Expert Systems With Applications 225 (2023) 120096

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Verification and performance comparison of CNN-based algorithms for two-step helmet-wearing detection

Ju-Yeon Lee [1], Woo-Seok Choi, Sang-Hyun Choi [*]

Department of BigData, Chungbuk National University, Cheongju 28644, Republic of Korea

ARTICLE INFO

Keywords:
YOLO
EfficientNet
Object detection
Image classification
Deep learning
Construction safety

ABSTRACT

Workplace accidents are on the rise, and the construction and manufacturing sectors account for more than half of all fatalities. In the construction and industrial sectors, most fatal accidents happen in small-scale workplaces with inadequate protective equipment. Wearing personal protective equipment, such as safety helmets, is critical for alleviating these issues. Many methods based on object-detection models have recently been proposed to assess whether safety helmets are worn; however, most of them train existing models to focus only on the wearing or non-wearing of safety helmets and do not determine whether a hat is worn instead of a safety helmet to cover the head. This study split the data into helmet, head, and hat classes, and trained the model to classify employees wearing hats. The model was built in two stages to attain high accuracy with a small amount of training data. The human head was detected in the image using the object-detection model in the first stage. In the second stage, the position of the human head identified in the first step was used to categorize it into three classes using a classification model: helmet, head, and hat. The performances of Faster R-CNN, RetinaNet, and YOLOv5 models were compared. The F1-score of the YOLO-EfficientNet model was 3.2–16.4% higher than those of the other models. The proposed method can help determine whether employees are wearing safety helmets and is expected to assist in the prevention of workplace accidents.

1. Introduction

Many policies and measures have been implemented across the globe to prevent workplace accidents (UNISDR, 2015; Ministry of Employment and Labor, 2004; Blanchard et al., 2022; EHS Today, 2022; Korea Occupational Safety and Health Agency, 2020). Similarly, policies to reduce industrial accidents have continually been implemented in South Korea (Ministry of Employment and Labor, 2021); however, the fatality rate[2] has been rising steadily since 2017 (Korea Occupational Safety and Health Agency, 2021b). According to the 2020 industrial accident analysis report of the Ministry of Employment and Labor, "the present number of occupational accidents and deaths" was 882, an increase of 27 (3.16%) from 855 in the previous year. In the occupational accident status survey conducted from January–September 2021, out of a total of 1,635 industrial fatalities, 850 deaths occurred in the construction and manufacturing sectors, accounting for 52% of the mortality rate by industry (Korea Occupational Safety and Health Agency, 2021a). A total of 66% of fatal accidents in the high-risk construction and industrial sectors occurred in workplaces with 5 to 49 people, accounting for a sizable proportion of all accidents (Korea Occupational Safety and Health Agency, 2021a).

To determine the cause of accidents, the Occupational Safety and Health Research Institute conducted a survey of employees in small-scale workplaces[3] in November 2012. The findings showed that self-protection by wearing protective equipment was not complied with in 76% of the cases, which was remarkably high (Shinwon et al., 2012). Personal protective equipment (PPE) is the clothing worn on all or part of a worker's body to block or reduce the impact of external risks and irritants. PPE must provide comprehensive protection against hazardous and dangerous objects, as well as being of high quality in terms of materials, construction, and processing (Korean Industrial Health Association, 2004).

Effective ways to prevent accidents in high-risk sectors generally include screening for safety hazards, providing PPE equipment, ensuring

* Corresponding author.
E-mail addresses: yeony_yy@chungbuk.ac.kr (J.-Y. Lee), cdt3017@naver.com (W.-S. Choi), chois@chungbuk.ac.kr (S.-H. Choi).
[1] ORCID: 0000-0001-6087-2001.
[2] Fatality rate: Number of deaths per 10,000 workers.
[3] Construction project sites of KRW 2 billion or less.

https://doi.org/10.1016/j.eswa.2023.120096
Received 3 January 2023; Received in revised form 23 March 2023; Accepted 8 April 2023
Available online 12 April 2023
0957-4174/

everyone wears suitable apparel, being vigilant on the job, and promoting safety awareness and education (Foulis, 2021). Among these accident prevention approaches, wearing PPE is the main focus of this study. PPE refers to safety equipment, and the safety helmet is confined to mitigating the dangers posed by falling items, electric shock, or shock (Korean Industrial Health Association, 2004). Conventional hats are considered unsafe because they do not qualify as safety helmets.

However, significant amount of time and effort are required for safety managers to evaluate workers' job assignments and conditions (Heesung et al., 2021), while the supply of safety managers is decreasing over time (EKU Online, 2020; Juyoung, 2021). In such circumstances, it is almost impossible for a safety manager to monitor whether all personnel identify dangerous circumstances and wear safety helmets (Song et al., 2017). As a result, the government launched the development of smart safety integrated control technology in 2020, with the objective of lowering construction site accidents by 25% by 2025. It is composed of safety technology for construction site workers, smart safety technology for temporary structures, and a smart safety integrated control system that combines and connects all these components. Since then, the government has announced a strategy to implement "Smart Safety Management" for private small and medium-sized construction sites, as well as private old and risky buildings, which are deemed safety management blind spots (Smart City Korea, 2021). A pilot project for 50 private construction sites was launched in February 2022 (Ministry of Land, Infrastructure, and Transport, 2022). Smart Safety Management is an intelligent monitoring system that detects unsafe circumstances by analyzing CCTV footage from private construction sites where AI is deployed in most sites. This system may be extended and applied to any situation that needs safety management as a preventative measure against potentially dangerous situations.

However, the safety helmet recognition technology for smart safety management was built based on the Faster R-CNN of the two-stage detector. Therefore, the algorithm speed is slow, making it difficult to use it in real-time, and there are limitations to optimizing the model owing to insufficient safety helmet datasets (Seoul Digital Foundation, 2022), which also affects most fields where object detection is required; many images are required to train the model. The difficulty in training object-detection deep-learning models is the scarcity of AI training datasets that are freely available (Perez & Wang, 2017).

We propose a safety-helmet-detection algorithm, YOLO-EfficientNet, that is capable of real-time performance with a small amount of training data to ensure the safety of construction workers. The algorithm is composed of two stages: first, a YOLO model is used to detect the object corresponding to a human head, and second, the safety helmet-wearing status of the detected object is classified using EfficientNet. In addition, we aimed to verify the superiority of the YOLO-EfficientNet model by comparing its performance with existing object-detection models, including RetinaNet (Lin et al., 2018) and YOLOv5 (Ge et al., 2021), as well as Faster R-CNN (Ren et al., 2016) used in the "Smart Safety Management" technology mentioned earlier. YOLO-EfficientNet is based on the one-stage detector YOLO, which is high algorithm speed. By dividing object detection into two simple problems and applying ensemble methods, we expect to achieve high accuracy by combining object detection and classification models. Furthermore, we anticipate that this algorithm can be quickly optimized for situations where monitoring of personal protective equipment use, in addition to helmets, is required.

The remainder of this paper is organized as follows: Section 2 discusses previous studies related to detecting safety-helmet-wearing status in the workplace. Section 3 describes the research methodology used in this study. Section 4 explains the overall design of the proposed model. Section 5 evaluates the model and compares its performance with other image detection models. Finally, Section 6 summarizes the study, draws conclusions, and discusses future research.

2. Related work

2.1. Deep learning-based object-detection model

In general, deep learning-based object detection can be divided into one-stage and two-stage detectors, depending on how they perform classification and regional proposal to find the area where objects are likely to exist (Lu et al., 2020). One-stage detectors simultaneously perform classification and regional proposal to obtain results, while two-stage detectors perform classification and regional proposal sequentially. One-stage detectors are relatively faster but less accurate, while two-stage detectors are relatively slower but more accurate (Elgendy, 2020). Therefore, algorithms are selected based on the purpose of object detection.

There has been significant progress in both methods of deep learning-based object detection. In 2015, a CNN-based deep-learning algorithm, ResNet, showed a performance of 97.7%—which was higher than the 95% human accuracy— during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), accelerating deep-learning research (Pak & Kim, 2017).

Deep learning-based object detection was first studied with a two-stage detector, which has mainly evolved into region-based convolutional neural networks (R-CNNs) structure (Du et al., 2020). R-CNN is a significant model that demonstrated the potential of CNN-based detection not only in classification but also in object detection (Mittal et al., 2022). R-CNN generates region proposals for images using selective search, and each generated region proposal is wrapped into a fixed size and used as the input for the CNN. The feature map obtained from the CNN is used for classification via a support vector machine and for adjusting the bounding box through a regressor to detect objects (Gkioxari et al., 2015).

