Active Learning for Deep Visual Tracking

Di Yuan, Member, IEEE, Xiaojun Chang, Senior Member, IEEE, Qiao Liu, Yi Yang, Senior Member, IEEE, Dehua Wang, Minglei Shu, Zhenyu He, Senior Member, IEEE, and Guangming Shi, Fellow, IEEE

Abstract—Convolutional neural networks (CNNs) have been successfully applied to the single target tracking task in recent years. Generally, training a deep CNN model requires numerous labeled training samples, and the number and quality of these samples directly affect the representational capability of the trained model. However, this approach is restrictive in practice, because manually labeling such a large number of training samples is time-consuming and prohibitively expensive. In this article, we propose an active learning method for deep visual tracking, which selects and annotates the unlabeled samples to train the deep CNN model. Under the guidance of active learning, the tracker based on the trained deep CNN model can achieve competitive tracking performance while reducing the labeling cost. More specifically, to ensure the diversity of selected samples, we propose an active learning method based on multiframe collaboration to select those training samples that should be and need to be annotated. Meanwhile, considering the representativeness of these selected samples, we adopt a nearest-neighbor discrimination method based on the average nearest-neighbor distance to screen isolated samples and low-quality samples. Therefore, the training samples’ subset selected based on our method requires only a given budget to maintain the diversity and representativeness of the entire sample set. Furthermore, we adopt a Tversky loss to improve the bounding box estimation of our tracker, which can ensure that the tracker achieves more accurate target states. Extensive experimental results confirm that our active-learning-based tracker (ALT) achieves competitive tracking accuracy and speed compared with state-of-the-art trackers on the seven most challenging evaluation benchmarks. Project website: https://sites.google.com/view/altrack/.

Index Terms—Active learning, limited budget, training samples selection, visual tracking.

I. INTRODUCTION

Single target tracking is a challenging and important task in the computer vision community, with numerous applications including video surveillance, autonomous vehicles, etc. In the tracking task, the core problem is to predict the target state in all subsequent frames when the target ground truth is provided at the beginning [1], [2], [3]. Although deep-learning-based trackers have obtained some notable tracking results in recent years, it remains a difficult problem to train a deep convolutional neural network (CNN) model with a strong representational ability due to the limitations associated with labeled training samples, as well as the time-consuming and expensive nature of annotating unlabeled samples.

Recently, trackers based on deep CNN models have achieved competitive tracking accuracy [4], [5]. In contrast to the traditional trackers [6], [7], [8], these trackers need to have a large number of labeled samples available in advance to train their deep CNN models to track the target of interest [9], [10]. The key to determining the representational ability of the deep CNN model lies in the diversity and comprehensiveness of the labeled training samples (sequences). However, the number of training samples that have labels is very limited, which directly limits the tracking performance of those trackers whose models are trained on these samples. To ensure diversity and comprehensiveness, it is necessary to collect as many training sequences as possible and label the objects in each sequence. Labeling small amounts of training sequences is tractable; however, labeling large amounts of training sequences is time-consuming and potentially expensive [11], [12]. Some methods attempt to use a small amount of labeled data and a large amount of unlabeled data for model training and have achieved acceptable results [13], [14], [15]. Another effective solution is to randomly select a subset of a large unlabeled
sequence set for labeling to train the deep CNN models. In the case of random selection, only the selected subset is sufficiently large to ensure the diversity and representativeness of the entire unlabeled training sample set [16]. The alternative approach to random selection is the use of an active learning method to select samples that can improve the diversity of the sample space [17], [18]. The general hypothesis is that a model trained on the sample subset, selected based on the active learning method, typically has a stronger representation ability than a model trained on the randomly selected subset under the same budget (number of samples) [19], [20]. Therefore, using an active learning method to select a subset of whole training samples for labeling can not only reduce the labeling costs but also improves the representation ability of models trained on this selected and labeled sample subset [21], [22], [23].

Based on the above analysis, we propose to perform an active learning method on the single target tracking task. The active learning method we used to select the labeling subset sequences from numerous unlabeled video sequences to train the deep CNN model of the tracker. The proposed method can ensure that the representational capability of the trained model can be improved as much as possible when the sample sequence budget is fixed. In this way, the tracker based on the trained deep CNN model can achieve competitive tracking performance while reducing the labeling costs. This differs from other tasks (such as image classification [22], [24], [25] and image segmentation [26], [27], [28]) in which active learning is used to select image samples, and we try to apply the active learning method for video sequences’ selection for the tracking task. These existing active learning methods are only based on selecting a single frame image (SAL) information in the sequence, and labeling the training sample sequences will affect the representational ability of the selected sample sequences due to background interference. Considering the time consistency of the moving target in each video, we present the active learning sequence selection method in a multiframe collaboration way. This approach can eliminate background interference through the temporal relationship between multiple frames in each video sequence, thereby ensuring the diversity of the selected sample sequences. Meanwhile, considering the representativeness of sample sequences, we adopt a nearest-neighbor discrimination method based on the threshold used to screen isolated sequences. Therefore, our proposed active learning method can ensure the diversity and representativeness of the selected training sample sequences using a limited budget.

In addition to the training samples affecting the tracking performance, another problem that directly affects the tracking performance is the accuracy of bounding box estimation. Current tracking algorithms mainly use the following methods to estimate the bounding box of the target: using multiscale factors to estimate the target state [29], [30], [31], or taking some proposals and using IoU scores to determine target states [16], [32]. The former mostly use certain fixed scale factors, which cannot accurately estimate the target boundary. The latter does not distinguish between the importance of the target and the background, which leads to insufficient attention being paid to the tracking target area. To mitigate the negative impact of this problem on tracking performance, we adopt the Tversky loss function [33] to improve the tracker’s bounding box estimation strategy, which enables it to obtain a more accurate target state. Compared with other loss functions, the adopted Tversky loss function can ensure that the tracker pays more attention to the target than the background. As shown in Fig. 1, benefiting from the training samples selected by the active learning method and the target boundary box estimation improved by the Tversky loss, our active learning-based tracker (ALT) tracker achieves more accurate target bounding boxes than the ROAM [34], Ocean [35], DiMP50 [32], and SiamBAN [36] trackers.

