Neurocomputing 525 (2023) 29–41

Contents lists available at ScienceDirect

Neurocomputing

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

DKTNet: Dual-Key Transformer Network for small object detection

Shoukun Xu [a], Jianan Gu [a], Yining Hua [b,*], Yi Liu [a,*]

[a] Changzhou University, Changzhou Jiangsu 213164, China
[b] University of Aberdeen, UK

ARTICLE INFO

Article history:
Received 18 August 2022
Revised 17 November 2022
Accepted 8 January 2023
Available online 11 January 2023

Keywords:
Small object detection
Transformer
Dual-key

ABSTRACT

Summary: Object detection is a fundamental computer vision task that plays a crucial role in a wide range of real-world applications. However, it is still a challenging task to detect the small size objects in the complex scene, due to the low resolution and noisy representation appearance caused by occlusion, distant depth view, *etc.* To tackle this issue, a novel transformer architecture, Dual-Key Transformer Network (DKTNet), is proposed in this paper. To improve the feature attention ability, the coherence of linear layer outputs Q and V are enhanced by a dual-K integrated from K_1 and K_2, which are computed along Q and V, respectively. Instead of spatial-wise attention, channel-wise self-attention mechanism is adopted to promote the important feature channels and suppress the confusing ones. Moreover, 2D and 1D convolution computations for Q, K and V are proposed. Compared with the fully-connected computation in conventional transformer architectures, the 2D convolution can better capture local details and global contextual information, and the 1D convolution can reduce network complexity significantly. Experimental evaluation is conducted on both general and small object detection datasets. The superiority of the aforementioned features in our proposed approach is demonstrated with the comparison against the state-of-the-art approaches.

1. Introduction

Nowadays, object detection has attracted great attention due to its broad range of real-world applications, such as autonomous driving [1–3], intelligent monitoring [4–7], virtual reality [8,9], augmented reality [10–12], *etc.* With the rapid development of deep learning techniques, two categories of object detection approaches have been proposed in the literature, which are 1) two-stage approaches based on region proposal network, such as Faster R-CNN [13], and Cascade R-CNN [14], and 2) one-stage approaches, such as SSD [15] and YOLO [16].

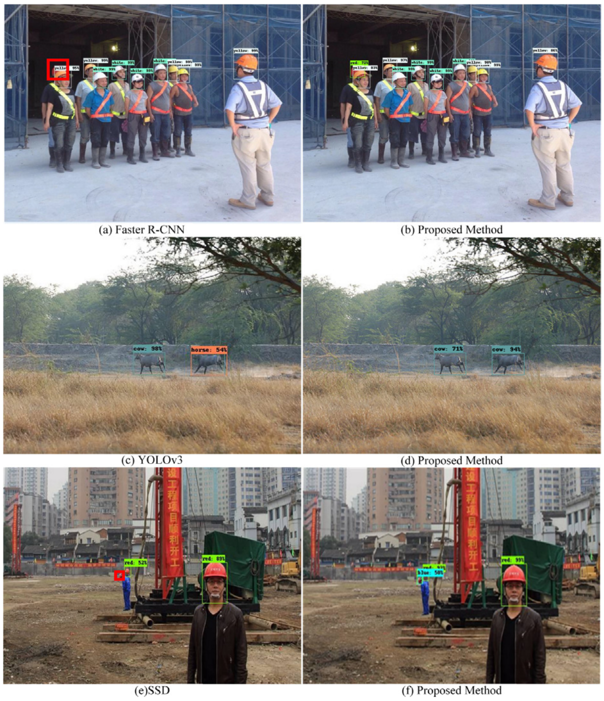
However, in the object detection field, small-size object detection in complex scenes is still a challenging task, which can be used for pedestrian and traffic sign detection in autonomous driving [17], helmet detection in industrial scenes [18,19], *etc.* Although the aforementioned deep object detection approaches can achieve remarkable performance in large object scenarios with high resolution and clear appearance, when they are applied to small object scenarios, the discrimination is depressed by the low resolutions, noisy representations, occlusions, and similar surrounding background, which will eventually cause misdetection or misrecognition. For example, as shown in Fig. 1(a), the occluded small-scale object marked by the red box is missed detected by the previous object detector, because its appearance and structure are destroyed by the occlusion. Besides, as shown in Fig. 1(c), when the object has a similar appearance to its surroundings, it will be wrongly recognized by the previous object detector. Moreover, as shown in Fig. 1(e), the small object with a distant depth view is hard to be detected by the previous object detector because of its low discrimination.

To address the above problems, in this paper, we introduce the transformer architecture for small object detection. Instead of directly incorporating the transformer architecture into the task of small object detection, we propose a Dual-Key Transformer Network (DKTNet) for small object detection. Specifically, to enhance the correlation between Q (Query) and V (Value), we develop a dual-key strategy to compute two keys, including one key along with the Q stream and one key along with the V stream. The use of different stream feature fusion allows the learning of different dimensions of feature information and the adequate extraction of target features. Different from the Swin transformer[1] that computes one key to connect Q and V, we compute two keys along with

* Corresponding authors.
E-mail addresses: yining.hua@abdn.ac.uk (Y. Hua), liuyi0089@gmail.com (Y. Liu).

[1] Specifically, in light of the superior performance of Swin transformer [20], we choose it as our baseline.

https://doi.org/10.1016/j.neucom.2023.01.055
0925-2312/© 2023 Elsevier B.V. All rights reserved.



(a) Faster R-CNN

(b) Proposed Method

(c) YOLOv3

(d) Proposed Method

(e)SSD

(f) Proposed Method

Fig. 1. Problems for small object detection. Left column: the problems of the previous object detectors, *e.g.*, occlusion in (a), similar background in (c), and distant depth view in (e); Right column: Results of the proposed method. It can be seen that the problems of the previous methods can be well addressed by our method.

the Q stream and the V stream, respectively, which are further integrated via addition to enhance the coherence between Q and V. Besides, instead of computing spatial attention adopted in most of transformer architectures, we compute the channel attention in our proposed dual-key transformer, which helps to promote those important channels of feature maps while suppressing whole confusing ones. In addition, we maintain the original size and replace the traditional fully-connection computation with the 2D convolution computation for Q, K, and V, which helps to preserve the local context. Moreover, inspired by the 1D convolution computation for tokens in natural language processing [21], we utilize 1D convolution to further replace 2D convolution for computing Q, K, and V to learn the relation within the adjacent range of the single dimension feature map, which reduces network parameters and computation. As shown in Fig. 1, our method can well detect the multi-scale objects and small-scale objects in various complex scenes, compared to the other methods.

In summary, the original contributions of the paper are as follows:

- A novel transformer architecture, Dual-Key Transformer Network (DKTNet), is proposed for the task of small object detection in complex backgrounds. The integration of K_1 and K_2 constructs the dependence between Q and V, which improves learning and characterisation of key features of the target.
- To avoid the local context loss caused by fully-connected layers, we develop a 2D convolution to compute Q, K^{dual} and V in MDKTA. In contrast to the fully-connected computation, the 2D convolution can preserve the local context for the transformer.
- A 1D convolution is developed to further enhance computational efficiency. By reducing the number of unnecessary reshaping operations, the 1D convolution can avoid the large-scale tensors of input features. As shown in the ablation study, with this significant reduction in computation complexity, the detection performance can keep competitive without sacrifice.
- Extensive qualitative and quantitative experiments proves the proposed dual-keys transformer achieves the state-of-the-art performance of small object detection in GDUT, SHW1, SHW2 and also achieves the state-of-the-art performance of general object detection in Pascal VOC 2007 datasets.

The rest of this paper is structured as follows. Section 2 successively summarises the related work from general object detection, small object detection and transformer in computer vision aspects. The proposed DKTNet method is presented in Section 3, where the details of the proposed dual-key mechanism, 2D and 1D convolution computations, and adopted loss function are illustrated. The experimental evaluation of the proposed method is conducted with the comparison against conventional works, the performance analysis is provided in Section 4, including an ablation study to identify the contributions of each component in our proposed method. Finally, the paper concludes in Section 5 with a discussion of future work.

2. Related work

2.1. General object detection

Object Detection is one of the core research tasks in computer vision, and its main task is to find the region of interest or target in an image. Due to the different shapes, colors and poses of targets, the noise of imaging devices, the limitations of shooting angles and ranges, and the complex background interference, generic target detection has been a challenging research task [22]. In recent years, general object detection milestone framework can be divided into two mainstream approaches. One is a two-stage object detection algorithm, which obtains the region suggestion frame by convolutional neural network. Faster R-CNN [13] predicted object scores and bounds with pooled features of proposed regions. R-FCN [23] introduced position-sensitive score maps to share the per-ROI feature computation to alleviate the contradiction between translation invariance between object classification and translation variability between object detection. Sparse R-CNN [24] learned a fixed set of sparse candidates for region proposal. Another is one-stage object detection algorithm, which obtains the object classification and bounding box location directly on feature maps. YOLOv5 [25] was a high-performance, general-purpose target detection method that could perform both target localization and target classification tasks in one time, with significant improvements in detection accuracy and speed compared to previous versions. SSD [15] improved the one-stage detection with various scales of multilayer features.

