

Article

# A Detailed Comparative Analysis of You Only Look Once-Based Architectures for the Detection of Personal Protective Equipment on Construction Sites

Abdelrahman Elesawy<sup>1</sup>, Eslam Mohammed Abdelkader<sup>1,2,\*</sup> and Hesham Osman<sup>1</sup>

<sup>1</sup> Structural Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt; abdelrahman\_elessawi@hotmail.com (A.E.); hesham.osman@gmail.com (H.O.)

<sup>2</sup> Department of Building and Real Estate, Faculty of Construction and Environment, Hong Kong Polytechnic University, Kowloon 999077, Hong Kong

\* Correspondence: eslam.abdelkader@polyu.edu.hk

**Abstract:** For practitioners and researchers, construction safety is a major concern. The construction industry is among the world's most dangerous industries, with a high number of accidents and fatalities. Workers in the construction industry are still exposed to safety risks even after conducting risk assessments. The use of personal protective equipment (PPE) is essential to help reduce the risks to laborers and engineers on construction sites. Developments in the field of computer vision and data analytics, especially using deep learning algorithms, have the potential to address this challenge in construction. This study developed several models to enhance the safety compliance of construction workers with respect to PPE. Through the utilization of convolutional neural networks (CNNs) and the application of transfer learning principles, this study builds upon the foundational YOLO-v5 and YOLO-v8 architectures. The resultant model excels in predicting six key categories: person, vest, and four helmet colors. The developed model is validated using a high-quality CHV benchmark dataset from the literature. The dataset is composed of 1330 images and manages to account for a real construction site background, different gestures, varied angles and distances, and multi-PPE. Consequently, the comparison among the ten models of YOLO-v5 (You Only Look Once) and five models of YOLO-v8 showed that YOLO-v5x6's running speed in analysis was faster than that of YOLO-v5l; however, YOLO-v8m stands out for its higher precision and accuracy. Furthermore, YOLOv8m has the best mean average precision (mAP), with a score of 92.30%, and the best F1 score, at 0.89. Significantly, the attained mAP reflects a substantial 6.64% advancement over previous related research studies. Accordingly, the proposed research has the capability of reducing and preventing construction accidents that can result in death or serious injury.

**Keywords:** construction safety; PPE detection; deep learning; computer vision; mAP score; You Only Look Once (YOLO)



**Citation:** Elesawy, A.; Mohammed Abdelkader, E.; Osman, H. A Detailed Comparative Analysis of You Only Look Once-Based Architectures for the Detection of Personal Protective Equipment on Construction Sites. *Eng* **2024**, *5*, 347–366. <https://doi.org/10.3390/eng5010019>

Academic Editor: F. Pacheco Torgal

Received: 30 November 2023

Revised: 19 January 2024

Accepted: 14 February 2024

Published: 21 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

One of the riskiest fields of work is thought to be the construction industry. Compared to workers in other industries, construction workers have twice as high a risk of injury. The nature of construction sites is well known for its high volume of activity, large machinery, frequent incidents, and numerous risks, all of which call for the careful consideration and application of safety precautions. Personal protective equipment (PPE) is the main line of defense against any threats that workers may encounter during their presence on construction sites. Manual inspections, which may be laborious and prone to human mistakes, are a major component of traditional ways of guaranteeing PPE compliance. Further, it was found that 70% of fall accidents in 2017 occurred because of workers not wearing personal protective equipment (PPE) on sites, as per a study conducted by Kang et al. [1]. Further, statistics [2–4] show that there is a significant risk of worker fatalities

and injuries in the construction sector. In addition, statistics on worker accidents in the construction industry are continuously increasing, which is alarming and points to the urgency of developing safety tracking systems for construction sites. For example, research by the Korea Occupational Safety and Health Agency states that among all industries, the construction sector has the second-highest rate of occupational accidents/injuries (25.5%) and the highest rate of fatalities (46.7%) [5].

Notwithstanding human efforts in manual and visual inspections, computer vision techniques have been developed and have progressed. This development is crystal clear to see in the use of automated PPE detection systems, which have more options nowadays than ever before. Such detection systems are a viable way to automate PPE recognition on building sites, improving safety and lessening the workload for engineers and site managers. Hence, significant efforts are currently being made to improve worker safety, which also greatly benefits construction companies because PPE can reduce the probability and severity of falling accidents. Creating recognition and monitoring systems for PPE used during working hours is one of the targeted initiatives. That is why Ferdous and Ahsan [6] created a YOLO-based architecture model for the recognition of workers wearing PPE on construction sites. Consequently, artificial intelligence (AI) capabilities can be adapted to create reasonably priced automation solutions for the construction industry, such as monitoring systems that can identify workers and PPE and determine whether or not they are adhering to safety requirements.

The major objective of this research is to exploit AI's capabilities to create a PPE detection system by using YOLO-based architecture. This aim can be established by attaining the following subtargets: (1) evaluating the accuracy of the performance of YOLO-based architectures in creating a PPE detection system; (2) comparing between models' performance matrixes, such as precision, recall, and mAP; (3) exploring trade-offs between the speed and accuracy of different YOLO architectural models; and (4) proposing future recommendations for optimizing PPE detection systems in real-world applications.

The outline of this research study is delineated in the following manner. Section 2 enumerates previous research endeavors in relation to computer vision's adaptation through detection and recognition systems in fields such as the construction industry. Section 3 describes steps that were taken during the preparation of this study; it also incorporates the research methodology framework. Section 4 illustrates the evolution of YOLO across the years, in addition to enumerating the framework used in the implementation of the YOLO model. Section 5 represents the dataset used in the training, validation, and testing of different YOLO models, as well as incorporating the code used for the training of the YOLO model. Section 6 highlights the performance evaluation matrix that can be considered as the basis of comparison between the different YOLO models. Section 7 clarifies the results of comparing the YOLO models in terms of the performance evolution matrix (recall, precision, F1 score, and mAP). Section 8 summarizes the whole study and provides readers with recommendations for future YOLO training to be more beneficial in the construction field.

## 2. Literature Review

Preserving the safety of construction workers during their presence on construction sites is the main aim of this study. This can be achieved by reducing the probability and/or severity of construction incidents. According to recent studies, as mentioned in the introduction, construction workers' safety is highly dependent on them wearing proper personal protective equipment (PPE) during their presence on construction sites. Hence, this study was conducted to continue work on recognizing PPE by using computer vision applications. Consequently, this literature review shows a number of recent studies on such computer vision techniques, including a number of investigations conducted within the latest six years. The recent studies that will be discussed and reviewed in this paper mainly concentrate on the progression of PPE recognition using computer vision techniques such as convolutional neural networks (CNNs) and the application of transfer learning principles. In addition, by examining these studies, a research gap can be identified, as well

as recommendations for future research for the development of an effective PPE detection system that can be tailored specifically for the construction field. As a result, and in terms of incident reduction and avoidance, this would be advantageous.

In this regard, the Web of Science (WoS), which incorporates a significant number of high-impact papers, is the most widely used platform for databases of the scientific literature. Thus, researchers often use this database to collect precise data for bibliometric analyses [7,8]. Consequently, the investigation of the literature review for this research is obtained from the WoS database. Accordingly, to find the desired papers in the database, a variety of criteria are analyzed, such as (construction worker) AND ((safety) OR (risk) OR (health)) AND ((machine learning) OR (deep learning) OR (computer vision) OR (vision-based)).

In a study conducted by Delhi et al. [9], the researchers applied a type of deep learning, which is computer vision, by recognizing the PPE on construction sites on an immediate basis. Accordingly, the researchers collected the dataset on which they conducted the research manually, in addition to applying web scraping. The dataset contained around 2500 images that were classified into four classes, as follows: NOHARDHAT, NOJACKET, SAFE, and NOT SAFE. Hence, YOLO-v3 was trained on that dataset. Furthermore, following the augmentation step based on the data, YOLO-v3 was trained on a sample of data. This gives the model resilience and generalization by performing flipping along with rotation on both sides, left and right, with an angle of 30 degrees. Further, by using a validation test strategy, the provided dataset was split into 90%, 8%, and 2% random segments for training, validation, and testing, respectively. Consequently, and based on the tested data, the model succeeded to fulfil an mAP and F1 score of 97%.

Deep learning neural networks were applied in the research of Wang et al. [10] for real-time detection and recognizing of objects to address the problem of worker safety by making sure that employees followed safety protocol. They consequently suggested applying YOLO-v3, YOLO-v4, and YOLO-v5, which are detectors based on deep learning of YOLO architectures. They used data from a high-quality dataset called CHV. Such data incorporated 1330 images extracted from Wang et al.'s [11] dataset and broken down into six categories: person, vest, and helmets with four colors. The research results showed that YOLO-v5s had the fastest GPU performance of 52 FPS, while YOLO-v5x had the best mAP of 86.55%. A newly introduced cognitive analysis of safety measures for a monitoring system was proposed by Torrese et al. in another study [12]. Such a system was used in this study to instantly determine whether personal protective equipment is being used appropriately based on data gathered by the monitoring of CCTV cameras. Further, the system employed a deep learning algorithm to identify objects. Hence, the study resulted in the creation of a YOLO-v4 system that could achieve an 80.19% mAP at 80 frames per second in real time. Most of the current deep learning detectors had limitations with far-away objects and close-range objects [13,14]. YOLO models perform with a higher accuracy more than other detection models.

A similar study by Hayat and Morgado Dias [15] adopted a deep learning method for real time for the sake of identifying the heads and helmets on construction site workers. This paper investigated three different iterations of the well-known deep learning architecture YOLO: YOLO-v3, YOLO-v4, and YOLO-v5x. The model was implemented by the authors using the public dataset made available by Make ML [16]. Therefore, a huge number of 3000 instances were used for training, and 1000 instances were used for testing. Furthermore, in this study, only the "Head" and "Helmet" were used as classes. To address the preprocessing of the images, power-law transformation [17] was used for image preprocessing so as to increase the quality of contrast and lighting in such data. With accuracy, precision, recall, and F1 scores of 92%, 92.4%, 89.2%, and 90.8%, respectively, the YOLO-v5x model gave the best accuracy and, hence, the best performance.

