Joint Representation Learning with Deep Quadruplet Network for Real-Time Visual Tracking

1st Dawei Zhang
College of Mathematics and Computer Science
Zhejiang Normal University
Jinhua, 321004, China
davidzhang@zjnu.edu.cn

2nd Zhonglong Zheng*
College of Mathematics and Computer Science
Zhejiang Normal University
Jinhua, 321004, China
zhonglong@zjnu.edu.cn

***Abstract*—Recently, trackers based on Siamese networks have attracted spread attention in the field of tracking because of a balance between accuracy and speed. Learning powerful representation via effective offline training strategy is critical for constructing high performance Siamese trackers. However, features extracted in most networks cannot accurately distinguish a tracked target from the background with semantic information in some challenging scenes. In this paper, we develop a Fully-Convolutional deep Quadruple Network (QuadFC) to learn more expressive representation via a novel multi-task loss function composed of a differential pairwise loss for tracking and a constructed triplet loss for similarity learning, which can be trained offline in an end-to-end mode. During inference, the proposed deep architecture does not need to update model and the positive-negative branches are removed to avoid unnecessary calculations. In particular, our approach is able to extract more discriminative features and perform robust visual tracking, due to joint representation learning and taking full use of original samples via the combination of positive-negative pairs. Furthermore, theoretical analysis of QuadFC is carried out through comparing the gradients of different loss functions. Extensive experiments on several tracking benchmarks, show that the proposed tracker achieves the state-of-the-art tracking performance while running at 68 FPS. The code can be available at https://github.com/DavidZhangdw/QuadFC.**

***Index Terms*—Siamese networks, Quadruplet network, Representation learning, Multi-task loss, Visual tracking.**

I. Introduction

Visual object tracking is one of the most fundamental and important tasks in computer vision and video analysis. It has a large range of applications in different fields, such as video surveillance, vehicle navigation, augmented reality and human-computer interaction, etc. We consider the single object tracking, which is aimed to find the location of a specific object in all subsequent frames with giving a bounding box of the unknown target in first frame of sequences. In fact, the core problem of visual tracking is how to accurately detect and localize the target in challenging scenes with deformation, illumination change, motion blur, occlusion, background clutter and other variations. Therefore, trackers are supposed to have the capability of robustness and discrimination simultaneously. Meanwhile, most applications demand real-time tracking.

Recently, some trackers based on Siamese networks [1], [3]–[6] have aroused extensive attention in tracking community, due to making a good balance between precision and speed. Specially, SiamFC [1] views the tracking problem as similarity metric in a certain feature space by constructing a full convolutional Siamese network. The inputs of networks contain an exemplar image patch centered the target and a search area with a larger size, in which the sub-windows with same size of template can be regarded as instances. Therefore, the logistic loss is generally adopted to maximize the similarity for positive instances and minimize scores for exemplar-negative pairs. In order to add training elements, SiamFC-tri [5] utilizes the underlying relations among the triplet (exemplar, positive as well as negative instances) and introduces triplet loss into SiamFC [1] for offline training. Although Siamese networks obtain balanced performance and tracking speed, feature utilized in most Siamese trackers can just distinguish foreground from the non-semantic background due to not utilizing the full potential of the training instances and not using deeper and wider networks. So the performance can not be guaranteed in some scene, like when the backgrounds are cluttered.

In this paper, we design a deep quadruplet network structure combined by a multi-task loss consisting of a differential pairs-loss and a triplet loss for joint representation learning. It

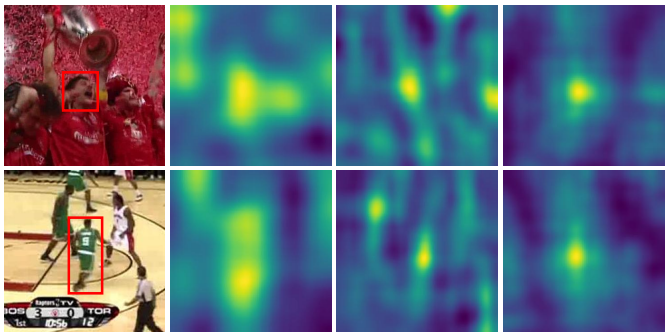


Fig. 1: Visualization of score maps. the left column is the search area, and the second to fourth columns denote response maps generated by SiamFC [1], SiamRPN [2] and our method, respectively. Siamese networks separate the object from the background ambiguously, while our approach obtains superior discriminability due to joint representation learning.

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

includes multiple pairs and a triplet mined online. To catch the underlying relation of instances, the designed architecture with shared weights has four inputs (exemplar, search area, positive and negative instance). We first sample an image pair including a template patch and a search image patch randomly as inputs of exemplar and search branches. Different from SiamFC [1] and SiamFC-tri [5], our purpose is to maximize the differential probability (e.g. $sigmoid(p) - sigmoid(n)$) through the combination of each positive and negative instances for training. The designed differential pairwise loss not only utilizes more samples for training, but also further capture the potential connection between exemplar-positive and exemplar-negative instances. As an example, we assume the numbers of positive and negative samples in a mini-batch as I and J, respectively. So the loss can generate $I \times J$ samples via combining I exemplar-positive with J exemplar-negative pairs, which is equal to [5] in terms of numbers of samples. In addition, for the triplet, representative positive and negative instances are selected from the search image patch as the input of positive and negative branches, which could further push away the similarity between positive and negative. Therefore, our framework aims to construct combination of pairs and mines a triplet by simple weighted summation of the differential pairwise loss and the triplet loss for effective training, which is helpful to learn discriminative features.

