Lightweight Detection of Small Tools for Safer Construction

Maryam Soleymani[a,*], Mahdi Bonyani[a] and Chao Wang[b]

[a]Ph.D. Student, Bert S. Turner Department of Construction Management, Louisiana State University, USA
[b]Associate Professor and Graduate Program Advisor, Bert S. Turner Department of Construction Management, Louisiana State University, USA

ARTICLE INFO

Keywords:
Small Tools Detection
Construction Worker Safety
Site Monitoring
YOLO
Safety Management
Attention Learning
Object Detection
Deep learning

ABSTRACT

Construction sites present significant potential safety hazards to the workers, with hand tools being a major source of injuries. This paper presents a ***L***ightweight approach for ***S***mall ***T***ools ***D***etection (LSTD) method using a deep neural network for real-time detection of small construction tools. LSTD utilizes a lightweight backbone with Dynamic Feature Extraction, Accurate Separated Head, and Integrated Feature Fusion, reducing parameters by 73% and computations by 28% versus YOLOv5 while achieving 87.3% mean Average Precision (mAP) on challenging construction site datasets. Additional modules enhance detection recall and robustness to appearance variation and scale changes. Extensive experiments demonstrate LSTD's superior performance in misty conditions and illumination changes. With high accuracy in a compact 2.87M parameter network, LSTD brings ubiquitous worker safety monitoring via edge devices closer to reality. The proposed model marks a significant advancement in improving safety in high-risk construction environments.

1. Introduction

Construction sites are hazardous environments where workers are exposed to many safety risks. According to the Occupational Safety and Health Administration (OSHA), 20% of worker fatalities in private industry in 2020 were in construction [1] [2]. Also, construction is among the most dangerous industries but has lagged behind others in technological adoption [3]. Cultural resistance to new techniques often exists, with preferences leaning towards conventional manual approaches [4]. Moreover, Studies show that the four leading causes of construction site fatalities in the United States are falls, electrocutions, being struck by objects, and getting caught in between objects [5]. A leading cause of these incidents is struck-by hazards from objects like falling tools and materials. Small hand and power tools, which are prevalent on construction sites, contribute to these incidents in various ways. For example, electric power tools can cause electrocutions through defective cords, and hand tools may be improperly secured and fall, striking workers below [6, 7]. Preventing such incidents requires effective safety protocols and risk mitigation methods tailored to the construction site environment.

Proper organization, storage, and transport of small construction tools are therefore paramount for site safety, but managing numerous small objects that are constantly in motion is an enormous challenge [8]. Computer vision techniques like object detection, however, now enable automated monitoring and analysis of construction sites. By automatically detecting small tools in images and video feeds, potentially unsafe conditions can be identified so that corrective actions may be taken. Vision-based models can accurately localize small objects like tools and equipment to identify fall and struck-by hazards in real-time [2]. This enables proactive interventions through warnings, relocation of objects, changes to site layout, and standardization of tool storage procedures. Among modern visual detection architectures, You Only Look Once (YOLO) v5 has emerged as a leading approach due to its speed and accuracy [9]. By leveraging YOLOv5 models tailored to construction sites, project managers can track on-site tools and enhance safety protocols in an efficient automated manner.

YOLOv5 [9] is a state-of-the-art one-stage object detector well-suited for real-time analysis of construction sites. As a one-stage detector, YOLOv5 directly predicts bounding boxes (BB) and class probabilities in one evaluation of an image. This allows the model to operate faster than previous two-stage detectors like Faster R-CNN [10] that first generate region proposals. YOLOv5 is also preferred in benchmarks, it achieves high accuracy while requiring fewer floating point operations and memory. These qualities make YOLOv5 well-matched to the domain of construction sites where both speed and accuracy are necessary.

In this paper, we propose a **L**ightweight approach for **S**mall **T**ools **D**etection (LSTD) based on the YOLOv5 models for detecting small construction tools on construction sites. We utilize a comprehensive dataset [11] of common hand and power tools in context within actual construction environments. Using this data, we train a LSTD model with robust performance for tool detection tasks. We additionally demonstrate the real-time capabilities of our tool detector by integrating it with an edge device that warns workers on-site when tools are spotted in hazardous areas.

The ability to accurately and rapidly recognize small construction tools is critical for mitigating safety incidents before they occur. Our LSTD approach provides intelligent situational awareness to identify small objects. As tools are the instruments used for virtually all construction activities, a specialized tool detector gives fulsome visibility into on-site risks. Also, automated tool detection with models like our LSTD should likewise be adopted as an indispensable safety mechanism. Just as essential safety gear protects individual workers, proactive detection systems protect the

*Corresponding author
ORCID(s):

entire work crew by preventing hazardous conditions from arising in the first place.

Monitoring small construction tools is clearly important for improving site safety, but few previous computer vision works have focused specifically on these small objects. Our LSTD model, with high accuracy and real-time performance, can therefore provide managers or site superintendents with an unprecedented ability to track and manage tools for safety. In the remainder of this paper, we provide further technical details on our approach, evaluations, and demonstrations of the system in operation. We believe the wide deployment of fast and accurate vision systems like ours could make substantial impacts by reducing injuries and fatalities on construction sites around the world.

In summary, the contributions of the proposed method are as follows:

- We propose a lightweight end-to-end network for small object detection that utilizes the Accurate Separated Head (ASH), the Integrated Feature Fusion (IFF), and the Dynamic Feature Extraction to capture a more comprehensive feature.
- We illustrate the performance of the proposed methods on the small object detection task on comprehensive dataset [11]. Compared with the baseline equivalents, our method decreases computational complexity and enhances accuracy.

2. Related works

Since small objects lack context and have indistinguishable characteristics, complicated backdrops, and poor resolution, it is challenging to recognize them using conventional object identification methods [12, 13, 14]. Training inputs with smaller-looking objects can help somewhat compensate for this low identification accuracy for little objects. Nevertheless, it might not be feasible to create more training picture datasets using different objects ranging in size from very tiny to very large given the available datasets [15]. In order to effectively recognize tiny objects in a variety of areas, researchers have thus tried to alter and enhance current algorithms without the need for new training picture datasets [16, 17, 15, 18, 19, 20].

When the resolution of the region filled by the small objects is increased, some developed algorithms can identify small things. For instance, Ku et al. [21] suggested a better YOLOv4-based technique that can identify a hard helmet in order to increase worker safety on building sites. Images were sharpened and localized tiny object features were extracted using an image super-resolution (ISR) module. Similar to this, Wang et al. [22] created a method based on YOLOv4 and integrated a feature texture transfer (FTT) module to capture the regional features of tiny objects and improve image resolution. The suggested technique successfully identified the tiny targets—student head movements—in college courses.

Contextual information was used in other attempts to identify tiny objects. This approach uses context to augment information for better identification at low resolutions by using more abstract higher-layer characteristics. A small object detection technique based on the SSD framework with segmentation and detection heads was created by Sun et al. [23]. This technique efficiently recognizes people as well as traffic signs by supplying more semantic features to the detection head via the segmentation head. Furthermore, Lim et al. [24] presented an SSD that uses integrated features to get semantic features as well as an attention module to extract features of the object in order to recognize tiny objects more precisely than traditional SSDs.

Deep learning-based object detection studies for construction sites can be divided into two categories: those that focus on worker behavior recognition [25, 26, 27] and those that just recognize objects like workers and heavy machinery [2, 28, 29, 30]. Luo et al. [29] investigated an object detection model based on a convolutional neural network (CNN) for the purpose of identifying 22 different kinds of heavy machinery and laborers on a construction site. Using CNN characteristics, Fang et al. [2, 28] sought to determine if employees on high floors wore hard helmets. Son et al. [30] reported a detection technique that could differentiate the workers from the backdrop using 3,241 images to create an object detection model for construction site workers. As an alternative, a number of academics have developed a more efficient technique that involves slicing or tiling the input image in order to enlarge small objects inside a wider pixel region, therefore enabling small object recognition [31]. For instance, a small object detection approach based on fine-tuning and slicing-aided hyper-inference was presented by Akyon et al. [31]. For object detection, they separated the input photos into overlapping slices without requiring unnecessary computing power. Although this approach enhanced small object detection performance, the larger pixel area occasionally decreased big object detection. Using the slicing-aided inference approach, Keles et al. [32] assessed the YOLOv5 and YOLOX models and found that sliced inference enhanced small object detection performance. Nevertheless, while cropping the input image, this study did not sufficiently take into consideration redundant objects in the overlapping area. EdgeDuet was developed by Wang et al. [33] to detect medium- to large-sized objects locally on mobile devices while offloading small object detection to the edge. By dividing a frame into many tiles, EdgeDuet allows for parallel offloading, which facilitates small object detection. Through overlap-tiling, this technique also lessens tile dependencies so that objects that span into neighboring tiles are not missed.

As indicated earlier, prior research primarily aimed at enhancing small object detection accuracy revealed that their suggested techniques raised the average precision (AP) in comparison to current algorithms. The majority of small object detection methods were evaluated on the precision of small object recognition in a GPU, despite the use of high-quality pictures. However, real-time object detection

taking into account processing as well as transmission of video data was not well evaluated. Thus, when real-time object identification is required, their field applicability is diminished. In this sense, edge computing has been used in recent construction studies to address automated construction demands by lowering monitoring latency. Chen et al.'s study [34] showed that edge nodes had performance comparable to local devices, suggesting that utilizing edge nodes is feasible for implementing hardhat-wearing detection based on YOLOv5 at a construction site. To solve the original problem of expensive processing, Xu et al. [35] also implemented harness-use detection based on the YOLOv5 on edge nodes. Additionally, Zhang et al. [36] demonstrated the accuracy and effectiveness of edge node detection for risky behavior to address efficiency as well as accuracy challenges. Furthermore, Zhao et al. [27] used YOLOv3 to manage construction sites' safety in real-time after identifying the activities that workers conduct in dangerous regions at outdoor sites, which is another study that looked at worker behavior recognition. By using object identification techniques that included a mask region-based CNN to establish a safe distance between the crane and workers, Yang et al. [25] were able to identify cranes and surrounding workers. The human body was separated into the head, chest, and arms by Zhao and Obonyo [26] in order to identify worker behavior and suggest ways to improve productivity at the site. Investigating whether edge inference may be used effectively for precise and instantaneous tiny object recognition is thus important.

