SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation

Jiale Cao[1], Rao Muhammad Anwer[2,3], Hisham Cholakkal[2,3], Fahad Shahbaz Khan[2,3], Yanwei Pang[1(✉)], and Ling Shao[2,3]

[1] Tianjin University, Tianjin, China
{connor,pyw}@tju.edu.cn
[2] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
{rao.anwer,hisham.cholakkal,fahad.khan,ling.shao}@mbzuai.ac.ae
[3] Inception Institute of Artificial Intelligence, Abu Dhabi, UAE

Abstract. Single-stage instance segmentation approaches have recently gained popularity due to their speed and simplicity, but are still lagging behind in accuracy, compared to two-stage methods. We propose a fast single-stage instance segmentation method, called SipMask, that preserves instance-specific spatial information by separating mask prediction of an instance to different sub-regions of a detected bounding-box. Our main contribution is a novel light-weight spatial preservation (SP) module that generates a separate set of spatial coefficients for each sub-region within a bounding-box, leading to improved mask predictions. It also enables accurate delineation of spatially adjacent instances. Further, we introduce a mask alignment weighting loss and a feature alignment scheme to better correlate mask prediction with object detection. On COCO `test-dev`, our SipMask outperforms the existing single-stage methods. Compared to the state-of-the-art single-stage TensorMask, SipMask obtains an absolute gain of 1.0% (mask AP), while providing a four-fold speedup. In terms of real-time capabilities, SipMask outperforms YOLACT with an absolute gain of 3.0% (mask AP) under similar settings, while operating at comparable speed on a Titan Xp. We also evaluate our SipMask for real-time video instance segmentation, achieving promising results on YouTube-VIS dataset. The source code is available at https://github.com/JialeCao001/SipMask.

Keywords: Instance segmentation · Real-time · Spatial preservation

1 Introduction

Instance segmentation aims to classify each pixel in an image into an object category. Different from semantic segmentation [6,10,32,34,39], instance segmentation also differentiates multiple object instances. Modern instance segmentation methods typically adapt object detection frameworks, where bounding-box

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12359, pp. 1–18, 2020.
https://doi.org/10.1007/978-3-030-58568-6_1

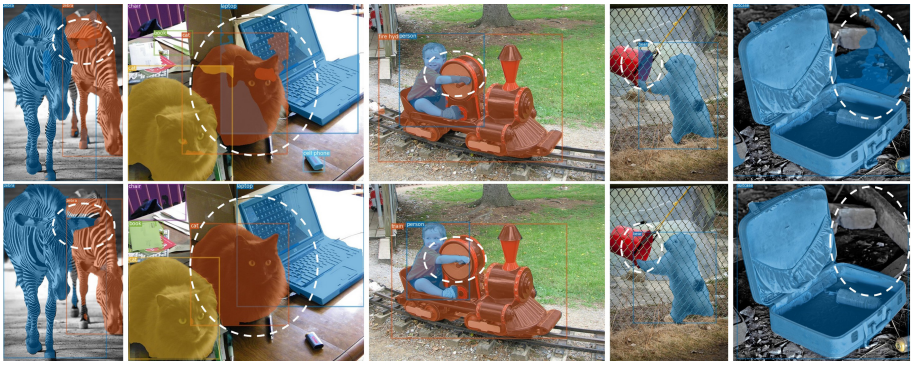


Fig. 1. Instance segmentation examples using YOLACT [2] (top) and our approach (bottom). YOLACT struggles to accurately delineate spatially adjacent instances. Our approach with novel spatial coefficients addresses this issue (marked by white dotted region) by preserving spatial information in bounding-box. The spatial coefficients split mask prediction into multiple sub-mask predictions, leading to improved mask quality.

detection is first performed, followed by segmentation inside each of detected bounding-boxes. Instance segmentation approaches can generally be divided into two-stage [8,17,21,23,31] and single-stage [2,13,36,37,42,47] methods, based on the underlying detection framework. Two-stage methods typically generate multiple object proposals in the first stage. In the second stage, they perform feature pooling operations on each proposal, followed by box regression, classification, and mask prediction. Different from two-stage methods, single-stage approaches do not require proposal generation or pooling operations and employ dense predictions of bounding-boxes and instance masks. Although two-stage methods dominate accuracy, they are generally slow, which restricts their usability in real-time applications.

As discussed above, most single-stage methods are inferior in accuracy, compared to their two-stage counterparts. A notable exception is the single-stage TensorMask [11], which achieves comparable accuracy to two-stage methods. However, TensorMask achieves this accuracy at the cost of reduced speed. In fact, TensorMask [11] is slower than several two-stage methods, including Mask R-CNN [21]. Recently, YOLACT [2] has shown to achieve an optimal tradeoff between speed and accuracy. On the COCO benchmark [29], the single-stage YOLACT operates at real-time (33 frames per second), while obtaining competitive accuracy. YOLACT achieves real-time speed mainly by avoiding proposal generation and feature pooling head networks that are commonly employed in two-stage methods. While operating at real-time, YOLACT still lags behind modern two-stage methods (*e.g.*, Mask R-CNN [21]), in terms of accuracy.