The main contributions of our research can be summarized as follows.

1) We formulate a novel active learning method to guide the training of the tracker, which can greatly reduce the cost of labeling training samples while ensuring acceptable tracking performance.

2) To select the most diverse and representative training samples under a given budget, we present a multiframe cooperation strategy for training sample selection to ensure the diversity of these selected samples. We also adopt a nearest-neighbor discrimination method based on the average nearest-neighbor distance to screen isolated samples, and this will ensure the representativeness of selected training samples.

3) In addition, we adopt a Tversky loss to improve the bounding box estimation strategy that can enable our ALT tracker to obtain a more accurate target state.

Extensive experimentation shows that our ALT tracker achieves more competitive results than other state-of-the-art trackers on some challenging tracking test datasets: OTB100 [1], UAV123 [2], TrackingNet [11], LaSOT [37], GOT10k [12], VOT2019 [38], and VOT2020 [39].

The remainder of this article is organized as follows. Some related works are first presented in Section II, after which the active-learning-based training samples selection method is proposed to retain the diversity and representativeness of training samples in Section III. Next, the experimental results are outlined in Section IV to verify the effectiveness of our proposed active learning method for the tracking task. Finally, a brief conclusion of this article is provided in Section V.

II. RELATED WORKS
A. Deep-Learning-Based Tracking Methods

Recently, deep-learning-based trackers have attracted increasing attention. Siamese-network-based trackers treat the tracking task as a cross correlation problem [40], [41]. The SiamFC [9] tracker incorporates a fully convolutional Siamese network in an end-to-end manner, which demonstrates the powerful representation ability of the offline trained feature extraction network. The HASN [42] tracker introduces an attention mechanism into the Siamese tracking framework and further introduces a hierarchical attention Siamese network for the target tracking task. Currently, Siamese-network-based trackers enhance their tracking accuracy by adding a region proposal network [43], [44], [45] or some other kind of
B. Active Learning for Training Samples’ Selection

Active learning methods play an important role in natural language processing, computer vision, engineering systems, and other fields [20], [21], [51], [52]. The most important role of active learning methods is that of querying the next sample that should be labeled. There are some existing query strategies in active learning methods: informativeness [24], representativeness [53], hybrid [54], and performance-based [55]. Through the use of these query strategies, active learning methods have been applied to text classification [56], image classification [22], [24], [25], object detection [57], [58], etc. In [24], the BALD method integrates Bayesian deep learning into an active learning framework for high-dimensional data. Yoo and Kweon [25] present a task-agnostic active learning method capable of training the network from a single loss prediction block. Other methods attempt to reduce the labeling cost with a focus on sparse subset approximation [23] or core-set selection [53]. Kim et al. [59] propose a task-aware variational adversarial active learning method for the semantic segmentation tasks. Ren et al. [20] made a comprehensive review of the development of deep active learning from the perspective of the application. Aiming at the uncertainty from measurement error and inherent input noise in the training samples, Yue et al. [60] proposed a variance-based weighted active learning algorithm and a D-optimal weighted active learning algorithm to solve these problems.

Separate from the task of choosing image samples based on active learning described above, our main task is to select video sequence samples based on active learning. First, most existing active learning methods use independent image samples to sort the training samples. In contrast, we use multiframe cooperation to sort the training sequence samples according to the internal time correlation of the sequences, which is well-suited for the tracking task. Second, most existing active learning methods indiscriminately use selected samples for model training. However, after considering the representativeness of these selected sequence samples, we use the nearest-neighbor verification method to reorder the selected sequences, which enables us to effectively exclude isolated and abnormal training samples.

C. Bounding Box Estimation in Tracking Task

Accurate target bounding box estimation is a key factor in ensuring the tracking performance of the tracker. There are two strategies generally adopted by most tracking methods to estimate their target boundary box. The first one is to use multiple fixed scale factors to determine the appropriate boundary box after the target center location is determined [29], [30], [61]. As a representative example, in the C-COT [29] tracker, several fixed scale factors are used to find the most suitable target bounding box. The other one is to select the most appropriate proposal for determining the target boundary box after being given certain proposals [16], [32], [43], [50]. The ATOM [16] tracker determines the appropriate target bounding box by calculating the intersection-over-union (IoU) score of certain proposals and the target in the reference frame. The former primarily uses the fixed scale factor, which is unable
to accurately estimate the target boundary frame [30], [61]. The latter does not distinguish the importance of intersection and union between the estimated and real bounding boxes, which leads to insufficient attention being paid to the tracking target area [16], [43].

To mitigate the negative impact of this problem on tracking performance, we adopt the Tversky loss function [33] to improve the tracker’s bounding box estimation strategy, which enables it to obtain a more accurate target state. Compared with other loss functions, using the Tversky loss function can make the tracker pay more attention to the overlap between the predicted area and the ground truth, which can in turn ensure that the tracker’s prediction is more in line with the actual target state.

III. ACTIVE LEARNING FOR OBJECT TRACKING

We propose an active learning method to select training samples for deep CNN model training in the tracking framework for the visual tracking task. First, we present an introduction to the basic deep CNN-model-based tracking framework in Section III-A. We then present the active learning approach designed to select the training sample sequences that train the deep CNN model for object tracking in Section III-B. Furthermore, we adopt the Tversky loss to improve the bounding box estimation of the target in Section III-C.

