Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11280-6

1200: MACHINE VISION THEORY AND APPLICATIONS FOR CYBER PHYSICAL SYSTEMS

YOLOv4 algorithm for the real-time detection of fire and personal protective equipments at construction sites

Saurav Kumar[1] · Himanshu Gupta[2] · Drishti Yadav[3] · Irshad Ahmad Ansari[4] · Om Prakash Verma[2]

Received: 13 October 2020 / Revised: 19 May 2021 / Accepted: 8 July 2021

Abstract

Many difficulties are encountered during evacuation from construction sites in hazardous situations, which may lead to severe fatalities. These fatalities, especially caused by fire, may be significantly reduced by ensuring personal protective equipment (PPE) compliance of construction site workers and fire detection through proper surveillance. Thus, the detection of PPEs, fire and injured or trapped persons, can greatly assist in the reduction of fatalities and economic loss. This article presents a novel approach towards the detection of fire and PPEs to assist in the monitoring and evacuation tasks. This work utilizes the YOLOv4 and YOLOv4-tiny algorithms based on deep learning for carrying out the detection task. A self-made dataset has been utilized to train the model in the Darknet neural network framework. Moreover, a comparative analysis with previous works has been carried out in order to endorse the real-time efficacy of the proposed work. The results verify the strength of YOLOv4 algorithm in real-time detection and surveillance at construction sites with maximum mean average precision (mAP) of 76.86 %.

✉ Om Prakash Verma
vermaop@nitj.ac.in

Saurav Kumar
saurav_k@ee.iitr.ac.in

Himanshu Gupta
guptah.nitj@gmail.com

Drishti Yadav
drish131196@gmail.com

Irshad Ahmad Ansari
irshad@iiitdmj.ac.in

[1] Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India

[2] Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar, India

[3] Faculty of Informatics, Technische Universität Wien, 1040 Vienna, Austria

[4] Department of Electronics and Communication Engineering, Design and Manufacturing, PDPM Indian Institute of Information Technology, Jabalpur, India

Published online: 31 August 2021

Keywords CNN · Fire · Object detection · PPE · Surveillance · YOLOv4 algorithm

Abbreviations

AP	Average Precision
CNN	Convolutional Neural Networks
DL	Deep Learning
IoU	Intersection over Union
mAP	Mean Average Precision
PPE	Personal Protective Equipment
RFID	Radio Frequency Identification
YOLO	You Only Look Once

1 Introduction

With blossoming economic development, construction activities continue to be one of the most physically demanding industries. It is more labour intensive in developing countries, involving 2.5 – 10 times as many workers per activity, as compared to the developed ones [4]. Moreover, with a quite significant number of worker injuries and workplace accidents, the construction industry proves to be one of the most dangerous among all other industries. Therefore, the increasing scale and complexity of constructions have introduced great challenges in the reduction of construction site fatalities. To ensure worker safety, Occupational Safety and Health Administration in the United States, Health and Safety Executive in the United Kingdom, and similar agencies in other countries have developed safety codes and regulations for construction [21]. Despite these measures, in 2016–17 alone, the total number of injuries (~19%) was the highest in construction compared to other industries in the United States, which is exorbitant [28].

The main causes of construction related casualties are falls, electrocution, stuck in the equipment, collision and fire. The majority of these injuries can be prevented if workers use appropriate personal protective equipments (PPEs). In view of the governing laws and safety regulations, suitable PPEs at the construction sites are being enforced. Despite this, workers usually disobey these regulations due to discomfort in wearing PPE, lack of safety awareness, and work interference [1, 9]. Consequently, to reduce casualties, the monitoring of appropriate usage of PPE and ensuring an effective fire detection system becomes essential. However, traditional PPE and fire detection techniques are quite inefficient for large spaces, large under-construction sites, or spaces with many disturbances. These traditional techniques not only require huge capital investment and maintenance costs, but also suffer from missed detection, false alarms, and detection delays. These drawbacks cause difficulties in efficient monitoring of PPE and relaying accurate fire warnings [16]. Consequently, the development of an efficient PPE and fire detection approach has recently gained much interest among the scientific community.

2 Motivation

Although fire and PPE detection algorithms based on CNNs have shown remarkable performance in detection accuracy than traditional algorithms, some problems still exist. Most of the existing works focus on detecting either fire or PPE [27, 40]. This cannot provide one place safety solution, which is extremely beneficial for monitoring and evacuation. The present work is not only limited to the detection of fire, but also it is capable of detecting PPEs (safety glasses, helmet, etc.) and to find the location of the fire extinguisher to assist in extinguishing or controlling small fires, often in emergency situations.

In general, the existing techniques for automatic monitoring of PPE and fire include two different tasks: PPE detection and fire detection. They can be further classified into two types: sensor based (traditional) and vision based (intelligent). The sensor based approach for PPE detection uses Radio Frequency Identification (RFID) tags. In this approach, RFID tags are installed on each PPE component. Also, the tags are either scanned (at the entrance) or continuously monitored *via* Local Area Network (LAN) or wireless system to verify if it meets the regulations [3, 12, 26, 37]. However, huge capital costs in the installation and maintenance of complex sensor networks might prevent its practical implementation. On the contrary, vision-based approaches analyze PPE components and fire events by recording the images or videos of the site. These approaches provide comparatively rich spatial information and help in understanding the complex construction sites more precisely, and comprehensively [33]. Moreover, these approaches require a low-cost and flexible system installation. Also, these approaches are highly capable of identifying PPE compliance and fire events in complex building structures [22]. These characteristics of vision-based approaches encourage for the automatic PPE and fire detection.

There are three main stages in the process of image PPE and fire detection algorithms: (1) image pre-processing, (2) feature extraction, and (3) PPE and fire detection. Among them, feature extraction is the heart of algorithms. In pre convolutional neural networks (CNNs) era, these methods depend upon the manual selection of PPE and fire features, and machine learning based classification (like hard cap detection using edge detection [20] and Histogram of Oriented Gradient (HOG) [18], and safety vest detection using Support Vector Machine (SVM) [34] and *k*-Nearest Neighbor (*k*NN) [30]). These methods provide fast detection, but their accuracy depends upon professional knowledge. However, even with expert knowledge, only simple features (like edges, colours, and simple texture) can be discovered. Consequently, these algorithms are inappropriate for fire and PPE detection at construction sites. The construction sites often have complex PPE and fire scenes as well as many inferential events in practical applications. This makes it quite difficult to distinguish between PPEs, fire and look-like events, thereby causing low accuracy and weaker generalization ability.

Further, with the advent of CNNs, these algorithms gained much popularity owing to their unparalleled ability to automatically learn and extract complex image features. Moreover, they provide superior performance on object detection and tracking, surveillance, self-driving vehicles, medical diagnosis, etc. [15]. Precisely, deep learning (DL) algorithms have proved to be a revolutionary step in the field of computer vision [10]. DL techniques have been successfully employed for waste segregation [14] and nuclear waste object detection [36]. Attributable to the ground-breaking achievements in complex object detection tasks [6, 8, 11, 31], previous literature employed CNNs into the field of PPE and image fire detection including detection of safety guardrails [13], objects on roof construction sites [35], workers potentially unsafe behaviour [7], common construction-related objects [27], and fire [17, 19, 23, 25, 40]. These methods generally employ two-stage (CNN, R-CNN, and Faster R-CNN) or one-stage (SSD, YOLOv2, and YOLOv3) detection approach. In addition, efficient CNNs have been proposed for the detection of fire in surveillance tasks [24]. Despite all these developments, the methods of real-time PPE and fire detection are inadequate, although significantly important for safety [38].