Regarding two-stage detectors, researchers have been improving and advancing the R-CNN-based structure by combining it with other models. Fast R-CNN and Faster R-CNN, which were introduced in 2015, replaced the R-CNN pooling layer with a region-of-interest (RoI) pooling layer that could reduce the RoI of the feature map to a fixed-length low-dimensional vector using max-pooling. This change significantly improved the learning speed of the model by over nine times (Girshick, 2015; Ren et al., 2016). In 2017, Mask R-CNN was introduced by adding a mask branch to Faster R-CNN to simultaneously process classification, Bbox regression, and predicting the object mask (He et al., 2020). Mask R-CNN also replaced the RoI pooling layer with an RoI align layer that could preserve the spatial location of objects. The new algorithm was easier to train and faster (5 fps) than the previous models (Su et al., 2020). Granulated R-CNN (G-RCNN), which is an improved version of Fast R-CNN and Faster R-CNN, was recently developed. G-RCNN integrates the unique granulation concept in deep convolutional neural networks and implements a new CNN architecture called G-AlexNet, which considerably improves object-detection accuracy (Pramanik et al., 2022). Although G-RCNN is slightly slower (5.5 fps) than Faster R-CNN (6 fps), it has shown an accuracy improvement of more than 1.34 times in terms of mean average precision (mAP). Despite the continuous development of two-stage detectors, there is a limitation in algorithm speed, which generally does not exceed 7 fps. Therefore, recent trends are moving toward one-stage detectors.

One-stage detector algorithms, namely single shot multibox detector (SSD) and you only look once (YOLO), have been researched and improved upon since their introduction in 2016. SSD is an object-detection model composed of two elements: a multi-scale feature layer and default box. The algorithm applies convolution multiple times to produce multiple feature maps, performs object detection on each feature map, and integrates the results to generate the final detection (Liu et al., 2016). One of the main advantages of SSD is its speed, as it can process a 300x300 image at 59 fps, which is close to real-time performance.

YOLO was the first model to introduce the one-stage detector approach and enable real-time object detection (Redmon et al., 2016). Unlike SSD, which divides an image into multiple sub-images for interpretation, YOLO reconstructs all image pixels, bounding boxes, coordinates, and class probabilities as a single regression problem, allowing the algorithm to scan the entire image only once. This feature results in an even faster algorithm compared to SSD (Redmon et al., 2016).

Redmon and Farhadi (2018) released YOLOv3 as an open-source model. Since then, numerous AI researchers have attempted to improve the hidden layers of YOLO. As a result, YOLO has rapidly evolved and improved, with YOLOv4 and YOLOv5 in 2020, YOLOv6 and YOLOv7 in 2022 (Bochkovskiy et al., 2020; Jocher, 2022; Li et al., 2022; Wang et al., 2022). Moreover, specialized versions such as Fast YOLO and YOLO-LITE have been developed to enable deployment on mobile environments (Shafiee et al., 2017; Huang et al., 2018).

However, compared to two-stage detectors, YOLO and SSD have approximately 10–40% lower accuracy because of the extreme class imbalance problem that occurs in the one-stage detector method. Therefore, in this study, we devised a YOLO-EfficientNet model to address the class imbalance problem and improve accuracy. We evaluated and verified the performance of the YOLO-EfficientNet model by comparing it with existing research models. YOLO-EfficientNet is based on YOLO, which is a one-stage detector, and has the advantage of a high algorithm speed. By dividing object detection into two simple problems and applying an ensemble method that combines object detection and classification models, high accuracy can be achieve. Additionally, since its approach to the problem is simple, it can achieve sufficient model learning with a small amount of data.

2.2. Hardhat detection

The task of checking whether workers are wearing safety equipment in a workplace is the simplest and most effective way to prevent accidents. However, it is difficult for safety managers to monitor all workers in real-time, which often results in noncompliance with safety regulations (Ahn et al., 2023). Therefore, many researchers have been studying various image-processing methods to manage the use of safety equipment by workers. Recently, the use of deep learning, which automatically extracts and learns the characteristics of safety equipment in images, has become popular (Nath et al., 2020). Thus, we review related deep learning-based hardhat detection studies (see Table 1).

Deep learning-based object-detection research focusing on hardhat detection is closely related to workplace accident prevention, which has led many studies to aim at improving model accuracy (Wójcik et al., 2021; Chen & Su, 2021). Wójcik et al. (2021) developed a ResNet R-CNN-based model using a two-stage detector to improve hardhat classification performance. They designed a rule-based inference algorithm using keypoint representations of the human head to enhance object-detection performance and compared the performance of ResNet-50, ResNet-101 (He et al., 2016), and ResNeXt-101 (Xie et al., 2017) networks. They used 7,035 data samples from Xie (2019) and achieved an AP of 71% for hardhat wearers and 64.1% for non-wearers according to MS COCO's class-based standard (Lin et al., 2014), with an mAP of 67.5%.

Table 1
Overview of deep-learning research on hardhat detection.

Type	Publications	Method	Key Point
Two-Stage Detector	Wójcik et al. (2021)	ResNet R-CNN	head keypoint localization simple rule-based reasoning
	Chen and Su (2021)	Mask R-CNN	Soft-NMS
One-Stage Detector	Li et al. (2020)	SSD-MobileNet	MobileNet
	Iannizzotto et al. (2021)	YOLOv5	fuzzy logic filtering
	Han et al. (2021)	SSD	cross-layer attention mechanism
	Xu et al. (2022)	YOLOv5	squeeze-and-excitation block
	Cheng et al. (2021)	SAS-YOLOv3-Tiny	sandglass-residual module and SPP module combined

Chen and Su (2021) proposed an improved Mask R-CNN algorithm to address the issue of decreased performance in hardhat detection caused by complex backgrounds, low image quality, target occlusion, and small targets in industrial environments. The proposed algorithm is based on the Mask R-CNN Network but increases the number of anchor frames and uses Soft-NMS instead of non-maximum suppression (NMS) to improve pixel-level contour detection performance and increase object-detection accuracy. NMS removes duplicate bounding boxes using an intersection over union (IoU) threshold, whereas Soft-NMS reduces confidence instead of removing bounding boxes (Bodla et al., 2017). The improved Mask R-CNN model achieved very high performance with mAP of 97.1%, precision of 96.0%, and recall of 97.4%.

However, research models based on two-stage detectors have the limitation of reflecting real-time performance compared with deep-learning models. In workplace object detection using deep learning, it is important to analyze videos in real-time and provide immediate results; therefore, several studies have focused on increasing algorithm speed (Li et al., 2020; Iannizzotto et al., 2021; Han et al., 2021; Xu et al., 2022; Cheng et al., 2021). Li et al. (2020) designed a base network of SSD as a MobileNet structure to improve the limitations of SSD, which slows down the learning speed as the operations increase. MobileNet is a lightweight network designed using the computational efficiency of depthwise separable convolution and has the advantage of being fast enough to run on mobile devices (Howard et al., 2017). The hardhat detection performance of the SSD-MobileNet proposed in the study achieved a precision of 95% and a recall of 77%.

Iannizzotto et al. (2021) proposed a real-time PPE detection framework based on a YOLOv5 algorithm that applies to embedded platforms. The proposed framework combined fuzzy logic filtering with a YOLO-based object-detection algorithm to enhance the speed of the algorithm. Fuzzy logic is a logical concept that expresses the uncertain or ambiguous state beyond the binary logic of true or false with multi-valued logic (Chowdhury et al., 2016). Fuzzy logic filtering is an image-processing technique that applies the degree of noise possibility in each RGB channel image of the image to fuzzy logic, which is used as a basis for the location of the mask source. However, the trade-off between real-time performance and object-detection accuracy could not be resolved, and the object-detection performance decreased compared to that of other models.

Object-detection deep-learning algorithms often sacrifice accuracy for speed. To address the issue of low accuracy in existing safety-helmet-detection methods, Han et al. (2021) proposed a novel object-detection algorithm based on SSD. This algorithm employs a spatial attention mechanism for low-level features and a channel attention mechanism for high-level features. The cross-layer attention mechanism further refines the feature information of the object region. The proposed algorithm achieved an mAP of 80.5% on the VOC 2007 dataset, 3.4% better than the baseline.

Xu et al. (2022) proposed a YOLOv5-based real-time hardhat detection framework that aimed to achieve both high speed and accuracy by incorporating the squeeze-and-excitation (SE) block. The proposed method obtains the weight of image channels and accurately separates the foreground and background of the image, significantly improving the performance. The mAP of the proposed algorithm with the SE block was 93.6%, which was 2–2.5% higher than that of the original YOLOv5 and the baseline algorithm with no SE block, which achieved an mAP of 91.2%.

Most hardhat detection studies approach the problem as a binary classification task, aiming to identify whether a safety helmet is being worn or not. However, in real-world settings, there may be various classes other than safety helmets, such as hats or headscarves. Therefore, binary classification approaches may lead to errors in detecting hats as safety helmets in work environments. To address this issue, Cheng et al. (2021) proposed a hardhat detection model capable of classifying four categories: helmet, cap, nowear, and safety-cap. Furthermore, they aimed to solve the

Fig. 1. A Sample of Roboflow, Kaggle, Google Image datasets used in the experiment.