A. Deep CNN-Model-Based Tracking Framework

We adopt an end-to-end DiMP [32] tracking architecture as the basic deep CNN-model-based tracking framework. The tracking problem in this framework incorporates a target classification component and a bounding box estimation component. The target classification component contains a convolutional block that can extract features of the image patch. Given labeled training samples, the classification model can generate weights of the target classifier. Applying the obtained weights to the features extracted from a new test image patch can enable the target confidence score to be obtained. The target classification loss can be defined as follows:

$$L(f) = \|r(x \ast f, c)\|^2 + \lambda \|f\|^2$$  \hspace{1cm} (1)

where $x$ is the image patch, $f$ is the filter weights, $\ast$ means the convolution, $x \ast f$ denotes the target confidence scores, $c$ is the ground-truth target center, $r(x \ast f, c)$ denotes the residual, and $\lambda$ is a regularization parameter. By introducing the discriminative learning loss, (1) can be rewritten as follows:

$$L_{cl} = \|l(x \ast f', z_c)\|^2$$  \hspace{1cm} (2)

where $z_c$ denotes the regression label being set to a Gaussian function centered as $c$, while $f'$ indicates the weight parameters of the feature extraction network.

The bounding box estimation component adopts an overlap maximization method which is proposed in the ATOM [16] tracker. Given a set of target candidate bounding boxes based on the target confidence score obtained in the target classification component, the bounding box estimation model is trained to find the maximum IoU score between these estimate bounding boxes ($B$) and the target ground-truth bounding box ($B^\text{gt}$). The IoU loss is defined as follows:

$$L_{ibb} = 1 - \text{IoU}$$  \hspace{1cm} (3)

where $L_{ibb}$ is the IoU loss of the bounding box estimation, and $\text{IoU} = (B \cap B^\text{gt})/(B \cup B^\text{gt})$. Each part in the IoU is clearly illustrated in Fig. 2. The full tracking framework of the DiMP [32] tracker is trained by combining the bounding box estimation loss and target classification loss as $L_{tot} = \eta L_{cl} + L_{ibb}$. More details can be found in literature [32].

B. Active Learning for Training Sequences’ Selection

Owing to the limitation of training samples with labels, which results in the tracking performance of trackers using these samples, trained models are unable to reach the ideal state. Although it is easy to obtain a large number of unlabeled samples, labeling them is time-consuming and prohibitively expensive [37], [62]. A compromise method is to randomly select a certain number of these samples for annotation; however, this random selection method cannot guarantee the performance of the trained deep CNN model. Consequently, a method based on active learning to select samples for annotation is proposed in this article. The hypothesis of active learning is to select those samples that can provide more valuable information for model training than other samples. Therefore, identifying an appropriate selection method is the key to selecting more representative samples. Most of the active learning methods are based on using the similarity between image samples to select and label these selected samples. Inspired by PointNet++ [63] using the farthest point sampling (FPS) algorithm to select the input point subset to cover the whole set, we used the FPS method to make an initial selection of samples to subsequently select a representative sample subset. In the simplest possible terms, the FPS algorithm always selects the sample that is furthest away from all the samples in the selected subset and adds it to the subset, which ensures the diversity of samples in the selected subset. Cosine distance is used to measure the distance between two samples in this article. Fig. 3 presents a simple example of training sample sequence selection. From this figure, we can clearly see that training samples selected through active learning can ensure the diversity and representativeness of samples.
as much as possible under the given budget (a.k.a the number of video sequences).

1) First-Frame-Based Selection Method: Given an annotated video sequence samples’ set $A = \{s_1, s_2, \ldots, s_n\}$, we use iterative FPS method to choose a subset of sequence samples $\{s_{i_1}, s_{i_2}, \ldots, s_{i_a}\}$, such that $s_{i_j}$ is the most distant sequence sample (in terms of metric distance) from the subset $\{s_{j_1}, s_{j_2}, \ldots, s_{j_{a-1}}\}$ with regard to the remaining sequence samples. To facilitate the selection of sample sequences, the first frame image $i_{s_i}$ of each sequence $s_i$ is selected as the representative of the sequence. In other words, given a set of images $I = \{i_1, i_2, \ldots, i_n\}$, a fixed number (budget) of the image subset is selected via active learning. Each image $i_{s_i}$ represents a sequence $s_i$, and the video sequence subset is labeled in correspondence to the selected image subset. In this case, our problem pertaining to active learning for video sequence sample subset selection becomes one of image sample subset selection. The sample subset that is selected based on the active learning method is more diverse than the sample subset based on random selection. Labeling these sequence samples selected based on active learning for deep CNN model training can improve the representational capacity of the trained model under the premise that the budget is fixed.

2) Multiframe Cooperation-Based Selection Method: The sequence selection method based on the first frame image may be influenced by the background information, which causes the selected sequence to pay too much attention to the background relative to the target. However, our goal is to learn a model focusing more on the target than the background. Considering the time consistency of the target in motion, we present an active learning sequence selection method based on multiframe collaboration. More specifically, we select multiframe images $\{i_{s_i}, i_{s_{i+1}}, i_{s_{i+2}}, i_{s_{i+3}}, i_{s_{i+4}}\}$ with the same interval $a$ in each sequence $s_i$, fuse the information, multiply the features of the corresponding positions) of these multiframe images selected from each sequence as $I$, and then select a subset $\{I_{i_1}, I_{i_2}, \ldots, I_{i_a}\}$ of these fused images $\{I_1, I_2, \ldots, I_n\}$ using an active learning method. The multiframe collaboration method can eliminate background interference through the use of temporal relationships between multiple frames in each video sequence, thus ensuring the diversity of selected video sequences.

