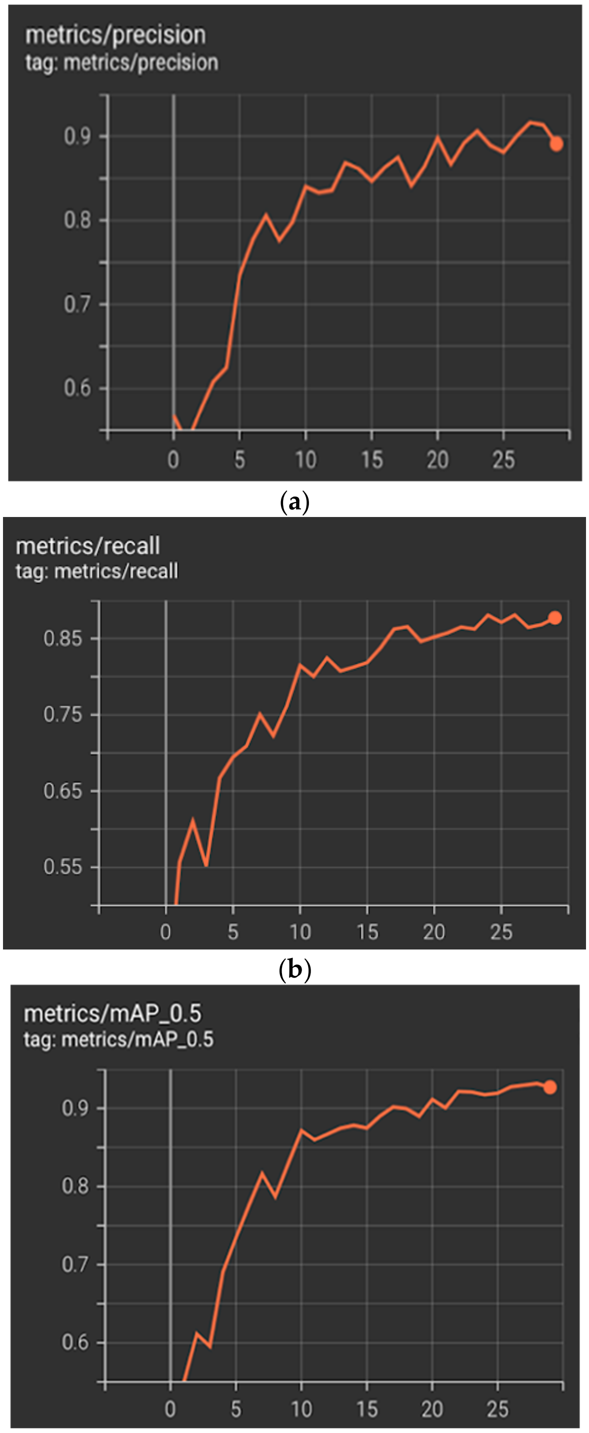
(**a**)

(**b**)

(**c**)

Figure 10. Performance of YOLO-v5 with worker/PPE training phase over epochs: (**a**) precision, (**b**) recall and (**c**) mAP at the 50 IoU threshold.

The model was trained for 30 epochs for the heavy equipment dataset, and the batch size was 16. The training time took approximately 18 minutes on a Google Colab GPU. Figure 11 shows the performance of the model with the heavy equipment dataset in terms of precision, recall and mAP at the 50 IoU threshold.



(**a**)

(**b**)

(**c**)

Figure 11. Performance of YOLO-v5 with heavy equipment training phase over epochs: (**a**) precision, (**b**) recall and (**c**) mAP at the 50 IoU threshold.

The performance of YOLO-v5 on the validation worker/PPE dataset is summarized in Table 3. The overall precision was approximately 90%, the recall was 77%, and the mAP at the 50 IoU threshold was 83%.

Table 3. Validation results on the worker/PPE dataset.

