Image Anal Stereol 2023;42:161-170 doi: 10.5566/ias.2929
Original Research Paper

SAMPLE-BALANCED AND IOU-GUIDED ANCHOR-FREE VISUAL TRACKING

JUEYU ZHU[1], YU QIN[2], KAI WANG[2] AND ZHIGAO ZENG✉,[2]

[1]School of Education, Hunan First Normal University, Changsha 410205, Hunan, China, [2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, Hunan, China
e-mail: anny_zjy@hnfnu.edu.cn, qinyu@stu.csust.edu.cn, kaiwang@stu.csust.edu.cn, zengzhigao@stu.csust.edu.cn
(Received May 19, 2023; revised September 21, 2023; accepted September 22, 2023)

ABSTRACT

Siamese network-based visual tracking algorithms have achieved excellent performance in recent years, but challenges such as fast target motion, shape and scale variations have made the tracking extremely difficult. The regression of anchor-free tracking has low computational complexity, strong real-time performance, and is suitable for visual tracking. Based on the anchor-free siamese tracking framework, this paper firstly introduces balance factors and modulation coefficients into the cross-entropy loss function to solve the classification inaccuracy caused by the imbalance between positive and negative samples as well as the imbalance between hard and easy samples during the training process, so that the model focuses more on the positive samples and the hard samples that make the major contribution to the training. Secondly, the intersection over union (IoU) loss function of the regression branch is improved, not only focusing on the IoU between the predicted box and the ground truth box, but also considering the aspect ratios of the two boxes and the minimum bounding box area that accommodate the two, which guides the generation of more accurate regression offsets. The overall loss of classification and regression is iteratively minimized and improves the accuracy and robustness of visual tracking. Experiments on four public datasets, OTB2015, VOT2016, UAV123 and GOT-10k, show that the proposed algorithm achieves the state-of-the-art performance.

Keywords: Cross-entropy, Intersetion over union, Machine vision, Siamese neural network, Target tracking.

INTRODUCTION

Visual tracking is a fundamental and important task in computer vision, and has been widely used in fields such as intelligent monitoring, human-computer interaction, and autonomous driving in recent years Cui *et al.* (2020). Its idea is to establish a model based on video information from sequence images, continuously infer the state of the target based on spatiotemporal correlation, and determine the parameters of the interested target at each frame. However, the appearance, posture, scale, and other variations of the target during its motion. Therefore, achieving robust visual tracking in complex environments still faces serious challenges Li *et al.* (2021).

In recent years, visual tracking methods based on siamese networks have attracted widespread attention due to their accuracy and robustness in tracking effects. The key lies in transforming tracking tasks into similarity matching. This type of method consists of two branches: template branch and search branch. It extracts deep convolution features for similarity calculation without using features for online modeling. Therefore, it is also known as end-to-end tracking methods. The SiamFC Bertinetto *et al.* (2016) (Siamese Fully-Convolutional) network was the first to apply this idea to visual tracking, establishing the overall framework for tracking tasks. SiamRPN Li *et al.* (2018) (Siamese Region Proposal Network) introduces the RPN (Region Proposal Network) structure Ren *et al.* (2017), which improves the cross-correlation operation into two branches: classification and regression. The former is used for the classification of foreground and background, and the latter is used for the regression of bounding boxes, improving the accuracy and robustness of tracking. SiamDW Zhang and Peng (2019) (Deeper and Wider Siamese Networks) and SiamRPN++ Li *et al.* (2019) remove the impact of padding and successfully apply deeper backbone networks to siamese tracking. ResNet He *et al.* (2016) replaces AlexNet Krizhevsky *et al.* (2017), greatly improving the performance of the tracking algorithm. SiamRD Cheng *et al.* (2021) introduces two modules, relationship detection and module optimization, into siamese tracking. The relationship detection section adopts a comparison training strategy to match and learn the same target, and also learns how to distinguish different

targets, improving the discrimination ability of the algorithm. The module optimization section combines classification and regression branches to alleviate the imbalance between the two branches.

In the two branches of the siamese tracking network described above, the classification branch uses a cross-entropy (CE) loss function. The original cross-entropy loss function is classified based on the probability that the samples are calculated as positive samples, ignoring the imbalanced distribution of positive and negative samples as well as the imbalance between hard and easy samples during the training process, which greatly reduces the robustness of the tracking model. In the regression branch, the intersection over union (IoU) loss function is usually used. However, using the intersection over union of the prediction box and the ground truth box area to predict the regression situation of the target cannot reflect the real situation in the regression process. Based on this, this paper proposed a sample-balanced and IoU-guided anchor-free visual tracking algorithm. Firstly, in the classification branch, the cross-entropy loss function is improved, and balance factors and modulation coefficients are introduced to reduce the weight of negative and easy samples, making the model pay more attention to positive and hard samples in training, thereby improving the tracking accuracy and robustness of the model. Secondly, in the regression branch, a new intersection over union loss function is proposed, using the minimum bounding box area between the prediction box and the ground truth box, as well as the difference in the aspect ratios of the two boxes, as penalty terms, to make the distance between the prediction box and the ground truth box closer, thereby improving the tracking accuracy of the model. Finally, the proposed algorithm is tested and compared on four public datasets: OTB2015, VOT2016, UAV123, and GOT-10k, and it reaches the state-of-the-art performance, achieving a real-time tracking speed of 54.40 FPS.

