ISDA: POSITION-AWARE INSTANCE SEGMENTATION WITH DEFORMABLE ATTENTION

Kaining Ying *Zhenhua Wang* ⋆ *Cong Bai* *Pengfei Zhou*

College of Computer Science and Technology, Zhejiang University of Technology

ABSTRACT

Most instance segmentation models are not end-to-end trainable due to either the incorporation of proposal estimation (RPN) as a pre-processing or non-maximum suppression (NMS) as a post-processing. Here we propose a novel end-to-end instance segmentation method termed ISDA. It reshapes the task into predicting a set of object masks, which are generated via traditional convolution operation with learned position-aware kernels and features of objects. Such kernels and features are learned by leveraging a deformable attention network with multi-scale representation. Thanks to the introduced set-prediction mechanism, the proposed method is NMS-free. Empirically, ISDA outperforms Mask R-CNN (the strong baseline) by 2.6 points on MS-COCO, and achieves leading performance compared with recent models. Code will be available soon.

Index Terms— Instance segmentation, end-to-end, deformable attention, position-aware kernel

1. INTRODUCTION

In vision community, removing hand-designed components and enabling the end-to-end training serve as stimulants to further performance improvement [1, 2, 3, 4, 5, 6]. In terms of instance segmentation [7, 8, 9, 10, 11, 12, 13], the main obstacles to train the model in an end-to-end way include two aspects. First, current segmentation methods, either top-down [7, 9, 10, 8, 11, 14] or bottom-up [15, 16] have decomposed the task into several consecutive sub-tasks. Second, as shown by Fig. 1, a post-processing step, namely NMS is typically taken to remove redundant predictions, which is non-differentiable and hinders back-propagating gradients.

Recently, DETR [1] proposed to train an end-to-end detector with a set-based loss and a Transformer encoder-decoder architecture. Nevertheless, DETR takes a long time to train due to the large computational overhead on dense attention computation, and it typically performs bad at detecting small objects as only a single-scale feature map is utilized. Very recently, deformable DETR [2] introduced a sparse attention mechanism and used multi-scale features, which boosts the detection performance. Inspired by this, we craft an end-to-end instance segmentation framework termed ISDA, which is shown by Fig. 2.

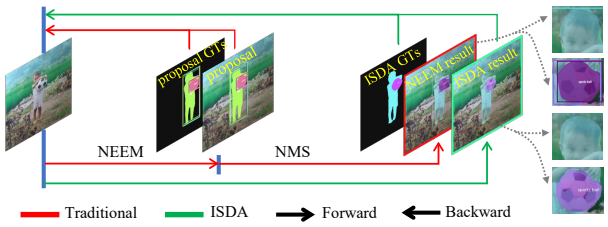

Fig. 1. Comparison of ISDA (ours) and traditional **N**on-**E**nd-to-**E**nd **M**odel (NEEM, *e.g.,* Mask R-CNN) in training (left arrow) and testing (right arrow) phases, best viewed by zooming in. Note training and testing of NEEM are not aligned due to usage of an extra NMS in testing. When redundant object proposals are provided, NEEM also gives redundant instance masks (see the football). ISDA seldom produces redundant masks, and it typically gives clearly better results in border areas of objects (see the head of the child).

Similar to our proposed ISDA, SOLOv2 [13] also learns object kernels from data. ISDA has two merits over SOLOv2. First, instead of predicting object kernels for each cell in a fixed grid, ISDA is designed to adaptively learn object queries from data, which is more flexible than SOLOv2. Second, in order to overcome the translation-invariance of convolution, SOLOv2 introduces additional channels of relative coordinates for both kernel and feature learning. Apart from embedding positional information into feature learning, ISDA further concatenates the learned object positions (namely the reference points) with the object features to enhance the positional awareness of the learned object kernels. The effectiveness of such a design is assessed via ablation experiments.

Our main contributions are of three aspects. First, we propose an instance segmentation framework based on deformable Transformer, which enables the end-to-end learning of object queries efficiently and effectively from data. Second, we present a method to generate position-aware object kernels which are especially useful for segmenting objects of similar appearances. Last but not least, our approach achieves leading performance on the challenging MS-COCO dataset.

⋆ indicates corresponding author (email: zhhwang@zjut.edu.cn).