	Evaluation Metrics		
Class	**Precision**	**Recall**	**mAP@50**
All	0.909	0.771	0.837
Worker	0.915	0.781	0.859
Safety helmet	0.886	0.712	0.782
Reflective vest	0.927	0.819	0.872

After training and validating the model, we tested the model with our worker/PPE test dataset. The performance of YOLO-v5 on the testing dataset is summarized in Table 4. The overall precision was approximately 90%, the recall was 76%, and the mAP at the 50 IoU threshold was 83%. The preprocess speed was 0.5 ms, and the inference speed was 5.1 ms per image. Figure 12 shows some examples of the actual labels of the testing dataset vs. the predicted labels for the model shown in Figure 13.

Table 4. Testing results on worker/PPE dataset.

	Evaluation Metrics		
Class	**Precision**	**Recall**	**mAP@50**
All	0.906	0.767	0.831
Worker	0.893	0.772	0.851
Safety helmet	0.895	0.638	0.71
Reflective vest	0.93	0.891	0.932

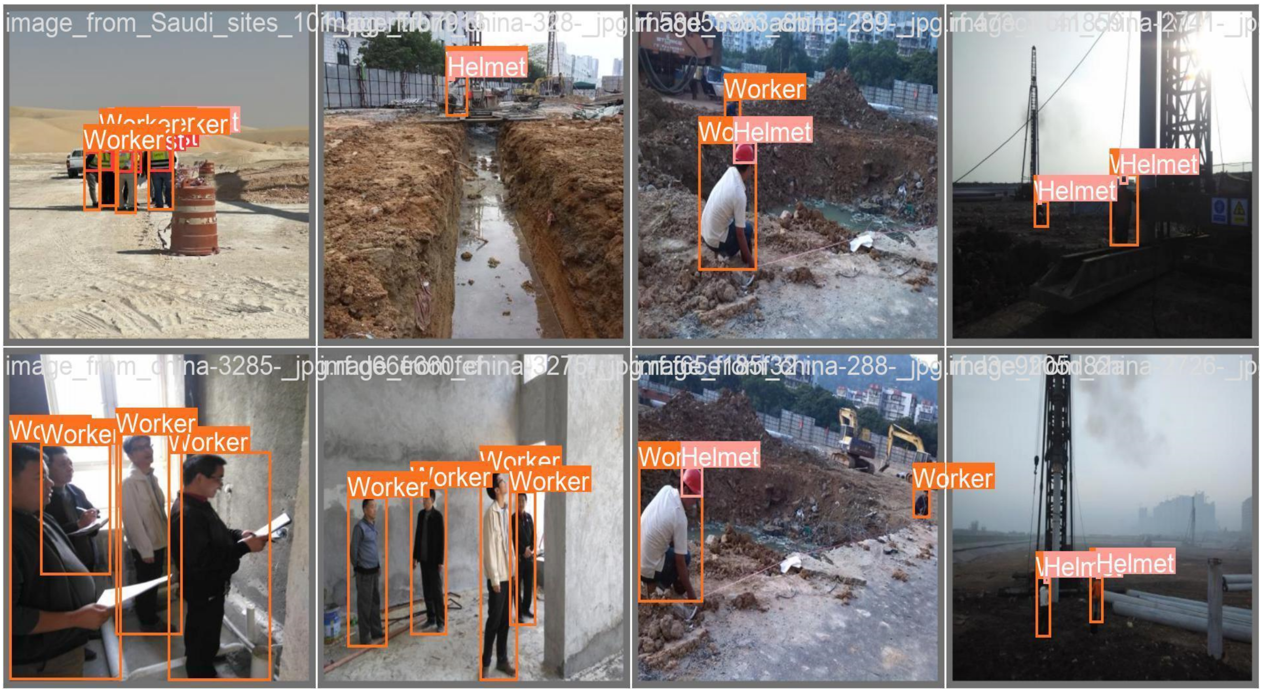


Figure 12. Actual labels of the worker/PPE testing datasets.

