Efficient Human-In-The-Loop Object Detection using Bi-Directional Deep SORT and Annotation-Free Segment Identification

Koki Madono* Teppei Nakano* Tetsunori Kobayashi* and Tetsuji Ogawa*
* Department of Communications and Computer Engineering, Waseda University
† Intelligent Framework Lab

***Abstract*—The present study proposes a method for detecting objects with a high recall rate for human-supported video annotation. In recent years, automatic annotation techniques such as object detection and tracking have become more powerful; however, detection and tracking of occluded objects, small objects, and blurred objects are still difficult. In order to annotate such objects, manual annotation is inevitably required. For this reason, we envision a human-supported video annotation framework in which over-detected objects (i.e., false positives) are allowed to minimize oversight (i.e., false negatives) in automatic annotation and then the over-detected objects are removed manually. This study attempts to achieve human-in-the-loop object detection with an emphasis on suppressing the oversight for the former stage of processing in the aforementioned annotation framework: bi-directional deep SORT is proposed to reliably capture missed objects and annotation-free segment identification (AFSID) is proposed to identify video frames in which manual annotation is not required. These methods are reinforced each other, yielding an increase in the detection rate while reducing the burden of human intervention. Experimental comparisons using a pedestrian video dataset demonstrated that bi-directional deep SORT with AFSID was successful in capturing object candidates with a higher recall rate over the existing deep SORT while reducing the cost of manpower compared to manual annotation at regular intervals.**

I. Introduction

In recent years, many researchers have shown interest in building video datasets for the development of object tracking systems, an elemental technology for self-driving cars, video surveillance systems and mobile robots [1], [2], [3], [4], [5], [6]. In such datasets, the trajectories of all objects in the video should be accurately and efficiently annotated. In this case, manual annotations, such as human verification and crowdsourcing, provide high accuracy, but are time-consuming and expensive. In contrast, automatic annotations based on object detection and tracking techniques, while efficient and economical, are generally not sufficiently accurate.

Human-supported or semi-automatic video annotation, which integrates these two approaches, is a good strategy to make annotation more efficient and economical. Several attempts have been made for reducing human intervention while maintaining high accuracy. For example, VIPER-GT and LabelMe [7], [8] manually annotated a small number of video frames and used the results to automatically detect the remaining objects. VATIC [9] exploited crowdsourcing on Amazon Mechanical Turk (AMT) [10] for manual annotation, and its extension [11] used active learning to select video frames to be manually annotated [12], [13]. PathTrack [14] also used crowdsourcing with active learning. Here, pre-labeling using automatic object detection reduced the time for manual annotation. These studies suggest that a combination of automatic annotation may contribute to lower the human annotation cost. It should be noted here that some objects, such as occluded objects, small objects, and blurred objects, are difficult to detect automatically, and such objects need to be annotated manually. This argument, therefore, suggests that it makes good sense to identify the areas that should be annotated manually while automatically annotating those that are not.

To this end, the present study proposes an efficient human-in-the-loop object detection method. We assume a human-supported video annotation framework in which object regions are detected without missing even if we allow for over-detection, and then the over-detected regions are removed manually. The proposed method corresponds to the previous stage of processing in this annotation framework and emphasizes the reduction in missed objects. The key to the success of minimizing the missed objects is accurate detection of occluded or small objects. An attempt, therefore, is made to estimate such object regions by tracking them from the objects detected in the previous and subsequent frames. For this purpose, the unidirectional object tracking technique, deep SORT [15], is extended to enable bi-directional tracking: backward tracking focuses on increasing the number of object candidates, while forward tracking focuses on accuracy, aiming to reduce misses without unduly increasing overdetection. For those objects that are still difficult to detect, manual annotation is inevitable. We, therefore, attempt to identify frames that contain objects to be annotated manually using a binary search. By designing both methods to reinforce each other, the proposed method aims to reduce the cost of human intervention while also reducing the number of oversights. Experimental comparisons using a pedestrian video dataset is carried out to demonstrate the effectiveness of the proposed human-in-the-loop object detection method in terms of the recall rate of detecting objects and the cost for manual annotation.

The rest of the present paper is organized as follows. Section II briefly explains an expected application, a human-supported video annotation framework. Section III proposes the effi-

