Multimedia Systems (2023) 29:3639–3653
https://doi.org/10.1007/s00530-023-01185-9

REGULAR PAPER

Workpiece tracking based on improved SiamFC++ and virtual dataset

Kaisi Yang[1] · Lianyu Zhao[1] · Chenglin Wang[2]

Received: 4 April 2023 / Accepted: 12 September 2023 / Published online: 11 October 2023

Abstract

Datasets play a crucial role in the training of deep learning models. For industrial datasets, the collection and annotation of images and videos is time-consuming, labor-intensive, and error-prone. In the past decade, With the development of rendering technology and hardware capability, more and more researches tend to use virtual datasets to overcome the shortcomings of real datasets. We studied the method of expanding the small sample data set of sprayed workpieces to solve the positioning problem of sprayed workpieces. We build the 3D model of sprayed workpieces and the factory environment in the virtual environment. We use blender software to render workpieces in different environments, and automatically generate the ground-truth label. In order to verify the effectiveness of this expansion method, We use real dataset, virtual dataset, and mixed dataset for model training. In our study, enhancements were made to the SiamFC++ model. Specifically, the backbone network was replaced with the ConvNeXt model, which boasts superior feature extraction capability. Additionally, we innovated the loss function by transitioning from IoU loss to CIoU loss, thereby introducing penalty terms for central point distance and shape consistency. Within the experimental section, we compared the performances of the SiamFC++ model using the AlexNet backbone network and the ConvNeXt backbone network. When trained solely on real datasets, the accuracy rates of the two model versions were 80.1% and 80.5% respectively. With virtual dataset training, the accuracy rates of the two versions improved by 6% and 7.4% respectively. When trained on mixed datasets, the accuracy rates of the two model versions saw respective enhancements of 8% and 8.6%. In all three training conditions, the ConvNeXt-based version of the model consistently outperformed the AlexNet-based version. Our improved model was further compared to mainstream object tracking models to validate its tracking efficacy. To substantiate the effectiveness of our model enhancements, we performed comprehensive ablation studies.

Keywords Object tracking · Virtual dataset · Artificial scenes · 3D model rendering

1 Introduction

In recent decades, computer vision has achieved great development, which is mainly due to the improvement of computer hardware and the development of big data. Computer vision algorithms are increasingly used in industrial fields, such as workpiece positioning, defect detection, and visual servoing. The application of deep learning algorithms in the industrial field is a challenging research direction. Compared with other fields, it is difficult to establish a general data set in the industrial field. Industrial datasets are highly targeted. Generally, industrial datasets are established according to a specific requirement, and there are large differences between datasets. As a result, most industrial datasets are small and difficult to integrate. Image and video annotation work is often tedious and expensive. Moreover, it is difficult

Communicated by J. Gao.

✉ Chenglin Wang
163109306@stud.tjut.edu.cn

Kaisi Yang
296280063@qq.com

Lianyu Zhao
193102409@stud.tjut.edu.cn

[1] School of Computer Science and Engineering, Tianjin University of Technology, Liqizhaung Street, Tianjin 300384, China

[2] School of Mechanical Engineering, Tianjin University of Technology, Liqizhaung Street, Tianjin 300384, China

to accurately annotate manually. It is difficult to train a tracking model with high accuracy using a dataset with low quality and few samples. Early data collection mainly depended on manual work, and the collected pictures were labeled one by one. This requires a lot of manpower, material and financial resources. In recent years, the collection of many datasets mainly relies on crawlers, which in turn leads to the low quality of annotation of datasets. It is difficult to obtain a dataset with large data volume and accurate annotation.

Virtual modeling technology can solve this problem well. The difficulty of rendering is that the lighting effect of the real physical world needs to be simulated by computer, and all the characteristics of light such as direct light, reflection, scattering, diffuse reflection, diffraction, interference, light attenuation and so on need to be fully considered. The more sufficient the presentation of light, the more complex the calculation, and the greater the amount of calculation. In the past 10 years, due to the development of computer graphics and virtual reality technology, rendering technology has been greatly improved. From forward rendering, delayed rendering to ray tracing, rendering of virtual scenes has become more and more realistic. Support for translucent objects and hardware anti-aliasing has become better and better, and material system design has become more free [1–4].

In fact, with the development of computer graphics and rendering technology, researchers have been able to build realistic scenes in virtual environments, and the labeling of virtual scenes can also be automated. Therefore, we adopted advanced static rendering technology, established an accurate industrial scene environment and realistic 3D models of sprayed workpieces in Blender. We performed video rendering to automatically generate the virtual dataset. The advantages of virtual datasets are:

- A large amount of annotation data can be obtained simply;
- Sample annotation is simple, accurate and automatically;
- Easy to impose interference and improve the robustness of the algorithm;
- The lighting conditions and environment can be switched conveniently.

In order to compare with the small sample real dataset, we collected a small amount of real data and made manual annotations. We compared the tracker TransT (Transformer Tracking) [5] based on transformer with the tracker SiamFC++ based on Siamese network. In the case of small data sets, the accuracy, robustness and FPS of TransT are lower than those of SiamFC++. TransT is difficult to learn the similarity between two objects in the case of small datasets. The performance of SiamFC++ on small sample datasets is significantly better than that of TransT.

2 Related work

2.1 Real datasets

Today, most of the publicly available datasets are obtained from real scenes. The main computer vision tasks include image classification, semantic segmentation, object detection, object tracking. Typical datasets include ImageNet [6], MNIST [7], CIFAR-100 [8], KITTI [9], LaSOT [SPScite1citeSPS][SPScite29citeSPS]. The ImageNet dataset contains 14,197,122 annotated images. It is a well-known image classification dataset. The MNIST dataset contains a large number of pictures of handwritten digits. The CIFAR-100 dataset consists of 60,000 images of size 32*32 divided into 100 categories. The KITTI dataset is one of the most commonly used datasets in the field of autonomous driving. Both the LaSOT dataset and the GOT dataset are commonly used for single-target tracking. The GOT dataset contains more than 10,000 manually labeled videos, and the targets are objects moving in the real world. The LaSOT dataset contains 70 categories, and each category contains 20 sequences. All videos are annotated with high quality. But inevitably they will spend a lot of time on ground-truth annotations. The more detailed the annotations, the heavier the workload for manual labeling. These publicly available generic object datasets are all well studied in the field of computer vision. However, there are few task-specific datasets (e.g., industrial object tracking). researchers often need to make their own datasets according to specific problems. The collection and labeling of datasets will take a lot of time.

2.2 Virtual datasets

Some researchers have conducted research on virtual datasets. They have applied them in research areas such as object detection, multimodal analysis, and eye-tracking [12–16]. Xuan et al. established a virtual traffic scene in Unity3D, which simulates real environmental changes [17]. They have automatically generated ground-truth labels in Unity3D, including semantic/instance segmentation, object bounding boxes, and so on. They combined the generated data with real data, and proved that adding virtual data can help improve the accuracy of deep learning models. Oliver et al. [18] established a facial expression dataset, which consists of 640 facial images from 20 virtual characters. The age of the character is between 20 and 80. This dataset can be used in the research of emotion recognition. Montulet et al. [19] collected the data in GTA5 and established a virtual pedestrian dataset under various scenes, environments, lighting and weather conditions. The annotation content is very rich, including boxes, bones, traceries, and segmentation. The data also includes depth channels. This dataset can be used for computer vision tasks, including pose estimation, person detection, segmentation, re-identification and tracking, individual and crowd activity recognition, and abnormal event detection. Jeon et al. [20] simulated scenes before and after disasters in a virtual environment, such as fire and building collapse. The dataset contains more than 300K high-resolution stereo image pairs, all of which are labeled in detail for semantic segmentation, surface normal estimation and camera pose estimation. Shen et al. [21] proposed a panoramic virtual dataset, and they created an automatically generated urban scene. In this scene, they collect data and annotate dataset automatically. These researches have made great contributions to the dataset expansion. Researchers can easily obtain a large number of pictures and videos in a virtual environment, and can automatically label without heavy data collection and labeling Due to the performance of rendering software or hardware, the rendering effect of many models may not be very realistic. This is mainly reflected in the treatment of sub surface scattering, global lighting, reflection, shadow and other effects. The texture of the model may be different from the real environment. The model may not be able to learn some key features, such as texture features or material features. This may have a certain impact on the learning of the model.