3) Neighborhood Validation-Based Reselection Method: Although the sequence selection method based on multiframe validation can eliminate the influence of the background and enhance the diversity of selected sequences, problems such as poor sequence quality (such as low resolution) and sequence isolation inevitably emerge during the collection of unlabeled sequences. Once these poor-quality or isolated sequences are selected for inclusion in the subset for labeling and used for model training, this will have a negative effect on the trained model. After considering the representativeness of the selected video sequence samples, we adopt a nearest-neighbor discrim-
Algo 1 : Active Learning for Training Samples’ Selection

Require: An unannotated sequence samples’ set $A$.
Ensure: A diverse and representative sample subset $subA$ is selected from $A$ with a given budget of $B$.

1: Given the distance matrix $M$ between samples in $A$;
2: $d_i$ is the distance between each sample $i$ in $A$ and its nearest neighbor $nn_i$, while $ave_d$ is the average nearest-neighbor distance;
3: Randomly select a sample $i$ with $d_i \leq ave_d$ to add to the subset $subA$;
4: if (the number of selected samples in $subA < B$) then
5: Use the FPS algorithm to select the next samples $i_t$;
6: if (the nearest neighbor $nn_i$ of $i_t$ is still not within $subA$ & $d_{i_t} \leq ave_d$) then
7: Add $i_t$ to the subset $subA$.
8: end if
9: end if

Algorithm 1 presents a general process for a training sample selection approach based on our proposed active learning method. For the purpose of disambiguation, the video sequence sampling method mentioned here refers to image sampling after multiframe fusion, which corresponds to the video sequence in which it is located. The selected subset of these samples corresponds to the selected subset of these unannotated video sequences. The subset of samples selected based on active learning can maintain the diversity (due to the FPS algorithm) and representativeness (due to the neighborhood validation) of the original sample set to the greatest extent possible under the fixed budget.

C. Tversky Loss for Bounding Box Estimation

To obtain more accurate tracking results, we adopt a Tversky loss to improve the bounding box estimation strategy in the tracking framework. The Tversky loss function can be defined according to the following format:

$$L_{tbb} = 1 - T(B, B^g)$$

where $L_{tbb}$ is the Tversky loss for bounding box estimation, while $T(B, B^g)$ is the Tversky coefficient

$$T(B, B^g) = \frac{|B \cap B^g|}{|B \cap B^g| + \alpha|B - B^g| + \beta|B^g - B|}.$$  (5)

In the Tversky coefficient $T(B, B^g)$, if $\alpha = \beta = 0.5$, $T(B, B^g)$ is the Dice coefficient; moreover, if $\alpha = \beta = 1$, $T(B, B^g)$ is the Jaccard coefficient. In $T(B, B^g)$, the $|B^g - B|$ term means to treat the background as the target, while the $|B^g - B|$ term means to treat the target as the background (shown in Fig. 2). In fact, it is even harder to accept that the tracker treats the background as the target, as this can shift the target’s appearance model and cause the tracker to lose its target. The Tversky loss function provides adjustable parameters used to adjust the weights of the $|B^g - B|$ and $|B - B^g|$ parts. Compared with the IoU loss, we can control the attention of the model by adjusting $\alpha$ and $\beta$ in the Tversky loss function, which can cause the model to pay more attention to the target than the background and avoid treating the background as the target, enabling it to obtain a more accurate boundary box of the tracking target. We train our full tracking framework by minimizing the total loss, which is the combination of this Tversky loss with the target classification loss

$$L_{tot} = L_{tbb} + \eta L_{cl}$$

where $\eta$ is the target classification weight used to adjust the specific gravity between the two loss parts.

IV. Experiments

In this section, we first introduce some experimental details of our ALT tracking method. We next report some ablation studies to verify the effectiveness of each active learning rule for training sample selection and the Tversky loss for target bounding box regression. Finally, we verified the effectiveness and competitiveness of our ALT tracker trained with limited budget samples through comparison with state-of-the-art trackers trained with extensive samples on seven standard benchmark datasets: OTB100 [11], UAV123 [2], G0T10k [12], LaSOT [37], TrackingNet [11], VOT2019 [38], and VOT2020 [39].

A. Experimental Details

We follow the DiMP [32] tracker, which applies the SGD with a momentum of 0.9 to train the deep CNN model. We use ResNet-50 as our backbone network in this article. The weight decay is $5e^{-4}$, the learning rate is $1e^{-5}$, and the target classification weight $\eta$ is $10^2$. The network is trained for 50 epochs with a mini-batch size of 32. Different from the DiMP [32] tracker, which uses training splits of the TrackingNet [11], LaSOT [37], GOT10k [12], and COCO [64] datasets, we use only those training samples selected from the GOT10k [12] dataset under a different budget. In the multiframe cooperation part, the interval $a$ is set to 10, so as to ensure the diversity of the same target in the sequence and ensure that the target does not disappear in the sequence. In the Tversky loss part, $\alpha$ and $\beta$ are set to 0.4 and 0.6, respectively. In this case, the boundary box estimation model can pay more attention to the target part without ignoring the background information. In the ablation study part, we incrementally change the training sample selection rules under the same budget to verify the effectiveness of each active learning rule. When the active learning rule is fixed, we gradually increase the budget value to verify that the performance of the trained network will be significantly improved as the number of training samples increases. Meanwhile, the Tversky loss was added under different training sample budgets to verify its effectiveness for target bounding box estimation. In the state-of-the-art comparison part, we used the deep CNN model...
TABLE I

| Budget | 50   | 100  | 500  | 1000 | 2000 |
|--------|------|------|------|------|------|
|        | Norm. Precision | AUC | Norm. Precision | AUC | Norm. Precision | AUC | Norm. Precision | AUC | Norm. Precision | AUC |
| Random | 49.9 | 42.5 | 54.3 | 46.5 | 57.6 | 51.5 | 58.6 | 52.6 | 59.2 | 53.0 |
| SAL    | 50.2 | 43.1 | 55.0 | 47.1 | 59.4 | 52.6 | 59.4 | 53.1 | 60.8 | 54.2 |
| MAL    | 52.3 | 43.9 | 55.4 | 47.9 | 60.2 | 52.9 | 60.7 | 53.9 | 61.9 | 55.4 |
| KMAL   | 52.5 | 44.6 | 55.9 | 48.3 | 61.1 | 53.6 | 62.9 | 55.6 | 65.3 | 57.1 |
| TKMAL(ALT) | 53.7 | 45.4 | 58.5 | 52.0 | 61.9 | 54.3 | 63.6 | 56.3 | 66.3 | 57.9 |