Gallo et al. suggested a system in [18] to recognize personal protective equipment (PPE) in hazardous industrial areas. Deep neural networks were used for the system's analysis of a video stream. Five models—YOLO-v4, YOLO-v4-Tiny, SSD, CenterNet, EfficientDet,

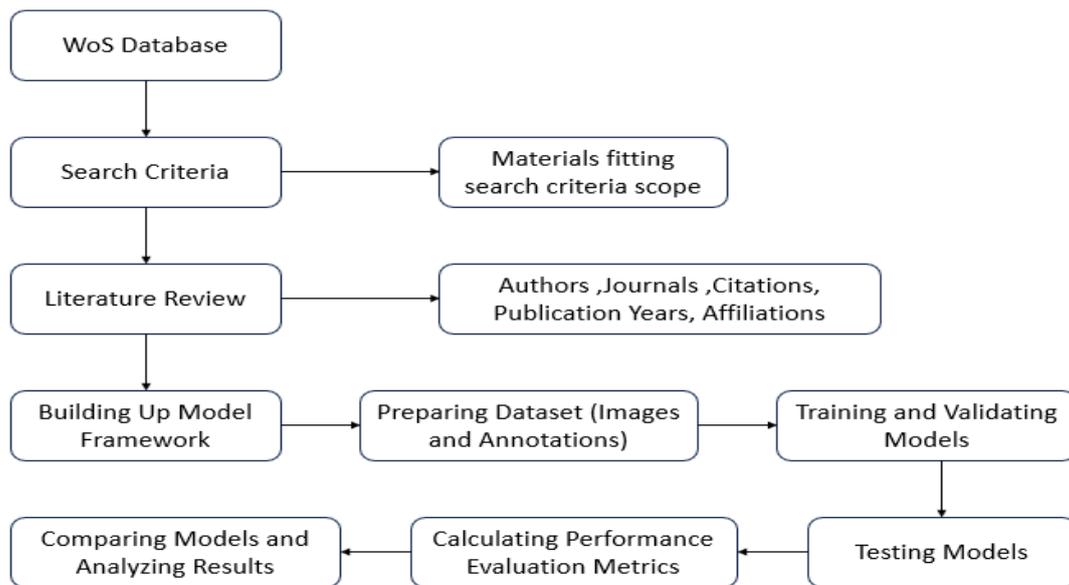
and MobileNet—were trained to determine whether or not the workers are implementing the safety measures by wearing safety equipment. The authors utilized three datasets: two were collected under controlled conditions and incorporated 215 and 236 images, respectively; the third one is an available public dataset [19] with 7035 images. Because of its rapid detection speed, YOLO-v4-tiny was implemented in the system, achieving an mAP of 86%. Further, and by using the INRIA person dataset [20], Li et al. [21] were able to train an autonomous safety-helmet-wearing recognition system. Furthermore, a safety helmet detection model was suggested by Wang et al. [22], trained using 10,000 photos taken on construction sites by 10 distinct surveillance cameras. Geng et al. [23] presented an enhanced helmet recognition method based on an unbalanced dataset of 7581 photos, the majority of which included a person wearing a helmet against a complicated backdrop. By testing it on 689 photos, it resulted in a label confidence of 0.982.

A transfer learning model-based automated technique was developed by Vibhuti et al. [24] to identify individuals who were not wearing masks in public in the period of the COVID-19 epidemic. InceptionV3, ResNet50, VGG16, MobileNet, MobileNetV2, and Xception were among the deep learning models that were employed in the intervention. Training, testing, and validation were conducted using the Simulated Masked Face dataset (SMFD) [25]. Through the use of fine-tuning strategy, the pretrained Inception (V3) model was developed and optimized. The greatest results, obtained with the SMFD dataset [26], were 100% accuracy and specificity in testing and 99.92% in training. The main outcomes highlighted the excellent accuracy of non-mask-wearer recognition automation achieved by the proposed transfer learning model.

Notwithstanding the above-mentioned studies, the previous studies and research did not address a detailed comparative analysis between the different YOLO-based architecture models' performance. This is deemed to be the research gap in the mentioned studies that is addressed and dealt with in this research by applying a detailed comparative performance analysis between the different YOLO models.

### 3. Research Methodology

In light of developing this research, we went through different phases to reach the optimum YOLO model in detecting PPE, as shown in Figure 1. The Web of Science (WoS) database was employed in this research to compile an extensive collection for the literature review to guide and inform our investigation. Moreover, in order to guarantee relevant articles and lay the groundwork for an extensive literature review, the search criteria were carefully crafted. Consequently, after gaining knowledge from previous research and studying the limitations of other research, building up different YOLO models was our goal to reach. This was achieved using Google Colab (accessed on 11 November 2023). The preparation of the dataset comprising both images and annotations was an important process to ensure that the results would be reliable. Subsequently, the stages of training and validating YOLO models were the highlighted phases. The model testing was performed by calculating performance evaluation metrics such as precision, recall, F1 score, and mAP. In the final stages, a comparative analysis of different YOLO models was conducted by analyzing the results to draw meaningful conclusions and contribute to the evolving field of detecting PPEs.

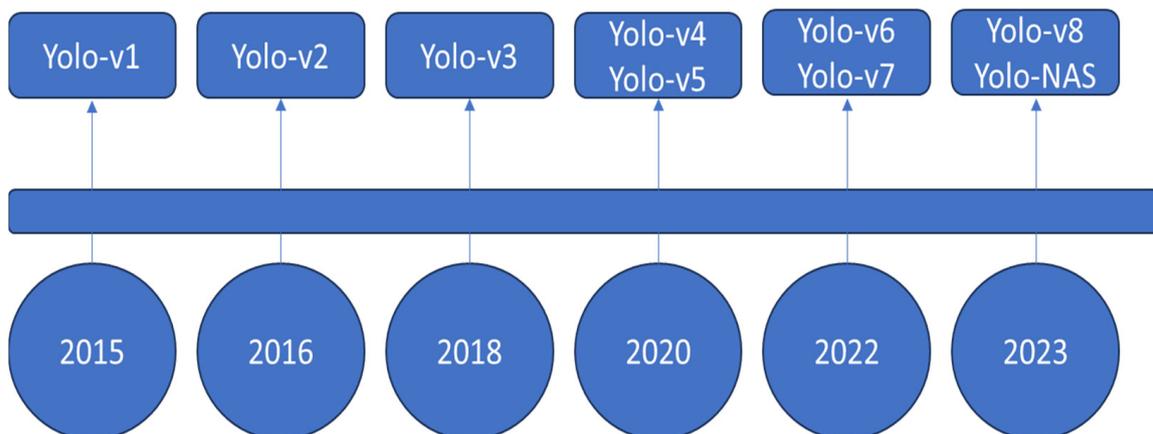


**Figure 1.** Framework of the proposed research methodology.

#### 4. Materials and Methods

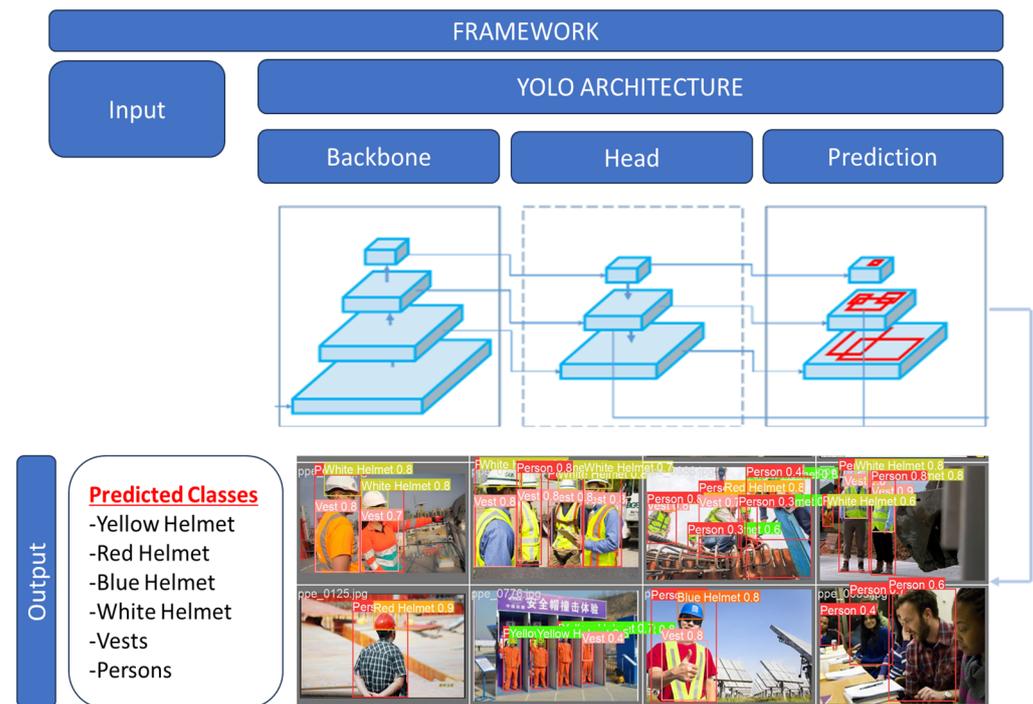
##### 4.1. Framework

In this study, YOLO-v5 and YOLO-v8 with their different versions were used as the primary models for classes detection. YOLO was first introduced to the computer vision field in 2015 by Joseph Redmon et al. [27] under a paper entitled “You Only Look Once: Unified, Real-Time Object Detection”. From 2015 to 2023, YOLO teams continued developing YOLO models and versions from one year to another. Figure 2 shows the evolution of YOLO across the years.