The development of small tools detection algorithms for safety monitoring and tools-manager robots encounters challenges including recognizing small tools in diverse construction environments and deploying efficient algorithms at the edge. This study aims to address these challenges by introducing a lightweight and accurate small tools detection algorithm suitable for deployment in complex construction sites. To enhance detection accuracy, the algorithm selectively expands the original dataset using on-the-fly data augmentation strategies, which improves the model's robustness and generalization ability. Additionally, the algorithm employs a Dynamic Feature Extraction (DFE) module to focus on capturing more related features, thereby improving detection accuracy. The suggested IFF module accurately captures features and detailed information of small tools while using a low computation. Furthermore, the use of an ASH module speeds up the convergence of the LSTD and enhances detection accuracy. Overall, the LSTD model demonstrates promise for managing robot operations in unstructured environments as well as presents insightful information for small tool detection development in the future.

3. Methodology

Following several iterations of development, the YOLO series has grown to be a well-liked family of object detection frameworks. YOLOv5, an anchor-based, one-stage detection method, is renowned for its excellent accuracy and quick detection speed. Ultralytics made YOLOv5 publicity available, offering four distinct scale variants. The structure of YOLOv5, which consists of a head, neck, as well as backbone, is shown in Fig. 1. In order to extract features from the input, the backbone component downsamples the input four times. The neck component uses the Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) architectures. YOLOv5's head structure consists of three linked heads. We used YOLOv5 as the basis for our study's LSTD algorithm, which we built as the baseline.

Table 1
Specifics of the LSTD output size of the feature, component, and connection technique.

No.	Module	From	Output size
0	CBR	-1	[32, 320, 320]
1	CBR	-1	[64, 160, 160]
2	RICC_v3	-1	[64, 160, 160]
3	CBR	-1	[128, 80, 80]
4	RICC_v3	-1	[128, 80, 80]
5	CBR	-1	[256, 40, 40]
6	RICC_v3	-1	[256, 40, 40]
7	AP	-1	[256, 40, 40]
8	CBR	-1	[128, 40, 40]
9	UpSample	-1	[128, 80, 80]
10	Concatenation	[−1, 4]	[256, 80, 80]
11	RIC	-1	[128, 80, 80]
12	CBR	-1	[128, 40, 40]
13	Concatenation	[−1, 8, 6]	[512, 40, 40]
14	RIC	-1	[256, 40, 40]
15	ASH	[11, 14]	[128, 80, 80] [256, 40, 40]

The architecture of our suggested LSTD is shown in Fig. 1. The three parts of LSTD are the Accurate Separated Head (ASH), the Integrated Feature Fusion (IFF), and the Dynamic Feature Extraction (DFE). The three primary modules of DFE are RICC (Robust Integrated Convolution based on CBAM), CBR(Convolution, Batch Normalization layer, ReLU function), and Adaptive Pooling (AP). The CBR, RIC (Robust Integrated Convolution), Concat, and UpSample modules make up the majority of IFF. The CBR module and 1×1 convolution make up the majority of the ASH.

Table 1 provides a detailed representation of the LSTD feature map variation, connecting components, as well as network composition. The model is small, with only 16 specially designed components. The entering information flow layer is indicated by the second column, where -1 denotes the layer that came before it. Customized modules are shown in Table 1's third column. The resulting feature map's dimensions—width, height, and number of channels—are listed in the last column. For instance, the feature maps from rows No. 9 and No. 4 are subjected to a Concat operation, as shown by the item [-1, 4] in row No. 10 of the table. A feature map with size [512,80,80] is produced by this process.

3.1. Dynamic Feature Extraction

The number of modules in DFE decreased and downscaled input images multiple (2, 4, 8, and 16) in order to address the challenges presented by the decrease in feature

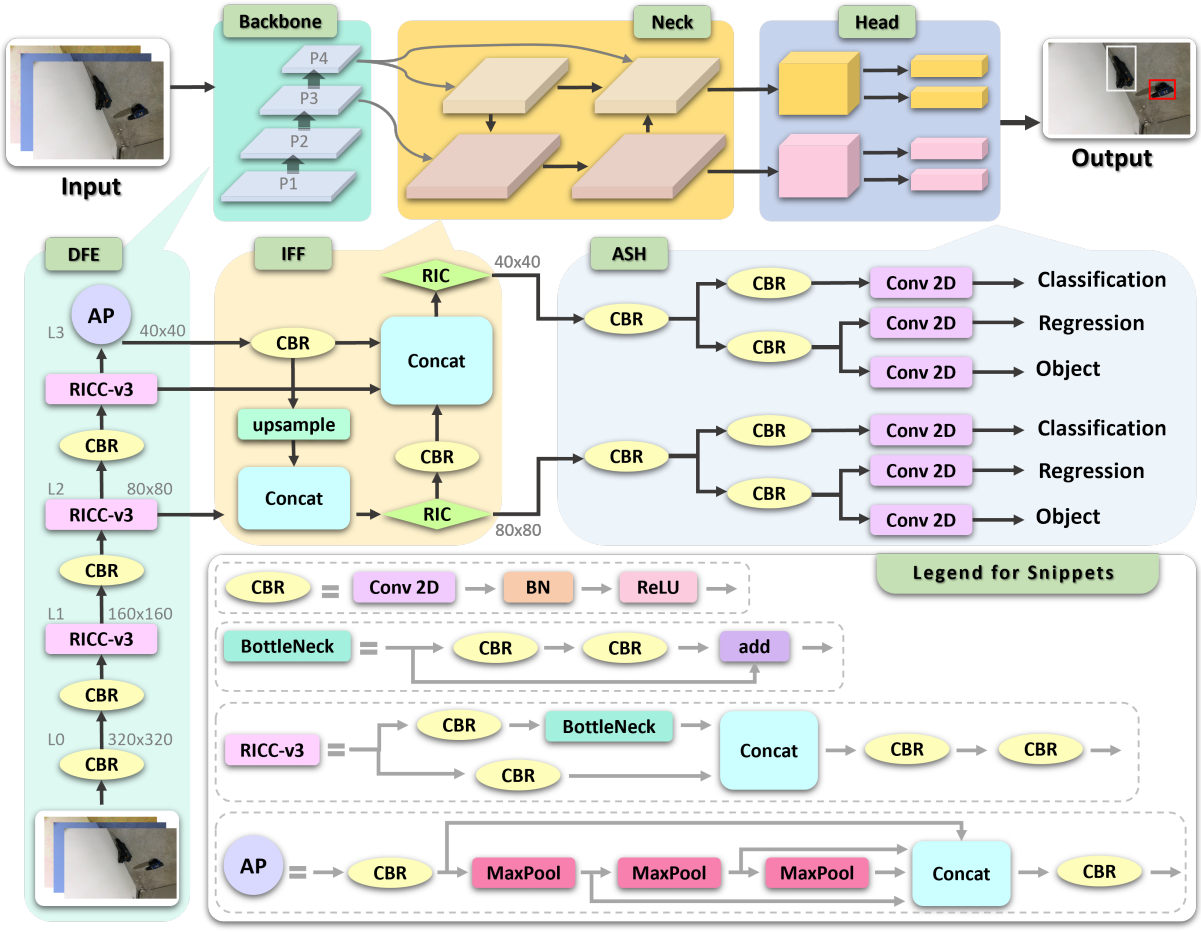

Figure 1: Overview of the LSTD architecture diagram based on YOLOv5. Three main parts make up the LSTD architecture, similar to YOLOv5: head, neck, and backbone. Components of the LSTD architecture are Dynamic Feature Extraction (DFE), Integrated Feature Fusion (IFF), and Accurate Separated Head (ASH).

dimension with more layers as well as the possible information loss brought on by smaller objects. By doing this, the feature maps' detail loss is decreased and smaller targets may be represented more accurately. We also included an attention technique to extract important information in an adaptable manner. DFE maintains a lightweight design while concentrating on useful feature information.

3.1.1. DFE Structure

Four layers make up the DFE, as seen in Fig. 1: one Lead Layer (L0), and three Level Layers (L1, L2, L3). A 6×6 convolutional kernel is present in the Lead Layer, which is a CBR module. It removes operations like channel concatenation and slicing in comparison to the the baseline, which lowers the amount of parameters and computational cost. Every CBR module and every RICC module make up the initial pair of Level Layers (L1, L2). ReLU activation function, Batch Normalization layer (BN), and Conv2d with a 3×3 filter size make up the CBR module. Finally, the Level Layer (L3) includes the AP module. The two CBR components with a 1×1 filter size and the three MaxPooling modules with a 5×5 filter size make up this AP layer. In baseline, the Average Pooling module is less efficient than the AP module in capturing multi-scale contextual information. From Level Layers (L2) and Level Layers (L3), DFE creates an output with the size [4, 512, 40, 40] and [4, 256, 80, 80], which are then sent to IFF.

3.1.2. RICC Modules

Figure 2 illustrates the three RICC modules that we suggested in this study, namely RICC_v1, RICC_v2, and RICC_v3, based on the [37]. These modules are critical to receptive field extension, adaptive augmentation, and feature extraction.

RICC_v1: The input is initially processed via a Conv2d with filter size 1 x 1 in a CBR module, after which the output is sent to two routes. The branch path is unprocessed, while the main route passes via CBR modules with Conv2d with filter size 3×3. Ultimately, a CBAM module receives the combined output feature maps from the two pathways.