Under the umbrella of the above discussion, this work aims towards the real-time detection of fire, PPE and persons at construction sites for effective monitoring and assisting in evacuation tasks. Concisely, the main contributions of this work can be sketched as follows:

- Proposing a novel deep learning based approach for the development of real-time fire detection system.
- Detecting a person with or without PPE including helmet: This may help in reducing casualties at construction sites.
- Finding the location of fire extinguisher to assist in extinguishing or controlling small fires in case of emergencies.

In the present work, the PPE is assumed to consist of helmet, safety vest, safety glasses, and fire extinguisher. Moreover, the literature reveals the supremacy of the YOLOv4 algorithm in handling complex object detection tasks with acceptable accuracy in real-time [5]. In view of this, the present work utilizes the YOLOv4 algorithm for the detection of PPE and fire. Moreover, to test the competence of YOLOv4 algorithm in the present application, its performance has also been compared with YOLOv4-tiny algorithm (a variant of YOLOv4).

The remainder of this paper is organized as follows. Section 3 describes the dataset utilized in the present investigation. This section also discusses the image acquisition method and augmentation techniques. A brief sketch of the YOLOv4 and YOLOv4-tiny algorithms is presented in Sect. 4. Then, details of the system specifications and parametric settings used to train the model have been presented in Sect. 5. In Sect. 6, the experimental results are presented and discussed. To end with, Sect. 7 gives the concluding remarks of the present work.

3 Dataset

The following subsections depict the strategy to gather and set up the image dataset, split the dataset into training and testing subsets, and lastly perform data augmentation to produce a bigger dataset from a moderately small number of images.

3.1 Data preparation

It is well known that the DL approach is data-driven and therefore, significant number of images with specific visual contents need to be acquired for training. For this purpose, open images dataset V6 by Google has been utilized to acquire approximately 10,000 images and their annotations. In this work, the image search has been carried out using the following keywords: helmet, bicycle_helmet, safety_vest, safety_glasses, fire, person and fire_extinguisher. In order to obtain a more accurate model, around 5,000 images have been acquired in JPEG format using the camera of Apple iPhone XR (64GB) with 1280 × 960 pixel resolution. After the pre-processing and cleaning of the collected data, 14,500 (97%) images have been utilized to form a self-made real-time dataset wherein each image is labeled with the name of the class to which it belongs. In the present investigation, these cleaned sample images have been grouped into six classes, namely: Fire, Person_With_Helmet, Person, Safety Vest, Fire Extinguisher and Safety Glass. Table 1 demonstrates the sample images of each class.

Table 1 Illustrative images of different classes

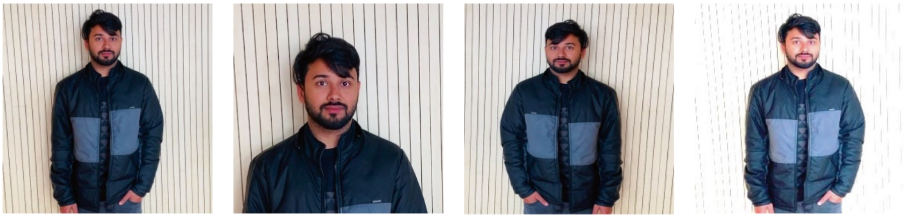
(a) Class 1: Fire	**(b)** Class 2: Person_With_Helmet	**(c)** Class 3: Person
(d) Class 4: Safety Vest	**(e)** Class 5: Fire Extinguisher	**(f)** Class 6: Safety Glass

3.2 Data splitting and augmentation

The prepared labeled dataset has been split into three parts namely training, validation, and testing. For this purpose, from the total images, 11,600 (80%) images are randomly selected as training images, 1,450 (10%) images are randomly chosen for validation, and the rest are utilized for testing purposes. After splitting, the training images are augmented to prevent overfitting by providing randomly distorted images. In this work, during each training epoch, training images are distorted by random scaling and horizontal flipping. Some of the augmentation techniques used for making the dataset are as follows: Flipping, Rotation, Shearing, Cropping, Zoom in, Zoom out, and Changing brightness or Contrast as depicted in Fig. 1. This provides the model an ability to recognize objects irrespective of the viewing angle and distance with respect to the camera.

4 Object detection using CNN

Any object detection problem in computer vision can be defined as identifying an object (a.k.a., classification) in an image, and then precisely estimating its location (a.k.a., localization) within the image. CNN performs this by subdividing the entire process into region



(a) Original image

(b) Scaling and cropping

(c) Horizontal flip

(d) Changing brightness / Contrast

Fig. 1 Illustration of Image augmentation

proposal, feature extraction, and classification. It takes an image or video sequence as an input and provides region proposals by convolution, stride, pooling, etc. It then predicts the presence or absence of PPE with fire in these regions using convolutional layers, residuals, fully-connected layers, etc. The convolutional layer is the heart of CNNs. It uses a set of image transform filters (known as kernels) to generate feature maps of original images which is responsible for accurate object detection. However, it has been found that kernels in earlier layers mainly learn spatial relationships by extracting simple features (like colour, edges, etc.). These spatial relationships cannot distinguish PPE, fire, and disturbances in noisy environments. Therefore, it becomes essential to use deep CNNs for extracting the semantic relationship for PPE and fire detection in real-time. Further, these networks require a substantial amount of time for accurate detection. Hence, there is always a trade-off between speed and accuracy.

Recently, You Only Look Once (YOLO) emerges as a highly noticeable and useful state-of-the-art DL technique which offers the advantage of real-time and synchronized object detection and classification [32]. Previous DL based object detection techniques (R-CNN and it's family) offer poor computational speed and high intricacy involved in optimization due to their pipeline architecture. These limitations of classical object detection techniques are addressed by YOLO by transforming the object detection task into a regression model. In contrast to its competitors, YOLO performs training on complete images resulting in optimized detection performance.

4.1 YOLOv4

The introduction of mosaic data enhancement in data processing and optimization of backbone, network training, activation, and loss function, has made YOLOv4 the best in the business. Moreover, YOLOv4 achieves the prominent balance between speed and accuracy for real-time object detection [5], as shown in Fig. 2.

YOLOv4 employs CSPDarknet53, an open source neural network module, as the fundamental backbone network to prepare and extract image features. After that, PANet (Path Aggregation Network) was utilized to achieve better extracted feature fusion and then, the head exploited YOLOv3 for object detection.

The structure of the PPE and fire detection model at construction sites based on YOLO v4 has been illustrated in Fig. 3. The composition and functions of the key modules are as follows:

- The CBL (Convolution, Batch Normalization and Leaky-ReLU) module was composed of a convolution layer, a batch normalization layer and a Leaky-ReLU activation function. It was similar to YOLOv3 network and the most repeatedly seen structure in the YOLOv4 network.
- The CBM (Convolution, Batch Normalization and MISH) module along with CBL was used for feature extraction. The only difference between these two was that in CBM, instead of Leaky-ReLU, MISH activation function was employed.
- The SPP (Spatial Pyramid Pooling) layer transforms convolution features of diverse magnitudes into pooled features of the same length.
- The CSP (Center and Scale Prediction) module was used to enhance the learning ability of CNNs by separating low-level features into two parts and then blending cross-level features.