**Figure 2.** Evolution of YOLO across the years.

YOLO is a one-stage single-shot detection. YOLO uses a convolutional neural network (CNN) to process an image. YOLO makes a single pass on the input image to make a prediction for targeted classes. It processes the entire image only in a single pass. Different YOLO models contain different architectures, but all of them contain the same structure, consisting of three main parts: backbone, neck, and prediction. Figure 3 shows the framework of PPE detection based on YOLO models.



**Figure 3.** YOLO framework.

The backbone helps to produce visual features with different shapes and types by using the convolutional neural network. The neck is a set of layers which mix and combine image features in order to pass them to the next prediction step. The prediction stage takes the input from the neck stage in order to perform classification for the targeted classes.

#### 4.2. YOLO History

YOLO is a powerful real-time detection model which was first introduced in 2015 by Joseph Redmon et al. [27]. Later, in 2018, Joseph Redmon [28] upgraded YOLO-v1 to YOLO-v3, which was faster. YOLO is a one-stage single-shot detection. YOLO makes a single pass on an input image to make a prediction for targeted classes. Different versions of YOLO are faster than the two-phase object detection model. Two-phase object detection uses two phases for detection. The first phase generates a pool of probabilities for object locations. The second phase ensures these probabilities to make a final decision regarding targeted classes. YOLO-v3 is faster than two-phase object detection models such as the R-CNN and fast R-CNN models [28]. YOLO-v3 is 1000 times faster than R-CNN and 100 times faster than fast R-CNN.

Bochkovskiy et al. [29] released an updated version for YOLO, which is YOLO-v4, in 2020. Comparing YOLO-v4 to YOLO-v3, YOLO-v4 enhances the average accuracy (AP) and FPS by 10% and 12% [29]. Regarding YOLO's structure, Darknet53 was replaced by CSPDarknet53, which had a backbone stage. Later in 2020, a company called Ultralytics released a new version for YOLO named YOLO-v5 [30,31]. Ultralytics released YOLO-v5 as a source code on the Github website. YOLO-v5 contains ten different models. Table 1 shows YOLO-v5's different models and its characteristics [30].

On January 2023, Ultralytics released YOLO-v8. YOLO-v8 contains five different models. The models of YOLO-v8 produce more efficient output while using an equivalent number of parameters. Table 2 shows YOLO-v8's different models and its characteristics [32].

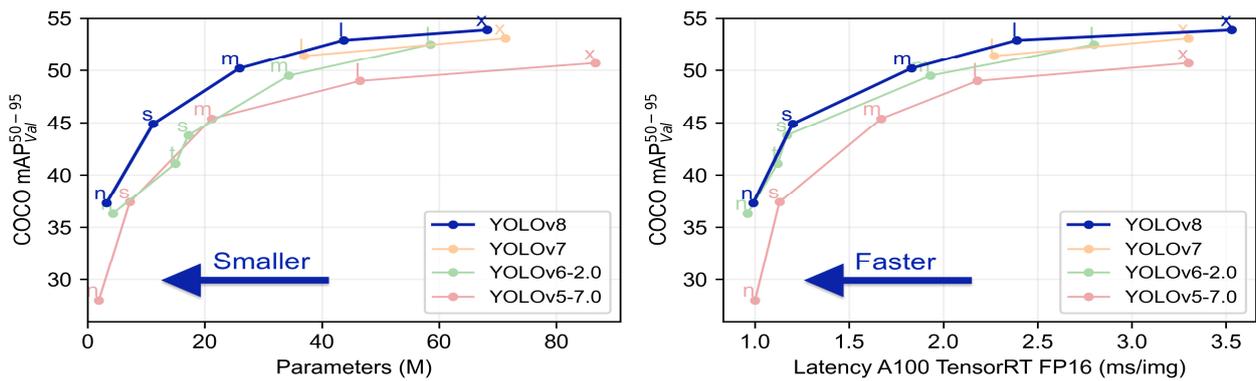
Figure 4 shows illustrates a comparison between different YOLO models trained on 640 image resolution [30].

**Table 1.** YOLO-v5’s different models [30].

Model	mAP (50–95)	Speed (ms)
YOLO-v5n	28.0	45
YOLO-v5s	37.4	98
YOLO-v5m	45.4	224
YOLO-v5l	49.0	430
YOLO-v5x	50.7	766
YOLO-v5n6	36.0	153
YOLO-v5s6	44.8	385
YOLO-v5m6	51.3	887
YOLO-v5l6	53.7	1784
YOLO-v5x6	55.0	3136

**Table 2.** YOLO-v8’s different models [32].

Model	mAP (50–95)	Speed (ms)
YOLO-v8n	37.3	80.4
YOLO-v8s	44.9	128.4
YOLO-v8m	50.2	234.7
YOLO-v8l	52.9	375.2
YOLO-v8x	53.9	479.1



**Figure 4.** Comparison between YOLO’s different models [30].

**5. Model Implementation**

*5.1. Dataset Description*

A public dataset containing PPE images was obtained from [10]. The dataset contains images for vests, colored helmets (blue, red, white, and yellow) and persons. The dataset is named the CHV dataset. The CHV dataset contains photos from real construction site conditions, unlike other datasets which contain images with backgrounds that are not from construction sites. The CHV dataset contains 1330 images with 9209 instances in total. The dataset contains different gestures (e.g., standing and bending), different angles (e.g., front, back, left, right, up), and distances (e.g., far away and close distance). Figure 5 shows the image distribution into training, testing, and validation sets.

The training set is used in the training process of the model; the testing set is used as the base for evaluation of the model; and the validation set is used to ensure that the model is predicting results as planned. Figure 6 shows the percentage of each class in training, testing, and validation sets.

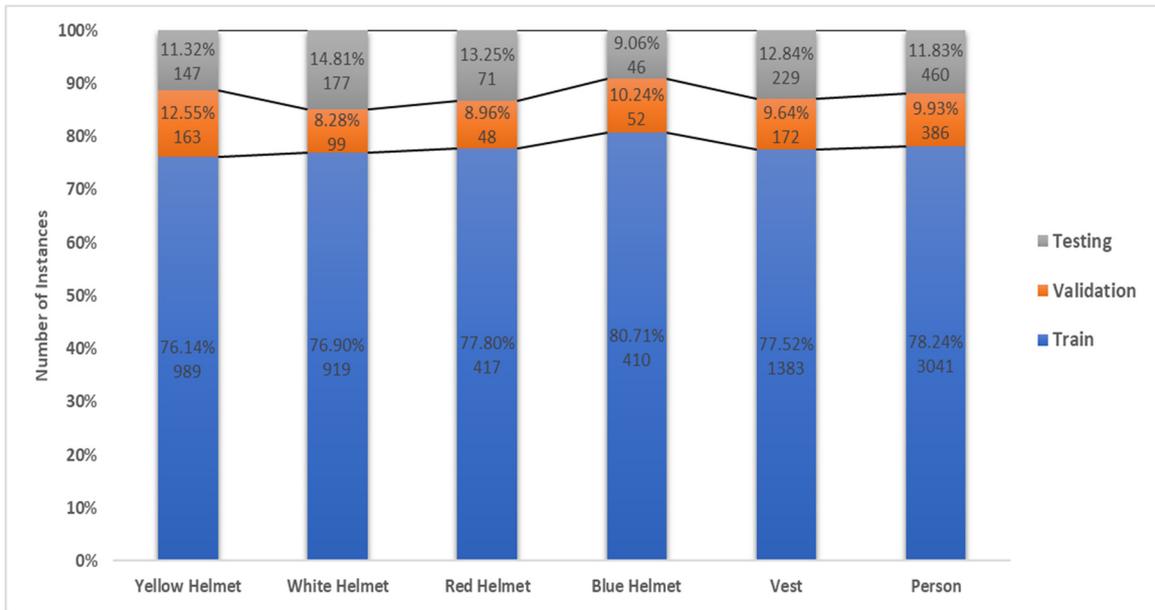


Figure 5. Image distribution into training, testing, and validation sets.

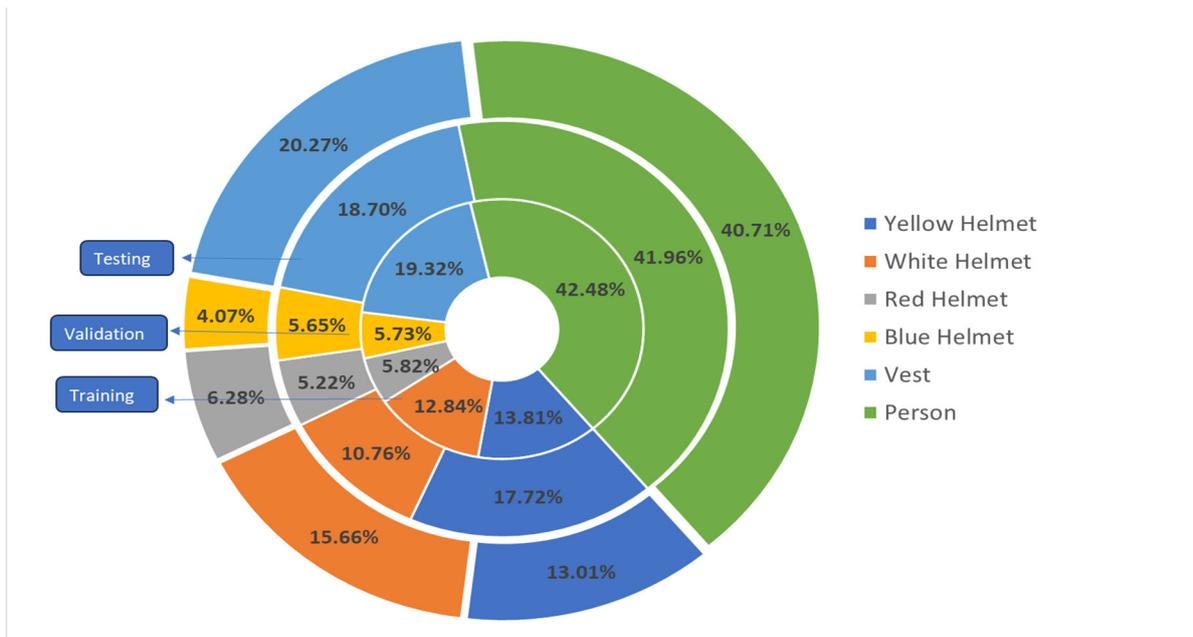


Figure 6. Percent of each class in training, testing, and validation sets.

