Part-MOT: A multi-object tracking method with instance part-based embedding

Xiaohu Liu | Yichuang Luo | Keding Yan | Jianfei Chen | Zhiyong Lei

1 School of Mechanical and Electrical Engineering, Xi’an Technological University, Xi’an, China
2 MeritData Technology Co., Ltd, Xi’an, China
3 School of Electronic and Information Engineering, Xi’an Technological University, Xi’an, China
4 Department of Electrical and Computer Engineering, Wayne State University, Detroit, Michigan, USA

Abstract
Part-MOT, a one-stage anchor-free architecture which unifies the object identification representation and detection in one task for visual object tracking is presented. For object representation, a position relevant feature is obtained using the center-ness information, which takes advantage of the anchor-free ideal to encode the feature map as the instance-aware embedding. To adapt to the object’s movement, the clustering-based method to get the global instance feature is introduced. This enables this approach more robust to make better tracking decisions. Part-MOT achieves the state-of-the-art performance on public datasets, with especially strong results for object deformation and movement changes.

1 | INTRODUCTION

Computer vision has made significant advances in image classification, instance segmentation, and object detection. Tracking, on the other hand, remains challenging, especially when multiple objects are involved. Recent results of tracking evaluations [1–3] show that bounding box-level tracking performance is saturating. Experiments [4] show that further improvements will be possible when using more information about pixel level.

The bounding box has been the dominant form for object annotation. One reason is that it is very convenient to annotate with little ambiguity, and the bounding box provides sufficient localization information for object detection. Another reason is that almost all image-feature extractors, both traditional [5] and deep-learning era [6, 7], are based on an input patch with a regular grid form. Moreover, it has been proved that bounding box representation facilitates the feature-extraction process [8, 9]. But for multi-object tracking, this annotation suffers from irregular shapes of different objects. In which, the bounding box annotation contains too much information from other objects; and make it hard to obtain the identified features. Also, the datasets that can be used to train and evaluate models for more precise representation, such as instance-wise level, usually do not provide annotations on video data or even information on object identities across different images. Taking this into consideration, dense boundary point prediction method is used to provide more accurate annotation.

More research attention has been paid to the one-shot MOT method, with the maturity of multitask learning [10–12] in deep learning. A single network is applied to simultaneously accomplish object detection and identity embedding. Due to sharing most of the computation, this network structure largely reduces the inference time. For example, a Re-ID branch is added in Track-RCNN [13] based on Mask-RCNN [14], in the meantime accomplishing the bounding box regression and classification.

However, due to many ID switches, one-shot methods usually have less accuracy than two-step methods. The reason is that the network does not learn the optimal features of objects because the embedding features are not aligned with the object centres, creating much ambiguity. To alleviate this problem, the anchor-free approach is used for object detection and identity...
YOLO [15], the popular one-stage anchor-based detector, predicts bounding boxes only at the points near the object center. The main idea is that the points near the center can provide more confidential detection. Because the number of center points is too small, YOLO suffers from low recall. While the FCOS [16] makes use of all points within the ground truth bounding box, and the “center-ness” branch is proposed to suppress the low-quality detection. So, a comparable recall is achieved compared with the anchor-based detectors.

Here, we propose an end-to-end trainable network, which takes detection, instance representation, and tracking as interconnected problems without instance-level annotations. Instead of detecting a rectangular bounding box, our detection localizes the parts of an object instance. More specifically, the aim of our detection is to predict a set of part-boundary points, which are more flexible for describing various shapes of objects. Using part-based boundary points has two advantages: (1) CNN features of irregular object regions can be accurately acquired with part-based boundary points, effectively eliminating the disturbance of background noise to the subsequent object representation and tracking; and (2) with part-based boundary points, an irregular shape can be easily transformed or rectified into a bounding box, which is a realistic annotation. Therefore, part-based boundary points appear to be a reasonable representation that can smoothly and effectively bridge representation and tracking modules.