2.2. Small object detection

Small object detection has been a challenging computer vision task, and existing detection algorithms have achieved good results for large-scale targets, but the detection of small objects is still less satisfactory. More and more methods have been proposed to improve the accuracy of small object detection [26–28]. Based on the Faster R-CNN, the researchers have made a number of improvements. Cascade R-CNN [29], which was served as one of the best single model detector, designed different regression normalization factors to adjust the aptitude of regression term in different stages. Libra R-CNN [30] proposed to promote the regression gradients from the accurate samples. The feature pyramid networks (FPN) [31] was proposed to solve the multi-scale problem in object detection. The image pyramid based approach used images at different scales to detect objects at different scales, such as small scale images to detect large scale objects and large scale images to detect small scale objects. SNIP [32] normalized the gradients from different object scales during training, such that the whole detector was scale-specific. TridentNet [33] constructed a parallel multi-branch architecture in which each branch shares the same transformation parameters but with different receptive fields. DSSD [34] upsampled the low-resolution features of SSD by the transposed convolution in the decoder to increase the internal space. [35] employed an automated approach to find the best data augmentation strategy for object detection. [36] proposed top-down modulations (TDM) as a way to incorporate fine details into the detection framework.

2.3. Transformer in computer vision

The successful application of Transformer [38] in natural language processing had inspired its application for the computer vision field. Vision Transformer (ViT) [39] was the first Transformer-based method for image classification, which outperformed CNNs. ViT [39] sliced the image into 16*16 patches and treated each patch as an output whole, and then used the Transformer architecture instead of the traditional CNN for global self-attention. Later, Detection Transformer (DETR) [40] innovatively treated target detection as an ensemble prediction problem, which was solved by using encoders and target queries to encode features and anchor frames, and then used decoders and Feed Forward Networks (FFNs) to obtain prediction frames and target classes. Although the Transformer architecture could avoid the problems of CNNs with limited receptive fields and maladaptive to the input content, its computational complexity grew quadratically with the spatial resolution. One of the latest approaches was Swin-Transformer (Shifted Window Transformer) [20], which introduced a moving-window-based multi-headed attention mechanism to balance performance and efficiency. But its contextual aggregation within local images against the main motivation of using self-attentiveness.

To alleviate the inadequate image feature extraction and low capability of small object feature extraction in conventional small object detection approaches, 1) We propose a dual-key mechanism for the transformer architecture by a dual-stream strategy, which enhances the coherence between Q and V to improve feature extraction; 2) We compute the channel attention instead of the spatial attention to promote those important features while suppressing those unimportant ones.

3. Methodology

The framework of the proposed DKTNet for small object detection is shown in Fig. 2. Faster R-CNN [13] is chosen as the baseline. Specifically, the image is input into the backbone network, *i.e.*, ResNet50 [41] and Feature Pyramid Network (FPN) [31], to extract multi-scale target features, which are subsequently fed into the dual-key Transformer for feature discrimination enhancement. Subsequently, Region Proposal Network (RPN) is utilized to calculate the foreground Region of Interest (RoI) and the first adjustment of the target bounding box. Finally, the features obtained from RPN are input into the RoI Align to achieve the RoI features, which are mapped into feature vectors using the full connectivity. The softmax [42] is used for classification and the target bounding box regression model is used to refine the target position. In the following, we will detail the core components of the proposed network, including dual-key transformer and computation for Q, K, and V.

3.1. Dual-key transformer

The procedure of the conventional Transformer based object detection is the Multi-head Self-Attention (MSA), which is shown in Fig. 3(a). The 2D image features $\mathbf{Y} \in \mathbb{R}^{C\times H\times W}$ is flattened and transposed t to $\mathbf{Y}' \in \mathbb{R}^{HW\times C}$, which is used to compute the $\mathbf{Q},\mathbf{K}$ and $\mathbf{V}$ as

$$\mathbf{Q} = f_{W_Q}(\mathbf{Y}'), \mathbf{K} = f_{W_K}(\mathbf{Y}'), \mathbf{V} = f_{W_V}(\mathbf{Y}'), \tag{1}$$

where $f_{W_Q}(\cdot), f_{W_K}(\cdot), f_{W_V}(\cdot)$ are three different linear transformations.

Different from the traditional transformer-based object detection method, as shown in Fig. 3(b), the proposed dual-key transformer based object detection attention, *i.e.*, Multi-head Dual-Key Transposed Attention (MDKTA), is shown in Fig. 3(b), which consists of three steps: two keys calculation, two keys integration, and attention computation.

Step 1: Two keys calculation.

Given the transposed features $\mathbf{Y}' \in \mathbb{R}^{HW\times C}$, two fully-connected layers are employed to learn two streams of knowledge, *i.e.*, $\mathbf{Q},\mathbf{K}_1$ and $\mathbf{K}_2,\mathbf{V}$. The procedure can be written by

$$(\mathbf{Q},\mathbf{K}_1) = f_{chunk}(f_{W_F^1}(\mathbf{Y}')), \tag{2}$$

$$(\mathbf{K}_2,\mathbf{V}) = f_{chunk}(f_{W_F^2}(\mathbf{Y}')), \tag{3}$$

where the $f_{chunk}(\cdot)$ operation is used to compute the values of $\mathbf{Q},\mathbf{K}_1,\mathbf{K}_2$, and $\mathbf{V}$. It can be implemented by the fully-connected linear computation. $f_{W_F^{(\cdot)}}(\cdot)$ is the fully-connected linear operation.

Step 2: Two keys integration.

$\mathbf{K}_1$ and $\mathbf{K}_2$ are computed along the stream of $\mathbf{Q}$ and $\mathbf{V}$, respectively. Therefore, $\mathbf{K}_1$ and $\mathbf{K}_2$ contains the dependence with $\mathbf{Q}$ and $\mathbf{V}$, respectively. Therefore, the integration of $\mathbf{K}_1$ and $\mathbf{K}_2$ will construct the dependence between $\mathbf{Q}$ and $\mathbf{V}$. To this end, we integrate $\mathbf{K}_1$ and $\mathbf{K}_2$ to achieve the final key values, *i.e.*,

$$\mathbf{K}^{dual} = \mathbf{K}_1 + \mathbf{K}_2. \tag{4}$$

With respect the vectors of key values, the addition operation can complement $\mathbf{K}_1$ and $\mathbf{K}_2$, and further enhance the coherence between $\mathbf{Q}$ and $\mathbf{V}$ to improve learning and characterisation of key features of the target, which will be verified in the section of Experiment, *i.e.*, Section 4.

Step 3: Attention computation.

In hidden layers, there are feature maps with multiple channels. Different channels of feature maps in essence contribute differently to the semantic understanding. Therefore, we apply the values of $\mathbf{Q},\mathbf{K}^{dual}$, and $\mathbf{V}$ to compute the channel-wise attention to enhance the important channels while suppressing those confusing channels. Specifically, considering the advantage of the multi-head operation, we divide the number of feature channels into h heads, each with $c = C/h$ channels, and learn separate attention maps in parallel. To this end, we reshape $\mathbf{Q}$ and $\mathbf{K}$ projections such that their dot-product interaction generates a channel-wise attention map with the shape of $\mathbb{R}^{c\times c}$, *i.e.*,

$$f_{Attention}(\mathbf{Q},\mathbf{K}^{dual},\mathbf{V}) = \mathbf{V}Softmax(\frac{\mathbf{K}^{dual}\mathbf{Q}}{\varepsilon}), \tag{5}$$

where $\mathbf{Q} \in R^{HW\times C}, \mathbf{K}^{dual} \in \mathbb{R}^{C\times HW}$, and $\mathbf{V} \in \mathbb{R}^{HW\times C}$ matrices are obtained after reshaping tensors from the original size $\mathbf{Y}' \in \mathbb{R}^{HW\times C}$. Here, ε is a learnable scaling parameter to control the magnitude of the dot product of $\mathbf{K}^{dual}$ and $\mathbf{Q}$ before applying the Softmax function.

Based on the above procedure, the MDKTA process can be written as:

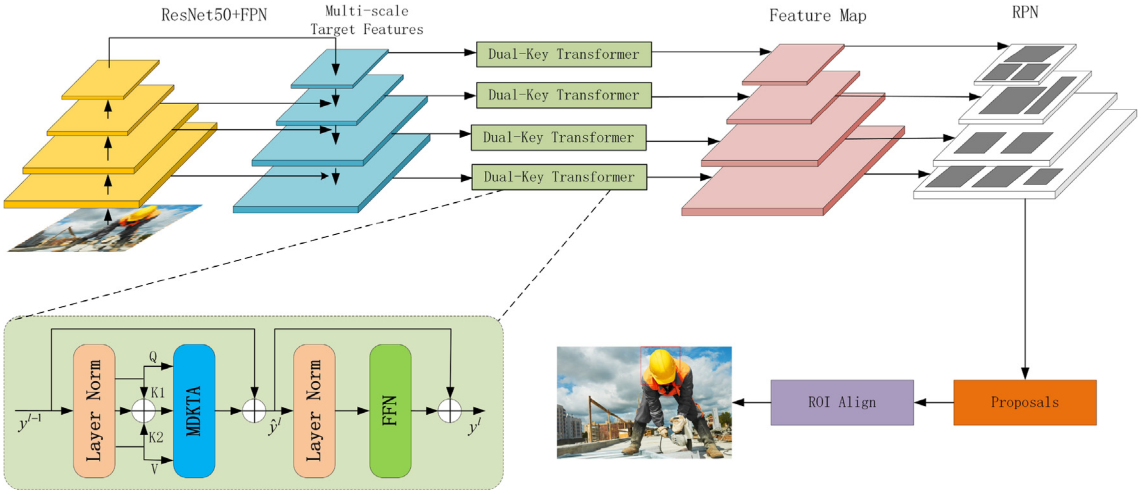
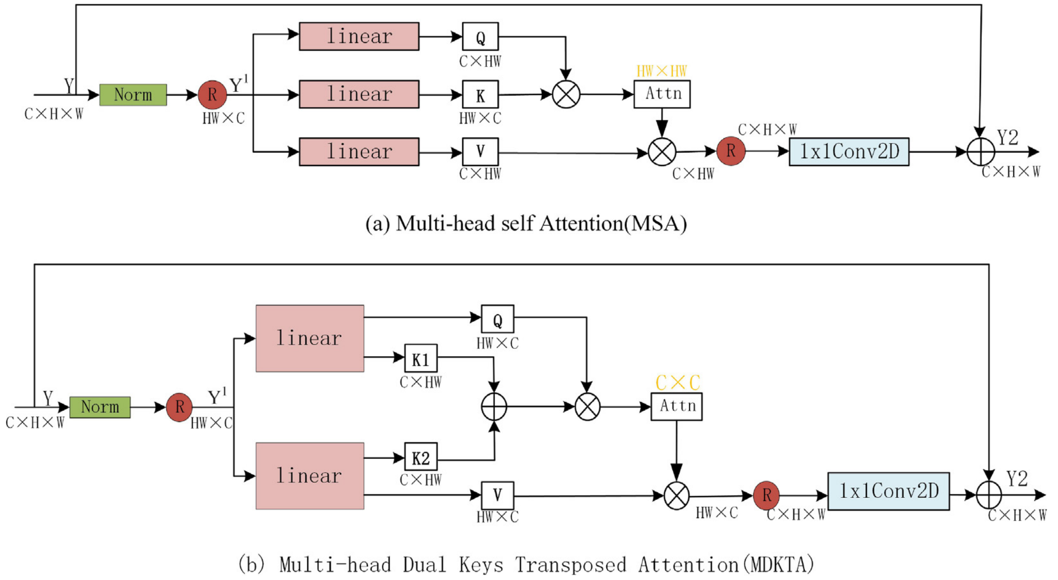


Fig. 2. Framework of the proposed DKTNet for small object detection. "ResNet50 + FPN" are chosen as the backbone network to extract features of the input image, which are fed into the proposed dual-key transformer to enhance the feature discrimination. On top of that, RPN and RoIAlign [37] are utilized to detect the objects.



(a) Multi-head self Attention(MSA)

(b) Multi-head Dual Keys Transposed Attention(MDKTA)

Fig. 3. Difference between the conventional MSA and the proposed MDKTA. (a) denotes the conventional MSA, which uses three fully-connected layers to compute the Q, K and V. (b) denotes the method MDKTA in this paper, which uses two fully-connected layers to compute the Q, K^{dual} and V. The usage of different stream feature fusion enhances the coherence between Q stream and V stream which improves feature extraction to obtain more discriminative object features.

$$\hat{\mathbf{Y}} = f_{W_p}(f_{Attention}(\mathbf{Q}, \mathbf{K}^{dual}, \mathbf{V})) + \mathbf{Y}, \tag{6}$$

where $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ are the output and input feature maps, $f_{W_p}(\cdot)$ is the 1x1 2D convolution operation.

As pointed out by conventional works, the Feed-Forward Network (FFN) in the standard Transformer presents limited capability to leverage local context. However, neighboring pixels are crucial references for object detection. To overcome this issue, as shown in Fig. 4, we use 1×1 two dimension convolution to capture local information, which is followed by the GELU as the activation function to implement the Feed-Forward Network (FFN), *i.e.*,

$$f_{FFN}(f_{LN}(\hat{\mathbf{Y}})) = f_{W_p^0}(\theta(f_{W_p^1}(f_{LN}(\hat{\mathbf{Y}}))) \odot f_{W_p^2}(f_{LN}(\hat{\mathbf{Y}}))) + \hat{\mathbf{Y}}, \tag{7}$$

where θrepresents the activation function of GELU, $\odot$denotes element-wise multiplication. $f_{LN}(\cdot)$ is the layer normalization operation [43]. $f_{W_p^{(\cdot)}}(\cdot)$ is the 1x1 two dimension convolution operation.

3.2. Convolutional computation for Q, K^{dual} and V

In this section, the details of 2D and 1D convolutions based on MDKTA are provided, successively.

2D convolution based MDKTA. The computation of Q, K, and V in the Transformer attention mechanism follows the processing method in the natural language task, in which the usage of fully-connected layers may cause some local context lost. To this end, we apply the convolutional computation to substitute for the fully-connected computation with the aim of preserving local context. Specifically, as shown in Fig. 5(a), the tensor of 2D image features is first normalized. The 2D convolution with the kernel of 1x1 is used to compute the $\mathbf{Q}, \mathbf{K}_1, \mathbf{K}_2$, and $\mathbf{V}$, *i.e.*,

$$(\mathbf{Q}, \mathbf{K}_1) = f_{chunk}(f_{W^1_{2D}}(\mathbf{Y}')), \tag{8}$$

$$(\mathbf{K}_2, \mathbf{V}) = f_{chunk}(f_{W^2_{2D}}(\mathbf{Y}')). \tag{9}$$

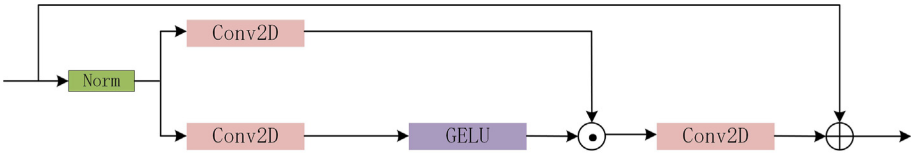
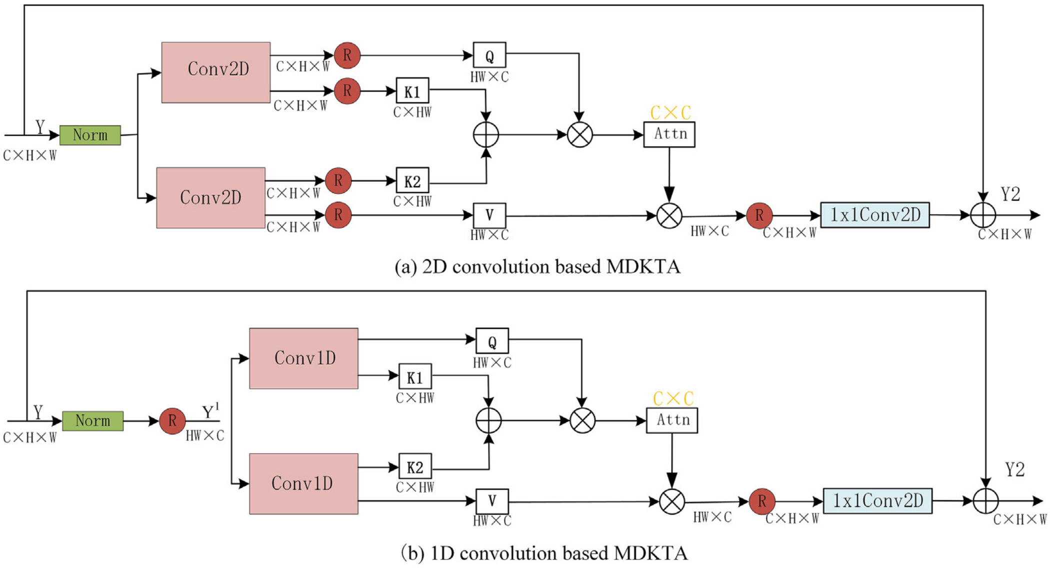


Fig. 4. Structure of Feed-forward Network (FFN). We use 2D convolution to capture the local information of the image, which solves the limitation of conventional transformer for local content acquisition.



(a) 2D convolution based MDKTA

(b) 1D convolution based MDKTA

Fig. 5. Two convolutional computations of the Q, K_1, K_2 and V. (a) shows the MDKTA, which uses 2D convolution to capture local information. (b) shows the MDKTA, which uses 1D convolution to compute the knowledge of Q, K_1, K_2 and V. The convolutional computation can capture the local context in feature layers, which is more effective than the fully connected computation. The 1D convolutional computation is further adopted to reduce the computational complexity while maintaining the performance.

On top of that, the $\mathbf{Q}, \mathbf{K}^{dual}$, and $\mathbf{V}$ can be obtained by Eq. (4). where $f_{W_{2D}^{(\cdot)}}(\cdot)$ is the 1x1 two-dimensional convolution,$\mathbf{Y}'$ is the tensor obtained after Layer Normalization [43].

1D convolution based MDKTA. It is well noticed that the above 2D convolution based computation still encounters large computing complexity is caused by the large-scale tensors of input features. To further enhance the computational efficiency, similar to the tokens in natural language processing, we apply the 1D convolution to compute the knowledge of $\mathbf{Q}, \mathbf{K}_1, \mathbf{K}_2$, and $\mathbf{V}$. As shown in Fig. 5(b), the 1D convolution is applied to compute the values of $\mathbf{Q}, \mathbf{K}_1, \mathbf{K}_2$, and $\mathbf{V}$, *i.e.*,

$$(\mathbf{Q}, \mathbf{K}_1) = f_{chunk}(f_{W_{1D}^1}(\mathbf{Y}')), \tag{10}$$

$$(\mathbf{K}_2, \mathbf{V}) = f_{chunk}(f_{W_{1D}^2}(\mathbf{Y}')), \tag{11}$$

where $f_{W_{1D}^{(\cdot)}}(\cdot)$ is the 1x1 single dimension convolution operation, and $\mathbf{Y}'$ is the tensor $\mathbf{Y}' \in \mathbb{R}^{HW \times C}$ that becomes to the tensor after the layer normalization is performed by the reshaping substitution.

It is proved that 1D convolution achieves the competitive performance while saving the unnecessary reshaping operations, which can be verified in the ablation study.

Different to the conventional transformer attention. The difference between our proposed transformer attention and the conventional transformer attention lies in three folds. First, instead of the one key computation adopted by the conventional transformer, we compute two keys to enhance the correlation between Q and V, which further improves the attention performance. Secondly, different from the spatial attention map adopted by the conventional transformer, we compute the channel-wise global self-attention map rather than traditional spatial-wise attention map. In Faster R-CNN architecture, the FPN already efficiently captures the multi-scale feature dependency at the 2D spatial level, but it is difficult to capture feature dependency at the channel level. To this end, our proposed channel-wise transformer attention is able to handle this problem. Thirdly, instead of adopting the heavy fully-connected layers for computation, we apply convolutional computation to preserve the local context. Notably, the 1D convolutional computation is further adopted to reduce the computational complexity while maintaining the performance.

3.3. Loss function

Similar to Faster R-CNN [13], we apply two loss functions to jointly train out network, *i.e.*, Eq. (12), including L_{RPN} for the RPN output classification results and the initial adjustment of the target location, and L_{RoI} for the refinement of the target location using Softmax classification and the target frame regression model. The expression of L_{RPN} can be written as follows.

$$L_{total} = L_{RPN} + L_{RoI}. \tag{12}$$

$$L_{RPN}(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i') + \lambda \frac{1}{N_{reg}} p_i' L_{reg}(t_i, t_i'). \tag{13}$$

Specifically, the binary cross-entropy loss function can be written as follows,

$$L_{cls}(p_i, p_i') = -\log[p_i p_i' + (1 - p_i)(1 - p_i')], \tag{14}$$

where i denotes the index of the anchor frame. p_i denotes the probability of the $i-th$ anchor frame being the foreground target. p_i' denotes the label of the $i-th$ anchor frame with only 0 (negative label) and 1 (positive label).

The bounding box regression loss function can be written as follows.

$$L_{reg} = \sum_{i}^{N} \begin{cases} 0.5(t_i - t_i')^2, & |t_i - t_i'| < 1 \\ |t_i - t_i'| - 0.5, & |t_i - t_i'| \geqslant 1, \end{cases} \tag{15}$$

where t_i represents the predicted target bounding box parameter vector, and t_i' represents the calibrated target bounding box parameter vector of $p_i' = 1$.

After obtaining the RoI feature vector, a Softmax function is used for multi-objective classification and the target box position is refined using a bounding box regression algorithm with the following loss function.

$$L_{RoI}(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geqslant 1]L_{reg}(t^u, v), \tag{16}$$

where $L_{cls}(p, u) = -\log p_u$ is the classification loss function. u is the true class label corresponding to the current candidate frame, and p_u is the probability vector p of the output of the Softmax classifier belonging to class u. $L_{reg}(t^u, v)$ denotes the target frame regression loss using the *SmoothL*1 [44] function, in which t^u denotes the parameter vector of the translation scaling transformation corresponding to this candidate frame to the true frame. v denotes the parameter vector obtained from the network calculation. The value of $\lambda$is 1 when the category label $[u \geqslant 1]$ is $[u \geqslant 1]$, otherwise it is 0.

4. Experimental evaluation and analysis

In this section, we conduct extensive experiments to verify the effectiveness and superiority of the proposed method. We first evaluate our method on the challenging general object detection benchmark, *i.e.*, PASCAL VOC 2007 [45]. Secondly, to evaluate our method for the task of small object detection, we choose three challenging safety helmet wearing datasets, including GDUT [46], SHW1 [47], and SHW2 [48], selected from real construction scenes, in which most of the target safety helmets occupy small areas in the images.

4.1. Datasets and evaluation metrics

4.1.1. Datasets

PASCAL VOC2007 [45]: contains 9973 images including 20 categories, in which the challenging classes (*e.g.*, bottle, chair, and boat) are small instances that are common in the real life.

GDUT [46]: 3,174 images in GDUT cover various view ranges, diverse operation scenarios, complex postures of workers and helmet coverings, *etc.* 1,587 images are split for testing, and another 1,587 images are used for training. These images are divided into five categories (red, yellow, white, blue and none) of 18,893 instances, and consist of three-scale objects, including small target ($area \leqslant 32 \times 32 pixels$), medium target ($32 \times 32 pixels \leqslant area \leqslant 96 \times 96 pixels$), and large target ($area > 96 \times 96 pixels$).

SHW1 [47]: The original SHW1 dataset contains six categories including (man, head, face, person_with_helmet, person_no_helmet, head_with_helmet), with a total of 75,578 tags. In this paper, this dataset is divided into a training set, a test set and a validation set according to 8:1:1.

SHW2 [48]: The SHW2 dataset contains 7,581 images including two categories (hat and person), and 9,044 markers for helmet wearers and 111,514 head markers for non-helmet wearers. SHWS can be directly loaded into normal PASCAL VOC format for training and testing.

Table 1
Performance comparison on PASCAL VOC2007. This table shows the comparative result of our proposed method against the conventional ones. Our proposed method can achieve the best performance in 12 categories out of 20. The best performance is marked by bold.

Metrics	YOLOv3 [16]	SSD [15]	Faster R-CNN [13]	OHEM [49]	R-DAD [50]	MLKP [51]	ESSD [52]	Feature-fused SSD [53]	STDN300 [54]	STDN513 [54]	MDSSD300 [55]	MDSSD512 [55]	ACFIM300 [56]	ACFIM500 [56]	DETReg [57]	Ours (DKT_2D)	Ours (DKT_1D)
mAP	73.4	76.8	73.2	78.9	80.2	80.6	79.4	78.9	78.1	80.9	78.6	80.3	79.7	82.1	84.16	84.7	**85.1**
areo	86.3	82.4	76.5	80.6	90	82.2	82.6	82	81.1	86.1	86.5	88.8	86.3	87.8	-	89.6	**90.5**
bike	82	84.7	79	85.7	86.6	83.2	86.1	86.5	86.9	89.3	87.6	88.7	86.9	88.7	-	89.5	**89.9**
bird	74.8	78.4	77.6	79.8	81.3	79.5	79.8	78	76.4	79.5	78.9	83.2	77.8	82.5	-	**84.2**	84.1
boat	59.2	73.8	65.6	69.9	71.2	72.9	72.2	71.7	69.2	74.3	70.6	73.7	74	75.2	-	**77.9**	73.6
bottle	51.8	53.2	54.9	60.8	66	70.5	54.7	52.9	52.4	61.9	55	58.3	56.9	64.4	-	**79.5**	79
bus	79.8	86.2	83.1	88.3	83.4	87.1	86.8	86.6	87.7	88.5	86.9	88.2	87.2	88.8	-	91.3	**92.1**
car	76.5	87.5	84.7	87.9	83.7	88.2	86.9	86.9	84.2	88.3	87	89.3	86.5	88.6	-	94.1	**94.5**
cat	90.6	86	**96.4**	89.6	94.5	88.8	88.2	88.3	88.3	89.4	88.1	87.4	88.1	88.7	-	87.3	88.8
chair	52.1	57.8	52	59.7	63.2	68.3	62.8	63.2	60.2	67.4	58.5	62.4	65	**67.5**	-	66.3	65.9
cow	78.2	83.1	81.9	85.1	84	86.3	85.2	83	81.3	86.5	84.8	85.1	84.8	88.1	-	91.4	**92.5**
table	58.5	70.2	65.7	76.5	64.2	74.5	78.2	76.8	77.6	79.5	73.4	75.1	79.8	**80.2**	-	79.2	80.1
dog	79.3	84.9	84.8	87.1	**92.8**	88.8	87.5	86.1	86.6	86.4	84.8	84.7	86.2	87.5	-	87.4	88.7
horse	82.5	85.2	84.6	87.3	90.1	88.7	88	88.5	88.9	89.2	89.2	89.7	88.1	88.2	-	90.6	**91.6**
mbike	83.4	83.9	77.5	82.4	88.6	82	87	87.5	87.8	88.5	88.1	88.3	86	88.9	-	90.9	**92.2**
person	81.3	79.7	76.7	78.8	87.3	82.2	80	80.4	76.8	79.3	78	83.2	79.9	84	-	90.1	**90.5**
plant	49.1	50.3	38.8	53.7	62.2	**83.2**	56.1	53.9	51.8	53	52.3	56.7	55.3	56.9	-	58.5	60.1
sheep	77.2	77.9	73.6	80.5	82.8	79.5	80.2	80.6	78.4	77.9	78.6	84	79.8	83.4	-	86.5	**87.8**
sofa	62.4	73.9	73.9	78.7	70.9	72.9	80.4	79.5	81.3	81.4	74.5	77.4	81.2	82.2	-	82.1	**82.6**
train	83.8	82.5	82.5	84.5	**88.8**	70.5	88.7	88.2	87.5	86.6	86.8	83.9	86.5	88.4	-	87.4	87.6
tv	68.7	75.3	75.3	80.7	77.6	87.1	78.1	77.9	77.8	85.5	80.7	77.6	77.8	82	-	90.2	**91.22**

4.1.2. Evaluation metrics

In the experiment, Precision(P), Recall(R) and Average Precision (AP) and Mean Average Accuracy (mAP) are used as performance evaluation metrics. Precision is defined as the ratio of the number of actual positive samples to the number of all positive samples in the prediction sample, and is used to evaluate the accuracy of the model. P can be computed by

$$P = \frac{TP}{TP + FP}, \tag{17}$$

where TP indicates that the true case is a positive case and the predicted case is also a positive case, and FP indicates that the true case is a negative case and the predicted case is a positive case.

The recall rate, *i.e.*, the check-all rate, is defined as the ratio of the actual number of positive samples in the predicted samples to the number of predicted samples, and is used to assess the comprehensiveness of the model detection. R can computed by

$$R = \frac{TP}{TP + FN}, \tag{18}$$

where FN indicates that the true case is a positive case and the predicted case is a negative case.

The average precision (AP) is defined as the area enclosed by the precision, recall and axes. AP can be computed by

$$AP = \int_0^1 P(x)dx, \tag{19}$$

where P(x) in Eq. (19) indicates the smoothed precision and recall curves.

The AP for each category is summed and divided by the number of categories to calculate the mAP, *i.e.*,

$$mAP = \frac{\sum_{i=1}^{c} AP_i}{c}, \tag{20}$$

where c denotes the number of categories, and $\mathrm{AP_i}$ denotes the average precision rate of the *ith* category.

In the experiment, mAP50 and mAP are used as evaluation metrics to evaluate the effectiveness of the model. mAP50 takes IoU of 0.5 and calculates the mean AP under IoU = 0.5. mAP takes IoU between 0.5 and 0.95 with a step size of 5% and calculates mAP under these IoUs. mAP_L, mAP_M, mAP_S represent mAP (IoU = 0.5) of different-size targets.

4.2. Implementation details

In this paper, we use the Pytorch platform [59] and one GPU for accelerated computing on a computer with an Intel(R) Core(TM) i7-6800 K CPU @ 3.40GHzTITAN-XP processor and 12G of RAM.

For fair comparisons, all of the datasets use the same training settings for all the methods. The network is trained with the batch size of 2 for 20 epochs. We apply normalization and randomly crop for data augmentation. SGD [60] optimizer is used with weight decay of 5×10^{-4}. The learning rate is initially set to 5×10^{-3} and decreases by the factor of 0.33 every 3 epochs. The momentum is set to 0.9.

To illustrate the performance superiority of our proposed methods (DKT_Conv1D and DKT_Conv2D), extensive experiments are conducted for both general and small object detection tasks, with the comparison of numerous conventional methods on different datasets. DKT_1D in Tables 1 and 2 indicates that DKT is calculated using 1D convolution (DKT_Conv1D). DKT_2D indicates that DKT is calculated using 2D convolution (DKT_Conv2D).

4.3. Results on general object detection

In order to prove the generalization and performance of the proposed method, we compare the proposed method with several object detectors on the PASCAL VOC2007 test set for general object detection. As shown in Table 1, our proposed method DKT_Conv2D is compared with 15 state-of-the-art methods, which shows that the DKT_Conv2D achieves the best performance in 14 out of 20 object categories. It proves the superiority of DKT_Conv2D. The detection effect of DKT_Conv1D is superior to that of DKT_Conv2D, which also shows that the move from 2D to 1D is a positive exploration process. It can be concluded from Table 1 that the DKT_Conv1D surpasses all other methods with the best performance in 12 of 20 categories. For classes that occupy a major area in the image, such as airlines, bicycles, horses and trains, the DKT_Conv1D in this paper obtains a significant boost of 4% - 14% compared to the benchmark Faster R-CNN. For some medium objects, such as boats, tables and chairs, the method in this paper has a boost of about 7% - 15% compared to the benchmark model Faster R-CNN. For some small targets, such as bottles, chairs and plants, our model shows a significant improvement of 14% - 25% over the benchmark Faster R-CNN. As can be seen from the table above, the performance of our method for target detection is optimal in many categories and outperforms most other object detection algorithms.

Table 2
Performance comparison with existing target detection methods on Safety Helmet Wearing Datasets. The best results are highlighted in bold and the second best results are underlined.

	Metrics	YOLOv3 [16]	SSD [15]	Faster R-CNN [13]	YOLOv5 [25]	Sparse R-CNN [58]	Ours (DKT_2D)	Ours (DKT_1D)
GDUT [46]	mAP50	76.78	72.64	82.5	85.9	86.7	88.2	**88.3**
	mAP	48.46	46.4	43.8	54.9	55.2	58.6	**58.8**
	mAP_L	64.8	62.7	64.8	69.0	71.7	72	**72.8**
	mAP_M	47.9	44.8	59.9	64.2	66.0	**69.9**	67.7
	mAP_S	35.5	36.7	39.3	43.4	45.3	47.4	**47.6**
SHW1 [47]	mAP50	78.53	63.8	82.0	84.7	78.0	87.7	**88**
	mAP	42.63	30.67	49.8	51.6	46.5	**53.8**	53.6
	mAP_L	59.3	44.4	63.5	64.3	62.4	**65.3**	62.4
	mAP_M	45.9	32.5	59.6	60.9	58.8	61.8	**62.2**
	mAP_S	27.2	18.3	32.7	34	32.1	36.1	**37.0**
SHW2 [48]	mAP50	84.43	85.22	85.0	86.3	81.7	**92.8**	91.4
	mAP	54.3	52.79	54.1	55.2	46.4	**59.3**	**59.3**
	mAP_L	65.8	67.5	71.1	71.0	68.5	74.7	**74.8**
	mAP_M	58.4	54.9	59.7	60.2	59.0	**65.3**	**65.3**
	mAP_S	32.0	31.4	37.6	38.6	36.7	39.5	**43.8**

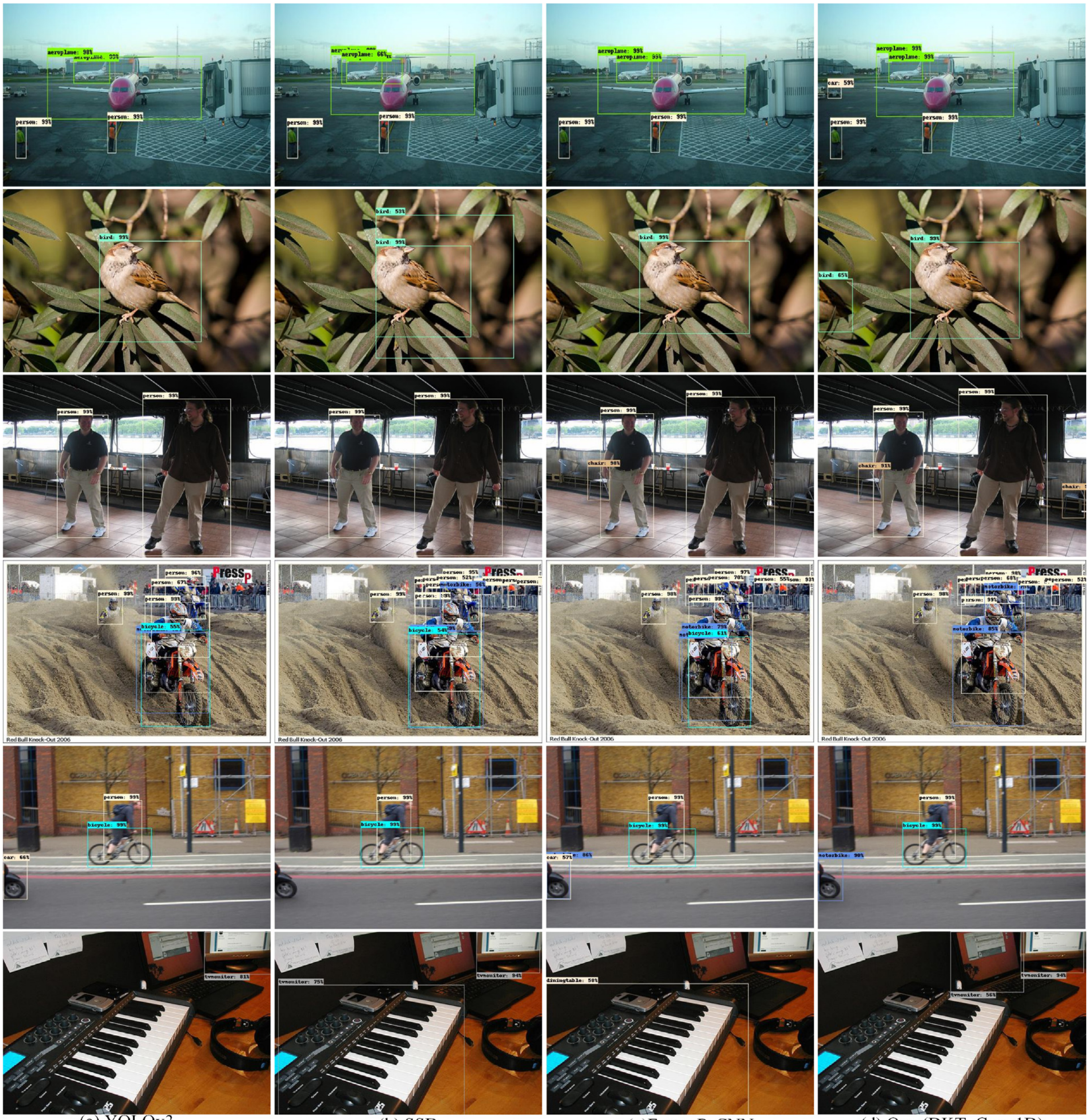
Fig. 6 shows the detection results of different methods. The top three rows of Fig. 6 illustrates YOLOv3, SSD, and Faster R-CNN tend to dismiss some tiny objects, while the DKT_Conv1D is more generalized and robust to detect both big objects and tiny objects. As shown in the bottom three rows the compared methods mostly produce errors when recognizing the objects. In contrast, our method is capable of extracting more identified features and output more faithful recognition results.

4.4. Results on small object detection

We have chosen Safety Helmet Wearing Detection, one of the fundamental research hotspots in industrial safety, to study the effectiveness of our method on small object detection. We evaluate our model on GDUT [46], SHW1 [47], and SHW2 [48], in which the safety helmets occupy small areas in the images. As shown in Table 2, DKT_Conv2D achieves the best detection performance compared to all five conventional methods on three datasets, and DKT_Conv1D further improves the detection performance on top of DKT_Conv2D. This is owing to the fact that the 1D convolutional computation computes the tokens for significant flexibility and universality, which helps for performance improvements.

4.4.1. Results on GDUT [46]

The results in Table 2 demonstrate our methods perform consistently better than other methods. Specifically, DKT_Conv1D achieves 88.3% mAP50 and 55.8% mAP on the GDUT. We can notice that the mAP50 of DKT_Conv1D exceeds Faster R-CNN + STL (Swin Transformer) by 1%, exceeds Spares R-CNN by 1.6%, exceeds YOLOv3 by 11.2%, and exceeds SSD by 15.36%. Compared with several other methods, the mAP of DKT_Conv1D also exceeds them a lot.



(a) YOLOv3 (b) SSD (c)Faster R-CNN (d) Ours(DKT_Conv1D)

Fig. 6. Visual results of different methods on PASCAL VOC2007 [45]. We use Dual-Key Transformer Network for PASCAL VOC2007 to better distinguish the objects.

Fig. 7(a) shows that YOLOv3 mistakenly detects a helmet on the table as a helmet worn on the head. And Fig. 7(b) illustrates that SSD fail to handle the crowed scenarios and dismissing more helmets. The DKT_Conv1D can solve these two problems comparing to YOLOv3 and SSD. Although the visualization of Faster R-CNN performs better in Fig. 7. DKT_Conv1D has larger confidence for the detection of the occluded dense targets.

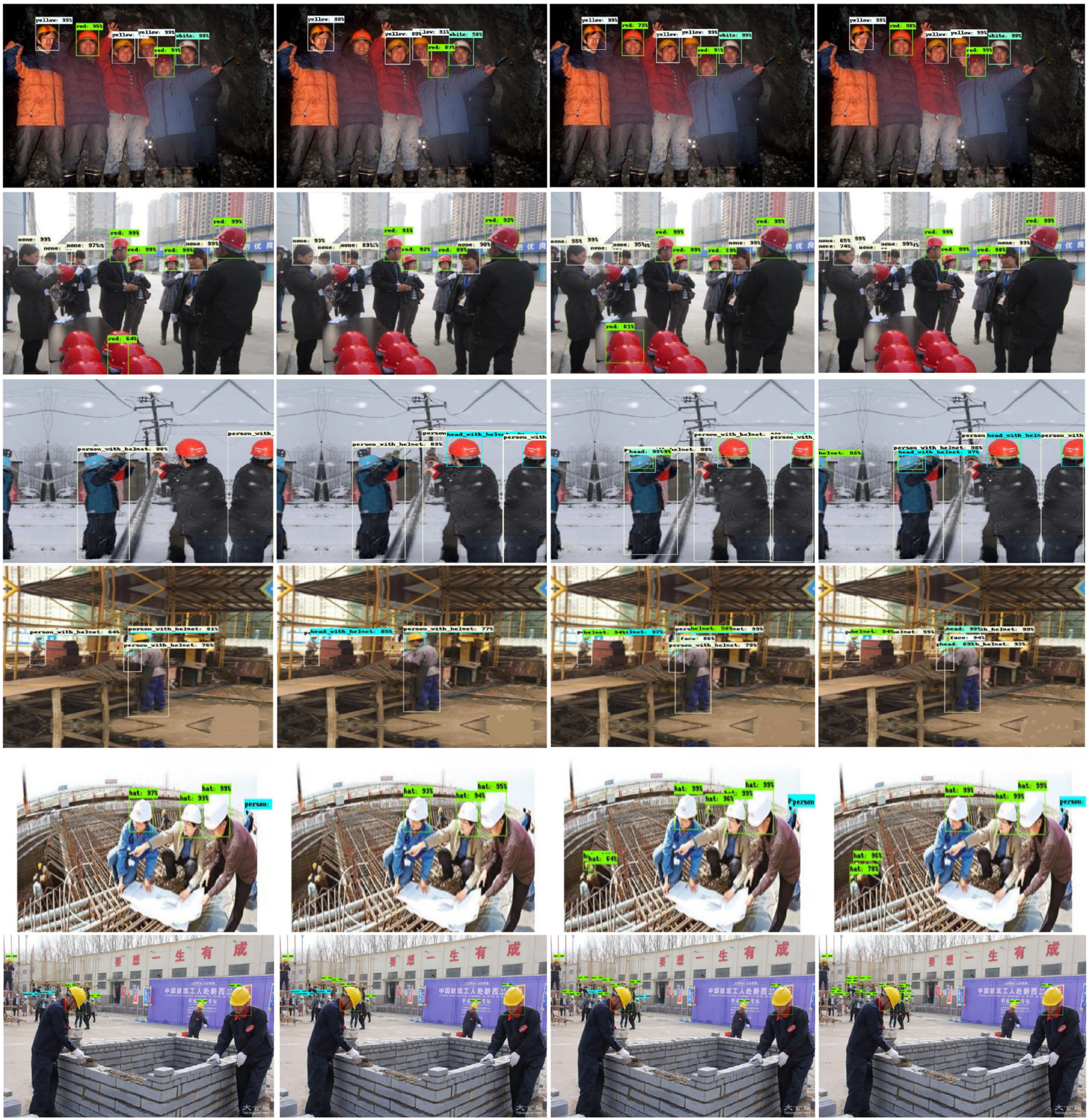
4.4.2. Results on SHW1 [47]

As shown from Table 2, the mAP50 of DKT_Conv1D on SHW1 is the highest reaching 88%, and the mAP is also the highest reaching 53.6%. As shown in the middle two rows of Fig. 8, YOLOv3 [16], SSD [15], and Faster R-CNN [13] performs not that well for the small helmets in complex scenes, *e.g.*, snowy weather, the indoor with many stripes and different directions, and distant views. In contrast, DKT_Conv1D can deal with these problems.

4.4.3. Results on SHW2 [48]

Table 2 illustrates a quantitative comparison between conventional methods and ours on SHW2. Specifically, the mAP50 of DKT_Conv1D exceeds Faster R-CNN + STL by 1.9%, exceeds YOLOv5 by 5.1%, exceeds SSD by 6.18%. We notice that the mAP of DKT_Conv1D exceeds Faster R-CNN + STL by 2.4%, exceeds YOLOv5 by 5.2%, and exceeds SSD by 6.51%. The improvements shows that DKT_Conv1D is feasible.

As shown in Fig. 7, the other methods mostly fail to detect the distant objects (the fifth row of Fig. 7) or miss some objects for dense target objects (the bottom row of Fig. 7). In contrast,



(a) YOLOv3 (b) SSD (c)Faster R-CNN (d) Ours(DKT_Conv1D)

Fig. 7. Visual results of different methods on GDUT [46], SHW1 [47] and SHW2 [48]. From the visualization, it can be seen that the detection results and confidence of the method in this paper have achieved the best performance.

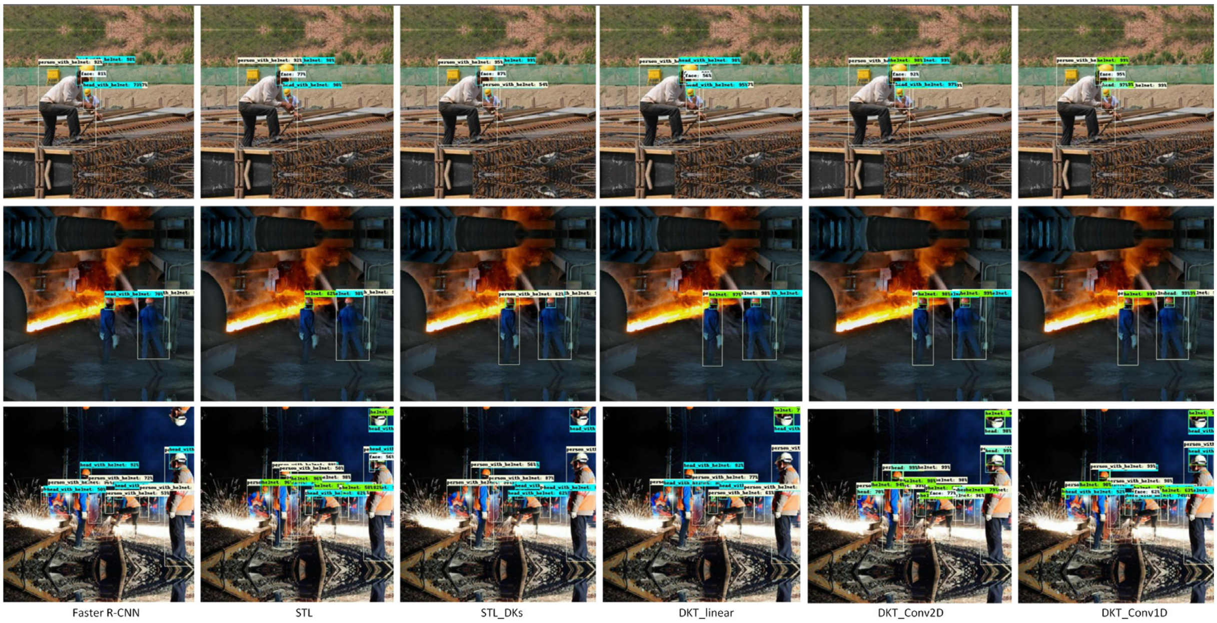


Fig. 8. Partial visual images improved by our method in each module of safety helmet dataset. Each column represents the visualization of a method. Through observation, it can be seen that our method has the highest detection accuracy and the better wholeness in complex environment.