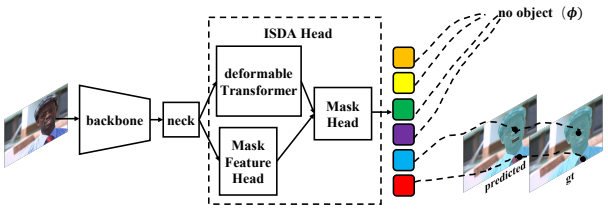

Fig. 2. An overview of the proposed ISDA. The architecture consists of three ingredients: 1) The backbone and neck module to extract multi-scale features; 2) The ISDA head which includes a deformable Tramsformer, a mask feature head and a mask head to predict object masks; 3) The Bipartite matching block which associates predictions with ground truth to compute loss. All parameters of ISDA could be trained in an end-to-end fashion.

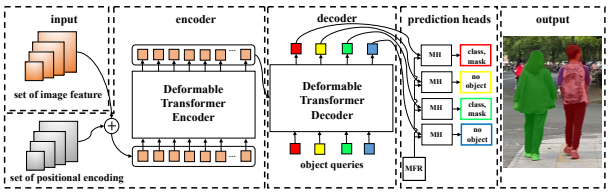

Fig. 3. Illustration of the proposed ISDA head. ISDA takes as inputs the feature maps from neck supplemented with a set of encoded positions. We feed the inputs into deformable Transformer (encoder and decoder) to get a fixed amount of $(object-feature,\ reference-point)$ pairs. Then these pairs together with the mask feature representation (MFR) generated by a mask feature head (omitted here) are input into a series of mask heads (with shared weights) to generate final predictions. The generated masks are visualized in the rightmost diagram.

2. RELATED WORK

2.1. Vision Transformer

Transformer [17] was first proposed for sequence-to-sequence machine translation and since then has become the *de facto* standard in Natural Language Processing tasks. The core mechanism of Transformers is self-attention, which is beneficial in terms of modeling long-range relations. Transformers have shown to be promising in terms of addressing various vision tasks [1, 2, 18, 19, 20, 21, 22, 23]. Very recently, VISTR [3] applied Transformer to instance segmentation in videos, a distinct task from our ISDA focusing on an image-level task.

2.2. End-to-End Instance-Level Recognition

An increasing number of works [1, 2, 18, 19, 24, 25] implement end-to-end detectors to achieve compelling results. To this end, the bipartite matching [26] has become the essential component for achieving end-to-end training of detectors. In the area of instance segmentation, this can be achieved by integrating sequential modules with recurrent networks [27, 28]. Nevertheless, these early methods were only evaluated on small datasets without comparing against modern approaches. Recently, [29] uses Transformer encoder only to fuse RoI features and image features to generate mask embedding. In contrast, ISDA uses deformable attention encoders and decoders to generate positional-aware kernels, which are then combined with mask features to generate masks directly.

3. THE ISDA MODEL

As illustrated in Fig. 2, ISDA contains three blocks: a CNN backbone and neck, a ISDA head and a matching module to supervise the model training. This section first introduces the the backbone and neck of ISDA (Section 3.1). Section 3.2 elaborates the ISDA head, and Section 3.3 describes the bipartite match cost and the set prediction loss.

3.1. Backbone and Neck

Given an image denoted by $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$, the CNN backbone extracts four feature maps with different resolutions, denoted by $\{C_i \in \mathbb{R}^{c_i\times H_i\times W_i}\}_{i=2}^{5}$. Here c_i, H_i, W_i denote the channel number, the height and the width of the feature map C_i. The neck takes the multi-scale features as the input and then enhances them separately as that done by deformable DETR [2]. Consequently, we get $\{P_i \in \mathbb{R}^{256\times H_i\times W_i}\}_{i=2}^{6}$, where P_6 is down-sampled form C_5.

3.2. ISDA Head

Fig. 3 illustrates the architecture of our ISDA head. It contains three components: an encoder-decoder deformable Transformer, a mask feature head (omitted in the figure) used to generate mask feature representations (MFR), and a mask head to make final predictions.

Encoder. We sum the feature maps $\{P_i\}_{i=3}^{6}$ and the encoded positions at different scales. This multi-scale representation is useful to identify which feature-scale the query pixel belongs to. Here queries and keys correspond to the embedded pixel-wise elements within the multi-scale feature maps. The outputs of the encoder take the same shapes as the inputs.

