IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 28, 2019, accepted February 14, 2019, date of publication February 20, 2019, date of current version March 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900296

An Integrated Deep Learning Framework for Occluded Pedestrian Tracking

KAI CHEN[1], XIAO SONG[1], XIANG ZHAI[2], BAOCHANG ZHANG[1,3], BAOCUN HOU[4], AND YI WANG[1]

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2]State Key Laboratory of Intelligent Manufacturing System Technology, Beijing 100854, China
[3]Shenzhen Academy of Aerospace Technology, Shenzhen 518057, China
[4]Beijing Aerospace Smart Manufacturing Technology Development, Co., Ltd., Beijing 100853, China

Corresponding author: Xiao Song (songxiao@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702703, and in part by the Open Fund of China State Key Laboratory of Intelligent Manufacturing System Technology, National Natural Science Foundation of China, under Grant 61473013.

ABSTRACT Numerous object-tracking and multiple-person-tracking algorithms have been developed in the field of computer vision, but few trackers can properly address the issue of when a pedestrian is partially or fully occluded by other objects or persons. In order to achieve efficient pedestrian tracking in various occlusion conditions, a pedestrian tracking framework is proposed and developed based on the deep learning networks. First, a pedestrian detector is trained as a tracking mechanism based on the Faster R-CNN, which narrows the search range and efficiently improves accuracy, as compared with the traditional gradient descent algorithm. Second, in the process of target matching, a color histogram and scale-invariant feature transform are combined to provide the target model expression, and a full convolution network (FCN) is trained to extract the pedestrian information in the target model, based on an FCN image semantic segmentation algorithm that can remove background noise effectively. Finally, the extensive experiments on a commonly used tracking benchmark show that the proposed method achieves better performance than the other state-of-the-art trackers in various occlusion situations.

INDEX TERMS Pedestrian tracking, Faster R-CNN, color histogram, SIFT, FCN.

I. INTRODUCTION

Pedestrian tracking is an important issue in the field of computer vision and has been widely used in many applications, such as unmanned vehicles, robots, and video surveillance [1]–[5]. Traditional trackers achieve good performance in simple scenes, but they perform more poorly with complex situations such as occlusion, illumination change, motion blur, and texture variation [9], [39]–[41]. Among these challenges, occlusion is a particularly important problem.

Normally, existing tracking algorithms can be divided into two categories, i.e., generative models and discriminative models. The former describe key characteristics of the target and then minimize the reconstruction error by searching the candidate target. Typical generative algorithms include sparse coding [6], locally orderless tracking (LOT) [7], distribution fields for tracking (DFT) [8], and incremental visual tracking (IVT) [9]. In contrast, the discriminant method distinguishes between the target and the background by training a classifier, and is therefore often called tracking-by-detection. The discriminant method is more robust because of the significant distinction between background and foreground information, and has come to gradually dominate the target tracking field.

Recently, a tracking method based on the correlation filter [10] has attracted attention because of its speed and accuracy. The correlation filter trains the filters by returning the input feature to the Gaussian distribution of the target, then finds the response peak in the forecast distribution to locate the target. The correlation filter employs the fast Fourier transform algorithm and therefore shows good performance. It has many extensions based on correlation filtering, including the kernelized correlation filter (KCF) [11], and the output constraint transfer for the kernelized correlation filter (OCT-KCF) [12].

These tracking algorithms have achieved good results, and most state-of-the-art generative and discriminative models

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

use the mean shift [13], [14] and particle filter [15] algorithms. These approaches often use the target in the previous frame as the initial location for the next search step. They might lose the target if a large portion of it is blocked or occluded for a relatively long time. To tackle this problem, a tracking framework will be proposed and implemented in the following sections.

The rest of this paper is organized as follows. Section II introduces related work. In Section III, we describe how the proposed framework can work with high accuracy in tracking pedestrians. Extensive experiments are discussed in Section IV. Finally, the conclusion is drawn in Section V.

II. RELATED WORK

Computer vision has been widely developed in terms of target detection and tracking. There exist many excellent tracking algorithms, such as CT [21], STC [22], CSK [23], KCF [11], OCT-KCF [12], and CN [37]. All these algorithms have made contributions in the field of target tracking, but most are sensitive to occlusion in practical environments. Experiments with pedestrian occlusion in OTB-50 [33] show that the algorithm has tracking drift problems if a pedestrian is partially or fully occluded by another object or person.

Object detection is an essential component of tracking. Recently, to provide better detection results, Ren [16] presented faster R-CNN, based on Fast Region-based Convolutional Network method (Fast R-CNN) [24], [25], proposing Region Proposal Networks (RPN) and the rapid generation of candidate areas. Faster R-CNN can self-generate a proposal box by using and sharing a convolution network with the target detection network, which greatly reduces the number of original proposed target region box while ensuring detection quality. It mainly uses the VGG-16 [20] as the feature extractor, which achieves the state-of-the-art accuracy on classification and localization tasks. VGG-16 contribution is an increasing network depth using smaller convolution filters, which shows a significant improvement on the network training effect by pushing the depth to 16-19 weight layers. Inspired by the faster R-CNN approach, we try to train a new pedestrian detection model and optimize it by expanding the training data set.

After detection of the proposal region, it is important to match that region with the subject pedestrian. In terms of target representation, one common practice is the RGB color histogram. This describes the number of color features in the image, which can reflect the statistical distribution of color in the image, as well as the basic tones. However, the histogram only contains the frequency of a color value, and each image has only one corresponding color histogram. The problem is that the same color histogram may correspond to various images. So, we can track the target by comparing color histograms in a relatively simple scene but cannot track it when occlusion occurs or the light changes. This is why we proposed combining an RGB color histogram with the SIFT feature to fulfill the pedestrian tracking task in this paper.

Lowe [17] proposed scale-invariant feature transform (SIFT), based on interest points of the object's local appearance rather than the size and rotation of the image. It also has high tolerance for light, noise, and change of micro-viewing angle. Based on these characteristics, it is relatively easy to capture objects accurately. With the SIFT feature, the detection rate of some occluded objects is further enhanced. Meanwhile, SIFT only requires three or more features to calculate the target position and orientation. With the development of computer hardware, SIFT can identify features in real time. With a large amount of feature information, SIFT is suitable for rapid and accurate matching in a massive image database.

Apart from these considerations, the background often contains many noise points when we try to match the features of a pedestrian using SIFT. To eliminate these noise points, we propose to implement full convolutional network (FCN) [19] image semantic segmentation. An FCN can judge whether a pixel belongs to a pedestrian, whereas a convolutional neural network (CNN) can only detect a rectangular region. In other words, an FCN can identify a pedestrian according to his shape while CNN cannot. Based on the idea in [19], this paper will construct a FCN based on the VGG-16 [20].

III. THE TARGET TRACKING FRAMWORK

In order to solve these various occlusion problems, a pedestrian tracking framework, shown in Figure 1, is proposed and studied in this paper. This framework has two major steps: first, pedestrian candidate region detection, and second, target representation and target matching. These will be studied in Section 3.B and 3.C, respectively.

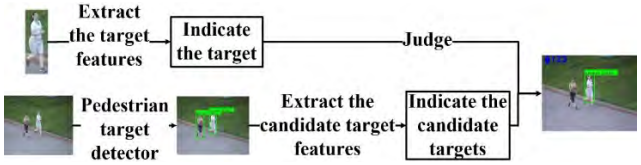

FIGURE 1. Main procedures of the proposed framework.

A. PROCEDURES OF PEDESTRIAN TRACKING FRAMWORK

Let us first present the overall process of the proposed pedestrian tracking framework. For each frame in the video, we propose to use the pre-trained Faster RCNN model to detect all pedestrians. Then, we select, in the first frame, the target most similar to the initial color histogram of the tracking target. For the following frames, the color histogram is used to represent all candidate targets and calculate similarity thresholds between the candidate and target models. The candidate target with the greatest similarity to the target model is used as the tracking target and updated with the target model.