SIAMESE TRACKING ALGORITHM

TRACKING ALGORITHM WITH ANCHOR

SiamFC transforms the tracking problem into a similarity matching problem, and performs cross-correlation operations on the features extracted from two siamese branches to determine the location of the target. SiamRPN introduces the RPN module, which sets the anchor ratio of 5 scales, namely 1/3, 1/2, 1, 2, and 3. Then, K boxes are generated at each location to predict the location and size of the target. SiamRPN uses classification branches to extract the features of the initial frame template, and regression branches are used to extract the features of the search area for the current frame. The features of the two branches are subjected to depthwise cross-correlation operations to predict the classification and regression offset of the target. However, the use of multi-scale anchor will increase computing costs, especially the burden of memory on GPU.

In order to better train the model, DaSiamRPN Zheng *et al.* (2018) (Distractor-aware Siamese Region Proposal Network) classifies samples, defines positive and negative samples by determining whether the overlap between the sample and the ground truth box is greater than a preset threshold, and defines negative samples that are easily removed as easy negative samples, which have a small impact on model training. This corresponds to hard negative samples, which are difficult to remove, but have an important role in improving the robustness of models. SiamRPN++ Li *et al.* (2019) introduces three RPNs to perform classification and regression on the feature of different layers, and combines the results of multiple classification and regression to determine the location and shape of the target in the current frame.

ANCHOR-FREE TRACKING ALGORITHM

In recent years, anchor-free visual tracking methods have become increasingly popular. These methods do not use anchor and can directly obtain the location of the target. The idea is to generate a possible location of the target with each pixel as the center, without using an anchor during the tracking process, which can greatly reduce the parameters and computational complexity. Typical methods include SiamCAR Guo *et al.* (2020), SiamBAN Chen *et al.* (2020), and SiamFC++ Xu *et al.* (2020). Based on the original siamese tracking framework, using the RPN module to perform depthwise cross-correlation operations on extracted features for classification and regression is a new trend in visual tracking tasks, including algorithms based on key point detection, such as CornerNet Law and Deng (2018) detecting the upper left and lower right corners of candidate boxes as key points to determine the location of the target. After performing classification and regression, the SiamCAR method adds the distance between the center points of the two boxes as a metric to the loss function, which can more accurately determine the distance between the two objects through the center metric, greatly improving the accuracy and robustness of tracking.

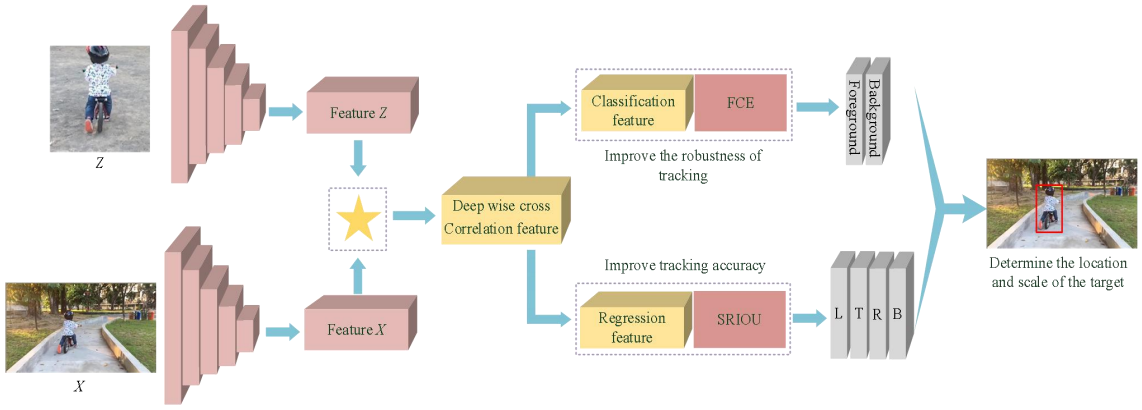


Fig. 1. *Schematic diagram of the sample-balanced and IoU guided anchor-free visual tracking algorithm.*

LOSS FUNCTION IN TRACKING ALGORITHM

In visual tracking, the loss of a model is mainly composed of classification loss and regression loss Li *et al.* (2018). Then, a more robust model is trained through iterative optimization of the loss. The lower the loss, the higher the tracking accuracy, but at the same time, there will be overfitting, manifested as low loss on the training set and good results. However, when applied to the test set, the effect is not satisfactory.