### 5.2. Dataset Processing

The training process was performed on the cloud platform Google Colab, with a GPU Nvidia K80/T4 and GPU memory 16 GB with performance 4.1 TFLOPS/8.1 TFLOPS. The CHV dataset [11] contains three categories: training, validation, and testing. Figure 7 shows the code written on Google Colab in order to train, validate, and test by cloning up different YOLO models from [30,32]. The models were trained with 50 training epochs.

```
!git clone https://github.com/ultralytics/yolov5.git

%cd /content/yolov5

!pip install -r requirements.txt

import torch

import utils

display = utils.notebook_init() # checks

!mkdir /content/datasets
!mkdir /content/datasets/coco128
!mkdir /content/datasets/coco128/annotations
!mkdir /content/datasets/coco128/images
!mkdir /content/datasets/coco128/labels

from google.colab import drive drive.mount('/content/drive')

!cp /content/drive/MyDrive/Msc/CHVG.rar /content/

!unrar x "/content/CHVG.rar" "/content/"

#Training

!mkdir /content/datasets/coco128/images/train2017

!cp /content/CHVG/Train/*.jpg /content/datasets/coco128/images/train2017

!mkdir /content/datasets/coco128/labels/train2017
!cp /content/CHVG/Train/*.txt /content/datasets/coco128/labels/train2017

#Testing

!mkdir /content/datasets/coco128/images/val2017

!cp /content/CHVG/Test/*.jpg /content/datasets/coco128/images/val2017

!mkdir /content/datasets/coco128/labels/val2017

!cp /content/CHVG/Test/*.txt /content/datasets/coco128/labels/val2017

yaml_str = r"""\# Train/val/test

# Classes

nc: 6 # number of classes
```

**Figure 7.** *Cont.*

```

names: [ 'Person', 'Vest', 'Blue Helmet', 'Red Helmet', 'White Helmet', 'Yellow
HeLmet'] # class names

with open('/content/datasets/coco128/coco128.yaml', 'w') as f: f.writelines(yaml_str)

%cd /content/yolov5

!python train.py --img 640 --batch 16 --epochs 50 --data /content/datasets/coco128
/coco128.yaml --bbox_interval 1 --weights yolov5n6.pt

#Results to Download

!zip -r /content/yolov5/runs.zip /content/yolov5/runs/

!cp /content/yolov5/runs.zip /content/drive/MyDrive

%cd /content/yolov5

!python val.py --weights /con
tent/drive/MyDrive/Msc/1yolov5/10yolov5n6/runs/train/exp/weights
/best.pt--data /content/datasets/coco128/coco128.yaml --img 640

```

**Figure 7.** Google Colab code for using YOLO models (accessed on 11 November 2023).

The number of weights a YOLO model learns in a single epoch is greatly increased when a large number of images are sent to the model concurrently during training. In order to handle this, datasets are usually split up into smaller batches, with “n” photos in each batch, and training the model batch by batch. The size of the input image was  $640 \times 640$  with batch size of 16 photos. After training on all batches, the results of each batch are stored. The memory consumption increases as the number of batches increases. With a batch size of 16, a momentum value of 0.937 was implemented in the model. Optimization was carried out using the stochastic gradient descent (SGD) optimizer during the training process. The learning rate was adjusted at an initial rate of 0.01 and then periodically using a warm-up approach which applied a decay weight of 0.0005.

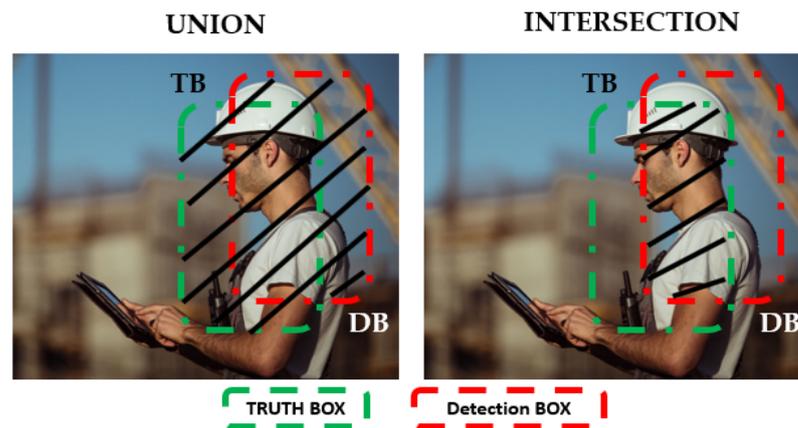
## 6. Metrics for Performance Evaluation

The CHV dataset was benchmarked by using state-of-the-art one-stage object detection models. The CHV dataset was tested by different YOLO models such as YOLO-v5x, YOLO-v5l, YOLO-v5m, YOLO-v5s, YOLO-v5n, YOLO-v5x6, YOLO-v5l6, YOLO-v5m6, YOLO-v5s6, YOLO-v5n6, YOLO-v8x, YOLO-v8l, YOLO-v8m, YOLO-v8s, and YOLO-v8n.

A basic metric to measure the performance of object detection algorithms is intersection over union (IOU). IOU is the ratio between the overlap of two boxes, ground truth box (TB) and detection box (DB). It is calculated using Equation (1) [33]. Figure 8 shows the relationship between the ground truth box (TB) and the detection box (DB).

$$\text{IOU} = \frac{\text{Intersection}}{\text{Union}} = \frac{\text{TB} \cap \text{DB}}{\text{TB} \cup \text{DB}} \quad (1)$$

After calculating an IOU, the confusion matrix criteria are applied using true positive (TP), false positive (FP), and true negative (TN). These basic concepts are described to aid in understanding the following equations, as follows. True positive (TP) is the correct detection of a ground truth bounding box [33]. False positive (FP) is the incorrect detection of a nonexistent object [33]. False negative (FN) is an undetected ground truth bounding box [33]. In object detection, true negative (TN) results do not apply as there are an infinite number of bounding boxes.



**Figure 8.** Relationship between ground truth box (TB) and detection box (DB).

One of the most important and difficult steps in machine learning is choosing appropriate metrics for performance evaluation. ROC curves, F1 score, precision, accuracy, and recall are frequently used metrics for comparison between different models [34,35]. They are not suitable for all datasets [36], especially when the positive and negative datasets are imbalanced [37]. Since accuracy and ROC curves do not accurately reflect the true classification performance of rare classes, they can be useless performance measures in unbalanced datasets [38,39]. In the proposed analysis, the precision, recall, F1 score, and mean average precision (mAP) were used as the evaluation metrics to perform comparison between YOLO's different models. Precision is the ability of a model to identify only the relative objects [33]. Precision shows the percentage of correct positive predictions among all detections [40], as shown in Equation (2).

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{All Detections}} \quad (2)$$

Recall is the ability of the model to find all the relevant cases [33]. Recall shows the percentage of true positives among all ground truths [41], as shown in Equation (3).

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\text{All Ground Truth}} \quad (3)$$

Moreover, the F1 score is the harmonic mean of precision and recall, as shown in Equation (4).

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Additionally, the most common metric used to measure the accuracy of the detection is mean average precision (mAP). The mAP is a metric used to measure the accuracy of object detectors over all classes, not only a specific class. The mAP is the score achieved by comparing the detected bounding box to the ground truth bounding box. If IOU is greater than or equal to 50%, the detection is counted as TP. The formula of the mAP is given in Equation (5).

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \quad (5)$$

where  $\text{AP}_k$  is the average precision of class  $k$  and  $n$  represents the number of classes. In this study,  $n = 6$  (person, vest, and four colored helmets).

In addition to assessing the above metrics, multiple metrics such as model layers, floating-point operations per second (FLOPs), and frames per second (FPS) were used to evaluate the performance and efficiency of the YOLO models. The complexity of the model is measured by FLOPs, which express the number of computations of the model. The

number of frames per second is represented by FPS. These metrics aid in the comprehension of variables including inference speed, computational cost, and model complexity.

## 7. Results and Discussion

### 7.1. YOLO Models Results Comparison

After running the model, precision  $\times$  recall curves were extracted from the model. For a precision  $\times$  recall curve, the accuracy of the model increases when the precision has a higher value accompanied with increase in the recall. Therefore, the curves which are closer to the right corners have higher performance. The precision  $\times$  recall curves for YOLO's different models are presented in Figure 9.