If the maximum value of the color histogram similarity comparison is below the similarity threshold,

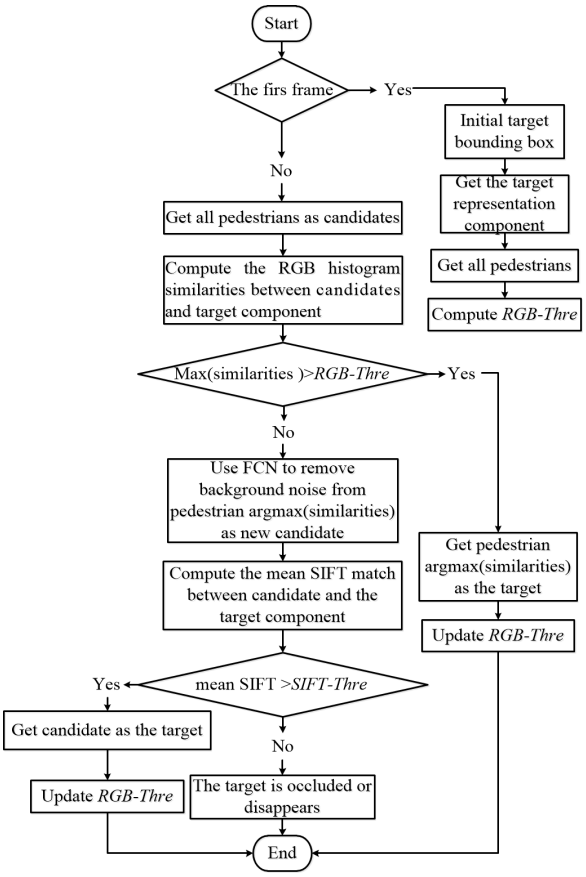

FIGURE 2. The flowchart of the framework.

the representation of the target is changed to the SIFT feature method. The target model is input into the FCN network to remove the background of the target model. After the new target model is obtained, the SIFT features of the new target model and all the candidate targets are calculated. The flowchart of the framework is shown in Figure 2. The pseudocode of the framework is shown in **Table 1**.

B. TARGET CANDIDATE REGION DETECTION

For the proposed pedestrian tracking framework, the first step is to locate the pedestrian's position accurately and quickly. As such, in Line 4 of the pseudocode in **Table 1**, the regional proposal network (RPN) of faster R-CNN is employed to generate the proposed area for each image. However, the accuracy of faster R-CNN detection [16] is not high when a pedestrian is partially or fully occluded. To tackle this problem, we propose to expand our training data with Caltech Pedestrian Dataset [43] (Figure 3) to enable our tracker to recognize partially blocked persons. The pedestrian detection model proposed in this paper includes four stages, which are shown in Figure 4.

Figure 4 describes the four stages of the proposed process of pedestrian detection model based on faster R-CNN.

In the 1st stage, a deep convolution network is used to extract the features of the regional proposals. The details

TABLE 1. Tracking framework pseudocode.

Pseudocode of the proposed Framework for Pedestrian Tracking	
1:	Initial target bounding box $\boldsymbol{b_0} = [x_0, y_0, w, h]$
2:	**if** the frame $n == 1$
3:	Randomly rotate and mirror the target to obtain 8 similar targets as a component
4:	Use pedestrian detector get all pedestrians and compute *RGB-Thre* using **Eqn.4**
5:	**if** the frame n > 1
6:	**repeat**
7:	Use pedestrian detector get all pedestrians
8:	**repeat**
9:	Compute the average RGB histogram similarities between each candidate pedestrian and the target component using **Eqn.1**
10:	**if** Max(similarities) > *RGB-Thre*
11:	Get pedestrian argmax(similarities) as the target and update target representation component refer to ***III.C.(3)***
12:	Update *RGB-Thre* using **Eqn.4**
13:	**else**
14:	Use FCN to remove background noise from pedestrian argmax(similarities) as the new candidate
15:	Compute the mean SIFT match between candidate and the target component
16:	**if** mean SIFT match > *SIFT-Thre*
17:	Get candidate as the target and update target representation component refer to ***III.C.(3)***
18:	Update *RGB-Thre* using **Eqn.4**
19:	**else**
20:	The target is occluded or disappears
21:	**end**
22:	**end**

are illustrated in Figure 5. VGG-16 models [20] are used as feature extractor for RPN and region of interest (RoI) Pooling layer. VGG-16 [20] uses several consecutive 3×3 convolution kernels instead of the larger convolution kernels in AlexNet [44] (11×11, 7×7, 5×5). For a given receptive field (the local size of the input picture associated with the output), stacked small convolution kernels are better than large convolution kernels because multiple layers of nonlinear layers can increase network depth to ensure more learning accuracy. Meanwhile, its cost is still small and has fewer parameters. As such, three 3×3 convolution kernels are used in VGG-16 instead of the 7×7 convolution kernels. Also, two 3×3 convolution kernels are used instead of 5×5 convolution kernels.

TABLE 2. Detection results on 30 pedestrian sequences of VOT2016-2018. The hardware configuration is Intel I7 4.2 GHz (4 cores) CPU, 16GB RAM and NVIDIA GeForce GTX 1080Ti GPU with memory of 11 GB.

Detectors	Train	Test	AP (%)	FPS
Faster R-CNN VGG-16[16]	VOC 2007+ 2012	30 Pedestrian Sequences of VOT 2016-2018 [45]	82.3	10
Faster R-CNN ZF[16]			69.8	20
YOLO VGG-16[46]			70.1	19
YOLO[46]			67.5	42
Fast YOLO[46]			55.7	150
Our	VOC 2007+2012+ Caltech		86.8	33

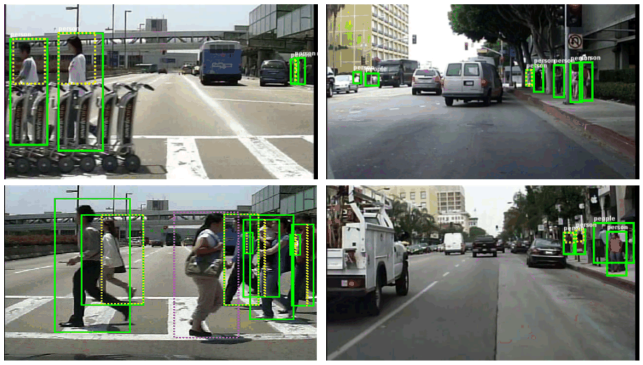

FIGURE 3. The solid green boxes denote the full pedestrian extent while the dashed yellow boxes denote the visible regions. The Caltech Pedestrian Dataset consists of approximately 10 hours of 640 × 480 30Hz video taken from a vehicle driving through regular traffic in an urban environment. About 250,000 frames (in 137 approximately minute long segments) with a total of 350,000 bounding boxes and 2300 unique pedestrians were annotated. The annotation includes temporal correspondence between bounding boxes and detailed occlusion labels.

In the 2nd stage, the regional proposal network (RPN) is used to generate the proposed area for each image. RPN uses pre-trained convolution neural networks (CNNs) to segment images. It requires an image of any size as input and outputs multiple rectangular area proposals. Due to the high flexibility of the CNN, there is no need to use similar behavior class objects to generate specialized regional proposals. Figure 4 shows the specific structure of the RPN network. It can be seen that the RPN network is divided into 2 components. The upper one is used to calculate the bounding box regression offset for the anchors. The lower one uses softmax to classify whether the anchors belong to the foreground or the background to obtain the precise proposal. The final proposal layer is responsible for synthesizing foreground anchors and bounding box traction offsets to acquire proposals, rejecting proposals that are too small and out of bounds.