In the classification process, the loss used is 0-1 loss, indicating that the sample is a positive or negative sample, that is, the weight of simple and complex interferences is the same Zheng *et al.* (2018), which leads to discontinuous loss functions and increased difficulty in optimization. Currently, the widely used cross-entropy loss function Giannakas *et al.* (2021) calculates the probability of a positive sample and performs a logarithmic operation. However, these efforts did not take into account the imbalanced distribution of positive and negative samples, as well as the fact that the vast majority of samples are easy samples.

In the regression process, the loss function is mainly caused by the error between the regression value and the ground truth, and is initially calculated by L_1 loss and L_2 loss Oprea *et al.* (2020). This calculation method is relatively simple, and it is not possible to judge whether the regression is accurate in the process of tracking the target. The existing work Rezatofighi *et al.* (2019) calculates regression loss by calculating the intersection over union between the prediction box and the ground truth box. However, this method only considers the degree of overlap between the two boxes. When there is no overlap or there is a significant difference in the shape of the two boxes, this metric loses its original effect.

SAMPLE-BALANCED AND IOU-GUIDED ANCHOR-FREE VISUAL TRACKING ALGORITHM

The visual tracking algorithm is used to estimate the state of the target in subsequent video sequences by learning the target feature of the initial frame. In recent years, the mainstream target tracking method based on siamese networks is to generate multiple candidate boxes at each pixel point by preset anchors, and calculate the similarity to the target one by one to improve the recall of the network. However, setting anchor requires a large amount of prior knowledge, which can bring a heavy computational burden. During model training, the dataset is mainly composed of negative samples with a small proportion of positive samples, which makes it difficult to learn effective information about the target and is not conducive to continuous tracking. In the regression process, the loss function is only calculated based on the intersection over union of the prediction box and the ground truth box area, which cannot truly and comprehensively reflect the regression state of the prediction box.

In order to effectively alleviate the above-mentioned problems, this paper proposed a sample-balanced and IoU guided visual tracking algorithm based on the efficient calculation of the anchor-free tracking framework. The algorithm schematic diagram is shown in Fig. 1. The tracking model consists of a feature extraction section and a classification and regression section. The backbone network of the feature extraction section is ResNet-50. The upper and lower branches share weights to extract the target template features Z and the current frame search area features X, respectively. The depthwise cross-correlation feature is obtained by performing correlation operations between the two, and is used for classification and regression. The focal cross-entropy (FCE) loss function Lin *et al.* (2020) is used in the

classification branch, which can effectively utilize the role of positive and hard samples, making the trained tracking model more robust. In the regression branch, a loss function called the square ratio intersection over union (SRIoU) was first proposed to measure the distance between the prediction box and the ground truth box, as well as the difference in shape. Then, the difference in the shape of the two boxes and the minimum area containing both were added to the regression loss function as a penalty term. Therefore, classification loss and regression loss together constitute the final loss. During training, the minimum loss is iteratively optimized to obtain the precise location and shape of the target.

FEATURE EXTRACTION

In the proposed method, the ResNet-50 network is used for feature extraction because of its deeper convolution layers and stronger feature extraction capabilities. Shallow convolution features usually contain more texture information, which has a good guiding role for target localization. Deep convolutional features contain more semantic information, which can well cope with the interference of similar objects on the target. We have introduced the features of layers 3, 4, and 5 in a network, and conducted cross-correlation operations on these features channel by channel to obtain different responses from different channels.

CLASSIFICATION LOSS FUNCTION CONSTRUCTION

Single target tracking algorithm regards visual tracking as a binary classification problem, and its purpose is to find the target in the surrounding background, so the accuracy of tracking depends largely on the accuracy of classification. We use the extracted features to classify and judge whether the current object is the target. The original cross-entropy function is defined as follows:

$$CE(p,y) = \begin{cases} x = -\log(p) & if(y=1) \\ y = -\log(1-p) & else \end{cases}, \quad (1)$$

where p is the probability that the sample is predicted as a positive sample. The value of label y is 1 and -1, where 1 means positive sample and -1 means negative sample.

The original cross-entropy loss function thinks that all samples have the same importance and contribution to training, ignoring the imbalance between positive and negative samples as well as the imbalance between hard and easy samples in the training process. Zheng et al. Zheng *et al.* (2018) explained that most of the samples in the training process are negative samples, and positive samples only account for a small part. In addition, most samples are relatively easy, and the proportion of hard samples is relatively small.

In order to solve the problem of imbalance between positive and negative samples, a balance factor is introduced into the cross-entropy loss function , and different weights are given to positive and negative samples. The loss function after addition is shown in Equation (2).

$$CE(p,y) = \begin{cases} -\log(p) \times \alpha & if(y=1) \\ -\log(1-p) \times (1-\alpha) & else \end{cases}, \quad (2)$$

where α is the balance factor, and the value is set to the proportion of the positive sample to the total sample. The proportion of cross-entropy loss for positive samples is α, and the proportion of loss for negative samples is $1-\alpha$. In this way, the loss of negative samples will increase, and the model will focus more on learning the features of positive samples.