SiamFC [1] is considered as our baseline. We apply our quadruplet framework to train the network and adopt the same online tracking mechanism. Our method takes full use of the samples for similarity metric learning. As shown in Fig. 1, our tracker obtains more discriminative representation and better tracking accuracy. Moreover, to demonstrate the effectiveness of our approach, we provide the theoretical analysis by comparing the original logistic in [1], triplet loss [5] and our differential pairwise loss. To summarize, the main contributions of this work are three-fold:

- An end-to-end deep quadruplet framework specifically developed for effective offline training of Siamese trackers is proposed. The architecture utilizes fully inherent connections among samples and is applied for robust online tracking successfully.
- We proposed a multi-task loss function composed of a differential pairwise loss as well as the constructed triplet loss for joint representation learning. Furthermore, we also provide theoretical analysis of our loss function to prove the reasonableness of QuadFC.
- Comprehensive experiments, on several representative tracking benchmarks, indicate that our proposed QuadFC framework achieves state-of-the-art performance while tracking with a far beyond real-time speed.

II. Related works

A. Siamese Network Based Trackers

Visual object tracking can be considered as a similarity learning problem to some extent. SINT [4] being the pioneering work searches for the candidates most similar to the template image cropped in first frame. However, it runs only 2 fps. Furthermore, Bertinetto et.al [1] proposed a fully convolutional Siamese network to calculate the feature similarity for image pairs. It takes an exemplar and a much larger image as inputs and estimate the similarity for all sub-windows of search area. Especially, the trained Siamese network is directly used for online tracking without extra fine-tuning strategy.

There are large numbers of follow-up work [2], [5]–[9] of SiamFC. CFNet [9] introduces the Correlation Filter into Siamese networks for online learning. Dong et al. [5] utilizes of the underlying connections among samples by using a triplet loss for the training of Siamese trackers. Meanwhile, SiamRPN [2] considers the tracking as local detection problems through adding region proposal networks after the Siamese network. It obtains significant improvement by end-to-end training offline with large-scale datasets. Furthermore, SiamRPN++ [10] and SiamDW [6] utilize successfully deeper network [11] as backbone networks to replace AlextNet used by the original SiamFC [1], improving the performance of Siamese tracker due to deeper semantic features.

B. Quadruplet Networks in Computer Vision

Similarity learning with deep CNNs have been widely used for many tasks of computer vision, such as image retrieval [12], [13], face recognition [14] and person re-identification [15], [16]. FaceNet [14] utilizes similarity learning for face recognition, while the triplet loss [12] is also used for data clustering. Quadruplet networks [16] are proposed to pull the distance between classes and reduce the distance within the class for person re-id. For visual object tracking, Dong et al. [5] employed a triplet loss for the training of Siamese trackers. Furthermore, Quad [17] first applied the quadruplet network to siamese trackers. Different from most existing approaches above, our method uses simultaneously triplets mined from instances and the differential combination of pairs to catch the underlying connections among samples for effective training.

C. Representation Learning for Tracking

Representation plays an important role in the task of visual object tracking. HCF [18] integrates multi-layer convolutional features for online tracking by combining with correlation filter. MDNet [19] which designed specially for tracking learns more detailed representation for each domain via updating continuously. Furthermore, UPDT [20] provides an approach to fuse shallow features with deep ones for tracking, while a twofold siamese network (SA-Siam) [8] is proposed to extract semantic and appearance features respectively at very different levels. Recently, MLT [21] introduces a meta-learner network to achieve target-aware features of the specific targets.

III. The Proposed Framework

A. Siamese Networks for Tracking

Before introducing the architecture of our method, we briefly review SiamFC [1], which is the basic framework of our discussion. The original Siamese networks takes an image pair as inputs, including a template image z and a search area x.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