In this work, we argue that one of the key reasons behind sub-optimal accuracy of YOLACT is the loss of spatial information within an object (bounding-box). We attribute this loss of spatial information due to the utilization of a *single*

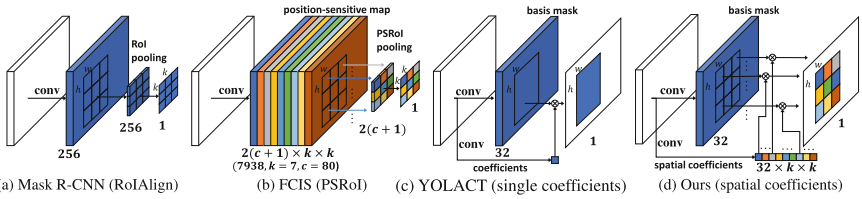
set of object-aware coefficients to predict the whole mask of an object. As a result, it struggles to accurately delineate spatially adjacent object instances (Fig. 1). To address this issue, we introduce an approach that comprises a novel computationally efficient spatial preservation (SP) module to preserve spatial information in a bounding-box. Our SP module predicts object-aware *spatial* coefficients that splits mask prediction into multiple sub-mask predictions, thereby enabling improved delineation of spatially adjacent objects (Fig. 1).

Contributions: We propose a fast anchor-free single-stage instance segmentation approach, called SipMask, with the following contributions.

- We propose a novel light-weight spatial preservation (SP) module that preserves the spatial information within a bounding-box. Our SP module generates a separate set of *spatial* coefficients for each bounding-box sub-region, enabling improved delineation of spatially adjacent objects.
- We introduce two strategies to better correlate mask prediction with object detection. First, we propose a mask alignment weighting loss that assigns higher weights to the mask prediction errors occurring at accurately detected boxes. Second, a feature alignment scheme is introduced to improve the feature representation for both box classification and spatial coefficients.
- Comprehensive experiments are performed on COCO benchmark [29]. Our single-scale inference model based on ResNet101-FPN backbone outperforms state-of-the-art single-stage TensorMask [11] in terms of *both* mask accuracy (absolute gain of 1.0% on COCO `test-dev`) and speed (four-fold speedup). Compared with real-time YOLACT [2], our SipMask provides an absolute gain of 3.0% on COCO `test-dev`, while operating at comparable speed.
- The proposed SipMask can be extended to single-stage video instance segmentation by adding a fully-convolutional branch for tracking instances across video frames. On YouTube-VIS dataset [48], our single-stage approach achieves favourable performance while operating at real-time (30 fps).

2 Related Work

Deep learning has achieved great success in a variety of computer vision tasks [12,20,24,25,35,43–45,52,53]. Existing instance segmentation methods either follow bottom-up [1,19,26,30,33] or top-down [2,8,21,31,36] paradigms. Modern instance segmentation approaches typically follow top-down paradigm where the bounding-boxes are first detected and second segmented. The top-down approaches are divided into two-stage [8,17,21,23,31] and single-stage [2,13,36, 42,47] methods. Among these two-stage methods, Mask R-CNN [21] employs a proposal generation network (RPN) and utilizes RoIAlign feature pooling strategy (Fig. 2(a)) to obtain a fixed-sized features of each proposal. The pooled features are used for box detection and mask prediction. A position sensitive feature pooling strategy, PSRoI [15] (Fig. 2(b)), is proposed in FCIS [27]. PANet [31] proposes an adaptive feature pooling that allows each proposal to access information from multiple layers of FPN. MS R-CNN [23] introduces an additional branch



(a) Mask R-CNN (RoIAlign) (b) FCIS (PSRoI) (c) YOLACT (single coefficients) (d) Ours (spatial coefficients)

Fig. 2. On the left (a and b), feature pooling strategies employed in Mask R-CNN [21] and FCIS [27] resize the feature map to a fixed resolution. Instead, both YOLACT [2] (c) and our approach (d) do not utilize any pooling operation and obtain mask prediction by a simple linear combination of basis mask and coefficient. Mask R-CNN is computationally expensive (*conv* and *deconv* operations after RoIAlign), whereas FCIS is memory demanding due to large number of channels in position-sensitive maps. Both YOLACT and our approach reduce the computational and memory complexity. However, YOLACT uses a single set of coefficients for a detected box, thereby ignoring the spatial information within a box. Our approach preserves the spatial information of an instance by using separate set of spatial coefficients for $k \times k$ sub-regions within a box.

to predict mask quality (mask-IoU). MS R-CNN performs a mask confidence rescoring without improving mask quality. In contrast, our mask alignment loss aims to improve mask quality at accurate detections.

Different to two-stage methods, single-stage approaches [2,13,42,47] typically aim at faster inference speed by avoiding proposal generation and feature pooling strategies. However, most single-stage approaches are generally inferior in accuracy compared to their two-stage counterparts. Recently, YOLACT [2] obtains an optimal tradeoff between accuracy and speed by predicting a dictionary of category-independent maps (basis masks) for an image and a single set of instance-specific coefficients. Despite its real-time capabilities, YOLACT achieves inferior accuracy compared to two-stage methods. Different to YOLACT, which has a single set of coefficients for each bounding-box (Fig. 2(c)), our novel SP module aims at preserving spatial information within a bounding-box. The SP module generates multiple sets of spatial coefficients that splits mask prediction into different sub-regions in a bounding-box (Fig. 2(d)). Further, SP module contains a feature alignment scheme that improves feature representation by aligning the predicted instance mask with detected bounding-box. Our SP module is different to feature pooling strategies, such as PSRoI [27] in several ways. Instead of pooling features into a fixed size ($k \times k$), we perform a simple linear combination between spatial coefficients and basis masks without any feature resizing operation. This preservation of feature resolution is especially suitable for large objects. PSRoI pooling (Fig. 2(b)) generates feature maps of $2(c+1)\times k\times k$ channels, where k is the pooled feature size and c is the number of classes. In practice, such a pooling operation is memory expensive (7938 channels for $k = 7$ and $c = 80$). Instead, our design is memory efficient since the basis masks are of only 32 channels for whole image and the spatial coefficients are a 32 dimensional vector for each sub-region of a bounding-box (Fig. 2(d)). Further, compared to contemporary work [7] using RoIpool based feature maps, our