In the 3rd stage, RoI pooling layer is responsible for collecting the proposal and calculating the proposal feature maps for delivery to the subsequent network which first maps the feature map into each region proposal so that the features are in the same location of the feature map. Then, it uses max pooling to transform this portion of the feature map into a small region of interest of size 7×7. The RoI is a rectangular window with four-tuple (x, y, w, h) which denotes the top left corner, width and height. Each $h \times w$ region proposal is max pooled using a sub-window of size approximately $h/7 \times w/7$. The RoI layer is also a special case of a SPPNet [25] which has only one level. The RoI is mapped to a fixed-sized vector using two networks, and this is input to classification layer to obtain the final confidence scores and refined bounding box coordinates.

In the 4th stage, the classification layer uses the adopted feature maps from RoI Pooling layer to determine whether each proposal is a pedestrian through the full connect layer and softmax, and outputs a probability vector, at the same time, using the bounding box regression to obtain the position offset of each proposal, and finally obtain a more accurate pedestrian detection box.

To test the feasibility of this proposed framework, 30 Pedestrian Sequences of Visual Object tracking benchmark (VOT) 2016-2018 [45] are tested. The measurements are detection Average Precision (AP) and frame per second (FPS). From **Table 2** we can observe that our pedestrian detector maintains the highest AP value while FPS can reach the real-time requirement.

C. TARGET REPRESENTATION AND MATCHING

In a visual tracker, target representation is a major component. A number of algorithms [26] have been proposed, such as global templates (raw gray scale values) [27]–[31], color histograms, Scale-invariant feature transforms (SIFT), HOGs [32], and covariance region descriptors. Of these approaches, the color histogram approach is fast and easy to implement but less accurate when there are environment changes: light, blocking, etc. In contrast, SIFT is a little slower but more accurate and shows good resistance to these environmental changes.

As such, in Line 9-11 of the pseudocode in Table 1, we first propose an improved form of color histogram to represent the target. The main advantage of the color histogram is that it has faster than real-time calculation speed. SIFT is used as a second choice because of its robust accuracy and real-time performance. The details are as follows.

1) COLOR HISTOGRAM MATCHING

In this paper, we convert all the images to the same size of $256 * 256$ pixels. Then, we calculate the similarity of the pictures' RGB channels. The average value of these three

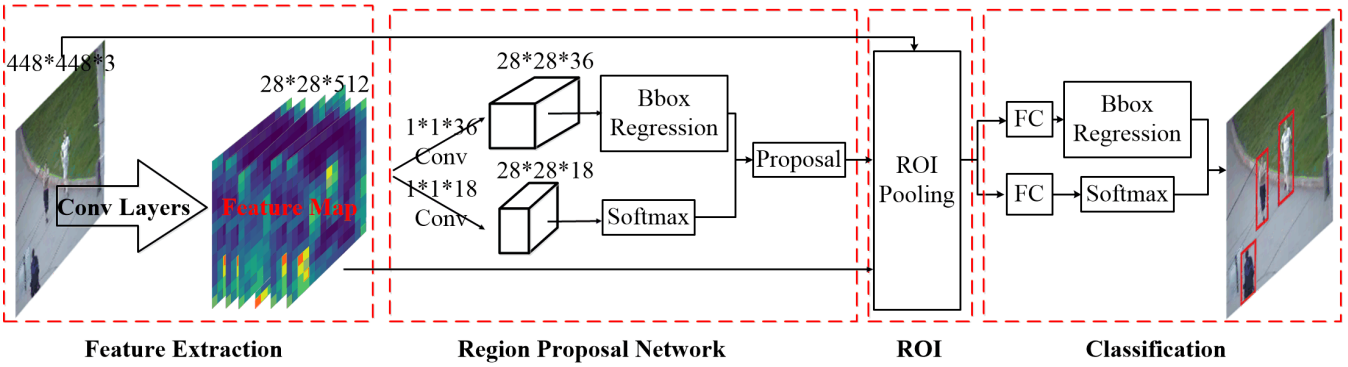

FIGURE 4. The process of pedestrian detection model based on faster R-CNN.

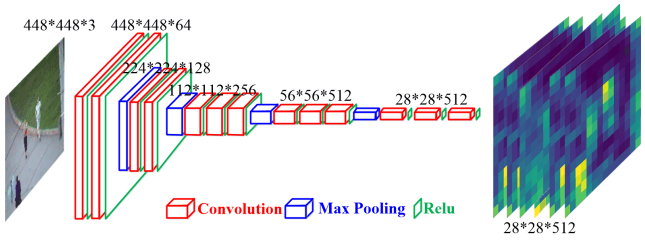

FIGURE 5. Red box represents convolution operation. Blue box and green box are Max Pooling and Relu operations, respectively. There are 13 convolution layers and 4 Max Pooling layers. The input image is resized to 448*448, resulting in a 512-level 28*28 size feature maps.

similarities is used as the final similarity of the compared images.

Although the traditional color similarity calculation methods [18], [40], [41] are useful, their discrimination rates are not high. We carried out several experiments with the video sequences in OTB-50 [33] and found that the main difference between the histograms of the image is the peak segment of the histogram, shown in Figure 6. Therefore, we hypothesize that we can compute the similarity of the peaks in order to discriminate between different persons.

To test this hypothesis, this paper proposes a new similarity calculation method: compute the mean RGB value (the

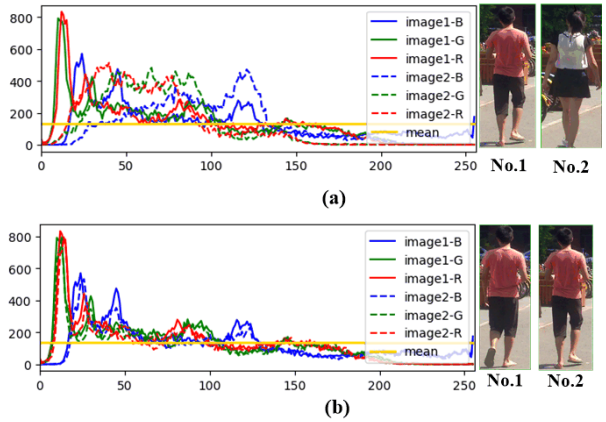

FIGURE 6. Comparison of color histograms of two images. The horizontal axis is the RGB brightness, ranging from 0 to 255. The vertical axis is the number of pixels that possess a certain RGB value in an image. (a) shows two different people in the same frame. (b) shows the same person in two frames.

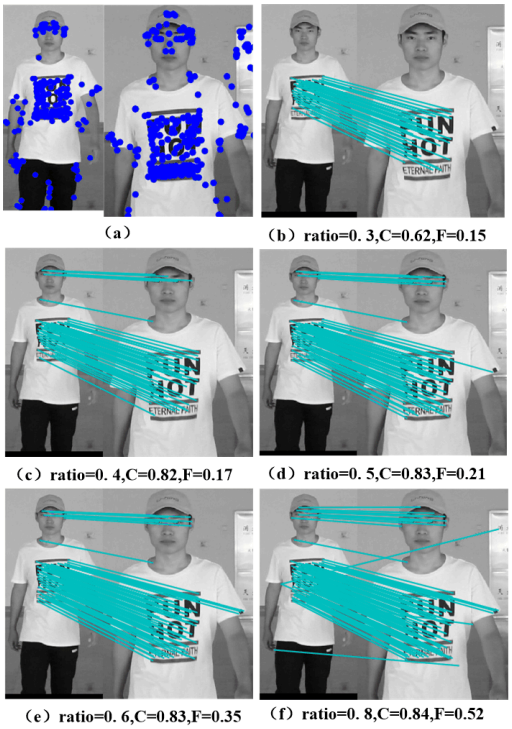

FIGURE 7. The effects of feature matching in different locations with various thresholds. C and F are correct and false matching rate, respectively. (b) ratio = 0.3, C = 0.62, F = 0.15. (c) ratio = 0.4, C = 0.82, F = 0.17. (d) ratio = 0.5, C = 0.83, F = 0.21. (e) ratio = 0.6, C = 0.83, F = 0.35. (f) ratio = 0.8, C = 0.84, F = 0.52.