2.3 Object tracking algorithm

Object tracking algorithms have made great progress in recent decades. The detection accuracy and algorithm performance are gradually increasing. Industrial workpiece tracking is also an important application direction. Industrial workpiece tracking can be used to locate the workpiece in real time. Object trackers mainly include correlation filtering based trackers, Siamese network based trackers and transformer based trackers. There are two main training methods for small data sets. The first is to migrate models trained on large data sets, such as COCO [22] and GOT, to small data sets. Some Siamese network based trackers such as SiamFC++ [23], SiamRPN++ [24], etc. use this method for training. The second is to obtain a trained backbone network through self-supervision learning of a large number of unlabeled data. For example, He et al. trained Vit Large/Huge [25] on ImageNet through the unlabeled MAE [26] pre-training.

Since the sprayed workpieces are industrial parts, their characteristic distribution is different from animals, vehicles, people, etc., we do not adopt the training strategy of transfer learning. Self-supervised learning requires the collection of large amounts of data, and we did not use this approach.

3 Construction of sprayed workpieces dataset

The sprayed workpieces dataset is used to train the tracking model of sprayed workpieces. The dataset provides a large number of video data with accurate annotation. In the production process of workpieces, researchers often use tracking algorithms to track the target workpiece. Since object tracking algorithms require large amounts of training data, using virtual datasets to train trackers is an actively explored topic. Virtual datasets can provide potentially infinite annotation data. In this section, we will introduce the collection process of sprayed workpieces dataset in detail. Data collection and labeling mainly include the following steps:

1. Build a real spraying production line, including conveyors, robotic arm, camera, drying boxes;
2. Build a production line model and workshop environment in virtual environment(including the whole workshop, equipment, cabinets, sundries, natural light, lighting, etc.);
3. Use a camera to capture video data of workpieces and manually annotate them in real environment;
4. Automatically generate annotation data for artifacts in virtual environment.

3.1 Establish real dataset

The purpose of creating the sprayed workpieces dataset is to train the tracker, so that the tracker can be used to solve the positioning problem of sprayed workpieces. During the production process, the sprayed workpiece can be positioned to facilitate spraying. The overall structure of the spraying production line is shown in Fig 1.

The mechanical arm we used is the FANUC six-axis industrial robot M10ID/12, the robot mass is 145 kg, the maximum load is 12 kg, the motion radius is 1441 mm, and the repeated positioning accuracy is ± 0.02 mm. The camera we used is the CA-G300C produced by HengDa Company, the photosensitive element is CMOS, the resolution is 1600*1200, the communication interface is Ethernet, and the pixel size is 3.5 µm. We calibrated the camera [27].

As shown in Fig. 2, the spraying production line is mainly composed of a robotic arm, a drying box, a controller, a conveyors and workpieces. The workflow includes the following:

1. The conveyor belt transport workpieces from the preparation area to the spray area;
2. The camera mounted on the spraying robot arm locates the sprayed workpiece;

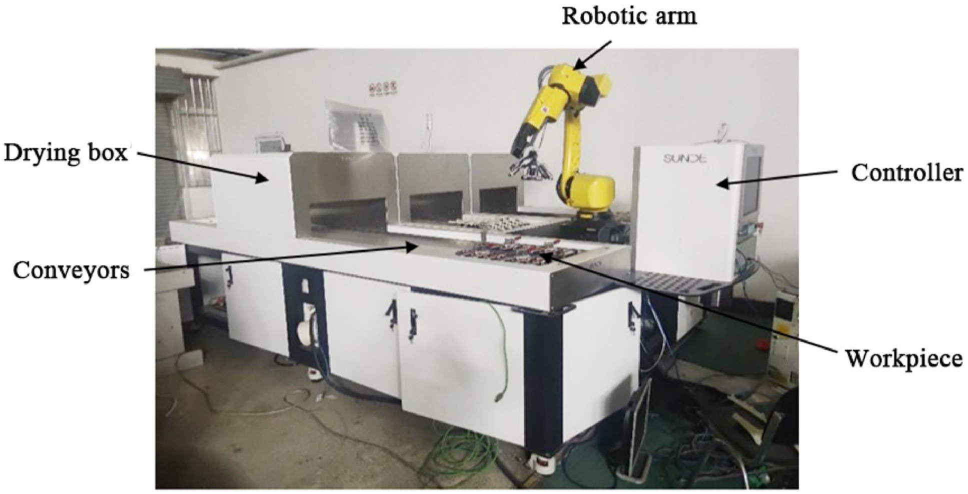

Fig. 1 Composition of spraying production line

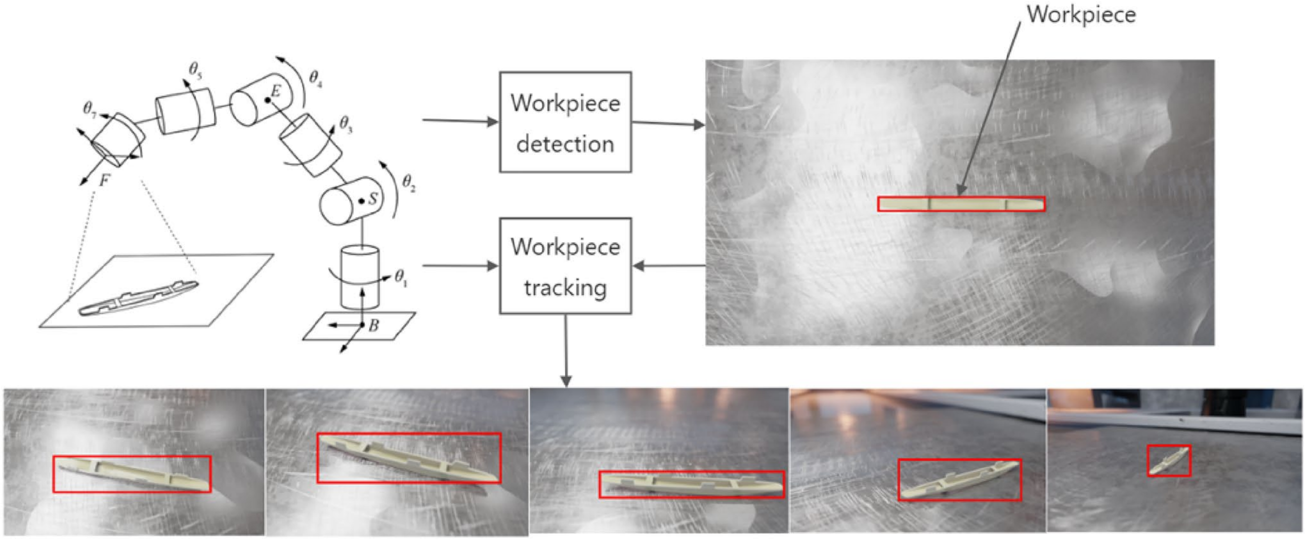

Fig. 2 Positioning of sprayed workpiece

3. The robotic arm sprays the workpiece according to the positioning position;
4. The conveyor belt returns the sprayed workpieces to the preparation area.

Mounting the camera on the robotic arm, real-time positioning of the spray target can be achieved through the use of a tracker. During the spraying process, maintaining accuracy is crucial due to the constantly changing relative positions between the robotic arm and the workpiece. By continuously monitoring and analyzing the image data captured by the camera, the system can rapidly and accurately calculate the current position of the target. This real-time positional information is then transmitted to the control system of the robotic arm for prompt adjustments of its posture and position. The robotic arm can closely follow the movements of the spray target. The red boxed area in the image represents the object being sprayed. Figure 3 shows the sprayed workpiece

We collected video data of a total of seven kinds of sprayed workpieces, and manually annotated the data. The data set is shown in Table 1.

3.2 Establish virtual dataset

In order to obtain a more realistic effect, we built the 3D model of the workshop scene and equipment in Blender, and added objects such as lighting, sky ball, and ground. Blender is a free and open source 3D graphics and image software. It provides a range of animation short film production solutions from modeling, animation, materials, rendering, to audio processing, video editing, and more.