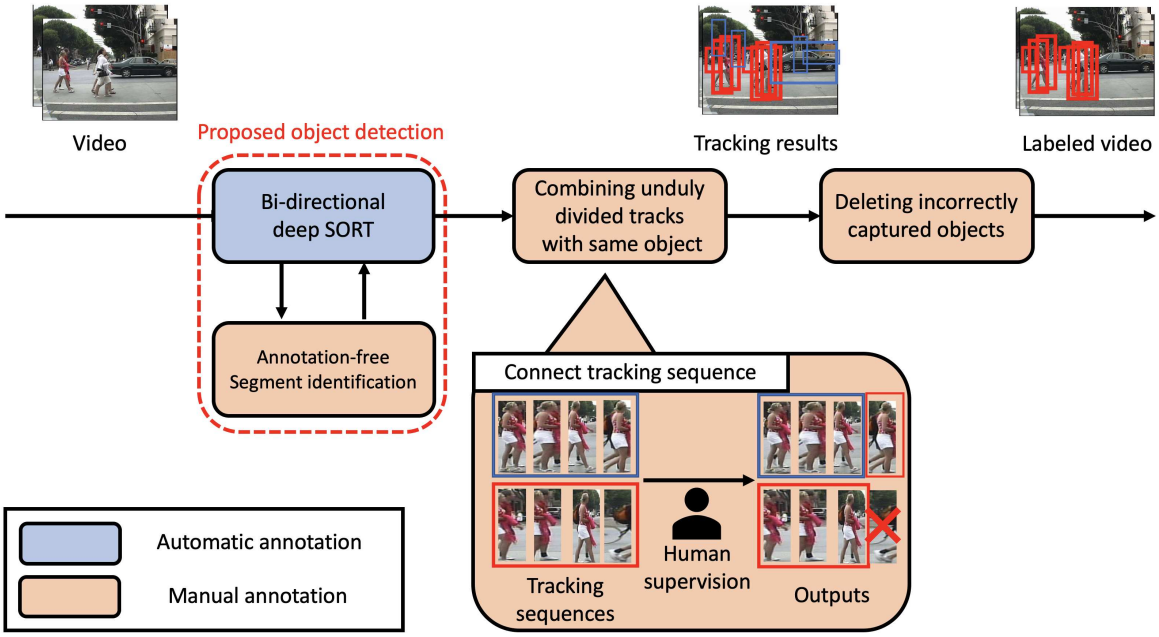

Fig. 1. Assumed human-supported video annotation framework for moving objects: 1) proposed efficient human-in-the-loop object detection using bi-directional deep SORT with annotation-free segment identification (AFSID); 2) manual combination of unduly divided tracks with same object; and 3) manual deletion of incorrectly captured objects.

cient human-in-the-loop object detection method. Section IV examines the performance of the developed object detection systems. Section V provides some concluding remarks.

II. Object Detection for Human-Supported Video Annotation Framework

This section clarifies the focus of this study and the assumed application. This study is intended to contribute to a human-supported (or semi-automatic) video annotation framework, which aims at effective integration of automatic and manual annotation to give high efficiency and accuracy. Figure 1 illustrates the pipeline of the assumed human-supported video annotation framework. It is composed of the following three stages of processing:

- **Stage 1**: Proposed efficient human-in-the-loop object detection using bi-directional deep SORT and annotation-free segment identification (AFSID)
- **Stage 2**: Manual combination of unduly divided tracks with the same object
- **Stage 3**: Manual deletion of incorrectly captured objects.

In this framework, it is important to avoid miss detection even at a sacrifice of yielding a number of over-detected (or false-positive) candidates in the first stage because manually deleting redundant object candidates might be easier than drawing bounding boxes to the missed objects. The present study, therefore, focuses on the first stage of the framework and aims at efficient object detection with a high recall rate. Although the first-stage processing of the framework does not check the identity of objects in the track, nor does it delete the false positive of object candidates, these processes can be performed by human verification in the subsequent two stages.

III. Human-In-The-Loop Object Detection

This section presents the proposed efficient and high recall object detection method designed as a human-in-the-loop architecture: bi-directional deep SORT is introduced with the aim of minimizing miss detection without unduly increasing overdetection, while identifying annotation-free segments, in which manual annotation is not required, to improve the efficiency in the annotation. The following subsections describe the details of the proposed method: Section III-A briefly explains deep SORT [15] that is the basis of this study, Section III-B describes an overview of the proposed object detection architecture, Section III-C describes the algorithm of AFSID, and Section III-D describes the algorithm of object tracking in bi-directional deep SORT.

A. Deep SORT and Its Drawbacks

This section provides an overview of deep SORT, which is the basis of this study, and discuss its shortcomings. Deep SORT exploits Kalman filter with deep appearance descriptors for reliable tracking of a sequence of object candidates. Here, a track begins with an object candidate region, which is the output of the object detector. Kalman filtering estimates the candidate regions in the following frames with the aim of capturing the missing objects, which are difficult to capture, such as highly occluded, small, or blurred objects. Deep SORT has shown to be robust against a long period of occlusions and