yellow line in Figure 6) and only consider the peak values higher than this average value. Then, calculate the similarity of each peak segment value and their average similarity value. To compute the similarity of a RGB channel, we first normalize the pixel values, then use Equation (1) to calculate the color histogram similarity of people i and j in image frame m:

$$S_{i_m,j_m} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{o_n' - o_n}\sum_{k=o_n}^{o_n'}\left(1 - \frac{\left|C_{i_m}^k - C_{j_m}^k\right|}{Max\left(C_{i_m}^k, C_{j_m}^k\right)}\right)\right) \quad (1)$$

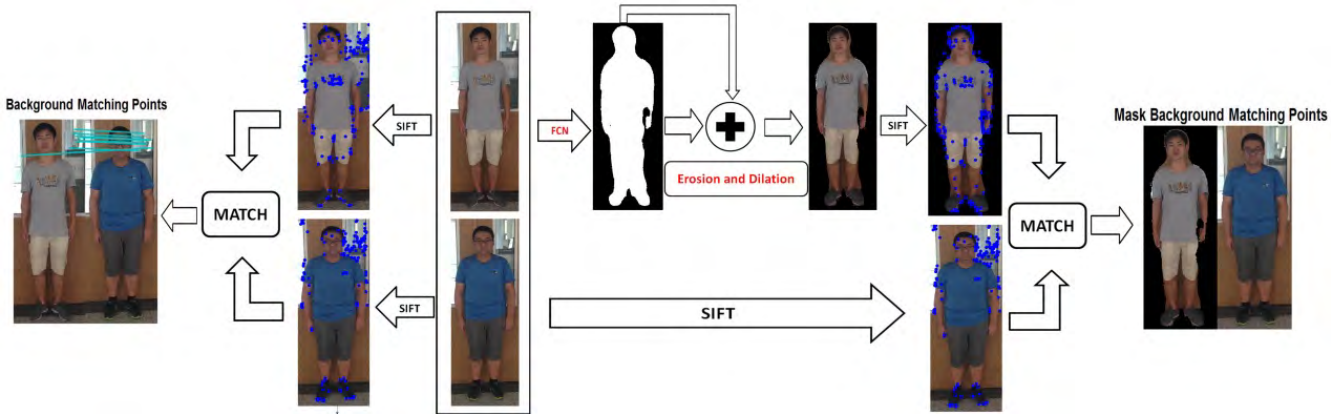

FIGURE 8. Removing the background matching points with FCN can generate more detection accuracy with SIFT. The two leftmost images (of different persons) might have the same SIFT key points because their backgrounds are similar. SIFT might confuse the target and the background noise. The upper images show that we only need to mask the background of the target. Then, SIFT will not find similar noise points in the upper and lower images.

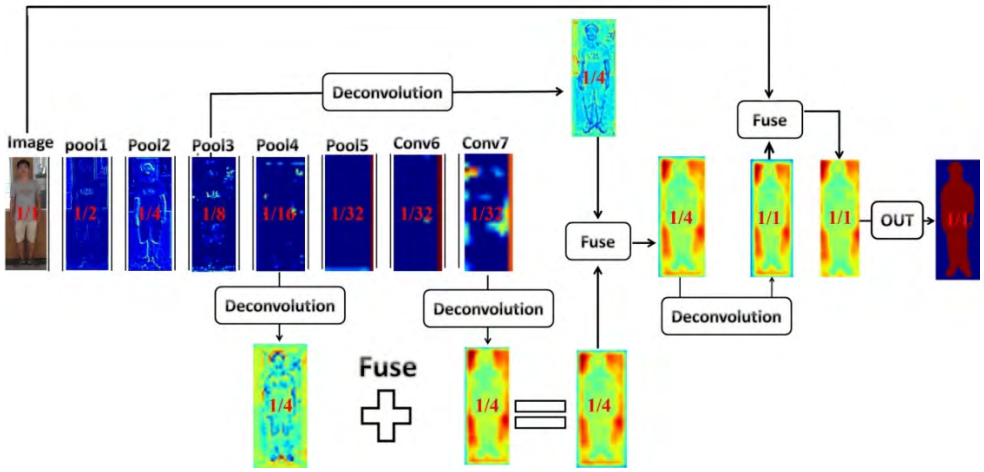

FIGURE 9. The structure of the full convolution network.

where, $S_{i_m,j_m} \in [0, 1]$, $C^k_{i_m}$ and $C^k_{j_m}$ are the color histogram pixel values of persons i and j in frame m. k is the index of RGB brightness. N is the number of peaks, $o_n^{'} - o_n (o_n < o_n^{'})$ is the interval of RGB brightness (horizontal axis of Figure 6) for each peak higher than the mean RGB value.

Moreover, to measure the performance of the similarity calculation method, a metric of degree of differentiation D is designed in Equation (2), where M is the number of frames in the video sequence, and P_m is the total number of people in image frame m. The numerator of D is the similarity of the target in two adjacent frames minus the mean similarity between the target and other non-targets in the same frame; the denominator of D is the similarity of the target in two adjacent frames.

$$D = \frac{\frac{1}{M-1}\sum_{m=2}^{M}\left(S_{i_m,i_{m-1}} - \frac{1}{P_m-1}\sum_{j=1,j\neq i}^{P_m} S_{i_m,j_m}\right)}{\frac{1}{M-1}\sum_{m=2}^{M} S_{i_m,i_{m-1}}} \tag{2}$$

Table 3 compares the traditional method and the proposed method in four groups of experiments (these video sequences are from OTB-50 [33]). It shows that the method proposed in this paper improves the ability to distinguish the target.

TABLE 3. The degree of differentiation D (equation 2) between two methods.

Video sequence	OTB(1)	OTB(4)	OTB(8)	OTB(15)
Tradition [18,42,43]	41.8%	33.3%	35.4%	28.5%
Proposed	43.5%	40.1%	47.3%	36.7%

2) SIFT MATCHING

In lines 22-26 of the pseudocode in **Table 1**, the color histogram is not valid when the above similarity S_{i_m,j_m} is below the threshold. We propose to use SIFT to detect the target pedestrian. The threshold of S_{i_m,j_m} is denoted by $Thre_{i_m}$, which is proposed to be affected by the number of pixels and the difference among pedestrians in the image. Therefore, we think $Thre_{i_m}$ is not a fixed value. As such, we design Equation (4) to compute $Thre_{i_m}$ of person i in frame m. If there

TABLE 4. Correct matching rate and False matching rate of OTB-50 (Video 1, 4, 8, 5 that have occluded persons).

Ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Correct matching rate	0.12	0.48	0.65	0.83	0.84	0.84	0.85	0.85	0.87	0.89
False matching rate	0.05	0.08	0.11	0.13	0.24	0.33	0.51	0.54	0.57	0.61

is only one person in the first frame, we set the threshold to 0. If there are more than two persons, we use the maximum of set L_m (Equation 3), the second largest value of L_m ($SubMax\ (L_m)$), and a parameter λ to calculate the threshold for the next tracking step.

$$L_m = \left(S_{i_m,1_m}, S_{i_m,2_m} \cdots, S_{i_m,j_m}\right), \quad (i \neq j) \tag{3}$$

$$Thre_{i_m} = \begin{cases} 0, & |L_m| = 1 \\ (Max\ (L_m) - SubMax\ (L_m)) * \lambda & \\ +SubMax\ (L_m), & |L_m| > 1 \end{cases} \tag{4}$$

In order to eliminate the key points which have no matching relationship due to image occlusion and background confusion, the SIFT matching method compares the nearest and second nearest neighbor distances. All key points have some nearest neighbor in the key point database. A key point match is considered positive when the ratio between its Euclidian distance and the distance of the second nearest neighbor is greater than a certain threshold T and the pair is matched.