In addition to the imbalance between positive and negative samples, there is also an imbalance between hard and easy samples during the training process. Easy samples account for the majority of the total samples, but hard samples contribute to the training of network models. Hard samples are often extreme cases, and learning the features of these samples can greatly improve the robustness of the model. Therefore, a modulation coefficient is added for the imbalance between hard and easy samples, as shown in Equation (3).

$$CE(p,y) = \begin{cases} -\log(p) \times (1-p)^{\gamma} & if(y=1) \\ -\log(1-p) \times p^{\gamma} & else \end{cases}, \quad (3)$$

the modulation coefficient is $(1-p)^{\gamma}$ for positive samples and p^{γ} for negative samples. The modulation coefficient also performs different processing for positive and negative samples. Therefore, the higher the predicted probability p of the sample, the easier the sample, and the lower the loss of the sample with the addition of a balance factor. The corresponding loss ratio of hard samples will increase, and the model will also pay more attention to hard samples.

By improving the problem of imbalance between positive and negative samples as well as imbalance between hard and easy samples, a new focal cross-entropy (FCE) loss function L_{FCE} is formed, as shown in Equation (4).

$$L_{FCE} = FCE(p,y) = \begin{cases} -\log(p) \times \alpha \times (1-p)^{\gamma} & if(y=1) \\ -\log(1-p) \times (1-\alpha) \times p^{\gamma} & else \end{cases}. \quad (4)$$

The new cross-entropy loss function not only classifies more accurately, but also considers more hard samples, thereby greatly improving the robustness of the model.

SRIOU REGRESSION LOSS FUNCTION CONSTRUCTION

Classification is used to determine whether the current object is a target, while regression is used to determine the offset between the predicted box and the ground truth box. The anchor-free tracking algorithm will generate a set of offsets for each pixel. As shown in Fig. 2, the pixel will obtain offsets L, T, R, and B in the left, top, right, and bottom directions respectively, to represent the predicted box B_p. The ground truth box B_g is also represented by a set of offsets L^T, R^T, T^T, and B^T. The area of the predicted box S_A and the ground truth box S_B can be calculated from these four offsets, as shown in Equation (5) and (6).

$$S_A = (L+R) \times (T+B), \quad (5)$$

$$S_B = (L^T + R^T) \times (T^T + B^T). \quad (6)$$

The intersection over union loss function is typically used to represent the degree of deviation between the predicted box and ground truth box, and its expression is shown in Equation (7). It only considers the intersection over union of the predicted box and the ground truth box areas, and does not accurately reflect the predicted box shape and distance from the ground truth box. Such metric are neither comprehensive nor effective in reflecting the true state of regression.

$$IoU = \frac{S_A \cap S_B}{S_A \cup S_B}. \quad (7)$$

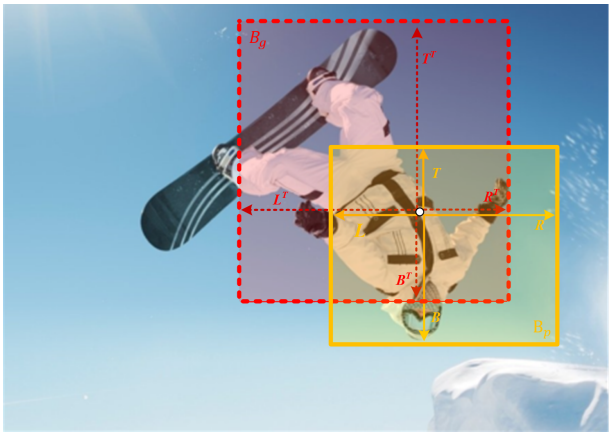


Fig. 2. *Schematic diagram of the intersection over union between the predicted box (B_p) and the ground truth box (B_g).*

In order to make the shapes of the predicted box and the ground truth box closer, the aspect ratio penalty term $\beta \times R$ is added on the basis of the intersection over union loss, as shown in Equation (8) and Equation (9) respectively.

$$R = \frac{4}{\pi^2}(\arctan\frac{W^T}{H^T} - \arctan\frac{W}{H})^2, \quad (8)$$

$$\beta = \frac{R}{1 - IoU + R}, \quad (9)$$

where R is a metric used to compare the aspect ratio of the predicted box and the ground truth box, which can reflect the shape difference between them. The greater the shape difference, the greater the value of the penalty term and the greater the corresponding loss. W, H and W^T, H^T are the width and height of the predicted box and the ground truth box, respectively. β it is the weight of the aspect ratio, which also contains the IoU, which can well reflect the shape between the predicted box and the ground truth box.