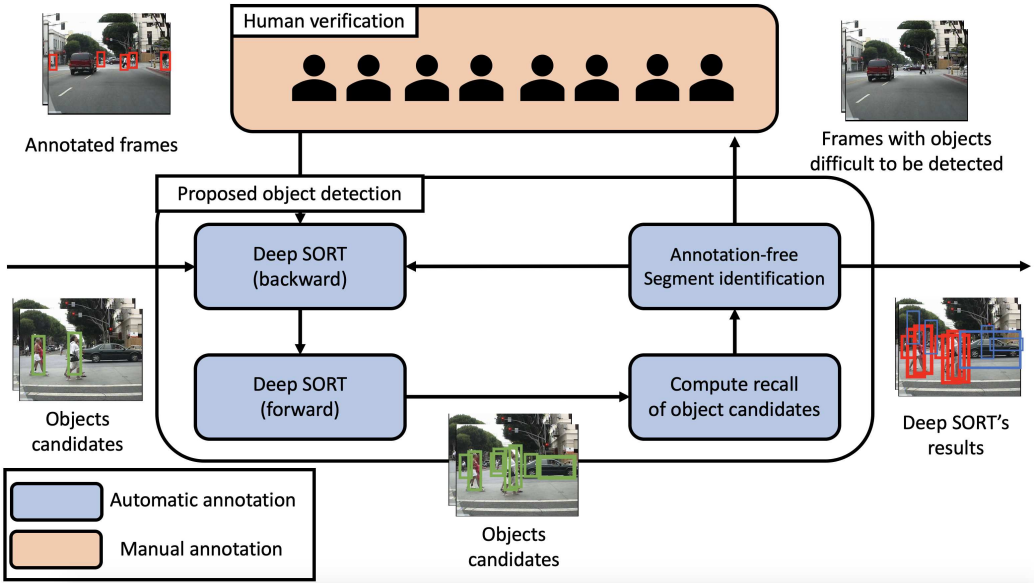

Fig. 2. Schematic diagram of proposed human-in-the-loop object detection. Bi-directional deep SORT employs (uni-directional) deep SORT [15] in not only forward but also backward directions on time axis to increase the recall rate of detecting objects. Annotation-free segment identification (AFSID) estimates time segments (i.e., consecutive frames) that do not require costly manual annotation. Human verification is performed for time intervals estimated by AFSID to require manual annotation.

effectively reduce the number of identity switches. The association measure and algorithm for object tracking is described in detail in Section III-D.

Note that the purpose of this study is to achieve object detection that avoids missed objects as much as possible. Estimation of object candidate regions by Kalman filtering helps to reduce such missed objects because it captures objects that are difficult to detect. Unidirectional tracking, however, has its limitations. Suppose an object that is occluded temporarily but long enough. If the object is tracked only in the forward direction, subsequent tracking may fail, and detecting the object just before it is no longer obstructed is unlikely. The object in such a situation, however, can be captured by tracking it from behind. Such tracking in both forward and backward directions, therefore, can contribute to a reduction in the missed objects. This argument suggests the effectiveness of our proposal, bi-directional deep SORT. Some object candidates, however, are still difficult to detect automatically with deep SORT. In order to treat such objects, human-powered annotation is inevitable. Further reduction in the missed objects can be expected by utilizing manually annotated object regions with deep SORT. For the efficient implementation of the aforementioned framework, a human-in-the-loop architecture is proposed.

B. Proposed Object Detection

1) Overview: Figure 2 illustrates the schematic diagram of the human-in-the-loop object detection architecture proposed to yield the high recall rate and less missed objects. Algorithm 1 lists its algorithm in detail. This algorithm mainly consists of three processing components as follows:

- **Backward deep SORT:** Objects are tracked using deep SORT in the backward direction. Here, the manually-annotated object regions as well as the outputs of the mask RCNN [16] are used as start points for tracking. Not only those regions, but also the regions estimated by backward tracking are used as starting points for tracking.
- **Forward deep SORT:** Objects are tracked using deep SORT in the forward direction. Not only the manually-annotated regions and outputs of the mask RCNN, but also the object candidate regions obtained during backward deep SORT are used as starting points for tracking. Unlike backward deep SORT, the regions estimated in a forward tracking are not used as the starting points.
- **Annotation-free segment identification (AFSID):** Annotation-free segments, i.e., consecutive frames in which manual annotation is not required, are identified by computing an intersection over unions (IoU) between the object candidate and the ground truth.

The purpose of bi-directional deep SORT is to reduce misses without unduly increasing overdetection. To that end, backward tracking focuses on increasing the number of object candidates in order to reduce missed objects. Subsequent forward tracking takes into account the accuracy of tracking. Therefore, the estimated results from tracking, which are sometimes unreliable, will be used as starting points during backward tracking, but not during a forward tracking. For the purpose of avoiding miss detection as much as possible, manual annotation is inevitable. Nevertheless, human intervention is costly and should be kept to a minimum. Annotation-free

Algorithm 1 Proposed Object Detection

Input: Object candidate regions $\mathcal{D}$, manually-annotated object regions $\mathcal{A}$, annotation-free segment indices $\mathcal{F} = \phi$, initial value of time interval for manual annotation $N = 32$, terminated value of time interval E

1: Give manual annotations to $\mathcal{A}$ every N frames
2: **while** $N > E$ **do**
3: Update $\mathcal{D}$ by backward deep SORT (Alg. 3)
4: Update $\mathcal{D}$ by forward deep SORT (Alg. 3)
5: Halve time interval N
6: Update $\mathcal{F}$ and $\mathcal{A}$ by AFSID (Alg. 2)
7: **return**