RICC_v2: Initially, a Conv2d with filter size 1 x 1 CBR module with the input feature map reduces the channel dimension by half. Subsequently, the output is sent to two routes: the branch route remains unprocessed, while the

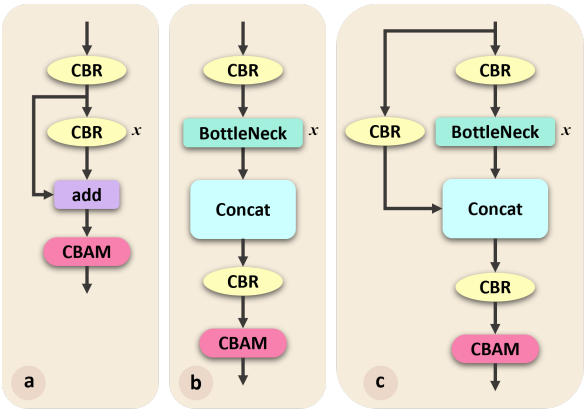

Figure 2: Three suggested RICC modules are illustrated. (a) RICC_v1, (b) RICC_v2, (c) RICC_v3.

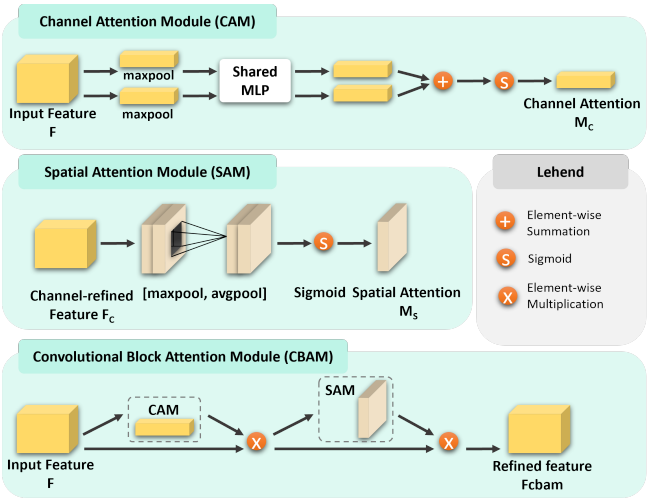

Figure 3: Overall architecture of CBAM that contains SAM and CAM.

primary route passes via the Bottleneck component. Subsequently, the two pathways' outputs are joined in the axis of the channel. After that, the fused output passes via a convolutional with kernel size 1×1 in a CBR module to increase the channel dimension to the intended feature map of output. Lastly, a CBAM component is used to filter the spatial features as well as feature channels.

RICC_v3: The input initially follows multiple routes. The primary route passes via the BottleNeck module after passing via a CBR component with Conv2d with kernel size 1×1 to decrease the channel dimension. The channel dimension is further decreased by the branch route, which passes via a CBR component Conv2d with kernel size 3×3. These two pathways' channel outputs are then concatenated. After that, the integrated feature passes through a Conv2d with kernel size 1×1 in a CBR component to increase the channel dimension to the intended output channels. Lastly, the spatial coordinates and feature channels are weighted using a CBAM module. Furthermore, the BottleNeck component has a residual design in which the input passes via a shortcut link after passing via two Conv2d layers with kernel size 3×3. The ultimate output, a feature map, is subsequently created by adding the input data to it.

After doing comparative studies on three RICC modules, we decided to include the RICC_v3 module in DFE; the specifics are provided in Section 3.3. There are variations in the number of BottleNecks in the three Level Layers (L1, L2, and L3) of the RICC_v3 module. L1, L2, and L3 specifically used 1, 2, and 3 bottlenecks, respectively, with 1, 2, and $3 \times$ values in line.

3.1.3. CBAM

As noted above, in order to increase the accuracy of small object detection, we decreased the number of network layers. As a result, the contextual understanding of the feature was weakened. Furthermore, the same background interferences have a major impact on the proper identification of small construction tools. Therefore, in order to improve recognition ability and concentrate on useful feature information, we added an attention module to the different Level Layers (L1, L2, and L3) of the DFE. In addition, the Convolution Block Attention Module (CBAM), suggested by Woo et al. [38], distinguishes itself from [39], [40], and [41] by being a lightweight attention module. It possesses the capability to adaptively boost the expressive capacity of crucial features of spatial dimension and channels. The two submodules that make up CBAM are the CAM and SAM, as seen in Fig. 3. First, CAM infers a feature map $M_c \in \mathbb{R}^{(C\times1\times1)}$ from the input feature map $F \in \mathbb{R}^{(C\times H\times W)}$. SAM then infers a feature map $M_s \in \mathbb{R}^{(1\times H\times W)}$. We included attention methods to improve recognition ability and concentrate on useful feature information in the three Level Layers (L1, L2, and L3) of the DFE.

'What' is significant in relation to an input is the focus of the channel attention. First, average pooling and max pooling processes are used to aggregate the input feature map for spatial information. The shared multi-layer perceptron (MLP) receives the aggregated feature map after that. Next, the resultant feature vectors are combined using element-wise summation. The sigmoid activation function is the final step in obtaining channel attention feature maps. To put it briefly, channel attention is calculated as follows:

$$M_c(F) = \sigma\left(\mathrm{MLP}\left(\mathrm{AvgPool}(F)\right) + \mathrm{MLP}\left(\mathrm{MaxPool}(F)\right)\right) \tag{1}$$

The Sigmoid activation function operation is represented by σ in the formula, the shared perceptron operation by MLP, and the global average pooling and maximum pooling operations by Avg-Pool and Max-Pool, respectively.

Where is an instructive portion of the feature map that receives spatial attention. First, two $(1 \times H \times W)$ feature maps are created from the channel attention module's output using the max pooling and average pooling processes. Next, a 7×7 convolution layer concatenates and convolves the feature maps.

Ultimately, a 2D spatial attention map is created by using the Sigmoid function to make the spatial attention output. To put it briefly, spatial attention is calculated as follows:

$$M_s(F_c) = \sigma\left(f^{7\times7}\left([\text{AvgPool}(F_c);\text{MaxPool}(F_c)]\right)\right) \quad (2)$$

A 7×7 convolution process is represented in the formula by $f^{7\times7}$. The CAM processes the feature F first, producing the output F_c in the channel dimension. The SAM processes the feature F to make the F_{cbam} in the spatial dimension. One way to sum up the attention process generally is as follows:

$$F_c = M_c(F) \times F$$
$$F_{cbam} = M_s(F_c) \times F_c$$

3.2. Integrated Feature Fusion

The middle layer of the network architecture, known as the "feature fusion," creates feature maps containing multi-scale information and is utilized for feature fusion and information transfer across various layers. This research proposes an Integrated Feature Fusion (IFF), which can help the model generate accurate features with fewer parameters. The feature fusion network is modified. Through examination of the information in Section 3.2, we see that small construction tools detection exhibit little variance in size and are generally modest in size. As a result, using the full FPN and PAN as seen in Fig. 4(a) is not required. Given the low pixel percentage of small tools, we eliminated the 32x downsampling layer from the PAN as well as FPN architecture, which is the lowest feature layer, in order to decrease the model size and improve flexibility. Fig. 4(b) depicts this structure. While making the model lighter and reducing computational complexity, simplifying the feature fusion network's structure may also make features less capable of being represented. Thus, as seen in Fig. 4(c), we created Integrated Feature Fusion (IFF) at the top layer based on the simplified network. In order to fuse multi-scale properties, IFF uses bidirectional connections. To be more precise, the bottom-up pathway uses downsampling to convey low-level detail information, whereas the top-down pathway uses upsampling to communicate high-level semantic information. Both high and low-level semantics are included in the fused feature. In order to get deeper semantic information and minimize detail loss, IFF also uses integrated links to combine features from higher levels. Two CBR modules, two RIC modules, one UpSample module, and two Concat modules make up IFF. For details on the precise arrangement and connections, please see Fig. 1. It is important to note that RICC is a reduced version of the RICC structure in DFE. For example, the final CBAM module is not present, and the BottleNeck module lacks a shortcut connection. Lastly, in accordance with the input from DFE, IFF sends two feature maps to the ASH. The corresponding tensor forms are (4, 256, 80, 80) and (4, 512, 40, 40). As a result, using the full FPN and PAN architectures as seen in Fig. 4(a) is not required.

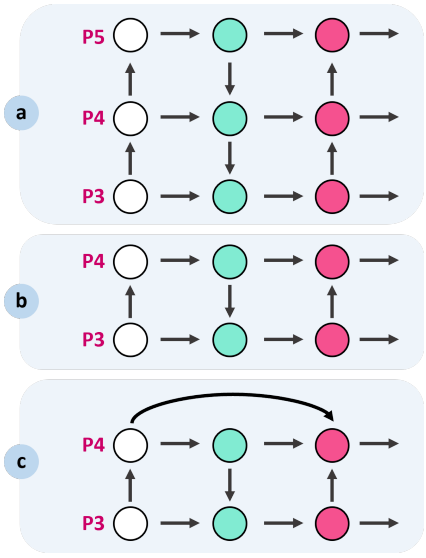

Figure 4: Overall Schematic of feature fusion. (a) FPN) + PAN, (b) Simplified IFF, (c) IFF.

3.3. Accurate Separated Head

The discrepancy between the regression and classification tasks is a major problem in object detection. The baseline's coupled detection head shares parameters with the localization as well as classification branches. However, employing common parameters may result in spatial misalignment problems because of the somewhat uneven focus of the localization as well as classification tasks [42]. Ge et al. [43] experiments have demonstrated that switching out the YOLOv5 connected head for a decoupled one may greatly increase convergence speed and improve detection performance. ASH eliminates the Object branch and separates the regression and classification branches for independent prediction. Furthermore, in contrast to YOLOX, we further minimize model complexity and inference delay by lowering from 2 to 1 the number of conv2d with filter size 3 × 3 on both routes.

Fig. 5 shows how the ASH is structured. The following are the particular operations: First, a Conv2d with kernel size 1 × × 1 is used to decrease the channel dimension of the IFF feature to 128 and 256, respectively. After that, it is divided into the regression as well as the classification branch, two parallel branches. A Conv2d with kernel size 3 x 3 is present in each branch for tasks involving regression and classification, respectively. The regression branch is expanded with an extra Object branch, and each branch is then subjected to an additional 1 × 1 convolution process. Furthermore, the regression branch forecasts the target's Object (confidence information) and regression (bounding box information), while the classification branch is in charge of forecasting the target's Classification (classification information). With two effectively separated heads, LOSD produces two different final output tensor shapes: (4, 7, 80, 80) and (4, 7, 40, 40).