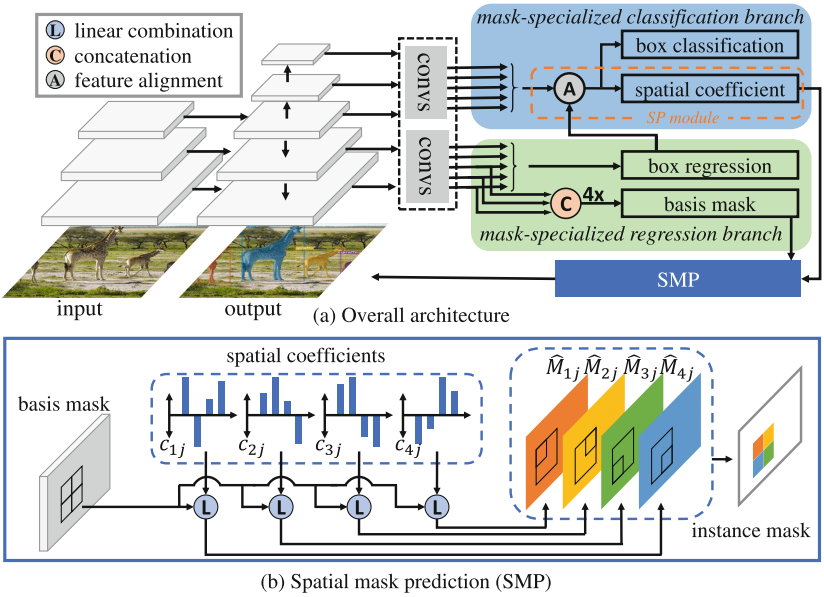


Fig. 3. (a) Overall architecture of our SipMask comprising fully convolutional mask-specialized classification (Sect. 3.1) and regression (Sect. 3.2) branches. The focus of our design is the introduction of a novel spatial preservation (SP) module in the mask-specialized classification branch. The SP module performs two-tasks: feature alignment and spatial coefficients generation. In our approach, a separate set of spatial coefficients are generated for each predicted bounding-box. These spatial coefficients are designed to preserve the spatial information within an object instance, thereby enabling improved delineation of spatially adjacent objects. The mask-specialized regression branch predicts both bounding-box offsets and a set of category-independent basis masks. The basis masks are generated by capturing contextual information from different prediction layers of FPN. (b) Both the basis masks and spatial coefficients along with predicted bounding-box locations are then input to our spatial mask prediction (SMP) module (Sect. 3.3) for predicting the final instance mask.

approach utilizes fewer coefficients on original basis mask. Moreover, our SipMask can be adapted for real-time single-stage video instance segmentation.

3 Method

Overall Architecture: Figure 3(a) shows the overall architecture of our single-stage anchor-free method, SipMask, named for its instance-specific spatial information preservation characteristic. Our architecture is built on FCOS detection method [40], due to its flexible anchor-free design. In the proposed architecture, we replace the standard classification and regression in FCOS with our mask-specialized regression and classification branches. Both mask-specialized classification and regression branches are fully convolutional. Our mask-specialized

classification branch predicts the classification scores of detected bounding-boxes and generates instance-specific spatial coefficients for instance mask prediction. The focus of our design is the introduction of a novel spatial preservation (SP) module, within the mask-specialized classification branch, to obtain improved mask predictions. Our SP module further enables better delineation of spatially adjacent objects. The SP module first performs feature alignment by using the final regressed bounding-box locations. The resulting aligned features are then utilized for both box classification and generating spatial coefficients required for mask prediction. The spatial coefficients are introduced to preserve spatial information within an object bounding-box. In our framework, we divide the bounding-box into $k \times k$ sub-regions and compute a separate set of spatial coefficients for each sub-region. Our mask-specialized regression branch generates both bounding-box offsets for each instance and a set of category-independent maps, termed as basis masks, for an image. Our basis masks are constructed by capturing the contextual information from different prediction layers of FPN.

The spatial coefficients predicted for each of $k \times k$ sub-regions within a bounding-box along with image-specific basis masks are utilized in our spatial mask prediction (SMP) module (Fig. 3(b)). Our SMP generates separate map predictions for respective regions within the bounding-box. Consequently, these separate map predictions are combined to obtain final instance mask prediction.

3.1 Spatial Preservation Module

Besides box classification, our mask-specialized classification branch comprises a novel spatial preservation (SP) module. Our SP module performs two tasks: spatial coefficients generation and feature alignment. The spatial coefficients are introduced to improve mask prediction by preserving spatial information within a bounding-box. Our feature alignment scheme aims at improving the feature representation for both box classification and spatial coefficients generation.

Spatial Coefficients Generation: As discussed earlier, the recently introduced YOLACT [2] utilizes a single set of coefficients to predict the whole mask of an object, leading to the loss of spatial information within a bounding-box. To address this issue, we propose a simple but effective approach that splits mask prediction into multiple sub-mask predictions. We divide the spatial regions within a predicted bounding-box into $k \times k$ sub-regions. Instead of predicting a *single set of coefficients* for the whole bounding-box j, we predict a *separate set of spatial coefficients* $c_{ij} \in R^m$ for each of its sub-region i. Figure 3(b) shows an example where a bounding-box is divided into 2×2 sub-regions (four quadrants, *i.e.*, top-left, top-right, bottom-left and bottom-right). In practice, we observe that $k = 2$ provides an optimal tradeoff between speed and accuracy. Note that our spatial coefficients utilize improved features obtained though a feature alignment operation described next.