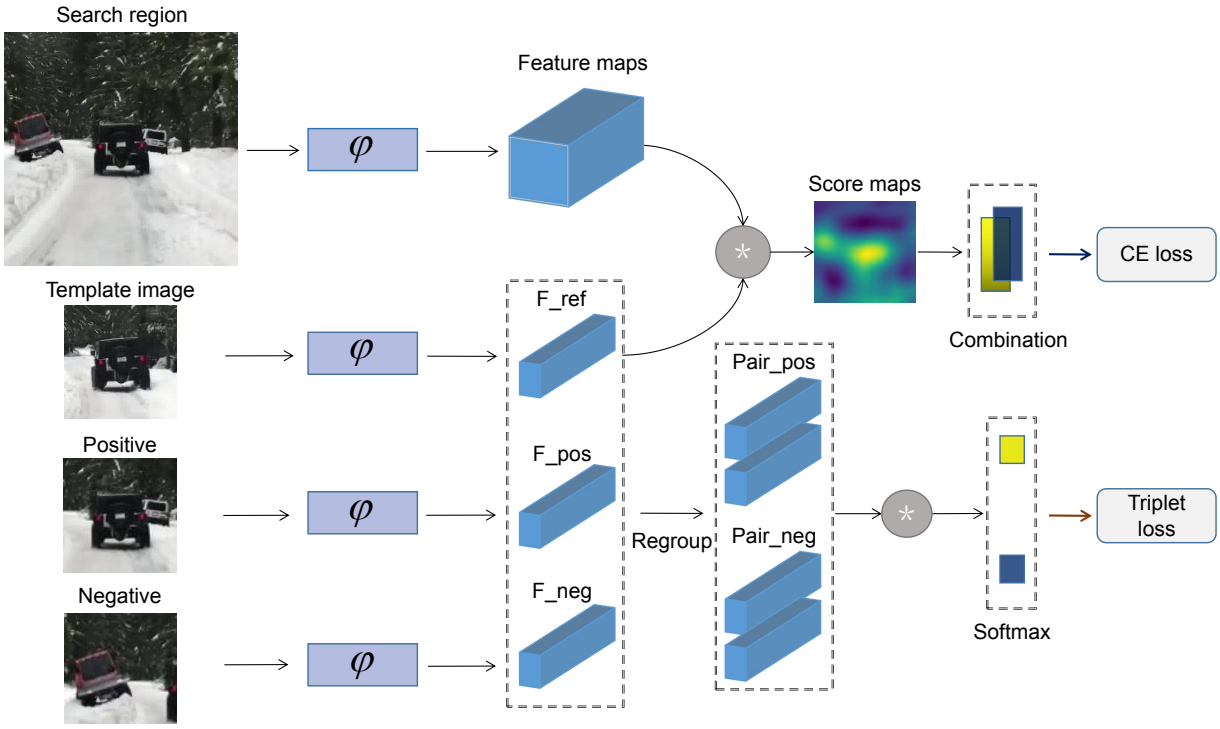


Fig. 2: The illustration of our proposed quadruplet architecture. There are four different inputs consisting of exemplar, instances, positive and negative branches. φ denotes the backbone network with shared weights, while $*$ represents the operation of cross-correlation. CE loss originates from the combination of exemplar-instances, and Triplet loss can be constructed by triplets.

The patch z cropped and scaled with the bounding box of the first frame denotes the tracked object, while x with a larger size indicates the candidate image from subsequent frames. Both inputs flow into a general CNNs φ with same weights θ, and we can obtain two feature maps. Thus, cross-correlation can be computed between them:

$$f_\theta(x,z) = \varphi_\theta(x) \star \varphi_\theta(z) + b \tag{1}$$

in which b represents a bias term. Eq. 1 equals to computing the similarity scores of template z over the search image x. So the maximum score in response map f indicates the most similar object. To this purpose, SiamFC [1] collected numerous image pairs (x, z) from video datasets for offline training, and designs the corresponding ground-truth y according to the distance from the center. Therefore, logistical loss can be used as objective function of this model, which is expressed as:

$$L_{logist}(y,f) = \sum_{v \in \mathcal{X}} \log\left(1 + e^{(-y[v] \cdot f[v])}\right) \tag{2}$$

where $\mathcal{X}$ represents the set of instances for the search branch, $f[v] = f(v,z)$, and the label of each exemplar-instance pair (v,z) can be denoted as $y[v] \in \{+1,-1\}$.

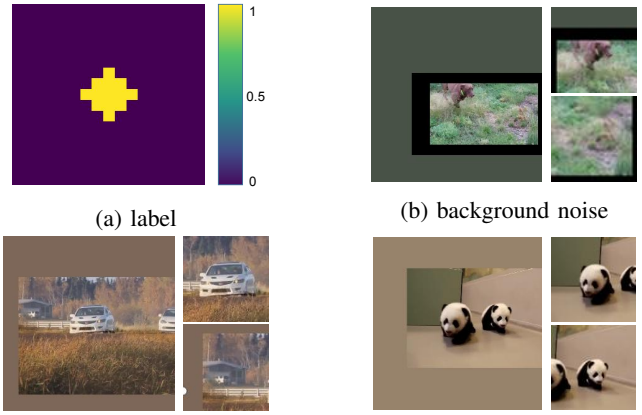
B. Quadruplet Networks

Inspired by triplet loss [5] and quadruplet network [16], we propose a deep quadruplet network containing four branches with shared backbone, while can be end-to-end trained through a joint loss. The proposed architecture is shown in Fig. 2.

Most existing siamese trackers [1], [2] utilize the classical but shallow AlexNet [22] as the backbone. To apply deeper and wider networks for Siamese trackers, SiamDW [6] introduces the residual unit containing a cropping operation. So the underlying position bias can be eliminated due to a simple yet effective cropping operation for feature maps. In our framework, CIResNet22W having doubled channel comparing with CIResNet22 [6] is employed as our backbone network, which is helpful to extract more rich semantic information and enhance tracking robustness and accuracy.