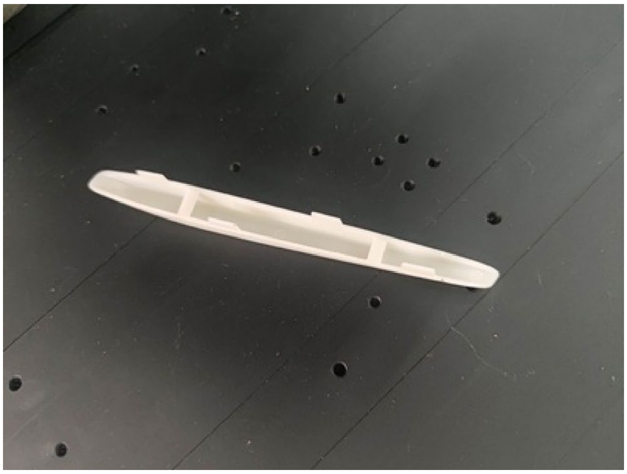

Fig. 3 Sprayed workpiece

Table 1 Real dataset of sprayed workpiece

Sprayed workpiece	Number of videos	Video length (frame)
1	8	219–577
2	13	168–292
3	14	187–299
4	14	179–245
5	15	165–231
6	12	171–235
7	8	288–349

We created the virtual dataset through the following steps:

1. Import the workpiece model. Ensure that the model's geometry and details are appropriate to simulate a real spraying scenario.
2. Create a spraying material for the workpiece in Blender. This includes texture information such as base color, roughness, metallicity, and normal maps to simulate different surface reactions.
3. Adjust the lighting and environmental settings of the scene to ensure that the workpiece looks realistic in the virtual environment.
4. Arrange cameras to achieve suitable angles and coverage.
5. Create an animation sequence that involves rotating and moving the workpiece to capture images from different perspectives. This enhances the diversity of the dataset.
6. Render each frame and save them as image files.
7. Add labels or annotations on the images to identify areas subjected to virtual spraying and other relevant information.
8. Export all rendered images and associated data as the dataset, including image files, workpiece models, material settings, and annotation data.

Creating a virtual dataset, especially for machine vision tasks, typically involves considering various aspects such as camera parameters, lighting conditions, materials, and scene setup. We use Blender to model the spray-painted workpiece. Appropriate materials are applied to the model to simulate real-world surface characteristics. Proper light sources are added to the scene to mimic lighting conditions in actual usage environments. A camera is positioned in the scene to emulate the camera's position during dataset capture, with a focal length of 24 mm and aperture of f5.6. In the rendering settings, a resolution of 640x480 is set for output. Rendered images and annotation information are saved. Blender offers both the Cycles renderer and the real-time rendering engine EEVEE. We use the Cycles renderer to render the scene model, ensuring a sufficiently realistic effect. The rendered effect of the scene in Blender is shown in the Fig. 4.

We established 3D models of six kinds of workpieces, we imported the 3D models of workpieces into Blender, and designed the trajectory of workpieces and cameras. Each

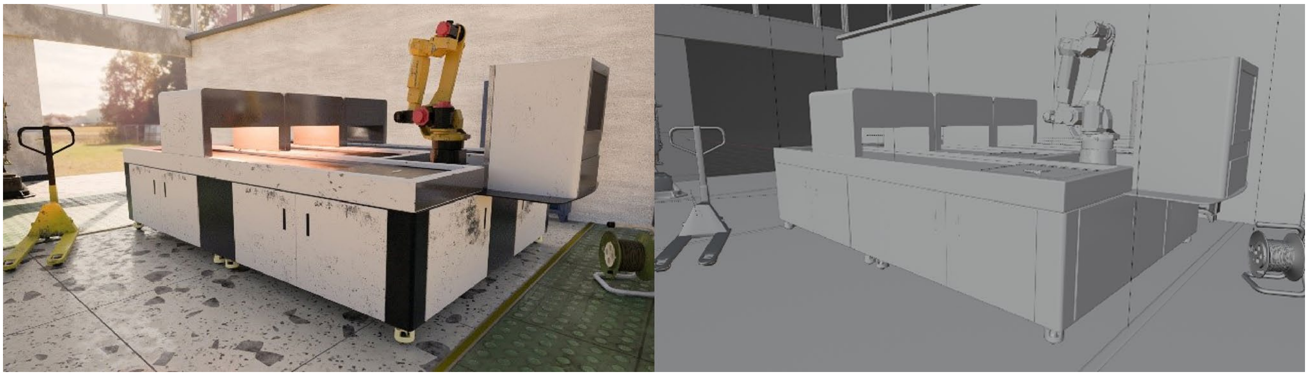

Fig. 4 Rendering model of spraying production line

workpiece renders several scenes and actions. The dataset is shown in Table 2.

Figure 5 shows the modeling and rendering effect of workpieces in virtual dataset. Because the factory environment is relatively fixed, we only render different lighting conditions. The rendering effect under different lighting conditions is shown in the Fig. 6.

Due to the absence of a standardized measure for evaluating feature distribution, during the process of generating virtual data, we relied on manual supervision to assess the similarity between the generated data and the virtual data, scrutinizing each generated video segment for authenticity. We aimed to align the feature distribution of virtual data as closely as possible to real data. However, disparities between virtual and real data are inevitable. To mitigate this, we diligently collected a substantial amount of video data from real datasets, striving to balance distribution discrepancies.

Table 2 Virtual dataset of sprayed workpiece

Sprayed workpiece	Number of videos	Video length (frame)
1	42	700
2	38	700
3	29	700
4	40	700
5	29	700
6	49	700

4 SiamFC++ based on ConvNeXt

We apply the tracking model to the spraying production line, which is mainly divided into two stages. In the first stage, we train a tracker to track the workpiece in the video. In the second stage, the position of the workpiece is identified in real time by the camera installed on the Mechanical arm. The Mechanical arm sprays the workpiece accurately according to the identified position. As shown in the Fig. 7.

4.1 Network structure

The method of SiamFC++ is to divide the tracking problem into two branches: classification and regression. The classification branch is responsible for accurate estimation of the position, and the regression branch is responsible for the regression of the bounding box. Traditional tracking algorithms rely on prior knowledge. For example, SiamFC [28] needs to perform three scale transformations on images, and SiamRPN [29] needs to design a priori frame for anchors, which will lead to poor model generalization ability.