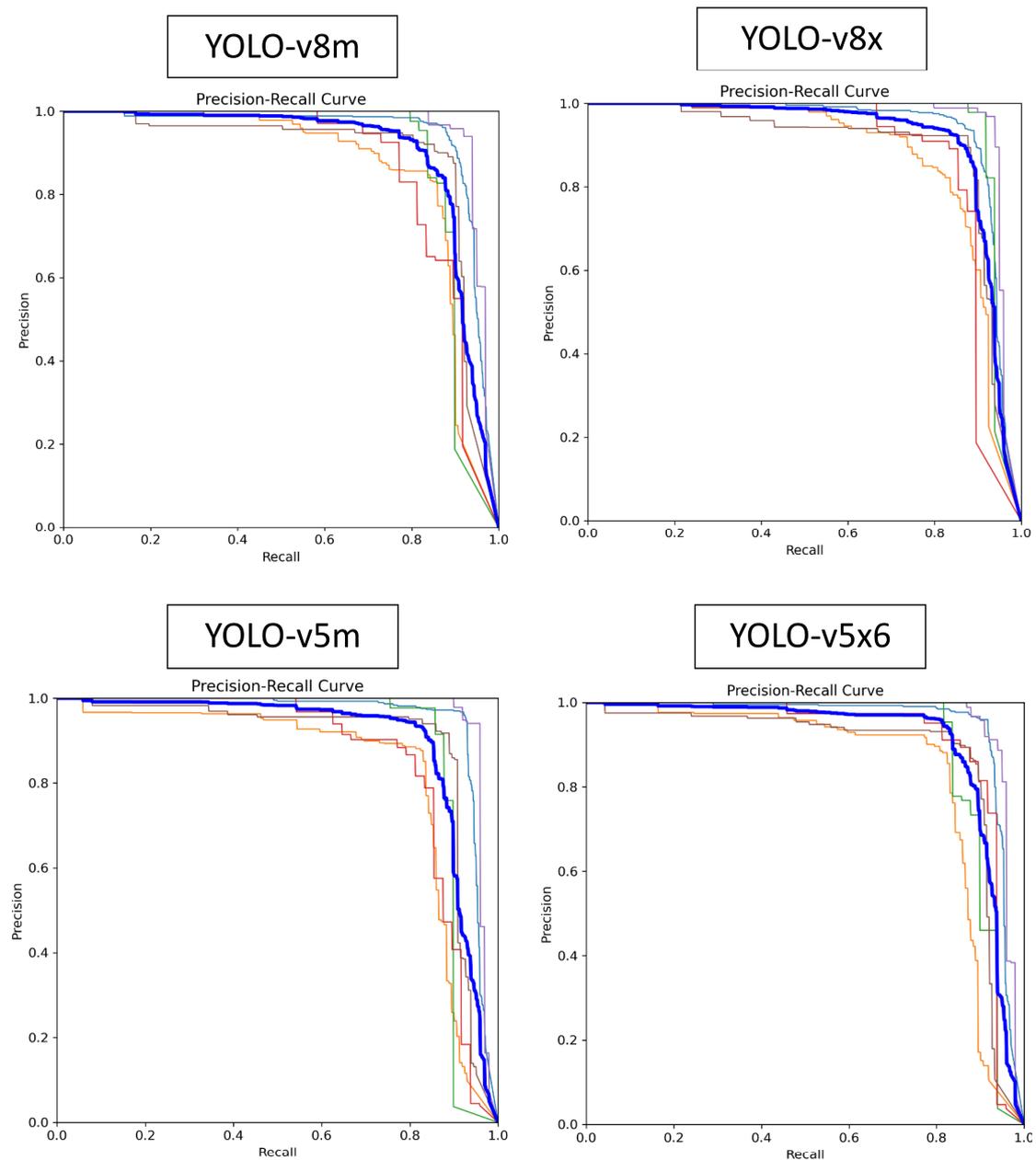
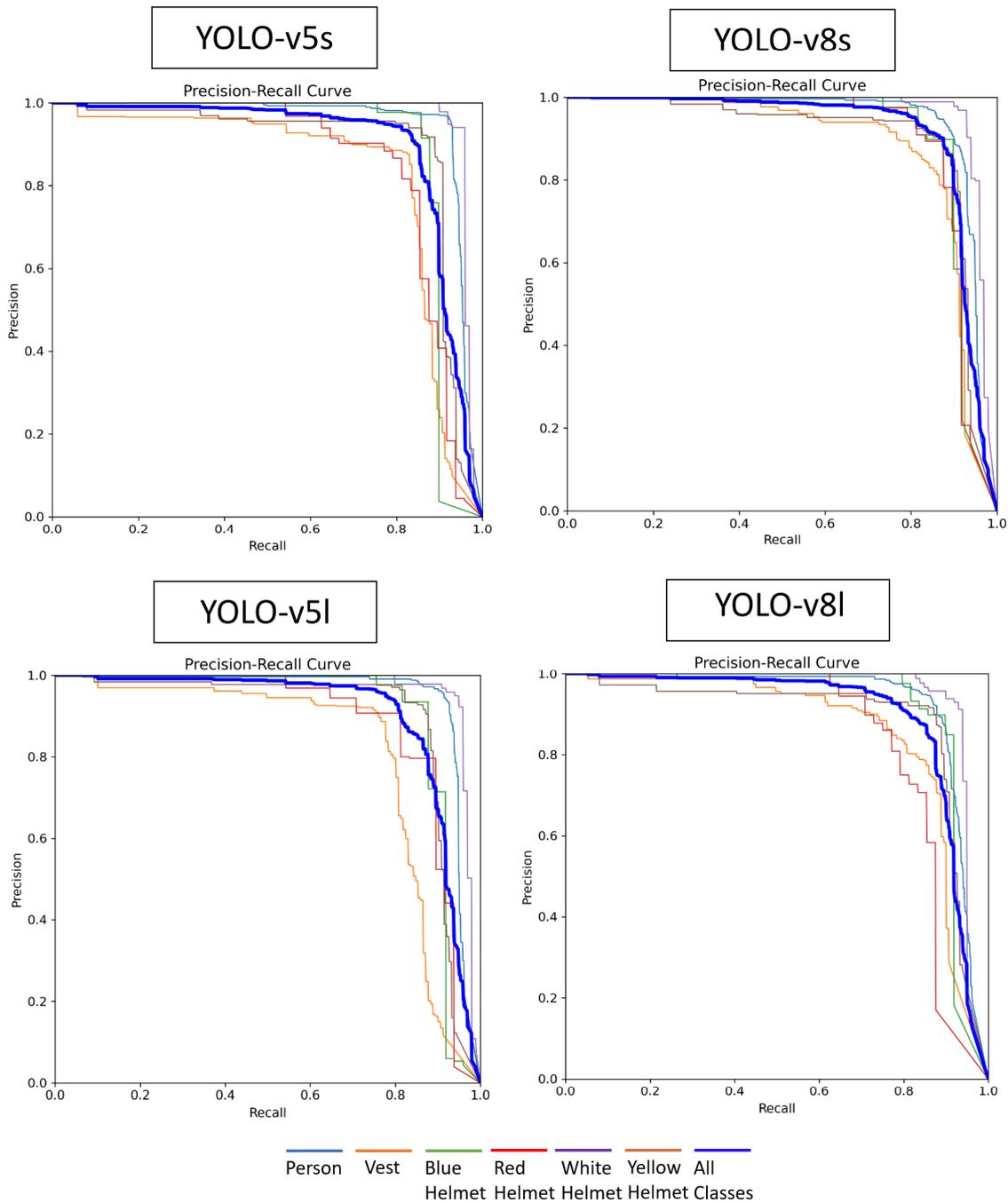


Figure 9. Cont.



**Figure 9.** Precision × recall curves of investigated YOLO-v5 and YOLO-v8 topologies.

In order to calculate the precision, recall, F1 score, and mean average precision (mAP), the TP, FP, and FN need to be extracted from the model after validating it. Table 3 shows the TP, FP, and FN for the ten YOLO-v5 models. Table 4 shows the TP, FP, and FN for the five YOLO-v8 models. Regarding the person class, YOLO-v5m6 scored the highest TP, which leads to an increase in precision and recall metrics. YOLO-v5n scored the highest FN, which leads to a decrease in recall metric. For the vest class, YOLO-v8s scored the highest TP, while YOLO-v5s6 scored the highest FN. For blue and red helmet classes, YOLO-v5n6 and YOLO-v8s scored the highest TP, while YOLO-v5N scored the highest FN. Other comparisons can be deduced from Tables 3 and 4.

**Table 3.** Performance comparative analysis of YOLO-v5 architectures.

Model	Person			Vest			Blue Helmet			Red Helmet			White Helmet			Yellow Helmet		
	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN
YOLO-v5x	346	27	48	136	20	25	44	9	11	44	9	7	94	7	14	139	13	11
YOLO-v5l	350	26	65	142	15	24	39	7	9	39	7	7	95	6	12	140	17	12
YOLO-v5m	345	24	72	133	12	25	42	5	9	43	7	4	93	11	18	141	8	13
YOLO-v5s	333	28	95	136	13	37	39	7	13	38	4	8	89	8	21	135	8	14
YOLO-v5n	313	31	104	125	12	41	36	7	15	31	12	8	86	8	19	133	10	19
YOLO-v5x6	345	23	50	141	13	33	40	3	11	39	2	5	89	8	16	136	10	14
YOLO-v5l6	348	23	63	139	13	39	40	3	10	41	2	7	90	4	17	143	13	16
YOLO-v5m6	355	30	48	135	16	30	39	2	12	39	3	5	90	9	16	134	15	15
YOLO-v5s6	344	31	68	132	20	45	36	6	21	35	4	7	85	6	26	126	8	19
YOLO-v5n6	346	85	51	135	108	33	44	9	11	44	9	7	82	30	28	139	13	11

**Table 4.** Performance comparative analysis of YOLO-v8 architectures.

Model	Person			Vest			Blue Helmet			Red Helmet			White Helmet			Yellow Helmet		
	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN
YOLO-v8x	352	44	52	143	14	27	45	7	15	43	3	5	91	8	13	144	11	16
YOLO-v8l	350	36	55	140	17	27	40	5	10	37	5	5	92	7	13	146	15	20
YOLO-v8m	352	34	68	149	13	30	41	5	11	40	1	4	92	5	16	140	16	19
YOLO-v8s	350	47	56	151	26	28	40	3	11	41	4	6	90	7	16	141	17	19
YOLO-v8n	344	54	61	140	15	43	36	6	15	36	2	7	88	8	21	136	12	23

After calculating TP, FP, and FN, the precision, recall, F1 score, and mean average precision (mAP) can now be calculated by applying Equations (2)–(5). Tables 5–8 show a comparative analysis between YOLO’s different models. Figures 10–13 elaborate the results shown in Tables 5–9.