For inputs, four branches of shared network are respectively denotes as exemplar, search area, positive and negative branches. Each tuple of template image patch and larger search image patch sampled randomly in the same video sequence is regarded as inputs of exemplar and search branches. After extracting features and computing cross-correlation, we use Sigmoid function to measure the probability that the positive or negative is similar to the exemplar. To capture the underlying relation of samples, instances of search area can be divided into I positives and J negatives in line with the label. Notice that each instance is selected from the search area without extra samples. On one hand, exemplar and search area are applied to construct an embedded function $f(x,z)$, which can computes similarity scores between z and different instances from x. On the other hand, as shown in Fig. 3, we further select typical positive and negative according to label and score map computed by $f(x,z)$ for similarity metric learning.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) label (b) background noise

(c) different categories (d) the same category

Fig. 3: label and three positive-negative samples are presented. Positive and negative are decided from search images according to the label and response map produced by our network.

Three positive-negative samples are presented in Fig. 3 (b-d). (b) shows an instance pair of positive and negative from background noise, while (c) is a pair from different semantic categories. It's noteworthy that Fig. 3 (d) indicates an instance pair of positive and negative from the same category. In this case, selecting the highest negative instance will result unstable training and may obtain a worse model and features. So how to choose positive and negative samples is very crucial.

Now, we introduce the strategy of selecting triplets. We select the highest score of exemplar-positive pairs as the input of positive branch, that means the positive with maximum confidence. Different from positive instance, the top k instances with the highest scores from the negative instance are first selected, and then one of them is randomly chosen as the input of the negative branch. It indicates that a negative is selected from the top k most difficult to distinguish negative, which avoids over-fitting and unstable training. Once selecting the triplet, our framework can take full use of the combination of pairs for classification and constructs typical triplets including the positive and negative for similarity metric learning.

C. Multi-task Loss

For our framework, the optimization goal is the corresponding positives could be similar to z as well as negatives keeps away from the exemplar. To make this purpose, we develop a multi-task objective function for representation learning. With the training process, the loss error will be reduced. In fact, four branches are processed by the identical transformation. Like standard Siamese networks, our differential pairwise loss is only for the exemplar and search branches. Meanwhile, the another loss serves the exemplar, negative and positive branches. Therefore, giving the positive score set p and negative score set n from the similarity score set of exemplar-instance pairs, the first loss is applied by formulated as follows:

$$L_{dif}(p,n) = -\frac{1}{I \times J}\sum_{i}^{I}\sum_{j}^{J}\log prob\left(p_i, n_j\right) \tag{3}$$

where I and J represent the numbers of positive and negative instances. the differential probability (e.g. $sigmoid(p) - sigmoid(n)$) can be further formulated:

$$prob(p_i, n_j) = \begin{cases} \frac{1}{1+e^{-p_i}} - \frac{1}{1+e^{-n_j}}, & p > n \\ \lim\limits_{x \to 0_+} x \quad , & else \end{cases} \tag{4}$$

Inspired from the quadruplet framework [16], each triplet can be constructed between exemplar, positive and negative branches, and the triplet loss can be obtained by comparing their soft-max results. The loss function can be formulated:

$$L_{tri}(z,p,n) = \left[s\left(z,n\right) - s\left(z,p\right) + \alpha\right]_+ \tag{5}$$

where $[z]_+ = max(z, 0)$, α is a threshold, $s(z,p)$ and $s(z,n)$ is as:

$$s(z,p) = \frac{e^{f(p,z)}}{e^{f(p,z)} + e^{f(n,z)}}, \; s(z,n) = \frac{e^{f(n,z)}}{e^{f(p,z)} + e^{f(n,z)}} \tag{6}$$

Therefore, the entire loss is a weighted summation of the above two losses:

$$L = L_{dif} + \lambda L_{tri} \tag{7}$$

D. Inference

During online tracking phase, positive and negative branches are deleted. As is shown in Fig. 4, the template object and search branch are only applied to perform tracking. For each sequence, we crop an image patch centered on target position of the first frame as the template, and the enlarged search images in subsequent frames. Keeping the image centered on target in previous frame, we resize the exemplar and search area to 127×127 and 255×255, respectively. With the input, our trained-well network can efficiently extract the features and calculate a response map. Next we perform up-sampling for the response map by using bicubic interpolation with a factor 16. Therefore, according to the maximum of response map, the target in search image can be tracked.

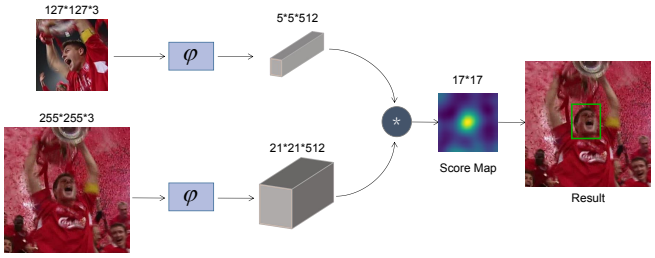


Fig. 4: The framework of our trackers during inference. Giving the exemplar patch and a search area, the response map can be generated by calculating cross-correlation between features. So the tracked object in search image can be located.