B. Ablation Study

We use the subset selected via active learning on the GOT10k [12] dataset to train the deep CNN model and carry out ablation studies on the LaSOT [37] benchmark to analyze the effect of different budgets and each active learning rule under the same budget in the training process. The comparison results are presented in Table I. To visually demonstrate the effect of each active learning rule in the tracker, we incrementally change the training sample selection rules. Starting with the basics of selecting unlabeled sequence samples based on SAL, then changing to multiframe coordination (MAL)-based samples’ selection, the unlabeled training samples are reselected using the nearest-neighbor verification (KMAL), while the target boundary box precision is improved using the Tversky loss (TKMAL). Meanwhile, we assess the tracking results of the deep CNN model trained on the random selection (Random) of the same number of samples. As shown in Table I and Fig. 4, with the same budget, the tracking performance of the trained model exhibits increasing improvement as the sample selection rules become increasingly perfect. Tracking performance improves with a budget under the same sample selection rules. At the same time, we can see that the introduction of the Tversky loss can effectively improve the tracking performance in each given budget.

C. State-of-the-Art Comparison

In this section, we present some quantitative comparisons of the proposed ALT tracker with some state-of-the-art trackers on the seven most challenging benchmark datasets to verify the effectiveness of our tracker.

1) OTB100: The OTB100 [1] benchmark contains two tracking performance evaluation criteria: precision score and success score. To test the proposed ALT tracker, we draw some experimental comparisons with some state-of-the-art trackers (namely, ATOM [16], GradNet [65], GCT [66], ARCF [67], UDT [68], DiMP50 [32], ROAM [34], SiamRCNN [45], DCFST [69], and PGNet [70]) on this dataset. Table II presents the experimental results of these comparisons over 100 testing videos. It can be seen that our ALT tracker, as well as the ROAM [34] tracker, achieved the best precision scores. The ROAM [34] is trained on large-scale datasets, which is highly time-consuming and expensive. In contrast, our ALT tracker only requires a limited number of training samples to achieve similar tracking results. Meanwhile, using the same backbone network (ResNet-50) as the DiMP50 [32] tracker, our tracker obtains some better tracking results than the DiMP50 tracker under the condition that limited training samples are used, and this fully demonstrates the effectiveness of our training sample selection strategy based on the active learning method and the accurate bounding box estimation improved by the Tversky loss.

2) UAV123: To evaluate the tracking performance of the proposed ALT tracker, we report some experimental comparisons between this tracker and other state-of-the-art trackers (namely, GFSDCF [61], DiMP50 [32], ARCF [67], SiamRPN++ [43], ATOM [16], GCT [66], CGACD [71], SiamBAN [36], SiamCAR [46], SiamAttn [44], and CLNet [72]) on this UAV123 [2] dataset. Table III presents the precision and success scores on 123 video sequences. From this table, we can determine that the proposed ALT tracker achieved the best tracking accuracy in terms of both the precision and success index. Compared with other tracking methods, the DiMP50 [32] tracker achieves superior tracking performance with regard to AUC (64.8%) and precision (85.8%) indexes. However, the proposed ALT tracker, which uses only limited training samples and uses the Tversky-loss-based bounding box estimation method, produces a certain degree of performance improvement. These experimental results demonstrate the effectiveness of our training sample...
TABLE II
COMPARISON RESULTS ON THE OTB100 [1] DATASET. THE TOP-3 SCORES ARE HIGHLIGHTED IN RED, BLUE, AND GREEN, RESPECTIVELY

| Trackers       | ALT   | ATOM[16] | GradNet[65] | GCT[66] | ARCF[67] | UDT[68] | DiMP50[32] | ROAM[34] | SiamRCNN[45] | DCFST[69] | PGNet[70] |
|----------------|-------|----------|-------------|---------|----------|---------|------------|----------|--------------|-----------|-----------|
| Reference      | Ours  | CVPR19'  | ICCV19'     | CVPR19' | ICCV19'  | CVPR19' | ICCV19'    | CVPR20'  | CVPR20'      | ECCV20'   | ECCV20'   |
| Precision      | 90.8  | 86.2     | 86.1        | 85.9    | 81.8     | 76.0    | 83.9       | 90.8     | 89.1         | 87.2      | 89.2      |
| Success(AUC)   | 69.2  | 66.1     | 63.9        | 64.8    | 61.7     | 59.4    | 68.7       | 68.1     | 70.1         | 70.9      | 69.1      |

TABLE III
COMPARISON RESULTS ON THE UAV123 [2] DATASET. THE TOP-3 SCORES ARE HIGHLIGHTED IN RED, BLUE, AND GREEN, RESPECTIVELY

| Trackers       | ALT   | GSDFC[61] | DiMP50[32] | ARCF[67] | SiamRPN++[43] | ATOM[16] | GCT[66] | CGACD[71] | SiamBAN[36] | SiamCAR[46] | SiamAttn[44] | CLNet[72] |
|----------------|-------|-----------|------------|---------|----------------|----------|---------|------------|--------------|--------------|--------------|-----------|
| Reference      | Ours  | ICCV19'   | ICCV19'    | ICCV19' | CVPR19'        | CVPR19'  | ICCV19' | CVPR20'   | CVPR20'      | CVPR20'      | CVPR20'      | ICCV20'   |
| Precision      | 87.1  | 76.7      | 85.8       | 67.6    | 80.7           | 85.6     | 73.2    | 83.3       | 83.3         | 76.0         | 84.5         | 83.0      |
| Success(AUC)   | 65.2  | 53.4      | 64.8       | 47.0    | 61.3           | 64.3     | 50.8    | 63.3       | 63.1         | 61.4         | 65.0         | 63.3      |