**Table 5.** Precision comparative analysis for YOLO’s different models.

Model	Person	Vest	Blue Helmet	Red Helmet	White Helmet	Yellow Helmet	Overall
YOLO-v5x	0.93	0.87	0.84	0.83	0.93	0.91	0.88
YOLO-v5l	0.93	0.90	0.84	0.84	0.94	0.89	0.89
YOLO-v5m	0.94	0.92	0.89	0.87	0.90	0.95	0.91
YOLO-v5s	0.92	0.91	0.84	0.90	0.91	0.94	0.90
YOLO-v5n	0.91	0.91	0.84	0.73	0.91	0.93	0.87
YOLO-v5x6	0.94	0.92	0.92	0.95	0.91	0.93	0.93
YOLO-v5l6	0.94	0.91	0.93	0.94	0.96	0.92	0.93
YOLO-v5m6	0.92	0.90	0.94	0.93	0.91	0.90	0.92
YOLO-v5s6	0.92	0.87	0.86	0.90	0.94	0.94	0.90
YOLO-v5n6	0.80	0.56	0.83	0.83	0.73	0.91	0.78
YOLO-v8x	0.89	0.91	0.87	0.94	0.92	0.93	0.91
YOLO-v8l	0.91	0.89	0.89	0.89	0.93	0.91	0.90
YOLO-v8m	0.91	0.92	0.89	0.98	0.95	0.90	0.92
YOLO-v8s	0.88	0.85	0.92	0.91	0.93	0.89	0.90
YOLO-v8n	0.87	0.90	0.86	0.96	0.92	0.92	0.90

**Table 6.** Recall comparative analysis for YOLO's different models.

Model	Person	Vest	Blue Helmet	Red Helmet	White Helmet	Yellow Helmet	Overall
YOLO-v5x	0.88	0.84	0.81	0.86	0.87	0.93	0.86
YOLO-v5l	0.84	0.86	0.82	0.85	0.89	0.92	0.86
YOLO-v5m	0.83	0.84	0.82	0.92	0.84	0.92	0.86
YOLO-v5s	0.78	0.79	0.75	0.82	0.81	0.91	0.81
YOLO-v5n	0.75	0.75	0.70	0.80	0.82	0.88	0.78
YOLO-v5x6	0.87	0.81	0.79	0.88	0.85	0.91	0.85
YOLO-v5l6	0.85	0.78	0.80	0.86	0.84	0.90	0.84
YOLO-v5m6	0.88	0.82	0.77	0.88	0.85	0.90	0.85
YOLO-v5s6	0.84	0.75	0.64	0.84	0.77	0.87	0.78
YOLO-v5n6	0.87	0.80	0.80	0.86	0.74	0.93	0.83
YOLO-v8x	0.87	0.84	0.75	0.90	0.88	0.90	0.86
YOLO-v8l	0.87	0.84	0.80	0.88	0.87	0.88	0.86
YOLO-v8m	0.84	0.83	0.80	0.90	0.85	0.88	0.85
YOLO-v8s	0.86	0.84	0.79	0.88	0.85	0.88	0.85
YOLO-v8n	0.85	0.77	0.71	0.84	0.81	0.86	0.80

**Table 7.** F1 score comparative analysis for YOLO's different models.

Model	Person	Vest	Blue Helmet	Red Helmet	White Helmet	Yellow Helmet	Overall
YOLO-v5x	0.90	0.86	0.82	0.84	0.90	0.92	0.87
YOLO-v5l	0.88	0.88	0.83	0.85	0.91	0.91	0.88
YOLO-v5m	0.88	0.88	0.85	0.89	0.87	0.93	0.88
YOLO-v5s	0.84	0.84	0.79	0.86	0.86	0.92	0.85
YOLO-v5n	0.82	0.82	0.76	0.76	0.86	0.90	0.82
YOLO-v5x6	0.90	0.86	0.85	0.91	0.88	0.92	0.89
YOLO-v5l6	0.89	0.84	0.86	0.90	0.90	0.91	0.88
YOLO-v5m6	0.90	0.86	0.85	0.90	0.88	0.90	0.88
YOLO-v5s6	0.87	0.80	0.73	0.87	0.84	0.91	0.84
YOLO-v5n6	0.84	0.66	0.81	0.85	0.74	0.92	0.80
YOLO-v8x	0.88	0.88	0.81	0.92	0.89	0.91	0.88
YOLO-v8l	0.89	0.86	0.84	0.88	0.90	0.89	0.88
YOLO-v8m	0.87	0.87	0.84	0.94	0.90	0.89	0.88
YOLO-v8s	0.87	0.85	0.85	0.90	0.89	0.89	0.87
YOLO-v8n	0.86	0.83	0.78	0.89	0.86	0.89	0.85

**Table 8.** mAP comparative analysis for YOLO's different models.

Model	Person	Vest	Blue Helmet	Red Helmet	White Helmet	Yellow Helmet	Overall
YOLO-v5x	0.91	0.87	0.84	0.90	0.92	0.95	0.90
YOLO-v5l	0.92	0.88	0.84	0.89	0.94	0.93	0.90
YOLO-v5m	0.91	0.88	0.84	0.91	0.91	0.94	0.90
YOLO-v5s	0.91	0.87	0.77	0.86	0.89	0.93	0.87
YOLO-v5n	0.88	0.84	0.76	0.85	0.87	0.90	0.85
YOLO-v5x6	0.92	0.88	0.83	0.93	0.90	0.93	0.90
YOLO-v5l6	0.92	0.86	0.84	0.90	0.92	0.94	0.90
YOLO-v5m6	0.91	0.86	0.81	0.90	0.91	0.91	0.88
YOLO-v5s6	0.91	0.83	0.71	0.88	0.86	0.89	0.85
YOLO-v5n6	0.89	0.80	0.73	0.73	0.78	0.88	0.80
YOLO-v8x	0.93	0.90	0.83	0.95	0.92	0.92	0.91
YOLO-v8l	0.93	0.90	0.84	0.93	0.92	0.94	0.91
YOLO-v8m	0.91	0.90	0.88	0.96	0.93	0.93	0.92
YOLO-v8s	0.91	0.89	0.86	0.90	0.91	0.91	0.89
YOLO-v8n	0.91	0.88	0.79	0.91	0.88	0.92	0.88

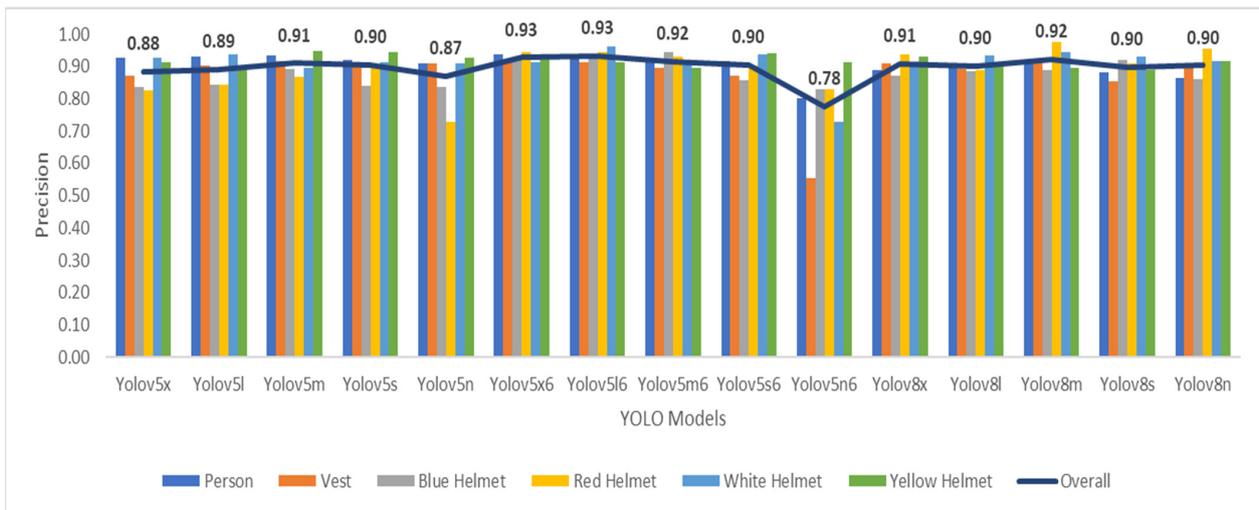


Figure 10. Precision comparative analysis for YOLO's different models.

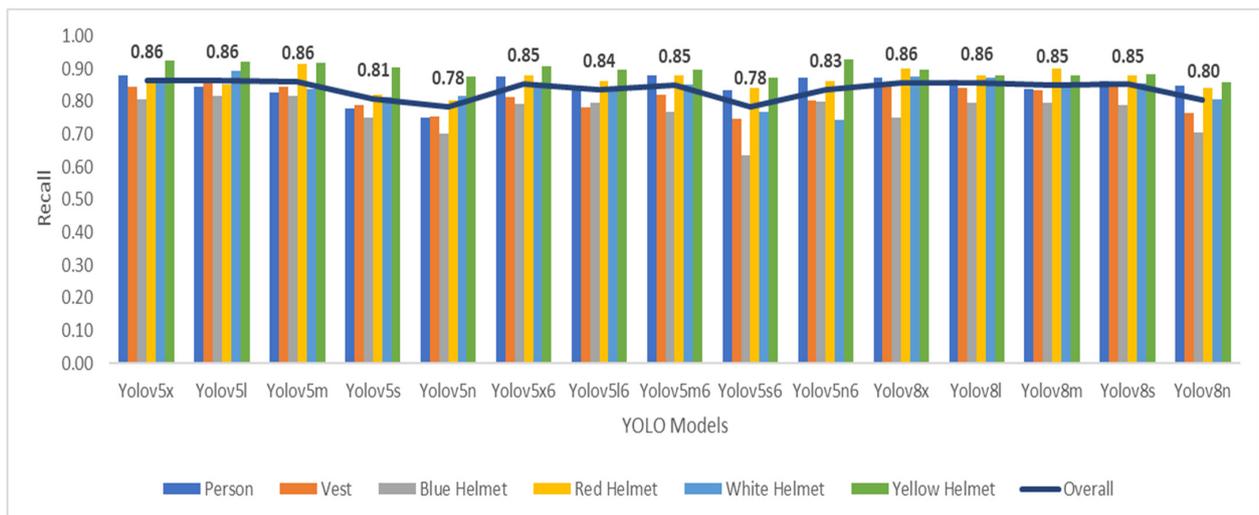


Figure 11. Recall comparative analysis for YOLO's different models.