Decoder. The inputs of the decoder include the output from encoder and an extra object query vectors. Note that these queries are learned during training with random initialization, and are fixed for testing. Each layer includes two components, namely the cross-attention module and the self-attention module. The cross-attention module takes object queries to extract object features from input feature maps using a deformable attention manner. The self-attention module enables the object queries interacting with each other. The decoder outputs a set of object features and their corresponding reference points, which are taken to compute object kernels as depicted by Fig. 4.

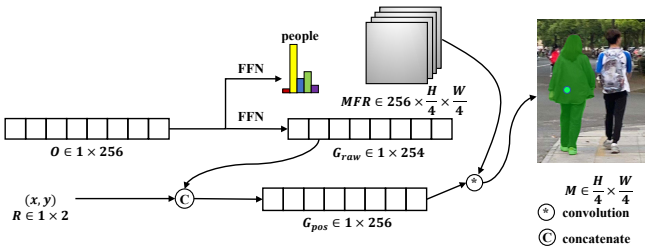

Fig. 4. The structure of the mask head (best viewed by zooming in). The predicted mask (for left person only) and its reference point (the green circle) is visualized here.

Mask Feature Representation. Inspired by SOLOv2, ISDA learns a compact and high-resolution mask feature representation (MFR) with feature pyramid. After repeated stages of 3×3 Conv, group-norm, ReLU and $2\times$ bilinear upsampling, the neck features $\{P_i\}_{i=2}^{5}$ are fused (via element-wise summation) to create one single output at $1/4$ scale. It is worth noting that normalized pixel coordinates are fed into the smallest feature map (at $1/32$ scale) before convolution and upsampling. In ablation study in Section 4.1, we show the importance of appending such positional-information.

The Mask Head. The Mask Head (MH) of ISDA is depicted by Fig. 4. Its inputs include 1) the obejct feature vector $O \in \mathbb{R}^{256}$, 2) the normalized reference points R (see an example in Fig. 5) and 3) the mask feature representation $MFR \in \mathbb{R}^{256\times H/4\times W/4}$. Our MH generates all object masks in parallel in three steps. First, the object feature vector is fed into two different feed-forward networks (FFNs), in order to compute object classification scores P_c and to obtain raw object kernel G_{raw}. Second, we concatenate G_{raw} with R to obtain the position-aware object kernel G_{pos}. Thanks to the the positional encoding in deformable Transformer, the object feature has included the positional information already. However, we find that appending the reference point to raw object kernel improves the position-awareness of the object kernel, hence is able to moderately improve the performance (see Section 4.1). Finally, G_{pos} is convolved with MFR to generate object masks M.

3.3. The Loss

We follow the procedure of [1] in computing loss, which includes bipartite matching and loss computation. The only difference is that we replace the bounding box loss with the mask IoU loss, which is defined as $1 - \text{IoU}(m_i, \hat{m}_{\sigma(i)})$. Here m_i and $\hat{m}_{\sigma(i)}$ denote the i^{th} ground truth and predicted mask in a permutation σ. Please refer to [1] for more details.

4. EXPERIMENTS

We implement ISDA based on the open source project MMDetection [30]. Unless otherwise specified, we use the ResNet-50 as our backbone network which are pretrained on ImageNet[31]. The deformable Transformer and the classification branch in ISDA are pretrained on MS COCO for fast convergence. In the ablation experiments, all models are trained in 12 epochs with learning rate decay, which drops the learning rate by a factor at 9th and 11th epoch, respectively. For testing the final model is trained in 24 epochs with learning rate decayed at 16th and 22th epoch, respectively. Due to the limitation on GPU memories (only $3\times$ GTX 1080 Tis are available), all compared models are trained by ourselves with the batch size equals 3. Following DETR[1], we use AdamW[32] optimizer and set the initial learning rate as 1.87e-5. We use a multi-scale training strategy and set loss weights λ_{cls} and λ_{mask} to 1 and 3, respectively.

Fig. 5. Visualization of learned reference points. Blue points represent background, while red, green and lemon points correspond to three zebras.

Table 1. Results on MS-COCO `val2017` by changing the resolution of MFR. Here the "resolution" column lists the ratio of the predicted mask to input image size. While increasing the resolution improves results on small objects (AP_S), it degrades results on large objects (AP_L).