Algorithm 2 AFSID

Input: Object candidate regions $\mathcal{D}$, manually-annotated object regions $\mathcal{A}$, annotation-free segment indices $\mathcal{F}$, time interval for manual annotation N

1: $t \leftarrow StartFrame$
2: **while** $t \leq LastFrame$ **do**
3: **if** $t \notin \mathcal{F}$ **then**
4: $\mathcal{A} \leftarrow \mathcal{A}$ + manual annotation at frame t
5: Compute recall $R(t)$ using $\mathcal{D}$(t) and $\mathcal{A}$(t)
6: $\mathcal{F} \leftarrow \mathcal{F} + \mathrm{AFF}(\mathcal{F}, t, N, R(t))$
7: $t \leftarrow t + N$
8: **return** $\mathcal{F}$, $\mathcal{A}$

Algorithm 3 Forward or backward deep SORT

Input: Object candidate regions $\mathcal{D}$, Manually-annotated object regions $\mathcal{A}$

1: **for** $t \in$ {StartFrame, ...,LastFrame} **do**
2: **if** $A(t)$ is not empty **then** $D(t) = \mathcal{A}(t)$
3: Compute association cost matrix $C = [c_{i,j}]$ using Eq. 8
4: Compute association gate matrix $B = [b_{i,j}]$ using Eq. 9
5: Initialize set of matches $\mathcal{M} \leftarrow \phi$
6: Initialize set of unmatched detections $\mathcal{U} \leftarrow D(t)$
7: **for** $n \in \{1, ..., A_{\max}\}$ **do**
8: Select tracks by age $\mathcal{T}_{\mathrm{n}} \leftarrow \{i \in \mathcal{T} | a_i = n\}$
9: $[x_{i,j}] \leftarrow min\_cost\_matching(C, \mathcal{T}_n, \mathcal{U})$
10: $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
11: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \cup \sum_i \{b_{i,j} \cdot x_{i,j} > 0\}$
12: **end for**
13: $\mathcal{M}, \mathcal{U} \leftarrow IoUMatching(\mathcal{M}, \mathcal{U}, \mathcal{T})$
14: **if** $\mathcal{A}(t)$ is not empty **then** $\mathcal{T} \leftarrow \mathcal{T} \setminus \{j \mid j(update) > 0\}$
15: **else** $\mathcal{T} \leftarrow \mathcal{T} \setminus \{j \mid j(update) > A_{\max}\}$
16: $\mathcal{T} \leftarrow$UpdateState$(\mathcal{T}, \mathcal{M})$
17: **if** Backward deep SORT and $\mathcal{A}(t)$ is empty **then**
18: $\mathcal{D}(t) \leftarrow \mathcal{D}(t) \cup \{j \mid j \in An\,estimated\,region\,by\,backward\,tracking\}$
19: **if** Forward deep SORT **then**
20: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid j \in An\,estimated\,region\,by\,forward\,tracking\}$
21: Initialize $\mathcal{U}$ as Track
22: **end for**

segment identification is introduced for this purpose.

2) Algorithm: The algorithm of the proposed object detection is shown in Alg. 1. A binary search [17] is used to efficiently determine the annotation-free segments by halving the time intervals for manual annotation. The iteration begins at a time interval of N and stops when the time interval reaches E. The initial value for N is empirically determined to be 32.

First, object regions are manually annotated every 32 frames. Such manually-obtained object regions as well as the object candidates detected by the mask RCNN are used as starting points for backward tracking (line 3 in Alg. 1). Using the object regions obtained in this step, forward tracking is then performed to predict further object candidates (line 4 in Alg. 1).

Here, time interval N is halved in a binary search manner i.e., annotations are manually given every 16 frames (line 5 in Alg. 1). After that, annotation-free segments are identified to reduce the cost for manual annotation, and then the annotation-free segment indices $\mathcal{F}$ and manually-annotated objects $\mathcal{A}$ are updated (line 7 in Alg. 1). The segments with $\mathcal{F}$ are unnecessary to manually annotate from the next iteration.

C. Annotation-Free Segment Identification (AFSID)

This section describes the details of annotation-free segment identification (AFSID). In a certain video frame, if the object regions annotated by humans match the object candidate regions detected automatically, it implies that the objects in such a frame are easy to detect and such a frame does not need to be manually annotated thereafter. In addition, if a pair of such annotation-free frames has the small time interval N, the objects in all frames between that frame pair $t-N$ and t are also considered easy to detect and those frames $t - N, \cdots, t$ are determined to be the annotation-free segment. The results of preliminary experiments showed that this assumption is considered to be satisfied if N is less than or equal to 32. This is the reason why the time interval for manual annotation N begins with 32 frames and then repeatedly decreases by half.