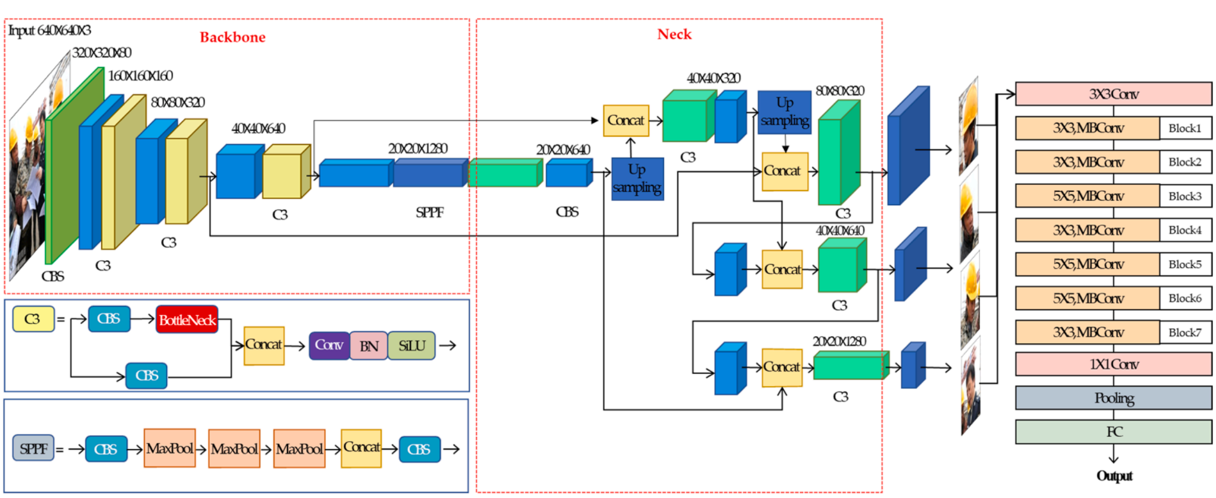
problem of YOLOv3-Tiny, which is small and easy to apply to embedded devices but had difficulty detecting multi-scale safety helmets. The proposed SAS-YOLOv3-Tiny combined sandglass-residual and SPP modules to balance the detection accuracy and speed, using an improved loss function. Training and validation were conducted using 7,656 crawled images, and the model achieved 71.6%, 80.9%, 80.3%, and 75.2%. in precision, recall, mAP, and F1, respectively.

The model was trained to detect the safety helmet by altering the backbone or using existing techniques. Current segmentation and object-detection algorithms can be used to effectively determine whether a safety helmet is worn. However, in most studies, only two cases have been identified: workers wearing or not wearing safety helmets, and workers wearing hats were not detected (Li et al., 2018; Dalal & Triggs, 2005; Cheng et al., 2021; Xu et al., 2022). A model trained on two classes identified a hat (with a similar shape and color) as a helmet (Dalal & Triggs, 2005). If this model is applied to a real-world workplace, it will detect workers who are wearing hats as wearing safety equipment. In fact, even in the dataset used in the experiment by this researcher, there were several workers wearing hats; thus, accidents can occur if class categorization is wrong. Therefore, this study focuses on designing a YOLO-EfficientNet model that includes hats in the classes, in addition to head and helmet, to meet the needs of real-world work environments.

3. Materials and methods

3.1. Overview

Fig. 2 shows the architecture of the proposed YOLO-EfficientNet for



(a) **(b)**

Fig. 2. Yolo-EfficientNet network structure, (a): Yolov5x architecture, (b): EfficientNetB7 baseline.

Table 2
Model used at each step, input data, output data, and tasks.

Step	Used Model	Input Data	Output Data	Task
1	YOLOv5x	Original image	Location of human head	Detection of human head
2	EfficientNetB7	Location of human head	Predicted class	Classification of head, hat, helmet

helmet detection. Unlike conventional object-detection deep-learning architecture, this architecture combines two deep-learning techniques, object detection and classification, in an ensemble-like manner. The architecture is divided into two steps: object detection, which detects the head of a person using a YOLOv5x algorithm, and classification, which predicts whether the detected head is wearing a helmet, a hat, or nothing using an EfficientNetB7 algorithm (see Table 2). The reason for dividing the problem into two stages is to simplify the multi-class object-detection problem into a single-class object-detection problem, reduce the amount of data required and computation for training, and improve model performance by applying a CNN-based classification model with good performance. Compared to existing models, the proposed YOLO-EfficientNet model can effectively train even with limited data, making it suitable for fields with limited publicly available AI training datasets. Additionally, by dividing a difficult problem into two simple problems, it is expected that high accuracy will be achieved. Unlike previous studies that classified only helmets and heads, this model can classify three classes: helmets, hats, and heads, making it suitable for use in work environments. To design this model, head images for detecting the head, helmet, and hat images for classifying PPE were collected and preprocessed.

3.2. Data collection and preprocessing

The Hard Hat Workers dataset from Roboflow (Northeastern University – China, 2022) and the Safety Helmet dataset from Kaggle (Larxel, 2020) were used for head and helmet training and validation. For hat training and validation, data obtained by crawling Google Images were used (Fig. 1). The dataset provided by Roboflow and Kaggle consists of images in JPG and PNG formats, respectively, and annotation files in XML format. The annotation file is labeled with three classes: helmet, head, and person, and since the person class is a labeling of the entire person, it was deleted because it was not necessary for this study that uses only the head of the person. In the first step, all two classes were changed to heads to detect human heads, and in the second step of classifying helmets, heads, and hats, the classes of helmet and head provided by Roboflow and Kaggle were used to train the EfficientNetB7 model. Regarding the hat class additionally collected using Google image, the image of the person wearing the hat in the workplace was collected as shown in Fig. 1 (third line), and the head wearing the hat was labeled hat using LabelMe.

For both Step 1 (YOLOv5x) and 2 (EfficientNet), a data split ratio of 8:1:1 was used to extract the training set, validation set, and test set for data validation. To use the collected 21,568 data samples, the same data was used for detection in Step 1 and classification in Step 2. To prevent the model from being trained on the test data in the first step, 2,250 data samples were removed from the training in Step 2 and the test data was chosen at random from the pool of 2,250 data samples (Table 3).

In Step 1, a total of 17,068 data samples, accounting for 80% of the total data from 21,568 samples, were used to train a model that identifies one class of head. Furthermore, 2,250 data samples, or 10% of the total, were used as verification data. Finally, the remaining 10% of the data, 2,250 samples, were used as the test set.

For training the model that classifies helmet, head, and hat into three classes in the second stage, 9,681 of 10,891 randomly selected data samples from 19,318 training data samples from the first stage were used as the training set; 1,210 data samples were used as the validation set and 1,210 data samples were used as test set, which is equivalent to 10% of the total data. The images belonging to each class were cropped for the second-step training by leveraging the position information of the XML file containing the information regarding each image. Table 4 reveals that, compared to 18,112 cropped images for helmets, 9,472 and 7,993 cropped images of heads and caps, respectively, are very small quantities. Thus, we used data augmentation to address class imbalances, making the amount of training data for the three classes similar (Suh et al., 2021). Data augmentation is a technique for generating new data through tasks such as flipping, cropping, rotation, and noise injection based on the original dataset to solve overfitting and underfitting problems because of insufficient data (Shorten & Khoshgoftaar, 2019). We applied rotation, left and right shift, horizontal flip, vertical flip, and shear techniques using Keras's ImageDataGenerator function.

Table 3
Summary of datasets generated for two-step AI algorithm in this study.

Dataset	Step	Subset	Number of Samples	Percentage
Roboflow/Crawling/ Kaggle dataset (21,568 helmet, head, hat images)	1	Training Set	17,068	80%
		Validation Set	2,250	10%
		Test Set	2,250	10%
	2	Training Set	9,681	80%
		Validation Set	1,210	10%
		Test Set	1,210	10%

Table 4
Using 2-step data-augmentation techniques to eliminate the training set (Step 2) class imbalance.

Dataset	Number of crop images	Data augmentation
Helmet	18,112	18,112
Head	9,472	18,895
Hat	7,993	17,883
Total	**35,577**	**54,890**

3.3. YOLO

YOLO is a typical one-stage object detection technique that performs object positioning and classification in the same network. That is, YOLO considers the bounding box and class probability in the image as a single regression problem and guesses the type and position of the object by looking at the image once. Since the processing process is simpler than two-stage detector, it has the advantage of showing low background error (False-Positive) due to a high contextual understanding of the class by looking at the entire image at once. In addition, generalized features of an object can be learned more than other algorithms, so it can have high accuracy. However, it is difficult to learn about regional features and is generally less accurate than two-stage detector (Redmon et al., 2016).

Here's how YOLO detects objects in images: First, divide the input image by S*S Grid. At this time, each grid cell has n Bounding Box, a Confidence Score, and a conditional Class Probability equal to the number of classes. In this case, the Confidence Score in Equation (1) means the product of $\Pr(object)$, which is the probability that an object exists in the grid cell, and IoU for the bounding box, and the Conditional Class Probability in Equation (2) means a probability value for each class. Finally, Test Time is determined by multiplying the conditional class probability with the confidence score of the bounding box to obtain a class-specific confidence score (Redmon & Farhadi, 2018).

$$Confidence\ Score = \Pr(Object)*IOU_{pred}^{trust} \tag{1}$$

$$\text{Conditional Class Probability} = \Pr(Class_i|Object) \tag{2}$$

Table 5
Summary of YOLOv5; Speed: Inference speed obtained using NVIDIA V100 GPU. AP is calculated using MS COCO's standard metrics (AP@[.5,.95]).