Resolution	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1/8	35.0	58.3	35.9	14.6	38.5	**54.7**
1/4	**36.5**	**58.9**	**38.3**	17.4	**39.5**	54.6
1/2	36.4	58.7	38.3	**17.6**	39.3	53.8

Table 2. Instance segmentation results with different auxiliary positional information. Including both MFR (MP) and Kernel positions (KP) delivers the best performance (with Delta $= 4.1$).

MP	KP	Delta	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		0	32.4	56.9	32.2	15.6	35.5	47.4
✓		+3.7	36.1	58.5	37.9	16.6	39.0	54.5
	✓	-0.6	31.8	56.0	31.9	15.4	34.8	47.1
✓	✓	+4.1	**36.5**	**58.9**	**38.3**	**17.4**	**39.5**	**54.6**

4.1. Ablation Experiments

To analyze ISDA, we conduct two ablation studies: 1) The choices of mask resolutions; 2) The positional information.

Table 3. Comparison with Mask R-CNN (a strong baseline) and SOTA methods.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [7]	36.1	58.2	38.5	20.1	38.8	46.4
SOLO [12]	35.1	55.9	37.4	13.7	37.6	51.6
SOLOv2 [13]	37.4	58.4	40.1	15.4	40.2	**57.4**
CondInst [33]	36.9	58.2	39.6	19.8	39.3	48.0
BlendMask [34]	37.0	58.0	39.4	19.5	39.9	53.1
ISTR [29]	37.6	-	-	**22.1**	40.4	50.6
ISDA (**ours**)	**38.7**	**62.0**	**41.1**	17.0	**41.2**	55.7

Recall that ISDA generates object masks by convolving the predicted object kernels with the mask features. Clearly, the mask resolution depends on the mask feature representation. We test three different resolutions which correspond respectively to $1/2$, $1/4$ and $1/8$ of the input image size, and the results are provided by Table 1. In general, $1/4$ scale admits the best performance (1.5 and 0.1 points better than $1/8$ and $1/2$ scales) in terms of average precision (AP). Not surprisingly, the highest resolution gives the best performance on small objects. However, it performs much worse on larger targets. Note all resolutions needs to be resized to the dimension of the original image, which inevitably results in the lost of details on object edges. As a trade-off, we use $1/4$ scale for all subsequent experiments because of its good performance and lower computational overhead.

Aside positional encoding [17], ISDA incorporates two extra sources of positional information to disentangle different objects taking similar appearance. The first source (MFR Pos.) is to add two channels of normalized pixel coordinates to the mask feature, as described in Section 3.2. The second source (kernel Pos.) is the reference points added into the object kernels, as illustrated by Fig. 4. From Table 2, we can find that the baseline already has spatial awareness to some extent due to the zero-padding operation [12]. We can see that the performance improves by 3.7% with MFR Positions. Note that kernel Pos. performs even worse than the baseline. This is because, without the aid of MFR Pos., the position-aware kernel (simply with kernel Pos.) is incapable of distinguishing similar objects due to the translation invariance of convolution. Interestingly, with the help of MFR Pos., kernel Pos. improves the result by 0.4% (the bottom row) in terms of AP. Specifically, when doing convolution on MFRs, the concatenated coordinate channels in MFR and the reference points in kernels are multiplied to compute positional similarities. Hence the kernels are able to better identify the corresponding objects using both appearance and positional features.

4.2. Qualitative and quantitative results

We compare ISDA against SOTA mothods on MS-COCO `test-dev2017`, see Table 3 for details. Thanks to the introduced end-to-end training paradigm and the learned position-aware kernels, ISDA surpasses all SOTA approaches. Also note that ISDA outperforms Mask R-CNN [7] (a well-known strong baseline) by 2.6 points in terms of AP, which is impressive in instance segmentation. While our ISDA performs best in general (on AP, AP_{50} and AP_{75}), its results on small (AP_S) and large (AP_L) objects are not the bests. This is probably because ISDA uses deformable Transformer to sample and synthesize features across different scales, which benefits most the mask generation of middle-sized objects.

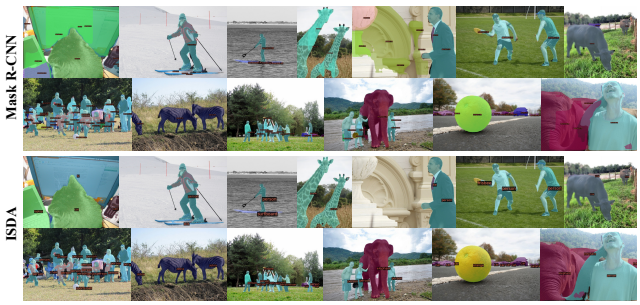

Fig. 6. Mask R-CNN results [7] (top) vs. ISDA results (bottom) on some examples. While Mask R-CNN gives duplicated masks and coarse edge segmentation results, our ISDA is able to address such issues and admits genuine masks.

We visualize some segmentation results obtained by Mask R-CNN and ISDA in Fig. 6. Mask R-CNN suffers from generating duplicated masks when NMS failed to filter out repeated instances, and it typically gives coarse masks along object edges. Thanks to the learned position-aware kernel and the informative mask features, ISDA is able to address such issues and gives almost perfect segmentation, at least on these examples. However, ISDA can still get trouble in segmenting overlapping objects, as that shown by the rightmost diagram in bottom row. We will address this issue in future.

5. CONCLUSION

We have proposed a novel single-stage instance segmentation method, named ISDA. ISDA introduced a Transformer-style framework for instance segmentation, which effectively removed NMS and achieved end-to-end training and inference. Moreover, ISDA is able to distinguish similar objects better by learning extra positional features. Empirically, ISDA admits genuine object masks and achieves leading performance compared with recent approaches.

Acknowledgement

This work is supported by Zhejiang Provincial Natural Science Foundation of China (LY21F020024, LR21F020002) and National Natural Science Foundation of China (61802348, U20A20196).

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229. 1, 2, 3

[2] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020. 1, 2

[3] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia, "End-to-end video instance segmentation with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8741–8750. 1, 2

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Region-based semantic segmentation with end-to-end training," in *European Conference on Computer Vision*. Springer, 2016, pp. 381–397. 1

[5] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang, "An end-to-end network for panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6172–6181. 1

[6] Kevin Lin, Lijuan Wang, and Zicheng Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1954–1963. 1

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. 1, 4

[8] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418. 1

[9] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2359–2367. 1

[10] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768. 1

[11] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al., "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983. 1

[12] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li, "Solo: Segmenting objects by locations," in *European Conference on Computer Vision*. Springer, 2020, pp. 649–665. 1, 4

[13] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural Information Processing Systems*, 2020. 1, 4

[14] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár, "Tensormask: A foundation for dense object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2061–2069. 1

[15] 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

[16] Bert De Brabandere, Davy Neven, and Luc Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017. 1

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017. 2, 4

[18] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng, "End-to-end object detection with fully convolutional network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15849–15858. 2

[19] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al., "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14454–14463. 2

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 2

[21] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6881–6890. 2

[22] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo, "Learning texture transformer network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5791–5800. 2

[23] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7464–7473. 2

[24] Rui Yao, Cunyuan Gao, Shixiong Xia, Jiaqi Zhao, Yong Zhou, and Fuyuan Hu, "Gan-based person search via deep complementary classifier with center-constrained triplet loss," *Pattern Recognition*, vol. 104, 2020. 2

[25] Zhenhua Wang, Jiajun Meng, Dongyan Guo, Jianhua Zhang, Javen Qinfeng Shi, and Shengyong Chen, "Consistency-aware graph network for human interaction understanding," in *International Conference on Computer Vision*, 2021. 2

[26] Harold W Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. 2

[27] Mengye Ren and Richard S Zemel, "End-to-end instance segmentation with recurrent attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6656–6664. 2

[28] Bernardino Romera-Paredes and Philip Hilaire Sean Torr, "Recurrent instance segmentation," in *European conference on computer vision*. Springer, 2016, pp. 312–329. 2

[29] Jie Hu, Liujuan Cao, Yao Lu, ShengChuan Zhang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji, "Istr: End-to-end instance segmentation via transformers," *arXiv preprint arXiv:2105.00637*, 2021. 2, 4

[30] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019. 3

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. 2009, pp. 248–255, IEEE Computer Society. 3

[32] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019, OpenReview.net. 3

[33] Zhi Tian, Chunhua Shen, and Hao Chen, "Conditional convolutions for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 4

[34] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8573–8581. 4