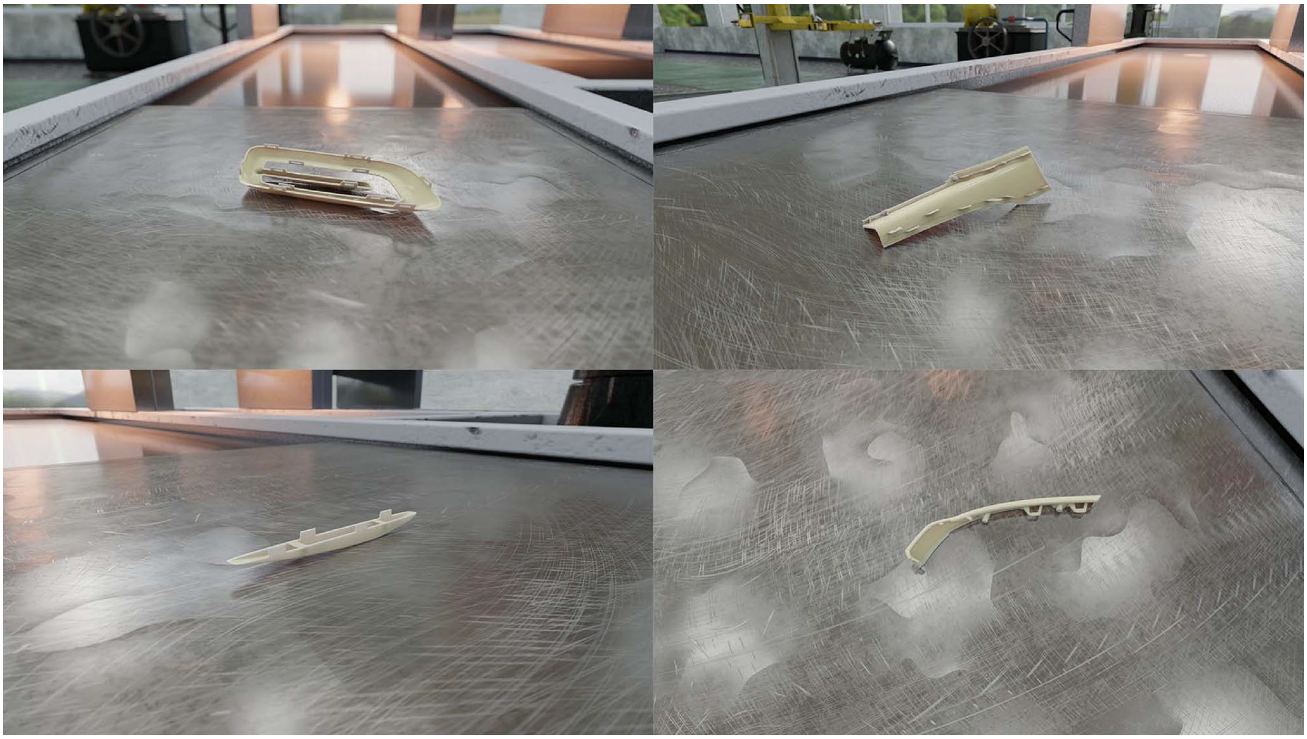

Fig. 5 Rendering effect of four kinds of workpieces

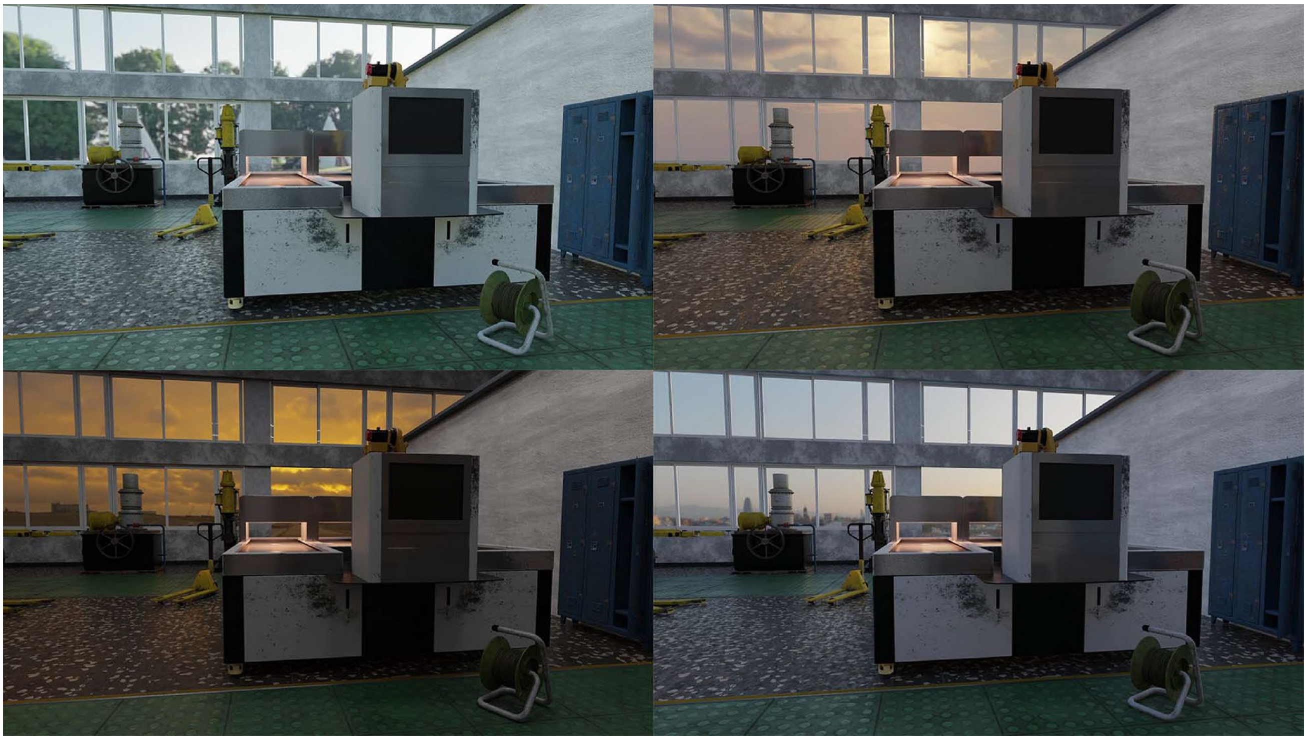

Fig. 6 Rendering effects under different lighting environments

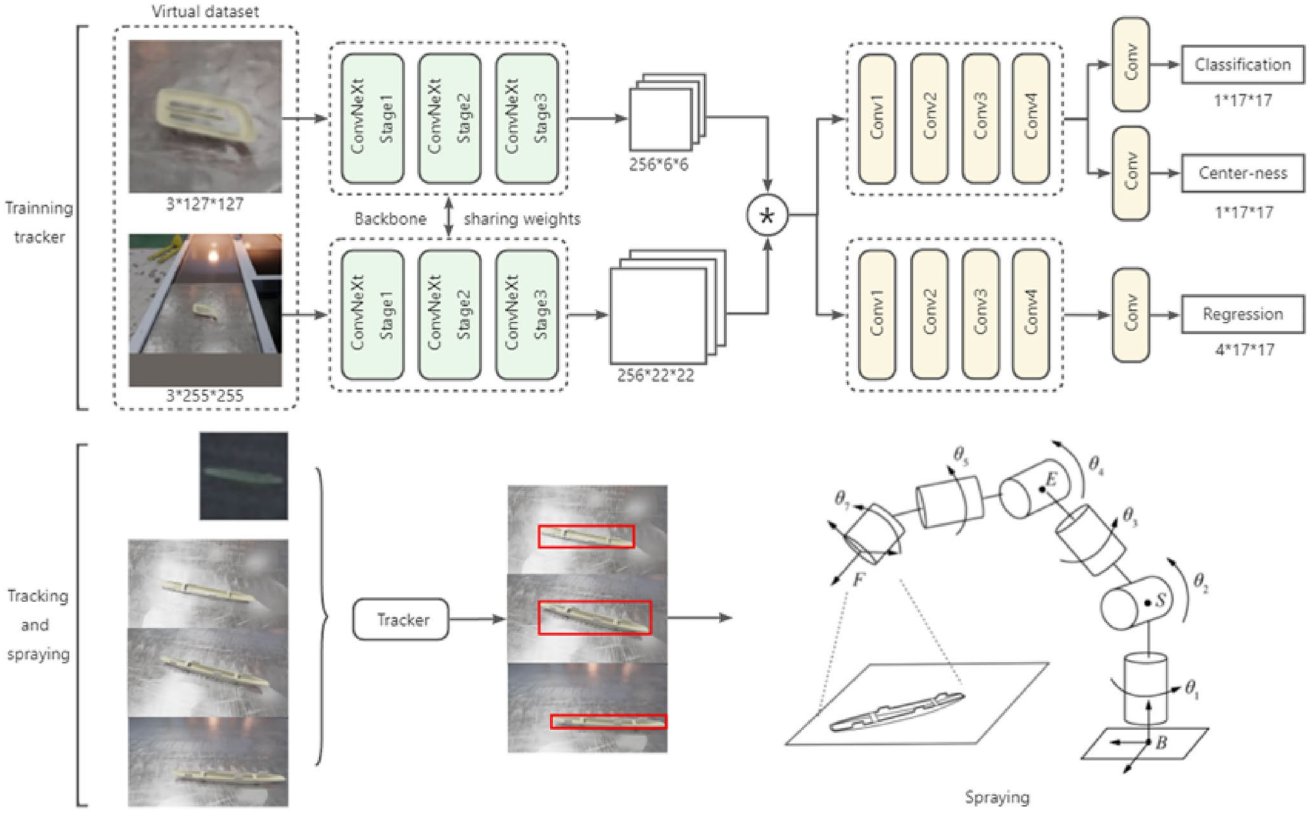

Fig. 7 Rendering effects under different lighting environments

The ConvNeXt model is a fully convolutional network proposed by Liu et al. in 2022 [30]. In ConvNeXt, its optimization strategy draws on Swin-Transformer [31]. The optimization strategies include: modifying the block structure of Resnet [32]; changing the optimizer from SGD to AdamW [33]; adding regular strategies, such as Stochastic Depth [34], Label Smoothing [35]. ConvNeXt can increase the depth and width of the model without adding a large number of parameters. This parameter efficiency enables the model to better adapt to larger image datasets while reducing the risk of overfitting. ConvNeXt performs exceptionally well in image classification tasks, demonstrating better classification performance compared to some traditional CNN architectures. It achieves competitive results on multiple benchmark image classification datasets, showcasing its superior capabilities in feature extraction and representation learning.

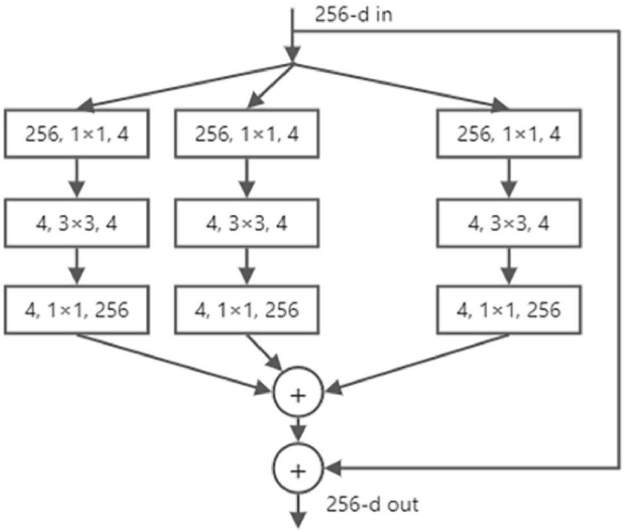

Fig. 8 The block structure of ConvNeXt

The block structure of the ConvNeXt model is shown in Fig. 8, and the depth-wise convolution [36] is used to form the convolution block. This structure greatly reduces the parameter scale of the network on the premise of sacrificing some accuracy.

We replace the backbone of the SiamFC++ model by Alexnet [37] with the ConvNeXt. In order to reduce the model parameters and ensure the output size of the backbone network, we adjust the number of blocks in each stage from (3, 3, 9, 3) in ConvNeXt tiny to (2, 2, 6). Notably, researchers have thoroughly investigated the distribution of computation [38, 39], and a more optimal design is likely to exist. Figure 9 shows the SiamFC++ model with ConvNeXt as the backbone network.

The tracker is designed in four steps:

1. Crop the pictures in the dataset to 127*127 template and 511*511 search region. The template is the ground-truth of the first frame, and the search region is the candidate box search region in the subsequent frames.
2. Input the template and search region into the backbone network. After feature extraction by the backbone, the output of the template branch is 17*17, and the output of the search region branch is 25*25.
3. Take the output of the template branch as the convolution kernel, and convolve the output of the search region branch to obtain the score map of the search region, which representing the similarity between each location in the search region and the template.
4. Input the score map into the classification branch and regression branch respectively. The classification branch and regression branch are respectively built by four layers of convolution layers. The classification branch classifies the pixels in the score map, and the regression branch is used to predict the bounding box.

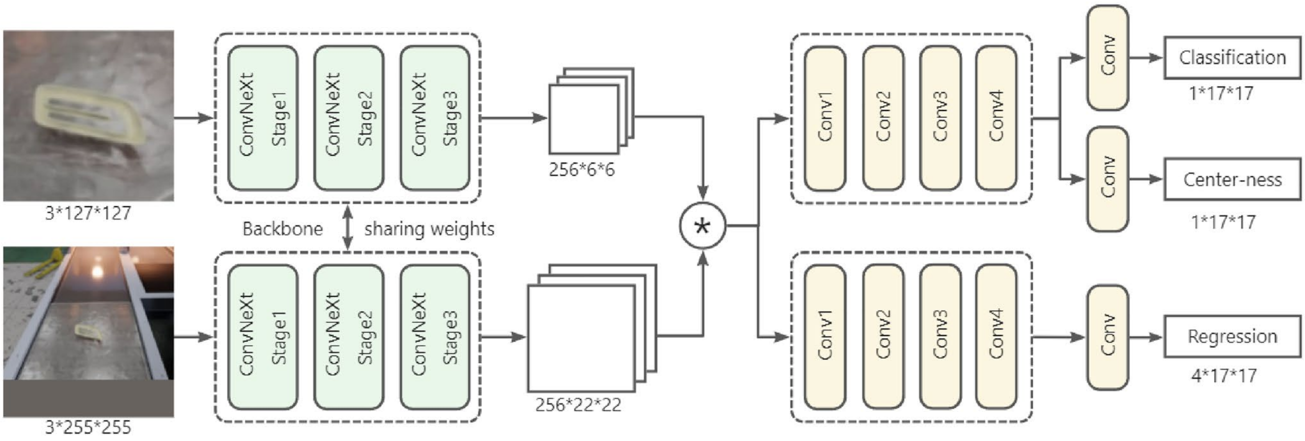

Fig. 9 Structure of the tracker

4.2 Loss function

The loss function of the model consists of three parts, including Center-ness loss, CIoU loss and classification loss. Multiply the three parts by the corresponding weights and add them together. The model updates parameters based on the final total loss.

4.2.1 Center-ness loss

Tian et al. proposed the FCOS network [40], which is a one stage anchor-free object detection network. In order to enhance the robustness of the algorithm, the network learns the center-ness parameter to suppress the weight of edge points. In the SiamFC++ network, the center-ness branch is a 17*17 matrix, and the score calculation formula is:

$$\text{centerness} = \sqrt{\frac{\min(l, r)}{\max(l, r)} \times \frac{\min(t, b)}{\max(t, b)}} \tag{1}$$

The closer the point is to the center, the closer the value of centerness is to 1. The closer the point is to the edge, the closer the value of centerness is to 0. This achieves the effect of reducing the weight of positions farther from the center of the target frame. The center-ness loss function is added as a branch to the total loss function. The algorithm generates multiple candidate boxes, some of which may overlap or even contain the same object. The fundamental idea behind the Non-Maximum Suppression (NMS) algorithm is to sort the object boxes for each class based on their confidence scores (or scores). Starting with the box with the highest score, it iterates through each box and compares its overlap with the rest of the candidate boxes. If the overlap exceeds a certain threshold, the box is removed from the candidate list, retaining only the box with the highest score. This ensures that each object is only kept once, preventing duplicate detections. In the tracking phase, the center-ness score is multiplied by the category score as the ranking reference for the nms algorithm.

4.2.2 CIoU loss

The traditional IoU loss only considers the overlap between two bounding boxes without accounting for their positional and size relationships. The CIoU loss captures the positional and size matching between predicted and ground truth boxes more accurately by considering the distance between their centers and the cross ratio of their widths and heights. The traditional IoU loss might exhibit imbalance between small and large objects, as even a slight deviation can lead to low IoU for small objects. The CIoU loss mitigates this imbalance by introducing size-normalized factors, making it more adaptable to objects of various sizes. The CIoU loss is more sensitive to the positional accuracy of predicted boxes, which makes it more likely for the model to predict object box locations more precisely during optimization. By integrating factors like position, size, and overlap, the CIoU loss provides a more comprehensive measure of bounding box quality during computation, aiding in better training guidance. We optimized the loss function part of the regression branch accordingly. We replaced the IoU loss of the original model with the CIoU [41] loss.

IoU formula:

$$\text{IoU} = \frac{b \cap b^{\text{gt}}}{b \cup b^{\text{gt}}} \tag{2}$$

b is the prediction box; b^{gt} is the ground-truth box.

CIoU formula:

$$\text{CIoU} = \text{IoU} - \left(\frac{\rho^2(b, b^{\text{gt}})}{c^2} + \alpha v \right) \tag{3}$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{w}{h} \right)^2 \tag{4}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{5}$$

$\rho^2(b, b^{\text{gt}})$ is the Euclidean distance between the center points of the predicted box and the ground-truth box; c is the diagonal distance of the smallest closure area that can contain both the predicted box and the ground-truth box; w^{gt} is the width of the ground-truth box; h^{gt} is the height of the ground-truth box; w is the height of the prediction box; h is the width of the prediction box.

The regression loss should consider three geometric parameters: overlap area, center point distance, and aspect ratio. The three items of CIoU correspond to the calculations of IoU, center point distance and aspect ratio. In this way, the predicted box will be more in line with the ground-truth box.

4.2.3 Classification loss

The output of the classification branch is a feature map of size 17*17. The feature map represents the probability that the pixel belongs to the target workpiece. The classification loss is obtained by calculating the cross entropy between the score of the classification branch and the classification label.

5 Experimental

When conducting a comparative analysis between real and virtual datasets, several sources of uncertainty need to be considered:

1. *Data Collection Process* Real datasets are collected through actual observations or measurements, while virtual datasets are generated using simulated or synthesized data. Uncertainty in real datasets arises from factors such as measurement errors, sensor limitations, or biases introduced during the collection process. Conversely, uncertainty in virtual datasets stems from assumptions, simplifications, or inaccuracies in the simulation model.
2. *Model Fidelity* It is commonly assumed that real datasets represent the true model or real-world phenomena. However, the fidelity of the model itself can introduce uncertainty due to measurement technique limitations or inherent variability in the observed phenomena. In contrast, the fidelity of the virtual dataset's model is based on the assumptions and models used in the simulation, introducing uncertainty associated with the accuracy of those assumptions.
3. *Generalization Capability* Real datasets often exhibit greater diversity and complexity, capturing a wider range of real-world scenarios and variations. Virtual datasets, on the other hand, are typically generated based on specific assumptions and scenarios, which may not fully capture the complexity or variability of the real world. This difference in generalization capabilities can lead to uncertainties regarding the applicability and transferability of results from virtual datasets to real-world scenarios.
4. *Model Validity* Due to differences in data characteristics and uncertainties, the performance of models trained or evaluated on real and virtual datasets can vary. Model validation becomes crucial in assessing the reliability and generalizability of results. Uncertainties related to model selection, hyperparameter tuning, and model performance evaluation should be considered in the analysis.

It is important to consider these uncertainties when performing a comparative analysis between real and virtual datasets.