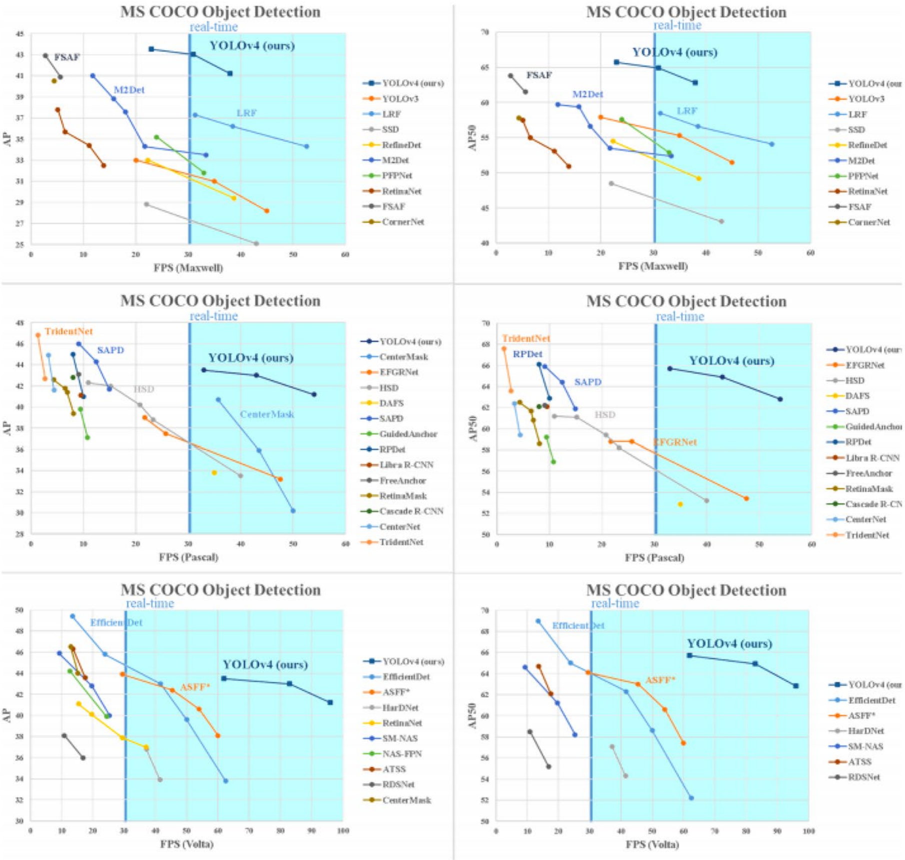


Fig. 2 Comparison of the speed and accuracy of different object detectors. (Some articles stated the FPS of their detectors for only one of the GPUs: Maxwell/Pascal/Volta) [5]

4.1.1 Deep transfer learning

YOLOv4, like other CNNs, can also be trained from scratch. However, in order to achieve optimal results, this approach requires substantial training data along with hyper-parameter tuning, which takes significant processing time. In order to overcome these difficulties, transfer learning has been implemented. It achieves significantly better and reliable performance for image PPE with fire detection at construction sites. In this process, a DL model has been pre-trained on a different, but large related dataset (a.k.a., source dataset). After that, the model has been re-trained on the comparatively smaller desired dataset (a.k.a., target dataset). Through this process, the model adapts itself to learn high- and mid-level features (edges, colours, shapes, etc.) from the source dataset that are relevant and useful for the classes in the target dataset. In this work, a pre-trained YOLOv4 model on COCO dataset has been utilized.

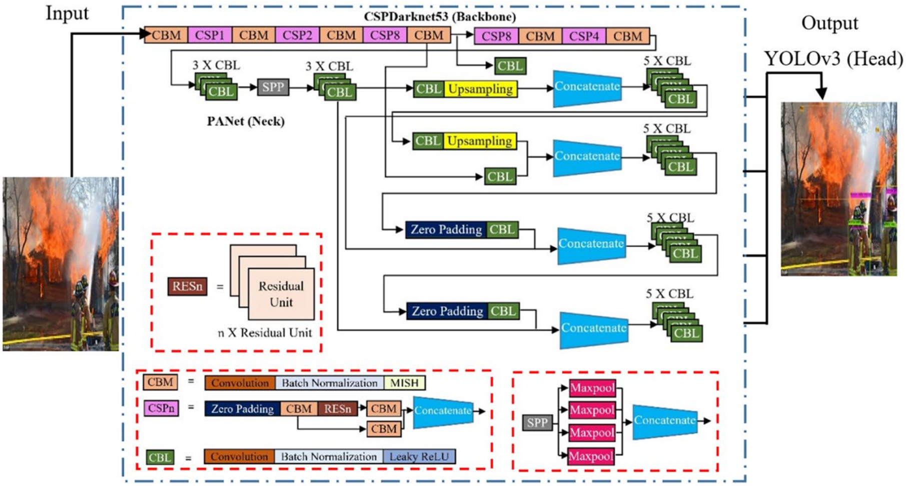


Fig. 3 PPE and fire detection based on YOLOv4 [5]

4.1.2 Performance parameters

In the present investigation, some of the fundamental key values [29] were inspected over the entire phase of training for exploring the performance of YOLOv4 in the recognition of face masked individuals. A brief discussion of these fundamental key values is as under:

(i) **Precision:** Precision is expressed in terms of the ratio of number of objects detected correctly to the number of total objects detected. Mathematically, Precision can be computed using Eq. (1):

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{1}$$

(ii) **Recall:** Recall is evaluated in terms of percentage of the number of objects which are correctly detected to the number of ground truth objects. Recall can be evaluated using Eq. (2):

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{2}$$

Here, NTP = Number of True Positives, i.e., number of objects detected correctly

NFP = Number of False Positives, i.e., number of detected objects which could not correspond to the ground truth objects

NFN = Number of False Negatives, i.e., number of ground truth objects that could not be detected

(iii) **Intersection over Union (IoU):** One of the recognized evaluation metrics in object detection tasks is IoU which is mathematically represented by Eq. (3). The concept of IoU has been illustrated in Fig. 4.

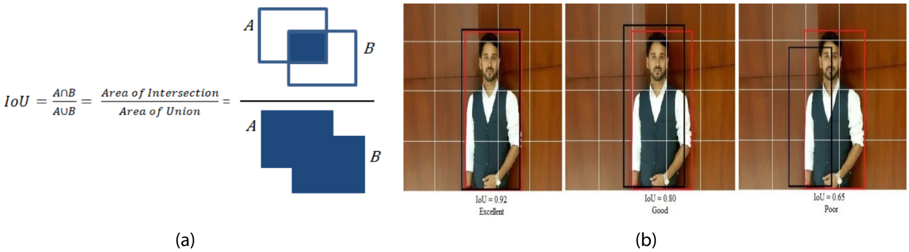


Fig. 4 The concept of IoU (**a**) Graphical representation (**b**) Example (Red colored bounding box represent ground truth and black is for predicted bounding box)

$$IoU = \frac{A \cap B}{A \cup B} \tag{3}$$

In Eq. (3) and Fig. 4, A and B represent the bounding boxes of prediction and ground truth respectively.