Furthermore, we effectively measure the distance between the predicted box and the ground truth box. The predicted box moves to the location of the ground truth box by adding the area of the minimum bounding box S_c. The added minimum bounding box is the smallest rectangular area that encloses the predicted box and the ground truth box, and its expression is shown in Equation (10).

$$S_c = [max(L, L^T) + max(R, R^T)] \times [max(T, T^T) + max(B, B^T)]. \quad (10)$$

Add the area of the minimum bounding box to the intersection over union loss function, as shown in Equation (11).

$$SIoU = IoU - \frac{S_c - (S_A \cup S_B)}{S_c}. \quad (11)$$

The aspect ratio of the predicted box and the ground truth box, as well as the area of the minimum bounding box, form a new SRIoU, where S represents the minimum area that can contain the predicted box and the ground truth box, and R represents the aspect ratio of the predicted box and the ground truth box, as shown in Equation ((12).

$$SRIoU = IoU - \beta \times R - \frac{S_c - (S_A \cup S_B)}{S_c}. \quad (12)$$

The loss construction of the corresponding regression part is completed, as shown in Equation (13).

$$L_{SRIoU} = 1 - IoU + \beta \times R + \frac{S_c - (S_A \cup S_B)}{S_c}. \quad (13)$$

By including two penalty items, the shape of the predicted box is closer to the ground truth box, allowing the predicted box to more accurately reflect the target's location and shape information. Finally, the loss of the proposed tracking model consists of the improved classification loss and regression loss, as shown in Equation (14).

$$L_{all} = L_{FCE} + L_{SRIoU}. \quad (14)$$

EXPERIMENTAL EVALUATION AND ANALYSIS

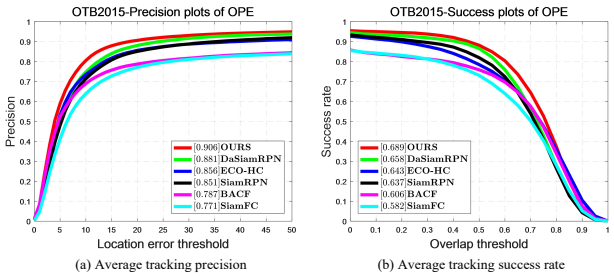
In order to verify the effectiveness of the innovative points in the proposed visual tracking algorithm, rigorous ablation experiments were conducted on two datasets, OTB2015 and VOT2016. The final algorithm is compared with five algorithms on OTB2015, VOT2016, UAV123 and GOT-10k respectively, which shows that the proposed algorithm is progressive. The proposed algorithm is implemented using Python. The deployment platform is Ubuntu 16.0, with 32G of memory, and a GPU of RTX2080Ti. The training set includes ImageNet Russakovsky *et al.* (2015) (VID, DET), YouTube-BB Real *et al.* (2015), and MS COCO Lin *et al.* (2014) datasets. During training and testing, 127 × 127 images are used as templates, and 255 × 255 images are used as search areas. The training batch size is set to 28, optimized using the stochastic gradient descent (SGD) method, and the initial learning rate is set to 0.005. γ is set to 2. In the last 10 batches, the last three layers of the tracking network are trained in combination with the backbone network loading dataset.

ABLATION EXPERIMENT

To verify the effectiveness of the improvement of the cross-entropy loss function and the intersection over union loss function, ablation experiments were conducted on OTB2015 and VOT2016, respectively. The experimental results are shown in Table 1. FCE is a focal cross-entropy loss function that considers positive and negative samples as well as hard and easy samples. After adding FCE, the improvement in robustness and EAO compared to the baseline is more obvious. SRIoU is a loss function that takes into account the aspect ratio of predicted box and the ground truth box, as well as the intersection over union that surrounds the minimum area of both. It has increased in precision and success rate, especially in terms of EAO. The proposed algorithm is based on the benchmark algorithm, introducing FCE and SRIoU, achieving state-of-the-art performance of precision and success rate.

PERFORMANCE COMPARISON EXPERIMENTS

OTB2015 Wu *et al.* (2015) is a widely used test dataset that contains 100 image sequences with varying challenges, including fast motion, background clutter, scale variation, motion blur, occlusion, rotation, and deformation. During testing, the various tracking algorithms are evaluated using success rate and precision in the one pass evaluation (OPE). The proposed algorithm is tested with DaSiamRPN Zheng *et al.* (2018), SiamRPN Li *et al.* (2018), SiamFC Bertinetto *et al.* (2016), ECO-HC Danelljan *et al.* (2017), and BACF Kiani *et al.* (2017) in the OTB2015 dataset, and the results of its success rate and precision are shown in Fig. 3.