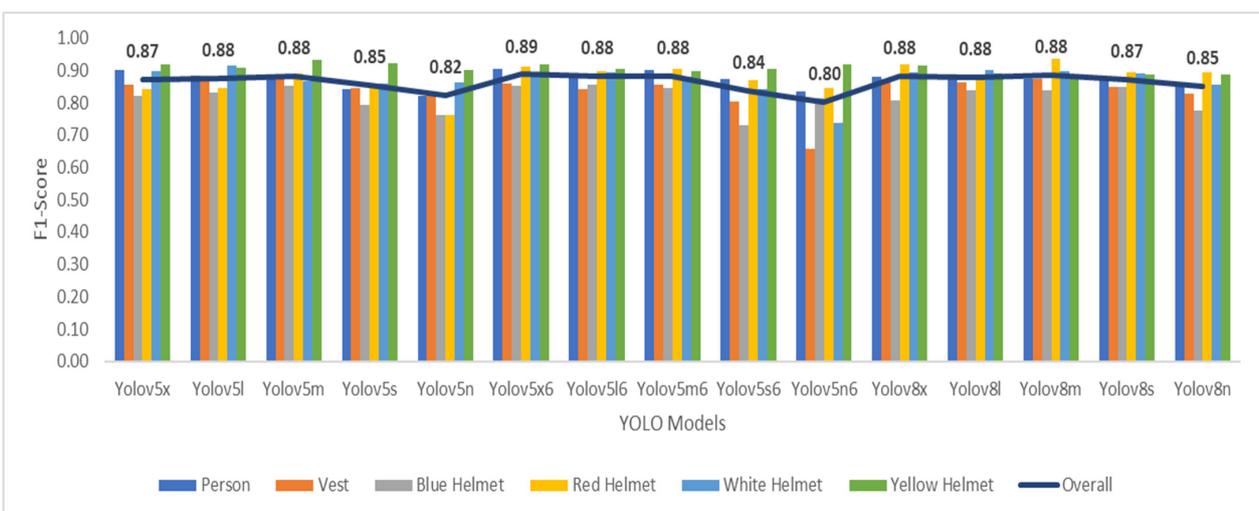


Figure 12. F1 score comparative analysis for YOLO's different models.

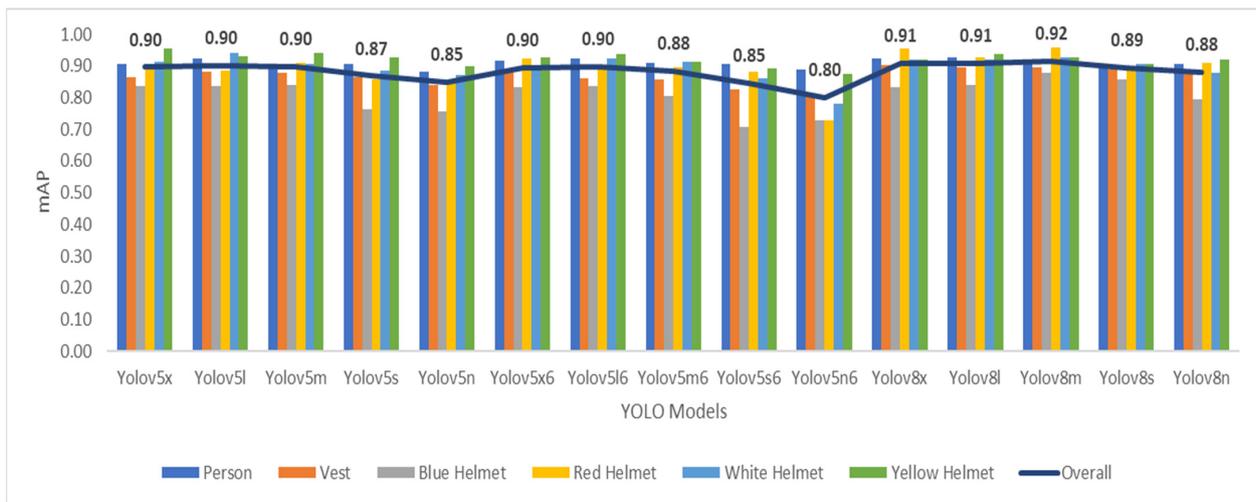


Figure 13. mAP comparative analysis for YOLO's different models.

Table 9. Complexity, FPS, and computational cost for YOLO's different models.

Model	Layers	FPS	Parameters (Million)	FLOPS (Giga)
YOLO-v5x	322	26.32	86.21	203.90
YOLO-v5l	212	52.36	20.87	47.90
YOLO-v5m	267	42.02	46.14	107.70
YOLO-v5s	157	63.69	7.03	15.80
YOLO-v5n	157	84.75	1.77	4.20
YOLO-v5x6	416	21.01	140.02	208.00
YOLO-v5l6	346	32.57	76.16	110.00
YOLO-v5m6	276	46.73	35.28	49.00
YOLO-v5s6	206	63.69	12.33	16.20
YOLO-v5n6	206	87.72	3.10	4.20
YOLO-v8x	268	20.37	68.13	214.55
YOLO-v8l	268	28.49	43.61	125.77
YOLO-v8m	218	41.32	25.84	55.41
YOLO-v8s	168	38.61	68.13	26.06
YOLO-v8n	168	59.17	3.01	6.23

### 7.2. Analysis of the Results

Regarding detailed comparative analysis of YOLO-based architectures for the detection of personal protective equipment (PPE) on construction sites, key findings were shown by various performance metrics. Notably, as shown in Table 5, in terms of precision, three models demonstrated exceptional accuracy in identifying PPE instances within construction site images. YOLO-v5x6 and YOLO-v5l6 were the top-performing models in terms of precision. Following them closely was YOLO-v8m. Turning our attention to the recall metric, the results show that YOLO-v5x, YOLO-v5l, and YOLO-v8m shared identical recall values. This suggests a comparable ability among these models to effectively capture instances of personal protective equipment in different construction site scenarios.

The F1 score, harmonic mean of precision, and recall revealed another set of top-performing models. YOLO-v5x6 stands out with the highest F1 score of 0.89, closely followed by YOLO-v5l, YOLO-v5m, YOLO-v8m, and YOLO-v8x, all achieving an F1 score of 0.88. These models demonstrated strong overall performance, balancing precision and recall effectively. YOLO-v5n6 and YOLO-v5n exhibited slightly lower F1 scores at 0.8 and 0.82, respectively. In general, these YOLO models offer reliable object detection capabilities, with specific variations in F1 scores that users can consider based on their application requirements.

The mAP (mean average precision) comparative analysis for various YOLO models indicates a consistent level of performance across different configurations. YOLO-v8m leads the group with the highest mAP of 0.92, closely followed by YOLO-v8x and YOLO-v8l, both achieving an mAP of 0.91. YOLO-v5x, YOLO-v5l6, YOLO-v5l, and YOLO-v5m share a common mAP of 0.90, highlighting their comparable precision in object detection. YOLO-v5s, YOLO-8n, and YOLO-v8s also delivered solid performance, with mAP values ranging from 0.87 to 0.89. YOLO-v5n6 and YOLO-v5s6 exhibited slightly lower mAPs at 0.8 and 0.85, respectively. Overall, these YOLO models showcase reliable and competitive mAP scores, with users able to choose based on specific application requirements and computational considerations.

Considering the values of model layers, YOLO-v5x6 has the highest number of layers (416), indicating a more complex architecture. YOLO-v5s and YOLO-v5n have the fewest layers (157), suggesting a simpler architecture. Regarding FPS, YOLO-v5n6 has the highest FPS (87.72), indicating faster processing of frames per second. YOLO-v5s and YOLO-v5s6 also have high FPS values, suggesting good real-time performance. YOLO-v8x has the lowest FPS (20.37), indicating slower frame processing. Concerning models' parameters, YOLO-v5x6 has the highest number of parameters (140.02 million), signifying a more complex model. YOLO-v5n has the lowest number of parameters (1.77 million), indicating a simpler model.

In addition to the abovementioned results, and after comparing computational cost for different models, it was deduced that YOLO-v8x has the highest FLOPS (214 million), suggesting higher computational efficiency. YOLO-v5x has the highest FLOPS among YOLO-v5 models, indicating higher computational load. Moreover, YOLO-v8x has the highest FLOPS among YOLO-v8 models. YOLO-v5n and YOLO-v5n6 have the lowest FLOPS (4.20 million), suggesting lower computational load. Accordingly, YOLO-v5n6 stands out for its high FPS and low FLOPS, indicating good real-time performance and computational efficiency. YOLO-v5x6, while having a high number of parameters, has a lower FPS, suggesting a trade-off between complexity and processing speed. The other models fall in between, offering a range of choices based on specific requirements for model complexity, real-time performance, and computational load. Notwithstanding the forementioned, the benefit of this detailed analysis is that YOLO-v8m benchmarks against previous related research studies with an increase of 6.64% advancement in mAP, making it more reliable and accurate.