(iv) **Average Precision (AP):** In general, a precision-recall curve (corresponding to a definite threshold value of IoU), can be drawn once the values of precision and recall are identified. Average Precision (AP) is the area under the precision-recall curve which is expressed by Eq. (4):

$$AP = \int_0^1 p(r)dr \tag{4}$$

(iv) **Mean Average Precision (mAP):** The mean of average precisions of all classes (say for N number of classes) specified in the test model is termed as mAP and is mathematically represented by Eq. (5):

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{5}$$

(vi) **Loss function:** A necessary criterion for performance analysis of YOLOv4 on the test model is the evaluation of the value of Loss function. Typically, the summation of bounding box location loss (L_{CIoU}), confidence loss ($L_{Confidence}$), and classification loss (L_{Class}) have been used as the loss function [39] for training the YOLOv4 PPE and fire detection model. The loss function is mathematically expressed by Eq. (6) as:

$$Loss = L_{CIoU} + L_{Confidence} + L_{Class} \tag{6}$$

Here, L_{CIoU} is the error associated with bounding box location and being expressed by Eq. (7).

$$L_{CIoU} = 1 - IoU + {}^{d^2}/_{c^2} + \alpha\vartheta \tag{7}$$

where,

$$\alpha = \frac{\vartheta}{(1 - IoU) + \vartheta}$$

$$\vartheta = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{8}$$

In Eq. (7), d and c respectively represent the diagonal distance of predicted and ground truth bounding boxes, and the distance between the two bounding box centers. The respective width and height of ground truth bounding box represented by h^{gt} and w^{gt} while that of predicted bounding box are represented by h and w.

In Eq. (6), $L_{Confidence}$ signifies the confidence error been represented by Eq. (9).

$$L_{Confidence} = \sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\{-\log(p) + BCE(\hat{n}, n)\} \tag{9}$$

where,

$$BCE\left(\hat{n}, n\right) = -\hat{n}\log(n) - (1 - \hat{n})\log(1 - n) \tag{10}$$

$$\text{Here, } 1_{ij}^{obj} = \begin{cases} \text{1, if the object falls into the j}^{\text{th}}\text{bounding box in grid i} \\ \text{0, otherwise} \end{cases}$$

S^2 = number of grids in the input image
B = number of bounding boxes generated by each grind
p = probability that the object is PPE and fire

Also, L_{Class} refers to the classification error which is usually expressed by Eq. (11). The L_{Class} corresponding to the ith grid is the summation of classification errors associated with all the objects within that grid where p_c is true class probability.

$$L_{Class} = \sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\{-\log(1 - p_c)\} \tag{11}$$

(vii) **F1 Score:** F1 Score is another most frequently used parameter for evaluating the performance of YOLOv4 algorithm. It is computed using Eq. (12).

$$F1\text{score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

The YOLOv4 algorithm disseminates the input image into a grid of cells of dimensions $\text{S} \times \text{S}$ wherein 3 bounding boxes can be predicted by each grid cell. Also, the bounding boxes are predicted by YOLOv4 at three distinct scales. Moreover, k-means clustering is utilized for the determination of bounding box priors. The present investigation uses 9 clusters and 3 scales. The selected clusters are consistently distributed among scales as $(17 \times 33), (39 \times 85), (105 \times 101), (59 \times 193), (121 \times 265), (265 \times 125), (206 \times 329), (359 \times 230), (370 \times 378)$.

4.2 YOLOv4-tiny

Although YOLOv4 performs exceptionally well in object detection task, but owing to its large structure size (137 layers), it is computationally expensive and not fast enough to run on embedded devices. However, YOLOv4-tiny has only 29 layers (combination of convolutional and pooling layers) in the backbone network. Consequently, it has comparatively small model size (<25 MB) with very high detection speed (~ 500 times faster) and reduced accuracy.

5 Simulation platform and parametric settings for training

The details of the simulation platform used in the present analysis have been presented in Table 2 along with their associated specific configurations. Notably, the simulation setting assembles the complete script in Visual Studio 2017.

Out of 14,500 images of the cleaned dataset, the training was performed on 80% of the images (11,600 images), the rest 20% (2,900 images) meant for testing and validation. The initial weight assignment at the starting phase of training was carried out using the pre-trained weights for the convolutional layers YOLOv4.conv.137 for YOLOv4 and yolov4-tiny.conv.29 for YOLOv4-tiny. In this work, a comparison of the performance of YOLOv4 and YOLOv4-tiny algorithms with previous works has been made. To accomplish this objective, the parameters specified in Table 3 were used for training.

For both algorithms (YOLOv4 and YOLOv4-tiny), the total number of iterations was set to 20,000. Initially, the learning rate was fixed at 0.001. However, after 16000 and 18000 iterations, the learning rate was divided by 10. The completion of training took approximately 80 hours (12 hours for YOLOv4-tiny and 68 hours for YOLOv4) on the specified simulation platform (Table 2). The entire training and testing was accomplished at the Computer Vision Research Laboratory in the Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab, India.

6 Results and discussion

As mentioned earlier, the training through YOLOv4 and YOLOv4-tiny algorithms have been carried out for 20,000 iterations in the system settings described in Table 2. The performance parameters (mAP, Average IoU, Precision, Recall and F1 score) have been

Table 2 Description of Simulation platform

Name	Related configuration
Operating System	Windows
CPU	Intel(R) Core (TM) i7-9700F CPU @ 3.00 GHz
RAM	8 GB
GPU	MSI Gaming GeForce GTX 1650
GPU acceleration library	CUDA10.0, CUDNN7.4

Table 3 Parameters of CFG used for training our model

Parameter	Value(s)	
	YOLOv4 Neural Network	**YOLOv4-tiny Neural Network**
Width	416	416
Height	416	416
Batch	64	64
Subdivisions*	64	32
Channels	3	-
Momentum	0.9	0.9
Decay	0.0005	0.0005
Learning rate	0.001	0.001
Maximum number of Batches*	20000	20000
Policy	Steps	Steps
Steps*	16000, 18000	16000,18000
Scale	0.1, 0.1	0.1, 0.1
Classes*	6	6
Filters*	$(4+1+6)\times 3=33$	$(4+1+6)\times 3=33$

Filters usually depend on the number of classes, bounding box properties, Prediction probability and the number of masks, *i.e.,* filters = {number of bounding box properties (4) + Prediction probability P_c (1) + Total number of classes (6)}× Number of mask, where mask denotes the indices of anchors (3)

*Represent the parameters modified in the original YOLOv4 CFG and YOLOv4-tiny CFG respectively

regularly monitored and evaluated during the training on an interval basis (at regular intervals of 1000 iterations). These performance parameters for YOLOv4 and YOLOv4-tiny algorithms have been presented in Table 4.

It is quite evident from Table 4 that the mAP settles at the best value of 76.86% for YOLOv4 algorithm after the completion of 20000 iterations. Conversely, the YOLOv4-tiny algorithm struggles at 54.19% of mAP even after 20000 iterations. Moreover, the variations in average loss and mAP values corresponding to the number of iterations in the training phase have been illustrated in Fig. 5a, b for YOLOv4 and YOLOv4-tiny algorithms respectively. The training results reveal that after 20000 iterations, YOLOv4 and YOLOv4-tiny algorithms yield an average loss of 2.6171 and 0.5240 respectively. Also, it can be observed that the mAP value obtained by YOLOv4 is 41.83% higher than that of YOLOv4-tiny (with respect to the best value). For comparative insight of YOLOv4 and YOLOv4-tiny (in terms of % mAP), the variations in mAP values with the number of iterations has been presented in Fig. 6.

In addition, Table 5 presents a summary of the present work in terms of mAP and detection speed (frames per second, i.e., FPS). The data presented in Table 5 reveals the effectiveness of YOLOv4 over the YOLOv4-tiny algorithm.