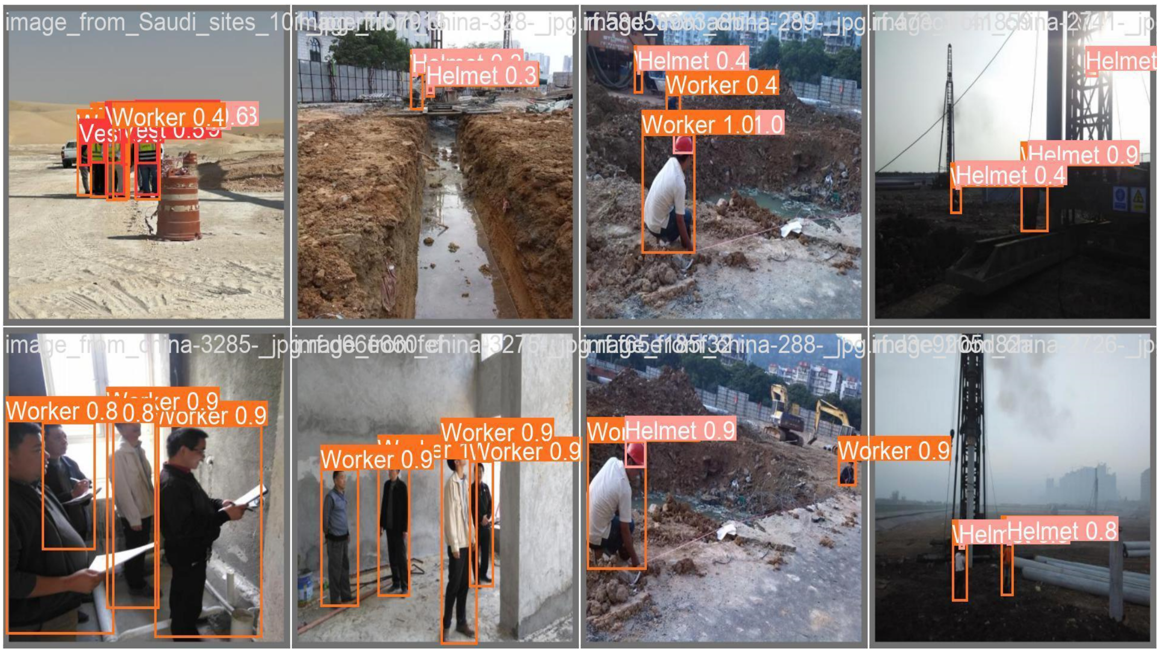


Figure 13. Predicted labels of the worker/PPE testing datasets.

From the validation and testing results, we can see that the model could detect the objects with high performance. However, the lowest performance was with the safety helmet class, which is considered a small object to be detected compared to the other classes in the dataset.

On the other hand, the performance of YOLO-v5 on the heavy equipment validation dataset is summarized in Table 5. The overall precision was approximately 91%, the recall was 86%, and the mAP at the 50 IoU threshold was 93%.

Table 5. Validation results on heavy equipment dataset.

	Evaluation Metrics		
Class	**Precision**	**Recall**	**mAP@50**
All	0.914	0.868	0.932
Bulldozer	0.963	0.958	0.98
Dump truck	0.844	0.819	0.887
Excavator	0.889	0.917	0.948
Grader	0.971	0.95	0.98
Loader	0.832	0.823	0.878
Mobile crane	0.922	0.647	0.852
Roller	0.976	0.965	0.992

After training and validating the model, the model was tested with our heavy equipment test dataset. The performance of YOLO-v5 on the testing dataset is summarized in Table 6. The overall precision was approximately 87%, the recall was 88%, and the mAP at the 50 IoU threshold was 92%. The preprocess speed was 0.2 ms, and the inference speed was 1.5 ms per image. Figure 14 shows some examples of the actual labels of the testing dataset vs. the predicted labels for the model shown in Figure 15.

Table 6. Testing results on heavy equipment dataset.