DKT_Conv1D can detect all the small helmets with promising confidence scores.

4.5. Ablation studies

In this subsection, we will discuss the roles of the important components of the proposed network.

4.5.1. Transformer for object detection

To probe into the role of the transformer adopted in our network for object detection, we compare two baselines, including Faster R-CNN and Faster R-CNN + STL. As shown in the two columns of Table 3, the transformer architecture can improve the performance by strengthening the representation ability of the feature layers. Besides, as shown in the first two columns of Fig. 8, with the involvement of Transformer, some missed small helmets can be recognized, which is owing to the feature discrimination enhancement achieved by the transformer.

Table 3
Ablation experiments of Faster R-CNN vs. Faster R-CNN + STL on Safety Helmet Wearing Datasets. The best results are highlighted in bold.

Dataset	Metrics	Faster R-CNN	Faster R-CNN + STL
GDUT [46]	mAP50	82.5	**87.3**
	mAP	43.8	**57.7**
	mAPL	64.8	**72.4**
	mAPM	59.9	**66.7**
	mAPS	39.3	**46**
SHW1 [47]	mAP50	82	**87**
	mAP	49.8	**52.3**
	mAPL	63.5	**65.3**
	mAPM	59.6	**61.8**
	mAPS	32.7	**36.1**
SHW2 [48]	mAP50	85	**89.5**
	mAP	54.1	**56.9**
	mAPL	71.1	**73.3**
	mAPM	59.7	**64.6**
	mAPS	37.6	**39.6**

4.5.2. Dual-key strategy

To illustrate the role the dual-key strategy in the transformer, we compare two modified versions, including Faster R-CNN + Swin Transformer and Faster R-CNN + Swin Transformer + DKs. The performance comparison can be seen in the second and third rows of Table 4. It is apparent that the dual-key strategy improves the performance. Besides, as displayed in the second and third columns of Fig. 8, the confidence coefficient of safety helmet in dark scenes has been greatly improved by the dual-key strategy, which thanks to the dual-key strategy that enhances the coherence between Q and V to extract more discriminatory features. As shown in the second and third columns of Fig. 9, the detection effect after using dual keys is more remarkable than that calculated by one key from the feature maps. Specifically, the usage of dual keys can reduce the interference of the background on the detection target with more discriminative features, which helps to improve the detection accuracy.

In our network, we transpose the feature map to calculate the global self-attention in channel dimension rather than spacial dimension. To study the superiority of the channel attention over the spatial attention, we compare two versions, including Faster R-CNN + Swin Transformer + DKs with Faster R-CNN + DKT_linear. It can be found in Table 4 that the channel-wise self-attention achieves some performance gains than the spatial-wise self-attention. Besides, as shown in the third and fourth columns of Fig. 8, the confidence scores can be improved by the channel-wise self-attention, compared to the spatial-wise

Table 4
Results of ablation experiments on PASCAL VOC2007. The best results are highlighted in bold and the second best results are underlined.

Backbone	mAP50	mAp	mAP_L	mAP_M	mAP_S
Resnet50	78.8	47.8	44.3	32.7	24.7
Resnet50 + STL	82.7	51.3	54.9	41.4	27.4
Resnet50 + STL_DKs	82.9	51.3	54.7	41.9	29.2
Resnet50 + DKT_linear	83.1	52.0	55.5	42.0	31.9
Resnet50 + DKT_Conv2D	84.7	54.3	58.2	44.1	31.7
Resnet50 + DKT_Conv1D	**85.1**	**54.5**	**58.4**	**44.5**	**33.4**

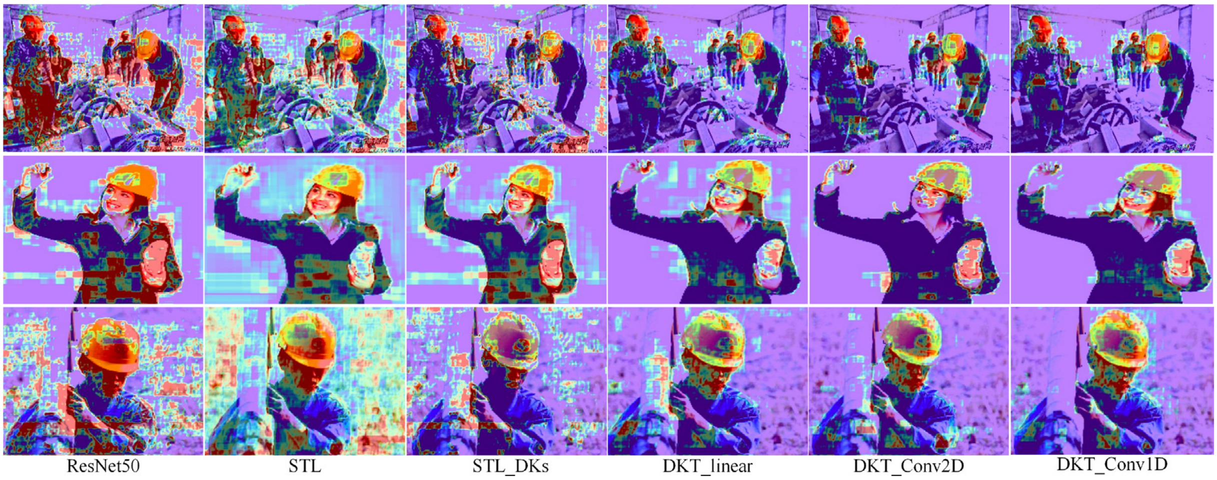


Fig. 9. Feature map visualization image of our method in the safety helmet dataset. The redder the color of the region in the image indicates the more attention the region receives.

attention architecture. As can be seen in the third and fourth columns of Fig. 9, using channel-wise attention to capture features is significantly more effective than spatial-wise attention.

4.5.3. Convolutional computation

To take insight into the convolutional computation for Q, K, and V, we compare the 2D/1D convolution based transformer and the fully-connected transformer. It can be found from the fourth row and the fifth row of Table 4 that the 2D convolutional computation can achieve the competitive performance. mAP$_L$, mAP$_M$, mAP$_S$ represent mAP (IoU = 0.5) of the best and the second-best results are highlighted and underlined. This improvement thanks to that the convolutional computation can capture the local context in feature layers. Besides, as illustrated in the fourth and fifth columns of Fig. 8, the 2D convolutional computation improves the detection confidence of small helmets than the conventional fully-connected computation manner.

To take deep insight into the convolutional computation, we comparing the bottom two rows of Table 4. It can be found that the 1D convolutional computation surpasses the 2D convolutional computation. With the same image size 640 × 640, the parameters number and FLOPS of Conv1D and Conv2D are listed as: Conv1D vs. Conv2D: 45.39 M vs. 56.16 M (Parameters) and 167.38G vs. 285.18G (FLOPS). It can be seen that Conv1D reduces the number of parameters and FLOPS by 10.77 M and 117.8GFLOPS, respectively, compared to Conv2D, which confirms that the Conv1D decreases the computation complexity significantly with respect to Conv2D without detection performance being sacrificed. As shown in the last two columns of Fig. 8, the 1D convolutional computation finds all the helmets compared to the 2D convolutional computation. This is owing to the fact that the the 1D convolutional computation computes the tokens for significant flexibility and universality, which helps for performance improvements. As shown in the last two columns of Fig. 9, using Conv1D to compute the feature map information is least affected by the background and most accurate for the target focus, thus improving the detection of small targets.

5. Conclusion

In the paper, transformer architecture DKTNet is proposed for the small object detection. Specifically, instead of directly embedding the transformer, we proposed a dual-key strategy to enhance the correlation between Q and V, which further improves the feature discrimination for most objects. Besides, to promote the important channels of features, we computed the channel attention instead of spatial attention in the transformer architecture. Moreover, in order to preserve the local context, we utilized the convolutional computation to substitute for the fully-connected computation for Q, K, and V in the transformer architecture. In the future, we will study robust deep learning [61] and part-whole relational saliency [62,63] to improve the robustness of our framework for various real-world scenes.

CRediT authorship contribution statement

Shoukun Xu: Investigation, Writing-review-editing. **Jianan Gu:** Visualization, Software, Methodology. **Yining Hua:** Writing-review-editing. **Yi Liu:** Supervision, Investigation, Writing-review-editing.

Data availability

Data will be made available on request.

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.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62001341 and Natural Science Foundation of Jiangsu Province under Grant BK20221379.

References

[1] Y. Shan, W.F. Lu, C.M. Chew, Pixel and feature level based domain adaptation for object detection in autonomous driving, Neurocomputing 367 (2019) 31–38.

[2] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, K. Dietmayer, Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges, IEEE Trans. Intell. Transp. Syst. 22 (3) (2020) 1341–1360.