After the training has been accomplished, the validation of the test images on the trained model has been performed. Fig. 7 presents the experimental results of the test images for YOLOv4 and YOLOv4-tiny algorithms. Moreover, for quantitative

Table 4 Training results of YOLOv4 and YOLOv4-tiny algorithms on the test model

Iterations	YOLO version	mAP (%)	Average IoU (%)	Precision	Recall	F1 score
1000	v4	16.72	24.45	0.34	0.30	0.32
	v4-tiny	7.04	22.35	0.33	0.07	0.12
2000	v4	46.18	45.16	0.59	0.50	0.54
	v4-tiny	20.23	36.77	0.51	0.19	0.27
3000	v4	43.63	45.45	0.60	0.48	0.53
	v4-tiny	35.52	36.58	0.50	0.27	0.35
4000	v4	50.71	48.02	0.62	0.57	0.59
	v4-tiny	37.99	37.19	0.50	0.32	0.39
5000	v4	54.96	58.17	0.73	0.54	0.62
	v4-tiny	40.19	39.09	0.53	0.33	0.41
6000	v4	47.72	55.48	0.70	0.69	0.64
	v4-tiny	42.24	37.06	0.50	0.35	0.41
7000	v4	58.19	58.14	0.71	0.60	0.65
	v4-tiny	44.38	39.15	0.53	0.34	0.41
8000	v4	54.82	55.67	0.71	0.64	0.67
	v4-tiny	45.41	39.43	0.53	0.36	0.43
9000	v4	44.11	43.92	0.55	0.66	0.60
	v4-tiny	47.29	40.74	0.55	0.37	0.44
10000	v4	47.72	54.70	0.68	0.66	0.67
	v4-tiny	47.41	39.38	0.53	0.40	0.46
11000	v4	60.46	60.18	0.75	0.69	0.72
	v4-tiny	49.12	41.86	0.56	0.38	0.45
12000	v4	61.75	54.78	0.68	0.72	0.70
	v4-tiny	49.54	40.44	0.54	0.41	0.47
13000	v4	58.34	49.97	0.64	0.74	0.69
	v4-tiny	51.09	42.41	0.57	0.44	0.49
14000	v4	71.85	55.36	0.70	0.73	0.72
	v4-tiny	50.33	41.63	0.56	0.42	0.48
15000	v4	67.91	67	0.83	0.71	0.76
	v4-tiny	51.68	43.95	0.59	0.43	0.50
16000	v4	68.83	65.39	0.80	0.76	0.78
	v4-tiny	51.19	42.36	0.57	0.46	0.51
17000	v4	78.18	68.56	0.83	0.77	0.80
	v4-tiny	53.62	44.22	0.58	0.49	0.53
18000	v4	63.40	64.43	0.80	0.75	0.77
	v4-tiny	54.28	43.57	0.57	0.50	0.53
19000	v4	64.91	63.65	0.78	0.77	0.77
	v4-tiny	54.46	46.44	0.61	0.49	0.55
20000	v4	76.86	70.59	0.85	0.77	0.81
	v4-tiny	54.19	43.60	0.58	0.51	0.54
BEST*	v4	76.86	70.59	0.85	0.77	0.81
	v4-tiny	54.19	43.60	0.58	0.51	0.54

***BEST** represents the iteration for which maximum mAP has been observed during training

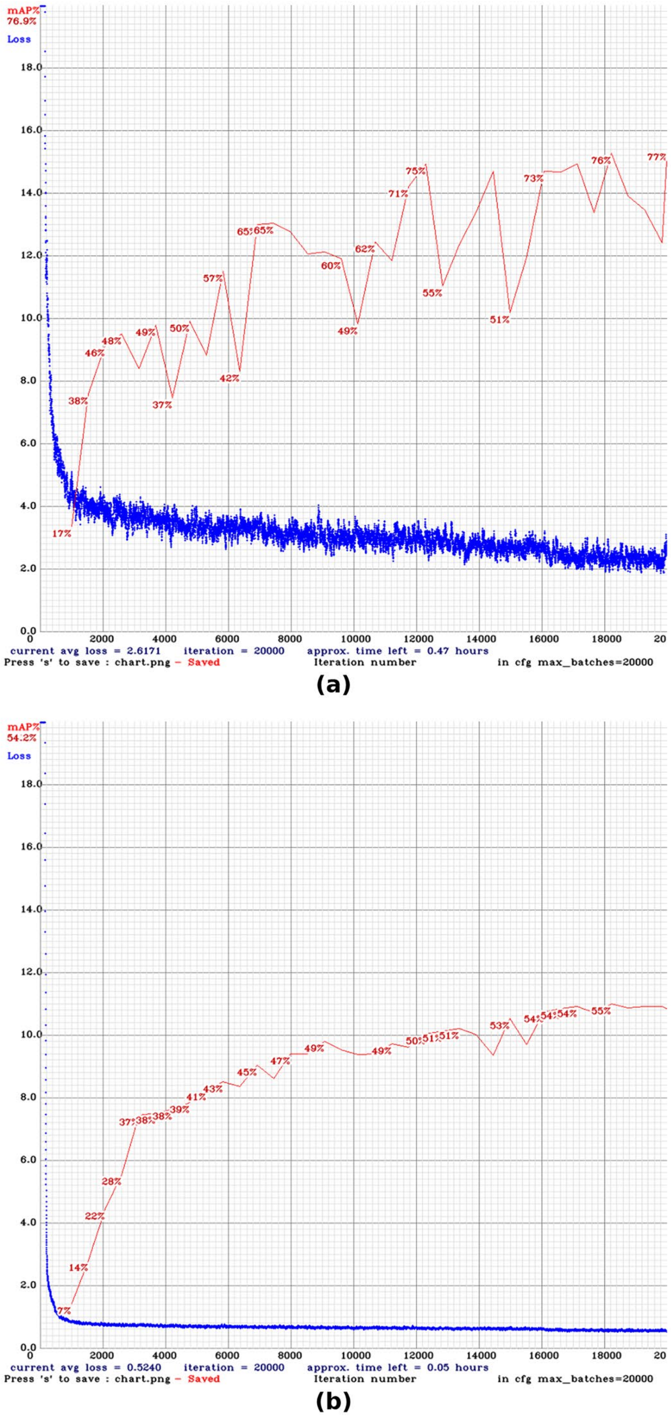


Fig. 5 Variations of loss function and mAP *w.r.t.* number of iterations for (**a**) YOLOv4 (**b**) YOLOv4-tiny

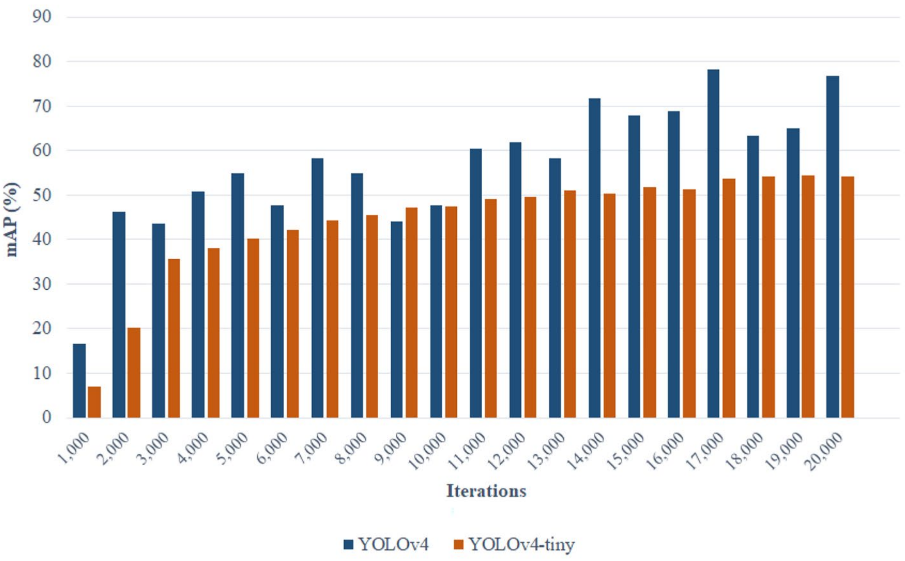


Fig. 6 Variation in mAP *w.r.t.* number of iterations for YOLOv4 and YOLOv4-tiny algorithms

assessment of the simulation results, a comparison has been made among YOLOv4 and YOLOv4-tiny algorithms in terms of prediction probability and prediction time.