(a) Average tracking precision (b) Average tracking success rate

Fig. 3. *Evaluation results of different trackers at OTB2015.*

Table 1. *Ablation experiments.*

Evaluating metrics	OTB2015		VOT2016		
	Precision	**Success**	**Accuracy**	**Robustness**	**EAO**
Baseline	0.676	0.885	0.646	0.186	0.443
Baseline + FCE	0.680	0.891	0.635	0.158	0.470
Baseline + SRIoU	0.684	0.893	0.629	0.168	0.468
Baseline + FCE + SRIoU	0.689	0.905	0.623	0.149	0.477

Table 2. *Evaluation results of different tracking algorithms on VOT2016.*

Tracker	ROAM	SPM	SiamFC	SiamRPN	DaSiamRPN	OURS
Accuracy	0.599	0.620	0.530	0.560	0.610	**0.623**
Robustness	0.174	0.210	0.460	-	0.220	**0.149**
EAO	0.441	0.434	0.235	0.344	0.411	**0.477**

Table 3. *Evaluation results of different trackers on GOT-10k.*

Tracker	SiamCAR	SiamRPN++	SPM	SiamFC	ECO-HC	OURS
AO	**0.569**	0.517	0.513	0.374	0.286	0.565
$SR_{0.5}$	0.670	0.616	0.593	0.404	0.276	**0.677**
$SR_{0.75}$	0.415	0.325	0.359	0.144	0.096	**0.420**
FPS	52.27	49.83	**72.30**	25.81	44.55	54.40

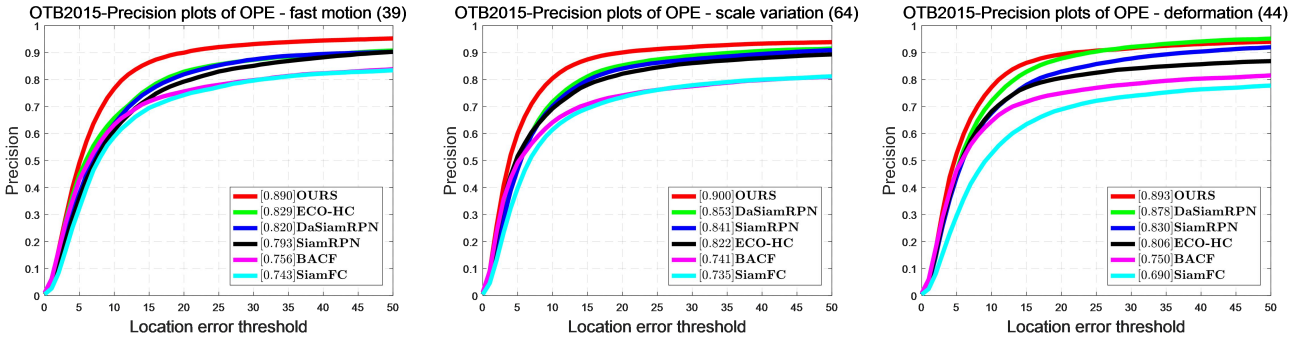
Fig. 3 shows that the proposed algorithm achieved good results in both precision and success rate, with values of 0.906 and 0.689, respectively. DaSiamRPN uses data augmentation to alleviate the problem of sample imbalance to some extent, but there is still a significant difference between the proposed algorithm and the overall experimental effect.

Furthermore, the proposed tracking algorithm improves the classification function and does not require any additional data augmentation. Fig. 4 and Fig. 5 compare the precision and success rate of different tracking algorithms in the subdivision properties of the OTB2015 dataset. We can see three challenges in fast motion, scale variation, and deformation. The proposed algorithm produces good results, demonstrating that it can adapt to fast motion, scale variation, and deformation of tracking targets.

VOT2016 Hadfield *et al.* (2016) is a popular evaluation dataset in the field of single target tracking in recent years, containing 60 image sequences with varying challenge factors and more accurate labeling for target location and size. Accuracy, robustness, and expected average overlap (EAO) are the general dataset evaluation metrics used in the VOT2016 evaluation system Yang *et al.* (2020). The proposed algorithm is compared to popular tracking methods in recent years, with experimental results shown in Table 2. The proposed tracking algorithm not only has the highest accuracy and EAO, but also the best robustness, demonstrating that the proposed algorithm is robust and can handle a variety of challenges Wang *et al.* (2019).

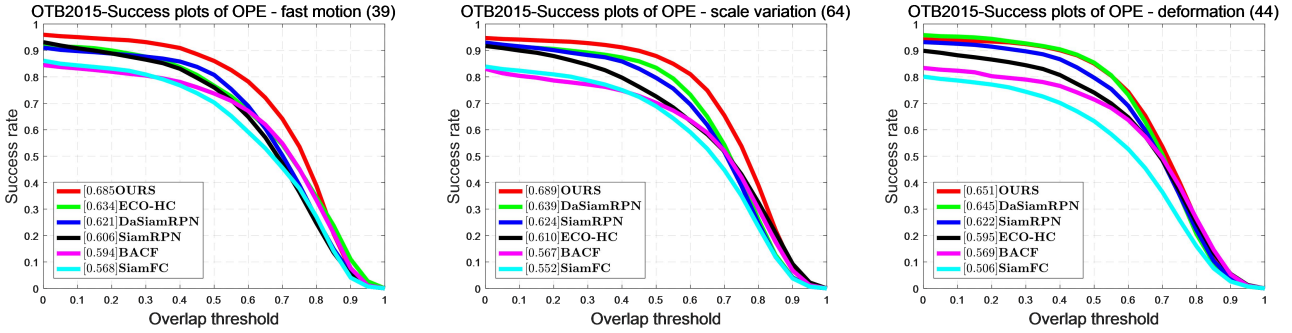
UAV123 Mueller *et al.* (2016) is an aerial photography test dataset that contains 123 image sequences obtained from low-altitude aerial photography, and the average length of each sequence is 915 frames, and all the sequences are marked with rectangular boxes, mainly including challenges such as fast motion speed, large scale variation, long video, and the target beyond the field of vision, which brings great challenges to the tracking task, so the difficulty of this dataset is high. UAV123 uses the same evaluation metrics as the OTB2015 dataset, and both use precision and success rate to measure the performance of the tracking algorithm. In this dataset, the proposed algorithm is compared to SiamCAR Guo *et al.* (2020), DaSiamRPN Zheng *et al.* (2018), SiamRPN Li *et al.* (2018), SRDCF Danelljan *et al.* (2015), and BACF Kiani *et al.* (2017), and the results are shown in Fig. 6. The proposed algorithm has some advantages and achieved the best precision, but its success rate is 0.3% lower than SiamCAR.