Model	Size (FP16)	Speed (NVIDIA V100)	mAP (MS COCO)
YOLOv5s	14 MB	2.2 ms	36.8
YOLOv5m	41 MB	2.9 ms	44.5
YOLOv5l	90 MB	3.8 ms	48.1
YOLOv5x	**168 MB**	**6.0 ms**	**50.1**

$$\text{Class Specific Confidence Score} = \text{Conditional Class Probability}*\text{Confidence Score} = \Pr(Class_i|Object)*\Pr(Object)*IOU_{pred}^{trust} = \Pr(Object)*IOU_{pred}^{trust} \tag{3}$$

The architecture of YOLOv1 was released in 2016 by Joseph Redmo and others. It was designed based on the GoogLeNet for image classification model, and YOLO continues to refine the structure of the hidden layer used in the architecture with each version. YOLOv5 used in this study uses CSPNet-based CSP-Darknet (Jocher, 2022). YOLOv5 divides backbones such as s, m, l, and x into four types by size based on depth multiple and width multiple, with x having the deepest depth and highest accuracy of the model (Ge et al., 2021).

In this study, YOLOv5x, which has the highest accuracy among YOLOv5 models, was adopted (Table 5).

3.4. EfficientNet

EfficientNet is a ConvNet-based model created by Google Brain in 2019. Instead of altering the channel (width), layer (depth), and input size (image resolution) individually to increase accuracy, a method of extending the model by integrating all three was proposed.

In general, scaling a model in CNN Architecture aims to find the best layer function ($\boldsymbol{F_i}$) by adjusting width, depth, and input resolution. However, the problem with scaling only one of the three dimensions is that it is an arbitrary adjustment based on the developer's or researcher's experience and experimental environment, resulting in suboptimal accuracy and efficiency in a limited resource range. On the other hand, EfficientNet fixes $\boldsymbol{F_i}$ and uniformly scales the remaining three dimensions: number of layers, number of channels, and input image size. In other words, it is an optimization algorithm to find a model with maximum accuracy within the memory and FLOPs constraints of a given model. In the case of ResNet and MobileNet, where only one of the three dimensions was typically scaled, Compound Scaling showed higher performance than before, and in the case of EfficienNet, the conventional scaling method and Compound Scaling method had the best performance with the same amount of computation (Brock et al., 2021).

EfficientNet is available in model sizes ranging from B0 to B7. To develop an efficient network design in the smallest network, EfficientNetB0, the size is increased by scaling up to EfficientNetB7. The EfficientNet model is lighter, quicker, and more accurate than previous networks (Tan & Quoc, 2019). EfficientNetB7, which has the highest accuracy among the EfficientNet models, was employed in this study. The baseline network of EfficientNetB7 is presented in Table 6.

Table 6
EfficientNetB7 baseline network.

Stage	Operator	Resolution	Channels	Layers
1	conv 3 × 3	112 × 112	64	1
2	Block 1—MBconv1 3 × 3	112 × 112	32	4
3	Block 2—MBconv6 3 × 3	56 × 56	48	7
4	Block 3—MBconv6 5 × 5	28 × 28	80	7
5	Block 4—MBconv6 3 × 3	14 × 14	160	10
6	Block 5—MBconv6 5 × 5	14 × 14	224	10
7	Block 6—MBconv6 5 × 5	7 × 7	384	13
8	Block 7—MBconv6 3 × 3	7 × 7	640	4
9	conv 1 × 1 & Pooling & FC layer	7 × 7	2560	1

3.5. Network architecture of Yolo-EfficientNet

Two models, YOLOv5x and EfficientNetB7, were used in combination to increase classification accuracy to detect not only whether a safety helmet is worn in the workplace but also if a hat other than the protective gear is worn.

The network structure of the combined Yolo-EfficientNet is shown in Fig. 2, and the corresponding parts of the backbone and neck are the network structure of YOLOv5x. Fig. 2(b) shows the baseline network of EfficientNetB7.

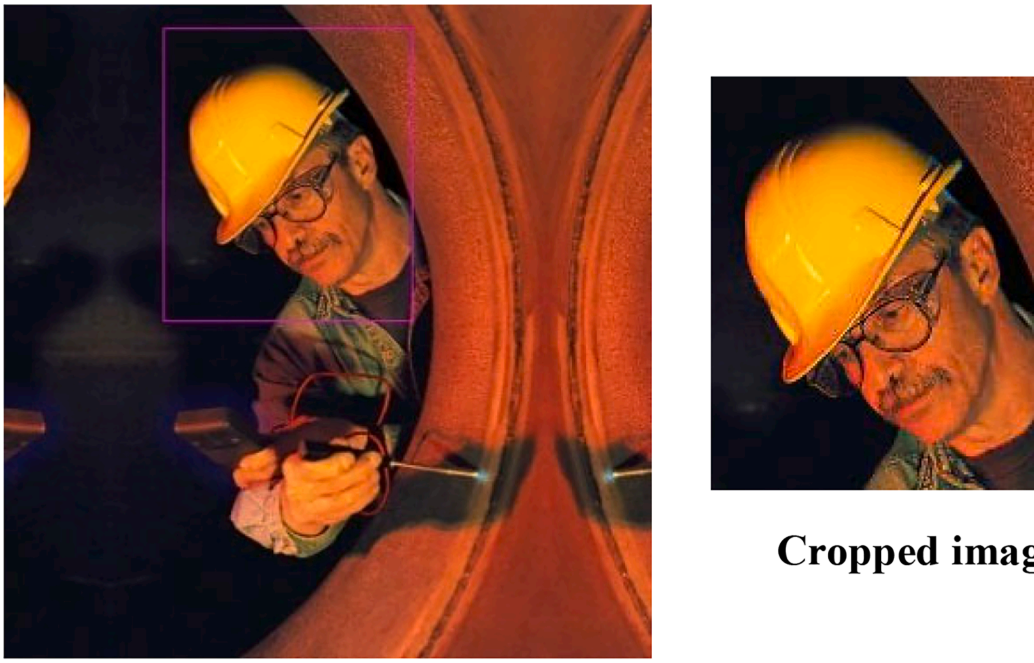
The backbone of Yolov5 involves extracting a feature map from an image through convolution with batch normalization and sigmoid linear unit (SiLU) (convolution-batchnorm-SiLU (CBS)), cross-stage partial network (CSP) bottleneck with three convolutions (C3), and SPP - Fast (SPPF). CBS is used to extract features from blocks consisting of convolutional layers, batch normalization, and activation functions of SiLU, and CSP is a method for dividing the feature map into two parts to reduce the heavy-inference computations caused by duplicate gradient information, performing only a partial convolutional operation and integrating with the rest (Wang et al., 2020). This method significantly reduces the amount of computation and improves accuracy by having a large correlation difference while passing through other network paths. C3 is a module that uses these CSPNet features and includes a bottleneck CSP layer that returns some of the input values as output values using residual connections. SPP transfers a fixed vector size to the fully connected (FC) layer by pooling the feature map into kernels of 5, 9, and 13 and then merging them again, regardless of the size of the input image (He et al., 2015). In Yolov5, SPP was simplified to apply SPPF using only a kernel size of 5.

The neck of Yolov5 purifies and reconstructs the feature map; it consists of a path aggregation network (PANet). PANet was developed in the feature pyramid network, which configures a feature map through neural networks step-by-step and then descends from the upper layer to combine features to detect objects (Liu et al., 2018). PANet was developed to solve the problem that the first extracted low-level feature is not sufficiently reflected in the high level, and unlike FPN, low-level information is transferred to the high level through 100 layers or more. PANet adds bottom-up path aggregation to transfer low-level information to the high level.

The bounding box generated based on the previously extracted feature map is passed to the EfficientNet model and used for classification. EfficientNet uses a mobile inverted bottleneck convolution (MBConv) block, which increases the number of channels with the expansion layer, performs depthwise convolution, reduces the number of input channels again through the projection layer, and adds a normalized vector block to the last layer (Hu et al., 2018).

3.6. Step 1 – Head detection

The model used in Step 1 was YOLOv5x. If the head is not detected correctly in the first step, the input data for the second-step classification process is lost. Thus, transfer learning using previously acquired weights was used for high accuracy. From the training set and validation set, 19,318 samples corresponding to the three classes of helmet, head, and hat were established, and the model was trained and validated using



Cropped image

Detection result

Fig. 3. Example of a first-step result.

these data. Images of workers crawling in construction sites and wearing hats in the workplace, which are shown in Fig. 3, were used as inputs, and the position of the bounding box, which is the detection result, was supplied as input data in Step 2. To overcome the difficulty of detecting non-human objects or structures similar to human parts, such as a human head, during the detection process, the confidence was increased such that the head could be detected only when the confidence was higher than 60%.

Table 7
Main considered parameters and values using hyperparameter evolution.

Model	Description	Value
YOLOv5x	Batch size	4
	Number of epochs	10
	Learning rate	1E-2
	Optimizer	SGD

The parameters for YOLOv5x model training are shown in Table 7, and hyperparameter evolution was used to select the optimal parameters. Hyperparameter evolution is a method of optimizing hyperparameters using a genetic algorithm (GA) by selecting candidates for the optimal solution (selection), creating derived algorithms, and combining them (crossover), modifying the created algorithm through mutation (mutation), and selecting the best algorithm through comparison (update) (He et al., 2021).

Fig. 4 shows the detection and classification losses for the training and validation datasets per epoch after training the YOLOv5x model to detect one head class. Regarding detection performance, mAP@0.5, precision, and recall were 0.986, 0.963, and 0.959, respectively, when the epoch was set to 10 based on 2,250 validation data samples.