Although Lowe [17] recommends this threshold ratio is 0.8, our experiments on correct matching rate and false matching rate of OTB-50 (videos 1, 4, 8, 5 that have occluded persons) show that, after a large number of matches between two pictures with arbitrary scale, rotation, and brightness changes, the best ratio for detecting a target pedestrian is between 0.4 and 0.6. As shown in **Table 4**, a ratio smaller than 0.4 will reduce the correct matching point and produce few successful matches. A ratio greater than 0.6 leads to a large number of false matches. For instance, Figure 7 a) shows all the constant invariant feature points of the same person in different positions in the video.

In this paper, SIFT feature matching is used to find pedestrians after occlusion. As this detection requires high accuracy, we take ratio $=$ 0.4. To enhance the real-time performance of SIFT, this paper uses the binary file provided by the open source toolkit VLFeat [42] to calculate the SIFT features of the image.

In the process of feature matching, we think that the background of the target persons is characterized by noise points, as shown in Figure 8. We only want to match the feature points on the target pedestrian. Therefore, we propose to employ full convolutional network (FCN)-based semantic image segmentation to remove the background noise.

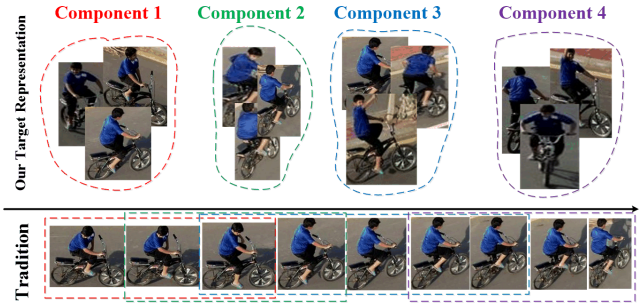

FIGURE 10. Visualization of the target representation in the traditional trackers (lower row) and our method (upper row).

Our effect is better than that achieved in [38], which used low- level feature video processing algorithms to extract the shot boundaries from a video scene and to identify dominant colors within these boundaries. As shown in Figure 8, we combine the result of the input of the target model in the FCN with the original image to eliminate the background. The comparison between the left and right graphs shows that our method can remove the influence of the background feature points.

VGG-16 is commonly used in object classification, but it cannot be used to detect the background and extract the foreground. To tackle this, we develop a full convolution network based on the VGG-16 network [20], which is a designed convolutional neural network (CNN). In this CNN structure, the first five layers are convolutions. The 6th and 7th layers are one-dimensional vectors of length 4096. The 8th layer is a one-dimensional vector of length 2—the target person and the background, respectively corresponding to the probability of a section of the image being either a pedestrian or part of the background.

Then, a convolution kernel of (4096, 1, 1), (4096, 1, 1), (2, 1, 1) is employed to transform the 6-8th layers into convolution layers.

Next, as shown in Figure 9, the original image is convoluted and pooled to 1/2, 1/4, …, 1/32 after conv1-5 and pool1-5. The image is reduced to 1/32 after the fifth convolution operation, conv5 and pool5. This produces a heat map, labeled 'Conv 7' in Figure 9. Now, we implement an up-sampling operation of this heat map, i.e., a de-convolution operation. This operation is iterated to restore the features in the original image, as Figure 9 shows.

TABLE 5. Video sequence attribute description.

Video sequence	Occlusion	illumination variations	scale variations	motion blur	low resolution	Image size	Pedestrian number
1	N	N	N	N	N	960*544	1
2	N	N	Y	Y	Y	640*480	1
3	Y	N	N	N	Y	352*288	3
4	Y	N	N	N	Y	352*288	3
5	Y	N	N	N	N	960*544	2
6	Y	Y	Y	N	Y	480*640	1
7	Y	N	Y	N	N	1280*720	2
8	Y	Y	Y	N	N	1920*1080	8

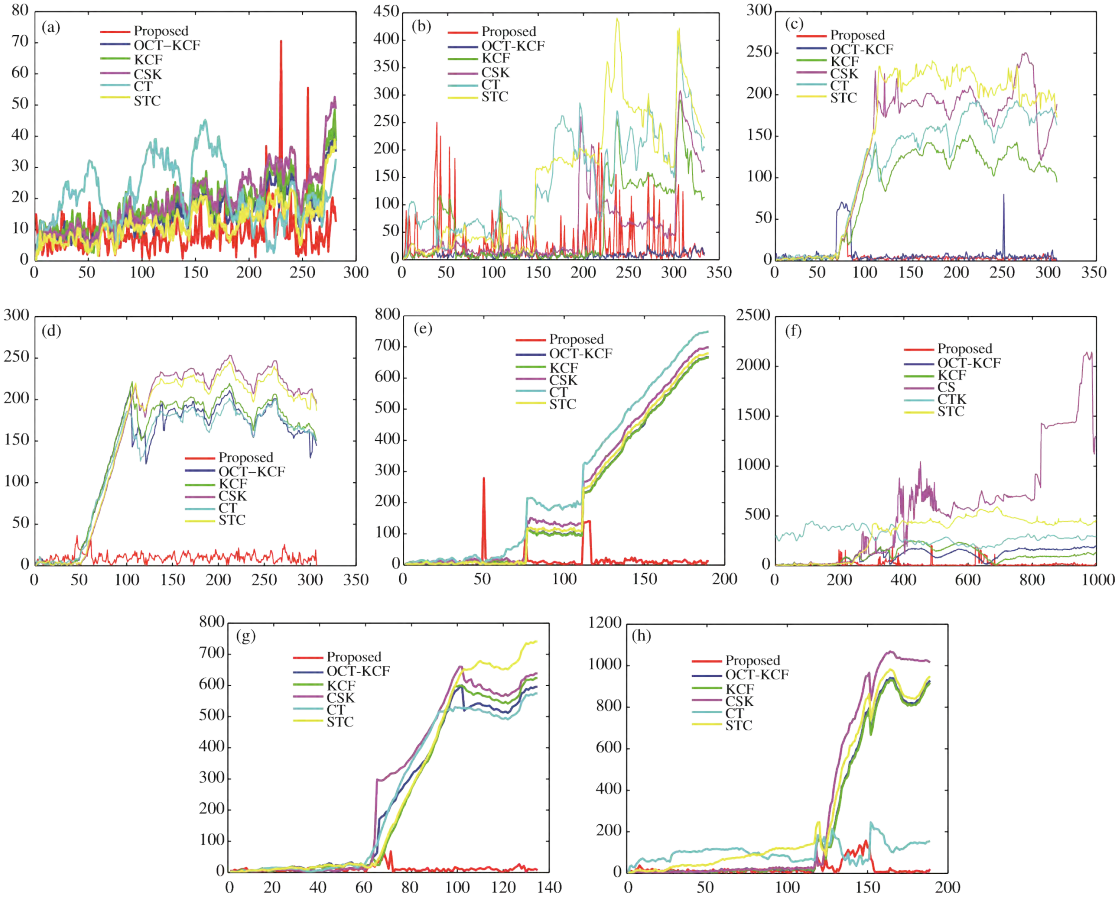

FIGURE 11. Comparison of the proposed frame and five existing tracking algorithms, based on CLE. Horizontal axis is the sequence number of the video frame. Vertical axis is central location error (CLE). (a) simple scene, (b) blurred body (no occlusion), (c) jogging 1 (occlusion), (d) jogging 2 (occlusion), (e) full occlusion, (f) partial occlusion, (g) pedestrian occlusion, (h) multi-pedestrian occlusion.