The China Academy of Sciences released GOT-10k Huang *et al.* (2021), a general dataset used for field target tracking. It has over 10,000 videos in over 560 classes, all of which are moving objects



(a) Tracking precision under fast motion (b) Tracking precision under scale variation (c) Tracking precision under deformation

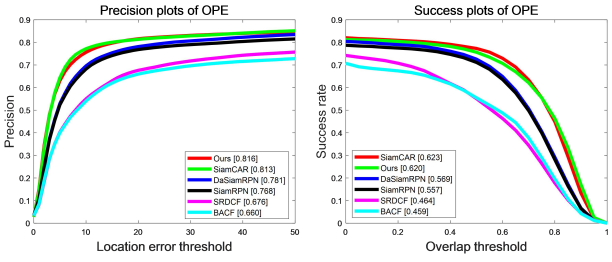
Fig. 4. *Precision of different tracking algorithms on OTB2015 single attributes.*



(a) Tracking success rate under fast motion (b) Tracking success rate under scale variation (c) Tracking success rate under deformation

Fig. 5. *Success rate of different tracking algorithms on OTB2015 single attributes.*

in the real world. This dataset evaluation metrics are average overlap (AO) and success rate (SR). AO denotes the average degree of overlap between the predicted box and the ground truth box, whereas SR denotes the proportion of successfully tracked frames. $SR_{0.5}$ and $SR_{0.75}$, for example, represent the proportion of successfully tracked frames whose overlapping rate exceeds 0.5 and 0.75, respectively. FPS is used to measure the running speed of tracking algorithms. Table 3 shows that the proposed algorithm achieves 0.565 on AO, which is 0.4% lower than SiamCAR, but ranks higher on SR. Furthermore, the proposed tracking algorithm has a speed of 54.40 FPS, indicating that the target can be tracked in real time.



(a) Average tracking precision (b) Average tracking success rate

Fig. 6. *Evaluation results of different trackers on UAV123.*

CONCLUSION

In this paper, the imbalance of positive and negative samples, as well as the imbalance of hard and easy samples, are thoroughly considered in the visual tracking training process, and a weight factor is introduced into the anchor-free tracking frame to reduce the influence of negative samples. The anchor-free tracking regression method requires less computation and has a higher real-time rate, as well as better location precision. Through the modulation coefficient, the model pays more attention to hard samples, which improves the model's robustness. To solve the problem of a single metric and only considering the intersection over union area of the predicted box and the ground truth box in the regression process, the shape difference between the predicted box and the ground truth box, as well as the minimum area surrounding the range of the predicted box and the ground truth box, are added to the regression loss function as penalty terms to train the tracking model to improve the tracking effect. The proposed algorithm was fully tested on four datasets, OTB2015, VOT2016, UAV123, and GOT-10k, reaching a state-of-the-art performance and running at 54.40 FPS. As a result, this can accurately predict the location and size of the target in the current frame in real time.

ACKNOWLEDGMENTS

This project is partially funded by the Hunan Province General Higher Education Teaching Reform Research Project under Grant HNJG-2022-1212. The all authors would like to acknowledge funding from the China University Industry-University-Research Innovation Fund grant number 2021RYA05001.

REFERENCES

Bertinetto L, Valmadre J, Henriques J F, Joao F, Vedaldi A, Torr P HS (2016). Fully-convolutional siamese networks for object tracking. In: Proceedings of the European Conference on Computer Vision, Cham: Springer, 850–65.

Cui Z J, An J S, Zhang Y F, Cui T S (2020). Light-weight siamese attention network object tracking for unmanned arial vehicle. ACTA BOT SIN 40: 1915001.

Cheng S, Zhong B, Li G, Liu X, Tang Z, Li X, Wang J (2021). Learning to filter: siamese relation network for robust tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New York: IEEE Press, 4421–31.