Considering these factors, we model objects by boundary points which adaptively position themselves in spatial extent, to enable semantically aligned feature extraction. Usually, each object corresponds to multiple detection results, which come from different feature representations. More clearly, the features extracted in detection branch represent different parts of the object. The Part-MOT is proposed, a framework that formulates objects as part-based representation and extracts instance-aware embedding features for tracking association. The proposed approach is shown in Figure 1. First, we use a simple anchor-free object-detection approach to estimate the class confidence, the part feature center-ness and boundary points. Then we adopt a parallel branch for extracting the instance-aware features as the objects’ identities by considering the feature parts’ center-ness. In particular, the low-dimensional identity representation can reduce the computation time as well as improve the robustness of instance association. Furthermore, to deal with differently scaled objects, we use the deep layer aggregation operator [17] with the FCOS [16] structure. To improve the robustness of association, we employ a clustering method, which aggregate the object features to refresh instance representation.

This work makes three contributions: (1) we introduce a new method for object representation using part-based boundary points that adopts the FCOS for part feature extraction; (2) we use the center-ness of object parts to extract instance features for the Re-ID module, which aggregates the parts feature to a global representation by the center-ness confidence; and (3) to improve the robustness of tracking, the object feature is clustering according to the object moving type, which further enrich the representation.

2 | METHODOLOGY

This section proposes the whole tracking framework, Part-MOT, which takes the part-based representation and instance-aware embedding network for multi-object tracking. The network architecture consists of two components. First, the detection branch is designed for object classification, part center-ness prediction, and pseudo-bounding box. The center-ness confidence is utilized to aggregate part features to a global

![Diagram](image-url)
representation, and the pseudo-bounding box is generated by
$n$ boundary points. Second, the instance-embedding branch is
used for embedding feature extraction at instance level to iden-
tify different instances.

Based on the instance-aware embedding, we introduce a
tracking strategy that considers movement. We use not only
traditional IoU but also the embedding similarity to associate
objects. Further, we employ the clustering method for object
movement to improve the tracking robustness.

2.1 Backbone encoder–decoder network

Per-pixel-level object detection is used in the FOCS [16]
structure. Specifically, we adopt the ResNet-34 [17] as the network
backbone, which achieves a good balance between accuracy and
speed. Further, we use a variant of DLA [18] to tackle objects
with different scales, which is shown in Figure 1. Multi-level
features are extracted to resolve the overlapped bounding box
ambiguity. And the recall is improved, compared with the Feature
Pyramid Network (FPN) [19], the high-level feature takes
more information from low level in this structure. Formally, the
depth aggregation function $Agg_n$, with depth $n$, is formulated as:

$$
Agg_n(x) = Nd(Rt^{n-1}_m(x), Rt^{n-2}_m(x), \ldots, Rt^1_m(x), L^1_n(x), L^2_n(x)),
$$

(1)

where $x$ is layer, $Nd$ is the aggregation node, $Rt$ propagates
the aggregation of all previous blocks, $Lt$ merges aggregation nodes
of the same depth. The definition is:

$$
L^2_n(x) = B(L^1_n(x)),
L^1_n(x) = B(Rt^1_m(x)),
Rt^m_n(x) = \begin{cases} 
Agg_m(x) & \text{if} \ m = n-1 \\
Agg_m(Rt^m_{n+1}(x)) & \text{otherwise},
\end{cases}
$$

(2)

where $B$ represents a convolutional block. In addition, the bilin-
ear interpolation method is used in the upsampling module, to
alleviate the alignment problem.

2.2 Part-based detection

Let $F_i \in \mathbb{R}^{H_i \times W_i \times C}$ be the feature maps of a CNN layer. Each
location $(x, y)$ on $F_i$, is usually taken as the object center. While
we take them as a sample of the object, inspired by semantic seg-
mentation [20]. So, every location of feature map can map to a
part of the input image. Specifically, the part within the ground-
truth box is considered as positive sample. Thus, the proposed
network can leverage as many samples as possible to train every
branch.