3) TARGET REPRESENTATION AND UPDATE STRATEGY

Moreover, in traditional trackers, the model set consists of a sequence of consecutive samples. This introduces large redundancies due to slow change in appearance, while previous aspects of the appearance are forgotten. This can cause over-fitting to recent samples. To overcome this shortcoming, we collect the target representation as a mixture of Gaussian components, where each component represents a different aspect of the target appearance, shown in Figure 10. This approach yields a compact yet diverse representation of the data, thereby reducing the risk of losing target, when the target appears after long-term occlusion.

IV. EXPERIMENTS

In this section, the tracking framework proposed in this paper is compared with state-of-the-art tracking algorithms. Our tracking target is a pedestrian who is partially or fully occluded by another object or person. Occluded pedestrian videos are not common in benchmarks. As such, some videos used in our experiment are from benchmark OTB-50 [33],

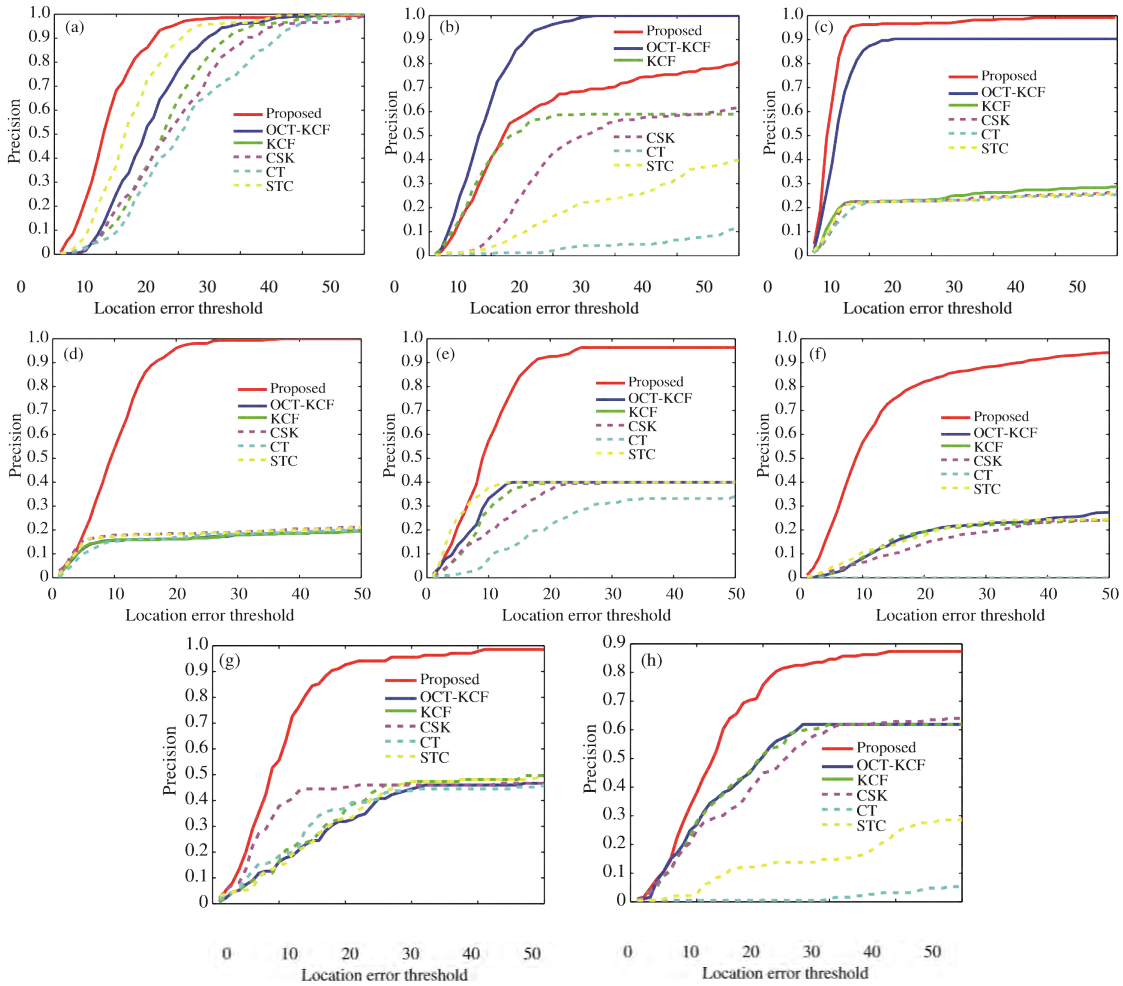

FIGURE 12. Precision plots for the 8 videos. The horizontal axis is the tracking center position offset threshold, in pixels. The vertical axis is the tracking accuracy. For the results, only pixel distance below 50 is tracked.

and some were recorded by us following the rules of this benchmark. These videos include typical occlusion situations, such as simple, unobstructed scenes, lens jitter, full occlusion, partial occlusion, occlusion by the crowd, and occlusion by a single pedestrian. The properties of the videos are shown in **Table 5**, and their serial numbers correspond to the sequences in Figures 11, 12, 13, and 15.

Our framework is tested on Intel 7 4.2GHz (4 cores) CPU, 16GB RAM and NVIDIA GTX 1080Ti GPU with memory of 11 GB.

In Figure 11, using the metric central location error (CLE), we compare our framework with five state-of-the-art tracking algorithms, namely OCT-KCF [12], KCF [11], CSK [23], CT [21], and STC [22]. We can observe that the CLE of these methods is almost the same in the simple scene, with a value around 50 pixels. The OCT-KCF algorithm [12] performs better in the case of (b), blurred body with no occlusion. Note that all five algorithms lose the target if it is blocked. In contrast, the proposed framework can always detect the target pedestrian after occlusion. For example, in the pedestrian occlusion scenario of Figure 11 (g), the target is blocked from frame 63 and the approach of this paper is able to redetect the target at frame 73, while other algorithms cannot. The reasons are as follows. First, Faster R-CNN can find all possible target candidates. Second, our model, combined with the FCN network, extracts the pedestrian information and excludes background noise. This is why our target can be found even after long-term occlusion, as shown in Figure 11 (e), i.e., the 5th video of Figure 11, where a pedestrian is fully occluded by a stone pillar with a width of 1.1m.

To fully analyze the performance of the algorithm, Figure 12 shows the precision of the proposed algorithm and other algorithms for tracking different sequences. It can be seen that this framework is similar to other algorithms in a simple scene without occlusion, but is clearly superior in scenes with occlusion. In the case of Figure 12 (b), lens jitter, the accuracy of the OCT-KCF algorithm is higher because the video pixels from the tracking benchmark are too low, which is not suitable for the SIFT feature matching module in our framework.

This paper adopts the success rate evaluation method proposed in [33]. Results are shown in Figure 13.