3) GOT10k: We conduct some experimental comparisons on the GOT10k [12] test set to evaluate our ALT tracker in relation to other state-of-the-art trackers, namely, LDES [73], SiamDW [74], ATOM [16], DiMP50 [32], DiMP50 [32], SiamCAR [46], D3S [76], ROAM [34], Ocean [35], DCFST18 [69], and DCFST50 [69]. The evaluation indexes include average overlap (AO) and success rate (SR0.50, SR0.75). We further show the tracking speed of each tracker. The comparative experimental results are presented in Table IV. Although our ALT tracker requires only limited samples to train the same network structure as the DiMP50 [32] tracker, the tracking performance of our tracker is superior to that of the DiMP50 [32] tracker, as well as achieving faster tracking speed. Meanwhile, our tracker achieved the best score on the SR0.75 index. The tracking accuracy of our ALT tracker is slightly lower than that of DCFST50 [69] on the indexes of AO and SR0.50; however, our tracker obviously exceeds this in tracking speed. These comparative results show that our tracker trained with only limited samples and equipped with a boundary box estimation strategy based on Tversky loss can still obtain tracking results comparable to those of state-of-the-art trackers.

4) LaSOT: To evaluate the proposed ALT tracker, we conduct several comparisons with several state-of-the-art trackers, namely, ATOM [16], SPM [75], D3S [76], DiMP50 [32], C-RPN [10], ROAM [34], GSDFC [61], CGACD [71], MAMLR [78], and UpdateNet [77] on the LaSOT [37] test set. Table V presents the comparison results in precision scores, normalized precision scores, and success scores. As can be seen from this table, while our tracker uses only a small number of training samples, it can achieve tracking performance resembling that achieved by these trackers when trained on a large-scale dataset. More specifically, in the success part, our ALT outperforms the third best tracker; compared with the best tracker (DiMP50 [32]), the proposed ALT tracker exhibited a decline in tracking performance of less than 0.5% when the training sample was far below it. This fully demonstrates the representational ability of our training samples selected based on active learning.

5) TrackingNet: To evaluate the proposed ALT tracker, we conduct some comparisons with several state-of-the-art trackers, namely, ATOM [16], SPM [75], D3S [76], DiMP50 [32], C-RPN [10], ROAM [34], GSDFC [61], CGACD [71], MAMLR [78], and UpdateNet [77] on the TrackingNet [11] test set. Table V presents the comparison results in precision scores, normalized precision scores, and success scores. As can be seen from this table, while our tracker uses only a small number of training samples, it can achieve tracking performance resembling that achieved by these trackers when trained on a large-scale dataset. More specifically, in the success part, our ALT outperforms the third best tracker; compared with the best tracker (DiMP50 [32]), the proposed ALT tracker exhibited a decline in tracking performance of less than 0.5% when the training sample was far below it. This fully demonstrates the representational ability of our training samples selected based on active learning.
Compared with the IoU-based ATOM [16] tracker, our ALT achieves an improvement of more than 2% on each metric. All these comparative results show that both the active-learning-based training samples selection method and the Tversky-loss-based bounding box estimation strategy are effective and worth practicing.

6) VOT2019: The VOT2019 dataset contains 60 test video sequences, and trackers are evaluated using expected AO (EAO), robustness, and accuracy. We show some experimental comparisons of our ALT tracker with the ATOM [16], DiMP50 [32], PrDiMP50 [50], TADT [81], SiamRPN++ [43], MemDTC [82], Ocean [35], CLNet [72], SiamBAN [36], SiamMask [83], and SPM [75] trackers on this dataset. The comparison results are listed in Table VI. From this table, we can know that our ALT tracker achieves the best EAO score compared with other trackers. Our ALT tracker adopts the same backbone network as the DiMP50 [32], PrDiMP50 [50], and SiamRPN++ [43] trackers, and the EAO score is significantly higher than these trackers, and this indicates that our proposed method can yield more accurate tracking results.

7) VOT2020: VOT2020 has the same dataset size as VOT2019, and trackers are also evaluated using EAO, robustness, and accuracy. We conduct some comparisons of our tracker with several state-of-the-art trackers, namely ATOM [16], DiMP50 [32], A3CTDmask [39], TRAT [39], UPDT [84], DCD [39], TCLCF [39], FSC2F [39], CSRDCF [85], and SiamFC [9] trackers on this dataset. The comparison results are shown in Table VII. Our ALT tracker has the best EAO score compared with other trackers. This is because our ALT adopts the same backbone network as the DiMP50 [32] tracker, and the accuracy score is slightly lower than this tracker, which not only indicates the importance of the Tversky loss in the target boundary box estimation but also indicates that our method can select more diverse and representative samples for deep model training.
TABLE VII