We treat object detection as a dense regression task on a high-
resolution feature map. Particularly, we estimate the heatmaps,
object center-ness, and bounding boundaries by appending
three parallel branches to the backbone network. We use a $3 \times
3$ convolution (with 256 channels) on the output feature maps
of the backbone network in each branch, followed by a $1 \times 1$
convolutional layer.

2.2.1 Heatmap head

This head is used for estimating the probability, that the part
belongs to a certain class, and the Gaussian-based ground-truth
representation is used to compute the loss. In particular, the
dimensions of the heatmap are $1 \times H \times W$. When a location
collapses with the ground truth, the heatmap is expected to have
a higher response, and this feature is more informative for the
classification task. With the distance between the heatmap and
the object center, the response decays exponentially.

2.2.2 Center-ness head

To reduce the performance gap between FOCS and anchor-
based detectors, a single layer branch, in parallel with the clas-
sification branch (as shown in Figure 1) is added to predict the
center-ness of a feature location. So, the further from the object
center, the lower the score, which largely suppresses the low-
quality detections, as shown in Figure 2.

As defined in PolarMask [21], given a set $\{d_1, d_2, \ldots, d_n\}$ for
the length of $n$ rays of one instance, where $d_{\text{max}}$ and $d_{\text{min}}$ are
the maximum and minimum of the set, and the center-ness is
defined as:

$$
\text{Centerness} = \sqrt{\frac{\min \{d_1, d_2, \ldots, d_n\}}{\max \{d_1, d_2, \ldots, d_n\}}},
$$

(3)

As mentioned before, the different locations on the feature
map correspond to a part of the object. The center-ness can be
used as an effective strategy to combine part-based features so
that the closer $d_{\text{max}}$ and $d_{\text{min}}$ are, the higher weight the part is
assigned. The sqrt function is used to reduce the decay of the
center-ness, and the binary-cross-entropy (BCE) loss is used to
train the center-ness branch, which ranges from 0 to 1. When
testing, to filter out the low-quality bounding boxes, the classi-
fication score is used by multiplying the predicted center-ness,
and then the non-maximum suppression (NMS) is adopted,
which can remarkably improve the detection performance.

2.2.3 Regression head

We estimate the bounding box of each object with this regres-
sion branch. Like [22], four convolutional layers are added
on the feature maps of the backbone networks. Moreover, to
ensure the prediction is positive, the $\exp(\chi)$ is utilized on the top
of the regression branch to map any real number to $(0, \infty)$.

As discussed before, the bounding box does not account for
shape and semantically important local areas. Thus, for later
instance representation, the bounding box considers only the
rectangular spatial scope of an object, which is too coarse for
object tracking. To obtain finer object localization and better features, we use a set of adaptive points $R$ as the object boundary:

$$R = \{(x_k, y_k)\}_{k=1}^n,$$

(4)

where $n$ is the total number of sample points, which is set to 9 by default in experiments.

Due to the dense distance regression, the imbalance between the regression and classification should be taken into consideration. Intuitively, the $n$ rays consist of one instance boundary. The $n$ rays should be trained as a whole, rather than as a set of independent examples. Considering this relevance, the imbalance issue can be largely resolved.

To take advantage of bounding box annotations in the training process, as well as train and evaluate the boundary-based object detectors, the predefined converting function $T: R_p \rightarrow B_p$ is used to convert boundary points into a bounding box, where $T(R_p)$ represents a pseudo-box. Three converting functions [23] can be applied for this purpose:

1. $T = T_1$: Min–max function. A min–max operation is performed over the detection points over both axes to determine $B_p$, equivalent to the bounding box over the sample points.
2. $T = T_2$: Partial min–max function. A min–max operation is performed over a subset of the sample points over both axes to obtain the rectangular box $B_p$.
3. $T = T_3$: Nearest point. The matching target of each point in the point $T$ is defined as the nearest point in $S$ based on L1 distance, where $S$ is even sample points of the ground-truth bounding box, as shown in Figure 3.