The result of classifying 2,250 test samples to evaluate the detection performance of the first-step head detection model obtained mAP@0.5, precision, and recall of 0.971, 0.973, and 0.953, respectively (see Fig. 5).

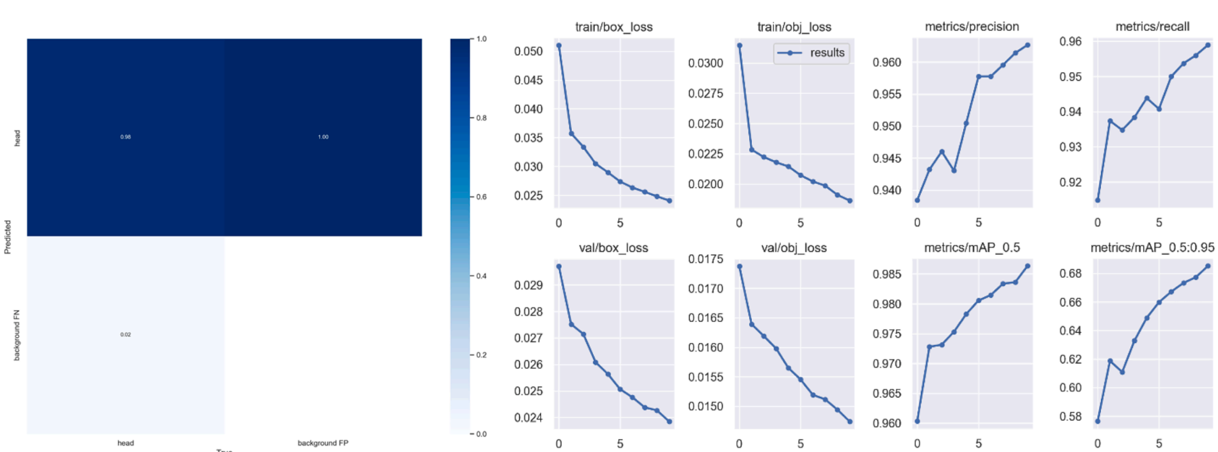


Fig. 4. Results of training with training and validation datasets to detect head.

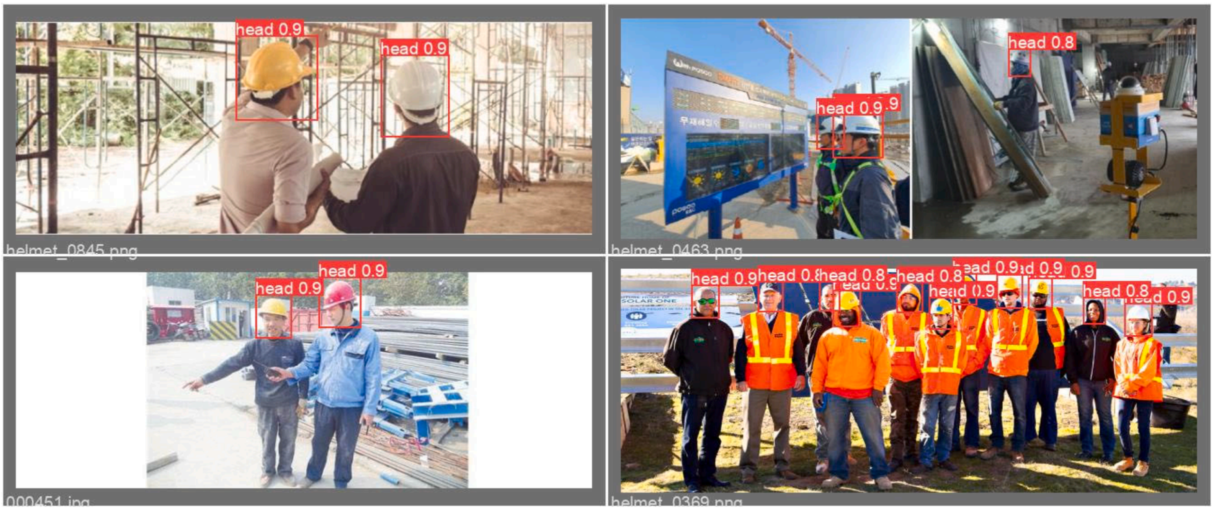


Fig. 5. Some examples of head detection after Step 1.

3.7. Step 2 – Classification of wearing a hardhat

Step 2 determines whether the safety helmet is worn by analyzing the human head position in the image from Step 1. Owing to the limited size of the training set, transfer learning was performed using EfficientNet, the previously described pretrained model, to achieve high performance. To prevent the features of the custom dataset from being trained for transfer learning, the convolutional base, wherein multiple layers of convolutional and pooling layers are stacked to extract features, was frozen, and only the classifier consisting of an FC layer was freshly trained to classify into three classes: head, hat, and helmet. Furthermore, a Softmax activation function, which is used to classify multiple classes, was applied to the feature map acquired from the FC layer. When the Softmax function was used, the output value of the FC layer was normalized between 0 and 1, indicating the probability that each belongs to the corresponding class. The object classification selects the class with the greatest probability, and the total of the output values is always 1. The Softmax layer of the model input data shape was (128, 1) in this study. Because the objective was to classify the output data into three classes of helmet, head, and hat, the form of the output data was (3, 1). The weight shape was (3, 128) in this case, and the Softmax layer of the model contained 384 parameters.

The EfficientNetB7 model was trained using 18,112 helmet images, 18,895 head images, and 17,833 hat images, as presented in Table 4. The parameters used during training are shown in Table 8.

We used the early stopping function in Keras to set the number of epochs. Early stopping is used to address the phenomenon where the loss on the training set decreases over the course of the optimization process but starts to increase again at some point (Mahsereci et al., 2017). It allows the user to choose a stopping point for the iterative algorithm based on pre-designed criteria such as validation loss and accuracy before model training; it stops the training when those criteria are met (Raskutti & Yu, 2014). We set the stopping criteria for early stopping to stop training if the validation loss did not decrease more than 20 times when the training was repeated 300 times. As shown in Fig. 6, the training was stopped at epoch 23, and the weights of epoch 10, which had the highest validation accuracy, were used. When the epoch was 10, the validation accuracy was 0.967, and the validation loss recorded 0.145. Using these weights, the final classification model achieved an accuracy of 96.6% on 1,210 test data points.

The validation set was combined with the training set, as shown in

Table 8
Main considered parameters and values for training EfficientNetB7 model.

Model	Description	Value
EfficientNetB7	Batch size	32
	Number of epochs	300
	Patience	20
	Learning rate	0.0005
	Optimizer	SGD

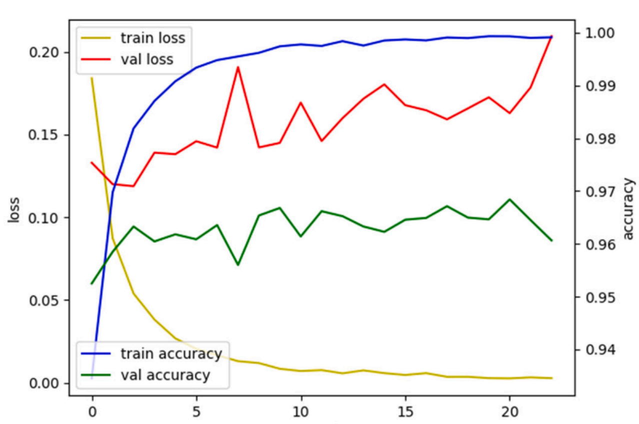


Fig. 6. Results of applying early stopping, which stops learning if the validation loss is not reduced more than 20 times.

Table 9
Final datasets used for training.

Class	Final subset for training	Number of cropped images	Data augmentation
Helmet	Training Set	18,112	18,112
Head	Training + Validation Sets	12,633	17,992
Hat	Training + Validation Sets	10,662	17,211

Table 10
Performance results on combination of training and validation set.

Model	Class	Precision	Recall	F1-score
EfficientNetB7	Helmet	0.982	0.969	0.975
	Head	0.927	0.971	0.948
	Hat	0.973	0.962	0.967
	Total	**0.961**	**0.967**	**0.963**

Fig. 7. Results of two-step method; 1. Human head detection using the Yolov5x model and 2. Helmet, head, and hat classification using the EfficientNetB7 model.

Table 9, to enhance model performance, and the training was repeated. To balance the classes, the validation data were combined only for the head and hat classes because the helmet class contained much more data than these two classes. Furthermore, the data were augmented only for two classes, head and hat. Finally, 18,112 helmet images, 17,992 head images, and 17,211 hat images were used for training. The accuracy of the classification model trained on the combined dataset was 96.8% (Table 10).

For the hat class, although in the previous work, images of people wearing hats were crawled from Google Images rather than images of workers wearing hats in the workplace, the YOLO-EfficientNet model detected the head class in the first step and then cropped the image of the head. In the second step, the cropped image was classified into one of the three classes: head, helmet, or hat. Results confirmed that the model accurately classified the hats worn by workers in the test set, as shown in Fig. 7.

4. Experimental environment

Table 11 lists the computer specifications and software environment used to train and test the model. GPU operation was performed using CUDA 11.3 and cuDNN 8.2.0, both of which are software tools for increasing the training speed for deep learning.