Feature Alignment Scheme: Generally, convolutional layer operates on a rectangular grid (*e.g.*, 3×3 kernel). Thus, the extracted features for classification and coefficients generation may fail to align with the features of regressed

bounding-box. Our feature alignment scheme addresses this issue by aligning the features with regressed box location, resulting in an improved feature representation. For feature alignment, we introduce a deformable convolutional layer [5,16,51] in our mask-specialized classification branch. The input to the deformable convolutional layer are the regression offsets to left, right, top, and bottom corners of ground-truth bounding-box obtained from mask-specialized regression branch (Sect. 3.2). These offsets are utilized to estimate the kernel offset Δp_r that augments the regular sampling grid G in the deformable convolution operator, resulting in an aligned feature $y(p_0)$ at position p_0, as follows:

$$y(p_0) = \sum_{i \in G} w_r \cdot x(p_0 + p_r + \Delta p_r), \tag{1}$$

where x is the input feature, and p_r is the original position of convolutional weight w_r in G. Different to [50,51] that aim to learn accurate geometric localization, our approach aims to generate better features for box classification and coefficient generation. Next, we describe mask-specialized regression branch.

3.2 Mask-Specialized Regression Branch

Our mask-specialized regression branch performs box regression and generates a set of category-independent basis masks for an image. Note that YOLACT utilizes a single FPN prediction layer to generate the basis masks. Instead, the basis masks in our SipMask are generated by exploiting the multi-layer information from different prediction layers of FPN. The incorporation of multi-layer information helps to obtain a continuous mask (especially on large objects) and remove background clutter. Further, it helps in scenarios, such as partial occlusion and large-scale variation. Here, objects of various sizes are predicted at different prediction layers of the FPN (*i.e.*, $P3 - P7$). To capture multi-layer information, the features from the $P3 - P5$ layers of the FPN are utilized to generate basis masks. Note that $P6$ and $P7$ are excluded for basis mask generation to reduce the computational cost. The outputs from $P4$ and $P5$ are first upsampled to the resolution of $P3$ using bilinear interpolation. The resulting features from all three prediction layers ($P3 - P5$) are concatenated, followed by a 3×3 convolution to generate feature maps with m channels. Finally, these feature maps are upsampled four times by using bilinear interpolation, resulting in m basis masks, each having a spatial resolution of $h \times w$. Both the spatial coefficients (Sect. 3.1) and basis masks are utilized in our spatial mask prediction (SMP) module for final instance mask prediction.

3.3 Spatial Mask Prediction Module

Given an input image, our spatial mask prediction (SMP) module takes the predicted bounding-boxes, basis masks and spatial coefficients as inputs and predicts the final instance mask. Let $B \in R^{h \times w \times m}$ represent m predicted basis masks for the whole image, p be the number of predicted boxes, and C_i be a $m \times p$

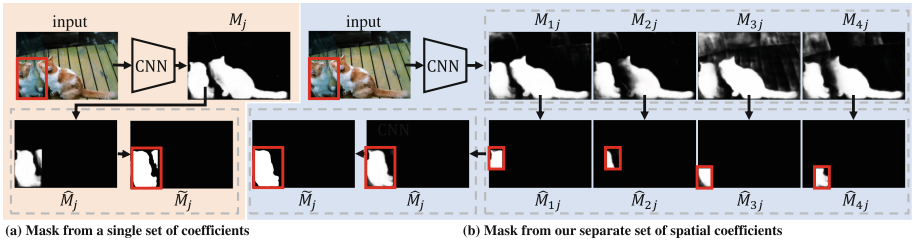


Fig. 4. A visual comparison between mask generation using (a) a single set of coefficients, as in YOLACT and (b) our SipMask. For simplicity, only one detected 'cat' instance and its corresponding mask generation procedure is shown here. A linear combination of single set of coefficients and basis masks leads to one map M_j. Then, the map M_j is pruned followed by thresholding to produce the final mask $\tilde{M}_j$. Instead, our SipMask generates a separate set of spatial coefficients for each sub-region (quadrant for $k = 2$) within a bounding-box. As a result, a separate set of spatial map M_{ij} is obtained for each quadrant i in the bounding-box j. Afterwards, these spatial maps are first pruned and then integrated (a simple addition) followed by thresholding to obtain final mask $\tilde{M}_j$. Our SipMask is able to reduce the influence of the adjacent object ('cat') instance, resulting in improved mask prediction.

matrix that indicates the spatial coefficients at the i^{th} sub-region (quadrant for $k = 2$) of all p predicted bounding-boxes. Note that the column j of C_i (*i.e.*, $c_{ij} \in R^m$) indicates the spatial coefficients for the bounding-box j (Sect. 3.1). We perform a simple matrix multiplication between C_i and B to obtain p maps corresponding to the i^{th} quadrant of all bounding-boxes as follows.

$$M_i = \sigma(B \times C_i) \quad \forall i \in [1, 4], \tag{2}$$

where σ is sigmoid normalization and $M_i \in R^{h \times w \times p}$ are the maps generated for the i^{th} quadrant of all p bounding-boxes. Figure 4(b) shows the procedure to obtain final mask of an instance j. Let $M_{ij} \in R^{h \times w}$ be the map generated for the i^{th} quadrant of a bounding-box j. Then, the response values of M_{ij} outside the i^{th} quadrant of the box j are set as zero for generating a pruned map $\hat{M}_{ij}$. To obtain the instance map $\hat{M}_j$ of a bounding-box j, we perform a simple addition of its pruned maps obtained from all four quadrants, *i.e.*, $\hat{M}_j = \sum_{i=1}^{4} \hat{M}_{ij}$. Finally, the instance map at the predicted bounding-box region is binarized with a fixed threshold to obtain final mask $\tilde{M}_j$ of instance j.