2.3 | Part-based representation and instance-aware embedding

To generate features that can distinguish different objects, we propose the instance-aware embedding branch. Ideally, the representation of the same object should be more similar than that of different objects. The standard 2D convolution with kernel size of $k \times k$ samples features using a fixed regular grid $G = \{(-\frac{k}{2}, -\frac{k}{2}), ..., (\frac{k}{2}, \frac{k}{2})\}$, where $\lfloor \cdot \rfloor$ denotes the floor function. The regular grid $G$ cannot guarantee the sampled features of the object in corresponding regions. Therefore, we use the deformable convolution [23] to convert the sampling positions from the fixed region to the predicted region, and we extract the part feature with the feature-alignment module, which is formulated as:

$$f[a] = \sum_{g \in G, \Delta t \in T} w[g] \cdot x[a + g + \Delta t],$$

(5)

where $x$ indicates the feature map, $w$ denotes the learned convolution weight, $a$ indicates a location on the feature map, and $f$ represents the output-part feature map. Then, weighted by the center-ness score, we obtain the instance-aware feature, as shown in Figure 4.

Because the transformations of the sampling positions are adaptive to the variations of the object, the extracted object-aware feature is robust to the changes of object scale, which is beneficial for feature matching during tracking. Moreover, the instance-aware feature provides a global description of the candidate targets, which distinguishes the object from the background more reliably.

This idea is explained in Figure 5. Due to the deformable convolutions, the features on the boundary can be extracted to
be more instance informative, which can be seen in the middle, and the boundary sample region is adaptive to the object level, as seen in the yellow samples. Traditional convolution contains more information about the background, as shown on the left, and because of that, the traditional detection methods use only the best classification score of the part features, dropping the others. For the tracking task, the object features of the other part beside the center play a very important role. So, to take advantage of the deformable convolution, we use all the features of different parts to distinguish different instances, as shown on the right with the different colour blocks. We combine the different part features with the center-ness score to alleviate the
impact of stride, where the final instance feature is obtained by weighted summation.

2.4 Clustering-based adaptive tracking

To link the instances, we employ the standard online tracking procedure. Features based on the instance-aware embedding are initialized in the first frame. In the subsequent frames, each instance-aware embedding feature is clustered to adapt to the instance variance, where the cosine distance is used to measure the similarity, as shown in Figure 6. And \( f_{\text{embedding}} \) indicates the final instance embedding of current frame.

To obtain the final tracking result, we perform the instance association based on similarities. Given the pseudo-bounding boxes \( B_{p_i} \) and \( B_{p_j} \), and their embedding \( f_{\text{embedding}_i} \) and \( f_{\text{embedding}_j} \), the similarity \( \phi \) is formulated as follows:

\[
\phi(I_{i}, I_{j}) = -\text{Dist}(f_{\text{embedding}_i}, f_{\text{embedding}_j}) + \alpha \times \text{IOU}(B_{p_i}, B_{p_j}),
\]

(6)

\[
\text{Dist}(f_{\text{embedding}_i}, f_{\text{embedding}_j}) = \| f_{\text{embedding}_i} - f_{\text{embedding}_j} \|,
\]

(7)

where \( \text{Dist} \) denotes the Euclidean distance and \( \text{IOU} \) represents the pseudo-bounding box IOU, \( \alpha \) is set to 0.5 by default. If an active track does not update for recent \( \beta \) frames, the track is ended automatically. For each instance, the similarity between the latest embedding of all active and current tracks is computed according to Equation (3). Following [24], the similarity threshold \( \gamma \) is set for instance association, the Hungarian algorithm [25] is exploited to perform instance matching, and the unassigned high-confidence detection will start new tracks. By default, \( \beta \) and \( \gamma \) are set at 24 and 0.3, respectively.

Due to the movement of the tracking objects, the extracted features continually change, as shown in Figure 7. The traditional association methods are usually hard to balance the complexity and efficiency, such as Kalman filter, which has some prior parameters to determine, including process variance, error variance. We use the clustering method to reduce the ID exchanges and improve tracking robustness, where the cosine function is used as the similar measurement. And furthermore, the method to update the features can be more exploited.