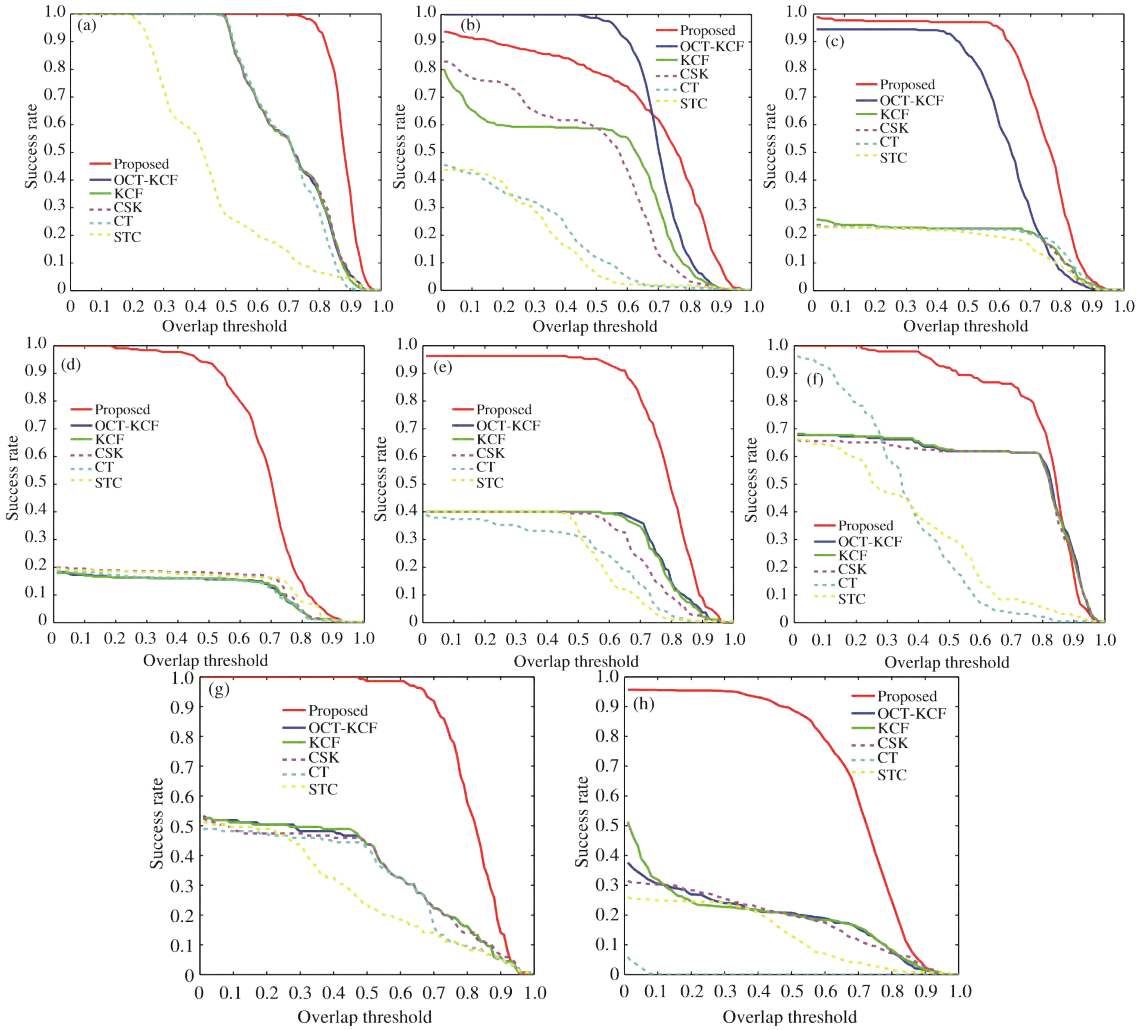

FIGURE 13. Success plots for the 8 videos. Horizontal axis is the intersection ratio S between the target rectangle and the ideal calibration rectangle. Vertical axis is the success rate of tracking, and the curve indicates that the crossings of all the detection targets are greater than the ratio of the different thresholds to the total number of frames.

TABLE 6. Speed comparisons with state-of-the-art trackers.

Speed	OUR	OCT-KCF [12]	KCF [11]	CSK [23]	STC [22]	CT [21]
FPS	29	53	132	430	410	400

The calculation method for the threshold S is formula (5) (R_i is the tracking rectangle, which represents the ideal rectangular box).The red line represents the algorithm in our paper. The crossings of the different algorithm curves and the ordinate show the ratio of the target to the whole sequence, and the area integral of the curve shows the overall success rate of the tracking algorithm. It can be seen from Figure 12 and Figure 13 that the proposed tracking framework has not only high tracking accuracy but also a better success rate than the other tracking algorithms. Moreover, the performance of our method is tested with 8 video sequences, the results are shown in **Table 6**. We can observe that our processing speed is about 29 FPS, which is slower than the other algorithms because our precision is always higher. However, the performance still meets the real-time requirement.

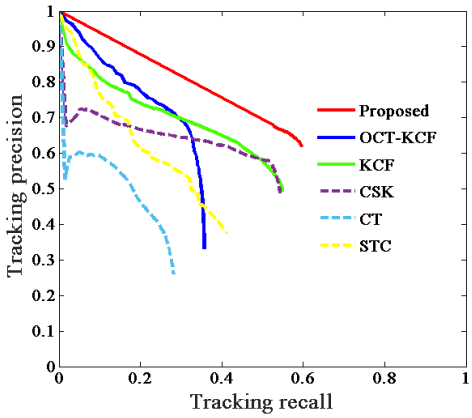

FIGURE 14. Long-term tracking performance. The average tracking precision-recall curves of VOT2018 long-term benchmark.

$$S = \frac{|R_i \cap R_0|}{|R_i \cup R_0|} \tag{5}$$

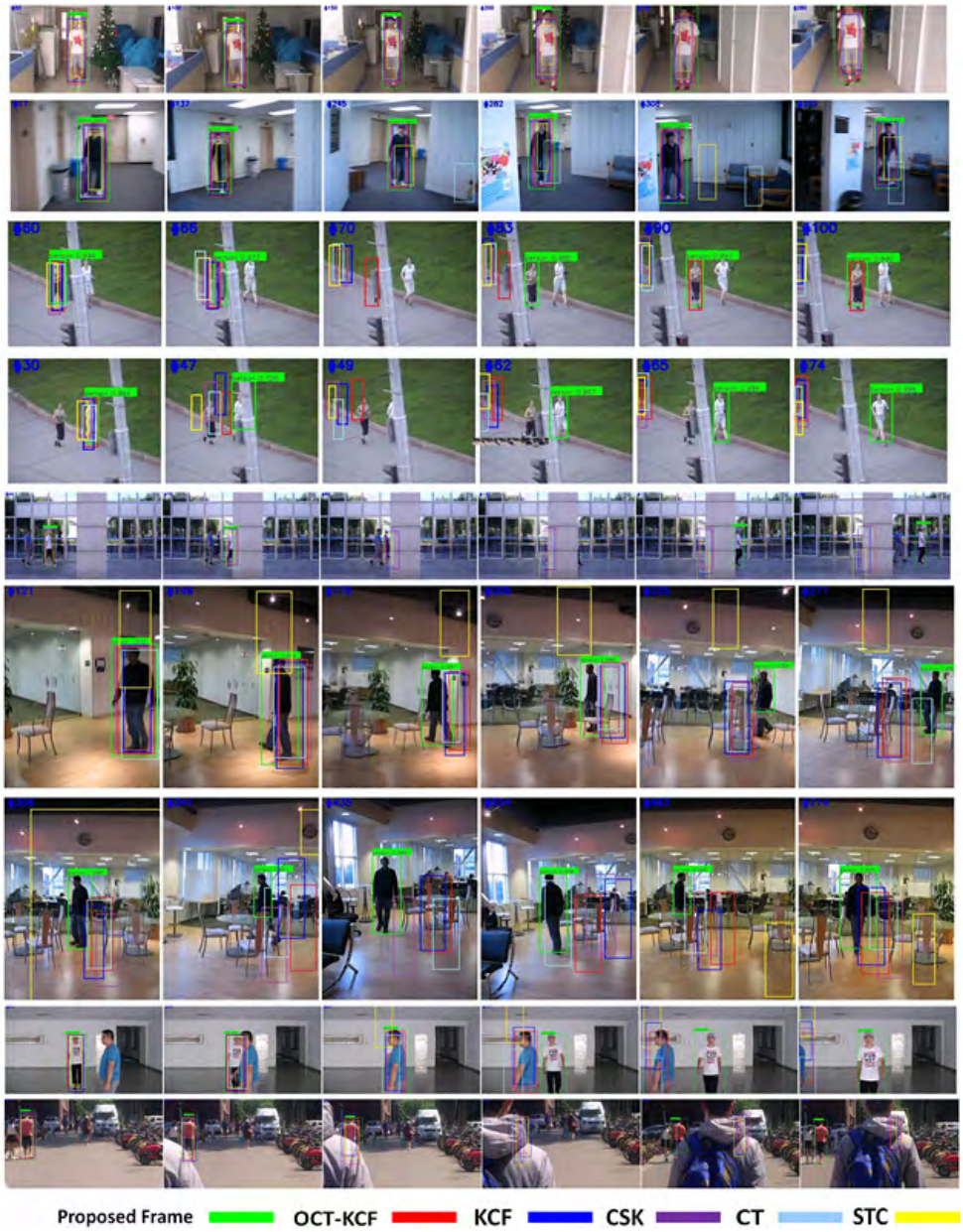

FIGURE 15. Illustration of some key frames.