1) Decision function: The frames that do not require manual annotation can be identified on the basis of the recall rate $R(t)$ of the object candidate regions $\{d_t^{(j)}\} \in \mathcal{D}(t)$ detected using bi-directional deep SORT for the correct object regions $\{a_t^{(i)}\} \in \mathcal{A}(t)$, which are given manually. The recall rate is written as:

$$R(t) = \frac{\sum_i \mathbb{1}[\max_j \mathrm{IoU}(d_t^{(j)}, a_t^{(i)}) \geq Th^{(1)}]}{\#_{\mathcal{A}}(t)}, \quad (1)$$

where $\#_{\mathcal{A}}(t)$ denotes the number of correct object regions at time t, $\mathrm{IoU}(d_t^{(j)}, a_t^{(i)})$ denotes the IoU between the object candidate region $d_t^{(j)}$ and the correct region $a_t^{(i)}$ to give

$$\mathrm{IoU}(d_t^{(j)}, a_t^{(i)}) = \frac{|d_t^{(j)} \cap a_t^{(i)}|}{|d_t^{(j)} \cup a_t^{(i)}|}, \quad (2)$$

and $Th^{(1)}$ denotes the IoU threshold, which is empirically determined to be 90%. Using the result in Eq. (1), annotation-free frames (AFFs) are given as:

$$\mathrm{AFF}(\mathcal{F}, t, N, R(t)) = \begin{cases} [t-N, t], & (R(t)=100\% \ \ \& \ \ t-N \in \mathcal{F}) \\ t, & (R(t)=100\%) \\ \mathrm{None}, & (\mathrm{otherwise}). \end{cases} \quad (3)$$

If the recall at frame t is 100%, the frame t is added to the annotation-free segment indices $\mathcal{F}$. In addition to that, if the previous frame $t-N$ has already been included in $\mathcal{F}$, the previous $N+1$ frames $t-N, t-N+1, \cdots, t$ are added to $\mathcal{F}$.

2) AFSID algorithm: The algorithm of AFSID is shown in Alg. 2. The iteration index t begins with the first frame of the video and ends when t reaches its last frame (lines 1 and 2 in Alg. 2). At every N frames (line 7 in Alg. 2), it is checked if the frame does not need to be manually annotated i.e., annotation-free.

If the target frame is not included in $\mathcal{F}$, manual annotation is performed and the results are added to the set of manually-annotated object regions $\mathcal{A}$ (lines 3 and 4 in Alg. 2).

The recall $R(t)$ between the object region $d_t^{(j)}$ estimated using bi-directional deep SORT and the correct region given manually $a_t^{(i)}$ is calculated using Eq. (1) (line 5 in Alg. 2). Using $R(t)$ and $\mathcal{F}$, the annotation-free frames are determined using Eq. (3) and added to $\mathcal{F}$ (line 6 in Alg. 2).

D. Object Tracking in Bi-directional Deep SORT

This section describes the details of object tracking, specifically focusing on the objective function and algorithm for object matching. The present study refers to the object matching algorithm used in deep SORT and extends it for bi-directional tracking.

1) Objective function for object matching: For object tracking, Kalman filtering is employed to robustly predict the location of the missed object against the occlusion problem. Here, the tracking scenario is defined in the eight-dimensional state space (u, v, γ, h, $\dot{x}$, $\dot{v}$, $\dot{\gamma}$, $\dot{h}$), where (u, v) denotes the center position of a bounding box, γ denotes the aspect ratio of a bounding box, h denotes the height of a bounding box, and four other variables express their respective velocities. If the object is missing less than or equal to $A_{\max}$ frames, tracking the object continues, and otherwise, the tracking is terminated. In addition, if an object in a track does not match manually-annotated object regions, the corresponding track is deleted.

When a Kalman filter is employed under the assumption of a constant velocity motion and a linear observation model, u, v, γ, and h are considered to be direct observations of the object state. The association cost between the predicted object state and a newly arrived object candidate region is given using the Mahalanobis distance [18] as

$$d^{(1)}(i,j) = (d_{t'}^{(j)} - y_t^{(i)})^{\mathrm{T}} {S_t^{(i)}}^{-1} (d_{t'}^{(j)} - y_t^{(i)}), \quad (4)$$

where ($y_t^{(i)}$, $S_t^{(i)}$) denotes the mean and variance of the i-th tracking state at time t, and $d_{t'}^{(j)}$ denotes the j-th bounding box state at time t', where $t < t'$. Here, unlikely associations are excluded on the basis of a 95% confidence computed from the inverse χ^2 distribution [19]. The association function

$$b^{(1)}(i,j) = \mathbb{1}[d^{(1)}(i,j) \le Th^{(2)}] \quad (5)$$

yields a value of one if the association between the i-th track and j-th object candidate is admissible, where the Mahalanobis threshold $Th^{(2)}$ is empirically set to 9.4877.