5.1 Implementation

The model parameters of the Convnext backbone are slightly larger than those of the Alexnet backbone, and the FLOPs are higher. However, on the premise of ensuring real-time performance, the Convnext network can achieve higher tracking accuracy. The tracker with AlexNet backbone runs at 84 FPS, while the one with ConvNeXt backbone runs at 61.1 FPS. The unit M (10 to the power of 6) refers to the quantity of parameters, indicating how many parameters are present in a model. The parameter of TransT is 10.7 M, which is slightly less than that of Alexnet. FLOPs is 86.96 G, which is 9 times of that of Convnext model, and FPS is 38. All evaluated on an NVIDIA RTX 3070 GPU. Table 3 shows the network characteristics.

For real datasets and virtual datasets, we divided them using different partitioning methods:

1. We divide the real data set into training set and test set, and only use the real data set for model training and evaluation. The Alexnet version trains 140 epochs, and the Convnext version trains 240 epochs.
2. We divide the real data set and virtual data set into training set and test set respectively, use only virtual data set for model training, and use real data set to evaluate the model. The Alexnet version trains 140 epochs, and the Convnext version trains 240 epochs.
3. We add a small amount of real data set data to the virtual data set, and use the mixed data set to train two versions of the model. Similarly, Alexnet trains 140 epochs, and Convnext trains 240 epochs.

We adopt the warm-up training method, which has achieved good results in YOLOv3 [42]. It is a dynamic learning rate adjustment strategy. The initial use of a smaller learning rate helps to slow down the early overfitting of the model to the mini-batch in the initial stage and keep the distribution stable. It also helps to keep the stability of the deep layers of the model. The learning rate adjustment curve is shown in Fig. 10.

We first train the model with a smaller learning rate, at this stage the learning rate is dynamically increased from 0.001 to 0.005, then use a cosine annealing learning rate schedule for the rest epochs until the learning rate is reduced to 0.0005.

Figure 11 shows the training results of three different training methods and three different trackers. The tracker trained on the real data set has a large deviation when the background and workpiece color are not much different. The trackers trained on virtual dataset and mixed dataset achieve good tracking results. With the addition of distractors, the tracker trained on the real dataset is greatly disturbed, and

Table 3 Network characteristics

Tracker	Parameters [M]	FLOPs [G]	Input size	FPS
SiamFC++ (Alexnet)	11.0	6.9	255*255	84
SiamFC++ (ConvNeXt)	12.6	9.59	255*255	61.1
TransT	10.7	86.96	255*255	38.3

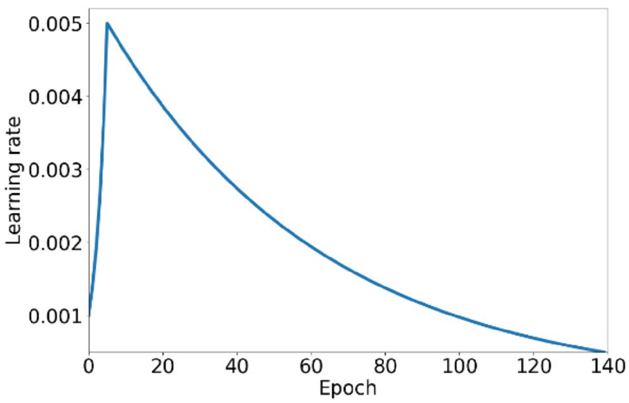

Fig. 10 The learning rate adjustment curve

the localization has a large deviation. Trackers trained with virtual and mixed datasets perform better. TransT model is more seriously affected by background and interferences.

We evaluate trackers using accuracy, robustness, and lost number. “lost number” is a metric used to measure the number of times a tracking algorithm loses the target throughout the entire video sequence. The objective of a tracking algorithm is to consistently and accurately track the target throughout its motion. If the algorithm loses the target at any point in time, the “lost number” increases. Therefore, a lower “lost number” value indicates better performance of the algorithm in target tracking, as it can maintain continuous tracking of the target and avoid losses. These evaluation metrics refer to the evaluation method in VOT [43]. Table 4 shows the tracking effect of trackers under the real testing set.

Trained on the same dataset, the Convnext version tracker outperforms the Alexnet version tracker both in accuracy and robustness. The ConvNeXt backbone shows better feature extraction. In terms of datasets, the effect of model training with manually-labeled real-world small sample datasets is unsatisfactory. The Alexnet version tracker misses the target 15 times, and the ConvNeXt version tracker misses the target 2 times. Using the virtual dataset for training achieves good results, and the accuracy and robustness of the tracker are significantly

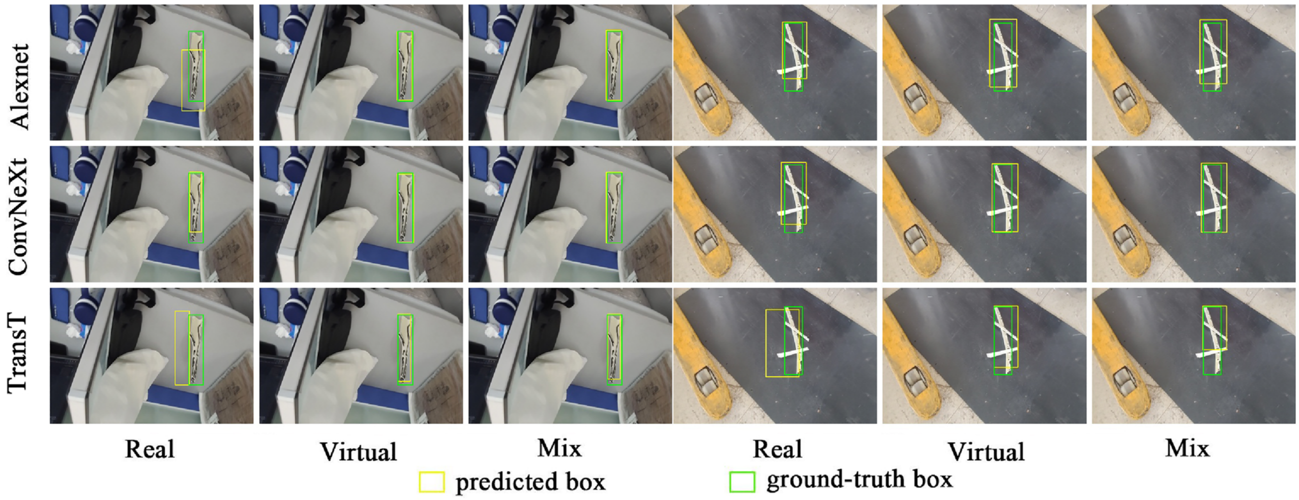

Fig. 11 Comparing the tracking performance of trackers

Table 4 Trackers evaluation

Tracker	Dataset	Accuracy	Robustness	Lost number
SiamFC++ (Alexnet)	Real	0.801	0.194	15
SiamFC++ (Alexnet)	Virtual	0.861	0.013	1
SiamFC++ (Alexnet)	Mix	0.881	0	0
SiamFC++ (ConvNeXt)	Real	0.805	0.026	2
SiamFC++ (ConvNeXt)	Virtual	0.879	0	0
SiamFC++ (ConvNeXt)	Mix	0.891	0	0
TransT	Real	0.786	0.543	41
TransT	Virtual	0.869	0.116	9
TransT	Mix	0.874	0.039	3

improved. The accuracies of the two versions of the tracker increased by 6% and 7.4%, respectively, and the lost number decreased to 1 and 0, respectively. Adding a small amount of real datasets to the virtual datasets for model training can further improve tracker performance. This improves the accuracy by 8% for the Alexnet version of the tracker and 8.6% for the ConvNeXt version of the tracker. TransT trained with real datasets has an accuracy of 0.786 and a robustness of 0.543. The accuracy and robustness of TransT trained using the virtual dataset reached 0.869 and 0.116. TransT using mixed datasets works best, achieving accuracy and robustness of 0.874 and 0.039. The overall effect of TransT is weaker than that of SiamFC++. We believe that the main reason is that the number of samples is not enough, and it is difficult for TransT to learn the similarities between the template and the ground truth.

We compare the performance of the trackers using the evaluation metrics employed in the OTB dataset [10]. The OTB dataset initializes the first frame with the location of the object in the ground-truth, and runs the tracking algorithm on the test set to get the average accuracy and success rate. This method is known as one-pass evaluation (OPE).

Figures 12 and 13 shows that training trackers with the virtual dataset produces leading result in overlap success and Location error. It can be seen that the model trained with virtual data has a higher success rate and precision than only a small amount of real data for training. The ConvNeXt backbone also has a certain improvement in model performance compared to the Alexnet backbone. The overall effect of TransT is weaker than that of the SiamFC++ model. Under the condition of using mixed data set training, SiamFC++ of convnext backbone improves 1.7% in overlap and 4.2% in precision compared to TransT.

