第 40 卷第 6 期
2021 年 12 月

红 外 与 毫 米 波 学 报
J. Infrared Millim. Waves

Vol. 40，No. 6
December，2021

文章编号：1001-9014(2021)06-0858-07 **DOI：10. 11972/j. issn. 1001-9014. 2021. 06. 021**

Real-time infrared target detection based on center points

MIAO Zhuang[1,2]， ZHANG Yong[1*]， LI Wei-Hua[1,2]

（1. Key Laboratory of Infrared System Detection and Imaging Technology，Shanghai Institute of Technical Physics，Chinese Academy of Sciences，Shanghai 200083，China；
2. School of Electronic，Electrical，and Communication Engineering，University of Chinese Academy of Sciences，Beijing 100049，China）

Abstract：A real-time target detection method based on center points is proposed for infrared imaging systems equipped with CPUs. Following the lightweight design principles，a backbone with low computational cost is first introduced for feature extraction. Correspondingly，an efficient feature fusion module is designed to exploit spatial and contextual information extracted from multi-stages. In addition，an auxiliary background suppression module is proposed to predict foreground regions to enhance the feature representation. Finally，a simple detection head predicts the target center point and its associated properties. Evaluations on the infrared aerial target dataset show that our proposed method achieves 90. 24% mAP at a speed of 21. 69 ms per frame on the CPU. It surpasses the state-of-the-art Tiny-YOLOv3 by 10. 16% mAP with only 21% FLOPs and 11% parameters while also runs 10. 02 ms faster. The results demonstrate its great potential for real-time infrared applications.

Key words：infrared image，target detection，real time，deep learning
PACS: 42. 30. Tz， 07. 05. Pj

基于中心点的实时红外目标检测方法

苗　壮[1,2]，张　湧[1*]，李伟华[1,2]

（1. 中国科学院上海技术物理研究所 红外探测与成像技术重点实验室，上海 200083；
2. 中国科学院大学 电子电气与通信工程学院，北京 100049）

摘要：针对仅配备CPU的红外成像系统，本文提出了一种基于中心点的实时目标检测方法。遵循轻量化的设计原则，首先引入了低计算成本的特征提取网络，并在此基础上设计了相应的特征融合模块以充分利用不同阶段提取的空间和上下文信息。同时为了进一步提高网络的表征能力，提出了一个背景抑制模块以完成对前景区域的特征增强，并最终通过轻量检测网络实现对目标中心点及其相应属性的预测。在红外空中目标数据集上的实验表明，本文所提方法能够在CPU上以21.69 ms每帧的速度达到90.24%的检测精度。与经典的Tiny-YOLOv3相比，在计算量和参数量仅为前者21%和11%的前提下，检测精度提高了10.94%，并且检测速度提高了10.02 ms，证明了方法在实时红外系统中的巨大应用潜力。

关　键　词：红外图像；目标检测；实时性；深度学习
中图分类号：TP391.4　　**文献标识码：**A

Introduction

Target detection is one of the most critical yet challenging tasks in infrared (IR) imaging systems，as it involves a combination of target classification and localization[1-3]. With the tremendous development of deep learning，many modern convolutional neural network (CNN) based detection models have been proposed and have significantly boosted detection accuracy. Despite the state-of-the-art accuracy these models have achieved，their deployment costs are increasingly expensive. Only high-end graphics processing units (GPUs) can ensure their inference efficiency due to the high computational complexity and large parameter size (model size). However，

Received date：2021- 02- 16，**revised date**：2021- 09- 13　　**收稿日期：**2021- 02- 16，**修回日期：**2021- 09- 13
Foundation items：Supported by Shanghai Key Laboratory of Criminal Scene Evidence funded Foundation（2017xcwzk08）
Biography：MIAO Zhuang (1988-)，male，Qingdao，China. Ph. D. The research area involves image processing and pattern recognition
E-mail：akkomz@163. com.
[*] **Corresponding author：**E-mail：zybxy@sina. com

most real IR systems are usually deployed on resource-constrained devices only equipped with central processing units (CPUs). Consequently, research on designing accurate real-time detection models suitable for IR systems is valuable and urgent.