This work proposes the anchor-based object identification algorithm LOSD. By employing the scale of the objects

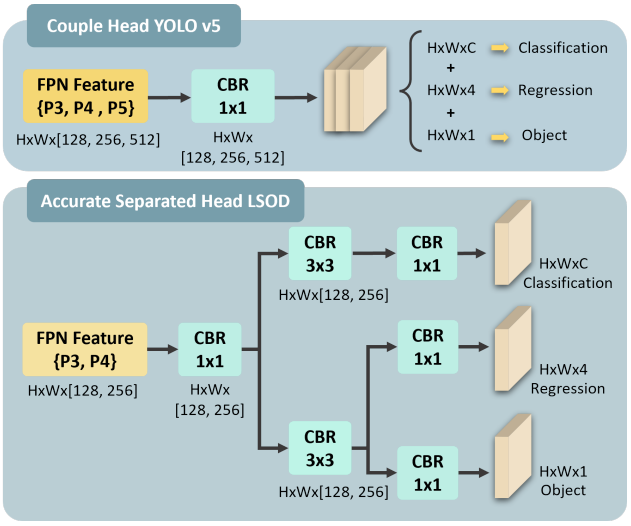

Figure 5: Overall structure of YOLOv5 head and the proposed ASH.

as the categorization metric, the anchor sizes are derived by categorization. This work uses the auto anchor technique to autonomously create as well as cluster anchor sizes depending on the inputs, developing them using a genetic method, as opposed to predefining anchor templates. Six sets of anchors are generated after K nearest neighbor clustering on 30,432 points: (25,25), (30,29), (35,45), (38,338), (45,42), and (52, 50).

3.4. Dataset

The dataset utilized in this study consists of images of 12 different small construction tools, including cutters, buckets, hammers, knives, saws, shovels, tackers, drills, grinders, spanners, and wrenches [11]. These tools were carefully selected based on their frequent usage in indoor construction sites, as determined by analyzing construction standard specifications and interviews with site managers [11]. To capture the diversity of appearances, sizes, shapes, colors, and backgrounds encountered in real-world construction environments, the dataset comprises 34,738 images. Approximately 18% (6,258 images) were acquired directly from actual construction sites, ensuring the inclusion of realistic conditions such as occlusions, varying illumination, and worker interactions. The remaining 82% (28,480 images) were captured with various controlled backgrounds like construction background sites to further enhance the dataset's diversity. The dataset was meticulously annotated with bounding boxes, indicating the location and class of each tool instance. The images were carefully curated to include variations in resolution, occlusion, lighting conditions, and backgrounds, factors known to influence the performance of object detection methods [44, 45]. The dataset was divided into training (60%, 20,842 images), validation (20%, 6,948 images), and test (20%, 6,948 images) sets. This dataset was constructed with the goal of improving object detection model performance while accounting for factors such as background diversity, illumination changes, occlusions, and resolution variations, all of which are prevalent in challenging construction site environments.

3.5. Experiment

Windows 11 is the operating system utilized in this paper, and CUDA version 11.8 is employed. The machine used for the trials included an Intel Core i7 13620H CPU and an NVIDIA GeForce RTX 4060 Laptop GPU. PyTorch 1.10.1 is used with Python 3.9 as the development language. The Adam optimizer [46] was used, with an initial learning rate of 0.0009 and a Cosine learning rate decay strategy. A weight decay of 0.0005 was applied to regularize the model. The loss function was the Focal Binary Cross-Entropy, with gamma set to 2 and alpha set to 0.25 for focal weighting. The model was trained for 200 epochs, with a batch size of 4. The detection findings can be categorized as true positive (TP), false positive (FP), true negative (TN), and false negative (FN) based on these studies. We present all the measures that are utilized in this research, such as FLOPs, mean average precision (mAP), recall (R), precision (P), and F1 score. In particular, recall (R) and precision (P) are defined as:

$$P = \frac{TP}{TP + FP} \times 100\%$$
$$R = \frac{TP}{TP + FN} \times 100\%$$

here recall is calculated by dividing the number of true positives by the sum of the true positives and false negatives, and precision is calculated by dividing the number of true positives by the sum of the true positives and the erroneous positives.

The AP for many categories is referred to as the mAP, and AP is defined as follows:

$$AP = \int_0^1 p(r)\, dr \tag{3}$$

In addition, the average mAP over various intersection over union (IoU) thresholds (from 0.5 to 0.95, step 0.05) is represented by mAP@0.5:0.95.

The mAP and recall, or F1-score, is a useful metric for assessing a model's overall performance in detection tasks. The F1-score is defined as follows and its value goes from 0 to 1.

$$F1\text{-}score = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

FLOPs is a measure of how many floating point operations the model needs to perform to simulate the output. It is an important indicator of the complexity of the model and can be used to compare to other models.

4. Result and Discussion

We carried out eight tests to assess the effectiveness of the suggested LOSD model, which are detailed in the results section.

Table 2
Influence of augmentation on the LSTD.

Dataset type	P(%)	R(%)	mAP(%)
Original	82.4	80.8	84.2
Augmented	85.0	83.5	87.3

Table 3
Comparison evaluation of three network architectures, PAN + FPN, DFE + FFN, and DFE + IFF on the test dataset.

Architecture	Layer	mAP (%)	Param (M)	FLOPs (G)
PAN+FPN	157	80.1	6.69	15.6
DFE + Simple IFF	107	83.2	1.79	11.1
DFE + IFF	107	83.9	1.85	11.3

Table 4
Comparison evaluation of three robust integrated convolution modules.

Component	mAP (%)	R (%)	P (%)	Param (M)
RICC_v1	83.4	81.3	82.2	4.10
RICC_v2	83.5	82.1	81.4	3.40
RICC_v3	87.3	83.5	85.0	2.87

Table 5
Utilization of different attention mechanisms in the LSTD and examination of their results.

Module	P (%)	R (%)	mAP (%)	Param (M)
Baseline w/o attention	81.2	80.6	84.7	2.86
SE	83.6	81.3	85.4	2.87
CA	82.4	83.6	86.1	2.87
SimAM	83.2	81.3	85.0	2.86
CBAM	85.0	83.5	87.3	2.87

4.1. Data Augmentation Effect

We validate our approach on the original dataset and on-the-fly (online) augmented dataset in order to look into the effects of data augmentation methodologies on the metrics. The augmentation strategy comprises various operations, including horizontal flipping, median blur, spatial shifting, adjustments in brightness and HSV, mosaic application, and the incorporation of Contrast Limited Adaptive Histogram Equalization (CLAHE) and simulated fog effects. The training dataset was the sole variable in the experimental setting, with all other parameters remaining constant. Table 2 displays the experimental outcomes. On-the-fly augmentation produced gains of 2.3% in recall, 3.4% in precision, and 3.1% in mAP over the original dataset. With recall, precision, and mAP of 83.5%, 85.0%, and 87.3%, respectively, the On-the-fly augmentation approach produced noteworthy gains in all measures in comparison to the original dataset. Thus, the On-the-fly augmentation technique used in this work successfully improves LSTD's detection performance.

4.2. Effect of Different Module

This study made several changes to the baseline model's structure according to the traits of small tools detection and the requirement to improve construction safety. We validated our approach on the suggested IFF to confirm the viability and efficacy of the changes. It should be noted that the baseline model was used for these trials, and no further modifications indicated in the study were used; instead, the only emphasis was on structural validation. Table 3 shows that the number of parameters and layers dropped to 6.69 M and 107 respectively, as well as the FLOPs dropped by 4.5 G after the 32x downsampling layers in the baseline model's neck and backbone were removed. Meanwhile, the mAP significantly increased by 2.1%. The mAP rose from 86.2% to 87.3% with IFF, while the FLOPs and number of parameters increased slightly to 11.3 G and 1.85 M, respectively. IFF adds top-layer integration connections in comparison with FFN. The DFE + IFF architecture obtained a 7.2% gain in mAP, a 73% decrease in parameters, as well as a 29% reduction in computation when compared to FPN + PAN. Thus, it can be said that despite significantly lowering the number of parameters, the suggested DFE + IFF architecture enhances small tools detection ability.

The experimental outcomes of the three robust integrated components suggested in the LSTD architecture are shown in Table 4. It is evident that the best detection performance is obtained when the RICC_v3 module is utilized. It uses the fewest parameters while achieving the best accuracy, recall, and mAP when compared to RICC_v1 and RICC_v2. For this reason, we decided to use the RICC_v3 module in this study to extract features from DFE.

4.3. Evaluating various attention mechanisms on LSTD

In this research, attention methods are included in the three Level layers (L1, L2, and L3) of DFE to improve the feature extraction capabilities. Using the expanded dataset and the suggested LSTD model, we carried out five comparison experiments to investigate the efficacy of attention modules in small tools detection: with the SE module, the CA module, the SimAM module, the CBAM module, and without the attention module. Table 5 shows that the baseline using the CBAM had the greatest mAP (87.3%) and precision (85%) while the baseline using the CA had the highest recall (83.6%). It is important to note that adding attention modules to any experiment almost completely prevented the model's parameter size from growing. In general, the CBAM-equipped LSTD model performs the best. The explainable results of LSTD employing various attention modules are shown in Fig. 6.