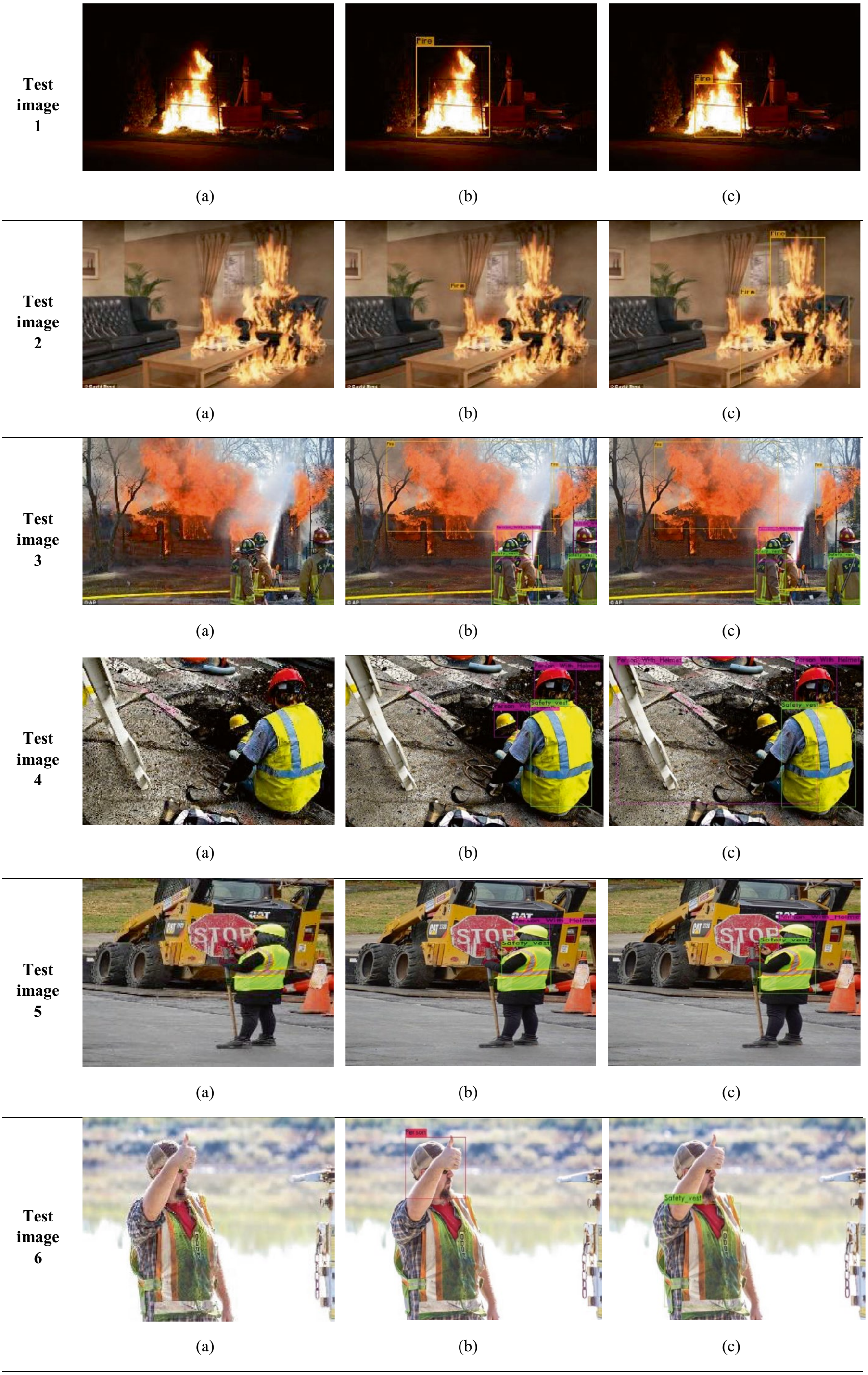
As evident from Fig. 7 and Table 6, all the test images (except test image 6) were correctly recognized by YOLOv4 with satisfactory prediction probability. In contrast, YOLOv4-tiny exhibits poor performance as it is incapable of recognizing some of the test images correctly. In addition, the test results presented in Table 6 confirm the poor computational speed of YOLOv4 over YOLOv4-tiny. It can be observed that the computational speed of YOLOv4-tiny is noteworthy. However, poor prediction probability and poor detection capability are observed (in case of YOLOv4-tiny). To compare the detection capabilities of YOLOv4 and YOLOv4-tiny, the total number of objects that were detected in the test images has been tabulated in Table 7.

For comparative analysis, the missed and false detection rates along with mAP values for YOLOv4 and YOLOv4-tiny algorithms have been presented in a tabular form (in Table 8). It can be observed that the detection rates in case of YOLOv4 are comparatively lower than YOLOv4-tiny.

Further, the obtained results for detection of PPE with fire have been compared with previous works (either PPE or fire) [2, 9, 16, 28] to validate the performance of the proposed

Table 5 Comparison of YOLOv4 and YOLOv4-tiny algorithms in terms of mAP and detection speed

Algorithm	mAP (%)	Detection Speed (FPS)	
		CPU (Intel(R) Core (TM) i5-4200U CPU @ 1.60GHz 2.30GHz)	On Specified Platform (Table 2)
YOLOv4	76.86	1	19.8
YOLOv4-tiny	54.19	5.3	109


Test image 1
(a)
(b)
(c)
Fire
Test image 2
(a)
(b)
(c)
Fire
Test image 3
(a)
(b)
(c)
Fire
Person_With_Helmet
Safety_vest
Test image 4
(a)
(b)
(c)
Person_With_Helmet
Safety_vest
Test image 5
(a)
(b)
(c)
Person_With_Helmet
Safety_vest
Test image 6
(a)
(b)
(c)
Person
Safety_vest