For more reliable estimation, appearance descriptors are extracted using a pre-trained convolutional neural network (CNN), as in deep SORT [15]. Let $r_{t'}^{(j)}$ and $\hat{r}_t^{(i)}$ be an appearance descriptor for the j-th object candidate at time t' and that for the object region in the i-th track at time t, respectively, where $||r_{t'}^{(j)}|| = 1$. For each object candidate, the best matching track is chosen and the corresponding association cost is obtained by computing the cosine similarity between their appearance descriptors to give

$$d^{(2)}(i,j) = \min_{\hat{r}_t^{(i)}} \{1 - (r_{t'}^{(j)})^{\mathrm{T}} \hat{r}_t^{(i)}\}. \quad (6)$$

Here, for the object regions in the i-th track up to 100 previous frames, i.e., $\hat{r}_{t-100}^{(i)}, \cdots, \hat{r}_t^{(i)}$, the cosine similarity from the object region $r_{t'}^{(j)}$ is calculated and its minimum value is taken as the association cost $d^{(2)}(i,j)$. The association function

$$b^{(2)}(i,j) = \mathbb{1}[d^{(2)}(i,j) \le Th^{(3)}] \quad (7)$$

yields a value of one if the association between the i-th track and the j-th object candidate is admissible, where $Th^{(3)}$ is empirically determined to be 0.3.

Finally, the association costs described in Eqs. (4) and (6) are combined to give

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j). \quad (8)$$

Here, the association is admissible if both the aforementioned conditions represented by Eqs. (5) and (7) are satisfied as

$$b_{i,j} = \prod_{m=1}^{2} b^{(m)}(i,j). \quad (9)$$

In [15], the system with $\lambda = 0$ yields the best accuracy when the camera is not stable. This parameter is also used in the present experiment.

2) Object Matching Algorithm: Algorithm 3 shows the algorithm of matching objects across frames used to track objects in the forward and backward directions. Unidirectional deep SORT is extended as follows:

- Backward tracking (lines 17 and 18 in Alg. 3) is introduced, and its results are used as object candidate regions.
- Manual annotation is incorporated into the tracking algorithm (lines 2, 14 and 15 in Alg. 3).

First, the association cost matrix $\mathcal{C} = \{c_{i,j}\}$ is computed as in Eq. (8) and used for object matching (line 3 in Alg. 3). Then, the association gate matrix $\mathcal{B} = \{b_{i,j}\}$ is computed as in Eq. (9) and used for removing infeasible associations (line 4 in Alg. 3). Using these two matrices, tracks $\mathcal{T}$ are mathing with object candidates.

The matching of the tracks $\mathcal{T}$ and object candidates $\mathcal{D}(\sqcup)$ is solved as a linear assignment problem (i.e., minimum cost bipartite matching) using the cost matrix $\mathcal{C}$ (line 9 in Alg. 3). In such an assignment, the number of pairs of the tracks $\mathcal{T}$ and object candidates $\mathcal{D}(\sqcup)$ should be maximized to reduce missed objects as much as possible. In addition, the tracks $\mathcal{T}$ are matched with the object candidate $\mathcal{D}(\sqcup)$ in order from the most recently updated track (line 8 in Alg. 3). This is based on the assumption that the track that has been recently paired with the object candidate can be tracked stably without occlusion or other problems and are likely to be paired with the object candidate at the current time as well. The adequacy of the assignments are verified using the gate matrix $\mathcal{B}$ (lines 10 and 11 in Alg. 3).

In order to be able to track objects that suddenly change in their appearance (e.g., occluded, small, or blurred objects), the $\mathrm{IoU}(y_t^{(i)}, d_{t'}^{(j)})$ is computed for the unmatched pair of the track and object candidate (lines 13 in Alg. 3). The pair with the IoU more than the empirically determined threshold, 30%, is considered to be a matching pair. If the IoUs for multiple candidates exceed the threshold, the candidate with the highest one is selected and connected to the track. If no object candidate that can be connected to the track is found, or the missing time is longer than $A_{\max}$, tracking is terminated (line 15 in Alg. 3). Especially, if the frame t is manually annotated, then the result of no match is reliable, and the unmatched tracks are terminated at that frame (line 14 in Alg. 3). The direction of movement of the track $\mathcal{T}$ is adjusted when the track and the object candidate region are connected (line 16 in Alg. 3). Especially, if the newly detected object candidate matches the manually annotated object, the Kalman state (u, v, γ, h) is initialized with information about the annotated object region because it is more accurate than the automatically detected region.

The purpose of backward tracking is to increase the number of object candidates. The regions estimated at time t, therefore, are added to the object candidates $\mathcal{D}(t)$ (lines 17 and 18 in Alg. 3) and exploited as starting points for subsequent tracking. In contrast, forward tracking (lines 19 and 20 in Alg. 3) aims at accurate tracking and it, therefore, uses the object regions detected by the mask RCNN and manually annotated regions as starting points for tracking, and does not use the regions estimated by tracking as such. Finally, the set of unmatched object regions $\mathcal{U}$ is initialized as the start point for tracking (line 21 in Alg. 3).

IV. Pedestrian Detection Experiment

Experimental comparisons were carried out to demonstrate the effectiveness of the proposed bi-directional deep SORT with AFSID on pedestrian detection in terms of 1) the recall rate of detecting pedestrians, and 2) the cost for manual annotation. For the former point, the accuracy of bi-directional deep SORT and that of deep SORT were compared and for the latter, the efficiency of using AFSID over manual annotation at regular intervals was examined.