Fig. 6 depicts the activation maps produced by the explainable Grad-CAM method in both simple and complicated situations, respectively. It is evident from the numbers that LSTD primarily targets small construction tools. Small construction tool identification is greatly impacted by comparable backgrounds, as was seen in the prior loss study. Additionally, the goal of this study is to develop a cutting-edge small tools detection model for monitoring safety and robots, specifically designed to identify and discriminate among

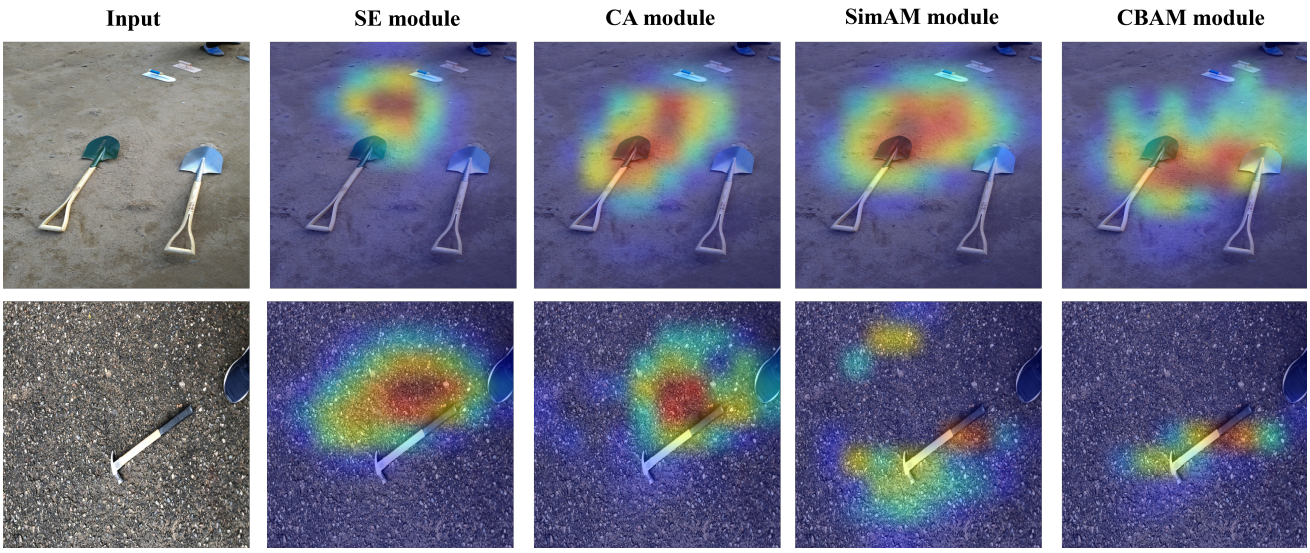

Figure 6: Grad-CAM visualizations generated by various attention modules.

various tools. To achieve this precision, the algorithm will be tailored to differentiate between tools positioned in the foreground and those in the background. Also, LSTD shows off a creative method for differentiating between small tools in the background and foreground. This feature of LSTD highlights how it may mimic some parts of human cognitive capacities, including perception and decision-making processes. Moreover, another area of interest for LSTD in the picture is the area containing small tools. It is evident from the class activation maps that distinct attention modules show differing levels of concentration on the small tools. In both complicated and straightforward circumstances, the LSTD architecture with the CBAM provides a more precise focus on the object zone and places greater attention on small tool regions. The SimAM and CA concentrate on specific locations in the complicated scenario, but the SE module has a lesser concentration on small tools. When compared to the CBAM, the SimAM, SE, and CA focus less on the small tool area in the simple scenario. According to the experimental results, the LSTD with the CBAM is able to identify small tools in the background and foreground more clearly, as well as concentrate and focus more effectively in that area.

4.4. Ablation Study

We carried out tests to evaluate the model's performance incrementally after each modification in order to further examine the efficacy of the improvement methodologies suggested in this research. Table 6 displays the test procedure and ablation experiment outcomes. The data shows that the B model, which uses the DFE + IFF architecture, reaches a 73% decrease in parameters to 1.85 M from 6.69 M while maintaining a little greater recall and accuracy compared to the baseline. Based on the B model, the C model increases the number of parameters by a small amount and improves recall, accuracy, mAP@0.5:0.95, and mAP. This is achieved by adding the ASH component. The CBAM module was added to the C model to create the LSTD model, which exhibits improvements in mAP@0.5:0.95, mAP, and precision but a minor decline in recall. In addition, our suggested LSTD model outperforms the baseline (A) model by 7.6%,7.2%,4.9%, and 6.8% in mAP@0.5:0.95, mAP, recall, and precision, achieving 77.8%, 87.3%, 83.5%, and 85%, respectively, with just 2.87 M parameters—a 57% decrease. The ablation experiment results show how efficient the enhancement tactics suggested in this study are, especially when it comes to lowering the number of parameters and improving detection accuracy. Also, the ASH component adds more to recall, the CBAM enhances accuracy more, and the DFE + IFF architecture has a bigger effect on the parameters.

In addition, LSTD, representing the fully augmented model, emerges as the top-performing method across all tools as shown in Fig. 7. For instance, LSTD achieves an accuracy of 82.87% for Cutter (CU) and 84.16% for Hammer (HA), outperforming configurations A, B, and C. Notably, the introduction of attention-guided spatial highlighting (ASH) and Convolutional Block Attention Module (CBAM) consistently contributes to performance improvement. For tools like Grinder (GR), LSTD attains remarkable accuracy at 90.18%, underscoring the efficacy of the proposed method. The attention mechanisms, particularly CBAM, play a pivotal role in elevating overall performance, evident in the substantial improvements from configuration B to LSTD across various tools. This ablation study provides valuable insights into the cumulative impact of depthwise feature enhancement, instance-level feature fusion, attention-guided spatial highlighting, and CBAM in enhancing small tool detection in construction site scenarios.

4.5. LSTD test results

The LSTD model's confusion matrix is displayed in Fig. 8. The anticipated labels are shown by the vertical axis, and the genuine labels are represented by the horizontal axis. The major diagonal probabilities indicate the likelihood that each

Table 6
Effects of each component on the performance of LSTD.

Model abbreviation	Components					Metrics				
	Baseline	DFE	IFF	ASH	CBAM	mAP@0.5:0.95 (%)	mAP (%)	R (%)	P (%)	Param (M)
A	✓					70.2	80.1	78.6	78.2	6.69
B	✓	✓	✓			74.2	83.9	80.1	80.2	1.85
C	✓	✓	✓	✓		77.2	84.7	80.6	81.2	2.86
LSTD	✓	✓	✓	✓	✓	77.8	87.3	83.5	85.0	2.87

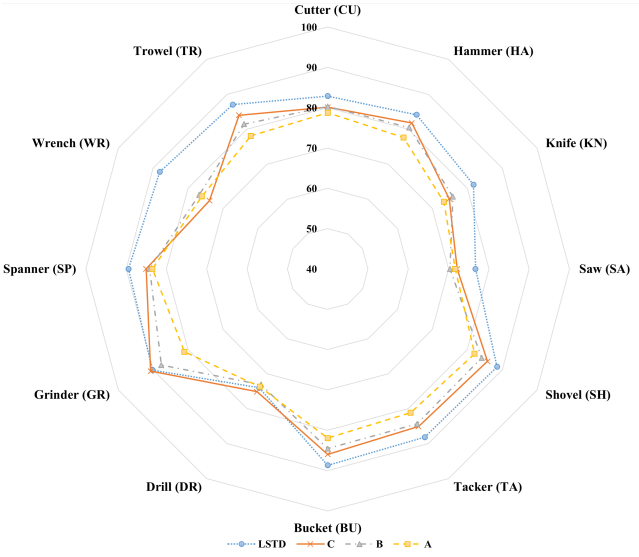

Figure 7: Radar chart for each category of objects in the test dataset, with different modules represented by different colored lines based on Table 6, displaying precision values.

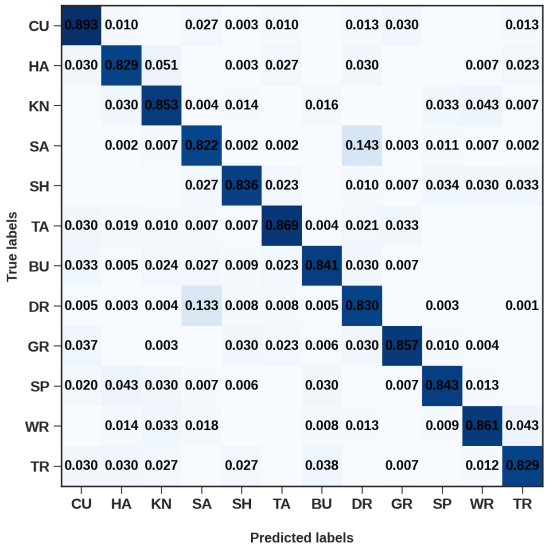

Figure 8: Comprehensive insight into Model Performance by Confusion matrix that shows the accuracy of each object.

category will be correctly classified. The precision of misclassification is shown by the numbers off the main diagonal, which indicates that overall, misclassification happens less frequently.

Using mAP as the assessment metric, we ran seven iterations of tests on the LSTD and baseline models to verify the validity of the training outcomes. Analysis of variance was used to examine the experimental outcomes for small objects, and this indicates that the difference between the total means of the two approaches was statistically significant (significant level of 0.05) and that it could be concluded that the results had a meaningful difference. The LSTD and Baseline models differ significantly from one another, indicating the dependability of our suggested approach.

Fig. 9 displays the LSTD result in the following unstructured environments: (a) standard illumination, (b) intense illumination, (c) subdued illumination, and (d) misty conditions. Although the environment has a notable effect on the model performance, LSTD is still able to identify the categories and detect small tools under conditions like misty conditions, intense illumination, and subdued illumination. It is evident that the suggested LSTD model has strong resilience and flexibility in intricate external contexts. The goal of this research is to create a detecting algorithm that can be used for robots or monitoring safety.

Another crucial metric for assessing the model's effectiveness is its capacity for generalization. Three distinct tool kinds, namely big (closer to camera) tool, medium, and very small, were utilized to obtain the generalization precision of the LSTD model, as seen in Fig. 6. The graphic shows that the model can identify tools from the background and recognize and categorize big tools with accuracy. Although the algorithm can detect the tools quite correctly. The LSTD works well for very small and big tools simultaneously.

