Learning to Track Instances without Video Annotations

Yang Fu[1*], Sifei Liu[2], Shalini De Mello[2], Umar Iqbal[2],
Humphrey Shi[1,3†], Jan Kautz[2]
[1]University of Illinois at Urbana-Champaign, [2]NVIDIA, [3]University of Oregon

1. Architecture Details

Video Instance Segmentation. As described in the paper, we propose an instance embedding head to learn the discriminative representation of different instances. This head shares a similar structure to the category prediction head in SOLO [13]. Specifically, we use four convolutional layers with 256 output channels followed by group normalization layers. We add an additional convolutional layer with 128 output channels for dimensionality reduction. This embedding module is adopted for features at different levels in FPN [7]. The video correspondence branch has the exact same structure as the embed branch.

Pose Tracking. Pose tracking [1] is more challenging for learning a discriminative feature embedding, since it focus on discriminating between different humans, which are instances of the same category. That is, compared to YouTube-VIS [16], pose tracking needs to learn a more fine-grained feature representation to discriminate different human instances across frames. Thus, we propose the keypoint embedding module (KEM) as demonstrated in Fig. 1.

Unlike the instance embedding module, the KEM is designed to learn the discriminative features of different joints. In particular, we first concatenate the predicted heatmap, which exists in the original PointSetAnchor [14], see Fig. 1, with FPN features as the input to the embedding head. In contrast to designing the head similar to the classification branch in the video instance segmentation framework, we introduced an encoder-decoder with one convolutional layer as the encoder and one de-convolutional layer as the decorder. This encoder-decoder structure is used to obtain the keypoint-level embedding. In addition, the keypoint prediction is also adopted as prior knowledge to indicate the location of each joint on the embedding feature map, to filter out the valid keypoint embeddings of different joint definitions, *i.e*. neck, shoulder, wrist etc. We apply the same instance contrastive (IC) objective both at the keypoint and the instance levels. In other words, we repeat the IC loss 17 times since there are 17 joints defined in the COCO [8] dataset. Besides these, we also average the embedding of all seventeen joints as a person-level embedding and apply the IC loss on it again. This KEM is added as a branch in parallel to the classification and shape regression branches in PointSetAnchor [14].

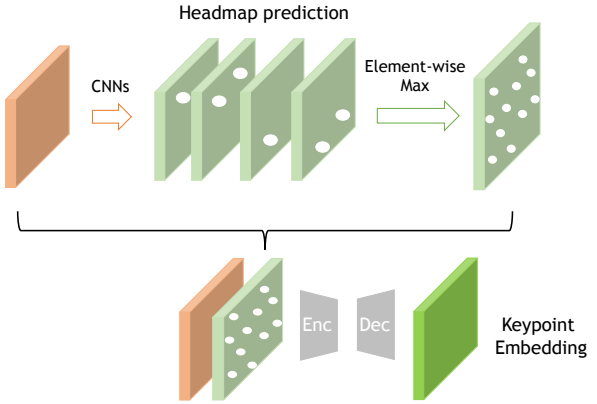

Figure 1. The architecture of the keypoint embedding module (KEM).

Difference with associative embedding (AE). The IC loss correlated to associative embedding approaches [9, 6]. However, both were designed to learn a keypoint embedding for spatial grouping within an individual image, e.g., AE in [6] was applied only to the SpatialNet that is independent of pose tracking, which was performed by another TemporalNet. AE in neither of them was utilized for learning cross frame correspondence, that we aim for.

2. Implementation Details

Training Details. For video instance segmentation, we use ResNet50 [5] pretrained on ImageNet [3] as the backbone network and train the SOLO [13] framework with the proposed instance embedding branch via a classification loss, mask prediction loss and an IC loss on COCO [8] instance segmentation annotations. The λ parameter in Eq (5) is set to 1. We further learn video correspondences across frames using unlabeled sequences of YouTube-VOS [15]. Each sequence is sampled from the same video randomly

∗ This work was done while Yang Fu was a research intern at NVIDIA
† corresponding author

with random intervals from 2 to 8.

Similarly, for pose tracking, we use HRNetW48 [12] as the backbone network and train PointSetAnchor [14] along with the KEM. The other steps are the same as those employed for video instance segmentation. Video correspondence is learnt on the PoseTrack2018 [1] training set without any annotations. Since the joint definitions of COCO [8] are different from PoseTrack [1], we further fine-tune the model on the MPII [2] training data.

Inference Details. During inference, we associate the objects in an online fashion following a procedure similar to the one proposed in [16] for video instance segmentation. A memory bank is established to store all detection results: object category, bounding box location, mask segment and the learned embedding feature. Object association is achieved by cosine similarity of the object embedding feature.