Moreover, the proposed pedestrian tracking framework is tested with the latest VOT2018 long-term tracking benchmark [45]. This benchmark has a total of 20 pedestrian video sequences. The target of each sequence averagely disappears 12 times (including occlusion), and the target disappears every 40 frames. As results, the average accuracy and recall values of these 20 sequences are shown in Figure 14. It is observed that our framework outperforms other algorithms in terms of accuracy and recall.

We also show tracking results of key frames of 8 sequences shown in table 4 in Figure 15. The 1st row is a simple scene; all the tracking algorithms can achieve good results. The 2nd row is the situation with lens jitter or a blurred body, in which the OCT-KCF tracking effect is better because of the very low resolution of the video. The 3rd to 8th rows are occlusion situations, in which all existing trackers failed to detect the blocked targets but the proposed tracking framework can ensure the occluded target will not be missed after occlusion.

V. CONCLUSION

In this paper, the performance and accuracy of occluded pedestrian tracking are jointly considered, combining RGB histograms with SIFT for target representation. Also, a new method of calculating the similarity of RGB histograms is proposed. To enhance tracking accuracy, the matching threshold for SIFT is calculated. To further enable accurate SIFT

feature matching, the full convolutional network method is utilized to implement image semantic segmentation in background denoising. Multiple experiments were carried out to compare our framework with five state-of-the-art tracking approaches. The results of benchmark OTB-50 and VOT2018 long-term show that our framework achieves better performance than most of the existing trackers and is highly efficient in dealing with high-resolution images. On the other hand, one shortcoming of the proposed framework is that its accuracy is lowered when dealing with low-resolution images with fewer SIFT features. To achieve higher tracking performance, future work might focus on finding a new target representation method. In addition, we need to improve the framework of this paper based on other advanced deep learning methods to address long-term tracking problems.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surveys*, vol. 38, no. 4, pp. 81–93, 2006.

[2] X. Song, H. Xie, J. Sun, D. Han, Y. Cui, and B. Chen, "Simulation of pedestrian rotation dynamics near crowded exits," *IEEE Trans. Intell. Transp. Syst.*, to be published. doi: 10.1109/TITS.2018.2873118.

[3] Y. Wang, J. Wang, X. Song, and L. Han, "An efficient adaptive fuzzy switching weighted mean filter for salt-and-pepper noise removal," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1582–1586, Nov. 2016.

[4] X. Song *et al.*, "Supporting real-world network-oriented mesoscopic traffic simulation on GPU," *Simul. Model. Pract. Theory*, vol. 74, no. 3, pp. 46–63, 2017.

[5] X. Song, L. Ma, Y. Ma, C. Yang, and H. Ji, "Selfishness- and selflessness-based models of pedestrian room evacuation," *Phys. A, Stat. Mech. Appl.*, vol. 447, no. 4, pp. 455–466, Apr. 2016.

[6] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling L1 tracker with occlusion detection," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2661–2675, Jul. 2013.

[7] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 213–228, 2015.

[8] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 1910–1917.

[9] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. CVPR*, Jun. 2010, pp. 2544–2550.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[12] K. Zhang *et al.*, "Output constraint transfer for kernelized correlation filter in tracking," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 4, pp. 693–703, Apr. 2017.

[13] D. Comaniciu and P. Meer "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 142–149.

[15] X. B. Jin, J. J. Du, and J. Bao, "Data-driven tracking based on Kalman filter," *Appl. Mech. Mater.*, vols. 226–228, pp. 2476–2479, Nov. 2012.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[18] K. Chen, S. G. Demko, and R. Xie, "Similarity-based retrieval of images using color histograms," *Proc. SPIE*, vol. 3656, pp. 643–652, Dec. 1998.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[21] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. ECCV*, 2012, pp. 864–877.

[22] K. Zhang *et al.*, "Fast tracking via spatio-temporal context learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2013.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, 2012, pp. 702–715.

[24] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[26] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. Hengel, "A survey of appearance models in visual object tracking," *Trans. Intell. Syst. Technol.*, vol. 4, no. 4, p. 58, 2013.

[27] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 1–10.

[28] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[29] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

[30] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[31] N. Alt, S. Hinterstoisser, and N. Navab, "Rapid selection of reliable templates for visual tracking," in *Proc. CVPR*, Jun. 2010, pp. 1355–1362.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[33] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.

[34] Y. Jia *et al.* (Jun. 2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: https://arxiv.org/abs/1408.5093

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

[36] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.

[37] R. M. Luque-Baena, J. M. Ortiz-de-Lazcano-Lobato, E. López-Rubio, E. Domínguez, and E. J. Palomo, "A competitive neural network for multiple object tracking in video sequence analysis," *Neural Process. Lett.*, vol. 37, no. 1, pp. 47–67, 2013.

[38] M. Mentzelopoulos, A. Psarrou, A. Angelopoulou, and J. García-Rodríguez, "Active foreground region extraction and tracking for sports video annotation," *Neural Process. Lett.*, vol. 37, no. 1, pp. 33–46, 2013.

[39] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 798–805.

[40] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, "Visual tracking via discriminative sparse similarity map," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1872–1881, Apr. 2014.

[41] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2363–2370.

[42] *Vlfeat*. Accessed: Feb. 21, 2019. [Online]. Available: https://github.com/menpo/cyvlfeat

[43] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[45] 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.

[46] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

KAI CHEN was born in Shanxi, China, in 1991. He currently pursuing the Ph.D. degree with Beihang university, Beijing, China. His research interests include object tracking and machine, and deep learning.

XIAO SONG received the Ph.D. degree in electrical engineering from Beihang University, Beijing, China, in 2006, where he is currently an Associate Professor with the Automation School. His research interests include pedestrian modeling and simulation, and cloud manufacturing.

XIANG ZHAI received the master's degree from the China Institute of Software, Chinese Academy of Sciences. He is currently an Engineer with the China State Key Laboratory of Intelligent Manufacturing System Technology. His main research interests include intelligent manufacturing and complex system simulation.

BAOCHANG ZHANG received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of the Technology, Harbin, China, in 1999, 2001, and 2006, respectively. He is currently an Associate Professor with the School of Automation, Beihang University, Beijing, China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.

BAOCUN HOU received the Ph.D. degree from Beihang University, in 2006. He is currently the Senior Manager of Beijing Aerospace Smart Manufacturing Technology Development, Co., Ltd. His main research interests include cloud computing, advanced manufacturing, and distributed simulation.

YI WANG received the Ph.D. degree from Beihang University, in 2017, where he currently holds a Postdoctoral position. His main research interests include computer simulation, image processing, and pattern recognition.