Figure 4 shows a visual comparison of a single set of coefficients based mask prediction, as in YOLACT, with our separate set of spatial coefficients (for each sub-region) based mask prediction. The top-left pixels of an adjacent 'cat' instance are appearing inside the top-right quadrant of the detected 'cat' instance bounding-box (in red). In Fig. 4(a), a linear combination of a single set of instance-specific coefficients and image-level basis masks is used to obtain a map M_j. The response values of the map M_j outside the box j are assigned with zero to produce a pruned mask $\hat{M}_j$, followed by thresholding to obtain the final mask $\tilde{M}_j$. Instead, our SipMask (Fig. 4(b)) generates a separate set of instance-specific

spatial coefficients for each sub-region i within a bounding-box j. By separating the mask predictions to different sub-regions of a box, our SipMask reduces the influence of adjacent (overlapping) object instance in final mask prediction.

3.4 Loss Function

The overall loss function of our framework contains loss terms corresponding to bounding-box detection (classification and regression) and mask generation. For box classification L_{cls} and box regression L_{reg}, we utilize focal loss and IoU loss, respectively, as in [40]. For mask generation, we introduce a novel mask alignment weighting loss L_{mask} that better correlate mask predictions with high quality bounding-box detections. Different to YOLACT that utilizes a standard pixel-wise binary cross entropy (BCE) loss during training, our L_{mask} improves the BCE loss with a mask alignment weighting scheme that assigns higher weights to the masks $\tilde{M}_j$ obtained from high quality bounding-box detections.

Mask Alignment Weighting: In our mask alignment weighting, we first compute the overlap o_j between a predicted bounding-box j and the corresponding ground-truth. The weighting factor α_j is then obtained by multiplying the overlap o_j and the classification score s_j of the bounding-box j. Here, a higher α_j indicates good quality bounding-box detections. Consequently, α_j is used to weight the mask loss l^j of the instance j, leading to $L_{mask} = \frac{1}{N}\sum_j l^j \times \alpha_j$. Here, N is the number of bounding-boxes. Our weighting strategy encourages the network to predict a high quality instance mask for a high quality bounding-box detections. The proposed mask alignment weighting loss L_{mask} is utilized along with loss terms corresponding to bounding-box detection (classification and regression) in our overall loss function: $L = L_{reg} + L_{cls} + L_{mask}$.

3.5 Single-Stage Video Instance Segmentation

In addition to still image instance segmentation, we investigate our single-stage SipMask for the problem of real-time video instance segmentation. In video instance segmentation, the aim is to simultaneously detect, segment, and track instances in videos.

To perform real-time single-stage video instance segmentation, we simply extend our SipMask by introducing an additional fully-convolutional branch in parallel to mask-specialized classification and regression branches for instance tracking. The fully-convolutional branch consists of two convolutional layers. After that, the output feature maps of different layers in this branch are fused to obtain the tracking feature maps, similar to basis mask generation in our mask-specialized regression branch. Different from the state-of-the-art MaskTrack R-CNN [48] that utilizes RoIAlign and fully-connected operations, our SipMask extracts a tracking feature vector from the tracking feature maps at the bounding-box center to represent each instance. The metric for matching the instances between different frames is similar to MaskTrack R-CNN. Our SipMask is very simple, efficient and achieves favourable performance for video instance segmentation (Sect. 4.4).

4 Experiments

4.1 Dataset and Implementation Details

Dataset: We conduct experiments on COCO dataset [29], where the `trainval` set has about $115k$ images, the `minival` set has $5k$ images, and the `test-dev` set has about $20k$ images. We perform training on `trainval` set and present state-of-the-art comparison on `test-dev` set and the ablations on `minival` set.

Implementation Details: We adopt ResNet [22] (ResNet50/ResNet101) with FPN pre-trained on ImageNet [38] as the backbone. Our method is trained eight GPUs with SGD for optimization. During training, the initial learning rate is set to 0.01. When conducting ablation study, we use a $1\times$ training scheme at single scale to reduce training time. For a fair comparison with the state-of-the-art single-stage methods [2,11], we follow the $6\times$, multi-scale training scheme. During inference we select top 100 bounding-boxes with highest classification scores, after NMS. For these bounding-boxes, a simple linear combination between the predicted spatial coefficients and basis masks are used to obtain instance masks.

4.2 State-of-the-art Comparison

Here, we compare our method with some two-stage [4,8,9,14,18,21,23,27,31] and single-stage [2,11,46,54] methods on COCO `test-dev` set. Table 1 shows the comparison in terms of both speed and accuracy. Most existing methods use a larger input image size, typically $\sim 1333 \times 800$ (except YOLACT [2], which operates on input size of 550×550). Among existing two-stage methods, Mask R-CNN [21] and PANet [31] achieve overall mask AP scores of 35.7 and 36.6, respectively. The recently introduced MS R-CNN [21] and HTC [8] obtain mask AP scores of 38.3 and 39.7, respectively. Note that HTC achieves this improved accuracy at the cost of a significant reduction in speed. Further, most two-stage approaches require more than 100 milliseconds (ms) to process an image.

In case of single-stage methods, PolarMask [46] obtains a mask AP of 30.4. RDSNet [42] achieves a mask AP score of 36.4. Among these single-stage methods, TensorMask [11] obtains the best results with a mask AP score of 37.1. Our SipMask under similar settings (input size and backbone) outperforms TensorMask with an absolute gain of 1.0%, while obtaining a four-fold speedup. In particular, our SipMask achieves an absolute gain of 2.7% on the large objects, compared to TensorMask.