Three distinct datasets of construction site settings, including intense illumination, subdued illumination, and misty conditions, were produced in order to evaluate the detection ability of the LSTD model in difficult scenarios. The particular test results are listed in Table 7. The subdued illumination situation yielded the best result (precision = 84.7%) and F1-score (84.1%) for small tools detection. Overall, the LSTD showed high accuracy in situations with intense illumination, subdued illumination, and misty conditions; however, scenarios with misty conditions and intense illumination had a greater impact on the detection performance.

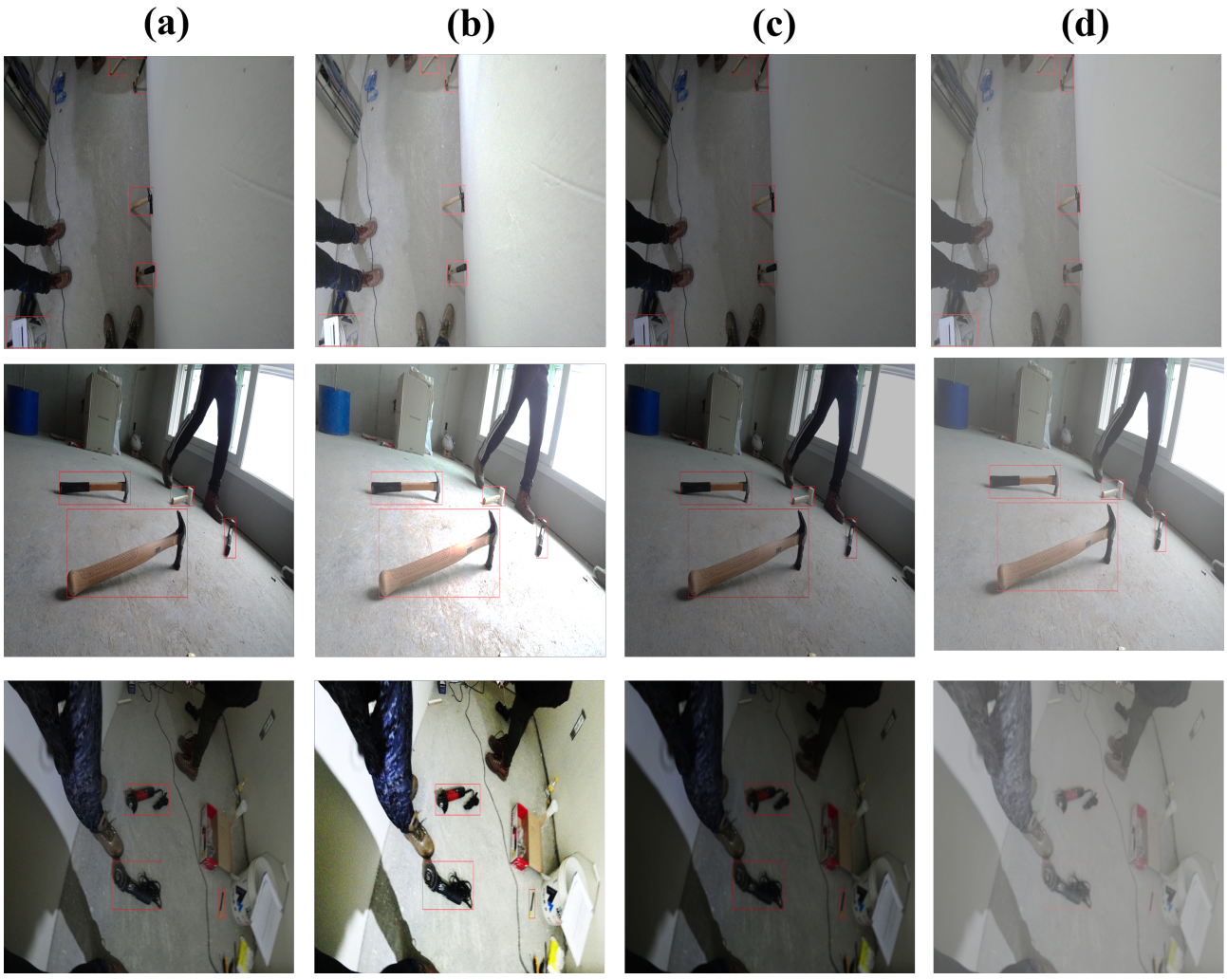

Figure 9: LSTD model's performance has proven to be consistent across different environment conditions, including (a) standard illumination, (b) intense illumination, (c) subdued illumination, and (d) misty conditions. The image shows a cluttered construction site with scattered tools in workplace, including hammers, knives, and cutters on the floor. This presents a significant tripping hazard for workers moving through construction sites.

Table 7
Effects of different brightness conditions on the performance of LSTD.

Dataset	P (%)	R (%)	mAP (%)	F1-score(%)	CDR(%)	EDR(%)	MDR(%)
intense illumination	82.2	82.6	85.7	82.4	84.6	6.4	15.4
subdued illumination	84.7	84.1	86.1	84.4	84.1	5.8	15.9
misty conditions	83.7	81.8	86.4	84.7	83.9	7.1	16.1

4.6. LSTD Robustness

Various external noises might create interference during the actual small tools detection procedure. For instance, problems like too little or too much illumination or misty conditions might exist. As a result, the detection algorithm that is created must be very resilient to noise and flexible. The LSTD model's detection ability in various construction site conditions will be evaluated in the future using four test datasets: misty conditions, intense illumination, standard illumination, and subdued illumination. Also, the normal light dataset's photos are subjected to brightness reductions (-30%), brightness enhancements (+30%), and fog additions (+30%) to produce the subdued illumination, intense illumination, and misty conditions datasets, respectively. As a result, for every extensive experiment, the quantity, classes, as well as locations of the labeled BBs for associated images are all the same. As seen in Fig. 9, we displayed the LSTD model's detection results for small tools in various environmental conditions. Although there are a few cases of wrong detection, the image illustrates

Table 8
Comparison of LSTD performance with state-of-the-art models.

Model	mAP (%)	Param (M)
Efficientnetv2	80.6	20.67
Lcnet	80.4	1.96
Ghostnet	80.2	1.08
Mobilenetv3	78.5	1.97
YOLOv8 - Small	84.3	11.15
YOLOv7 - Tiny	83.1	6.03
YOLOv6 - Small	83.2	17.18
YOLOv5 - Small	80.1	6.69
YOLOv3 - Tiny	82.2	2.18
LSTD (ours)	87.3	2.87

how well our suggested LSTD model can detect small tools in various settings. The experimental findings show that while intense illumination in the construction site has a significant influence, misty conditions have little effect on the LSTD model's detection ability. Thus, one avenue for future development is to enhance the detection accuracy in misty conditions.

4.7. Comparisons with state-of-the-art model

We contrasted the LSTD model with the most advanced object identification techniques in order to further corroborate the effectiveness of the suggested methodology.

Table 8 lists the models that are being compared. According to Table 8, LSTD reaches the maximum mAP value of 87.3%. LSTD demonstrates higher mAP values, showcasing improvements of 4.1% compared to the recently published YOLOv7-Tiny [47] when the number of parameters decreases by 52.3%. Similarly, when contrasted with YOLOv6-Small[48], LSTD reveals a notable mAP increase of 2.2%, emphasizing its enhanced detection performance with significant reductions in the number of parameters to 83.3% against YOLOv8-Small, LSTD exhibits a substantial improvement of 3.0% in mAP, highlighting its superior object detection capabilities when the number of parameters decreases by 74.2%. Furthermore, although LSTD has a bit more parameters than Ghostnet [49], YOLOv3-Tiny [50] Mobilenetv3 [51], and Lcnet [52], the LSTD mAP is higher than them significantly. As a result, the LSTD's high mAP and comparatively limited number of parameters lead to an impressive overall performance.

4.8. Discussion

Accurate small tool detection in unstructured construction sites is difficult due to a number of factors, including illumination and mistiness. The goal of this work is to create a detection architecture for monitoring safety and robots. In order to tackle these problems, a LSTD method for detecting small tools in intricate and unstructured construction sites is suggested. With fewer parameters and computation, this algorithm's unique neural network design delivers excellent detection accuracy. Furthermore, enhancements to the head and backbone networks efficiently mitigate interference. The following are the specific contributions made by this paper: (1) To increase the generalization and resilience of the model, on-the-fly data augmentation techniques are used to semantic enlarge the original dataset. (2) DFE is proposed To improve the feature extraction performance for small tools. (3) To achieve lighter weight and richer feature representations, an IFF is suggested. (4) To enhance the background interference discriminating capability, an ASH is employed. In comparison to sophisticated object identification algorithms, the suggested technique achieves greater detection accuracy for small tools in complicated construction site conditions while displaying superior robustness, adaptation, and generalization in unstructured sites. This approach may also be used in monitoring safety and robots that are used in construction sites.

The lightweight and efficient nature of our proposed LSTD model, demonstrated by the significant reduction in parameters (73%) and computations (28%) compared to YOLOv5, makes it a promising candidate for integration into existing safety monitoring systems or robotic platforms used on construction sites. Although explicit evaluations on edge devices were not conducted in this study, the low computational requirements and compact architecture of our model suggest its suitability for deployment on resource-constrained devices or embedded systems commonly found in construction site monitoring setups. The ability to accurately detect and localize small construction tools in real time is crucial for enabling proactive safety measures and interventions. By integrating our LSTD model into on-site monitoring systems, potential hazards such as tools being dropped, misplaced, or left in high-risk areas can be identified in a timely manner. This real-time awareness can trigger various safety protocols and warning mechanisms to mitigate risks and prevent accidents. For instance, upon detecting a tool in an unauthorized or hazardous zone, automated alerts or notifications can be sent to site supervisors or workers in the vicinity, prompting immediate action to secure the tool or evacuate the area if necessary. Additionally, real-time tool tracking can support the implementation of standardized tool storage and management procedures, ensuring that tools are properly accounted for and stored in designated areas when not in use. Furthermore, the integration of our LSTD model with robotic platforms or autonomous systems employed for construction site monitoring can enable autonomous detection and response to potential hazards. Robots equipped with our model could actively patrol the site, identifying and flagging instances of misplaced or dropped tools, and even potentially retrieving or securing them to prevent accidents.

While the current study focuses on demonstrating the lightweight and accurate performance of our LSTD model, future work should involve explicit evaluations on various edge devices and embedded systems commonly used in construction site monitoring scenarios. By assessing the model's inference times and resource requirements on these target platforms, we can further validate its real-time capabilities and suitability for deployment in practical safety

monitoring applications. Moreover, additional research can explore the seamless integration of our model with existing safety monitoring systems, robotic platforms, and site management software. Developing user-friendly interfaces, alert mechanisms, and decision-support tools based on real-time tool detection outputs can facilitate the adoption of our approach and enhance its practical impact on construction site safety.

The need for lightweight models for detecting small objects in construction sites arises from several practical considerations. In this case, construction sites often have limited access to high-performance computing resources, making it challenging to deploy computationally intensive models on-site. Lightweight models can be more easily deployed on edge devices or embedded systems with modest hardware capabilities. To put it in another way, edge devices and robotic systems used for on-site monitoring may have power and battery constraints, making it essential to use efficient models that minimize energy consumption while maintaining high accuracy. Also, in many construction safety scenarios, real-time monitoring and immediate detection of potential hazards are crucial. Small tools, despite their size, can pose significant risks if mishandled or left unattended. A lightweight model can enable faster inference times, allowing for more responsive safety monitoring and hazard detection. Specific examples where immediate detection of small tools is essential include:

- Small tools inadvertently dropped from heights can pose serious risks to workers below. Immediate detection of such incidents can trigger alerts or safety measures to prevent injuries.
- Certain tools, if used incorrectly or in unauthorized areas, can create hazardous situations. Real-time detection can enable timely interventions or safety reminders.
- Misplaced or stolen tools can disrupt workflows and potentially lead to unsafe practices. Immediate detection can aid in tool tracking, management, and accountability.
- By integrating small tool detection with worker tracking, unsafe behaviors or proximity violations involving tools can be identified and addressed promptly.

While modern robots may have GPU capabilities to run computationally intensive models, the use of lightweight models can still offer advantages in terms of power efficiency, reduced hardware requirements, and the potential for deploying multiple models concurrently for various safety monitoring tasks. Furthermore, as construction sites evolve and incorporate more edge devices, sensors, and Internet of Things (IoT) technologies, the demand for efficient and lightweight models will become increasingly important to enable real-time safety monitoring and decision-making at the edge.

As can be observed from the results of the ablation experiment (Table 6), the addition of ASH increases the parameters, while also achieving overall better results. Anchor-based detectors are somewhat more sophisticated since they need to do clustering analysis before the learning phase in order to identify the ideal anchor set. The act of moving detection results between hardware adds extra delay in particular specific edge applications. Conversely, the anchor-free method can increase detection speed and has a simpler decoding logic [43]. As a result, by using the without anchor approach to decrease the parameters and increase speed, the model may be further improved.

With an accuracy of just 87.3%, the robustness of the model (Table 7) shows that the LSTD model's detection ability in the misty scenario is rather weak. The model's detection efficacy may be lowered in the misty conditions, which might result in missed detections. As a result, in order to enhance the model's adaptability to this setting and bolster the LSTD model's resilience, it is feasible to include an even greater number of misty conditions input in the training.

Comparisons with smaller or lightweight state-of-the-art architecture showed that LSTD earned the highest mAP, demonstrating the effectiveness of our improvement efforts based on extracting useful features. Extracting relevant feature information from tiny targets is the main goal of the DFE. The lightweight aspect of the model is maintained while acquiring richer feature representations through the use of the IFF. The model's detection performance is further improved by the use of ASH. Furthermore, it is allowed to slightly raise the parameter of LSTD in order to considerably enhance detection precision, even though the parameter size is not minimum.

4.9. Limitation

This study presents promising results and significant advancements in small tools detection in construction environments, however, it still has some limitations, like any other research project. Nonetheless, our primary goal was to overcome the shortcomings of current methods for detecting small tools in intricate and unstructured construction sites. The design of the proposed method allows us to model the system dynamics smoothly with fewer parameters and computation and capture the complex interactions among features. Firstly, the effectiveness of our proposed LSTD model can be influenced by environmental conditions such as illumination levels and mistiness. While we have demonstrated robustness across various scenarios, including intense illumination, subdued illumination, and misty conditions, further research may be needed to enhance the model's adaptability to extreme environmental conditions. Additionally, although our proposed LSTD model achieves superior performance with relatively fewer parameters compared to some state-of-the-art models, there is ongoing research to optimize model complexity further without compromising detection accuracy. Striking a balance between model complexity and efficiency is crucial for real-world deployment. While our lightweight approach is designed to be suitable for

edge device deployment, we did not explicitly implement or test the model on actual edge devices in this study. Future work will involve evaluating the model's performance and conducting experiments on various edge computing platforms to validate its real-time capabilities and resource efficiency in practical construction site monitoring scenarios. However, the proposed method has specific advantages over those methods, especially in the context of the task we focus on, where we are interested in detecting small tools in intricate and unstructured construction sites, and extracting comprehensive features when applying ASH, DFE, and IFF. Also, it is true that the proposed method falls within the broader domain of small object detection, we want to emphasize that our focus is specifically on detecting small tools in intricate and unstructured construction sites.

One limitation of our study is that, while the dataset was carefully curated to include variations in factors such as background diversity, occlusions, and resolution changes, our experimental evaluation primarily focused on the impact of fog and lighting conditions on the model's performance. Due to the lack of comprehensive metadata in the dataset regarding other factors, we were unable to quantitatively evaluate the robustness of our approach to these additional challenges. While this dataset aimed to capture a diverse range of real-world conditions encountered in construction sites, the explicit evaluation of our model's performance under varying levels of occlusion, background complexity, and resolution changes was not conducted. Future work should incorporate detailed annotations and controlled experiments to assess the model's resilience to these factors, which are known to influence the performance of object detection methods in practical scenarios. Furthermore, collecting and annotating additional data with an emphasis on these specific factors would enable a more comprehensive evaluation of our approach's capabilities and potential limitations in handling the full spectrum of challenges present in construction site environments. However, it's essential to note that the data utilized in our study is a small tools dataset [11], currently the largest dataset available for this specific domain. This dataset comprises real-world image capture in standard settings, providing a diverse range of construction site contexts. We have meticulously processed these images to enable comprehensive testing of our proposed model's detection capabilities for different small tools in intricate and unstructured construction sites. As a limitation, it is crucial to highlight that due to the unique labeling process and the distinctive number of classes in the used datasets compared to other existing datasets, we faced challenges in testing our model on alternative datasets to evaluate the generalization of the proposed method. The lack of a standardized labeling schema and class distribution in other datasets limits the direct applicability of our model beyond the used datasets. Despite these constraints, we have conducted a series of extensive experiments on the dataset [11] to showcase the effectiveness of our proposed in detecting small tools in intricate and unstructured construction sites.

Also, our future research directions include refinement of environmental adaptability, further investigation into techniques to enhance the model's adaptability to extreme environmental conditions, such as misty environments or varying illumination levels. Additionally, tailoring the LSTD model for specific applications within the construction domain, considering factors such as camera placement, scene complexity, and tool diversity, to optimize detection performance is a priority. Furthermore, incorporating real-time feedback mechanisms into the LSTD model to enable continuous learning and adaptation in dynamic construction environments is an important avenue for exploration. By addressing these limitations and pursuing future research directions, we aim to further advance the field of small tools detection in construction environments and contribute to the development of safer construction practices.

5. Conclusions

This paper presents LSTD, a Lightweight Small object detection architecture for difficult and unstructured construction sites. By utilizing on-the-fly data augmentation techniques, the mAP, recall, and precision are increased by 3.1%, 2.7%, and 2.6%, respectively, in comparison to the original dataset. In comparison to the PAN + FPN architecture in the YOLOv5, the DFE + IFF architecture achieves a 73% decrease in parameters and a 27% reduction in computation. The advantages of CBAM integration allow the LSTD to reach the highest precision (85%) and mAP (87.3%). In addition, it increases the LSTD capacity to concentrate on small tool areas. The influence of the DFE + IFF network topology is further demonstrated by the ablation experiment, whereby the CBAM module enhances accuracy and the ASH module effects more on recall. Additionally, the study shows that misty condition in construction sites has a more significant effect on the LSTD's detection ability than illumination. The suggested object identification approach obtains the greatest mAP (87.3%) when compared to other cutting-edge techniques.

To automate the monitoring of safety in construction sites, the proposed LSTD model could be incorporated into robotic systems to monitor onsite small tools for worker safety enhancement and tool tracking and management. Further study will investigate more to make sure it can reach the necessary speed and accuracy when deployed on edge devices. To further enhance the model's capacity to discriminate between small tools, it is also worthwhile to investigate the incorporation of complicated construction site environments into the model input. Also, given the properties of small tools, it is promising to investigate ways to improve BB generation techniques or create loss functions that are more suited for small object recognition in order to improve small tools detection performance.

6. Disclosures

The authors declare no conflict of interest.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2222881. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Census of fatal occupational injuries summary, https://www.bls.gov/news.release/cfoi.nr0.htm, [Accessed 24-01-2024] (2023).

[2] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose, W. An, Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, Automation in Construction 85 (2018) pp. 1–9. doi:10.1016/j.autcon.2017.09.018.

[3] W. Yi, A. P. Chan, Critical review of labor productivity research in construction journals, Journal of Management in Engineering 30 (2) (2014) pp. 214–225. doi:10.1061/(ASCE)ME.1943-5479.0000194.

[4] C. Mao, Q. Shen, W. Pan, K. Ye, Major barriers to off-site construction: The developer's perspective in china, Journal of Management in Engineering 31 (3) (2015) p. 04014043. doi:10.1061/(ASCE)ME.1943-5479.0000246.

[5] U.S. Bureau of Labor Statistics, https://www.bls.gov/, [Accessed 25-01-2024].

[6] J. Hinze, J. N. Devenport, G. Giang, Analysis of construction worker injuries that do not result in lost time, Journal of Construction Engineering and Management 132 (3) (2006) pp. 321–326. doi:10.1061/(ASCE)0733-9364(2006)132:3(321).