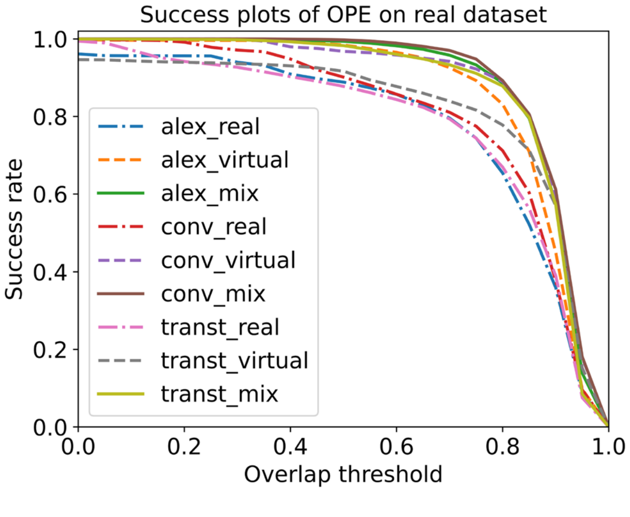

Fig. 12 Success plots of OPE on real dataset

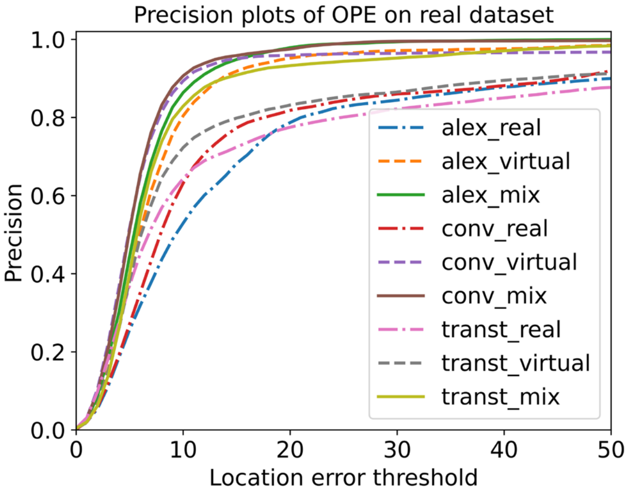

Fig. 13 Precision plots of OPE on real dataset

5.2 Comparative experiments

We compared our method with other mainstream object tracking methods, including TransT, KCF, SiamFC, and SiamRPN. We employed the pre-trained models of these methods and designed a series of comparative experiments to systematically evaluate the performance of different object tracking algorithms. Our testing and evaluation were based on real dataset. By comparing the performance of different algorithms in terms of accuracy, real-time applicability, and robustness, we were able to identify their respective strengths and weaknesses in various scenarios.

Figures 14 and 15 depict the accuracy and robustness of different models under the spray painting dataset. The experimental results suggest that, in comparison to other models, our model produces leading results in terms of overlap success. When compared with TransT, KCF, SiamFC, and SiamRPN models, our model demonstrates a significant improvement in both accuracy and robustness. This underscores the fact that model fine-tuning with industrial minor datasets can yield superior results in industrial scenarios.We tested several competitive trackers, with SiamFC achieving 83.7 FPS, SiamRPN at 37.5 FPS, TransT at 38.3 FPS, and ours at 61.1 FPS. Our approach achieves competitive efficiency with higher accuracy.

5.3 Ablation study

This paper delves into the function of different critical components through a series of object tracking ablation studies, conducting a profound analysis and discussion. We chose the

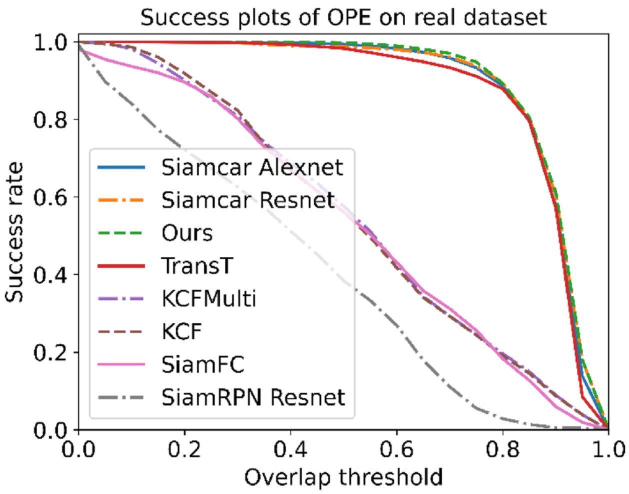

Fig. 14 Success plots of trackers on real dataset

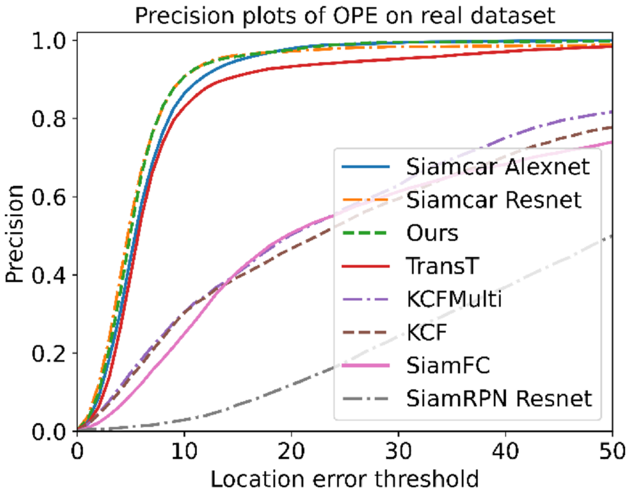

Fig. 15 Precision plots of trackers on real dataset

feature extractor and the loss function parts and performed experiments using real dataset. In the analysis of the ablation study results, we focused on accuracy and robustness performance metrics. The experimental results are as shown in the Table 5.

AlexNet did not use padding to expand the dimensions of the original features, resulting in strict translational invariance. SiamFC++ employs AlexNet as the backbone network. We consider the appropriate target region selection strategy, coupled with the more powerful feature extraction network ConvNeXt, can compensate for issues arising from the inclusion of padding. Using ConvNeXt, a stronger feature extraction network, in place of AlexNet, we observed a 1% increase in accuracy. This indicates that a network with stronger feature extraction capabilities can learn from the biases introduced by the addition of padding. By successively eliminating each component and comparing the experimental results, we arrived at the following conclusions:

Table 5 Ablation study

Method	Accuracy	Robustness	Lost number
SiamFC++	0.866	0	0
SiamFC++ + CIoU	0.881	0	0
SiamFC++ + ConvNeXt	0.884	0	0
SiamFC++ + CIoU + ConvNeXt	0.891	0	0

1. The feature extractor has a significant impact on object tracking performance. In the ablation studies, we found that using the ConvNeXt feature extractor, which is more discriminative and robust, could notably enhance the accuracy and robustness of object tracking, improving the accuracy by 1.8% compared to using the AlexNet backbone network.
2. Different loss functions have a noticeable influence on tracking results. According to our experimental results, using CIoU as the loss function for the position regression branch instead of IoU could boost the tracker's accuracy from 86.6 to 88.1%.

5.4 Application effect

By employing this method for workpiece location tracking, the robot spray paints the workpiece accurately, following a regular and smooth path, resulting in a finely painted surface as illustrated in the Fig. 16.

Quality assessment was conducted on the experimental samples, with observations and comparisons made on the spray paint color and surface. According to the experimental

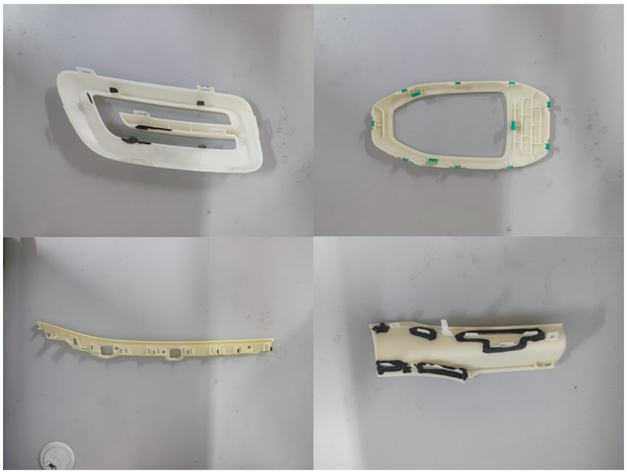

Fig. 16 Spray painting effects on four types of workpieces

data and analysis results, we discovered that the spray painting robot was able to achieve high uniformity in coating thickness and spray coverage. However, in certain unique circumstances, such as curved surface painting and complex-shaped objects, unevenness in the coating might occur.

In most instances, the spray painting robot was able to deliver excellent painting results. We will continue to optimize the robot in our subsequent research, which will include refining painting parameters, enhancing control precision in spray painting, selecting suitable spray guns and nozzles, adjusting nozzle pressure, etc.