2.5 Loss function

According to the description above, the loss functions are defined as follows:

2.5.1 Heatmap loss

For each ground-truth box \( b_i = (x_i^1, y_i^1, x_i^2, y_i^2) \) of the image, the object center \( (c_i^x, c_i^y) \) can be computed by \( c_i^x = \frac{x_i^1 + x_i^2}{2} \) and \( c_i^y = \frac{y_i^1 + y_i^2}{2} \), respectively. The corresponding location on the feature map can be calculated as \( (\hat{c}_i^x, \hat{c}_i^y) = \left( \frac{x_i^1 + x_i^2}{4}, \frac{y_i^1 + y_i^2}{4} \right) \). Specialy, the Gaussian function is used to get the heatmap response at the location \( (x, y) \), defined as \( M_{xy} = \sum_{i=1}^{N} \exp \left( -\frac{(x-c_i^x)^2 + (y-c_i^y)^2}{2\sigma^2} \right) \), where \( N \) is the total number of objects and \( \sigma \).
is the standard deviation. The pixel-wise logistic function is used as the heatmap loss with focal loss [26]:

\[
L_{\text{heatmap}} = -\frac{1}{N} \sum_{x,y} \left\{ \begin{array}{ll}
(1 - \hat{M}_{xy})^\alpha \log(\hat{M}_{xy}), & \text{if } M_{xy} = 1 \\
(1 - \hat{M}_{xy})^\alpha \log(1 - \hat{M}_{xy}), & \text{otherwise}
\end{array} \right.,
\]

(8)

where \( \hat{M} \) represents the estimated heatmap, and \( \alpha, \beta \) are super-parameters.

2.5.2 Center-ness loss and regression loss

The center-ness loss and regression loss are defined as follows:

\[
L\left(\{p_{x,y}\}, \{b_{x,y}\}\right) = \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(p_{x,y}, c_{x,y}, \epsilon_{x,y}) + \lambda \frac{1}{N_{\text{pos}}} \sum_{x,y} 1[\epsilon_{x,y} > 0] L_{\text{reg}}(b_{x,y}, b_{\epsilon_{x,y}}),
\]

(9)

where \( L_{\text{cls}} \) indicates the focal loss as in [26] and \( L_{\text{reg}} \) represents the pseudo-bounding box IoU loss as UnitBox [25]. \( \epsilon \) indicates the location \((x, y)\) within the ground-truth box, \( c \) is the class label, and \( b_{x,y} \) is the bounding-box coordination. \( N_{\text{pos}} \) is the number of positive samples and \( \lambda \) is used to balance the classification loss and regression loss. \( 1[\epsilon_{x,y} > 0] \) is defined as the indicator function, being 1 if \( \epsilon_{x,y} > 0 \) and 0 otherwise. The total loss is calculated by summation over the whole feature map.

2.5.3 Instance-embedding loss

Object identity embedding is defined as a classification problem. Particularly, all object instances of the same identity of the training set are taken as one class [27]. For each ground-truth box \( b_i = (x_i', y_i', x_i'', y_i'') \) in the image, extract the embedding feature vector to get the class distribution \( p(k) \). When we denote the ground-truth class label as \( L_{ij}(k) \) with one-hot encoding, the embedding loss is computed as:

\[
L_{\text{embedding}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} L_{ij}(k) \log(p(k)),
\]

(10)

where \( K \) is the number of classes.

3 EXPERIMENTS

3.1 Experimental setup

3.1.1 Datasets and evaluation metrics

We composed a large training data set by combining the training data from the MOT16, MOT17 [1], and CUHK-SYSU [30] data sets. They contain sequences with varying viewing angle, size and number of objects, camera motion and frame rate. Average precision (AP) [30] was used to evaluate the detection performance, and true positive rate (TPR) was applied to evaluate the embedding representation. The tracking accuracy was evaluated with multiple object tracking accuracy (MOTA) [30] and IDF1 score (IDF1) [31], as they quantify two of the main aspects of multiple object tracking, namely, object coverage and identity preservation.