[7] M. Bonyani, M. Soleymani, C. Wang, Construction workers' unsafe behavior detection through adaptive spatiotemporal sampling and optimized attention based video monitoring, Automation in Construction 165 (2024) p.105508. doi:10.1016/j.autcon.2024.105508.

[8] E. D. Marks, J. Teizer, Method for testing proximity detection and alert technology for safe construction equipment operation, Construction Management and Economics 31 (6) (2013) pp. 636–646. doi:10.1080/01446193.2013.783705.

[9] Ultralytics, https://github.com/ultralytics/, [Accessed 25-01-2024] (2021).

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems 28, [Accessed 25-01-2024] (2015). URL https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[11] K. Lee, C. Jeon, D. H. Shin, Small tool image database and object detection approach for indoor construction site safety, KSCE Journal of Civil Engineering 27 (3) (2023) pp. 930–939. doi:10.1007/s12205-023-1011-2.

[12] Y. Liu, P. Sun, N. Wergeles, Y. Shang, A survey and performance evaluation of deep learning methods for small object detection, Expert Systems with Applications 172 (2021) p. 114602. doi:10.1016/j.eswa.2021.114602.

[13] K. Tong, Y. Wu, F. Zhou, Recent advances in small object detection based on deep learning: A review, Image and Vision Computing 97 (2020) p. 103910. doi:10.1016/j.imavis.2020.103910.

[14] H. Wang, Y. Song, L. Huo, L. Chen, Q. He, Multiscale object detection based on channel and data enhancement at construction sites, Multimedia Systems 29 (1) (2023) pp. 49–58. doi:10.1007/s00530-022-00983-x.

[15] G. X. Hu, Z. Yang, L. Hu, L. Huang, J. M. Han, Small object detection with multiscale features, International Journal of Digital Multimedia Broadcasting 2018 (2018) p. 4546896. doi:10.1155/2018/4546896.

[16] B. Bosquet, M. Mucientes, V. M. Brea, Stdnet-st: Spatio-temporal convnet for small object detection, Pattern Recognition 116 (2021) p. 107929. doi:10.1016/j.patcog.2021.107929.

[17] C. Eggert, S. Brehm, A. Winschel, D. Zecha, R. Lienhart, A closer look: Small object detection in faster r-cnn, in: Proceedings - IEEE International Conference on Multimedia and Expo, IEEE, 2017, pp. 421–426. doi:10.1109/ICME.2017.8019550.

[18] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, C. Piao, Uav-yolo: Small object detection on unmanned aerial vehicle perspective, Sensors (Switzerland) 20 (8) (2020) p. 2238. doi:10.3390/s20082238.

[19] H. Luo, J. Liu, W. Fang, P. E. D. Love, Q. Yu, Z. Lu, Real-time smart video surveillance to manage safety: A case study of a transport mega-project, Advanced Engineering Informatics 45 (2020) p. 101100. doi:10.1016/j.aei.2020.101100.

[20] J. Ren, Y. Guo, D. Zhang, Q. Liu, Y. Zhang, Distributed and efficient object detection in edge computing: Challenges and solutions, IEEE Network 32 (6) (2018) pp. 137–143. doi:10.1109/MNET.2018.1700415.

[21] B. Ku, K. Kim, J. Jeong, Real-time isr-yolov4 based small object detection for safe shop floor in smart factories, Electronics (Switzerland) 11 (15) (2022) p. 2348. doi:10.3390/electronics11152348.

[22] Z. Z. Wang, K. Xie, X. Y. Zhang, H. Q. Chen, C. Wen, J. B. He, Small-object detection based on yolo and dense block via image super-resolution, IEEE Access 9 (2021) pp. 56416–56429. doi:10.1109/ACCESS.2021.3072211.

[23] C. Sun, Y. Ai, S. Wang, W. Zhang, Mask-guided ssd for small-object detection, Applied Intelligence 51 (6) (2021) pp. 3311–3322. doi:10.1007/s10489-020-01949-0.

[24] J. S. Lim, M. Astrid, H. J. Yoon, S. I. Lee, Small object detection using context and attention, in: 3rd International Conference on Artificial Intelligence in Information and Communication, IEEE, 2021, pp. 181–186. doi:10.1109/ICAIIC51459.2021.9415217.

[25] Z. Yang, Y. Yuan, M. Zhang, X. Zhao, Y. Zhang, B. Tian, Safety distance identification for crane drivers based on mask r-cnn, Sensors (Switzerland) 19 (12) (2019) p. 2789. doi:10.3390/s19122789.

[26] J. Zhao, E. Obonyo, Convolutional long short-term memory model for recognizing construction workers' postures from wearable inertial measurement units, Advanced Engineering Informatics 46 (2020) p. 101177. doi:10.1016/j.aei.2020.101177.

[27] Y. Zhao, Q. Chen, W. Cao, J. Yang, J. Xiong, G. Gui, Deep learning for risk detection and trajectory tracking at construction sites, IEEE Access 7 (2019) pp. 30905–30912. doi:10.1109/ACCESS.2019.2902658.

[28] W. Fang, L. Ding, H. Luo, P. E. D. Love, Falls from heights: A computer vision-based approach for safety harness detection, Automation in Construction 91 (2018) pp. 53–61. doi:10.1016/j.autcon.2018.02.018.

[29] X. Luo, H. Li, D. Cao, F. Dai, J. Seo, S. Lee, Recognizing diverse construction activities in site images via relevance networks of construction-related objects detected by convolutional neural networks, Journal of Computing in Civil Engineering 32 (3) (2018) p. 04018012. doi:10.1061/(ASCE)CP.1943-5487.0000756.

[30] H. Son, H. Choi, H. Seong, C. Kim, Detection of construction workers under varying poses and changing background in image sequences via very deep residual networks, Automation in Construction 99 (2019) pp. 27–38. doi:10.1016/j.autcon.2018.11.033.

[31] F. C. Akyon, S. O. Altinuc, A. Temizel, Slicing aided hyper inference and fine-tuning for small object detection, in: Proceedings - International Conference on Image Processing, IEEE, 2022, pp. 966–970. doi:10.1109/ICIP46576.2022.9897990.

[32] M. C. Keles, B. Salmanoglu, M. S. Guzel, B. Gursoy, G. E. Bostanci, Evaluation of yolo models with sliced inference for small object detection, arXiv (2022). doi:10.48550/arXiv.2203.04799.

[33] X. Wang, Z. Yang, J. Wu, Y. Zhao, Z. Zhou, Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision, in: Proceedings - IEEE Annual Joint Conference: INFOCOM, IEEE Computer and Communications Societies, Vol. 2021-May, IEEE, 2022. doi:10.1109/INFOCOM42981.2021.9488843.

[34] C. Chen, H. Gu, S. Lian, Y. Zhao, B. Xiao, Investigation of edge computing in computer vision-based construction resource detection, Buildings 12 (12) (2022) p. 2167. doi:10.3390/buildings12122167.

[35] Z. Xu, J. Huang, K. Huang, A novel computer vision-based approach for monitoring safety harness use in construction, IET Image Processing 17 (4) (2023) pp. 1071–1085. doi:10.1049/ipr2.12696.

[36] J. Zhang, C. C. Liu, J. J. C. Ying, Deepsafety: a deep neural network-based edge computing framework for detecting unsafe behaviors of construction workers, Journal of Ambient Intelligence and Humanized Computing 14 (12) (2023) pp. 15997–16009. doi:10.1007/s12652-023-04554-4.

[37] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, I. H. Yeh, Cspnet: A new backbone that can enhance learning capability of cnn, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Vol. 2020-June, 2020, pp. 1571–1580. doi:10.1109/CVPRW50498.2020.00203.

[38] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 3–19, [Accessed 25-01-2024].
URL https://link.springer.com/chapter/10.1007/978-3-030-01234-2_1

[39] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7132–7141. doi:10.1109/CVPR.2018.00745.

[40] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021, pp. 13708–13717. doi:10.1109/CVPR46437.2021.01350.

[41] L. Yang, R. Y. Zhang, L. Li, X. Xie, Simam: A simple, parameter-free attention module for convolutional neural networks, in: Proceedings of Machine Learning Research, Vol. 139, PMLR, 2021, pp. 11863–11874, [Accessed 24-01-2024].
URL https://proceedings.mlr.press/v139/yang21o

[42] G. Song, Y. Liu, X. Wang, Revisiting the sibling head in object detector, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 11560–11569. doi:10.1109/CVPR42600.2020.01158.

[43] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv (2021). doi:10.48550/arXiv.2107.08430.

[44] M. Shi, D. Zheng, T. Wu, W. Zhang, R. Fu, K. Huang, Small object detection algorithm incorporating swin transformer for tea buds, Plos One 19 (3) (2024) p. e0299902. doi:10.1371/journal.pone.0299902.

[45] C. Chen, H. Ding, M. Duan, Discretization and decoupled knowledge distillation for arbitrary oriented object detection, Digital Signal Processing 150 (2024) p.104512. doi:10.1016/j.dsp.2024.104512.

[46] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv (2014). doi:10.48550/arXiv.1412.6980.

[47] C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, arXiv (2022) pp. 7464–7475[Accessed 24-01-2024].
URL https://openaccess.thecvf.com/content/CVPR2023/html/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.html

[48] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Yolov6: A single-stage object detection framework for industrial applications, arXiv (2022). doi:10.48550/arXiv.2209.02976.

[49] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: More features from cheap operations, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 1577–1586. doi:10.1109/CVPR42600.2020.00165.

[50] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv (2018). doi:10.48550/arXiv.1804.02767.

[51] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, Q. Le, H. Adam, Searching for mobilenetv3, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-October, 2019, pp. 1314–1324. doi:10.1109/ICCV.2019.00140.

[52] H. Yu, L. Zhang, Lcnet: a light-weight network for object counting, in: International Conference on Neural Information Processing, Springer, 2020, pp. 411–422. doi:10.1007/978-3-030-63830-6_35.