In terms of fast instance segmentation and real-time capabilities, we compare our SipMask with YOLACT [2] when using two different backbone models (ResNet50/ResNet101 FPN). Compared to YOLACT, our SipMask achieves an absolute gain of 3.0% without any significant reduction in speed (YOLACT: 30 ms vs. SipMask: 32 ms). A recent variant of YOLACT, called YOLACT++ [3], utilizes a deformable backbone (ResNet101-Deform [55] with interval 3) and a mask scoring strategy. For a fair comparison, we also integrate the same two ingredients in our SipMask, called as SipMask++. When using a similar input size and same backbone, our SipMask++ achieves improved mask accuracy while

Table 1. State-of-the-art instance segmentation comparison in terms of accuracy (mask AP) and speed (inference time) on COCO `test-dev` set. All results are based on single-scale test and speeds are reported on a single Titan Xp GPU (except TensorMask and RDSNet that are reported on Tesla V100). When using the same large input size ($\sim$1333 $\times$ 800) and backbone, our SipMask outperforms all existing single-stage methods in terms of accuracy. Further, our SipMask obtains a four-fold speedup over the TensorMask. When using a similar small input size ($\sim$550 $\times$ 550), our SipMask++ achieves superior performance while operating at comparable speed, compared to the YOLACT++. In terms of real-time capabilities, our SipMask consistently improves the mask accuracy without any significant reduction in speed, compared to the YOLACT.

Method	Backbone	Input size	Time	AP	AP@0.5	AP@0.75	AP_s	AP_m	AP_l
Two-Stage									
MNC [14]	ResNet101-C4	$\sim$1333 $\times$ 800	-	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [27]	ResNet101-C5	$\sim$1333 $\times$ 800	152	29.2	49.5	-	7.1	31.3	50.0
RetinaMask [18]	ResNet101-FPN	$\sim$1333 $\times$ 800	167	34.7	55.4	36.9	14.3	36.7	50.5
MaskLab [9]	ResNet101	$\sim$1333 $\times$ 800	-	35.4	57.4	37.4	16.9	38.3	49.2
Mask R-CNN [21]	ResNet101-FPN	$\sim$1333 $\times$ 800	**116**	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN* [21]	ResNet101-FPN	$\sim$1333 $\times$ 800	116	38.3	61.2	40.8	18.2	40.6	54.1
PANet [31]	ResNet50-FPN	$\sim$1333 $\times$ 800	212	36.6	58.0	39.3	16.3	38.1	53.1
MS R-CNN [23]	ResNet101-FPN	$\sim$1333 $\times$ 800	117	38.3	58.8	41.5	17.8	40.4	**54.4**
HTC [8]	ResNet101-FPN	$\sim$1333 $\times$ 800	417	39.7	**61.8**	43.1	21.0	42.2	53.5
D2Det [4]	ResNet101-FPN	$\sim$1333 $\times$ 800	168	**40.2**	61.5	**43.7**	**21.7**	**43.0**	54.0
Single-Stage: *Large input size*									
PolarMask [46]	ResNet101-FPN	$\sim$1333 $\times$ 800	-	30.4	51.9	31.0	13.4	32.4	42.8
RDSNet [42]	ResNet101-FPN	$\sim$1333 $\times$ 800	113	36.4	57.9	39.0	16.4	39.5	51.6
TensorMask [11]	ResNet101-FPN	$\sim$1333 $\times$ 800	380	37.1	59.3	39.4	17.1	39.1	51.6
Our SipMask	ResNet101-FPN	$\sim$1333 $\times$ 800	**89**	**38.1**	**60.2**	**40.8**	**17.8**	**40.8**	**54.3**
Single-Stage: *Small input size*									
YOLACT++ [3]	ResNet101-Deform	550 $\times$ 550	**37**	34.6	53.8	36.9	**11.9**	36.8	55.1
Our SipMask++	ResNet101-Deform	544 $\times$ 544	**37**	**35.4**	**55.6**	**37.6**	11.2	**38.3**	**56.8**
Real-Time									
YOLACT [2]	ResNet50-FPN	550 $\times$ 550	**22**	28.2	46.6	29.2	9.2	29.3	44.8
Our SipMask	ResNet50-FPN	544 $\times$ 544	24	31.2	51.9	32.3	9.2	33.6	49.8
YOLACT [2]	ResNet101-FPN	550 $\times$ 550	30	29.8	48.5	31.2	**9.9**	31.3	47.7
Our SipMask	ResNet101-FPN	544 $\times$ 544	32	**32.8**	**53.4**	**34.3**	9.3	**35.6**	**54.0**

operating at the same speed, compared to YOLACT++. Figure 5 shows example instance segmentation results of our SipMask on COCO `test-dev`.

4.3 Ablation Study

We perform an ablation study on COCO `minival` set with ResNet50-FPN backbone [28]. First, we show the impact of progressively integrating our different components: spatial preservation (SP) module (Sect. 3.1), contextual basis masks (CBM) obtained by integrating context information from different FPN prediction layers (Sect. 3.2), and mask alignment weighting loss (WL) (Sect. 3.4), to the baseline. Note that our baseline is similar to YOLACT, obtaining the basis masks by using only high-resolution FPN layer ($P3$) and using a single set of coefficients for mask prediction. The results are presented in Table 2. The

Fig. 5. Qualitative results on COCO `test-dev` [29] (corresponding to our 38.1 mask AP). Each color represents different object instances in an image. Our SipMask generates high quality instance segmentation masks in challenging scenarios. (Color figure online)

Table 2. Impact of progressively integrating (from left to right) different components into the baseline. All our components (SP, CBM and WL) contribute towards achieving improved mask AP.