	Evaluation Metrics		
Class	**Precision**	**Recall**	**mAP@50**
All	0.879	0.882	0.92
Bulldozer	0.94	0.971	0.985
Dump truck	0.793	0.834	0.864
Excavator	0.886	0.935	0.952
Grader	0.991	0.955	0.992
Loader	0.749	0.824	0.871
Mobile crane	0.808	0.681	0.791
Roller	0.987	0.971	0.981

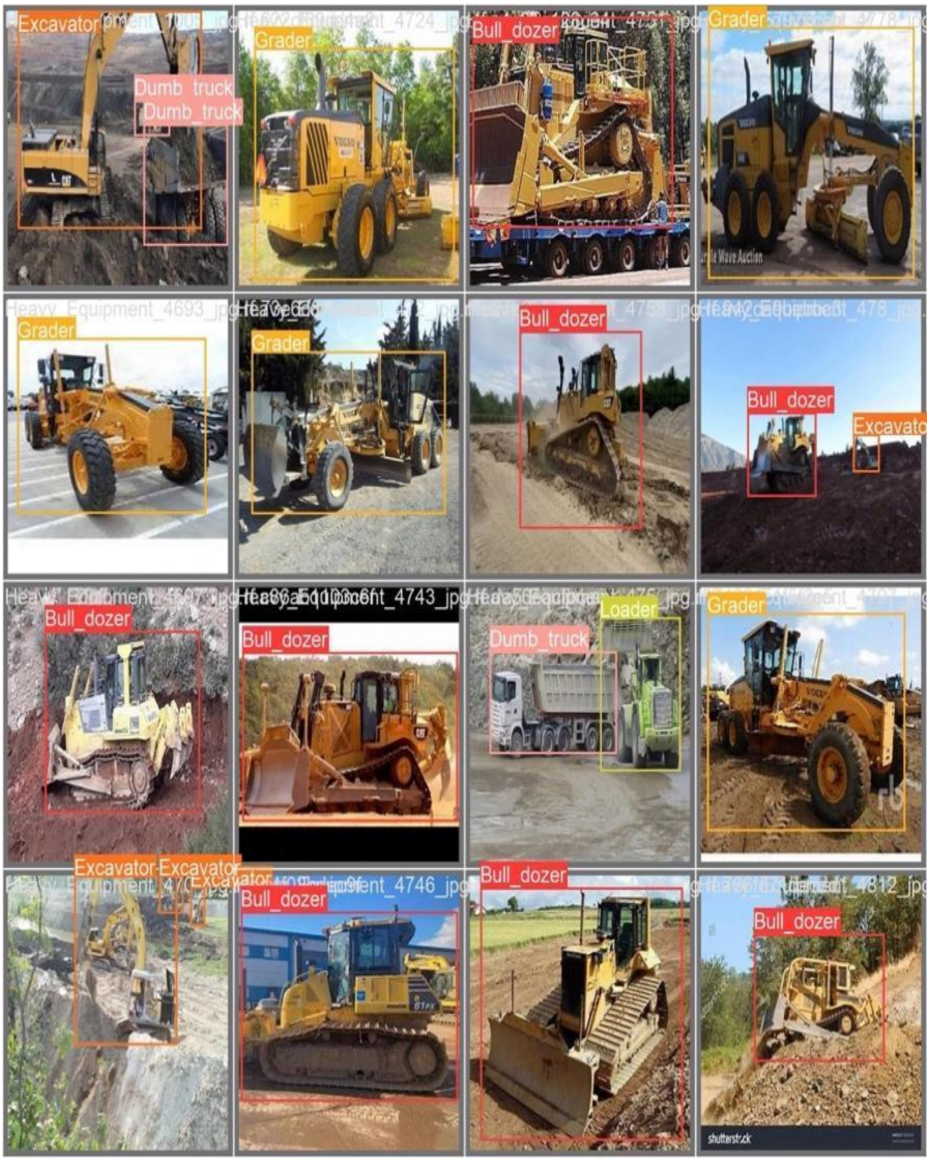


Figure 14. Actual labels of the heavy equipment testing datasets.