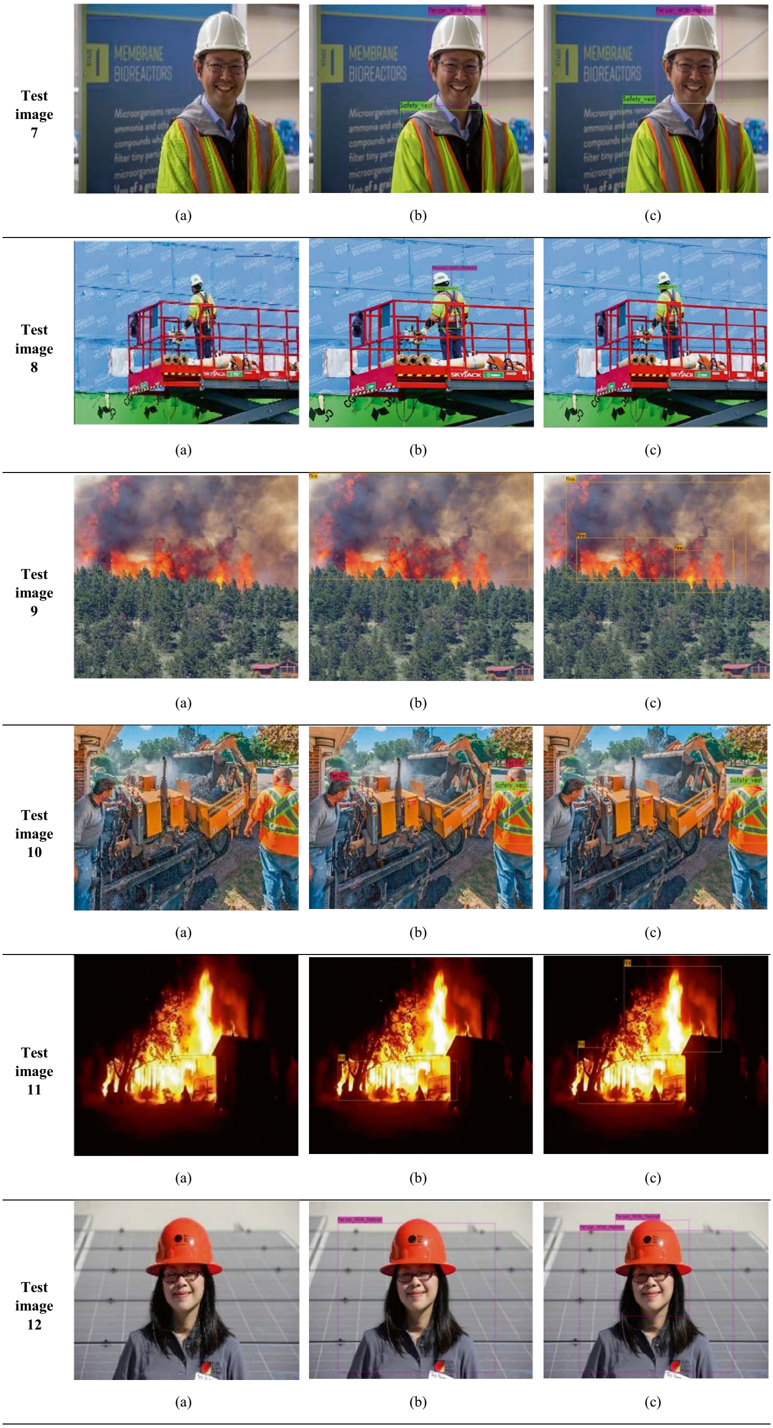


Fig. 7 Experimental results on sample test images for PPE and fire detection tasks (**a**) Original image (**b**) Detection by YOLOv4 (**c**) Detection by YOLOv4-tiny

Table 6 Quantitative comparison of the Experimental results of the test model

Test Image	Name	Prediction probability (in %)		Prediction Time (in ms)	
		YOLOv4	YOLOv4-tiny	YOLOv4	YOLOv4-tiny
1	Fire	98	64	145.75	10.56
2	Fire	99	80	145.87	10.76
3	Fire	99	26	150.77	11.12
	Fire	98	38		
	Person_With_Helmet	100	67		
	Person_With_Helmet	100	No detection		
	Safety vest	89	56		
	Safety vest	87	64		
4	Person_With_Helmet	99	54	149.56	11.10
	Person_With_Helmet	98	No detection		
	Safety vest	87	54		
5	Person_With_Helmet	100	68	148.75	11.05
	Safety vest	98	57		
6	Person	100	No detection	148.59	11.03
	Safety vest	No detection	24		
7	Person_With_Helmet	100	68	147.97	11.01
	Safety vest	98	29		
8	Person_With_Helmet	87	No detection	148.04	11.04
	Safety vest	92	24		
9	Fire	100	84	149.89	11.13
			False detection		
			False detection		
10	Person	100	No detection	149.68	11.12
	Person	100	No detection		
	Safety Vest	84	26		
11	Fire	95	20	147.87	10.99
			21		
12	Person_With_Helmet	100	58	147.93	11.01
			False detection		

methodology. This comparison has been summarized in Table 9. Reported literature reveals that although [9, 16] and [2] achieve high precision, but they can only detect either (i) persons with helmet, or (ii) person, or (iii) fire. Moreover, most of the training and testing images were randomly fetched from consecutive video frames. This does not yield fruitful results with practical implications. In contrast, the present work utilizes images that have been taken from a variety of sources, devices, at different locations, perspectives, and times. Therefore, the performance of the proposed methodology reflects the most probable performance, especially on unseen images. Table 9 also reveals that previous works are unable to provide real-time speed. However, the present work ensures 19.8 FPS and 109 FPS with YOLOv4 and YOLOv4-tiny respectively. Moreover, to distinguish between fire and look-a-like events, the previous works require video frames. In contrast, the present work is applicable to both still images and video frames. Therefore, it has greater generalization capability and adaptability for practical usage.

Table 7 Comparison of detection capability

Test Image	Ground Truth	Objects Detected	
		YOLOv4	YOLOv4-tiny
1	1	1	1
2	1	1	2
3	6	6	5
4	3	3	2
5	2	2	2
6	2	1	1
7	2	2	2
8	2	2	1
9	1	1	3
10	3	3	1
11	1	1	2
12	1	1	2

Table 8 Comparison of detection rates and mAP for YOLOv4 and YOLOv4-tiny algorithms

Algorithm	Missed Detection Rate (%)	False Detection Rate (%)	mAP (%)
YOLOv4	0	0.68	76.86
YOLOv4-tiny	8.47	19.75	54.19

Table 9 Performance of different PPE and fire detection models

S. N.	Authors	Methodology	Class						FPS
			PWH	Person	Fire	Safety Vest	Fire Hydrant	Safety Glasses	
1.	Nath et al. [28]	YOLOv3	73.97%	x	x	72.3%	x	x	11
2.	Fang et al. [9]	Faster R-CNN	>90%	x	x	x	x	x	10
3.	Balakreshnan et al. [2]	Azure Custom Vision AI	x	>90%	x	x	x	50%	x
4.	Li and Zhao [16]	YOLOv3	x	x	>90%	x	x	x	28
5.	Proposed*	YOLOv4	84.93%	81.15%	67.72%	98.41%	52.52%	76.45%	19.9
6.	Proposed*	YOLOv4-tiny	46.28%	32.71%	44.66%	83.83%	100.00%	22.20%	109

PWH Person_With_Helmet and ‘x’ represents unavailability of data

7 Conclusions

This work presents a maiden attempt towards the detection of PPE and fire at construction sites for overall assessment of occupational safety. For the real-time detection of PPE and fire, YOLOv4 and YOLOv4-tiny algorithms have been employed. The experimental results prove the efficiency of YOLOv4 algorithm over YOLOv4-tiny algorithm in terms of prediction probability. It is observed that YOLOv4 offers mAP of 76.86%. This value is 41.83% higher than the best mAP obtained by YOLOv4-tiny. Clearly, the results recommend the development of smart surveillance systems based on YOLOv4 algorithm for PPE and fire direction to reduce construction site casualties. However, the present application needs to be trained on a large dataset to get more generalized and robust model. This may be addressed in future works.