Baseline	SP	CBM	WL	AP
✓				31.2
✓	✓			33.4
✓	✓	✓		33.8
✓	✓	✓	✓	34.3

Table 3. Impact of integrating different components individually into the baseline. Our spatial coefficients (SC) obtains the most improvement in accuracy.

Baseline	SC	FA	CBM	WL	AP
✓					31.2
✓	✓				32.9
✓		✓			31.7
✓			✓		31.9
✓				✓	32.0

baseline achieves a mask AP of 31.2. All our components (SP, CBM and WL) contribute towards achieving improved performance (mask accuracy). In particular, the most improvement in mask accuracy, over the baseline, comes from our SP module. Our final SipMask integrating all contributions obtains an absolute gain of 3.1% in terms of mask AP, compared to the baseline. We also evaluate the impact of adding our different components individually to the baseline. The results are shown in Table 3. Among these components, the spatial coefficients provides the most improvement in accuracy over the baseline. It is worth mentioning that both the spatial coefficients and feature alignment constitute our spatial preservation (SP) module. These results suggest that each of our components individually contributes towards improving the final performance.

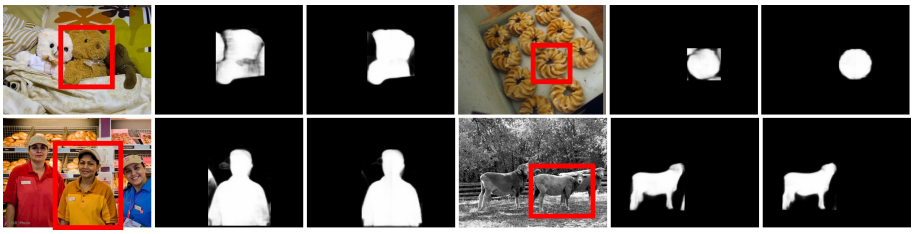


Fig. 6. Qualitative results highlighting the spatial delineation capabilities of our spatial preservation (SP) module. Input image with a detected bounding-box (red) is shown in column 1 and 4. Mask prediction obtained by the baseline that is based on a single set of coefficients is shown in column 2 and 5. Mask prediction obtained by our approach that is based on a separate set of spatial coefficients in a bounding-box is shown in column 3 and 6. Compared to the baseline, our approach is able to better delineate spatially adjacent object instances, leading to improved mask predictions. (Color figure online)

Table 4. The effect of varying the number of sub-regions to compute spatial coefficients. A separate set of spatial coefficients are generated for each sub-region.

	1×1	1×2	2×1	2×2	3×3	4×4
AP	31.2	32.2	32.1	32.9	33.1	33.1

Table 5. The effect of classification (class confidences) and localization (ground-truth overlap) scores on our mask alignment weighting loss (cls. + loc.).

	Baseline	Only cls	Only loc	cls.+loc
AP	31.2	31.8	31.7	32.0

Figure 6 shows example results highlighting the spatial delineation capabilities of our spatial preservation (SP) module. We show the input image with the detected bounding-box (red) together with the mask prediction based on a single set of coefficients (baseline) and our mask prediction based on a separate set of spatial coefficients. Our approach is able to provide improved delineation of spatially adjacent instances, leading to superior mask predictions.

As discussed in Sect. 3.1, our SP module generates a separate set of spatial coefficients for each sub-region within a bounding-box. Here, we perform a study by varying the number of sub-regions to obtain spatial coefficients. Table 4 shows that a large gain in performance is obtained going from 1×1 to 2×2. We also observe that the performance tends to marginally increase by further increasing the number of sub-regions. In practice, we found 2×2 to provide an optimal tradeoff between speed and accuracy. As discussed earlier (Sect. 3.4), our mask alignment weighting loss re-weights the pixel-level BCE loss using both classification (class scores) and localization (overlap with the ground-truth) information. Here, we analyze the effect of classification (only cls.) and localization (only loc.) on our mask alignment weighting loss in Table 5. It shows that both the classification and localization are useful to re-weight the BCE loss for improved mask prediction.

Table 6. Comparison with state-of-the-art video instance segmentation methods on YouTube-VIS validation set. Results are reported in terms of mask accuracy and recall.

Method	Category	AP	AP@0.5	AP@0.75	AR@1	AR@10
OSMN [49]	Mask propagation	23.4	36.5	25.7	28.9	31.1
FEELVOS [41]	Mask propagation	26.9	42.0	29.7	29.9	33.4
OSMN [49]	Track-by-detect	27.5	45.1	29.1	28.6	31.1
MaskTrack R-CNN [48]	Track-by-detect	30.3	51.1	32.6	31.0	35.5
Our SipMask	Track-by-detect	32.5	53.0	33.3	33.5	38.9
Our SipMask *ms-train*	Track-by-detect	**33.7**	**54.1**	**35.8**	**35.4**	**40.1**

Fig. 7. Qualitative results on example frames of different videos from Youtube-VIS validation set [48]. The object with same predicted identity has same color.