Table 11
Experimental environment.

CPU	Intel® Core™ i7-10700 K CPU @ 3.80 GHz
RAM	24 GB
SSD	512 GB
GPU	NVIDIA GeForce RTX 3060 Ti
SW	CUDA 11.3 & cuDNN 8.2.0 TensorFlow 2.5.0 Python 3.7

5. Experimental results for hardhat detection

5.1. YOLO-EfficientNet

The performance of the YOLO-EfficientNet-based safety-helmet-detection model in a workplace was evaluated using mAP@IoU = 0.5 and the F1-score index. One of the indices used to assess the accuracy of object detection is the IoU. In general, it has a value between 0 and 1, and is used as a criterion for object-detection accuracy. This is a technique for calculating the difference between the real object location B_{gt} (ground truth) and the predicted object position B_p (prediction) by measuring the area where two bounding boxes overlap. It can be expressed as Equation (4). A larger overlapping region indicates a higher accuracy of prediction.

$$IoU = \frac{area(B_{gt} \cap B_p)}{area(B_{gt} \cup B_p)} \tag{4}$$

The IoU value was fixed at 0.5 in this study. Thus, if it was more than 0.5, it was classified as true positive; if it was less than 0.5, it was classified as false positive.

The AP metric was established in response to the difficulties in quantitative performance analysis for models having different precision-recall (PR) curves, which depict precision and recall values according to the variation in the confidence level threshold. AP presents performance as a single value, and techniques for its calculation include 11-point interpolation and interpolating all data. In the case of 11-point interpolation, the average accuracy at 11 points is determined by putting a point on the biggest value among the PR values larger than the current recall value. Equation (5) represents the 11-point interpolation. $\rho_{interp}(r)$ in Equation (6) is the accuracy measured when recall is $\widetilde{r}$. Moreover, all data is interpolated to further minimize the error. In contrast to the 11-point interpolation, it reflects the average accuracy at all points, as stated in Equation (7), which is determined by the area under the line of the graph for the PR curve.

Step-1 result

helmet	0.881836	33	75	100	153
hat	0.879395	125	14	249	163
helmet	0.877441	421	117	488	201
helmet	0.872559	315	113	396	195
helmet	0.849121	478	61	571	158
helmet	0.823242	213	53	281	135

Step-2 results

Fig. 8. YOLO-EfficientNet model results.

$$AP = \frac{1}{11}\sum_{r\in\{0,0.1,0.2,\cdots,1\}} \rho_{interp}(r) \tag{5}$$

$$\rho_{interp}(r) = \frac{max}{\widetilde{r}:\widetilde{r}\geq r} p(\widetilde{r}) \tag{6}$$

$$AP = \sum_i (r_{i+1}-r_i)\rho_{interp}(r_{i+1}) \tag{7}$$

$$\rho_{interp}(r_{i+1}) = \frac{max}{\widetilde{r}:\widetilde{r}\geq r_{i+1}} \rho(\widetilde{r}) \tag{8}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{9}$$

As shown in Equation (9), mAP is a performance index employed in this study when there are many object classes. It is calculated by adding all the APs for each class and then dividing by the number of classes, *n*. F1-score is a classification model performance index that is derived as the harmonic average of accuracy and recall in Equation (10). When accuracy and recall are considered separately, a trade-off relationship occurs, making it difficult to make a judgment. Therefore, it is employed to consider both indices. The F1-score was used in this study because it reflects the correct performance result of the model without being biased toward a single value when the number of classes in the test data is unbalanced.

$$F1Score = \frac{2\times Precision\times Recall}{Precision + Recall} \tag{10}$$

The YOLO-EfficientNet model was evaluated using 1,210 photos from the testing set that had previously been classified. Because the model is divided into two steps, the test images were fed into the YOLO model in the first step, and the resulting head position information was fed into the trained classification model in the second step. The two-step classification model output the classification result (class name) of the detected object, the probability of belonging to the class, and the position information of the bounding box, as shown in Fig. 8, in the form of a.txt file.

According to Table 12, the performance results indicate that the APs were 92.7%, 87.3%, and 94.2% for the helmet, head, and hat classes, respectively, whereas the F1-scores were 95.2%, 89.5%, and 96.5%, respectively. Finally, the mAP@0.5 was 91.4% and the macro F1-score, which is the average F1-score for each class, was 93.7%.

Table 12
APs and F1-scores for three classes; AP_{50}: Average Precision (IoU = 0.5).

Detection site	AP_{50}	F1-Score
helmet	0.927	0.952
head	0.873	0.895
hat	0.942	0.965

5.2. Comparison results and discussion

Faster R-CNN (Ren et al., 2016), YOLO, and RetinaNet (Lin et al., 2018) were used to compare the performances of the safety helmet-wearing detection model proposed in this paper and other object-detection models.

5.2.1. Faster R-CNN

Faster R-CNN (Ren et al., 2016) introduced the region proposal network (RPN), which extracts candidate areas to address the issue that the selective search method employed in Fast R-CNN (Girshick, 2015) runs on the CPU and generates bottlenecks. RPN is positioned between feature maps and RoI pooling in the Fast R-CNN framework, and it generates region proposals by using anchor boxes of varying sizes and horizontal and vertical ratios. The center of each grid cell in the source image is used to build the anchor box. Anchors, or reference points, in the original image are fixed depending on the sub-sampling ratio. Many region proposals that suggest object positions of varying sizes are constructed by applying the anchor box obtained by the sliding window technique to the feature map. Regression and classification are performed for each point.

5.2.2. RetinaNet

Among the object-detection models, RetinaNet (Lin et al., 2018) provides a novel loss function, known as focused loss, to overcome the class imbalance issue that occurs with extremely few positive samples relative to the number of negative samples in a one-stage detector. It is a single integrated model that is built on the Resnet101 + FPN backbone and subnetworks for two tasks, classification and box regression, and in which focal loss is applied to the classification subnet output. A one-stage detector is faster and simpler than a two-stage detector, but it has the same limitation in that numerous background class samples are created because dense sampling is used instead of approaches such as selective search and RPN to identify candidate sites. The focal loss proposed to overcome this issue takes the form of a dynamic scaling factor that varies depending on the class, which is added to the cross-entropy loss. Easy examples may be used with lower weights during focal loss training, whereas challenging examples can be used with higher weights.

5.2.3. Comparison of results

For fair comparison between CNN-based models, epochs were all set identically (Rao & Ni, 2016), and the same datasets used for training and validation of the first step YOLOv5x model were used as the training, validation, and test sets. Regarding the setting of the main parameters, Table 13 summarizes the set default values by model.

The macro F1-score was used, and only cases having a confidence score of 0.6 or above were set to be detected. The YOLOv5x model, which is a one-stage detector, and the Faster R-CNN model with a

Table 13
Main considered parameters and related default values.

Model	Description	Value
Faster R-CNN	Batch size	4
	Number of epochs	10
	Learning rate	0.005
	Optimizer	SGD
YOLO	Batch size	4
	Number of epochs	10
	Learning rate	1E-2
	Optimizer	SGD
RetinaNet	Batch size	4
	Number of epochs	10
	Learning rate	1E-5
	Optimizer	Adam

backbone of ResNet50 + FPN, which is a two-stage detector, both achieved the highest mAP of 0.918. YOLOv5x achieved the best precision, but recall was the highest for Faster R-CNN at 0.935. In the case of the RetinaNet model, precision, recall, mAP, and F1-score were the lowest compared to the other models. Upon experimenting with increasing the number of epochs to 100 to analyze the cause, it was confirmed that precision, recall, mAP, and F1-score recorded 0.887, 0.695, 0.672, and 0.773, respectively, and the model did not train well. The YOLO-EfficientNet model proposed in this study did not achieve the best performance for precision, recall, and mAP@0.5. However, its mAP was 0.914, ranking second. Unlike other models, which have poor recall when precision is high, or have high recall when precision is low, the proposed model performed well in both evaluation indices, with a precision of 0.942 and recall of 0.933. Consequently, the F1-score when both evaluation indices were considered was 0.937, which was the highest among the comparison models (Table 14). Among the YOLOv5 models, the smaller YOLOv5s model had the shortest training time, with the RetinaNet model showing the longest training time of 6.27 h, and for the YOLO-EfficientNet model, the YOLOv5x model used in the first human head detection phase took 4.87 h, and the second safety helmet, head, and hat classification phase, 1.27 h. In the first step, only the human head was detected in the image; therefore it was 0.19 h faster than detecting three classes of YOLOv5x model alone, and in the second step, the EfficientNetB7 model used for the classification of the three classes of helmet, head, and hat took a training time of 1.27 h, not detection. In terms of inference time, YOLO, a one-stage detector, was the fastest, YOLOv5s, the smallest model size, achieved the shortest testing time of 9 s. Although the YOLO-EfficientNet model took longer for inference than YOLO, because it performs two steps, it recorded a faster speed than the two-stage detector, Faster R-CNN, with a difference of 6.5 s to 25.3 s in comparison.