From the perspective of detection methods, current CNN-based detection models can be roughly divided into anchor-based detectors and anchor-free detectors. Anchor-based detectors start with setting a huge number of pre-defined rectangle bounding boxes (anchors) with different ratios and scales on high-level feature maps extracted from images. Taking these anchors as proposal candidates, two-stage detectors such as Faster R-CNN[4] and its variants[5, 6] introduce two modules to detect targets precisely. The first module is a regional proposal network (RPN), which predicts the probabilities that each anchor belongs to a target or not and regresses the coordinate offsets between each anchor and its labeled boundary. After non-maximum suppression (NMS), RPN sends all selected anchors to the second module called R-CNN. R-CNN estimates the category probabilities and refines the boundaries. Compared with two-stage detectors, one-stage anchor-based detectors get rid of RPN and directly predict all anchor categories and regress their boundaries. As the architectures are much simpler, one-stage detectors usually have faster detection speed but lower accuracy due to the extreme class imbalance during training. YOLO series[7-10] is one of the most successful one-stage detectors. Its real-time version Tiny-YOLO has been widely implemented in many applications that require fast detection.

By avoiding the intricate design and heavy computation of anchors, anchor-free detectors based on key points have drawn much attention recently[11-13]. CornerNet[11] proposes to detect a target bounding box as a pair of key points, the top-left corner and the bottom-right corner. It adopts the associative embedding technique to group the corner pairs belonging to the same target. Compared with CornerNet, CenterNet[13] introduces a much simpler architecture that simultaneously predicts the target center and its size. Since it does not rely on complicated post-processing decoding strategies, CenterNet achieves state-of-the-art accuracy while having a fairly fast inference speed.

To alleviate the resource consumption of CNNs, a lot of efficient architectures have been designed, including SqueezeNet[14], MobileNet series[15-16], and ShuffleNet series[17-18], etc. Depth-wise separable convolution and group convolution are two primary forms of convolution that construct these architectures. In addition to efficient architecture design, methods such as network pruning[19-20], and quantization[21-22] can further accelerate the inference speed based on pre-trained networks.

To achieve a better balance between detection accuracy and speed for CPU-only IR systems, we propose a real-time infrared target detection model inspired by both the neatly anchor-free detector CenterNet and the lightweight units introduced by ShuffleNetV2[18]. In this paper, it is named TCPD, a tiny center point detector. TCPD contains four main modules: Feature Extraction Module (FEM), Feature Fusion Module (FFM), Background Suppression Module (BSM), and Target Prediction Module (TPM). FEM extracts feature maps at different levels, and FFM combines all these feature maps to leverage spatial and semantic information. BSM is responsible for enhancing the target region, and TPM predicts the target size and its center point. Due to its low computational cost and anchor-free design, TCPD can be efficiently trained on a single GPU and be easily adapted to different application scenarios (from infrared to visible). Without bells and whistles, evaluations on the self-built infrared dataset have shown that TCPD has a better accuracy-speed tradeoff. Compared with state-of-the-art lightweight detector Tiny-YOLOv3, TCPD obtains gains of 10.16% mAP with only 21% FLOPs and 11% parameters at an inference speed of 21.69 ms per image on CPU, which is 10.02 ms faster.

1 Proposed method

In this section, we present the details of TCPD, including the network design and workflow. Although our model is designed mainly focusing on detection efficiency, its accuracy still reaches a high level. Figure 1 illustrates the overall architecture of TCPD. In TCPD, FEM is lightweight designed to reduce the computation cost, which is usually very heavy in modern detection models. FFM and BSM are introduced to mitigate the accuracy degradation due to lightweight design. Furthermore, TPM is for final detection.

1.1 Feature extraction module

Feature extraction module, commonly called the backbone network, is the heaviest part of a detection

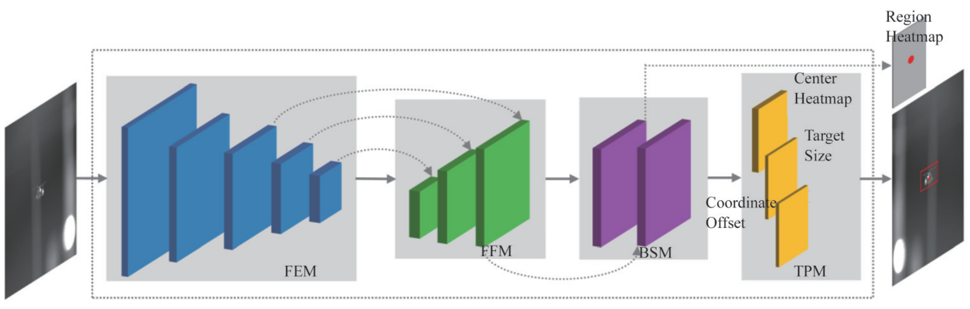

Fig. 1 The overall architecture of TCPD
图 1 TCPD整体框架

model in terms of computation. Therefore, designing a lightweight backbone with strong representation power is fundamental to accurate fast detection. Starting from ShuffleNetV2, we build a new lightweight FEM. It only requires 365 million FLOPs when the input resolution is 384×384 pixels. The detailed structure of FEM is listed in Table 1.