[3] L.-H. Wen, K.-H. Jo, Deep learning-based perception systems for autonomous driving: A comprehensive survey, Neurocomputing.

[4] D. Cheng, J. Zhou, N. Wang, X. Gao, Hybrid dynamic contrast and probability distillation for unsupervised person re-id, IEEE Trans. Image Process. 31 (2022) 3334–3346.

[5] C. Wang, C. Xin, Z. Xu, M. Qin, M. He, Mix-vaes: A novel multisensor information fusion model for intelligent fault diagnosis, Neurocomputing 492 (2022) 234–244.

[6] D. Cheng, Y. Gong, X. Chang, W. Shi, A. Hauptmann, N. Zheng, Deep feature learning via structured graph laplacian embedding for person re-identification, Pattern Recogn. 82 (2018) 94–104.

[7] E. Jove, R. Casado-Vara, J.-L. Casteleiro-Roca, J.A.M. Pérez, Z. Vale, J.L. Calvo-Rolle, A hybrid intelligent classifier for anomaly detection, Neurocomputing 452 (2021) 498–507.

[8] S. Cheng, X. Chen, H. Qu, Rapid real-time collision detection for large-scale complex scene based on virtual reality, in: International Conference on Applications and Techniques in Cyber Security and Intelligence, Springer, 2021, pp. 605–610.

[9] L. Chen, H. Yang, Z. Gao, Person re-identification from virtuality to reality via modality invariant adversarial mechanism, Neurocomputing 414 (2020) 303–312.

[10] H. Tavakoli, S. Walunj, P. Pahlevannejad, C. Plociennik, M. Ruskowski, Small object detection for near real-time egocentric perception in a manual assembly scenario, arXiv preprint arXiv:2106.06403.

[11] W. Yan, Augmented reality instructions for construction toys enabled by accurate model registration and realistic object/hand occlusions, Virtual Real. (2021) 1–14.

[12] N. Xu, C. Huo, X. Zhang, Y. Cao, G. Meng, C. Pan, Dynamic camera configuration learning for high-confidence active object detection, Neurocomputing 466 (2021) 113–127.

[13] 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.

[14] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[16] R. Huang, J. Pedoeem, C. Chen, Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 2503–2510.

[17] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, S. Yan, Perceptual generative adversarial networks for small object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1222–1230.

[18] L. Huang, Q. Fu, M. He, D. Jiang, Z. Hao, Detection algorithm of safety helmet wearing based on deep learning, Concurr. Comput.: Pract. Exp. 33 (13) (2021).

[19] S. Guo, D. Li, Z. Wang, X. Zhou, Safety helmet detection method based on faster r-cnn, in: International Conference on Artificial Intelligence and Security, Springer, 2020, pp. 423–434.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[21] D. Rothman, Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing Ltd., 2021.

[22] D. Cheng, T. Liu, Y. Ning, N. Wang, B. Han, G. Niu, X. Gao, M. Sugiyama, Instance-dependent label-noise learning with manifold-regularized transition matrix estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16630–16639.

[23] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, Advances in neural information processing systems 29.

[24] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al., Sparse r-cnn: End-to-end object detection with learnable proposals, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 14454–14463.

[25] X. Zhu, S. Lyu, X. Wang, Q. Zhao, Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2778–2788.

[26] Z. Shao, J. Han, D. Marnerides, K. Debattista, Region-object relation-aware dense captioning via transformer, IEEE Transactions on Neural Networks and Learning Systems.

[27] C. Chen, K. Debattista, J. Han, Semi-supervised object detection via virtual category learning, arXiv preprint arXiv:2207.03433.

[28] Y. Miao, Z. Lin, X. Ma, G. Ding, J. Han, Learning transformation-invariant local descriptors with low-coupling binary codes, IEEE Trans. Image Process. 30 (2021) 7554–7566.

[29] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.

[30] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 821–830.

[31] G. Ghiasi, T.-Y. Lin, Q.V. Le, Nas-fpn: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7036–7045.

[32] B. Singh, L.S. Davis, An analysis of scale invariance in object detection snip, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3578–3587.

[33] G. Cheng, J. Han, P. Zhou, L. Guo, Multi-class geospatial object detection and geographic image classification based on collection of part detectors, ISPRS J. Photogramm. Remote Sens. 98 (2014) 119–132.

[34] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: Deconvolutional single shot detector, arXiv preprint arXiv:1701.06659.

[35] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection, in: European conference on computer vision, Springer, 2020, pp. 566–583.

[36] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: Top-down modulation for object detection, arXiv preprint arXiv:1612.06851.

[37] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30.

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16 × 16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929.

[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European conference on computer vision, Springer, 2020, pp. 213–229.

[41] Z. Wu, C. Shen, A. Van Den Hengel, Wider or deeper: Revisiting the resnet model for visual recognition, Pattern Recogn. 90 (2019) 119–133.

[42] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks., in: ICML, vol. 2, 2016, p. 7.

[43] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450.

[44] A.R. Sutanto, D.-K. Kang, A novel diminish smooth l1 loss model with generative adversarial network, in: International Conference on Intelligent Human Computer Interaction, Springer, 2020, pp. 361–368.

[45] M. Everingham, A. Zisserman, C.K. Williams, L. Van Gool, M. Allan, C.M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, et al., The pascal visual object classes challenge 2007 (voc2007) results.

[46] J. Wu, N. Cai, W. Chen, H. Wang, G. Wang, Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset, Autom. Constr. 106 (2019).

[47] M.-E. Otgonbold, M. Gochoo, F. Alnajjar, L. Ali, T.-H. Tan, J.-W. Hsieh, P.-Y. Chen, Shel5k: an extended dataset and benchmarking for safety helmet detection, Sensors 22 (6) (2022) 2315.

[48] njvisionpower, Safetyhelmetwearing-dataset,https://github.com/njvisionpower/Safety-Helmet-Wearing-Dataset, online accessed 17 Dec 2019.

[49] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, J. Sun, Megdet: A large mini-batch object detector, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6181–6189.

[50] S.-H. Bae, Object detection based on region decomposition and assembly, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 8094–8101.

[51] H. Wang, Q. Wang, M. Gao, P. Li, W. Zuo, Multi-scale location-aware kernel representation for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1248–1257.

[52] L. Zheng, C. Fu, Y. Zhao, Extend the shallow part of single shot multibox detector via convolutional neural network, Tenth International Conference on Digital Image Processing (ICDIP 2018), vol. 10806, SPIE, 2018, pp. 287–293.

[53] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, J. Wu, Feature-fused ssd: Fast detection for small objects, Ninth International Conference on Graphic and Image Processing (ICGIP 2017), vol. 10615, SPIE, 2018, pp. 381–388.

[54] P. Zhou, B. Ni, C. Geng, J. Hu, Y. Xu, Scale-transferrable object detection, in: proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 528–537.

[55] C. Termritthikun, Y. Jamtsho, J. Ieamsaard, P. Muneesawang, I. Lee, Eeea-net: An early exit evolutionary neural architecture search, Eng. Appl. Artif. Intell. 104 (2021).

[56] C. Song, X. Cheng, L. Liu, D. Li, Acfim: Adaptively cyclic feature information-interaction model for object detection, in: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Springer, 2021, pp. 379–391.

[57] A. Bar, X. Wang, V. Kantorov, C.J. Reed, R. Herzig, G. Chechik, A. Rohrbach, T. Darrell, A. Globerson, Detreg: Unsupervised pretraining with region priors for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14605–14615.

[58] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al., Sparse r-cnn: End-to-end object detection with learnable proposals, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 14454–14463.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32.

[60] N.S. Keskar, R. Socher, Improving generalization performance by switching from adam to sgd, arXiv preprint arXiv:1712.07628.

[61] M. Ren, W. Zeng, B. Yang, R. Urtasun, Learning to reweight examples for robust deep learning, in: International conference on machine learning, PMLR, 2018, pp. 4334–4343.

[62] Y. Liu, D. Zhang, N. Liu, S. Xu, J. Han, Disentangled capsule routing for fast part-object relational saliency, IEEE Transactions on Image Processing.

[63] Y. Liu, D. Zhang, Q. Zhang, J. Han, Part-object relational visual saliency, IEEE Transactions on Pattern Analysis and Machine Intelligence.

Shoukun Xu received the Ph. D. degree from China University of Mining and Technology, China in 2001. He is currently a Professor at Changzhou University. He is Chair of China Computer Federation Changzhou Branch and the distinguished member of China Computer Federation. His research interests include digital twins, computer vision and blockchain.

Jianan Gu received the M.S. degree from Changzhou University, China in 2020. She is currently an M.S. candidate at ChangZhou University. Her research interests include computer version and deep learning, etc.

Yining Hua received the B.Eng. degree in information security from Northeastern University, Shenyang, China, in 2016, and the Ph.D. degree in computer science from the Department of Computer Science, Loughborough University, Loughborough, U.K., in 2020. She was a Postdoctoral Research Assistant with the School of Computer Science, University of Glasgow, Glasgow, U.K., in 2021. She is currently a Lecturer in computer science with the School of Arts, University of Roehampton, London, U.K. Her research interests include autonomous systems, Internet-of-Things, edge/fog computing, and next-generation networks.

Yi Liu received the Ph. D. degree from Xidian University, China, in 2019. He is currently a Professor at Changzhou University, China. From 2018 to 2019, he was a visiting scholar at Lancaster University. His research interests include machine learning and computer vision, especially on saliency detection, capsule network, 3D point cloud, and object detection.