5.3. Failure cases

During the evaluation of the proposed model, several failure cases were observed at each step. Fig. 9(a) illustrates the cases in which the head detection using the first-stage YOLOv5x model failed or missed detection depending on the characteristics of the workspace images. Based on an analysis of the failed data, the first-stage model had difficulty detecting the heads of people in cases where 1) people were too small in the workspace, 2) there were too many people in the image, or 3) people overlapped each other and were not clearly visible in the image.

Fig. 9(b) presents failure cases in the second-stage model, EfficientNetB7, which classifies three classes of helmet, head, and hat, depending on the image characteristics, as in the first stage. The cases where classification failed were 1) crop images being too small to clearly show the shape of a helmet or hat, and 2) people overlapping each other and their shapes being unclear. The Softmax probability in Fig. 9(b) represents the predicted probabilities of each class through the Softmax classifier applied to the classification layer, and it can be used to confirm the classes confused during the prediction of the classification model.

6. Conclusions and future work

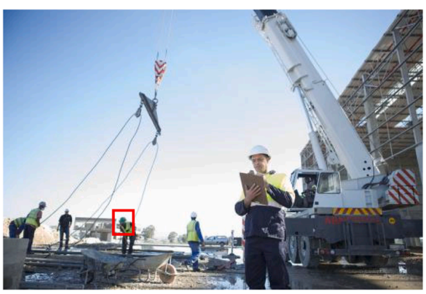
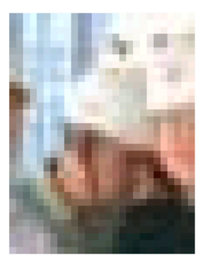
In this study, we designed a deep learning-based workplace safety helmet classification model to detect whether workers are wearing safety helmets. We trained a first-step model that can recognize heads using 21,568 image data samples from Roboflow, Kaggle, and crawling datasets. Thereafter, the position information of the XML file containing information of 10,891 data for second-step training was used to crop the images corresponding to the head, hat, and helmet classes and separate them into 50,392 head parts, which were used to create a helmet-head-hat classification model based on EfficientNetB7. A YOLO-EfficientNet-based safety helmet classification model was then designed by combining it with a YOLO-based human head detection model, and this model finally achieved an mAP of 91.4% and F1-score of 93.7%.

The model proposed in this study not only employs the YOLO transfer learning model with existing weights trained on a large amount of data, but also employs the classification model for higher accuracy. Therefore, it is significant in that the existing training data can be used in two steps to achieve higher performance than when just one model is used. The model could be expanded if a classification model with higher performance than the EfficientNetB7 model employed in this study is developed in the future. It offers the advantage of being more accurate than the current model if more training datasets for the hat and head classes are obtained and the second-step model is trained. Importantly, it is a model that addresses the real safety of workers because it can detect hats similar to a safety helmet that do not belong to PPE with high accuracy. The Ministry of Land, Infrastructure and Transport and the Korea Digital Foundation's technology using Faster R-CNN model for recognizing non-wearing of safety helmets, mentioned in the introduction, was compared with the YOLO-EfficientNet model used in this

Table 14
Evaluation of object-detection models using precision, recall, mAP, and F1-score on the hardhat dataset.

Models	Epochs	Precision	Recall	*m*AP0.5	F1-score	Training Time (h)	Testing Time (s)
Faster R-CNN$_{resnet50}$	10	0.828	0.909	0.892	0.866	4.3	48.6
Faster R-CNN$_{resnet50fpn}$	10	0.849	0.935	0.918	0.890	2.59	67.4
YOLOv5s	10	0.943	0.798	0.884	0.864	2.3	9
YOLOv5m	10	0.951	0.843	0.908	0.894	2.76	13
YOLOv5l	10	0.951	0.851	0.913	0.898	5.63	17
YOLOv5x	10	0.955	0.858	0.917	0.904	5.06	22
RetinaNet	10	0.264	0.138	0.201	0.179	6.27	205
YOLO-EfficientNet	10	0.942	0.933	0.914	0.937	6.14	42.1

(a)

Original Image	Crop Image	Class	Softmax Probability	Label	Predict
		Helmet	0.3979498	●	
		head	0.6020406		●
		hat	0.0000096		
		helmet	0.3762282	●	
		head	0.0181904		
		hat	0.6055813		●

(b)

Fig. 9. Failure cases of (a) head detection network, (b) helmet, head, hat classification network. In (a), pink boxes denote the head. In (b), red boxes indicate where the misclassified crop image is in the original image.

paper. The results showed that the YOLO-EfficientNet model achieved a higher F1-score and faster testing time, and higher performance in evaluation considering the class imbalance and demonstrated a more suitable speed for real-time use.

The proposed method can replace the safety manager in monitoring whether employees are wearing safety helmets. Thus, it is expected to assist in the prevention of workplace accidents. However, in this study, the hat class was not trained on images of workers wearing hats in the workplace—unlike the helmet and head classes—which may limit its optimization for real workplace environments. To overcome this limitation, it is necessary to collect a dataset of workers wearing hats in the workplace for training. Although the YOLO model has been updated to v8, this study used YOLOv5. If more advanced object-detection and classification models are used in both the first and second stages of this study in the future, it is possible to achieve better results.

In the future, we plan to collect training datasets for the head and helmet classes to train a two-step classification model, as well as to focus on reducing the model's weight to achieve real-time capability for detecting whether or not a safety helmet is worn.

CRediT authorship contribution statement

Ju-Yeon Lee: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Woo-Seok Choi:** Conceptualization, Methodology, Software, Writing – review & editing, Visualization. **Sang-Hyun Choi:** Resources, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Funding: This work was supported by the BK21 FOUR of the National Research Foundation of Korea (NRF) funded by the Ministry of Education [grant number: 5199990314333].

References

Ahn, J., Park, J., Lee, S. S., Lee, K. H., Do, H., & Ko, J. (2023). SafeFac: Video-based smart safety monitoring for preventing industrial work accidents. *Expert Systems with Applications, 215*. https://doi.org/10.1016/j.eswa.2022.119397

Brock, A., De, S., Smith, S. L., & Simonyan, K. (2021, July). High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning (PMLR)*, 1059-1071.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection*. arXiv preprint arXiv:2004.10934*.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, 5562-5570, Venice, Italy. doi: 10.1109/ICCV.2017.593.

Chen, Z., & Su, M. (2021, November). Improved Mask R-CNN Method for Intelligent Monitoring of Helmet in Power Plant. *2021 Photonics & Electromagnetics Research Symposium (PIERS)*, 844-848, Hangzhou, China. doi: 10.1109/PIERS53385.2021.9695098.

Cheng, R., He, X., Zheng, Z., & Wang, Z. (2021). Multi-scale safety helmet detection based on SAS-YOLOv3-Tiny. *Appl. Sci.*, 11(8), Article e3652. 10.3390/app11083652.

Chowdhury, M., Gao, J., & Islam, R. (2016, July). Fuzzy logic-based filtering for image de-noising. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2372-2376, Vancouver, BC, Canada. doi: 10.1109/FUZZ-IEEE.2016.7737990.

Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886–893. 10.1109/CVPR.2005.177.

Du, L., Zhang, R., & Wang, X. (2020, March). Overview of two-stage object detection algorithms. In *Proceedings of Journal of Physics Conference on International Conference on Intelligent Computing and Signal Processing (ICSP)*, Suzhou, China. doi: 10.1088/1742-6596/1544/1/012033.

EHS Today. (2022, August 16). *ASSP to enhance standards to guide safety at construction sites*. Retrieved from https://www.ehstoday.com/construction/article/21248716/assp-to-enhance-standards-to-guide-safety-at-construction-sites. Accessed October 20, 2022.

EKU Online. (2020, July 28). *The Demand for Safety Professionals in the U.S*. Retrieved from https://safetymanagement.eku.edu/blog/the-demand-for-safety-professionals-in-the-u-s/. Accessed March 17, 2023.

Elgendy, M. (2020). Deep Learning for Vision Systems (1st ed.). Object Detection With R-CNN, SSD, and YOLO (Chapter 7).

Foulis, M. (2021, September 22). *7 Ways to prevent workplace accidents.* Canadian Occupational Safety. Retrieved from https://www.thesafetymag.com/ca/topics/safety-and-ppe/7-ways-to-prevent-workplace-accidents/310921. Accessed October 20, 2022.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*.

Girshick, R. (2015). Fast R-CNN. *arXiv preprint arXiv:1504.08083v2*.

Girshick, R. (2015, December). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440-1448. doi: 10.1109/ICCV.2015.169.

Gkioxari, G., Girshick, R., & Malik, J. (2015, December). Contextual Action Recognition With R*CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1080-1088, Santiago, Chile. doi: 10.1109/ICCV.2015.129.

Han, G., Zhu, M., Zhao, X., & Gao, H. (2021). Method based on the cross-layer attention mechanism and multiscale perception for safety helmet-wearing detection. *Computers and Electrical Engineering, 95*. https://doi.org/10.1016/j.compeleceng.2021.107458

He, K., Gkioxari, G., Doller, P., Girshick, R. (2020). Mask R-CNN. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386-397. doi: 10.1109/TPAMI.2018.2844175.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell, 37*(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. 10.1109/CVPR.2016.90.

He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems, 212*, Article 106622.