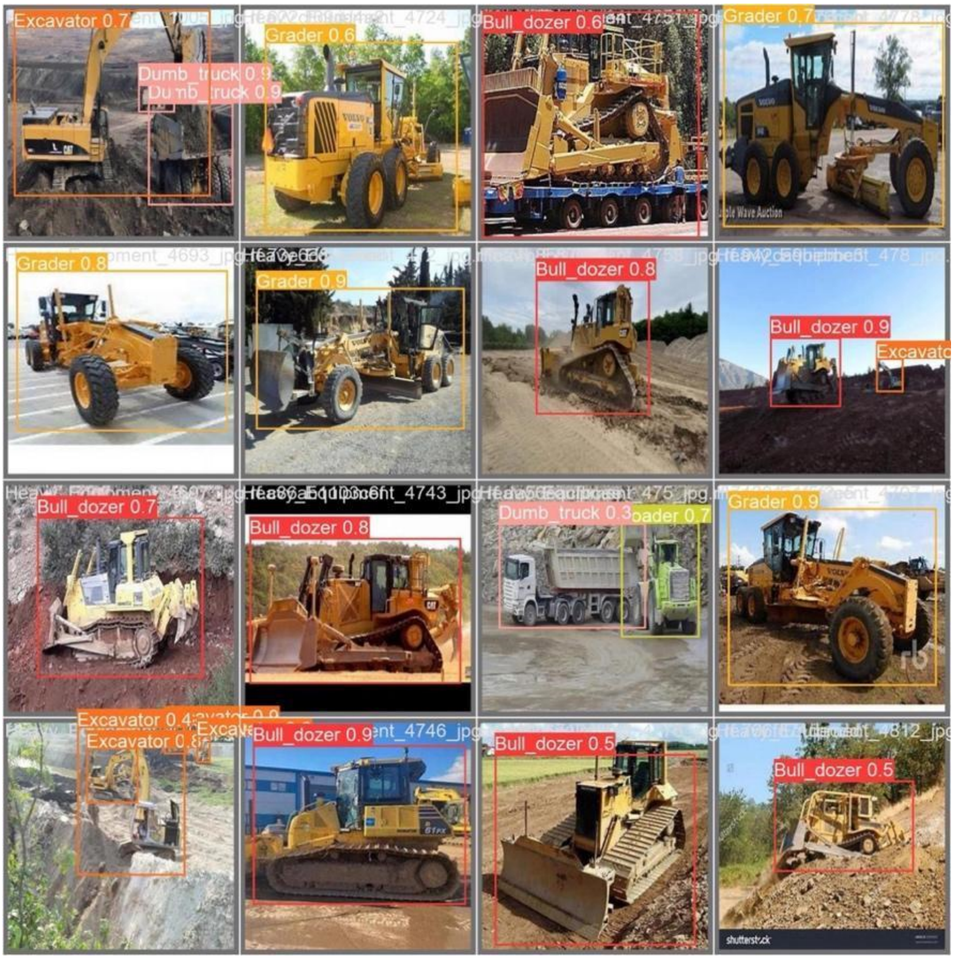


Figure 15. Predicted labels of the heavy equipment testing datasets.

In addition, our model was tested with videos from local construction sites. The model was able to detect the objects with a high inference speed, which was 141 FPS. Figures 16 and 17 show an example of the results of the detection.