6 Conclusion

We believe that by establishing the 3D model of the workpiece and the factory environment in a virtual environment, the working situation in the real environment can be well reproduced. The closer the virtual scene is to the real scene, the closer the feature distribution of the virtual dataset is to the real dataset. This method of generating datasets from virtual scenes can well solve the problem of insufficient data in industrial datasets, so that deep learning methods can be better applied in industrial production. Most of the workpiece model are modeled by the mechanical designer in the design stage, which can be easily obtained. Researchers can also create 3D models of workpieces through 3D scanning technology. Interfering objects, lighting changes, occlusions, and pose changes can be easily simulated in the virtual environment. Automatic annotation can save a lot of time. Using virtual datasets can train high-precision and robust deep learning models.

Our following research will focus on building an industrial deep learning system. The user only needs to import the workpiece model and the factory environment, and the tracker of the workpiece can be automatically and quickly trained in the virtual environment. The trained model can be deployed automatically and put into production quickly.

Acknowledgements This is a part research accomplishment of the project "Learning and Verification of Smart Coordinated Operation of Autonomous Agents for Space On-orbit Maintenance", which is supported by the Ministry of Science and Technology of the People's Republic of China (Grant number [2018AAA0103004]). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions KY conceived the experiments, KY and CW conducted the experiments, KY analysed the results. All authors reviewed the manuscript.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests

References

1. Chandran, P., Winberg, S., Zoss, G., Riviere, J., Gross, M., Gotardo, P., Bradley, D.: Rendering with style: combining traditional and neural approaches for high-quality face rendering. ACM Trans. Graph. (2021). https://doi.org/10.1145/3478513.3480509
2. Zhu, J., Zhao, S., Xu, Y., Meng, X., Wang, L., Yan, L.-Q.: Recent advances in glinty appearance rendering. Comput. Vis. Media **8**(4), 535–552 (2022). https://doi.org/10.1007/s41095-022-0280-x
3. Choi, M., Park, J.-H., Zhang, Q., Hong, B.-S., Kim, C.-H.: Deep representation of a normal map for screen-space fluid rendering. Appl. Sci. Basel **11**(19), 1 (2021). https://doi.org/10.3390/app11199065
4. Neuhauser, C., Wang, J., Westermann, R.: Interactive focus plus context rendering for hexahedral mesh inspection. IEEE Trans. Vis. Comput. Graph. **27**(8), 3505–3518 (2021). https://doi.org/10.1109/TVCG.2021.3074607
5. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8122–8131 (2021). https://doi.org/10.1109/CVPR46437.2021.00803
6. Deng, J., Dong, W., Socher, R., Li, L.J., Kai, L., Li, F.-F.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
7. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017). https://doi.org/10.48550/arXiv.1708.07747
8. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do CIFAR-10 Classifiers Generalize to CIFAR-10? (2018). https://doi.org/10.48550/arXiv.1806.00451
9. Liao, Y., Xie, J., Geiger, A.: Kitti-360: a novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Trans. Pattern Anal. Mach. Intell. (2022). https://doi.org/10.1109/TPAMI.2022.3179507
10. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Harshit, Huang, M., Liu, J., Xu, Y., Liao, C., Yuan, L., Ling, H.: Lasot: a high-quality large-scale single object tracking benchmark. Int. J. Comput. Vis. **129**(2), 439–461 (2021). https://doi.org/10.1007/s11263-020-01387-y
11. Huang, L., Zhao, X., Huang, K.: Got-10k: a large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. **43**(5), 1562–1577 (2021). https://doi.org/10.1109/TPAMI.2019.2957464
12. Kang, Y., Yin, H., Berger, C.: Test your self-driving algorithm: an overview of publicly available driving datasets and virtual testing environments. IEEE Trans. Intell. Veh. **4**(2), 171–185 (2019). https://doi.org/10.1109/tiv.2018.2886678
13. Li, D.C., Lin, L.S., Chen, C.C., Yu, W.H.: Using virtual samples to improve learning performance for small datasets with multimodal distributions. Soft. Comput. **23**(22), 11883–11900 (2019). https://doi.org/10.1007/s00500-018-03744-z
14. Tian, Y.L., Li, X., Wang, K.F., Wang, F.Y.: Training and testing object detectors with virtual images. IEEE-CAA J. Autom. Sin. **5**(2), 539–546 (2018). https://doi.org/10.1109/jas.2017.7510841
15. Xue, Z.F., Chen, L., Liu, Z.T., Liu, Y., Mao, W.J.: Virfd: a virtual-realistic fused dataset for rock size analysis in tbm construction. Neural Comput. Appl. **34**(16), 13485–13498 (2022). https://doi.org/10.1007/s00521-022-07179-4

16. Zhou, Y.Z., Feng, T., Shuai, S.H., Li, X.D., Sun, L.Y., Duh, H.B.L.: Edvam: a 3d eye-tracking dataset for visual attention modeling in a virtual museum. Front. Inf. Technol. Electron. Eng. **23**(1), 101–112 (2022). https://doi.org/10.1631/fitee.2000318
17. Li, X., Wang, K.F., Tian, Y.L., Yan, L., Deng, F., Wang, F.Y.: The paralleleye dataset: a large collection of virtual images for traffic vision research. IEEE Trans. Intell. Transp. Syst. **20**(6), 2072–2084 (2019). https://doi.org/10.1109/tits.2018.2857566
18. Oliver, M.M., Alcover, E.A.: Uibvfed: virtual facial expression dataset. PLoS ONE **15**(4), 1–10 (2020). https://doi.org/10.1371/journal.pone.0231266
19. Montulet, R., Briassouli, A.: Densely annotated photorealistic virtual dataset generation for abnormal event detection. In: Proceedings of the International Conference on Pattern Recognition, pp. 5–19. Springer, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-68799-1_1
20. Jeon, H.-G., Im, S., Lee, B.-U., Choi, D.-G., Hebert, M., Kweon, I.S.: Disc: a large-scale virtual dataset for simulating disaster scenarios. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 187–194 (2019). https://doi.org/10.1109/IROS40897.2019.8967839
21. Shen, Q.Y., Huang, T.G., Ding, P.X., He, J.: Training real-time panoramic object detectors with virtual dataset. In: ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1520–1524 (2021). https://doi.org/10.1109/ICASSP39728.2021.9414503
22. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision-ECCV 2014, pp. 740–755 (2014). Springer. https://doi.org/10.1007/978-3-319-10602-1_48
23. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence 34, pp. 12549–12556 (2020) https://doi.org/10.1609/aaai.v34i07.6944
24. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4277–4286 (2019). https://doi.org/10.1109/CVPR.2019.00441
25. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations, pp. 1–22 (2010). https://doi.org/10.48550/arXiv.2010.11929
26. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners (2021). https://doi.org/10.48550/arXiv.2111.06377
27. Kaji, F., Nguyen-Huu, H., Budhwani, A., Narayanan, J.A., Zimny, M., Toyserkani, E.: A deep-learning-based in-situ surface anomaly detection methodology for laser directed energy deposition via powder feeding. J. Manuf. Process. **81**, 624–637 (2022). https://doi.org/10.1016/j.jmapro.2022.06.046
28. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) Computer Vision—ECCV 2016 Workshops, pp. 850–865. Springer (2016). https://doi.org/10.1007/978-3-319-48881-3_56
29. Bo, L., Yan, J., Wei, W., Zheng, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8971–8980 (2018). https://doi.org/10.1109/CVPR.2018.00935
30. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11976–11986 (2022)
31. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9992–10002 (2021). https://doi.org/10.1109/ICCV48922.2021.00986
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
33. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (2017). https://doi.org/10.48550/arXiv.1711.05101
34. Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243
35. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016). https://doi.org/10.1109/CVPR.2016.308
36. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017). https://doi.org/10.48550/arXiv.1704.04861
37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
38. Radosavovic, I., Johnson, J., Xie, S., Lo, W.Y., Dollar, P.: On network design spaces for visual recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1882–1890 (2019). https://doi.org/10.1109/ICCV.2019.00197
39. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollár, P.: Designing network design spaces. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10425–10433 (2020). https://doi.org/10.1109/CVPR42600.2020.01044
40. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional one-stage object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9626–9635 (2019). https://doi.org/10.1109/ICCV.2019.00972
41. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. (07), pp. 12993–13000 (2020). https://doi.org/10.1609/aaai.v34i07.6999
42. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement (2018). https://doi.org/10.48550/arXiv.1804.02767
43. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernández, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. **38**(11), 2137–2155 (2016). https://doi.org/10.1109/TPAMI.2016.2516982

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