Comparison Results on the VOT2020 [39] Dataset. The Top-3 Scores Are Highlighted in Red, Blue, and Green, Respectively

| Trackers | ATOM[16] | DiMP50[32] | A3CTDmask[39] | TRAT[39] | UDPT[84] | DCDA[39] | TCLCF[39] | FSC2F[39] | CSRDCF[85] | SiamFC[9] | ALT |
|----------|----------|------------|----------------|----------|----------|----------|----------|----------|------------|----------|-----|
| Reference | CVPR19' | ICCV19' | ECCVW20' | ECCVW20' | ECCV18' | ECCVW20' | ECCVW20' | ECCVW20' | CVPR17' | ECCVW16' |
| EAO (%)  | 0.237    | 0.241     | 0.260       | 0.256    | 0.237   | 0.232    | 0.202    | 0.156    | 0.193     | 0.172    | 0.269 |
| Robustness (%) | 0.687 | 0.700     | 0.498       | 0.724    | 0.688   | 0.624    | 0.582    | 0.465    | 0.580     | 0.479    | 0.757 |
| Accuracy (%) | 0.440  | 0.434     | 0.634       | 0.445    | 0.443   | 0.456    | 0.430    | 0.397    | 0.405     | 0.422    | 0.426 |

Fig. 6. Qualitative comparison results of our ALT tracker with the SiamRPN++ [43], PrDiMP50 [50], DiMP50 [32], ROAM [34], Ocean [35], SiamBAN [36], and ATOM [16] trackers (sequences from top to bottom are: bird1, jump, matrix, skating1, soccer, and trans). Zooming in for a better view.

In summary, comparative results on these seven standard benchmark datasets show that our active-learning-based ALT tracker has achieved competitive tracking performance relative to these state-of-the-art tracking methods. It further shows that these training samples, selected based on the active learning method, have the perfect representational ability under a fixed budget. This can effectively solve the issue of it being prohibitively expensive and time-consuming to label samples in large quantities. It, therefore, provides a practical direction for the selection of those training samples with strong representational ability. Meanwhile, the Tversky-loss-based bounding box estimation strategy can also provide some performance gain to the tracker, as it makes the tracker give more attention to the target than the background.

D. Qualitative Comparison

To further visually display the tracking results of the comparison, we present a qualitative comparison of our ALT tracker with seven state-of-the-art trackers, namely, SiamRPN++ [43], PrDiMP50 [50], DiMP50 [32], ROAM [34], Ocean [35], SiamBAN [36], and ATOM [16]. Fig. 6 illustrates the comparison results on some challenging sequences. Although the SiamRPN++ [43] tracker trains its network over a large number of training datasets, its tracking performance is unacceptable when the tracking scenarios are under deformation (jump) and occlusion (bird1). In contrast, our ALT tracker uses limited training samples selected via active learning to train the network and achieves accurate tracking results in the same tracking scenarios (jump and...
bird1). The DiMP50 [32] tracker and the PrDiMP50 [50] tracker readily interfere in the scenarios of occlusion, background cluster, and deformation (e.g., bird1 and soccer). Our ALT tracker adopts the Tversky loss to obtain more accurate bounding box estimation for the tracking task. Compared with other trackers, our tracker can achieve more satisfactory tracking results in these complex tracking scenarios. In summary, compared with other trackers, our tracker shows more accurate tracking results, which indicates that our training samples selected based on active learning are more diverse and representative.

V. Conclusion

In this article, we propose a multiframe cooperation-based active learning method for training sequences’ selection to train deep CNN model on the target tracking task. In the proposed tracker, after considering the temporal relationship of the target in the sequence, we formulate an active learning method based on multiframe cooperation for selecting these training samples for labeling, which ensures the diversity of these selected samples, and then adopt a nearest-neighbor discrimination method based on the average nearest-neighbor distance to screen out isolated or low-quality sequences and thereby ensure the representativeness of these selected samples. Moreover, we adopt the Tversky loss function to improve the tracker’s bounding box estimation strategy, enabling it to obtain a more accurate target state. Extensive experimental results verify that our ALT tracker, when trained with limited budget samples, can achieve comparable tracking results when measured against state-of-the-art trackers that require extensive training samples. This work can provide guidance on the selection of samples for model training, but there is still room for improvement in using unlabeled data. We hope that we can use active learning to select unlabeled data directly for model training in future research, so that model training can completely get rid of the limitation of sample labels.