References

1. Akbar-Khanzadeh F (1998) Factors contributing to discomfort or dissatisfaction as a result of wearing personal protective equipment. J Hum Ergol (Tokyo) 27:70–75
2. Balakreshnan B, Richards G, Nanda G et al (2020) PPE Compliance Detection using Artificial Intelligence in Learning Factories. Procedia Manuf 45:277–282. https://doi.org/10.1016/j.promfg.2020.04.017
3. Barro-Torres S, Fernández-Caramés TM, Pérez-Iglesias HJ, Escudero CJ (2012) Real-time personal protective equipment monitoring system. Comput Commun 36:42–50. https://doi.org/10.1016/j.comcom.2012.01.005
4. Bhole SA (2016) Safety Problems and Injuries on Construction Site: A Review. Int J Eng Tech 2:24–35
5. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934
6. Chen RC (2019) Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. Image Vis Comput 87:47–56. https://doi.org/10.1016/j.imavis.2019.04.007
7. Ding L, Fang W, Luo H et al (2018) A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. Autom Constr 86:118–124. https://doi.org/10.1016/j.autcon.2017.11.002
8. Dundar A, Jin J, Martini B, Culurciello E (2017) Embedded streaming deep neural networks accelerator with applications. IEEE Trans Neural Netw Learn Syst 28:1572–1583. https://doi.org/10.1109/TNNLS.2016.2545298
9. Fang Q, Li H, Luo X et al (2018) Detecting non-hardhat-use by a deep learning method from far-field surveillance videos. Autom Constr 85:1–9. https://doi.org/10.1016/j.autcon.2017.09.018
10. Hassaballah M, Awad AI (2020) Deep Learning in Computer Vision. CRC Press
11. Karthik R, Hariharan M, Anand S et al (2020) Attention embedded residual CNN for disease detection in tomato leaves. Appl Soft Comput 86:105933. https://doi.org/10.1016/j.asoc.2019.105933
12. Kelm A, Laußat L, Meins-Becker A et al (2013) Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. Autom Constr 36:38–52. https://doi.org/10.1016/j.autcon.2013.08.009
13. Kolar Z, Chen H, Luo X (2018) Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images. Autom Constr 89:58–70. https://doi.org/10.1016/j.autcon.2018.01.003
14. Kumar S, Yadav D, Gupta H et al (2020) A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management. Electronics 10:14. https://doi.org/10.3390/electronics10010014
15. Lee D-H (2020) CNN-based single object detection and tracking in videos and its application to drone detection. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-09924-0
16. Li P, Zhao W (2020) Image fire detection algorithms based on convolutional neural networks. Case Stud Therm Eng 19. https://doi.org/10.1016/j.csite.2020.100625
17. Luo Y, Zhao L, Liu P, Huang D (2018) Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. Multimed Tools Appl 77:15075–15092. https://doi.org/10.1007/s11042-017-5090-2
18. Man-Woo P, Nehad E, Zhenhua Z (2015) Hardhat-Wearing Detection for Enhancing On-Site Safety of Construction Workers. J Constr Eng Manag 141:4015024. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000974
19. Mao W, Wang W, Dou Z, Li Y (2018) Fire Recognition Based On Multi-Channel Convolutional Neural Network. Fire Technol 54:531–554. https://doi.org/10.1007/s10694-017-0695-6
20. Mneymneh BE, Abbas M, Khoury H (2017) Automated Hardhat Detection for Construction Safety Applications. Procedia Eng 196:895–902. https://doi.org/10.1016/j.proeng.2017.08.022
21. Mneymneh BE, Abbas M, Khoury H (2019) Vision-Based Framework for Intelligent Monitoring of Hardhat Wearing on Construction Sites. J Comput Civ Eng 33:1–20. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000813

22. Muhammad K, Ahmad J, Mehmood I et al (2018) Convolutional Neural Networks Based Fire Detection in Surveillance Videos. IEEE Access 6:18174–18183. https://doi.org/10.1109/ACCESS.2018.2812835
23. Muhammad K, Ahmad J, Baik SW (2018) Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing 288:30–42. https://doi.org/10.1016/j.neucom.2017.04.083
24. Muhammad K, Khan S, Baik SW (2020) Efficient Convolutional Neural Networks for Fire Detection in Surveillance Applications. https://books.google.com https://doi.org/10.1201/9781351003827-3
25. Namozov A, Cho YI (2018) An efficient deep learning algorithm for fire and smoke detection with limited data. Adv Electr Comput Eng 18:121–128. https://doi.org/10.4316/AECE.2018.04015
26. Naticchia B, Vaccarini M, Carbonari A (2013) A monitoring system for real-time interference control on large construction sites. Autom Constr 29:148–160. https://doi.org/10.1016/j.autcon.2012.09.016
27. Nath ND, Chaspari T, Behzadan AH (2019) Single- And multi-label classification of construction objects using deep transfer learning methods. J Inf Technol Constr 24:511–526. https://doi.org/10.36680/J.ITCON.2019.028
28. Nath ND, Behzadan AH, Paal SG (2020) Deep learning for site safety: Real-time detection of personal protective equipment. Autom Constr 112:103085. https://doi.org/10.1016/j.autcon.2020.103085
29. Nie X, Yang M, Liu RW (2019) Deep Neural Network-Based Robust Ship Detection Under Different Weather Conditions. In: 2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019. Inst Electr Electr Eng Inc 47–52
30. Park M-W, Brilakis I (2012) Construction worker detection in video frames for initializing vision trackers. Autom Constr 28:15–25. https://doi.org/10.1016/j.autcon.2012.06.001
31. Rangel JC, Martínez-Gómez J, Romero-González C et al (2018) Semi-supervised 3D object recognition through CNN labeling. Appl Soft Comput 65:603–613. https://doi.org/10.1016/j.asoc.2018.02.005
32. Redmon J, Farhadi A (2018) YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767
33. Seo J, Han S, Lee S, Kim H (2015) Computer vision techniques for construction safety and health monitoring. Adv Eng Inform 29:239–251. https://doi.org/10.1016/j.aei.2015.02.001
34. Seong H, Son H, Kim C (2018) A Comparative Study of Machine Learning Classification for Color-based Safety Vest Detection on Construction-Site Images. KSCE J Civ Eng 22:4254–4262. https://doi.org/10.1007/s12205-017-1730-3
35. Siddula M, Dai F, Ye Y, Fan J (2016) Unsupervised Feature Learning for Objects of Interest Detection in Cluttered Construction Roof Site Images. Procedia Eng 145:428–435. https://doi.org/10.1016/j.proeng.2016.04.010
36. Sun L, Zhao C, Yan Z et al (2019) A novel weakly-supervised approach for RGB-D-based nuclear waste object detection. IEEE Sens J 19:3487–3500. https://doi.org/10.1109/JSEN.2018.2888815
37. Tran Q-H, Le T-L, Hoang S-H (2019) A fully automated vision-based system for real-time personal protective detection and monitoring. KICS Korea-Vietnam Int Jt Work Commun Inf Sci 2019:1–6
38. Wu J, Cai N, Chen W et al (2019) Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. Autom Constr 106:102894. https://doi.org/10.1016/j.autcon.2019.102894
39. Wu D, Lv S, Jiang M, Song H (2020) Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Comput Electron Agric 178:105742. https://doi.org/10.1016/j.compag.2020.105742
40. Yin Z, Wan B, Yuan F et al (2017) A Deep Normalization and Convolutional Neural Network for Image Smoke Detection. IEEE Access 5:18429–18438. https://doi.org/10.1109/ACCESS.2017.2747399

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.