3.1.2 Training details

A variant of DLA-34 [18] is used as our default backbone (see Section 2.1), and the model weights is initialized on the COCO dataset [34]. And for the instance-aware embedding branch, which is pretrained with the task of Re-Identification (ReID) on two publicly datasets: Market1501 [35] and CUHK03 [36]. On MOT dataset, we do data augmentation by rotation, scaling, and cropping, and jointly finetune the detection and instance-aware embedding branches by adding the losses (Equations 8–10) together. The batch size is set to be 16. The input image is resized to 1088 × 608 and the feature map resolution is 272 × 152. We train for 22 epochs with a learning rate 1e−4, weight decay term 1e−5 and an Adam Optimizer with β1 and β2 set to 0.9 and 0.999, respectively.

3.2 Ablative study

3.2.1 Part-based instance embedding and center-based embedding

This section evaluates the impact of our proposed part-based instance embedding. Particularly, we compare with the
TABLE 1 Evaluation of the part-based instance-aware embedding and traditional center-based embedding, and the impact of the feature dimensions

| Embedding    | Dim | MOTA | IDF1 | IDs | FPS | TPR |
|--------------|-----|------|------|-----|-----|-----|
| Center based | 512 | 40.4 | 52.1 | 210 | 28.7 | 61.5 |
| Part based   | 512 | 40.4 | 55.3 | 182 | 29.6 | 64.2 |
| Center based | 256 | 40.2 | 51.1 | 160 | 30.6 | 63.5 |
| Part based   | 256 | 40.5 | 56.3 | 157 | 32.0 | 65.3 |
| Center based | 128 | 40.4 | 53.9 | 136 | 31.0 | 67.3 |
| Part based   | 128 | **41.3** | 55.7 | **127** | 32.5 | **69.2** |

The center-based embedding method, where the embedding feature is extracted using only the center features, and the feature map is referred to as the object center position [32]. The remaining factors stay the same, and the 2DMOT15 training set is split into eight training videos and three validation videos. The feature of 512 dimension has been used in previous works, and to evaluate the importance of the feature dimension, we compare the different dimensions. The results are shown in Table 1.

One can see that the overall MOTA score is improved using our proposed part-based instance-aware embedding method. Through analysis of the detailed metrics, the importance of fewer ID switches becomes clear, and the detection metric measured by TPR improves from 61.5 to 69.2. Furthermore, when the representation dimension decreases from 512 to 128 gradually, the TPR improves, the number of ID switches decreases from 182 to 127, and the inference speed also improves. By comparison, with the increase of the feature dimension for the part-based method, the MOTA score declines. This is because there will be more information from the boundary parts that contain more features from other instances, which makes the fused parts of the model difficult to identify. So, proper low-dimension representation satisfies the tracking task.

Figure 8 shows the embedding features visualization using different method. Different instance features are mixed for a center-based approach, especially for high dimensions. The proposed part-based approach distinguishes them well. We think the backbone also plays an import role, which enables more flexible receptive fields for objects of different sizes, it is very important for anchor-free method. Since anchor-free method has no anchors for scale variance, and the different scale convolutions can be a substitution for that, which can be confirmed in Table 2.

3.2.2 Cluster-based association versus similarity matching

We evaluate the different association mechanisms in Table 3. The basic tracking structure follows the explanation above, but with different association heads. For the proposed method, the association matches with detections up to $\beta$ frames in the past. While the other mechanisms do not need initialization, only the adjacent frames are associated.

FIGURE 8 Visualization of the embedding features with t-SNE [28]. Different colours indicate different instances
TABLE 4 Comparison of the proposed one-shot tracker with the SOAs on two datasets

| Dataset | Method       | MOTA  | IDF1  | IDs   | FPS  |
|---------|--------------|-------|-------|-------|------|
| MOT15 train | JDE[29]    | 67.6  | 66.7  | 218   | 22.5 |
|         | SORT[24]    | 33.4  | 40.4  | 1001  | 260.5|
|         | DeepSORT[38]| 61.4  | 62.2  | 781   | 107.0|
|         | DeepMOT[39] | 44.1  | 46.0  | 1347  | 1.6  |
|         | MPNTrack[40]| 51.5  | 58.6  | 375   | 6.5  |
|         | CSTrack[41] | 67.3  | 67.9  | 2994  | 16.9 |
|         | CTracker[42]| 67.6  | 57.2  | 1897  | 34.4 |
|         | CenterTrack[43]| 77.1 | 76.0  | 80    | 30.9 |
|         | FairMOT[33] | 78.3  | 77.7  | 73    | 32.5 |
|         | PartMOT(Ours)| 71.3  | 73.7  | 127   | 30.9 |