REFERENCES

[1] Y. Wu, J. Lim, and M. H. Yang, “Object tracking benchmark,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[2] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for UAV tracking,” in Proc. ECCV, 2016, pp. 445–461.
[3] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, and D. Tao, “Robust visual tracking revisited: From correlation filter to template matching,” IEEE Trans. Image Process., vol. 27, no. 6, pp. 2777–2790, Jun. 2018.
[4] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for correlation filter based tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2085–2083.
[5] S. Ge, C. Zhang, S. Li, D. Zeng, and D. Tao, “Cascaded correlation refinement for robust deep tracking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 99, pp. 1276–1288, Apr. 2020.
[6] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, “Robust multitask multiview tracking in videos,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 11, pp. 2874–2890, Nov. 2015.
[7] B. Ma, H. Hu, J. Shen, Y. Zhang, L. Shao, and F. Porikli, “Robust object tracking by nonlinear learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 10, pp. 4769–4781, Oct. 2018.
[8] Y. Zheng, H. Song, K. Zhang, J. Fan, and X. Liu, “Dynamically spatiotemporal regularized correlation tracking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 7, pp. 2336–2347, Jul. 2020.
[9] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional Siamese networks for object tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 850–865.
[10] H. Fan and H. Ling, “Siamese cascaded region proposal networks for real-time visual tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7952–7961.
[11] M. Müller, A. Bibi, S. Giancola, S. Alsabahi, and B. Ghanem, “TrackingNet: A large-scale dataset and benchmark for object tracking in the wild,” in Proc. ECCV, 2018, pp. 300–317.
[12] L. Huang, X. Zhao, and K. Huang, “GOT-10k: A large high-diversity benchmark for generic object tracking in the wild,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1562–1577, May 2021.
[13] Z. Ling, X. Li, W. Zou, and S. Guo, “Semi-supervised learning via convolutional neural network for hyperspectral image classification,” in Proc. 24th Int. Conf. Pattern Recognit. (ICPR), Aug. 2018, pp. 1–6.
[14] W. Liu et al., “Pair-wise uncertainty and diversity promoting early active learning for person re-identification,” ACM Trans. Intell. Syst. Technol., vol. 11, no. 2, p. 21, 2020.
[15] H. Yu, X. Yang, S. Zheng, and C. Sun, “Active learning from imbalanced data: A solution of online weighted extreme learning machine,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 4, pp. 1088–1103, Apr. 2019.
[16] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ATOM: Active tracking by overlap maximization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4660–4669.
[17] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, “Multi-class active learning by uncertainty sampling with diversity maximization,” Int. J. Comput. Vis., vol. 113, no. 2, pp. 113–127, 2014.
[18] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, “Compound rank-k projections for N-D data analysis,” IEEE Trans. Neural Networks Learn. Syst., vol. 27, no. 7, pp. 1502–1515, 2016.
[19] W. Liu, X. Chang, L. Chen, and Y. Yang, “Early active learning with pairwise constraint for person re-identification,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2017, pp. 103–118.
[20] P. Ren et al., “A survey of deep active learning,” ACM Comput. Surv., vol. 54, no. 9, p. 180, 2022.
[21] H. H. Aghdam, A. Gonzalez-Garcia, A. Lopez, and J. Weijer, “Active learning for deep detection of neural networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3672–3680.
[22] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, “The power of ensembles for active learning in image classification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 9368–9377.
[23] R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato, “Bayesian batch active learning as sparse subset approximation,” in Proc. NIPS, 2019, pp. 6359–6370.
[24] Y. Gal, R. Islam, and Z. Gahramani, “Deep Bayesian active learning with image data,” in Proc. ICML, 2017, pp. 1183–1192.
[25] D. Yoo and I. S. Kweon, “Learning loss for active learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 93–102.
[26] X. Shu, Y. Yang, and B. Wu, “A neighbor level set framework minimized with the split Bregman method for medical image segmentation,” Signal Process., vol. 189, Dec. 2021, Art. no. 108293.
[27] Y. Siddiqui, J. Valentin, and M. Niefner, “ViewAL: Active learning with viewpoint entropy for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9433–9443.
[28] X. Shu, Y. Yang, and B. Wu, “Adaptive segmentation model for liver CT images based on neural network with multi-level set method,” Neurocomputing, vol. 453, pp. 438–452, Sep. 2021.
[29] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in Proc. ECCV, 2016, pp. 472–488.
[30] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ECO: Efficient convolution operators for tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6638–6646.
[31] Q. Liu, X. Li, Z. He, J. Shen, and H. Wang, “Learning deep multi-level similarity for thermal infrared object tracking,” IEEE Trans. Multimedia, vol. 23, no. 21, pp. 2114–2126, 2020.
[32] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, “Learning discriminative model prediction for tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6182–6191.
[33] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, “Tversky loss function for image segmentation using 3D fully convolutional neural networks,” in Proc. Int. Workshop Mach. Learn. Med. Imag., 2017, pp. 379–387.
[84] G. Bhat, J. Johander, M. Danelljan, F. S. Khan, and M. Felsberg, “Unveiling the power of deep tracking,” in Proc. ECCV, 2018, pp. 483–498.

[85] A. Lukezic, T. Vojir, L. C. Vajec, J. Matas, and M. Kristan, “Discriminative correlation filter with channel and spatial reliability,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6309–6318.

Di Yuan (Member, IEEE) received the Ph.D. degree in computer applied technology from the Harbin Institute of Technology, Shenzhen, China, in 2021. He was sponsored by the China Scholarship Council as a Visiting Ph.D. Student with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia, from 2019 to 2021, where he was working with Prof. Xiaojun Chang. He is currently a Lecturer with the Guangzhou Institute of Technology, Xidian University, Guangzhou, China. His current research interests include computer vision, object tracking, machine learning, self-supervised learning, and active learning.

Xiaojun Chang (Senior Member, IEEE) was an ARC Discovery Early Career Researcher Award (DECRA) Fellow, from 2019 to 2021. He worked as a Post-Doctoral Research Associate with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, a Senior Lecturer with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia, and an Associate Professor with the School of Computing Technologies, RMIT University, Melbourne, VIC. He is currently a Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia. He mainly worked on exploring multiple signals for automatic content analysis in unconstrained or surveillance videos and has achieved top performance in various international competitions.

Qiao Liu received the Ph.D. degree in computer science from the Harbin Institute of Technology, Shenzhen, China, in 2021. He is currently a Lecturer with the National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing, China. His research interests include thermal infrared object tracking, infrared image processing, and machine learning.

Yi Yang (Senior Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He was a Post-Doctoral Researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with the University of Technology Sydney, Sydney, NSW, Australia. His current research interests include machine learning, multimedia content analysis and computer vision, such as multimedia indexing and retrieval, surveillance video analysis, and video content understanding.

Mingfei Shu received a B.S. degree in automation, the M.S. degree in power electronics and power transmission, and the Ph.D. degree in communications and information systems from Shandong University, Jinan, China, in 2003, 2006, and 2016, respectively. He is currently working with the Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan. His research interests include computer vision, the Internet of Things, and wireless sensor networks.

Guangming Shi (Fellow, IEEE) received the M.S. degree in computer control and the Ph.D. degree in electronic information technology from Xidian University, Xi’an, China, in 1988, and 2002, respectively. He is currently a Professor with the School of Artificial Intelligence, Xidian University. His research interests include artificial intelligence, intelligent communications, and human-computer interaction. Prof. Shi is the Chair of the IEEE CASS Xi’an Chapter, a Senior Member of ACM and CCF, and a fellow of Chinese Institute of Electronics and IET. He was awarded Cheung Kong Scholar Chair Professor by the Ministry of Education in 2012. He won the second prize of the National Natural Science Award in 2017.