Table 1 Network structure of FEM
表1 FEM网络结构

Stage	Output Size	Output Channels	Layer
Input	384×384	3	Image
Stage1	192×192	24	3×3, Conv, s2
Stage2	96×96	24	3×3, Max Pooling, s2
Stage3	48×48	116	Block1×1
			Block2×4
Stage4	24×24	232	Block1×1
			Block2×8
Stage5	12×12	464	Block1×1
			Block2×4

As listed, FEM consists of five stages in total. After the process of each stage, the feature resolution is halved while the feature channel increases. In "Stage1" and "Stage2", FEM first quickly down-samples the input resolution to 1/4 and expands the feature channel to 24 through a simple 3×3 convolution and a 3×3 max pooling. From "Stage3" to "Stage5", each stage is stacked by several repeated blocks shown in Fig. 2. "Block1" is used to down-sample the feature map and expands the feature channel (116, 232, 464 for Stage 3, 4, and 5, respectively) at the beginning of each stage. Controlling the number of feature channels expanded can trade off network efficiency and accuracy. "Block2" is repeated to enhance the feature representation ability. To minimize memory access costs, the amount of its input and output feature channels keeps the same.

1.2 Feature fusion module

Image features extracted by FEM at different stages

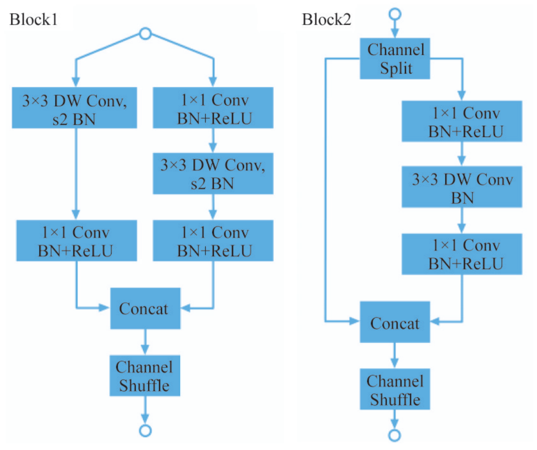

Fig. 2 The structure of blocks in FEM
图2 FEM的单元结构

represent different levels of information. Low-level features in early-stage feature describe more spatial details. By contrast, high-level features in late-stage feature maps capture more contextual information. As a result, localization is more sensitive to larger early-stage feature maps, while classification relies more on smaller late-stage feature maps. To better leverage both spatial and contextual information for detection, a simple feature fusion module is designed. Figure 3 shows its network structure.