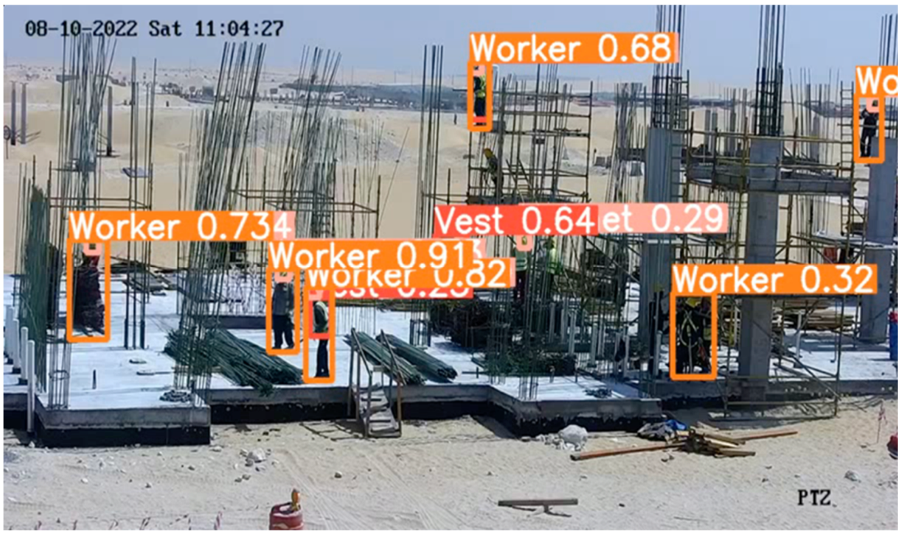


Figure 16. Result of testing the model in Video (1).

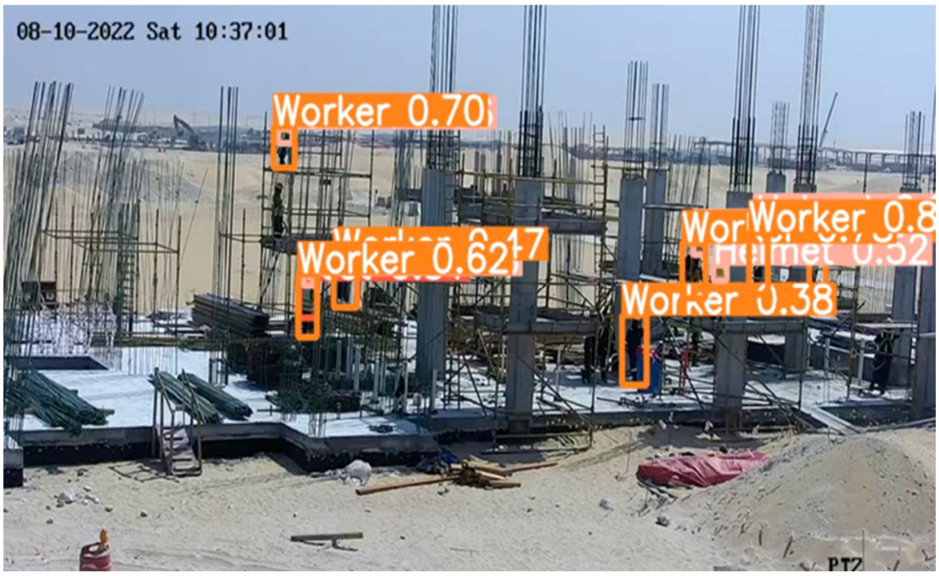


Figure 17. Result of testing the model in Video (2).

4. Discussion

The model gave promising results for detecting workers, PPE and heavy equipment on the construction site, accurately and in real-time. The model was able to detect partial objects in the images and video frames. The inference speed of the YOLO-v5 model is considered very high compared with the other versions of YOLO in the literature. The inference speed reached 141 FPS with YOLO-v5 and 13 FPS with YOLO-v3 in the Pictor-v3 dataset. In addition, the mAP of YOLO-v5 in the worker/PPE reached 83%, while the highest mAP with YOLO-v3 was 72%. Moreover, we can notice that YOLO-v5 can handle imbalanced classes very well. Underrepresented classes such as safety vests in the worker/PPE dataset and loader and crane in the heavy equipment dataset were detected by the model, with high precision, recall and mAP in the testing dataset. However, not all objects can be detected by the model. For example, the model did not detect some workers behind rebars and objects far from the camera. Benjumea et al. [34] proposed an enhancement of the YOLO-v5 model to overcome this limitation, by altering the structural elements of the model. This improvement can be used in future studies to enhance construction site management.

Table 7 shows a general comparison between our proposed YOLO-v5 model and the related work. We can see that YOLO-v5 gave a competitive performance in terms of mAP. It also outperformed all the methods in the literature in terms of inference speed (141 FPS), which would help in real-time detection.