TABLE I
Number of pedestrians included in V000 and V007 of Caltech pedestrian dataset.

set	Reasonable	Partial	Heavy	Full	# of frame
V000	5149	587	992	2050	1842
V007	472	6	22	3	1842

A. Dataset

The Caltech pedestrian dataset [1] contains the realistic situations for annotation, such as frequent occlusions of pedestrians and blurred images. The object tracking performance was evaluated for several occlusion levels defined in the Caltech pedestrian dataset as follows:

- **Reasonable**: no occlusions in a bounding box
- **Partial**: less than 35% occlusions in a bounding box
- **Heavy**: 35% to 70% occlusions in a bounding box
- **Full**: more than 70% occlusions in a bounding box

In this experiment, V000 and V007 in set07 were selected for evaluation. V000, which has a large number of pedestrians and includes many occlusions, is suitable for evaluating difficult cases of pedestrian tracking, and V007, which has few pedestrians, is suitable for evaluating the situation easy to track pedestrians. The number of pedestrians included in these subsets is listed in Table I.

The Caltech pedestrian dataset has labels of pedestrians such as "person" and "people," which are exploited as the ground truth. In the present experiment, these two classes were not distinguished and the bounding boxes whose sizes were smaller than 50 pixels were ignored.

B. Experimental Setups

Experimental setups were the same as those in [15]. The wide residual network [20] was trained with the Mars dataset [21], which is designed for the person re-identification task, and used for appearance matching between the estimated track and subsequent object candidates. The object candidates were extracted by using the mask RCNN [16] in both the deep SORT and bi-directional deep SORT. The confidence score of these objects is more than 1%. Here, the mask RCNN[1] was trained using the MS COCO dataset [22]. This network was built using ResNet-101-FPN as a backbone network [23].

The object candidate with an IoU rate between the predicted region and ground truth of more than 50% was regarded as a positive example (i.e., pedestrian). The systems developed were evaluated with variously varying final values of time intervals for manual annotation (e.g., two, four, and eight frames) to evaluate the trade-off between the object detection accuracy and manual annotation cost. Here, the correct labels were given to the first and last frames for reliable tracking.

C. Experimental Results

Tables II and III list the recall rates for the "Reasonable," "Partial," "Heavy," and "Full" cases, the number of required

[1]The present experiment used the pre-trained model of the mask RCNN obtained from https://github.com/matterport/Mask_RCNN.

TABLE II
EXPERIMENTAL RESULTS OF RECALL RATE AND NUMBER OF ANNOTATIONS IN V000 DATASET. TIME INTERVAL EXPRESSES TERMINATED VALUE OF TIME INTERVAL FOR MANUAL ANNOTATION.

method	time interval	Reasonable	Partial	Heavy	Full	# annotations	FP rate	# box
Deep SORT	8	90.66%	84.67%	81.05%	82.83%	232	49.63%	14235
Deep SORT	4	94.76%	92.84%	93.75%	94.49%	462	35.9%	12084
Deep SORT	2	97.98%	99.15%	98.79%	96.63%	922	20.27%	10041
Bi-dir. Deep SORT	8	**95.59%**	**95.57%**	**85.08%**	**93.46%**	232	55.69%	17338
Bi-dir. Deep SORT	4	**98.41%**	**98.47%**	**95.97%**	**98.88%**	462	46.48%	15034
Bi-dir. Deep SORT	2	**99.46%**	**99.66%**	**99.19%**	**99.46%**	922	32.98%	12142
Bi-dir. Deep SORT + AFSID	8	95.46%	95.57%	84.98%	93.61%	**218**	56.55%	17670
Bi-dir. Deep SORT + AFSID	4	98.41%	98.47%	95.97%	98.88%	**415**	48.76%	15704
Bi-dir. Deep SORT + AFSID	2	99.48%	99.49%	99.40%	99.41%	**758**	38.32%	13206

TABLE III
EXPERIMENTAL RESULTS OF RECALL RATE AND NUMBER OF MANUAL ANNOTATIONS IN V007 DATASET. TIME INTERVAL EXPRESSES TERMINATED VALUE OF TIME INTERVAL FOR MANUAL ANNOTATION.

method	time interval	Reasonable	Partial	Heavy	Full	# annotations	FP rate	# box
Deep SORT	8	96.40%	66.67%	95.45%	33.33%	232	78.44%	2212
Deep SORT	4	99.15%	83.33%	90.90%	33.33%	462	64.57%	1380
Deep SORT	2	100%	100%	95.45%	66.67%	922	40.86%	837
Bi-dir. deep SORT	8	**96.40%**	**66.67%**	**100%**	**33.33%**	232	81.19%	2541
Bi-dir. deep SORT	4	**99.79%**	**100%**	**90.91%**	**33.33%**	462	71.19%	1708
Bi-dir. deep SORT	2	**100%**	**100%**	**95.45%**	**66.67%**	922	53.65%	1068
Bi-dir. deep SORT + AFSID	8	96.40%	66.67%	100%	33.33%	**145**	85.74%	3352
Bi-dir. deep SORT + AFSID	4	99.79%	100%	90.91%	33.33%	**200**	84.26%	3126
Bi-dir. deep SORT + AFSID	2	100%	100%	95.45%	66.67%	**307**	82.99%	2910

manual annotations, and false positive rates in V000 and V007, respectively. Note that manual annotation at regular time intervals was incorporated into deep SORT and bi-directional deep SORT and compared with bi-directional deep SORT with AFSID to demonstrate both the accuracy and efficiency of the proposed approach.