## 8. Conclusions

To sum up, the construction workers have a higher risk of becoming injured. It was figured out that most of the construction accidents resulted from the construction workers' negligence in wearing the personal protective equipment (PPE). Hence, the researchers agreed that the PPE is the main line of defense against any threats that workers may encounter during their presence on construction sites, as per Delhi et al. [9]. Therefore, the YOLO-based architectural model was thought to be one of the AI applications in terms of tracking the workers' safety on construction sites by detecting the workers' who are not wearing the PPE.

In the light of the abovementioned conclusions and after experimenting with the CHV dataset on YOLO-v5 ten models and YOLO-v8 five models, the precision, recall, F1 score, mean average precision (mAP), number of layers, number of parameters, FPS, and FLOPS were calculated to compare between different YOLO models. YOLO-v8m benchmarks against previous related research studies with an increase of 6.64% advancement in mAP, making it more suitable for applications where detection performance is the measure for decision making. These findings collectively help to understand the strengths and capabilities of each YOLO-based architecture in the context of PPE detection on construction sites, providing valuable insights for the development and deployment of computer vision solutions in occupational safety applications.

In this regard, one drawback of different YOLO models is their difficulty in identifying small objects at a long distance, especially when there are some huge objects nearby. This might be resolved by changing or developing the architecture. Additionally, even though the model was extended to include six distinct PPE classes, it may be expanded more to detect more PPE sets, such as masks, glasses, and gloves. Regarding future research, our studies will be expanded to optimize each model and enhance its performance by gathering original datasets from different actual construction sites, which could help more in the model training process. This will be reflected in more accurate predictions.

**Author Contributions:** Conceptualization, A.E., H.O. and E.M.A.; methodology, A.E., H.O. and E.M.A.; formal analysis, A.E., H.O. and E.M.A.; data curation, A.E., H.O. and E.M.A.; investigation, A.E., H.O. and E.M.A.; resources, A.E., H.O. and E.M.A.; writing—original draft preparation, A.E., H.O. and E.M.A.; writing—review and editing, A.E., H.O. and E.M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Kang, Y.; Siddiqui, S.; Suk, S.J.; Chi, S.; Kim, C. Trends of Fall Accidents in the U.S. Construction Industry. *J. Constr. Eng. Manag.* **2017**, *143*, 04017043. [CrossRef]
- Keller, J.R. Construction Accident Statistics. Available online: <https://www.2keller.com/library/construction-accident-statistics.cfm> (accessed on 27 November 2023).
- U.S. Bureau of Labor Statistics (BLS). National Census of Fatal Occupational Injuries in 2014. 2015. Available online: [https://www.bls.gov/news.release/archives/foi\\_09172015.pdf](https://www.bls.gov/news.release/archives/foi_09172015.pdf) (accessed on 27 November 2023).
- Jeon, J.-H. Staff Reporter. 971 S. Korean Workers Died on the Job in 2018, 7 More than Previous Year. 2019. Available online: [https://www.hani.co.kr/arti/english\\_edition/e\\_natio%5Cnal/892709.html](https://www.hani.co.kr/arti/english_edition/e_natio%5Cnal/892709.html) (accessed on 27 November 2023).
- Lee, Y.R.; Jung, S.H.; Kang, K.S.; Ryu, H.C.; Ryu, H.G. Deep learning-based framework for monitoring wearing personal protective equipment on construction sites. *J. Comput. Des. Eng.* **2023**, *10*, 905–917. [CrossRef]
- Ferdous, M.; Ahsan, S.M.M. PPE detector: A YOLO-based architecture to detect personal protective equipment (PPE) for construction sites. *PeerJ Comput. Sci.* **2022**, *8*, e999. [CrossRef]
- Birkle, C.; Pendlebury, D.A.; Schnell, J.; Adams, J. Web of Science as a data source for research on scientific and scholarly activity. *Quant. Sci. Stud.* **2020**, *1*, 363–376. [CrossRef]
- Butt, N.S.; Malik, A.A.; Shahbaz, M.Q. Bibliometric analysis of statistics journals indexed in web of science under emerging source citation index. *SAGE Open* **2021**, *11*, 2158244020988870. [CrossRef]
- 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]
- Wang, Z.; Wu, Y.; Yang, L.; Thirunavukarasu, A.; Evison, C.; Zhao, Y. Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches. *Sensors* **2021**, *21*, 3478. [CrossRef] [PubMed]
- Wang, Z. Real-Time PPE Detection Based on YOLO. Open High-Quality Dataset. Available online: [https://github.com/ZijianWang-ZW/PPE\\_detection](https://github.com/ZijianWang-ZW/PPE_detection) (accessed on 11 November 2023).
- Torres, P.; Davys, A.; Silva, T.; Schirmer, L.; Kuramoto, A.; Itagyba, B.; Salgado, C.; Comandulli, S.; Ventura, P.; Fialho, L.; et al. A Robust Real-time Component for Personal Protective Equipment Detection in an Industrial Setting. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021), Online Streaming, 26–28 April 2021; pp. 693–700.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
- Hayat, A.; Morgado-Dias, F. Deep Learning-Based Automatic Safety Helmet Detection System for Construction Safety. *Appl. Sci.* **2022**, *12*, 8268. [CrossRef]
- Hard Hat Workers Dataset | MakeML—Create Neural Network with Ease. Available online: <https://makeml.app/datasets/hardhat-workers> (accessed on 11 November 2023).

17. Kumar, D.; Ramakrishnan, A.G. Power-law transformation for enhanced recognition of born-digital word images. In Proceedings of the 2012 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 22–25 July 2012; pp. 1–5.
18. Gallo, G.; Di Rienzo, F.; Garzelli, F.; Ducange, P.; Vallati, C. A Smart System for Personal Protective Equipment Detection in Industrial Environments Based on Deep Learning at the Edge. *IEEE Access* **2022**, *10*, 110862–110878. [[CrossRef](#)]
19. Hard Hat Workers Dataset. Available online: <https://public.roboflow.com/object-detection/hard-hat-workers> (accessed on 11 November 2023).
20. Dalal, N. INRIA Person Dataset. Available online: <http://pascal.inrialpes.fr/data/human/> (accessed on 26 November 2023).
21. Li, K.; Zhao, X.; Bian, J.; Tan, M. Automatic Safety Helmet Wearing Detection. In Proceedings of the 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Honolulu, HI, USA, 31 July–4 August 2017.
22. Wang, H.; Hu, Z.; Guo, Y.; Yang, Z.; Zhou, F.; Xu, P. A Real-Time Safety Helmet Wearing Detection Approach Based on CSYOLOv3. *Appl. Sci.* **2020**, *10*, 6732. [[CrossRef](#)]
23. Geng, R.; Ma, Y.; Huang, W. An improved helmet detection method for YOLOv3 on an unbalanced dataset. *arXiv* **2020**, arXiv:2011.04214.
24. Vibhuti; Jindal, N.; Singh, H.; Rana, P.S. Face mask detection in COVID-19: A strategic review. *Multimed. Tools Appl.* **2022**, *81*, 40013–40042. [[CrossRef](#)]
25. SMFD Dataset. Available online: <https://github.com/prajnasb/observations> (accessed on 11 November 2023).
26. Ferdous, M.; Ahsan, S.M.M. CHVG Dataset. Figshare. 2022. Available online: [https://figshare.com/articles/dataset/CHVG\\_Dataset/19625166/1](https://figshare.com/articles/dataset/CHVG_Dataset/19625166/1) (accessed on 11 November 2023).
27. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
28. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
30. Ultralytics. YOLOv5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 11 November 2023).
31. Jocher, G.; Stoken, A.; Borovec, J.; Christopher, S.T.A.N.; Laughing, L.C. Ultralytics/YOLOv5: v4.0-nn.SiLU() Activations, Weights& Biases Logging, PyTorch Hub Integration. Zenodo. 2021. Available online: <https://zenodo.org/record/4418161> (accessed on 11 November 2023).
32. Ultralytics. YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 11 November 2023).
33. 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, Marrakech, Morocco, 4–6 June 2020; pp. 237–242.
34. Ma, C.; Zhang, H.H.; Wang, X. Machine learning for Big Data analytics in plants. *Trends Plant Sci.* **2014**, *19*, 798–808. [[CrossRef](#)]
35. Müller, A.C.; Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016.
36. Liu, Y.; Zhou, Y.; Wen, S.; Tang, C. A Strategy on Selecting Performance Metrics for Classifier Evaluation. *Int. J. Mob. Comput. Multimed. Commun.* **2014**, *6*, 20–35. [[CrossRef](#)]
37. Orozco-Arias, S.; Isaza, G.; Guyot, R.; Tabares-Soto, R. A systematic review of the application of machine learning in the detection and classification of transposable elements. *PeerJ* **2019**, *7*, 18311. [[CrossRef](#)] [[PubMed](#)]
38. Kamath, U.; De Jong, K.; Shehu, A. Effective automated feature construction and selection for classification of biological sequences. *PLoS ONE* **2014**, *9*, e99982. [[CrossRef](#)]
39. Eraslan, G.; Avsec, Ž.; Gagneur, J.; Theis, F.J. Deep learning: New computational modelling techniques for genomics. *Nat. Rev. Genet.* **2019**, *20*, 389–403. [[CrossRef](#)] [[PubMed](#)]
40. Schietgat, L.; Vens, C.; Cerri, R.; Fischer, C.N.; Costa, E.; Ramon, J.; Carareto, C.M.A.; Blockeel, H. A machine learning based framework to identify and classify long terminal repeat retrotransposons. *PLoS Comput. Biol.* **2018**, *14*, e1006097. [[CrossRef](#)] [[PubMed](#)]
41. Rawal, K.; Ramaswamy, R. Genome-wide analysis of mobile genetic element insertion sites. *Nucleic Acids Res.* **2011**, *39*, 6864–6878. [[CrossRef](#)] [[PubMed](#)]

**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.