Table 7. General comparison between the related work and YOLO-v5.

Paper Reference	PPE	Heavy Equipment	Methodology	mAP (%)	Inference Speed
[5]	Yes	Yes	Fast-RCNN	Not provided	Not provided
[6]	Yes	Yes	Faster-RCNN	92.55	Not provided
[7]	Yes	No	YOLO-v3	89.00	18 FPS
[8]	Yes	No	YOLO-v3	72.30	13 FPS
[9]	Yes	No	CNN, YOLO-v3	96.00	2 FPS
Proposed method	Yes	Yes	YOLO-v5	PPE 83.00 Heavy equipment 93.00	141 FPS

5. Conclusions

As construction sites are highly hazardous environments, many efforts have been made to improve safety management. Computer vision and deep learning algorithms have made safety management at these sites more competent and efficient. In this study, we implemented the You Only Look Once model, version 5 (YOLO-v5), which was put to use as an object detection model for our dataset to detect workers, personal protective equipment (PPE) and heavy equipment. In addition, weather conditions were considered when designing the system GUI, due to their importance in detecting hazards, especially in extreme weather conditions such as those in Saudi Arabia. The model's results were promising to detect workers, PPE and heavy equipment on the construction site in real time and with high precision. Moreover, we noticed that the model gave the lowest performance in detecting small objects such as the safety helmets worn by workers. This project may be the first step toward smarter and safer construction sites.

The future work includes enhancements of the proposed system to improve its small object detection. These enhancements will consist of using more data in the training datasets. Another improvement can be to determine the safety status of workers based on the detected PPE and heavy equipment and automatically notify safety officers to prevent accidents. Moreover, spatial–temporal analysis can be added to predict hazardous situations before they happen.

Author Contributions: Formal analysis, M.A.S.A.; funding acquisition, M.A.S.A.; investigation, F.R.P.P.; methodology, M.M.A. and M.A.S.A.; resources, M.M.A.; supervision, M.A.S.A.; writing—original draft, M.M.A., F.R.P.P. and M.A.S.A.; writing—review and editing, M.M.A., F.R.P.P. and M.A.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Deanship of Scientific Research, King Faisal University, Saudi Arabia, with grant number GRANT2457.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not available.

Acknowledgments: We deeply acknowledge the Deanship of Scientific Research, King Faisal University, Saudi Arabia, as they funded our project with grant number GRANT2457.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Arab News. Work Injuries Cost over sr370 Million in Saudi Arabia. *Arab News*, 17 May 2017. Available online: https://www.arabnews.com/node/1100966/saudi-arabia(accessed on 12 December 2021).
2. Moosa, M.M.; Oriet, L.P.; Khamaj, A.M. Measuring the Causes of Saudi Arabian Construction Accidents: Management and Concerns. *Int. J. Occup. Saf. Health* **2020**, *10*, 108-14. [CrossRef]
3. Abukhashabah, E.; Summan, A.; Balkhyour, M. Causes of occupational accidents and injuries in construction industry in Jeddah City. *JKAU Met. Environ. Arid Land Agric. Sci.* **2019**, *28*, 105–116. [CrossRef]
4. Park, C.; Doyeop, L.; Numan, K. An analysis on safety risk judgment patterns towards computer vision based construction safety management. In Proceedings of the Creative Construction e-Conference 2020, Opatija, Croatia, 28 June–1 July 2020; pp. 31–38.
5. Zhang, J.; Zhang, D.; Liu, X.; Liu, R.; Zhong, G. A framework of on-site construction safety management using computer vision and real-time location system. In Proceedings of the International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving Data-Informed Decision-Making, Cambridge, UK, 8–10 July 2019; pp. 327–333.
6. Wang, M.; Wong, P.; Luo, H.; Kumar, S.; Delhi, V.; Cheng, J. Predicting safety hazards among construction workers and equipment using computer vision and deep learning techniques. In Proceedings of the International Symposium on Automation and Robotics in Construction, Banff, AC, Canada, 21–24 May 2019; Volume 36, pp. 399–406.
7. Zhao, Y.; Chen, Q.; Cao, W.; Yang, J.; Xiong, J.; Gui, G. Deep Learning for Risk Detection and Trajectory Tracking at Construction Sites. *IEEE Access* **2019**, *7*, 30905–30912. [CrossRef]
8. Nath, N.D.; Behzadan, A.H.; Paal, S.G. Deep learning for site safety: Real-time detection of personal protective equipment. *Autom. Constr.* **2020**, *112*, 103085. [CrossRef]
9. Delhi, V.S.K.; Sankarlal, R.; Thomas, A. Detection of Personal Protective Equipment (PPE) Compliance on Construction Site Using Computer Vision Based Deep Learning Techniques. *Front. Built Environ.* **2020**, *6*, 136. [CrossRef]