The results showed that proposed bi-directional deep SORT improved the recall rate over existing deep SORT. Bi-directional deep SORT sacrificed an increase in the false positive rates. The false positive candidates, however, can be deleted by the subsequent human verification in assumed human-supported video annotation framework as shown in Fig. 1. The present study, therefore, does not see it as a serious problem although there is room for improvement. In addition, the results of comparison between bi-directional deep SORT with and without AFSID suggest that AFSID efficiently reduced the number of manual annotations without the reduction in the recall rate.

Since the number of pedestrians in V007 is small (e.g., 6, 22, and 3), the comparison in the recall rate between V000 and V007 will not be discussed here. When the time interval for manual annotation was two frames, the reduction in human annotation by AFSID was about 18% in V000, which contains more pedestrians, and about 67% in V007, which contains fewer pedestrians. In V000, a reduction in recall with AFSID was observed, but only marginally. This suggests that the reduction in human annotation is more effective in datasets with fewer objects.

Figure 3 shows erroneous examples for analysis: the left images are the results of proposed bi-directional deep SORT, and the right images are the ground truth provided in the dataset. From the top-right image, it can be seen that some correct labels are too hard to capture due to their small sizes. The bottom-right image indicates that the shape of the bounding box is subtly incorrect. Since the shape of such a ground truth was different from the estimated shape of the object even though the developed system reasonably predict the object region, this evaluation can yield unduly decrease in recall rates.

V. CONCLUSION

The present study proposed a high-recall object detection approach based on object tracking and efficient human annotation. Specifically, bi-directional deep SORT was introduced to reduce the number of missed objects, and AFSID was incorporated to reduce the burden of manual annotation. Experimental comparisons using pedestrian detection demonstrated that the proposed method improved the recall rate by 11 % at most over deep SORT and reduced the number of manual annotations by 67 % at most over bi-directional deep SORT without AFSID.

In the present paper, bi-directional deep SORT was proposed and used as an object tracking method, but in future, it can be replaced by a more accurate tracking method tfor challenging video datasets.

o reduce the false positive rates for challenging video datasets.

REFERENCES

[1] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 743–761, 2012.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

Fig. 3. Analysis of failed examples in V000. Right images are ground truth and left images are corresponding results of bi-directional deep SORT with AFSID.

[3] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015.

[4] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler, "Mot16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016.

[5] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele, "Citypersons: A diverse dataset for pedestrian detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4457–4465, 2017.

[6] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z. Li, and Guodong Guo, "Widerperson: A diverse dataset for dense pedestrian detection in the wild," *ArXiv*, vol. abs/1909.12118, 2019.

[7] David Mihalcik and David Doermann, "The design and implementation of viper," 2005.

[8] Jenny Yuen, Bryan C. Russell, Ce Liu, and Antonio Torralba, "Labelme video: Building a video database with human annotations," *2009 IEEE 12th International Conference on Computer Vision*, pp. 1451–1458, 2009.

[9] Carl Vondrick, Deva Ramanan, and Donald J. Patterson, "Efficiently scaling up video annotation with crowdsourced marketplaces," in *ECCV 2010*, 2010.

[10] "Amazon mechanical turk," https://www.mturk.com/.

[11] Carl Vondrick and Deva Ramanan, "Video annotation and tracking with active learning," in *NIPS*, 2011.

[12] Michael Prince, "Does active learning work ? a review of the research," 2004.

[13] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge J. Belongie, "Visual recognition with humans in the loop," in *ECCV*, 2010.

[14] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool, "Pathtrack: Fast trajectory annotation with path supervision," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 290–299, 2017.

[15] Nicolai Wojke, Alex Bewley, and Dietrich Paulus, "Simple online and realtime tracking with a deep association metric," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, 2017.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[17] Louis F. Williams, "A modification to the half-interval search (binary search) method," in *ACM-SE 14*, 1976.

[18] R. De Maesschalck, D. Jouan-Rimbaud, and Desire L. Massart, "The mahalanobis distance," 2000.

[19] Henry Lancaster, "Chi-square distribution," 2005.

[20] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016.

[21] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian, "Mars: A video benchmark for large-scale person re-identification," in *ECCV*, 2016.

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context," *ArXiv*, vol. abs/1405.0312, 2014.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.