Heesung, C., Byungju, C., Jungheon, K., Sangyoung, L., Taeheon, K., Eunchul, A & Gyuwon, K. (2021, December). Recommendations for issues and improvement measures in the application of smart construction technology on construction sites: focusing on the case of DL Engineering & Construction. *Construction Engineering and Management*, 22(6), 34–41.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, J., Shen, L., & Sun, G. (2018, June). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132–7141, Salt Lake City, UT. 10.1109/CVPR.2018.00745.

Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In *Proceedings of the IEEE international conference on big data (BIG DATA)*, 2503-2510, Seattle, WA, USA. doi: 10.1109/BigData.2018.8621865.

Iannizzotto, G., Bello, L. L., & Patti, G. (2021). Personal Protection Equipment detection system for embedded devices based on DNN and Fuzzy Logic. *Expert Systems with Applications, 184*. https://doi.org/10.1016/j.eswa.2021.115447

Jocher, G. (2022). YOLOv5 SOTA Realtime Instance Segmentation. https://github.com/ultralytics/yolov5/releases/. Accessed March 7, 2023.

Juyoung, K. (2021, November 15). *[2nd anniversary] Lack of safety managers, are there any countermeasures?* Mechanical Equipment Newspaper. Retrieved from https://www.kmecnews.co.kr/news/articleView.html?idxno=23439. Accessed March 17, 2023.

Korea Occupational Safety and Health Agency. (2020, November 30). *A comparative analysis of changing trends in occupational accident rate among major countries*. Retrieved from https://www.kosha.or.kr/oshri/publication/researchReportSearch.do?mode=view&articleNo=419720. Accessed November 1, 2022.

Korea Occupational Safety and Health Agency. (2021a, November 29*). Status of industrial accidents at the end of September 2021*. Retrieved from https://www.kosha.or.kr/kosha/data/industrialAccidentStatus.do?mode=view&articleNo=427063. Accessed January 19, 2022.

Korea Occupational Safety and Health Agency. (2021b, December 31). *Industrial accident status analysis in 2020*. Retrieved from https://www.kosha.or.kr/kosha/data/industrialAccidentStatus.do?mode=view&articleNo=436868. Accessed January 19, 2022.

Association, K. I. H. (2004). Safety Class of the Month - Importance of safety protection equipment and management tips. *Journal of the Safety Technology, 79*(6), 92–107.

Larxel (2020). Safety Helmet Detection. https://www.kaggle.com/datasets/andrewmvd/hard-hat-detection.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv: 2209.02976*.

Li, K., Zhao, X., Bian, J., & Tan, M. (2018). Automatic safety helmet wearing detection. *arXiv preprint arXiv:1802.00264v1*.

Li, Y., Wei, H., Han, Z., Huang, J., & Wang, W. (2020). Deep learning-based safety helmet detection in engineering management based on convolutional neural networks. *Advances in Civil Engineering*, Article e9703560. 10.1155/2020/9703560.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014, September). Microsoft COCO: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., & Tuytelaars, T. (Eds), *Computer Vision – ECCV 2014*: *Lecture Notes in Computer Science*, 8693 (pp. 740–755). Springer, Cham. 10.1007/978-3-319-10602-1_48.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection*. arXiv preprint arXiv:1708.02002v2.*

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018, June). Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8759–8768, Salt Lake City, UT. 10.1109/CVPR.2018.00913.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *In Computer Vision–ECCV 2016: 14th European Conference*, 21-37, Amsterdam, The Netherlands. 10.1007/978-3-319-46448-0_2.

Lu, X., Li, Q., Li, B., & Yan, J. (2020). MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection. *Computer Vision – ECCV 2020*, 541-557. 10.1007/978-3-030-58568-6_32.

Maren Mahsereci., Lukas Balles., Christoph Lassner., Philipp Hennig. (2017). Early Stopping without a Validation Set. *arXiv preprint arXiv:1703.09580v3.*

Ministry of Employment and Labor. (2004, November 5). *Comparison of occupational health laws of the US, Japan and Korea*. Retrieved from https://www.moel.go.kr/policy/policydata/view.do?bbs_seq=6023. Accessed October 20, 2022.

Ministry of Employment and Labor. (2021, March 25). *Measures to reduce deaths from industrial accidents in 2021*. Retrieved from https://www.moel.go.kr/news/enews/report/enewsView.do?news_seq=12068. Accessed January 19, 2022.

Ministry of Land, Infrastructure and Transport. (2022, February 17). *Seoul implements real-time control of dangerous situations at private construction sites by AI: Pilot project for 50 sites*. Retrieved from https://www.seoul.go.kr/news/news_report.do#view/356681. Accessed June 8, 2022.

Mittal, P., Sharma, A., & Singh, R. (2022). Dilated convolution based RCNN using feature fusion for Low-Altitude aerial objects. *Expert Systems with Applications, 199*. https://doi.org/10.1016/j.eswa.2022.117106

Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction, 112*. https://doi.org/10.1016/j.autcon.2020.103085

Northeastern University - China (2022). Hard Hat Workers. https://public.roboflow.com/object-detection/hard-hat-workers.

Pak, M., Kim, S. (2017). A review of deep learning in image recognition. In *Proceedings of the IEEE Conference on Computer Applications and Information Processing Technology (CAIPT)*, 1-3, Kuta Bali, Indonesia. doi: 10.1109/CAIPT.2017.8320684.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Pramanik, A., Pal, S. K., Maiti, J., & Mitra, P. (2022). Granulated RCNN and Multi-Class Deep SORT for Multi-Object Detection and Tracking. *IEEE Transactions on Emerging Topics in Computational Intelligence, 6*(1), 171–181. https://doi.org/10.1109/TETCI.2020.3041019

Rao, Y., & Ni, J. (2016, December). A deep learning approach to detection of splicing and copy-move forgeries in images. In *2016 IEEE international workshop on information forensics and security (WIFS)*, 1–6. 10.1109/WIFS.2016.7823911.

Raskutti, G., Wainwright, M. J., & Yu, B. (2014). Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *Journal of Machine Learning Research, 15*(1), 335–366.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788, Las Vegas, NV, USA. doi: 10.1109/CVPR.2016.91.

Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497v3*.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *In IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Seoul Digital Foundation. (2022, February 17). *A Study on the Utilization of Digital Technology in the Construction Sector in accordance with the Enforcement of the Serious Disaster Punishment Act*. Retrieved from https://sdf.seoul.kr/research-report/1678?srchKey=sj&srchText=%EA%B1%B4%EC%84%A4. Accessed March 17, 2023.

Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *Journal of Computational Vision and Imaging Systems, 3*. https://doi.org/10.15353/vsnl.v3i1.171

Shinwon, B., Wonhee, L., Hanjoong, K., & Jonggeun, P. (2012, November). A study on accident reduction strategies for small construction sites. Seoul: Occupational Safety and Health Research Institute.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Smart City Korea. (2021, September 13). *Seoul introduces 'smart safety management' to private construction sites for accident prevention with AI and IoT*. Retrieved from https://smartcity.go.kr/en/2021/09/13/%EC%84%9C%EC%9A%B8%EC%8B%9C-%EB%AF%BC%EA%B0%84-%EA%B3%B5%EC%82%AC%EC%9E%A5%E2%80%A4%EA%B1%B4%EC%B6%95%EB%AC%BC%EC%97%90-%EC%8A%A4%EB%A7%88%ED%8A%B8-%EC%95%88%EC%A0%84%EA%B4%80%EB%A6%AC-%EB%8F%84%EC/. Accessed June 8, 2022.

Song, D.-Y., Cho, S. W., & Lee, S. H. (2017). Study on the necessity of improving safety manager reinforcement and replacement regulation system. *Journal of the Korea Safety Management & Science*, 19(4), 77–85. 10.12812/KSMS.2017.19.4.77.

Su, L., Wang, Y., & Tian, Y. (2020). R-SiamNet: ROI-Align Pooling Based Siamese Network for Object Tracking. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 19–24, Shenzhen, China. doi: 10.1109/MIPR49039.2020.00012.

Suh, S., Lee, H., Lukowicz, P., & Lee, Y. O. (2021). CEGAN: Classification Enhancement Generative Adversarial Networks for unraveling data imbalance problems. *Neural Networks, 133*, 69–86. https://doi.org/10.1016/j.neunet.2020.10.004

Tan, M., & Quoc, V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 97:6105-6114. https://proceedings.mlr.press/v97/tan19a.html.

UNISDR, U. (2015, March). Sendai framework for disaster risk reduction 2015–2030. In *Proceedings of the 3rd United Nations World Conference on DRR*, *Sendai, Japan* (Vol. 1).

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv: 2207.02696*.

Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020, June). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 390–391, Seattle, WA. 10.1109/CVPRW50498.2020.00203.

Wójcik, B., Żarski, M., Książek, K., Miszczak, J. A., & Skibniewski, M. J. (2021). Hard hat wearing detection based on head keypoint localization. *arXiv preprint arXiv: 2106.10944*.

Xie, L. (2019). Hardhat. Harvard Dataverse, v1. 10.7910/DVN/7CBGOS.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5987–5995. 10.1109/CVPR.2017.634.

Xu, Z. P., Zhang, Y., Cheng, J., & Ge, G. (2022). Safety helmet wearing detection based on YOLOv5 of attention mechanism. In *Journal of Physics: Conference Series*, 2213(1), p. 012038. IOP Publishing. 10.1088/1742-6596/2213/1/012038.