Different from [16], which has an additional category of "new object" while training with identity annotations (track ids) across frames, we do not have such a category definition. Thus we make several modifications to the original tracking procedure. Assume M objects are detected in previous frames, and N objected are detected in the current frame. Then the similarity scores should form a $N \times M$ association matrix. To effectively figure out the new objects, we employ a bi-directional softmax [11] instead of the original softmax. Bi-directional softmax computes the softmax operation along the row and column directions. The new object cannot guarantee good consistency in both directions, resulting in a lower score for new objects. Based on the similarity matrix, we assign every detected object $(1 : N)$ a unique identity through the row-wise argmax operation. If the similarity score is lower than a threshold, this object is considered as a new object and its embedding feature is concatenated to the memory bank. On the other hand if it is higher than the threshold, it is assigned to an existing object and its embedding is updated by the newly tracked object's with a momentum value of 0.7.

Post-processing. To be consistent with the previous approaches and to improve tracking performance, we also apply the post-processing procedure introduced in [16], which combines category confidence, bounding box Intersection over Union (IoU), embedding similarity and category consistency through a weighted sum. In particular, the final similarity between newly detected objects and the existing candidates in the memory bank can be computed as:

$$s(n, m) = \mathrm{sim}(n, m) + \alpha \mathrm{c(n)} + \beta \mathrm{IoU}(\mathrm{b_n}, \mathrm{b_m}) + \gamma \delta(\mathrm{c_n}, \mathrm{c_m}) \quad (1)$$

where $\mathrm{c(n)}$ is the classification confidence score of the nth object, $\mathrm{c_n}$ is the predicted category and $\delta(\mathrm{c_n}, \mathrm{c_m})$ is the Kronecker delta function, which returns one if and only if $\mathrm{c_n}$ is equal to $\mathrm{c_m}$, otherwise it returns zero. Note that as discussed in our paper, the current post-processing method can only bring a limited improvement on our approach compared to others, due to the obvious domain gap between the training set of COCO, and the validation set of YouTube-VIS. In this work, we mainly focus on learning a tracking embedding representation while leaving domain adaptation of the original SOLO heads to further work.

Method	runtime(fps)
LightTrack [10]	0.8
AlphaPose [4]	2.2
Ours	4.1
Ours (ms)	1.3

Table 1. Average running time of different pose tracking methods on the PoseTrack 2018 validation set. "ms" represents multi-scale testing.

3. More Experiments

3.1. Pose Tracking Running Time

The running time of the proposed semi-supervised tracking approach on PoseTrack2018 [1] is shown in Table 1. Compared with the top-down methods, LightTrack [10] and AlphaPose [4], our approach performs more efficiently since it estimates all joint locations of different persons at the same time. In addition, LightTrack [10] utilizes precomputed human detection results and its efficiency can further decrease on considering the detection step as well.

3.2. Comparison with Associate Embedding

We also compare our proposed instance contrastive loss with Associate Embedding (AE) loss in [9]. For a fair comparison, we apply AE to our pose tracking experiments. We replace our joint-level embedding with the original form of AE, while keeping all the other settings the same. The AE model achieves **63.5**% on MOTA, which is lower than ours, i.e., **64.7**%(Table.4 in the paper).

3.3. More visualization

We show more qualitative results of our proposed semi-supervised tracking approach on the video instance segmentation and pose tracking tasks and compare them with the baseline model, i.e., image-based instance segmentation/pose estimation models with spatial distance association, as described in our main paper (see Sec. 4.3) in Fig. 2. It can be observed that compared to the baseline model, our proposed semi-supervised tracking can detect instance masks, human poses and associate different instances across frames much more accurately.

Figure 2. Visualization results of our proposed semi-supervised tracking approach compared to baseline method mentioned in our paper on video instance segmentation and pose tracking. Each row has five sampled frames from a video sequence. Categories and instance masks are shown for each object. Note that objects with the same predicated identity across frames are marked with the same color. Zoom in to see details.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Conf. Comput. Vis. Pattern Recog.*, pages 5167–5176, 2018. 1, 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Conf. Comput. Vis. Pattern Recog.*, pages 3686–3693, 2014. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conf. Comput. Vis. Pattern Recog.*, 2009. 1

[4] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Int. Conf. Comput. Vis.*, pages 2334–2343, 2017. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1

[6] Sheng Jin, Wentao Liu, Wanli Ouyang, and Chen Qian. Multi-person articulated tracking with spatial and temporal embeddings. In *CVPR*, 2019. 1

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Conf. Comput. Vis. Pattern Recog.*, 2017. 1

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014. 1, 2

[9] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *arXiv preprint arXiv:1611.05424*, 2016. 1, 2

[10] Guanghan Ning, Jian Pei, and Heng Huang. Lighttrack: A generic framework for online top-down human pose tracking. In *Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 1034–1035, 2020. 2

[11] Jiangmiao Pang, Linlu Qiu, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense instance similarity learning. *arXiv preprint arXiv:2006.06664*, 2020. 2

[12] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703, 2019. 2

[13] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *Eur. Conf. Comput. Vis.*, 2020. 1

[14] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[15] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1

[16] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Int. Conf. Comput. Vis.*, 2019. 1, 2