4.4 Video Instance Segmentation Results

In addition to instance segmentation, we present the effectiveness of our SipMask, with the proposed modifications described in Sect. 3.5, for real-time video instance segmentation. We conduct experiments on the recently introduced large-scale YouTube-VIS dataset [48]. The YouTube-VIS dataset contains 2883 videos, 4883 objects, 131k instance masks, and 40 object categories. Table 6 shows the state-of-the-art comparison on the YouTube-VIS validation set. When using the same input size (640×360) and backbone (ResNet50 FPN), our SipMask outperforms the state-of-the-art MaskTrack R-CNN [48] with an absolute gain of 2.2% in terms of mask accuracy (AP). Further, our SipMask achieves impressive mask accuracy while operating at real-time (30 fps) on a Titan Xp. Figure 7 shows video instance segmentation results on example frames from the validation set.

5 Conclusion

We introduce a fast single-stage instance segmentation method, SipMask, that aims at preserving spatial information within a bounding-box. A novel light-weight spatial preservation (SP) module is designed to produce a separate set

of spatial coefficients by splitting mask prediction of an object into different sub-regions. To better correlate mask prediction with object detection, a feature alignment scheme and a mask alignment weighting loss are further proposed. We also show that our SipMask is easily extended for real-time video instance segmentation. Our comprehensive experiments on COCO dataset show the effectiveness of the proposed contributions, leading to state-of-the-art single-stage instance segmentation performance. With the same instance segmentation framework and just changing the input resolution (544×544), our SipMask operates at real-time on a single Titan Xp with a mask accuracy of 32.8 on COCO `test-dev`.

This work was supported by National Key R&D Program (2018AAA0102800) and National Natural Science Foundation (61906131, 61632018) of China.

Author contributions. Jiale Cao, Rao Muhammad Anwer

References

1. Arnab, A., Torr, P.H.: Pixelwise instance segmentation with a dynamically instantiated network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
3. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact++: better real-time instance segmentation. arXiv:1912.06218 (2020)
4. Cao, J., Cholakkal, H., Anwer, R.M., Khan, F.S., Pang, Y., Shao, L.: D2det: towards high quality object detection and instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
5. Cao, J., Pang, Y., Han, J., Li, X.: Hierarchical shot detector. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
6. Cao, J., Pang, Y., Li, X.: Triply supervised decoder networks for joint detection and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
7. Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., Yan, Y.: Blendmask: top-down meets bottom-up for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
8. Chen, K., et al.: Hybrid task cascade for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
9. Chen, L.C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: instance segmentation by refining object detection with semantic and direction features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
11. Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: a foundation for dense object segmentation. In: Proceedings of the IEEE International Conference Computer Vision (2019)

12. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
13. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 534–549. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_32
14. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
15. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Proceedings of the Advances in Neural Information Processing Systems (2016)
16. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
17. Fang, H.S., Sun, J., Wang, R., Gou, M., Li, Y.L., Lu, C.: Instaboost: boosting instance segmentation via probability map guided copy-pasting. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
18. Fu, C.Y., Shvets, M., Berg, A.C.: Retinamask: learning to predict masks improves state-of-the-art single-shot detection for free. arXiv:1901.03353 (2019)
19. Gao, N., et al.: SSAP: single-shot instance segmentation with affinity pyramid. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (2014)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE International Conference on Computer Vision (2016)
23. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
24. Jiang, X., et al.: Density-aware multi-task learning for crowd counting. IEEE Trans. Multimedia (2020)
25. Khan, F.S., Xu, J., van de Weijer, J., Bagdanov, A., Anwer, R.M., Lopez, A.: Recognizing actions through action-specific person detection. IEEE Trans. Image Process. **24**(11), 4422–4432 (2015)
26. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: Instancecut: from edges to instances with multicut. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
27. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
28. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
29. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

30. Liu, S., Jia, J., Fidler, S., Urtasun, R.: SGN: sequential grouping networks for instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
31. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
32. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
33. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
34. Pang, Y., Li, Y., Shen, J., Shao, L.: Towards bridging semantic gap to improve semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
35. Pang, Y., Xie, J., Khan, M.H., Anwer, R.M., Khan, F.S., Shao, L.: Mask-guided attention network for occluded pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
36. Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., Zhou, X.: Deep snake for real-time instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
37. Pinheiro, P.O., Lin, T.-Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 75–91. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_5
38. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. (2015)
39. Sun, G., Wang, B., Dai, J., Gool, L.V.: Mining cross-image semantics for weakly supervised semantic segmentation. In: ECCV 2020. Springer, Cham (2020)
40. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
41. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: fast end-to-end embedding learning for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
42. Wang, S., Gong, Y., Xing, J., Huang, L., Huang, C., Hu, W.: RDSNet: a new deep architecture for reciprocal object detection and instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
43. Wang, T., Anwer, R.M., Cholakkal, H., Khan, F.S., Pang, Y., Shao, L.: Learning rich features at high-speed for single-shot object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
44. Wang, T., Yang, T., Danelljan, M., Khan, F.S., Zhang, X., Sun, J.: Learning human-object interaction detection using interaction points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
45. Wu, J., Zhou, C., Yang, M., Zhang, Q., Li, Y., Yuan, J.: Temporal-context enhanced detection of heavily occluded pedestrians. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
46. Xie, E., et al.: Polarmask: single shot instance segmentation with polar representation. arXiv:1909.13226 (2019)

47. Xu, W., Wang, H., Qi, F., Lu, C.: Explicit shape encoding for real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
48. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
49. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
50. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: point set representation for object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
51. Yang, Z., et al.: Reppoints: point set representation for object detection. In: ECCV 2020. Springer, Cham (2020)
52. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.H.: Deep learning for person re-identification: a survey and outlook. arXiv:2001.04193 (2020)
53. Ye, M., Zhang, X., Yuen, P.C., Chang, S.F.: Unsupervised embedding learning via invariant and spreading instance feature. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
54. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
55. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: more deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)