IV. Theoretical Analysis

As mentioned before, our differential pairwise loss and triplet loss [5] contain $I \times J$ samples while the amount in logistic loss is $I + J$. Now, we analyze and compare these

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

functions. For a set of instances $\mathcal{X}$, the original logistic loss of SiamFC [1] can be expressed as follows.

$$L_l = \frac{1}{I \times J} \sum_{i}^{I} \sum_{j}^{J} \frac{1}{2} \left(\log(1 + e^{-p_i}) + \log(1 + e^{n_j}) \right) \quad (8)$$

The triplet loss [5] in Siamese networks is expressed as:

$$\begin{aligned} L_t &= -\frac{1}{I \times J} \sum_{i}^{I} \sum_{j}^{J} \log \frac{e^{p_i}}{e^{p_i} + e^{n_j}} \\ &= \frac{1}{I \times J} \sum_{i}^{I} \sum_{j}^{J} \log \left(1 + e^{n_j - p_i}\right) \end{aligned} \quad (9)$$

In contrast, our differential cross-entropy loss (L_d) is also simplified for further analysis.

$$\begin{aligned} L_d &= -\frac{1}{I \times J} \sum_{i}^{I} \sum_{j}^{J} \log\left(\frac{1}{1 + e^{-p_i}} - \frac{1}{1 + e^{-n_j}}\right) \\ &= \frac{1}{I \times J} \sum_{i}^{I} \sum_{j}^{J} \Big(\log(1 + e^{-p_i}) + \log(1 + e^{n_j}) \\ &\quad - \log(1 - e^{n_j - p_i}) \Big) \end{aligned} \quad (10)$$

Comparison of Gradients. The gradient acts as an important effect in the offline training, because it directly participate in the back propagation phase. Thence, we analyze the characteristics of different terms. Firstly, for the logistic loss and SiamFC-tri [5], the gradients are respectively derived as:

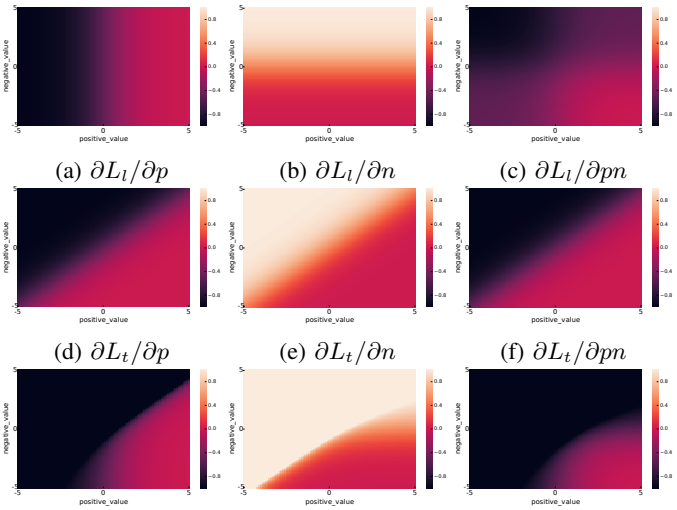
$$\frac{\partial L_l}{\partial p} = -\frac{1}{2\left(1 + e^{p}\right)}, \quad \frac{\partial L_l}{\partial n} = \frac{1}{2\left(1 + e^{-n}\right)} \quad (11)$$

$$\frac{\partial L_t}{\partial p} = -\frac{1}{1 + e^{p-n}}, \quad \frac{\partial L_t}{\partial n} = \frac{1}{1 + e^{p-n}} \quad (12)$$

For ours, the gradients of its term are given as:

$$\begin{aligned} \frac{\partial L_d}{\partial p} &= -\frac{1}{(1 + e^{p})} - \frac{1}{e^{p-n} - 1} \\ \frac{\partial L_d}{\partial n} &= \frac{1}{(1 + e^{-n})} + \frac{1}{e^{p-n} - 1} \end{aligned} \quad (13)$$

From Eq. 11, Eq. 12 and Eq. 13, we can find that $\partial L_d / \partial p$ and $\partial L_d / \partial n$ considering simultaneously both p and n, are able to make full use of the information provided by p and n, while $\partial L_l / \partial p$ and $\partial L_l / \partial n$ of logistic loss just hinge on p and n respectively. Comparing with $\partial L_t / \partial p$ and $\partial L_t / \partial n$, our $\partial L_d / \partial p$ and $\partial L_d / \partial n$ also consider the absolute gradient value of p and n respectively. It means that our differential pairwise loss can offer suitable feedback for back-propagation when the similarity value of exemplar-positive p is small than 0 (e.g. the network prediction is wrong) or less than the exemplar-negative pair n (e.g. the distance between positive and negative is not pulled well). For further analysis, visualization is presented in Fig. 5 via using the heatmaps of different gradients. It illustrates that $\partial L_l / \partial n$ and $\partial L_l / \partial p$ are independent of p and n, respectively, and $\partial L_t / \partial n$ and $\partial L_t / \partial p$ only depend on the relative value of the gradients between n and p. In contrast, $\partial L_d / \partial p$ and $\partial L_d / \partial n$ effectively pay more attention for $p < 0$, $n > 0$ as well as $p < n$ on the training phase.



(a) $\partial L_l / \partial p$ (b) $\partial L_l / \partial n$ (c) $\partial L_l / \partial pn$

(d) $\partial L_t / \partial p$ (e) $\partial L_t / \partial n$ (f) $\partial L_t / \partial pn$

(g) $\partial L_d / \partial p$ (h) $\partial L_d / \partial n$ (i) $\partial L_d / \partial pn$

Fig. 5: Gradient visualization of three losses (each row) for different items (each column) containing positive, negative and their difference $pn = p - n$, respectively. With comparison, our differential pair-wise loss is more effective than others.