Chen Z, Zhong B, Li G, Zhang S, Ji R (2020). Siamese box adaptive network for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York: IEEE Press, 6668–77.

Danelljan M, Bhat G, Shahbaz K F, Felsberg M (2017). Eco: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA. New York: IEEE Press, 6638–46.

Danelljan M, Hager G, Shahbaz K F, Felsberg M (2015). Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. Cham: Springer, 4310–18.

Guo D, Wang J, Cui Y, Wang Z, Chen S (2020). SiamCAR: siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York: IEEE Press, 6269–77.

Giannakas F, Troussas C, Voyiatzis I, Sgouropoulou C (2021). A deep learning classification framework for early prediction of team-based academic performance. APPL SOFT COMPUT 106: 107355.

Hadfield S J, Bowden R, Lebeda K (2016). The visual object tracking VOT2016 challenge results. LECT NOTES COMPUT SC 9914: 777–823.

He K, Zhang X, Ren S, Sun J (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA. Cham: Springer, 770–78.

Huang L, Zhao X, Huang K (2021). A large high-diversity benchmark for generic object tracking in the wild. IEEE T PATTERN ANAL 43: 1562–77.

Zhang J, Feng W, Yuan T, Wang J, Sangaiah A K (2022a). SCSTCF: Spatial-Channel Selection and Temporal Regularized Correlation Filters for Visual Tracking. APPL SOFT COMPUT 118: 108485.

Zhang J, Sun J, Wang J, Li Z, Chen X (2022b). An object tracking framework with recapture based on correlation filters and Siamese networks. COMPUT ELECTR ENG 98: 107730.

Zhang J, Sun J, Wang J, Yue X-G (2021). Visual object tracking based on residual network and cascaded correlation filters. J AMB INTEL HUM COMP 12: 8427–40.

Zhang J, He Y, Wang S (2023). Learning adaptive sparse spatially-regularized correlation filters for visual tracking. IEEE SIGNAL PROC LET 30: 11–15.

Krizhevsky A, Sutskever I, Hinton G E (2017). Imagenet classification with deep convolutional neural networks. COMMUN ACM 60: 84–90.

Kiani G H, Fagg A, Lucey S (2017). Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. New York: IEEE Press, 1135–43.

Li C, Yang D D, Song P, Guo C, Guo C (2021). Global-aware siamese network for thermal infrared object tracking. ACTA BOT SIN 41: 0615002.

Li B, Yan J, Wu W, Zhu Z, Hu X (2018). High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA. New York: IEEE Press, 8971–80.

Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2019). SiamRPN++: evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA. New York: IEEE Press, 4282–91.

Law H, Deng J (2018). Cornernet: detecting objects as paired keypoints. In: Proceedings of the European

Conference on Computer Vision, Cham: Springer, 734–50.

Lin T Y, Goyal P, Girshick R, He K, Dollár P (2020). Focal loss for dense object detection. IEEE T PATTERN ANAL 42: 318–27.

Lin T Y, Michael M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick C L (2014). Microsoft coco: common objects in context. In Proceedings of the European Conference on Computer Vision, Cham: Springer, 740–55.

Mueller M, Smith N, Ghanem B (2016). A benchmark and simulator for uav tracking. In: Proceedings of the European Conference on Computer Vision, Cham: Springer, 445–61.

Oprea S, Martinez-Gonzalez P, Garcia-Garcia A, Castro-Vargas J A, Orts-Escolano S, Garcia-Rodriguez J, Argyros A (2020). A review on deep learning techniques for video prediction. IEEE T PATTERN ANAL 44: 2806–26.

Ren S, He K, Girshick R, Sun J (2017). Faster r-cnn: towards real-time object detection with region proposal networks. IEEE T PATTERN ANAL 39: 1137–49.

Rezatofighi H, Tsoi N, Gwak J Y, Sadeghian A, Reid I, Savarese S. Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York: IEEE Press, 658–66.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015). Imagenet large scale visual recognition challenge. INT J COMPUT VISION 115: 211–52.

Real E, Shlens J, Mazzocchi S, Pan X, Vanhoucke V (2015). Youtube-boundingboxes: a large high-precision human-annotated data set for object detection in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA. New York: IEEE Press, 5296–305.

Wu Y, Lim J, Yang M H (2015). Object tracking benchmark. IEEE T PATTERN ANAL 37: 1834–48.

Wang G, Luo C, Xiong Z, Zeng W (2019). Spm-tracker: series-parallel matching for real-time visual object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA. New York: IEEE Press, 3643–52.

Xu Y, Wang Z, Li Z, Yuan Y, Yu G (2020). SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence, 34: 12549–56.

Yang T, Xu P, Hu R, Chai H, Chan A B (2020). ROAM: recurrently optimizing tracking model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE Press, 6718–27.

Zhang Z, Peng H (2019). Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA. New York: IEEE Press, 4591–600.

Zheng Z, Wang Q, Li B, Wu W, Yan J, Hu W (2018). Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision, Munich, Germany. Cham: Springer, 101–17.