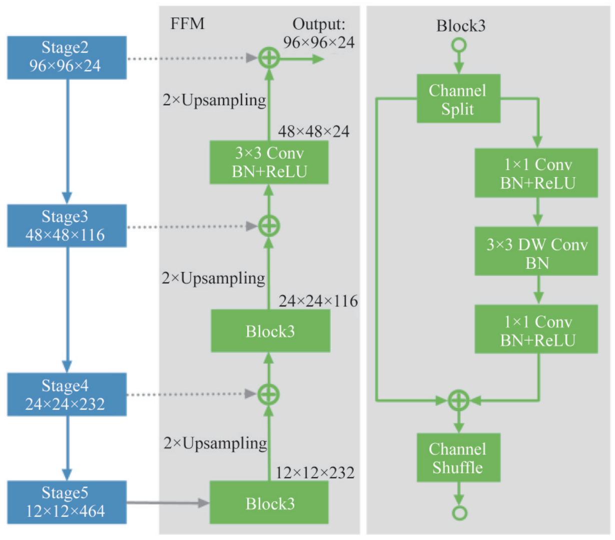

Fig. 3 The network structure of FFM
图3 FFM的网络结构

Starting from "Stage5", FFM combines four stages of FEM through a "bottom-to-up" structure. As the dimension of feature maps (size and channel) varies between two adjacent stages, two steps are needed to complete a single feature fusion. The first step is channel compression. It is through "Block3" for the first two times, while through a 3×3 convolution for the last time. As shown in Fig. 3, "Block3" is similar to "Block2" in FEM. It first divides the input feature maps into two groups equally. Then, one group of feature maps passes through a depth-wise convolution and two element-wise convolutions. Instead of concatenating, these two groups are added together before channel shuffling, which eventually halves the channel. After channel compression, the second step is to upsample feature maps by bilinear interpolation. Compared with deconvolution, bilinear interpolation achieves better performance in practice while reducing extra parameters that need to be learned and stored. The final output size of FFM is 1/4 of the input image, which is large enough for accurate localization of small infrared targets and avoiding collisions of target center points.

1.3 Background suppression module

Generally speaking, a high-performance network is expected to focus on features in the foreground region rather than the background counterparts. To achieve this goal, we design a computation-friendly Background Suppression Module (BSM) to guide the network to learn proper feature distribution explicitly. Figure 4 shows the structure of BSM.

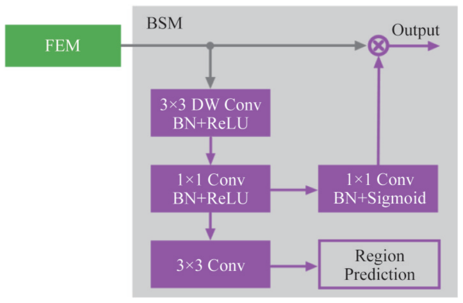

Fig. 4 The network structure of BSM
图4 BSM的网络结构

BSM has two functions: predicting foreground regions and re-weighting feature maps over spatial dimensions. Foreground prediction is the basis of feature re-weighting. During training, BSM first passes the input from FEM to a single-layer detection head through two convolutional layers. The detection head then predicts foreground regions within one heatmap. Ground-truth foreground regions are defined as the combination of all ground-truth targets mapped to the heatmap. The region of each ground-truth target is produced by a 2D-Gaussian kernel, formulated as:

$$\mathrm{K}(x, y) = \exp\left(-\frac{\left(x - x_c\right)^2}{2\sigma_x^2} - \frac{\left(y - y_c\right)^2}{2\sigma_y^2} \right), \quad (1)$$

where (x_c, y_c) is the center point of the mapped ground-truth box, $\sigma_x = \frac{\alpha w}{6}$ and $\sigma_y = \frac{\alpha h}{6}$, which are determined by width and height of the ground-truth target and the hyper-parameter α. α is set to 0.95 by default. All points inside the kernel are regarded as positive samples. If two kernels overlap, the element-wise maximum is taken. Focal loss[23] is applied to train the network:

$$L_r = \frac{-1}{N} \sum_{xy} \begin{cases} \left(1 - \hat{Y}_{xy}\right)^2 \log\left(\hat{Y}_{xy}\right) & \text{if } Y_{xy} = 1 \\ \left(\hat{Y}_{xy}\right)^2 \log\left(1 - \hat{Y}_{xy}\right) & \text{otherwise} \end{cases}, \quad (2)$$

where N is the total number of ground-truth targets, Y_{xy} specifies the ground-truth foreground regions, $\hat{Y}_{xy}$ denotes the estimated probability for the foreground regions.

As the trained BSM has the ability to predict foreground regions, the intermediate layer before the detection head can guide the feature distribution. For computational efficiency, only an element-wise convolution followed by the sigmoid function is used to re-weight the input feature maps over the spatial dimensions.

1.4 Target prediction module

Target prediction module is the last module of TCPD. It is responsible for predicting all information that is needed to localize and classify targets. To match the light-weight design of other modules, a unified structure including only one 3×3 convolutional layer is used in TPM, as shown in Fig. 5.