V. Experiments

In this section, we firstly provide the details of implementation. Next, the comparing results of QuadFC with the state-of-the-art tracking algorithms are presented on five popular tracking benchmarks including OTB-2013, OTB-50, OTB-2015, VOT-2016 and VOT-207. Besides, ablation study is also shown to analyze the influence of each component for our proposed tracker. Finally, we present and analyze the qualitative evaluation by visualizing the tracking results.

A. Implementation Details

Networks. For our variants, we consider the deeper and wider CIResNet-22 [6] which is similar to ResNet [11] with Cropping Inside Residual(CIR) Units, but different size of receptive field and network stride. In our final framework, CIResNet22W with doubled channels comparing with CIResNet22 is effectively employed as our backbone network.

Training. To maintain the generalization property of feature representation, CIResNet-22 [6] is firstly trained offline from scratch on ImageNet Large Scale Visual Recogonition Challenge (ILSVRC2015) [23]. Therefore, the backbone network of our QuadFC is initialized with the weights pre-trained on ImageNet [23]. Next, like [6], the weight of the first 7×7 convolution layer is frozen, and other layers are gradually fine-tuned from back to front for stabilizing the training process. For optimization method, we use standard stochastic gradient descent (SGD) with momentum of 0.9 to train QuadFC and set the weight decay to 10^{-4}. To ensure fairness, we keep the setting to be consistent with SiamFC [1]. The learning rate is progressively reduced logarithmically from 10^{-2} to 10^{-5}. There are 50 epochs in total, and the batch size is 8. For our designed multi-task loss, we set $\lambda = 1$, $\alpha = 1$, $k = 5$.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

The training image pairs for QuadFC can be picked from video datasets with annotations, such as ImageNet VID [23] or the GOT10K [24]. VID contains more than 4000 sequences with about 1.3 million labeled frames, while GOT10K is a large-scale and high-diversity dataset including over 10000 long sequences, 9335 of which form the training-set is widely utilized for training in tracking methods recently. In each video sequence, we pick each image pair within the nearest 100 frames. For both training and testing, we set exemplar and search images of 127×127 and 255×255 respectively.

Tracking. On the inference phase, positive and negative branches are removed. Thus, the proposed tracker follows the same protocols as in [1], [6] and runs at similar speed with baseline trackers. The feature extraction $\varphi(z)$ for the template target is only calculated once in the first frame, and $\varphi(z)$ repeatedly matches to the feature $\varphi(x)$ from subsequent frames by cross-correlation. In addition, QuadFC matches the target over three scales $1.102^{\{-1,0,1\}}$ and utilizes the linear interpolation with a factor 0.5986 to avoid huge scale change.

The proposed architecture is implemented in Python with PyTorch 0.4.1 and all the experimental results are obtained on a workstation with Intel(R) Xeon(R) CPU E5-2683 v4 @2.10GHz and a NVIDIA GeForce GTX 1080 Ti GPU.

B. Comparison with the State-of-the-Art

We compare the proposed tracker with several state-of-the-art tracking algorithms, as well as some recent Siamese trackers on five popular datasets of OTB and VOT benchmarks.

OTB Benchmarks. The object tracking benchmarks (OTB) is composed of three datasets, named as OTB-2013 [25], OTB50 [26] and OTB-2015 [26]. The evaluation on OTB follows the standard protocols [25], [26]. Two metrics including center location error (CLE) of precision plots and area-under-curve (AUC) of success plots, are applied to estimate all trackers. Fig. 6 shows that QuadFC achieves the best performance with 0.676 and 0.680 of AUC scores on OTB-2013 and OTB-50, respectively. Note that our tracker surpasses other start-of-the-art tracking algorithms, such as the recent proposed SA-Siam [8], CREST [27] and SiamRPN [2]. Moreover, Fig. 7 also presents that compared with recent Siamese trackers, such as DaSiamRPN [28], CIRes-FC [6] and GradNet [29], our tracker still ranks first with the scores of 0.665 and 0.885 in terms of CLE and AUC on OTB-2015 with absolute advantage. That demonstrates the effectiveness of our architecture.

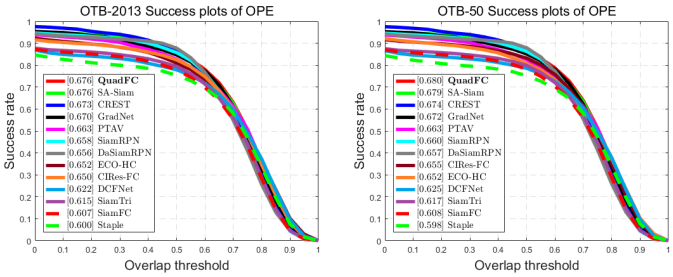


Fig. 6: The comparison results on OTB2013 and OTB-50. Thirteen trackers are ranked by AUC scores of success plots.