| MOT16 test | JDE[29]    | 64.4  | 55.8  | 1544  | 18.5 |
|           | SORT[24]   | 59.8  | 53.8  | 1423  | 230.3|
|           | DeepSORT[38]| 65.4  | 64.3  | 751   | 106.3|
|           | DeepMOT[39]| 54.8  | 53.4  | 645   | 1.6  |
|           | MPNTrack[40]| 58.6  | 61.7  | 354   | 6.5  |
|           | CSTrack[41]| 58.6  | 67.9  | 958   | 16.9 |
|           | CTracker[42]| 69.6  | 58.9  | 1762  | 34.4 |
|           | CenterTrack[43]| 65.5 | 61.6  | 113   | 22   |
|           | FairMOT[33]| 68.7  | 70.4  | 953   | 25.9 |
|           | PartMOT(Ours)| 71.3  | 73.7  | 127   | 30.9 |

The table reveals that the proposed cluster-based association method achieves a notably better MOTA score, due to the representation clustering. In particular, the number of ID switches decreases from 182 to 127, which indicates that the applied method largely improves the robustness. And the traditional bounding box IoU association mechanism is also evaluated, it can be found that just the IoU mechanism large decrease the performance, because it does not take the instance feature into consideration in detection at the meantime. But the association speed is largely improved.

3.3 Qualitative results

In this section, we compare the proposed Part-MOT with state-of-the-art one-shot methods, including the JDE [29] and FairMOT [33], which also put together object detection and identity-feature embedding. And other SOTA methods such as SORT[24], DeepSORT[38], DeepMOT[39], MPNTrack[40], CSTrack[41], CTracker[42], CenterTrack[43] and FairMOT[33]. Specifically, we use the 2DMOT15-train and MOT16-test dataset to validate our model. The performance results are shown in Table 4 with the MOTA metric [30] and IDF1 [31], showing that the proposed approach outperforms the other two methods. This validates the effectiveness of the one-shot part-based approach over the traditional center-based one.

To validate the proposed approach, we compare some non-real-time top-performing trackers, including MIFTv2, ISE_MOT15R, Tracktor++v2, TrctrD15, AP_HWDPL_p, SST_MOT15, AMIR15, STRN, Tube_TK, RAR15, AP_HWDPL, EAMTT, EAMTT_pub, NSH2015, SLA_public and MDP_SubCNN [37].

The MOTA score versus speed curve on 2DMOT15 is shown in Figure 9. One can see that the proposed Part-MOT method ranks first among all the trackers, with higher MOTA score and faster inference speed, by taking advantage of the simple structure.

Figure 10 shows the successful tracking cases on MOT2015 and MOT2016. Owing to the robustness of the part-based instance-aware embedding, our tracker can detect and track targets with huge scale deformation. The combination of backbone and the instance-aware embedding branch can also adapt to the varying object shapes. The failure cases are shown in Figure 11. When the target is severely occluded by a similar object, the tracking box drifts. The same failure can be caused when the ground truth contains only a small part of an object, due to our tracker taking more semantic information about objectness and tracking the whole instance.

4 CONCLUSION

Here, we propose a one-shot, anchor-free, multiple-objects tracking method that consists of two parallel branches. The detection branch classifies instances and regresses the dense points of the sampled locations around contours, and it estimates center-ness to weight the part features of a global instance representation. The instance-aware embedding branch extracts the part-based global instance features, using a combination of deformable convolution and center-ness scores. To improve the linking robustness, we first cluster the features of the instance, and use the IoU and feature-similarity measure to do the association. Part-MOT is designed as a simple one-shot object tracker, and the experiments show its effectiveness, accuracy, and robustness. In the future work, we will study applying our framework to other video tasks, for example, video object detection and segmentation, and at the meantime, explore other instance representation learning method, such as contrast learning [44] to this framework.

FUNDING INFORMATION

This research was supported by the National Natural Science Foundation of China (Project Number: 11804263).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest to report regarding the present study.

ACKNOWLEDGEMENT

The authors are grateful for the constructive suggestions from anonymous reviewers that significantly enhance the presentation of this paper.
FIGURE 9  Performance-speed trade-off of trackers on 2DMOT15 benchmark

FIGURE 10  Visualization of successful tracking sequences from MOT2015 on the top, and MOT2016 at bottom

ORCID
Xiaohu Liu  https://orcid.org/0000-0003-0896-1810

REFERENCES
1. Milan, A., et al.: MOT16: a benchmark for multi-object tracking. arXiv:1603.00831 (2016)
2. Caelles, S., et al.: The 2018 DAVIS challenge on video object segmentation. arXiv:1803.00557 (2018)
3. Kristan, M., et al.: A novel performance evaluation methodology for single-target trackers. IEEE Trans. Pattern Anal. Mach. Intell. 38(11), 2137–2155 (2016)
4. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP J. Image Video Process. 2008, 1–10 (2008)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings ofCVPR, San Diego, pp. 886–893 (2005)

FIGURE 11  Visualization of failure cases
25. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Quarterly 2(1–2), 83–97 (1955)
26. Lin, T.Y., et al.: Focal loss for dense object detection. In: Proceedings of CVPR, Las Vegas, pp. 770–778 (2016)
27. Wang, Z., et al.: Towards real-time multi-object tracking. arXiv:1909.12605 (2019)
28. Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. 9(86), 2579–2605 (2008)
29. Xiao, T., et al.: Joint detection and identification feature learning for person search. In: Proceedings of CVPR, Honolulu, pp. 3415–3424 (2017)
30. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear MOT metrics. EURASIP J. Image Video Process. 2008(26309), 1–10 (2008)
31. Kasturi, R., et al.: Framework for performance evaluation for face, text and vehicle detection and tracking in video: data, metrics, and protocol. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) 31(2), 319–336 (2009)
32. Zhou, X., Wang, D., Krahenbühl, P.: Objects as points. arXiv:1904.07850 (2019)
33. Zhang, Y., et al.: A simple baseline for multi-object tracking. arXiv:2004.01188v4 (2020). https://arxiv.org/abs/2004.01188v4
34. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Proceedings of European Conference on Computer Vision, pp. 740–755. Springer, Heidelberg (2014)
35. Zhu, J., et al.: Online multi-object tracking with dual matching attention networks. 1, 2 (2018)
36. Li, W., et al.: Deepreid: deep filter pairing neural network for person re-identification. In CVPR, (2014)
37. Data set: https://motchallenge.net/results/MOT15/?det=All
38. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. arXiv:1703.07402 (2017)
39. Xu, Y., et al.: How to train your deep multi-object track-er. arXiv:1906.06618v2 (2019)
40. Braso, G., Leal-Taixe, L.: Learning a neural solver for multiple object track-ing. arXiv:1912.07515 (2019)
41. Liang, C., Zhang, Z., Lu, Y., et al.: Rethinking the competition between detection and ReID in multi-object tracking. arXiv:2010.12138 (2020)
42. Peng, J., et al.: Chained-tracker: chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. arXiv:2007.14557 (2020)
43. Zhou, X., Koltun, V., Krahenbühl, P.: Tracking objects as points. arXiv:2004.01177v2 (2020). https://arxiv.org/abs/2004.01177v2
44. Wang, W., et al.: Exploring cross-image pixel contrast for semantic segmentation. arXiv:2101.11939 (2021). https://arxiv.org/abs/2101.11939
45. Huang, Z., et al.: Mask scoring R-CNN. In: Proceedings of CVPR, Long Beach, pp. 6409–6418 (2019)