10. "Pictor-PPE" Google Drive. Available online: https://drive.google.com/drive/folders/19uUR6EJPQzMeK0YpsxRm51wMZzDmcsv6 (accessed on 15 November 2022).
11. "ACID7000" Roboflow Universe. Roboflow. Available online: https://universe.roboflow.com/imsmile2000-naver-com/acid7000 (accessed on 15 November 2022).
12. "Dataset TTM Dataset Dataset" Roboflow Universe. Roboflow. Available online: https://universe.roboflow.com/object-nfasp/ttm (accessed on 15 November 2022).
13. "Construction YOLOv5 Dataset" Roboflow Universe. Roboflow. Available online: https://universe.roboflow.com/shinyj1385-gmail-com/construction-yolov5 (accessed on 15 November 2022).
14. "YOLO_LABEL" GitHub. Available online: https://github.com/developer0hye/Yolo_Label (accessed on 15 November 2022).
15. Redmon, J.; Santosh, D.; Ross, G.; Ali, F. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
17. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
18. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
19. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788. [CrossRef]
20. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
21. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
22. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
23. "Python" Python.org. Available online: https://www.python.org/ (accessed on 15 November 2022).
24. "PYTORCH Documentation" PyTorch documentation-PyTorch 1.13 Documentation. Available online: https://pytorch.org/docs/stable/index.html (accessed on 15 November 2022).
25. "Scikit Learn" Scikit. Available online: https://scikit-learn.org/stable/ (accessed on 15 November 2022).
26. "Tensorboard" TensorFlow. Available online: https://www.tensorflow.org/tensorboard (accessed on 15 November 2022).
27. Google Colab. Available online: https://research.google.com/colaboratory/faq.html (accessed on 15 November 2022).
28. OpenWeatherMap.org. "Current Weather and Forecast" OpenWeatherMap. Available online: https://openweathermap.org/ (accessed on 15 November 2022).
29. Sharma, V.; Mir, R.N. A comprehensive and systematic look up into deep learning based object detection techniques: A review. *Comput. Sci. Rev.* **2020**, *38*, 100301. [CrossRef]
30. "Ultralytics/yolov5" GitHub. Available online: https://github.com/ultralytics/yolov5 (accessed on 15 November 2022).
31. "Construction-Management" Sign in to Roboflow. Available online: https://app.roboflow.com/kfu-ye4kz/construction-management/3 (accessed on 15 November 2022).
32. "Construction-Management-Yolo-V5-with-Gui" GitHub. Available online: https://github.com/muneerah1992/Construction-management-YOLO-v5-with-GUI (accessed on 15 November 2022).
33. "Tkinter" Tkinter-Python Interface to Tcl/Tk-Python 3.11.0 Documentation. Available online: https://docs.python.org/3/library/tkinter.html (accessed on 15 November 2022).
34. Benjumea, A.; Teeti, I.; Cuzzolin, F.; Bradley, A. YOLO-Z: Improving Small Object Detection in YOLOv5 for Autonomous Vehicles. *arXiv* **2021**, arXiv:2112.11798.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.