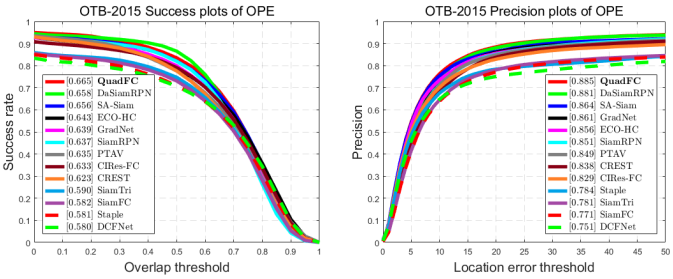


Fig. 7: The comparison of QuadFC with state-of-the-art trackers is shown by precision and successful plots on OTB-2015.

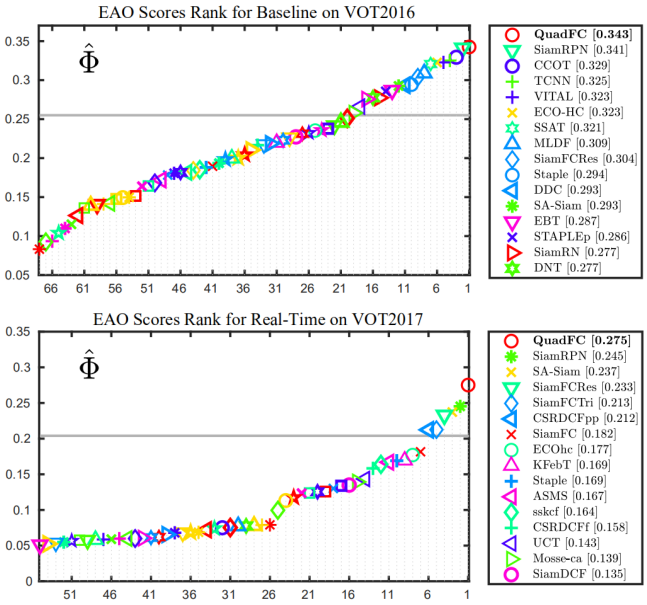


Fig. 8: Illustration of the expected average overlap scores for VOT-2016 baseline and VOT-2017 real-time challenges.

VOT Benchmarks. The visual object tracking (VOT) have several versions, and we select VOT-2016 [30] baseline challenge and VOT-2017 real-time challenge [31] to perform evaluation by the official toolkit. Different from OTB, VOT serves A (accuracy), R (robustness) and EAO (expected average overlap) as the metrics. VOT2016 consists of 60 challenging videos, which the bounding box is precisely auto-annotated. Our approach is compared with a large of competitive trackers including SiamRPN [2], VITAL [32], SiamFCRes [6], SA-Siam [8] and the other trackers in VOT-2016 reports. Fig. 8 shows that QuadFC surpasses CCOT [33] which is the winner and SiamRPN With a slight advantage, and ranks first with a 0.343 score of EAO on VOT-16 baseline challenge. For VOT-2017 dataset, 10 sequences of VOT-16 are replaced with more difficult ones. We also compare our QuadFC with SiamRPN, SA-Siam, SiamFCRes, SiamFCTri [5] and the total 51 trackers in VOT-2017 reports. As shown in Fig. 8, QuadFC achieves the best performance with a 0.275 EAO score and surpasses SiamFCRes, SiamRPN [2], SA-Siam [8] and CSRDCFpp [34] being the winner in VOT-17 real-time challenge by a large margin. That effectively proves the advantage of the proposed tracker in terms of performance and tracking speed.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

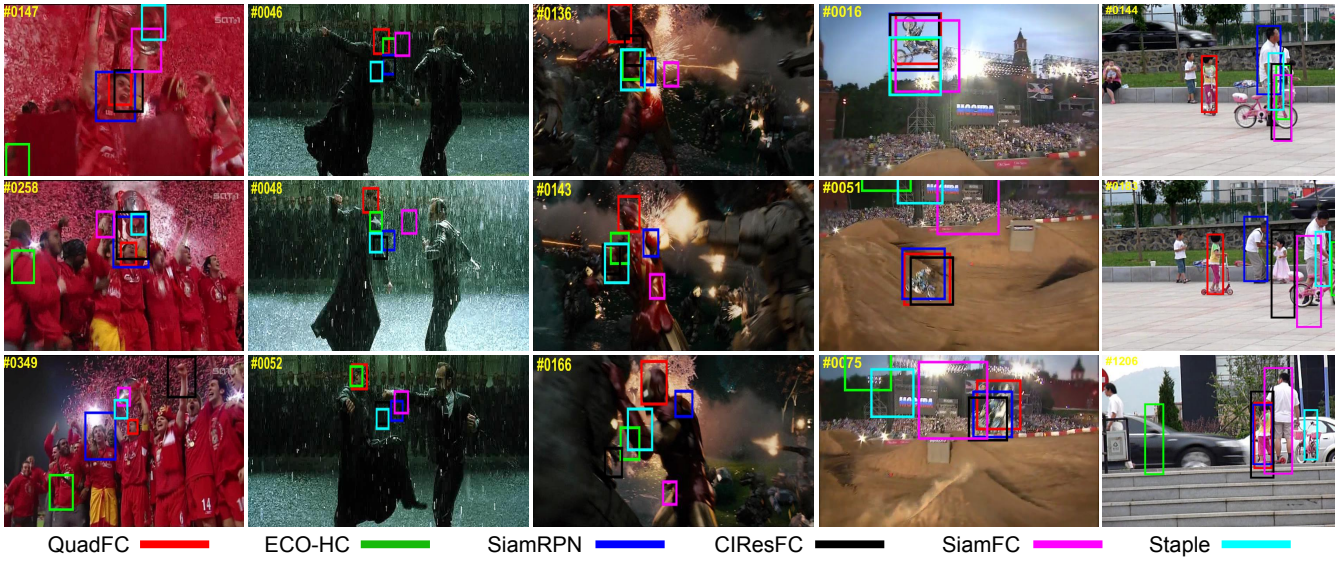


Fig. 9: Qualitative results by comparing QuadFC with other trackers. QuadFC tracks robustly in these hard cases.