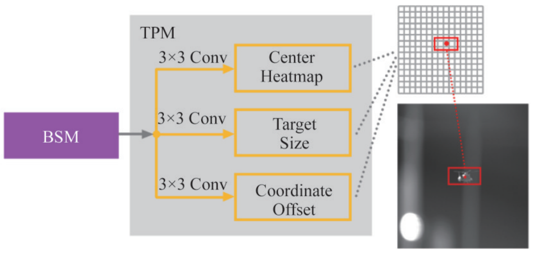

Fig. 5 The network structure of TPM
图5 TPM的网络结构

TPM treats target detection as its center localization and size regression. For center localization, it predicts center confidence scores of different target categories on corresponding center heatmaps. The ground-truth heatmaps are produced by the same Gaussian kernel defined in Eq. 1, enabling negative samples around positive center points to get less penalization than those far away. Compared to the kernel used in CenterNet, our variant kernel is more reasonable. It fits the target shape better with two standard deviations determined by target width and height, respectively. The hyper-parameter α is set to 0.75 by default. The center heatmaps are trained with a variant focal loss[11]:

$$L_k = \frac{-1}{N} \sum_{xyc} \begin{cases} \left(1 - \hat{Y}_{xyc}\right)^2 \log\left(\hat{Y}_{xyc}\right) & \text{if } Y_{xyc} = 1 \\ \left(1 - Y_{xyc}\right)^4 \left(\hat{Y}_{xyc}\right)^2 \log\left(1 - \hat{Y}_{xyc}\right) & \text{otherwise} \end{cases}, \quad (3)$$

where N is the total number of ground-truth center points, $\hat{Y}_{xyc}$ is the center confidence score of class c at location (x, y), Y_{xyc} is its corresponding ground-truth value. Additional to center heatmaps, TPM also predicts coordinate offsets to compensate for the discretization error caused by downsampling. Center locations are adjusted slightly by offsets when remapping from the heatmap to the original image. L1 loss is adopted for training defined as:

$$L_o = \frac{1}{N} \sum_{p} \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right|, \quad (4)$$

where N is also the total number of ground-truth center points, $\hat{O}_{\tilde{p}}$ is the predicted offset, p is the center coordinate on the original image, R is the downsampling factor (default is 4), $\tilde{p}$ is the center coordinate of discretization $\frac{p}{R}$ on the heatmap.

For size regression, TPM directly predicts the target size on the center point with width and height. The target size is also trained with L1 loss:

$$L_s = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{s}_k - s_k \right|, \quad (5)$$

where $\hat{s}_k$ is the predicted k-th target size, and s_k is the ground-truth k-th target size.

Combined with the loss L_r in BSM, the total loss L for training is:

$$L = L_r + L_k + \lambda_s L_s + \lambda_o L_o \quad , \quad (6)$$

where λ_s and λ_o are weights for loss L_s and L_o respectively. They are set to 0. 1 in our experiments unless otherwise specified.

Different from training, a simple post-processing method is introduced to generate the final predictions during inference. Instead of using IoU-based NMS, a 3×3 max-pooling layer is used on the center heatmaps to select the top 100 center points with the highest confidence scores. After adjusting by coordinate offsets, all selected center points and their corresponding target sizes are remapped to the original image. The final results are those with confidence scores above a manual threshold.

2 Experiments

In this section, we first evaluate the performance of TCPD on both the self-built infrared aerial target dataset and the public visible dataset PASCAL VOC. An ablation study is then conducted to evaluate our design furthermore.

2. 1 Dataset and implementation details

In our experiments, an infrared aerial target dataset is built for training and testing. There are 2 758 images with 3 000 labeled infrared targets in the dataset. All images are captured from ground-to-air infrared videos. The labeled targets consist of five categories: bird, helicopter, airliner, trainer, and fighter. The ratio of the training set and test set is 7:3. Results on the public dataset PASCAL VOC are also reported to verify the generalization ability of TCPD. PASCAL VOC dataset has natural images from 20 categories. The VOC 2007 and 2012 trainval sets are combined for training, while the VOC 2007 test set is used for testing.

We implement TCPD with Pytorch. It is trained on a single GPU 1080ti and tested on CPU 9900ks. During training, the input resolution is set to 384×384. Standard data augmentation is applied, including random flipping, random scaling, cropping, and color jittering. Adam is adopted to optimize the total loss. By default, TCPD is trained with a batch size of 32 for 150 epochs. The learning rate starts from 1. 25e-3 and decays by a factor of 0. 1 at the 70th epoch, and 120th epoch.

2. 2 Target detection

Accuracy is one of the most critical metrics for a detection model. A good light-weight model requires accurate classification and localization while keeping efficiency. We first evaluate our model on the infrared dataset. The results are shown in Table 2. The first two rows are classic GPU-based detection models[9, 13], while the last three rows are CPU-based ones[3, 9-10]. The backbone network of CenterNet and YOLOv3 is Res18 and Darknet53. Tiny-YOLOv3 and Tiny-YOLOv4 use the light version of Darknet53 and CSPDarknet53 as their backbones.

As shown in Table 2, TCPD achieves 90. 24% mAP, which surpasses all other light-weight models. For example, it outperforms Tiny-YOLOv3 and FKPD by 10. 16% and 1. 26%. Tiny-YOLOv4 can only achieve Tiny-YOLOv3 level accuracy with a larger input size, which sacrifices its computational efficiency greatly. It is noteworthy that TCPD even surpasses CenterNet by 2. 2%, which benefits from the design of FFM and SAM. As small targets dominate the infrared dataset, TCPD with these modules has significant advantages. Figure 6 visualizes some examples on the infrared dataset. It is clear that the detection accuracy, including classification and localization, achieves a high level.

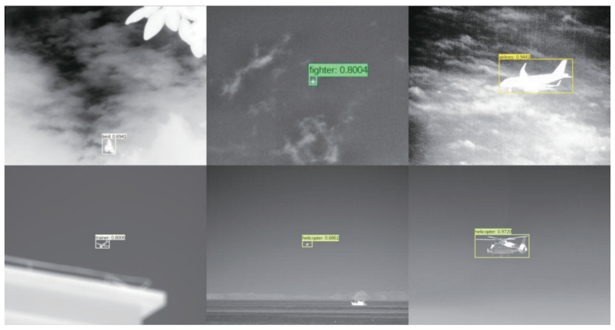

Fig. 6 Examples on infrared dataset
图6 红外数据集图例

In addition to evaluating TCPD on the infrared dataset, the model is also trained on the VOC dataset to verify its generalization ability. The network and all training hyperparameters keep the same as those used on the infrared dataset. The results are reported in Table 3.

As the VOC dataset contains more types of targets and more complex scenarios, it is reasonable that large GPU-based models with more powerful representation

Table 2 Detection results on infrared dataset
表2 红外数据集检测结果

Model	Input Size	mAP/(%)	AP /(%)				
			Bird	Fighter	Airliner	Helicopter	Trainer
CenterNet	384×384	88. 04	76. 73	88. 95	94. 91	90. 77	88. 84
YOLOv3	416×416	93. 02	87. 70	93. 97	95. 97	94. 84	92. 66
Tiny-YOLOv3	416×416	80. 08	66. 58	83. 16	93. 85	84. 92	71. 90
Tiny-YOLOv4	512×512	82. 87	85. 60	91. 06	95. 35	89. 13	53. 23
FKPD	384×384	88. 98	79. 40	90. 84	95. 01	90. 27	89. 39
TCPD	384×384	90. 24	79. 44	90. 69	96. 02	94. 68	90. 35

Table 3　Detection results on VOC dataset
表 3　VOC 数据集检测结果

Model	Input Size	mAP/(%)
CenterNet	384×384	68. 24
YOLOv3	416×416	76. 80
Tiny-YOLOv3	416×416	58. 40
Tiny-YOLOv4	416×416	65. 71
FKPD	384×384	61. 61
TCPD	384×384	66. 76

abilities perform better than TCPD. However, TCPD still achieves 66. 76% mAP, which is close to CenterNet while two times faster. Compared with Tiny-YOLOv3 and FKPD, TCPD surpasses them by 8. 26% and 5. 05%, respectively. As for the latest Tiny-YOLOv4, TCPD still outperforms it by 1. 05%. The results demonstrate that TCPD can adapt target detection better in different applications. Some examples are shown in Fig. 7.

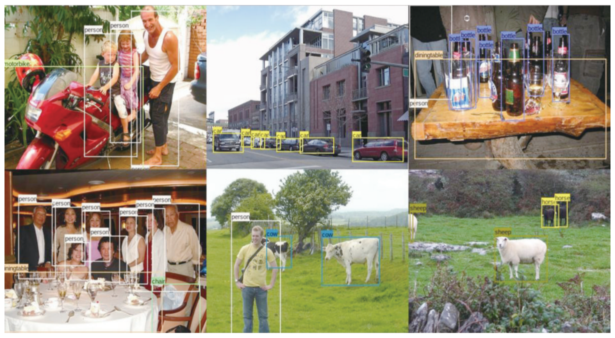

Fig. 7　Examples on the VOC dataset
图 7　VOC 数据集图例

2. 3　Inference speed

As discussed, inference speed plays a significant role in determining whether the model can be applied in most IR systems without GPU acceleration. Computational cost (FLOPs) and model size (Parameters) are two key metrics to evaluate a light-weight model. The computational cost has a direct influence on the inference speed. Lower FLOPs always mean faster detection. While the model size directly affects the storage cost. A model with fewer parameters makes it easier to deploy and has lower FLPOs. Table 4 shows the efficiency test on the infrared dataset. In addition to calculates FLOPs and the number of parameters, we also test the inference speed running on the CPU in practice. For fairness, we input one infrared image once a time on a single thread. The resolution of all input images is fixed to 384×384. The final inference time is the average of 100 images calculated.

With only 0. 49 billion FLOPs and 0. 95 million parameters, TCPD achieves real-time single frame detection on the CPU at a speed of 21. 69 ms. It is 10. 02 ms and 4. 17 ms faster than Tiny-YOLOv3 and FKPD, with merely 21% and 34% FLOPs. The speed of Tiny-YOLOv4 is on par with FKPD, which is 4. 54 ms slower than TCPD. Compared with the other two GPU-based models, the speed advantage of TCPD is more significant. Combined with the discussion in subsection 2. 2, TCPD achieves a better performance, which keeps the balance of accuracy and speed. As a result, it is more suitable for the application in CPU-only IR systems, which requires accurate target detection at a real-time speed.

Table 4　Real-time analysis of TCPD
表 4　TCPD 实时性分析

Model	FLOPs/Bn	Parameters/M	Inference Time/ ms
CenterNet	8. 69	14. 22	48. 90
YOLOv3	27. 93	61. 63	134. 07
Tiny-YOLOv3	2. 34	8. 68	31. 71
Tiny-YOLOv4	2. 91	5. 88	26. 23
FKPD	1. 55	2. 03	25. 86
TCPD	0. 49	0. 95	21. 69

2. 4　Ablation study

In this subsection, we first evaluate the network design of TCPD. Experiments include varying input resolution, compressing the feature channel, and module ablation. The results are shown in Table 5. We choose the default configuration used in 2. 2 as the baseline. All experiments are conducted on the infrared dataset.

Input resolution is an important factor that has a notable influence on the performance of TCPD. Smaller images mean low-resolution feature maps, which leads to the loss of detailed features. Larger images can improve detection accuracy while slows down the inference speed. Line 2 and line 3 in Table 5 verify this conclusion. TCPD-compressed (line 4) is proposed for more sensitive applications to the computational cost and detection speed. It decreases the channels of FEM starting from "Stage3" with {48, 96, 192}, and the inference speed is up to 17. 90 ms with only 0. 19 billion FLOPs and 0. 19 million parameters. The last two lines explore the effectiveness of FEM and BSM. From the experiments, both have positive effects on the detection accuracy while having limited overheads on the inference speed. The mAP decreases 0. 95% and 1. 49% when eliminating FFM and BSM, respectively.

Table 5　Ablation study on the design of model
表 5　模型设计的消融实验

Model	Input Size	mAP/(%)	Inference Time/ ms
TCPD(baseline)	384×384	90. 24	21. 69
TCPD-small	320×320	89. 85	18. 22
TCPD-large	512×512	92. 38	32. 70
TCPD-compressed	384×384	88. 60	17. 90
TCPD w/o FFM	384×384	89. 29	20. 15
TCPD w/o BSM	384×384	88. 75	20. 43

In addition to the network design, we also investigate the influence of the Gaussian kernel defined in Eq. 1. As the kernel is only used to define the penalty weights of negative samples around center points in heat maps during training, it does not affect the inference speed. Table 6 shows the detection results on both datasets with different standard deviations controlled by α.

Ranging from 0. 35 to 0. 95, the variation of α actually affects the scale of negative samples with penalization other than 0 inside the ground-truth box. An appropriate α can improve the detection accuracy. For the infrared dataset with more small targets, the choice is more flexible. While for the VOC dataset with larger targets, the impact is significant. As a result, the choice should be more careful.

Table 6 Ablation study of Gaussian kernel
表6 高斯核的消融实验

α	Dataset (mAP/(%))	
	Infrared	VOC
0. 35	89. 42	64. 81
0. 55	90. 56	66. 00
0. 75	90. 24	66. 76
0. 95	90. 31	66. 24

3 Conclusion

We proposed a new real-time infrared target detection model TCPD based on center points. With the benefit of lightweight design, its computational cost is low, and it can keep the fast inference speed on CPU-only devices. In addition to fundamental feature extraction and target prediction, the Feature Fusion Module and Background Suppression Module are designed to improve feature representation. Evaluations on both infrared and VOC dataset demonstrate the outstanding performance of TCPD as it achieves a better balance between accuracy and speed. In summary, it provides a new choice for real-time detection in IR systems. In the future, we plan to investigate methods such as network pruning to speed up the model while keeping detection accuracy and finally deploy it as a key module in real infrared tracking systems.

References

[1] Wu S C, Zuo Z R. Small target detection in infrared images using deep convolutional neural networks[J]. *J. Infrared Millim. Waves*, 2019, **38**(3): 371-380.

[2] Xie J R, Li F M, Wei H, *et al.* Enhancement of single shot multibox detector for aerial infrared target detection[J]. *Acta Optica Sinica*, 2019, **39**(6): 0615001.

[3] Miao Z, Zhang Y, Chen R M, *et al.* Method for fast detection of infrared targets based on key points[J]. *Acta Optica Sinica*, 2020, **40**(23): 2312006.

[4] Ren S Q, He K, Girshick R, *et al.* Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems (NIPS), 2015: 91-99.

[5] He K, Gkioxari G, Dollár P, *et al.* Mask r-cnn[C]. Proceedings of the IEEE international conference on computer vision (CVPR), IEEE, 2017: 2961-2969.

[6] Cai Z W, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection[C]//Proceedings of the IEEE international conference on computer vision (CVPR), IEEE, 2018: 6154-6162.

[7] Redmon J, Divvala S, Girshick R, *et al.* You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE, 2016: 779-788.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE, 2017: 7263-7271.

[9] Redmon J, Farhadi A. Yolov3: An incremental improvement[EB/OL]. (2018-04-08)[2021-02-10]. https://arxiv. org/abs/1804.02767.

[10] Bochkovskiy A, Wang C Y, Liao H Y. YOLOv4: Optimal Speed and Accuracy of Object Detection[EB/OL]. (2020-04-23)[2021-02-10]. https://arxiv.org/abs/2004.10934.

[11] Law H, Deng J. Cornernet: Detecting objects as paired keypoints [C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 116-131.

[12] Duan K W, Bai S, Xie L X, *et al.* Centernet: Keypoint triplets for object detection[C]//Proceedings of the IEEE International Conference on Computer Vision (CVPR), IEEE, 2019: 6569-6578.

[13] Zhou X Y, Wang D Q, Krähenbühl P. Objects as points[EB/OL]. (2019-05-25)[2021-02-10]. https://arxiv.org/abs/1904.07850.

[14] Iandola F N, Han S, Moskewicz M W, *et al.* SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.**5** MB model size [EB/OL]. (2016-11-4)[2021-02-10]. https://arxiv. org/abs/1602.07360.

[15] Howard J, Zhu M L, Chen B, *et al.* Mobilenets: Efficient convolutional neural networks for mobile vision applications [EB/OL]. (2017-05-17)[2021-02-10]. https://arxiv.org/abs/1704.04861.

[16] Sandler M, Howard J, Zhu M L, *et al.* Mobilenetv2: Inverted residuals and linear bottlenecks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE, 2018: 4510-4520.

[17] Zhang X Y, Zhou X Y, Lin M X, *et al.* Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CPVR), IEEE, 2018: 6848-6856.

[18] Ma N N, Zhang X Y, Zheng H T, *et al.* Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 116-131.

[19] Wen W, Wu C, Wang Y, *et al.* Learning structured sparsity in deep neural networks[C]//Advances in neural information processing systems (NIPS), 2016, 2074-2082.

[20] Li C, Shi C J. Constrained optimization based low-rank approximation of deep neural networks[C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 732-747.

[21] Rastegari M, Ordonez V, Redmon J, *et al.* Xnor-net: Imagenet classification using binary convolutional neural networks[C]//European conference on computer vision (ECCV), 2016: 525-542.

[22] Hubara I, Courbariaux M, Soudry D, *et al.* Quantized neural networks: Training neural networks with low precision weights and activations[J]. *The Journal of Machine Learning Research*, 2017, **18**(1): 6869-6898.

[23] Lin T Y, Goyal P, Girshick R, *et al.* Focal loss for dense object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE, 2017: 2980-2988.