C. Ablation Analysis

To identify the influence of different elements, we compare the baselines and several variants of our tracker. As presented in Fig. 10, our approach outperforms the baselines and all variants on OTB-2015. Compared with CIRes-FC [6], CIResW-FC obtains 1.4% and 1.1% gains due to more rich features with the doubled channel. Furthermore, QuadFC-d improves the performance with 1.2% and 0.9% by using our differential pair-wise cross-entropy loss for training. However, QuadFC-t just selects a positive and negative sample in each image pair by using the triple loss and achieves 1.1% and 1.2% decrease. This is because only the difficult instances are concerned and most of the samples are ignored during training, resulting in unstable convergence. Therefore, the final model QuadFC trained via our multi-task loss function can achieve the highest precision (0.885) and area-under-curve scores (0.665).

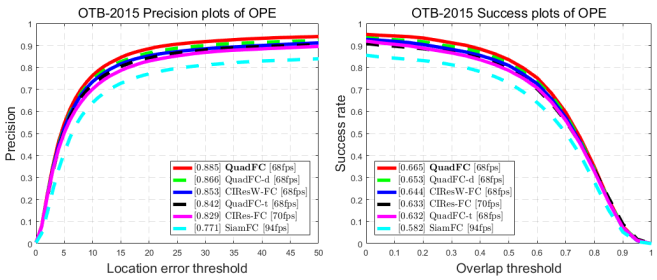


Fig. 10: Precision and success plots using OPE on OTB-2015. CIResW-FC indicates that the backbone network has doubled channels. QuadFC-d and QuadFC-t denote the model trained by the differential pair-wise loss and triple loss, respectively.

D. Qualitative Evaluation

As displayed in Fig. 9, we compare our QuadFC tracker with other five state-of-the-art real-time models (SiamFC [1], SiamFCRes [6], SiamRPN [2], ECO-HC [35] and Staple [36]) on five challenging sequences of OTB-2015 benchmark. The sequences in first column (Soccer), third column (Ironman) and in fifth column (Girl) are examples of semantic distractor, background clutter and partial occlusion situation. QuadFC can track the target successfully in terms of either precision or overlap, while SiamFC, ECO-HC, SiamRPN and SiamFCRes suffer a drift problem. The second column (Matrix) and fourth column (MotorRolling) are the challenging sequences where include fast motion and scale deformation. Our tracker still performs more accurate than the others. All trackers except QuadFC lost the target occasionally due to different challenging factors in Fig. 9, which proves that our tracker is robust to semantic background clutter and scale deformation.

VI. Conclusion

In this paper, we provide a full-convolutional deep quadruplet architecture for tracking, referred as QuadFC, to train Siamese networks in an end-to-end mode. we have especially designed a multi-task loss including a differential pairwise loss and a triplet loss by utilizing the combination of instances and mining of the potential connections among samples for joint representation learning. The resulting tracker greatly benefits from this scheme and achieves more discriminative representation. In addition, we also prove the reasonableness of our loss function in theory and experiments. Extensive results on five popular tracking benchmarks indicate that QuadFC outperforms the start-of-the-art tracking algorithms while keeping real-time speed. In the future, we will continue to explore more effective representation learning and the fusion strategy of features at different levels.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 61672467, 11871434 and 61877055).

REFERENCES

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision*, 2016.

[2] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conference on Computer Vision*, 2016.

[4] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[6] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional siamese network for high performance online visual tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] B. Zhuang, G. Lin, C. Shen, and I. Reid, "Fast training of triplet-based deep binary embedding networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[15] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv preprint arXiv:1703.07737*, 2017.

[16] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 403–412.

[17] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, July 2019.

[18] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3074–3082.

[19] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[21] J. Choi, J. Kwon, and K. M. Lee, "Deep meta learning for real-time target-aware visual tracking," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *In neural information processing systems*, 2012.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[24] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *arXiv preprint arXiv:1810.11981*, 2018.

[25] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[26] Y. Wu, J. Lim, and M. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[27] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *IEEE International Conference on Computer Vision*, 2017, pp. 2555 – 2564.

[28] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[29] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "Gradnet: Gradient-guided network for visual object tracking," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[30] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin, G. Vojir, A. Lukezic, and G. Fernandez, "The visual object tracking vot2016 challenge results," in *IEEE International Conference on Computer Vision Workshops*, 2016.

[31] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, and G. Fernandez, "The visual object tracking vot2017 challenge results," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[32] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang, "Vital: Visual tracking via adversarial learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[33] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *ECCV*, 2016.

[34] A. Lukežič, T. Voj'iř, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *International Journal of Computer Vision*, 2018.

[35] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE