Image and Vision Computing 120 (2022) 104401

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Review article

A review on 2D instance segmentation based on deep neural networks

Wenchao Gu [a], Shuang Bai [a,*], Lingxing Kong [b]

[a] School of Electronic and Information Engineering, Beijing Jiaotong University, No.3 Shang Yuan Cun, Hai Dian District, Beijing, China
[b] State Key Laboratory of Operation and Control of Renewable Energy and Storage Systems, China Electric Power Research Institute, Beijing, China

ARTICLE INFO

Article history:
Received 1 November 2021
Received in revised form 28 January 2022
Accepted 1 February 2022
Available online 5 February 2022

Keywords:
Instance segmentation
Deep neural networks
Computer vision
Review

ABSTRACT

Image instance segmentation involves labeling pixels of images with classes and instances, which is one of the pivotal technologies in many domains, such as natural scenes understanding, intelligent driving, augmented reality and medical image analysis. With the power of deep learning, instance segmentation methods that use this technique have recently achieved remarkable progress. In this survey, we mainly discuss the representative 2D instance segmentation methods based on deep neural networks. Firstly, we summarize current fully-, weakly- and semi-supervised instance segmentation methods, and divide existing fully-supervised methods into three sub-categories depending on the number of stages. Based on our investigation, we conclude that currently, two-stage methods dominate the frontier of general instance segmentation; single-stage methods can achieve a better speed-accuracy trade-off, and multi-stage methods can achieve higher accuracy. Secondly, we introduce eleven datasets and three evaluation metrics for evaluating instance segmentation methods that can help researchers decide which one to choose to meet their needs and goals. Then the innovation and quantitative results of state-of-the-art general instance segmentation methods and specific instance segmentation methods (including salient instance segmentation, person instance segmentation, and amodal instance segmentation) are reviewed. In what follows, the common backbone networks are reviewed to better explain the reasons that why deep neural networks-based instance segmentation methods can achieve excellent performance. Finally, the future research directions and potential applications of instance segmentation are discussed, which can facilitates researchers to realize the existing technical difficulties and recent research hotspots.

Contents

* Corresponding author.
E-mail address: shuangb@bjtu.edu.cn (S. Bai).

https://doi.org/10.1016/j.imavis.2022.104401
0262-8856/

1. Introduction

Image segmentation has been a critical problem in computer vision, which has a considerable impact on many areas, including autonomous driving [1–3], medical systems [4,5] and agricultural analysis [6,7], *etc.* Differentiating from tasks like image classification [8–10,12] and object detection [13,14,15,16] that focus on object-level information, image segmentation can be deemed as a pixel-level classification task. In general, image segmentation task contains three sub-tasks: semantic segmentation [17–19], instance segmentation [20,21] and panoptic segmentation [22,23]. Semantic segmentation is a fine prediction task to label each pixel of an image with a corresponding object class; instance segmentation is designed to identify and segment pixels that belong to each object instance; going further, panoptic segmentation unifies

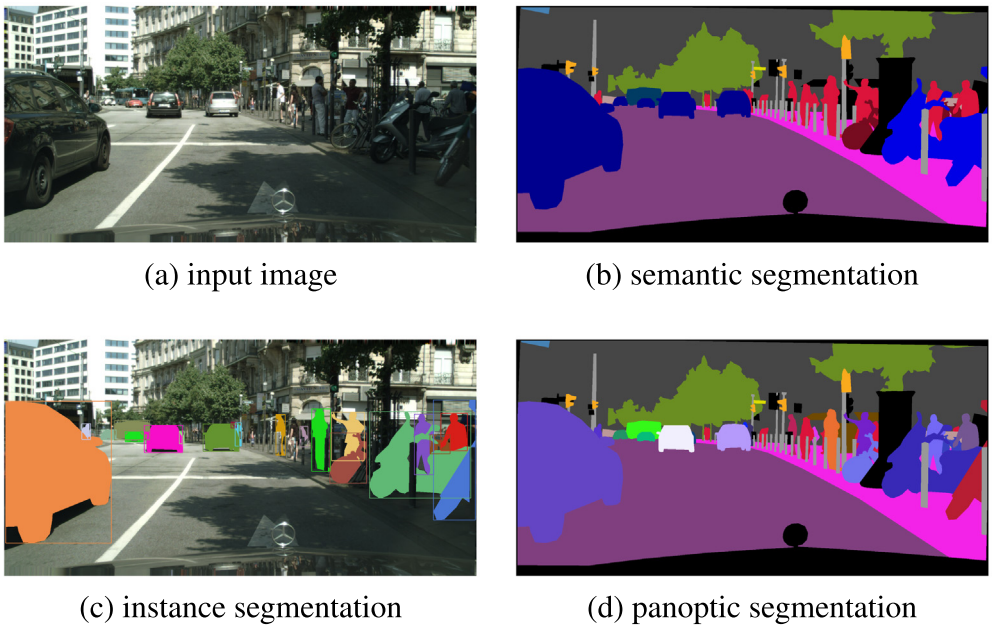


Fig. 1. The difference of three image segmentation tasks: an input image (a) and the result of semantic segmentation (b), instance segmentation (c), and panoptic segmentation (d).

semantic segmentation and instance segmentation such that all pixels are given both a class label and an instance ID. Fig. 1 shows the differences between the three sub-tasks.

In this paper, we will focus on image instance segmentation. To demonstrate the importance of instance segmentation, we will make a comparison among the aforementioned three sub-tasks of image segmentation. Image semantic segmentation, as one of the tasks, has been widely researched with the development of deep neural networks (DNNs) [24,25,26] over the past years. However, semantic segmentation fails to segment different instances of the same semantic category in images, that is the task of instance segmentation. Compared to semantic segmentation, instance segmentation assigns both a semantic label and an instance ID to all pixels to segment object instances [27,28]. Thus, instance segmentation can provide more detailed information of a given image than semantic segmentation, such as *location* and *quantity* of detected objects. Lastly, panoptic segmentation combines them to provide more comprehensive and detailed segmentation results, but instance segmentation is important prior knowledge for implementing it.

Apart from tasks related to 2D images, three-dimensional (3D) scene understanding [29,30] is also a fundamental computer vision problem, which has also been researched and mainly involves cloud points [37–41]. Since the 3D imaging technology is not quite mature and the 3D instance segmentation dataset is scarce now, the 3D instance segmentation methods will not be discussed in this paper. Video instance segmentation is another research direction of instance segmentation task [32,33], which can be formulated as adding a tracking head to the 2D instance segmentation methods. Therefore, we mainly focus on reviewing works on two-dimensional (2D) image instance segmentation in this paper. For simplicity, *instance segmentation* in this paper refers to *2D instance segmentation*.

Although the *anchor* property is very straightforward for classifying instance segmentation methods, as it only focuses on how to locate instances and ignores the way of generating instance masks. According to the number of stages required to achieve positioning object and generation mask, existing fully-supervised instance segmentation methods can be divided into three categories: multi-stage, two-stage and single-stage method. Among them, the object positioning and mask generation in the multi-stage and two-stage instance segmentation methods have a certain sequence of dependence. While object positioning and mask generation can be realized at the same time for the single-stage instance segmentation methods.

Because of the idea of "*detect then segment*", the instance segmentation task is dominated by two-stage methods currently. In particular, Mask R-CNN is the most representative work [21]. Based on the priority of detection and segmentation, two-stage works can be further categorized into two groups: top-down methods and bottom-up methods. As shown in the Fig. 2, top-down methods first predict a bounding box for each object and then generate an instance mask within each bounding box [21,42]. This type of approach is heavily dependent on the detection results and prone to systematic artifacts on overlapping instances [21]. On the other hand, bottom-up approaches associate pixel-level projection with each object instance and adopt a post-processing procedure to distinguish each instance, *e.g. embedding projection and pixel-level clustering* [43,44]. It's easy to know that this type of approach relies on the performances of post-processing and tends to suffer from under-segment or over-segment problems [27]. The above drawbacks will be further discussed in Sec 3.1 in detail.

To obtain better performances, multi-stage instance segmentation methods are designed to refine instance segmentation results stage by stage. For instance, Chen et al. use a hybrid task cascade architecture (as illustrated in Fig. 11) to interweave mask and box information [45] for achieving better performance. Besides, based on the framework of Transformers [170], [171–173,176] introduce the way of enabling end-to-end instance segmentation that can reduce non-maximum suppression (NMS) during inference. More details about the multi-stage methods will be described in Sec 3.2.

However, both two-stage and multi-stage methods rarely consider computation cost. Taking Mask R-CNN [21] as an example, it only achieves 8.6 *fps* with ResNet-101-FPN when processing 800 $\times$ 1333 image on a Titan Xp. As for single-stage object detection methods, on the contrary, they achieve a better trade-off between speed and accuracy [46,47], so single-stage instance segmentation is getting more and more critical attention, which aims to perform instance segmentation in a direct way to gain a better trade-off between speed and accuracy [48]. The most representative single-stage instance segmentation method YOLACT [48], motivated by YOLO [46] and SSD [47], divides instance segmentation tasks into two parallel tasks: generating prototype

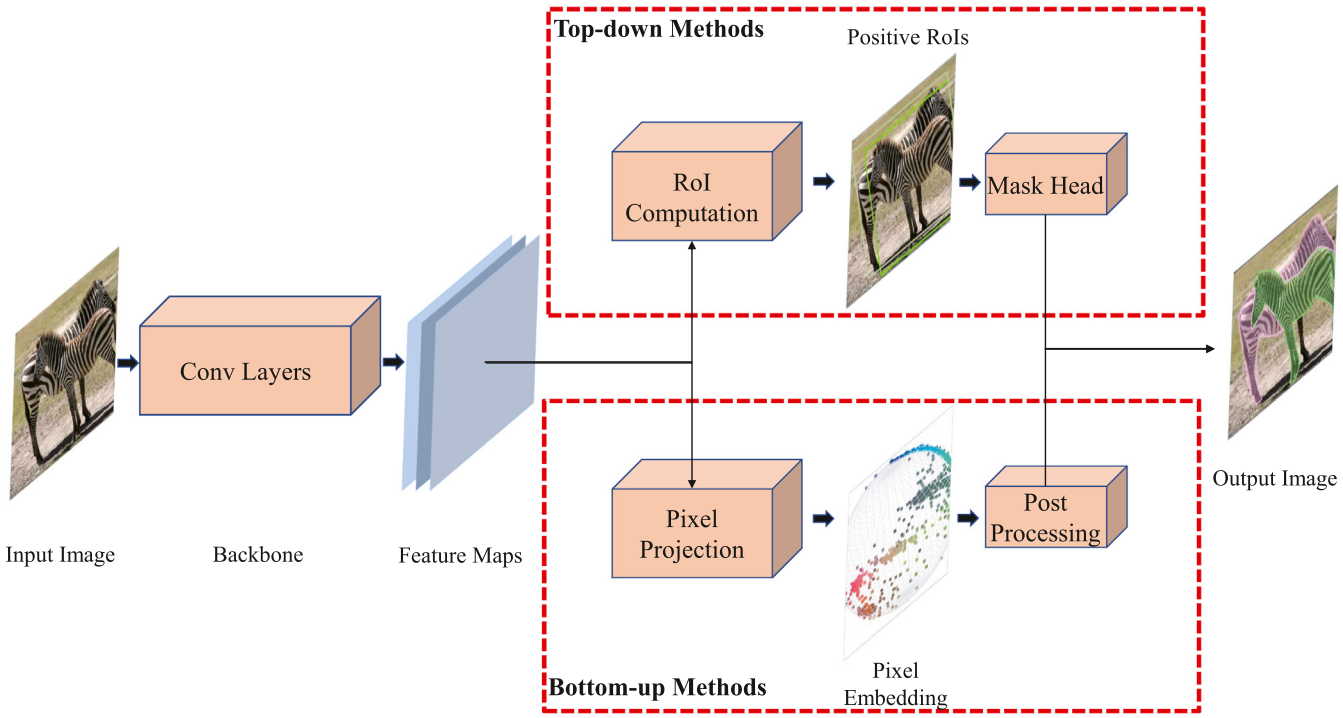


Fig. 2. The common architecture of *two-stage* instance segmentation methods. The differences between top-down methods and bottom-up methods are shown in the above red dashed boxes. Top-down methods first predict top-level bounding box information and then segment foreground object for each box. Bottom-up methods start with pixel-level projection and then distinguish each object by clustering processing.

masks and producing per-instance mask coefficients. As a result, the instance segmentation task can be performed by linearly combining the prototype masks with the mask coefficients. The main key to distinguish single-stage methods from two-stage and multi-stage methods is that location information and mask representation are of equal status for the former methods, rather than one information being dependent on the other. Fig. 3 shows the simple pipeline of single-stage instance segmentation methods.

With a large number of labeled pixel-wise mask annotations, fully-supervised learning-based instance segmentation methods have achieved decent performance. Unfortunately, annotating pixel-wise masks is far more time-consuming. In the COCO dataset, it takes over 22 worker hours to label per 1000 segmentations [50]. Therefore, there are many weakly-supervised learning-based [51] and semi-supervised learning-based [52] based instance segmentation methods that have been proposed. These methods used box-level, image-level, or point-level annotations [53–55] as supervised information to guide the learning of the model. We suggest that readers refer to Sec 3.4 for more details about methods applied weakly- and semi-supervised instance segmentation problem.

In addition to general instance segmentation, there are several specific instance segmentation tasks, including salient instance segmentation [56], amodal instance segmentation [57,58], and human instance segmentation [59]. In line with salient object detection [60–62], salient instance segmentation further segments each instance in the detected object regions. Salient instance segmentation is essential for many

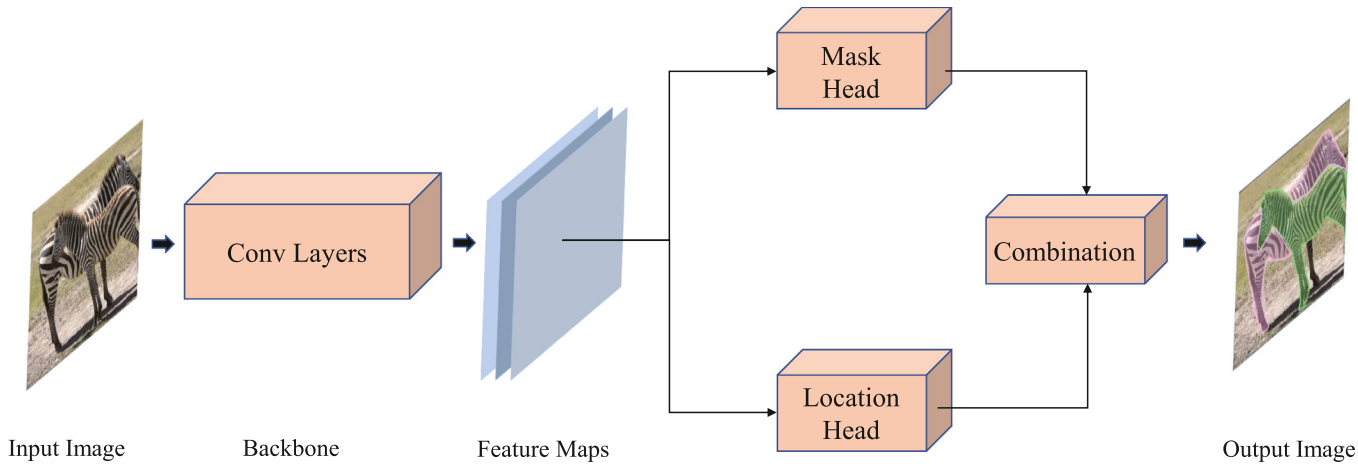


Fig. 3. The general architecture of *single-stage* instance segmentation methods. The location head locates where each object is and the mask head distinguishes what class each pixel belongs to. Different from two-stage methods, the execution of the positioning head and the mask head in the single-stage method has no sequence.

model applications: image editing [63,99] and video summarization [64]. Amodal instance segmentation is another new direction of instance segmentation, which aims to depict instance shapes even in the case of partial invisibility and occlusion. Amodal instance segmentation can analyze more invisible information and help us avoid potential dangers. Human instance segmentation, as its name implies, only needs to segment and distinguish humans in an image. The human instance segmentation task that is related to people attract a lot of attention because of the high demand for real-life application [59]. Methods of salient instance segmentation, amodal instance segmentation and human instance segmentation will be discussed briefly in Sec 3.5.

Although there are many remarkable methods on image instance segmentation, few works have been devoted to making a thorough review and comprehensive analysis of instance segmentation methods. Various image segmentation surveys focused on discussing image semantic segmentation methods and only introduced a few of instance segmentation methods [111,167,168]. Those existing image instance segmentation surveys, such as [65,169], do a great summarization and classification of some of the representative methods, present the performance on these methods, and provide potential future research directions. Unfortunately, they did not fully describe the standard benchmark datasets, evaluation metrics and backbone networks, which are most essential for instance segmentation task. In addition, they lack some of the most recent state-of-the-art methods, especially the Transformer-based methods.[1] And they also neglect the specific instance segmentation methods, and weakly- and semi-supervised instance segmentation methods. The missing information from previous reviews will be presented in our work.

Considering all the conditions, we are going to exhaustively discuss the existing instance segmentation methods, datasets, evaluation metrics and backbone networks. Therefore, the main contributions of our survey are as follows:

- We provide a detailed description of the commonly used benchmark datasets and evaluation metrics for instance segmentation tasks.
- A well-organized and comprehensive survey of the most representative instance segmentation methods (including general instance segmentation and specific instance segmentation) is reviewed.
- An exhaustive performance evaluation on different datasets is presented, which can help researchers to follow the state-of-the-art.
- We briefly discuss about the deep neural networks including the representative combination of networks (ResNet + FPN + DCN), and the attention-based architecture (Swin Transformer [177,178]).
- The application prospects and future directions of instance segmentation are discussed.

The overall structure of this paper is organized as follows: Section 1 briefly introduces the instance segmentation task and the structure of this paper. Section 2 shows the standard 11 datasets and 3 metrics, which are essential for evaluating instance segmentation methods. The following Section is concerned with the methodologies used for instance segmentation (include general instance segmentation and specific instance segmentation); in this section, most of the representative methods are reviewed. Section 4 summarizes the performances of current instance segmentation methods. Followed by the the commonly used backbone networks of instance segmentation in Section 5. Next, Section 6 gives possible applications and future research directions that might set the course of upcoming advances. At last, Section 7 concludes this paper.

2. Datasets and metrics

Datasets and metrics, which are essential for deep learning problems, can be used to help researchers to follow the state-of-the-art. Therefore, an extensive description of common instance segmentation datasets and metrics for the 2D image instance segmentation task is provided in this section.

2.1. Common challenge and datasets

The amount of available data and the quality of labeled data heavily affect the deep learning model performances [66]. In other words, current deep learning models are data-hungry [67,68]. It is worthy to know that labeling instance annotations in images is very time-consuming. Compared to bounding box annotation, instance mask annotation takes about 15 times longer. For example, finely annotating a single image in Cityscapes needs more than 1.5 h [69]. For the convenience of instance segmentation research, some institutions have made some publicly available instance segmentation datasets. As shown in Tables 1, 11 common instance segmentation datasets are enumerated. The main categories of these datasets include: street scenes, natural scenes, pathology images, humans, and common objects. And the performances of current instance segmentation methods on these datasets will be presented in Section 4.

2.1.1. Microsoft common objects in context (COCO)

The Microsoft Common Objects in COntext dataset is a large-scale dataset for image classification, object detection, semantic segmentation, and instance segmentation [50]. It contains 2.5 million labeled instances in 328 k images and 91 object classes with 80 of them used for instance segmentation. The images of COCO are captured from daily life scenes with different resolutions. Because of the large scale of this dataset, it has gradually become a widely used benchmark for instance segmentation competitions and model evaluation. Up to now, the COCO dataset has two popular versions: COCO 2014 and COCO 2017. The difference between these two versions is the division of the dataset. COCO 2014 provides more than 83 k images for training, 41 k images for validation, and 41 k images for testing. While COCO 2017 dataset has 118 k training images, 5 k validation images, and 41 k testing images.

2.1.2. Pascal visual object classes (VOC)

The Pascal Visual Object Classes (VOC) challenge dataset is a benchmark for image classification, object detection, image segmentation, and person layout taster [70]. The main goal of this challenge is to recognize objects in realistic scenes. This dataset contains 21 object classes, which can be grouped into person (person), animal (bird, cat, cow, dog, horse, and sheep), vehicle (aeroplane, bicycle, boat, bus, car, motorbike, and train), indoor (bottle, chair, dining table, potted plant, sofa, and tv/monitor) and background (except the above categories). The number of training and validation images are 1464 and 1449 respectively. The Pascal Visual Object Classes (VOC) challenge provides an open online evaluation server for instance segmentation methods evaluation with 1456 testing images, whose annotations are not publicly available.

2.1.3. Cityscapes

The Cityscapes dataset is a benchmark suite for semantic urban scene understanding: semantic-level, instance-level and panoptic-level segmentation [69]. The Cityscapes dataset contains rich urban scene images captured from 50 daytime European cities. After manual selection, images in Cityscapes have 3 characteristics: numerous objects, varying scene layouts, and varying backgrounds. There are 5000 high-quality pixel-level annotated images with 2975 training, 500 validation, and 1525 testing images. The instance segmentation task contains eight object categories, whose name are *person*, *rider*, *car*, *truck*, *bus*, *train*, *mcycle* and *bicycle*.

2.1.4. KITTI

The KITTI dataset is a meaningful benchmark for autonomous driving [1,71]. The KITTI dataset is captured from a moving platform driving in and around Karlsruhe, Germany. Although this dataset is popular for

[1] https://paperswithcode.com/task/instance-segmentation.

Table 1
The overview of common instance segmentation datasets.

Name	Scene	Classes[a]	Resolution	Description
MS COCO	General	80	Dynamic	The most common dataset for instance segmentation task. COCO 2014 contains 82,783 training images, 40,504 validation images and 40,775 testing images, and COCO 2017 has 118,287 training images and 5000 validation images and its test set is same as COCO 2014.
PASCAL VOC	General	20	Dynamic	The number of train images and validation images with segmentation is 1464 and 1449 respectively, and the number of test images is 1456.
Cityscapes	Street	30	2048×1024	This dataset has finely annotated 2975 training images, 500 validation images, and 1525 testing images for instance segmentation.
KITTI	Street	Dynamic	Dynamic	For most evaluation, http://liangchiehchen.com/projects/beat_the_MTurkers.htmlKITTI car segmentation is the popular choice and open to the public. This subset holds 3172 training images, 120 validation images and 144 testing images.
PASCAL 2012 SBD	General	20	Dynamic	It contains 5623 training images and 5732 testing images.
CVPPP	Plants	1(leaf)	530×500[b]	This dataset is developed for plant phenotyping and its common subset A1 has 128 training images and 33 testing images.
MVD	Street	60	Dynamic	It is the worldâ€™s largest and most diverse publicly available, pixel-accurately and instance-specifically annotated street-level imagery dataset, which contains 18,000 training images, 2000 validation images and 5000 testing images.
GlaS	Medical	Dynamic	Dynamic	This dataset has 85 training images and 80 testing images. There are 37 benign sections and 48 malignant ones in the training set, 37 benign sections and 43 malignant ones in the testing set.
KINS	Street	8	Dynamic	The total number of it is 14,991 and it split into two parts where 7474 images are for training and the other 7517 images are for testing.
LVIS	General	1200	Dynamic	This dataset is obtained by annotating COCO2017 in entry-level. Now LVIS dataset has 57,263 training images, 5000 validation images and 19,761 testing images. According to the statistics, the LVIS dataset has 693,958 instances in the training set and 50,763 instance in the validation set.
Open Image	General	350	Dynamic	This dataset selects 350 classes to annotate instance mask based on strict criteria. There are 944,037 training images, 13,524 validation images and 40,386 testing images.
Others	*	*	*	There are many dataset are designed for some purpose, *e.g.* road line detection, medical analysis, remote sensing technique and human segmentation.

[a] This only contains the classes of instance segmentation.
[b] The A1 subset of CVPPP.

visual odometry, 3D object detection, and 3D object tracking, it does not consist of instance segmentation annotations in the beginning. In the later, several researchers complete the mask labeling mission on the basis of their own needs. For example, KITTI car segmentation is the most representative subset [2,72], which contains 3172 training images, 120 validation images and 144 testing images.

2.1.5. Semantic boundaries dataset (SBD)

The SBD re-annotates 11,355 PASCAL VOC 2012 images, which is split into 5623 training images (some images might be part of the VOC2012 validation set) and 5732 testing images [73]. For each image, this dataset provides segmentation and boundary annotations at the category-level and instance-level. The same as the Pascal VOC dataset, there are 20 object categories and a background category in the SBD.

2.1.6. Computer vision problems in plant phenotyping (CVPPP)

The CVPPP plant leaf dataset is developed for plant detection, plant segmentation, leaf segmentation, leaf detection, leaf counting, leaf tracking, boundary estimation, classification and regression. This dataset mainly uses the leaves of Arabidopsis (*Arabidopsis thaliana*) and tobacco (*Nicotiana tabacum*) grown by biologists, which mainly consists of five subsets. In this five subsets, the most commonly used subset contains 128 training images and 33 testing images with a size of 530×500 pixels [74].

2.1.7. Mapillary vistas dataset (MVD)

MVD is a relatively new and large-scale urban street scene dataset for empowering autonomous driving at a global scale [75]. Statistically speaking, 25,000 high-resolution images (18,000 for training, 2000 for validation, and 5000 for testing) are collected in MVD dataset. With pixel-level annotations, semantic segmentation and instance segmentation are the main goals of this dataset. More specifically, in this dataset, 66 categories are annotated for semantic segmentation and 37 categories are annotated for instance annotations. Considering the diversity criteria, the MVD images are captured from Europe, North America, South America, Asia, Africa, and Oceania. In addition to the diversity of locations, the MVD images are recorded under conditions varying in weather, season, viewing perspectives, region and daytime.

2.1.8. Gland segmentation DataSet(GlaS)

This dataset is provided by Medical Image Computing and Computer Assisted Intervention (MICCAI) 2015 Gland Segmentation Challenge [76], which consists of 165 labeled colorectal cancer histological images (85 training images and 80 testing images). The original resolution of most of the images in this dataset is 775×522.

2.1.9. KITTI INStance dataset (KINS)

KINS finely re-annotates 14,991 images from the KITTI dataset to form a large-scale dataset for amodal instance segmentation [77]. The dataset is divided into two parts: 7474 training images and 7517 testing images. Amodal instance masks, semantic labels, and relative occlusion orders are included to form annotations, from which initial instance masks can be easily inferred. In particular, the pixels of KINS images may have multiple semantic labels due to occlusion. In the KINS dataset, there are two general categories: people and vehicle. Similar to the KITTI dataset, the general category people is further split into 3 classes: pedestrian, cyclist, and person-siting. And the general category vehicle is also divided into 5 classes: car, train, truck, van, and misc.

2.1.10. Large vocabulary instance segmentation (LVIS)

This dataset is an extension of COCO dataset by reannotating 164 k images in COCO 2017, which is designed for the open problem of low-shot learning [78]. LVIS has the largest entry-level object categories and high-quality mask annotations. Up to now, the LVIS dataset contains 1200 object categories and more than 2.2 million instances. And its training set has 693,958 instances in 57,263 images and the validation set consists of 50,763 instances in 5000 images.

2.1.11. Open image

This dataset consists of 9.2 M images which are annotated with image-level labels, object bounding boxes, object segmentation masks, visual relationships, and localized narratives [79,80]. There are 6

versions of this dataset so far. In Open Image dataset V6, instance segmentation annotations cover 350 classes. There are 944,037 images and 2,686,666 instance masks in the training set, 13,524 images and 24,730 instance masks in the validation set, and 40,386 images with 74,102 instance masks in the test set. Rather than manually annotating object segmentation, object instances in the training set are annotated with an interactive segmentation approach, where professional annotators iteratively check the segmentation neural network outputs. As a result, this interactive segmentation approach provides accurate masks and is 3 times faster than the traditional labeling method. And the validation set and testing set are manually annotated with a free-painting tool.

2.1.12. Others

In addition to the above-mentioned datasets, there are several other existing instance segmentation datasets that are designed for different purposes. These datasets include, but are not limited to, the OCHuman dataset(for occluded human) [59], the SpaceNet dataset (for remote sensing) [81], the tuSimple lane dataset(for road lane detection) [82], JSRT/SCR (for medical analysis) [83] and Portrait (for portrait segmentation) [84].

2.2. Evaluation metrics

Appropriate evaluation metrics are extremely important for comparing different methods. For most of the existing instance segmentation works, better accuracy is the main goal. However, it is also desirable that the models are real-time and light-weight in the real life. Therefore, a relevant discussion about accuracy, inference time, and model complexity is given in this subsection.

2.2.1. Accuracy

There are many evaluation metrics for assessing the accuracy of different instance segmentation datasets, including but not limited to AP (averaged precision), DiC (Difference in Count), SBD (Symmetric Best Dice), MWCov (mean weighted coverage loss), MUCov (mean unweighted coverage loss), ABO (Average Best Overlap), and others. In this survey, we will describe commonly used evaluation metrics of the ones mentioned above.

- Average Precision (AP^r): as defined in *Simultaneous Detection and Segmentation* (SDS), average precision is calculated using mask Intersection-over-Union (IoU). It measures the precision between predictions and ground-truth annotations in a range of IoU threshold values. For most datasets, AP^r is the standard metric that is similar to AP^b for object detection. This metric is commonly used for instance segmentation method evaluation on the COCO dataset [50]. The standard mean average precision metrics of instance segmentation are defined as follows:

$$AP(\{y\},\{y^*\}) = \frac{1}{N}\sum_c area(\mathrm{Pr}) \tag{1}$$

$$\mathrm{Pr} = \sum_{\alpha}\sum_{ij}\mathrm{Pr}\left(y_i, y_j^*\right)\cdot \mathrm{I}\left[\mathrm{IoU}\left(y_i, y_j^*\right) > \alpha\right] \tag{2}$$

where y and y^* represent the prediction and ground-truth instance masks respectively, N is the number of object classes. Pr indicates the smoothed precision recall (PR) curve, $area(\cdot)$ calculates the area under the PR curve. α denotes the IoU threshold from 0.5 to 0.95 with an interval of 0.05, and $I(\cdot)$ is the indicator function. In the same way, AP_{50} and AP_{75} show the value of average precision at thresholds of 0.5 and 0.75 respectively. Similarly, AP_S, AP_M and AP_L are designed for evaluating average precision of small instances (number of pixels is less than 32^2), medium instances (number of pixels between 32^2 and 96^2) and large instances (number of pixels greater than 96^2).

- Difference in Count (DiC): Difference in count is designed for counting the difference between the predicted number of leaves and the ground truth, and $|DiC|$ is the absolute value of the mean of DiC averaged across all images [86].

$$|\mathrm{DiC}| = \frac{1}{N}\sum_i^N \left|\mathrm{count}_i - \mathrm{count}_i^*\right| \tag{3}$$

For the *i-th* image, count_i is the number of predicted instances and count_i^* is the number of ground-truth instances, and N is the number of images.

- Symmetric Best Dice (SBD): SBD provides a measurement to estimate the average instance segmentation accuracy. For each input label, in the Best Dice (BD) function, the ground truth label yielding maximum Dice is used for averaging.

$$\mathrm{BD}\left(L^a, L^b\right) = \frac{1}{M}\sum_{i=1}^{M}\max_{1\le j\le N}\frac{2\left|L_i^a \cap L_j^b\right|}{\left|L_i^a\right| + \left|L_j^b\right|} \tag{4}$$

Where $|\cdot|$ denotes instance area (number of pixels), L_a and L_b are two different instance sets. $L_i^a (1 \le i \le M)$ and $L_j^b (1 \le j \le N)$ are subsets of leaf object segments belonging to leaf segmentations L_a and L_b, respectively. The SBD between L^{gt} (ground-truth masks) and L^{pre} (predicted masks) can be defined as follow:

$$SBD(L^{pre}, L^{gt}) = min\left(\mathrm{BD}(L^{pre}, L^{gt}), \mathrm{BD}(L^{gt}, L^{pre})\right) \tag{5}$$

- Mean Coverage Loss: mean weighted coverage loss (MWCov) and mean unweighted coverage loss (MUCov) are used for KITTI car segmentation [85]. MUCov is defined as the maximum IoU of the ground truth with a predicted mask in instance level, and MWCov additionally weighs the IoUs by the area of the ground-truth mask.

$$\mathrm{MUCov}\left(\{y_i\}, \left\{y_j^*\right\}\right) = \sum_i \frac{1}{N}\max_j \mathrm{IoU}\left(\mathrm{y_i}, \mathrm{y_j^*}\right) \tag{6}$$

$$\mathrm{MWCov}\left(\{y_i\}, \left\{y_j^*\right\}\right) = \sum_i \alpha \max_j \mathrm{IoU}\left(\mathrm{y_i}, \mathrm{y_j^*}\right) \tag{7}$$

$$\alpha = \frac{|y_i|}{\sum_i |y_i|} \tag{8}$$

where y and y^* denote the predictions and ground-truth instance masks respectively.

- Average Best Overlap (ABO): ABO first picks the best proposal for each ground-truth instance, then computes the Jaccard index (or Overlap, or IoU) between the selected proposal and the ground truth, and finally performs the mean over all instances.

$$ABO(\{y_i\},\{y_i^*\}) = \frac{1}{N}\sum_i^N f(y_i, y_i^*) \tag{9}$$

where f can be formulated as a function to calculate Jaccard index, or overlap, or IoU.

- Others: In addition to the above-mentioned metrics, there are still some useful metrics for evaluating an instance segmentation method. For instance, *AvgFP* is used to average the number of false positive instances in all images (predicted instance does not overlap with any ground-truth instance); *AvgFn* is used to average the number of false negative instances in all images (the ground-truth instance does not overlap with any predicted instance); *InsPr* denotes the quotient of the number of GT-prediction pairs divided by the number of predictions; *InsRe* is the quotient of the number of GT-prediction pairs divided by the number of ground-truth; InsF1 is the corresponding F1 score of *InsPr* and *InsRe* [2].

2.2.2. Inference time

There is training time and inference time to determine instance segmentation model efficiency. Since all of the existing instance segmentation methods is offline process, it is not important to report the training time unless it is exaggeratedly slow. Nevertheless, the inference time of model indicates the possibility of its practical application. For simplicity, *FPS*(Frames Per Second) is a commonly used measure for evaluating efficiency of instance segmentation methods. It is convenient for researchers to pick a faster inference model under the same conditions.

2.2.3. Model complexity

Model complexity is another significant evaluation factor for comparing different instance segmentation methods. It is usually measured with respect to physical memory and computation. The former is about model parameter storage and the latter is about model forward processing. Both of them need to be considered when performing instance segmentation on mobile devices and robotic platforms. Therefore, the amount of parameters and *FLOPs*(floating point operations) is often used as the metric to evaluate physical memory usage and computation complexity respectively.

3. Instance segmentation methods

Based on the number of stages involved in the process of performing instance segmentation, fully-supervised instance segmentation methods can be divided into three main categories: single-stage, two-stage, and multi-stage methods. Since Hariharan et al. propose the first instance segmentation method based on object detection results [20], two-stages instance segmentation methods have been dominating the instance segmentation task [21,42,87]. Based on the sequence of positioning objects and generating masks, two-stage instance segmentation methods can be further divided into top-down methods and bottom-up methods. The main idea of top-down methods is to segment a single instance in the predicted object bounding box. On the contrary, bottom-up methods first map pixels to high-dimensional space, and then group pixels into different instances through clustering or metric learning methods.

Despite the success of two-stage instance segmentation methods, they still inherit the problems of object detector or pixel-level projection method. Intuitively, two-stage instance segmentation methods could not work well when the object detector or pixel-level projection method has bad performance [88]. Besides, two-stage instance segmentation methods, perform instance segmentation with a step-by-step process, does not fully explore the reciprocal information between object detection and instance segmentation [89].

On the other hand, if a deep neural network is able to perform segmentation and detection in parallel, the network can automatically reciprocate information about the task of detection and segmentation. Based on this inspiration, the single-stage instance segmentation methods came into being [48,93]. Obviously, performing segmentation and detection at the same time will also greatly reduce the inference time.

To further refine segmentation results and leverage the reciprocal relationship between detection and segmentation, researchers have designed the multi-stage cascade structure to perform instance segmentation stage by stage [45,88]. And there are also many instance segmentation methods segment which object instances one by one [85,92] based on recurrent neural networks (RNN) [90,91]. Considering that RNN-based methods only segment one instance at once, the RNN-based methods are also be classified as multi-stage methods. In addition to these two type methods, the self-attention-based instance segmentation methods [171–173,176] also refine query boxes and mask predictions with the recurrent refinement strategy.

Compared with image- and box-level labels, annotating instance masks is much time- and labor-consuming. Therefore, there are some methods being proposed to perform instance segmentation with image-level or point-level annotation based on *weakly-supervised* learning and *semi-supervised* learning. More discussion about *weakly-supervised* and *semi-supervised* instance segmentation methods will be presented in Sec. 3.4.

Besides the above-mentioned general instance segmentation task, several specific instance segmentation tasks (such as salient instance segmentation, amodal instance segmentation and human instance segmentation) will be discussed briefly in Sec. 3.5.

As shown in Fig. 4, current instance segmentation task can be classified into two major problems: general instance segmentation and specific instance segmentation. In addition, we subdivide these two kinds of problems according to the different implementation forms of each method. In this section, we are going to introduce different type methods of instance segmentation.

3.1. Two-stage instance segmentation methods

The two-stage instance segmentation methods to perform instance segmentation mainly depend on the completion of the detection and segmentation. Depending on the order of detection and segmentation, two-stage instance segmentation methods can be further divided into top-down methods which are based on detection [20,21,94], and bottom-up methods which are based on segmentation [43,95,96].

In the top-down methods, as shown in the upper part of Fig. 2, top-level bounding boxes are predicted through object detection methods, and then segmentation is carried out within each bounding box. In each bounding box, the segmentation result is output as an instance mask. Apparently, the performances of the object detector have a significant impact on the results of top-down methods. Therefore, designing a state-of-the-art detector is also one of the goals of such methods.

The bottom-up methods, as shown in the bottom part of Fig. 2, split the instance segmentation task into two cascading tasks: (1) mapping each pixel as a vector embedding; (2) grouping vector embedding into different instances through clustering methods. Intuitively, bottom-up methods require high-quality pixel-level (*bottom-level*) mapping results and well-designed clustering methods.

3.1.1. Top-down instance segmentation

At the very beginning, researchers usually considered traditional image processing methods and deep learning to perform image instance segmentation task. The pioneering work of instance segmentation task, named SDS, first use MCG [97] to detect the region of interest (RoI) and then complete classification and mask refinement with the power of convolutional neural network [20]. As research on deep neural networks (DNNs) has made great progress [8–10,98], the structure of the instance segmentation method is entirely composed of deep neural networks. To facilitate readers' understanding, we simply classify top-down instance segmentation methods according to the different implementation forms of each method (as shown in Table 2).

3.1.1.1. Dense sliding window. The most straightforward way to achieve top-down instance segmentation is to predict a candidate mask on

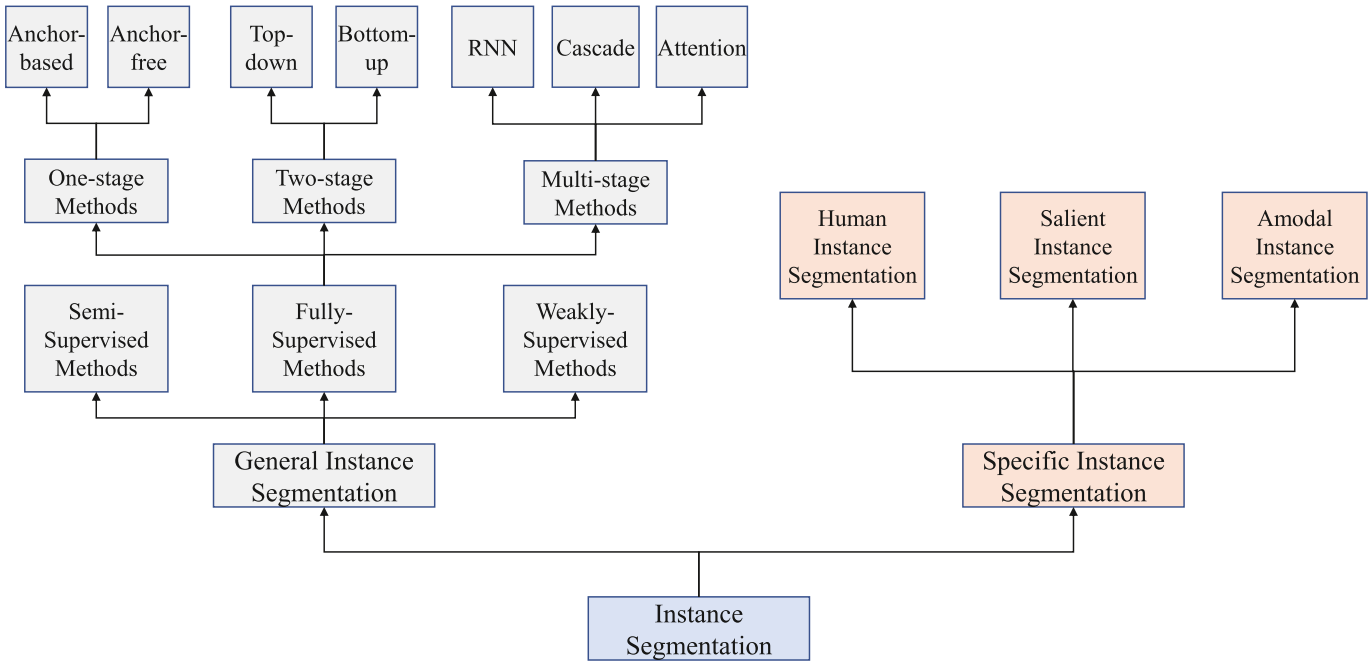


Fig. 4. The taxonomy of current instance segmentation methods.

each spatial region through a sliding window method. Obviously, sliding the window pixel by pixel is definitely suboptimal. Pinheiro et al. propose DeepMask, a dense instance segmentation method, which predicts a candidate mask on each spatial region by sliding windows on the feature maps [42]. However, results of DeepMask are predicted by using only upper-layer CNN features, which results in coarse accuracy. Going step further, Pinheiro et al. add a top-down module on DeepMask to combine feature maps of different scales and refine the quality of coarse masks [100]. The architecture of DeepMask [42] and its improved version SharpMask [100] are shown in Fig. 5.

3.1.1.2. Mulit-level features. DeepMask [42] and SDS [20] only use the output of the last fully connected (*fc*) layer as feature representations to generate instance masks. As a result, this procedure will lose precise low-level localization information. To overcome this problem, Hariharan et al. propose the Hypercolumns method to extract features from a set of layers of neural networks [102]. This architecture can use location information and semantic information of deep neural networks as much as possible. Based on the detector SSD [47] and RoIAlign [21], Zhang et al. integrate multi-level features of ResNet to predict the instance segmentation mask for each object detection box [165].

Following this idea, Liu et al. propose a simple Path Aggregation Network (PANet) [103] to integrate comprehensively low-level location information and high-level semantic information. Based on Feature Pyramid Networks (*FPN*) [104], this model designs a bottom-up context information aggregation structure, which can integrate different level features of FPN from bottom to up. Experiments show that the PANet method considering information from different levels can gain about 3 points improvement than Mask R-CNN on the COCO dataset. The framework of PANet is illustrated in Fig. 6.

Table 2
The representative top-down instance segmentation methods.

Top-down Methods	Dense Sliding Window	DeepMask [42]
		SharpMask [100]
	Multi-level Features	Hypercolumns [102]
		PANet [103]
		Mask SSD [165]
	R-CNN	Mask R-CNN [21]
		Mask Scoring R-CNN [87]
	Contour Information	MaskLab [107]
		Explicit Shape Encoding (ESE) [108]
		DeepSnake [109]

3.1.1.3. R-CNN. With the power of region-based convolutional neural networks (R-CNN), anchor-based object detection has achieved promising performances [15,94,105]. Inspired by the success of the R-CNN, top-down instance segmentation methods have also been further developed. He et al. [21] extend Faster R-CNN [94] by adding a mask branch for predicting an object mask in parallel with the existing branch for bounding box regression. It also replaces RoIPooling in Faster R-CNN [94] with RoIAlign for fixing the misalignment issue and saving more context information. The framework of Mask R-CNN is illustrated in Fig. 7. Without additional bells and whistles, Mask R-CNN [21] achieves unprecedented instance segmentation results. Thanks to its simplicity and robustness, Mask R-CNN gradually becomes the benchmark of instance segmentation task.

Huang et al. point out that the mask scores of Mask R-CNN depend on the classification results. However, classification scores only focus on semantic information and is not aware of the actual quality of the bounding -boxes and instance masks [87]. To solve this problem, MS R-CNN is proposed to directly evaluate the quality of predicted instance masks by designing a mask IoU head [87]. Experiments show that this simple modification can improve the performance of Mask R-CNN by about 1 point on the COCO benchmark.

3.1.1.4. Contour information. Although Mask R-CNN can achieve promising performances, the internal defects of Mask R-CNN are that it is insensitive to the object overlaps and the fine contour information. From another point of view, the contour information also is an alternative representation for differentiating each instance. Considering these problems, Chen et al. employ a direction prediction module (first proposed in [106]) to predict pixel directions towards the center of their corresponding instances [107]. Then the direction features are concatenated to further refine the original masks.

To perform instance segmentation efficiently, Xu et al. encode instance shapes into low-dimensional shape vectors to reduce

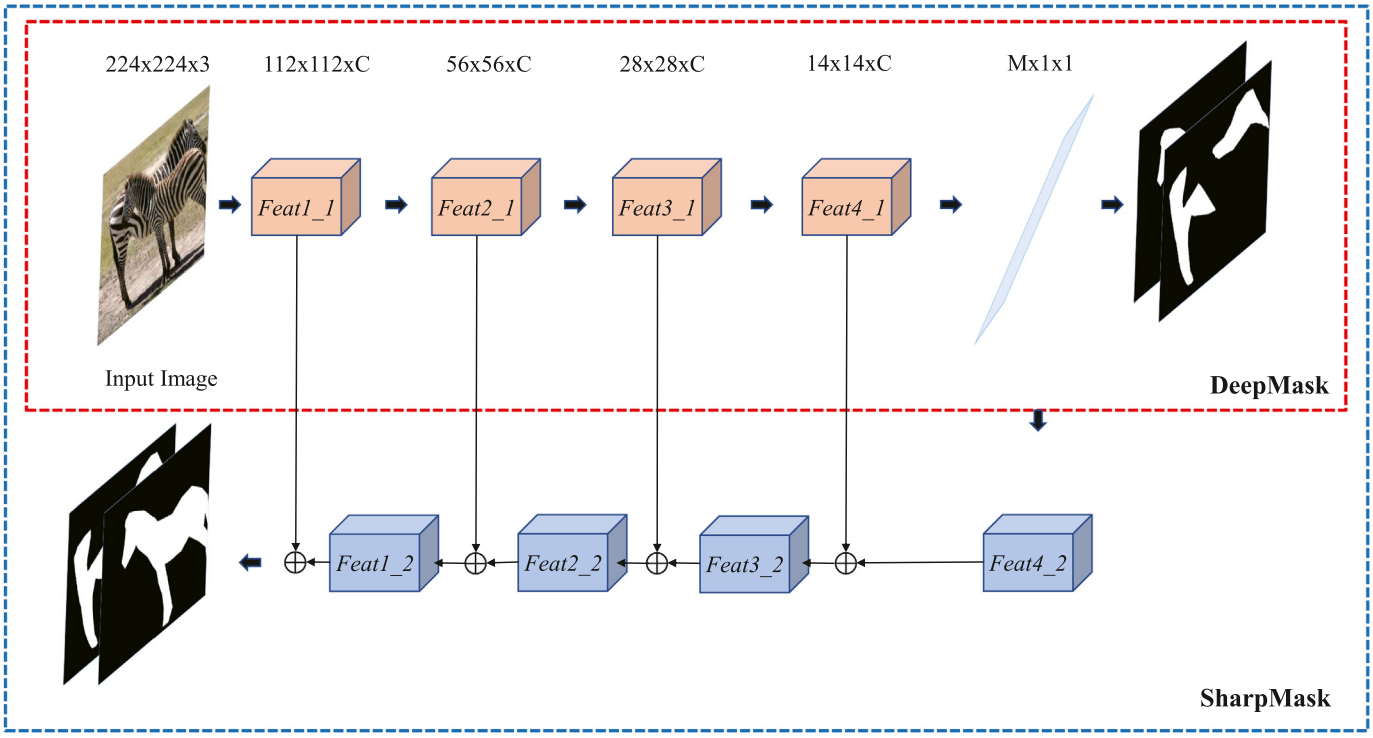


Fig. 5. The schematic diagram of dense sliding window methods. The top red dashed box is the original architecture of DeepMask, whose instance masks are generated from the upper-layer feature maps with size Mx1x1. The whole blue dashed box is the improved architecture of DeepMask, named SharpMask. The SharpMask adds a top-down module to utilize different level feature maps to enhance the mask quality.

computational consumption [108]. They sample the contour points according to the angles at the fixed interval around the inner centroid in polar coordinates. Similar to 4D bounding box vectors, the shape vector is obtained through the object detection network. Then a decoding module is implemented to map the polar coordinate to the cartesian coordinate. Next a Chebyshev polynomial is devised to fit a continuous instance contour which can further shorten the shape vector and improve the robustness of the model. With the same backbone, *i.e.* ResNet-50 [10], it can be 7 times faster than Mask R-CNN and only has a marginal performance penalty.

Motivated by the traditional snake algorithm [109], Peng et al. initialize an initial diamond contour based on detection results first, and then deform the initial diamond contour to object shape. The vertices of initial diamond contour come from the midpoints of the four sides of the detected bounding box, which are used to deform to object extreme points by deep snake. And then an octagon contour is constructed by connecting the extreme points and the midpoints in sequence. Finally, taking the fixed number points on the octagon as the initial contour, the deep snake model iteratively deforms it to the object boundary [110].

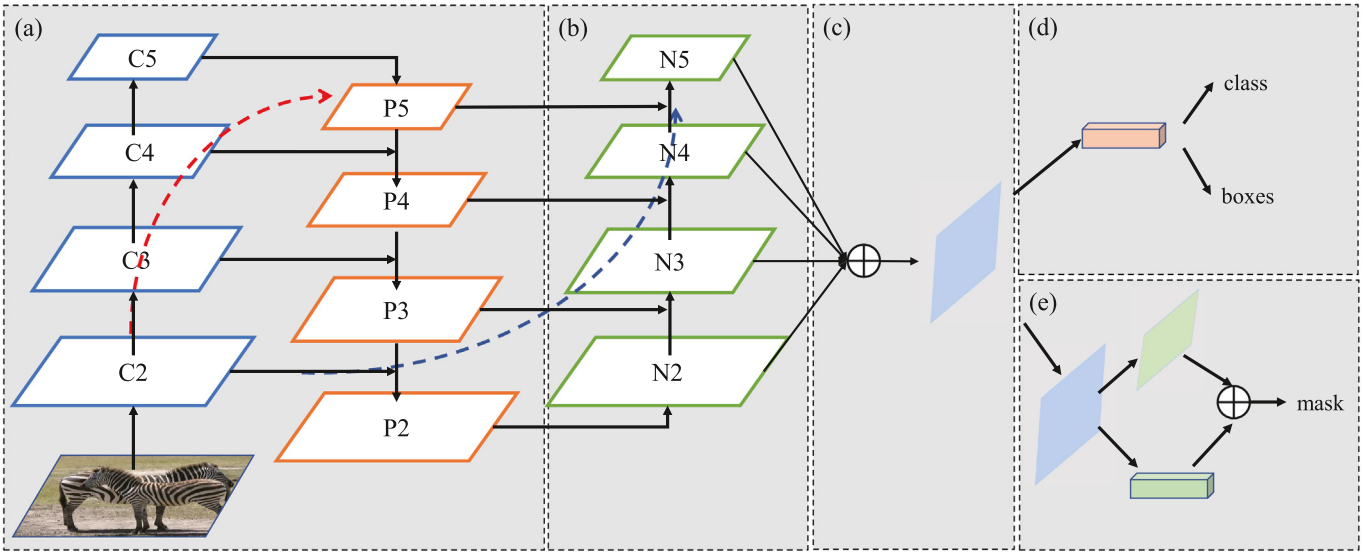


Fig. 6. The framework of PANet. (a) FPN backbone. (b) Bottom-up path augmentation. (c) Adaptive feature pooling. (d) Box branch. (e) Fully-connected fusion. Figure is reproduced from [103].

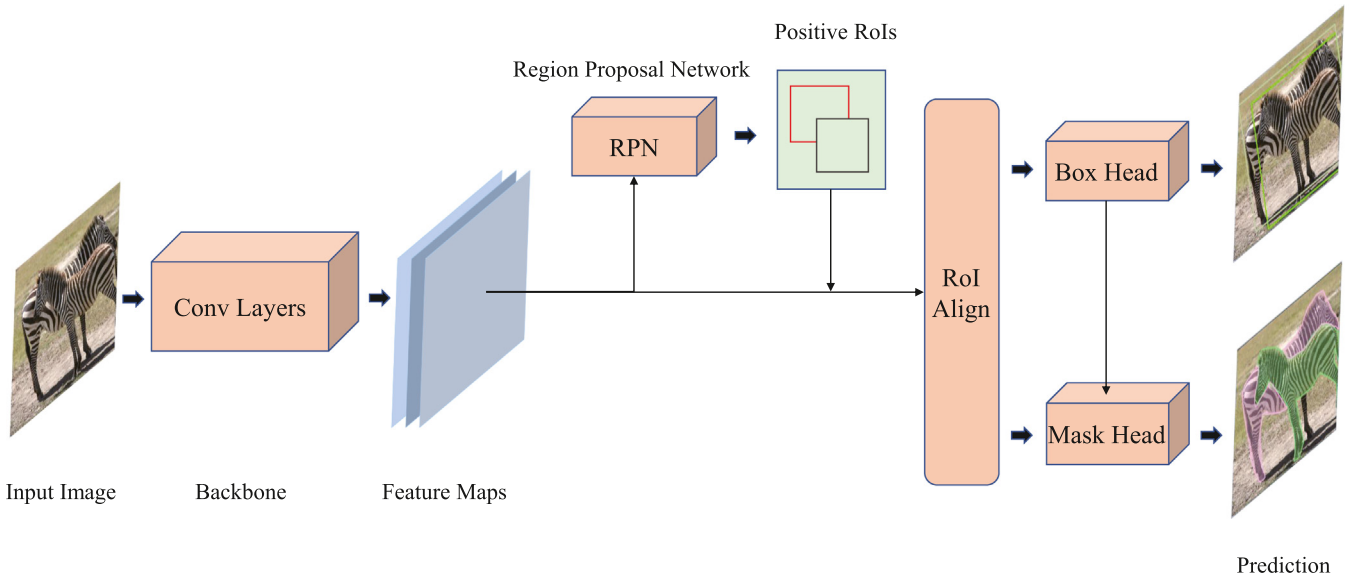


Fig. 7. The framework of the representative top-down instance segmentation method Mask R-CNN. Each instance mask is generated based on the result of object detection.

3.1.1.5. Top-down methods' advantages. Since an object detector can determine where instances are located in a given image and which category each instance belongs to, it is relatively simple and robust to perform instance segmentation by adding a mask branch to existing object detection methods. As long as the object detector performs well, top-down instance segmentation methods can enjoy a promising performance.

3.1.1.6. Top-down methods' disadvantages. However, top-down instance segmentation methods highly depend on object detection results. In other words, the upper limit of the performance of object detection determines the upper limit of the performance of instance segmentation. For instance, the predicted instance mask will be ambiguous when dealing with unconnected and occluded objects. Since humans do not need to locate the object through the bounding box before segmenting the instance, top-down methods are not in line with the human intuitive experience.

3.1.2. Bottom-up instance segmentation

Now great progress has been made in semantic segmentation [95,111,112]. And the biggest difference between instance segmentation and semantic segmentation is that instance segmentation can distinguish different instances of the same semantic category. Therefore, some researchers proposed bottom-up instance segmentation methods to distinguish each instance based on existing semantic segmentation methods [27,28]. The main idea of bottom-up instance segmentation is to perform pixel embedding projection first, and then use a post-processing method such as clustering or metric learning to map each pixel to the corresponding instance [96,113]. The main difference among different bottom-up instance segmentation methods is on how to perform pixel-level semantic projection and aggregate semantic projection into different object instances [27,28]. Similarly, Table 3 simply categorize bottom-up methods according to the implementation form of each method.

3.1.2.1. Object box cropping. The most simplest way to realize bottom-up instance segmentation is to crop semantic segmentation masks using the results of object detection and then refine the coarse masks using a learning-based algorithm. Arnab et al. use a semantic segmentation sub-network to perform semantic segmentation, and an instance segmentation sub-network to generate final instance masks. In the semantic segmentation sub-network, a higher-order Conditional Random Field (CRF) layer takes as input the pixel-wise semantic outputs of CNN and results of object detection to assign each pixel a semantic label. In the instance segmentation sub-network, coarse instance masks can be obtained by combining the outputs of detection and semantic segmentation. Finally, an instance CRF layer is used to refine the final instance segmentation masks [43]. However, this method is not end-to-end trainable and not sensitive to object occlusions and disconnection. Considering these problems, Arnab et al. further integrate semantic segmentation and instance segmentation network as an end-to-end architecture, which can enhance the flow of information and improve the performance of both of semantic segmentation and instance segmentation. Rather than distinguishing each instance with object detection information directly, a shape term branch and global term branch are added to solve disconnection and occlusion problems [114]. The overall framework of the dynamically instantiated network (DIN) [114] is shown in Fig. 8.

3.1.2.2. Contour information. To differentiate object instances based on semantic segmentation results, contour information is a clue worth considering. Bai et al. combined watershed transform with deep learning methods to directly learn the boundary energy of the watershed transform. In their method, each basin is assumed to correspond to an object instance [115].

Table 3
The representative bottom-up instance segmentation methods.

Bottom-up Methods	Object Box Cropping	DeepCRFs [43]
		Dynamically Instantiated Network (DIN) [114]
	Contour Information	Deep Watershed Transform (DWT) [115]
	Pixel Center Prediction	Box2pix [117]
		Spatial Embeddings (SE) [116]
	Depth Information	Pixel-level Encoding and Depth Layering (PEDL) [106]
		Monocular [118]
	Clustering	Affinity Derivation and Graph Merge (ADGM) [119]
		SSAP [120]
		Recurrent Pixel Embedding (RPE) [44]
		Deep Metric Learning (DML) [27]
		Discriminative Loss Function (DLF) [28]
		SGN [121]
		Proposal-free Network (PFN) [113]

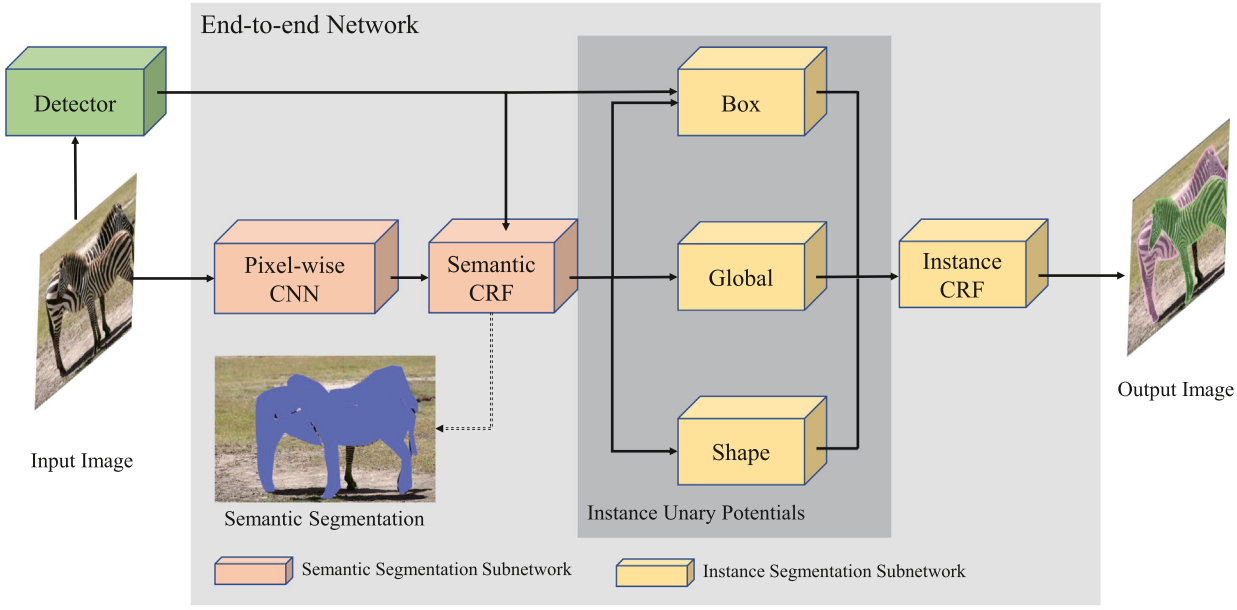


Fig. 8. The framework of DIN [114]. The intermediate category-level segmentation and instance-level detection are used to reason about instances. Figure is reproduced from [114].

3.1.2.3. Pixel center categorization. If each instance can elect a center to represent itself, then only instance center categorization of pixels is needed to complete the instance segmentation task. In accordance with this idea, Uhrig et al. integrate the SSD [47] object detector and a simple FCN as a whole framework. The FCN module is designed to output semantic labels and pixel center-offset vectors. Taking these outputs as input, a simple pixel-to-box assignment module is adopted to assigns each pixel to their corresponding box [117].

In addition to predicting pixel center-offset vectors, Neven et al. further introduce a learnable margin, which can force pixels to lay within a specified margin (the hinge margin) around the instance centroid [116]. In parallel, a separate branch is used to predict the probability of each pixel as the central seed point of instances. Finally, the instance segmentation is performed with a clustering algorithm.

3.1.2.4. Depth information. The depth information is another useful tool for differentiating object instances in monocular images. By assigning a depth ordering to instances, it is possible to differentiate between different instances and to determine non-connected instances. Based on this idea, Zhang et al. formulate instance segmentation as a pixel-level labeling where each state denotes an instance and its label ID encodes the ordering. Considering the difference of scales of objects, they propose a partition-based method, which can predict instance-level segmentation and depth ordering in an image patch. And then a Markov Random Field (MRF) module is utilized for merging patches of different scales to complete the generation of instance masks [118].

Furthermore, Uhrig et al. design a network to simultaneously output semantic labels, depth ordering and direction information [106]. The depth ordering information can be used to judge whether the disconnected areas belong to the same instance; the pixel center directions of different instances are oriented almost opposite to each other in the overlapping area; and the direction information is related to the center of the visible area of an object.

3.1.2.5. Clustering. Based on off-the-shelf semantic segmentation architectures, pixels can be projected into high dimensional embedding space. And then clustering or metric learning approaches can be used to distinguish each instance. Theoretically, it is reasonable that any off-the-shelf clustering method can be used to differentiate object instances from the outputs of semantic segmentation method.

Following this idea, Liu et al. use the pixel affinity information as the clustering clue to distinguish object instances. In this work, two parallel networks are designed to get semantic information and pixel affinity information from images. Based on this two information, they use a graphic model to determine whether pairs of neighboring pixels belong to the same instance or not [119]. Similar to [119], Kong et al. compute the cosine similarity between pixels on a high-dimensional hypersphere to measure the similarity between pixels. Then pixels are grouped into instances through a variant form of mean-shift clustering that is schemed as a recurrent neural network [44].

To explore more mutual benefits between semantic information and affinity information in the [119], Gao et al. design a single backbone to generate semantic segmentation information and affinity pyramid of different scales. And then a cascaded graph partition module fuses these two features to distinguish different instances in semantic segmentation [120]. The framework of [120] is described in Fig. 9.

In another way, Fathi et al. modify a pre-trained semantic segmentation model as a two branches architecture, in which an embedding prediction branch is designed to project each pixel into semantic dimension and an object grouping branch is used to distinguish object instance [27]. Specifically, the object grouping branch is used to complete two tasks: (1) predicting the probability of each pixel becoming a seed point; (2) assigning a category label to the instance mask centered at each pixel. And then method utilizes the learned distance metric and mask classification scores to sample high-quality instance masks. For better discriminating different instances, Bert et al. design a discriminative loss function, which can increase distances of pixel embeddings of different instances and decrease distances of pixel embeddings of the same instance [28]. With a simple post-processing thresholding operation, pixel embeddings are clustered into different instances.

To obtain the number of clusters for the clustering algorithm in the instance segmentation methods, Liang et al. add a branch for the prediction of the number of instances based in the instance-level segmentation [113]. More specifically, [113] considers the instance segmentation task as three subtasks: semantic segmentation, instance location prediction for each pixel, and the number of instances prediction. The target of location prediction is to assign each pixel with bounding boxes information with respect to that of the top-left corner and the bottom-right corner of the corresponding instance. It is very crucial to predict precise instance locations for segmenting the heavily occluded

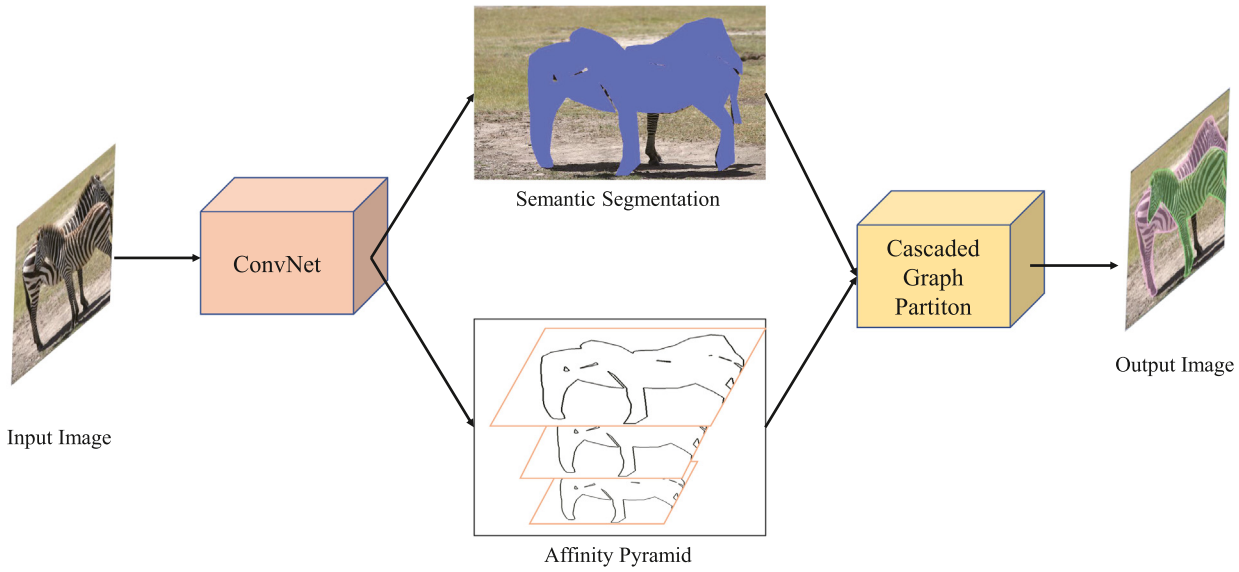


Fig. 9. The brief description diagram of [120]. Figure is reproduced from [120].

instances. And then the instance location predictions and the feature maps of the last convolutional layer from the semantic segmentation network are combined together to form the fused feature maps for predicting the number of instances. The final instance mask is obtained by clustering the instance location predictions and semantic segmentation results by using the number of instances as clusters number.

All the mentioned bottom-up clustering methods perform instance segmentation on the basis of pixel-level semantic segmentation and clustering. That treating a region as a *super-pixel* and classifying these *super-pixels* is another direction to perform instance segmentation. Liu et al. treat the instance segmentation task as a sequence of sub-grouping problems by generating and aggregating horizontal and vertical line segmentation. Firstly, the method prdicts horizontal and vertical object breakpoints to create line segments. Secondly, it groups horizontal and vertical lines into connected components. Finally, it adpots a CNN-based network to form the final set of object instance by gruoping these components [121].

3.1.2.6. Bottom-up methods' advantages. It is worth mentioning that bottom-up instance segmentation approaches conform to human intuition. When humans perform instance-level recognition, the object box information is not important. That is we can directly distinguish which instance the visual region belongs to. Therefore, it is advisable that we can achieve boxes-free or proposal-free bottom-up instance segmentation methods.

3.1.2.7. Bottom-up methods' disadvantages. However, there still exit certain limitations of the bottom-up instance segmentation methods. First, a robust semantic segmentation network backbone is needed to project each pixel into high-level dimensional space. Second, the post-processing method has poor generalization ability and cannot handle complex cases, especially occlusion and disconnected cases. Compared to top-down methods, the post-processing procedure of bottom-up methods always is more complicated.

3.2. Multi-stage methods

From the discussion of two-stage instance segmentation methods, we can see that the sequential execution of detection and segmentation tasks results in not fully exploiting their relationship. To fuse more useful information, there are many researchers turning to the multi-stage cascade structure to perform instance segmentation. Table 4 lists some typical multi-stage instance segmentation methods, which include both cascade-structure-based methods, RNN-based methods and self-attention-based methods.

3.2.1. Cascade architecture

Dai et al. divide the instance segmentation task into three cascaded sub-tasks: instance discrimination (discriminating different instances with non-semantic bounding boxes), mask generation (generating pixel-level masks), and object classification (assigning a semantic label to each instance) [122]. The multi-task cascaded structure is shown in Fig. 10. In this cascade architecture, the mask generation branch does not completely rely on the object bounding box information.

Based on the architecture of Cascade R-CNN [123], Cai et al. triple the modules of Mask R-CNN. The first stage is similar to Mask R-CNN, which takes as input the outputs of region proposal network (RPN) and the original feature maps. And the second and third stage take as the original feature maps and previous stage box branch outputs to predict the new instance box and mask [88]. This idea can improve instance segmentation performances by about 4 *mAP* than Mask R-CNN on the COCO dataset.

In the Cascade Mask R-CNN [88], the box branch and the mask branch are not directly interacted within a stage. Considering this problem, Chen et al. design a hybrid task cascade (HTC) network structure, which can transfer bounding box information and mask information of the previous stage to the next stage [45]. The HTC framework is shown in Fig. 11. In this framework, each stage mask branch M_i is

Table 4
The representative multi-stage instance segmentation methods.

Multi-stage Methods	Cascade Architecture	Multi-task Network Cascades (MNC) [122]
		Cascade Mask R-CNN [88]
		Hybrid Task Cascade (HTC) [45]
	RNN	Recurrent Instance Segmentation (RIS) [92]
		End-to-End Instance Segmentation (ETE) [85]
		Recurrent neural networks for semantic instance segmentation (RSIS) [124]
	Self-attention	ISTR [173]
		QueryInst [171]
		SOLQ [172]
		Mask2Former [176]

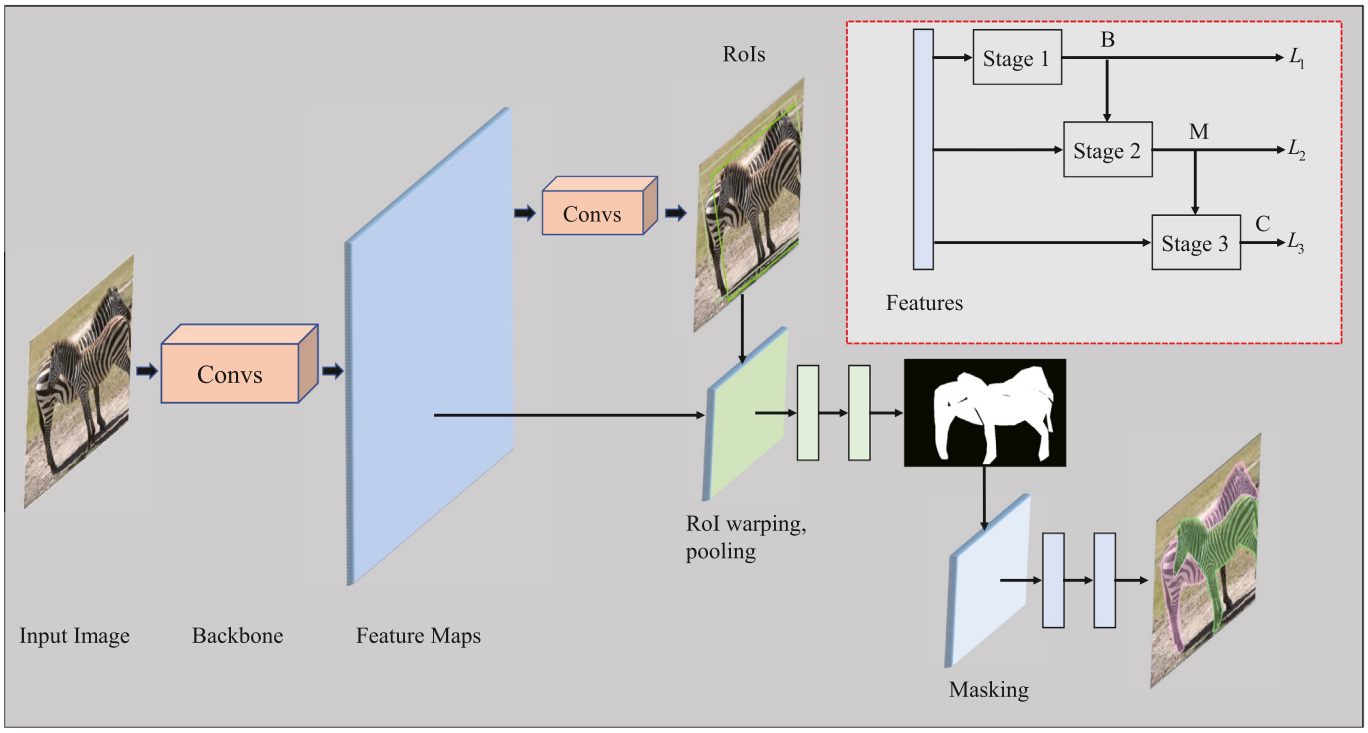


Fig. 10. The architecture of Multi-task Network Cascades (MNC). The top-right red dashed box is a simplest illustration. Figure is reproduced from [122].

correlated to the same stage box branch B_i. To further improve the accuracy of mask prediction, the second and the third mask branches also take mask outputs of the previous stage as part of the input. For better distinguishing the foreground from the complex background, the spatial contexts branch is also being sent to each mask branch. Experiments show that this multi-stage method can outperform other instance segmentation methods used for comparison.

3.2.2. RNN

Inspired by the human-like iterative and attentive counting process, using an RNN architecture to segment one instance at a time is another direction of instance segmentation. The basic problems of RNN-based instance segmentation are how to record which objects have been counted and how to locate the next object. Because the RNN-based instance segmentation methods need to repeat the segmentation module, the RNN-based instance segmentation methods are classified as multi-stage methods.

To this end, Romera-Paredes et al. use a fully convolutional network (FCN) [112] to extract instance features from images, and then directly feed it into an RNN architecture to segment one single instance at a time [92]. Unfortunately, this architecture segments all instances on a global scale which can not segment each instance very well. Therefore, Ren et al. view instance segmentation as a recurrent attentive process which can only segment one object instances at a time. An external memory module is utilized to memorize the state of the segmented objects, and a region proposal network is used to locate the next region of interest with the LSTM scheme [85].

For fully integrating low-level location information and high-level semantic information into the RNN architecture, Salvador et al. design an encoder-decoder architecture, which is similar to typical semantic segmentation methods [124]. The framework of this method is shown in Fig. 12. The encoder module adopts ResNet101 [10] pre-trained on ImageNet [68], which truncates the last convolutional layer. The decoder module takes as input the features of the encoder and the output of Convolutional LSTMs [91]. When the network detects that all instances have been segmented, a stop prediction branch is used to suspend the network. This merit allows this framework to segment a variable number of instances.

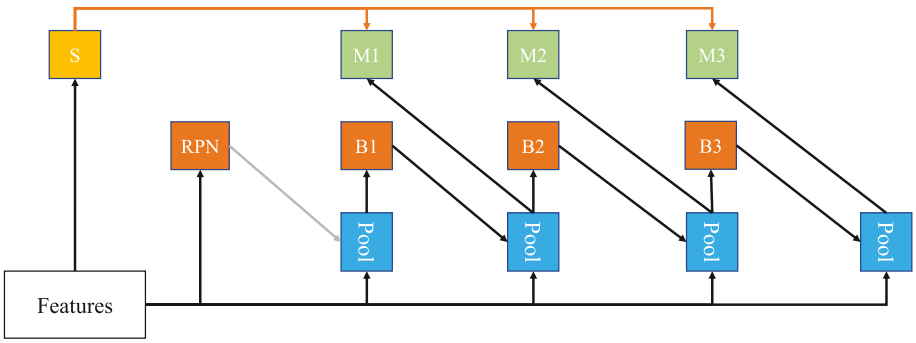


Fig. 11. The architecture of Hybrid Task Cascade (HTC), which can achieve the best performance at that time. Figure is reproduced from HTC [45].

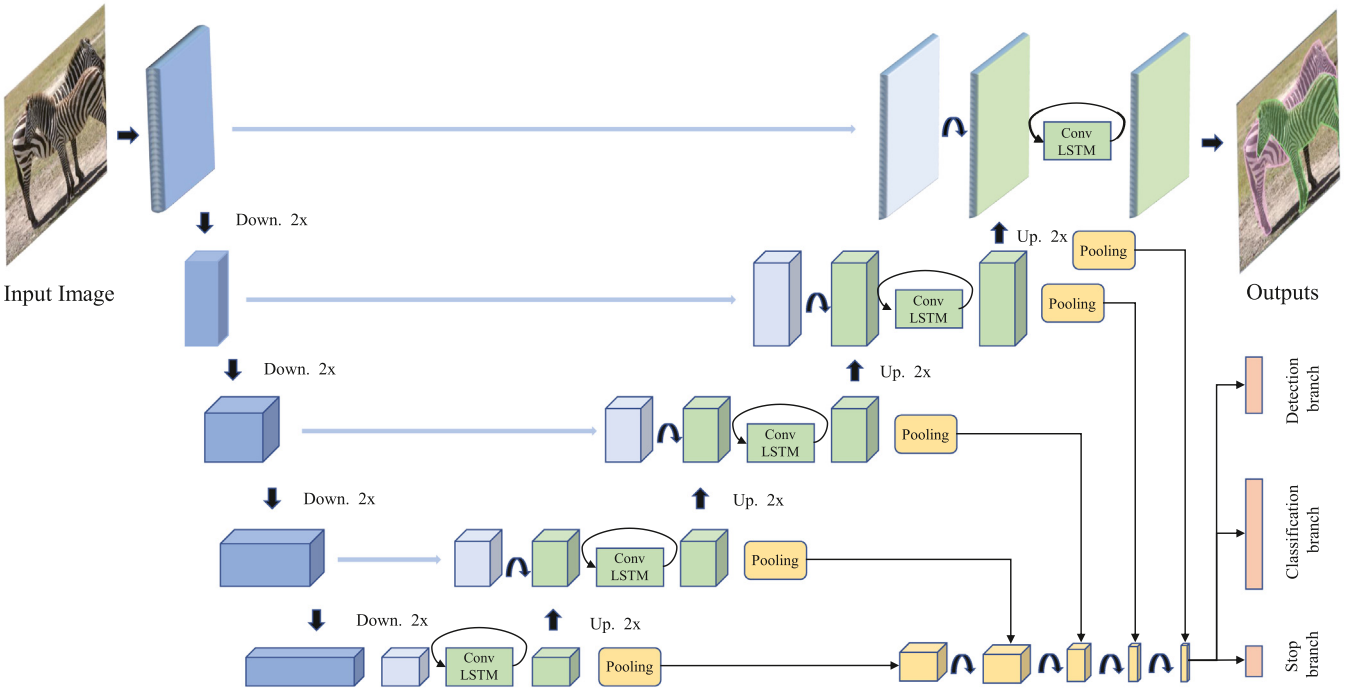


Fig. 12. The architecture of recurrent neural networks for semantic instance segmentation. Figure is reproduced from [124].

3.2.3. Self-attention

With the power of self-attention [170,180], query-based end-to-end object detection methods have been successfully developed [179,181,182]. Because of the good performance of self-attention-based object detection methods, self-attention-based instance segmentation methods have also received more attention.

The main idea of end-to-end object detection methods [179,181,182] is that they introduce learnable query embeddings and bipartite matching loss to match sparse ground truth objects. Based on the query-based object detection method Sparse R-CNN [181], Fang et al. propose a query based multi-task learning instance segmentation method [171]. As shown in the Fig. 13, QueryInst adds a mask pooling operator $\mathcal{P}^{\text{mask}}$ and a box dynamic convolution module $\text{DynConv}_t^{\text{mask}}$ on the Sparse R-CNN. Similar to the pooling operator $\mathcal{P}^{\text{bbox}}$, $\mathcal{P}^{\text{mask}}$ extracts the current stage instance mask features x_t^{mask} by using common RoI-Pooler, such as RoIAlign. Then the $\text{DynConv}_t^{\text{mask}}$ block is designed to link the relationship between instance mask features and the query embedding q_{t-1}^* of current stage. Specially, the $\text{DynConv}_t^{\text{mask}}$ block output mask prediction by enhancing the instance mask features x_t^{mask} with two consecutive convolutional layers, whose kernel parameters are produced by the query embedding q_{t-1}^*. The object box prediction of the current stage is fed into the next stage as input, which is repeated many times for achieving better performance.

Compared with the object detection task, the instance segmentation task is limited by the irregularity of the instance masks and the inadequacy of the samples to learn the mask head. Considering these problems, Hu et al. use a dimensionality reduction algorithm, such as principal component analysis (PCA) [174], to transform the mask representations to a fixed- and low-dimension embeddings [173]. Therefore, an end-to-end instance segmentation architecture with Transformers (ISTR) is designed, which enables the training phase to efficiently and efficiently process a small number of matching samples. In the ISTR, a CNN backbone and FPN is used to extract the features, which are cropped and aligned by learnable query boxes with RoIAlign [21] to get the RoI features. For better leveraging the global image information in the Transformer-blocks, ISTR extract image-level features by

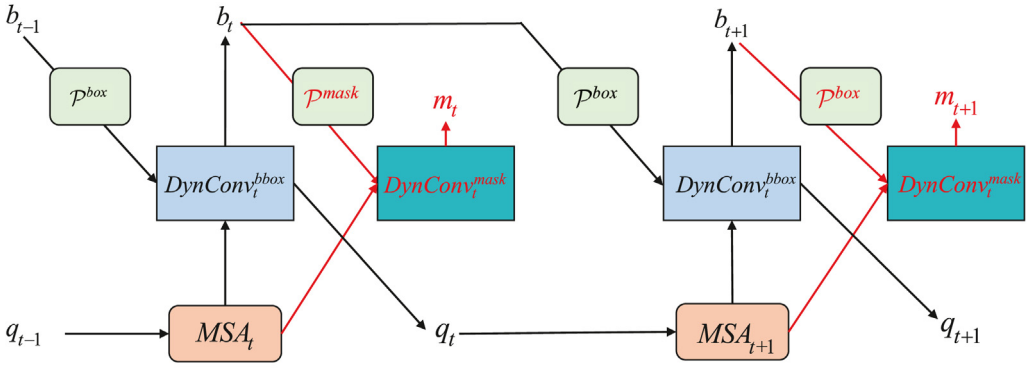


Fig. 13. The architecture of query-based instance segmentation: QueryInst. The black parts indicate the architecture of Sparse R-CNN [181], and the red parts indicate the mask blocks. Figure is reproduced from QueryInst [171].

averaging (global average pooling) and summing the features from P2 to P5. Then the sum of image-level features and learnable position embeddings multiply with three learnable weight matrices to obtain the inputs of Transformer blocks. The output of the Transformer blocks is sent to a fully connected layer to dynamically generate the filter parameters for processing object query features. Finally, three task specific heads (a class head, a box head, a mask head) and a fixed mask decoder work together to complete the problem of classification, localization, and segmentation. Similar to DETR, ISTR also adopts bipartite matching cost to match the set of predictions and ground-truth. While the matching costs for bounding boxes and classes are the same as DETR, the matching cost for mask embeddings is defined as the cosine similarity. Because the initial object queries are randomly initialized and the image features hardly focus on local information, a recurrent refinement strategy is required to make the output of the Transformer blocks focus on instance object details.

Clearly, there are three heads in ISTR that learn classification, localization and segmentation tasks respectively. In fact, it is helpful to link the segmentation task and detection task as much as possible to improve the instance segmentation performance, which has been verified in Mask Scoring RCNN [87] and HTC [45]. Therefore, Dong et al. formulate the instance segmentation as the joint learning of unified query representation (UQR) [172], which completes classification, boxes regression and mask segmentation in the one head. Similar to ISTR, SOLQ (See Fig. 14.) also utilizes the backbone network and FPN for extracting image features. The difference between the two methods is that SOLQ adopts the latest backbone network Swin Transformer [177,178] as the feature extractor and obtains higher experimental performance. The feature map of the *Res5* block is sent to the Transformer encoder layers $\mathcal{F}(\cdot)$ for enhancing the image-level features. Then the initial object queries q^0 and the refined feature maps x^K are interacted in the Transformer decoder layers to obtain instance-aware query embeddings q^K. At the same time, compared with ISTR, SOLQ select the Discrete Cosine Transform (DCT) [175] as the mask compression coding method. Finally, Experimental results show that SOLQ with ResNet101 achieves 40.9 mask AP and 48.7 box AP on the MS COCO dataset without bells and whistles, outperforming ISTR by 1.0 mask AP and 0.6 box AP.

With the successful application of bipartite matching in DETR, researchers began to try to unify semantic segmentation, instance segmentation and panoramic segmentation into a unified, simple, and effective network structure [176,183].

Under this premise, Zhang et al. proposed to dynamically generate semantic kernel and instance kernel to complete semantic segmentation and instance segmentation respectively [183]. In K-Net [183], the sum of the number of semantic kernel and instance kernel is a fixed number *N*. Then the *N* masks can be obtained by multiplying these *N* learnable kernels with the original mask feature map. Then a kernel update head takes *N* learnable kernels, *N* masks and original mask feature maps as input to dynamically update *N* kernels. In the kernel update head, the product of *N* mask predictions from the previous stage and the feature maps is used to represent *N*"meaningful groups". And then an adaptive feature update module is proposed to improve the representation ability of kernels by using the feature maps and *N* learnable kernels. Finally, a kernel interaction is presented to inform each kernel with contextual information from other groups and update the *N* learnable kernels.

Unlike K-Net, Mask2Former utilizes *N* learnable query and Transformer structures to directly predict instance and semantic masks [176]. Similar to the adaptive feature update module in K-Net, Mask2Former devises a masked attention approach to learn attention only within the predicted masked foreground regions of each query in the Transformer blocks. Experiments show that Mask2Former outperforms the state-of-the-art HTC++ by 0.3 mask AP with Swin-L backbone.

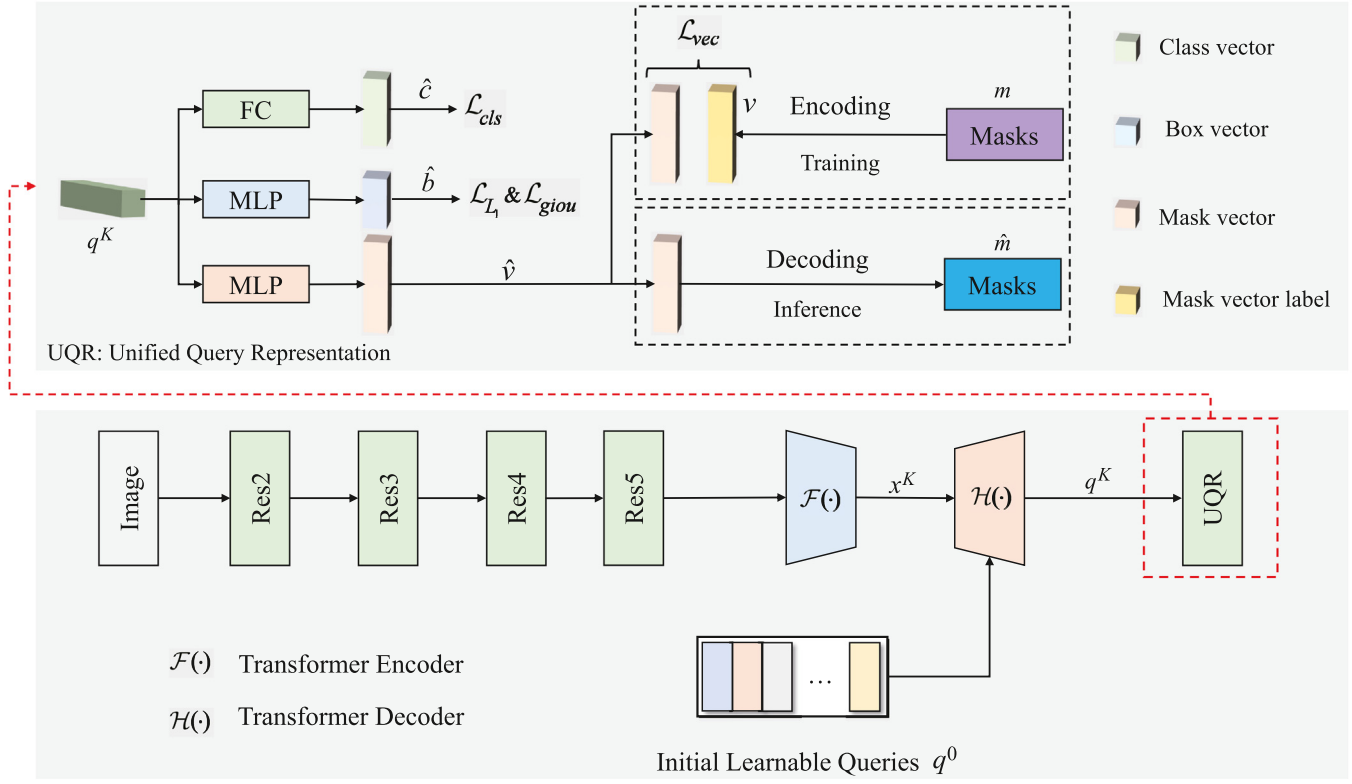


Fig. 14. The architecture of the directly-predict style instance segmentation SOLQ [172]. q^0 and q^K are the learnable object queries in the different stage. $\hat{c}, \hat{b}, \hat{v}, \hat{m}$ are the predicted class, box, mask vectors and binary masks reconstructed, respectively. $\mathcal{L}_{cls}$, $\mathcal{L}_{L_1}$ &$\mathcal{L}_{giou}$ and $\mathcal{L}_{vec}$ are the losses for classification, box regression and mask segmentation, respectively.

3.2.4. Multi-stage methods' advantage

Compared with two-stage methods, the cascade-structure-based multi-stage instance segmentation methods can enjoy better performances with a well-designed network and refine the quality of instance masks. That RNN-based instance segmentation methods exploits context of an image can segment each instance without overlapping shadows as much as possible. In addition, thanks to the development of Transformer, the performance of multi-stage instance segmentation has been further improved.

3.2.5. Multi-stage methods' disadvantage

However, there are no schemes to balance accuracy and computing resource consumption in the above methods. For example, using the ResNet-101 backbones HTC achieves 39.7 *mAP* on COCO at 2.4 *fps* on a single TITAN Xp GPU. This is definitely far from real-time instance segmentation.

3.3. Single-stage methods

As mentioned above, two-stage methods did not comprehensively consider the correlation between detection and segmentation. And multi-stage methods take too much time to achieve real-time instance segmentation. Can we coalesce detection and segmentation into a single network that also can achieve real-time instance segmentation?

Considering these problems, it is reasonable to perform both detection and segmentation in the single architecture. Therefore, single-stage instance segmentation methods are proposed. Since single-stage methods consider the relationship between object detection and semantic segmentation, it can achieve great performances and lessen processing time [48]. Motivated by the idea of single-stage object detection, single-stage instance segmentation can be further divided into two categories: (1) anchor-based methods and (2) anchor-free methods.

3.3.1. Anchor-based methods

The majority of state-of-the-art object detectors such as RetinaNet [14], YOLOv3 [46], and Faster R-CNN [94] rely on pre-defined anchor boxes. The main idea of anchor-based single-stage object detection methods is (1) generating candidate bounding boxes by using well-designed region proposal algorithms; (2) selecting the most optimal outputs by scoring candidate boxes and discarding redundant boxes with NMS.

Similar to anchor-based single-stage object detection, single-stage anchor-based instance segmentation methods start with using a network to generate a set of class-agnostic candidate score maps or masks on the candidate region. And a parallel semantic branch is adopted to extract instances. Table 5 lists several representative anchor-based instance segmentation.

3.3.1.1. Sliding window. On the basis of FCN [112], Dai et al. propose an instance-sensitive fully convolutional network (InstanceFCN). The simplest sliding window approach is used to generate a set of instance-sensitive score maps and corresponding object confidence maps on each position. Then an assembling module is used to generate a candidate object instance on each position from instance-sensitive score maps [125]. Based on the object confidence map and box-level IoU, the final instance masks is generated by using NMS approach [125]. However, the whole architecture is divided into two separate branches and not end-to-end. Besides, the score map of InstanceFCN [125] is insensitive to semantic information. For achieving end-to-end instance segmentation, Li et al. integrate the RPN into the whole architecture to achieve fully convolutional instance segmentation (FCIS) [93]. Rather than predicting a single score map, FCIS produces position-sensitive internal and external score maps. Internal score maps are only for segmentation, while the set of external score maps are only for classification. With an assembling module, FCIS integrates two score maps to simultaneously output instance masks and semantic category prediction [93].

Different from object boxes, which have a fixed and low-dimensional representation regardless of scale, segmentation masks can benefit from more richer and structured representations [101]. To bridge this gap and provide a foundation for exploring dense instance segmentation, Chen et al. [101] propose to use structured 4D tensors to represent features of instance masks. Experimental results show that TensorMask [101] can achieve more reasonable results than DeepMask.

Table 5
The representative anchor-based instance segmentation methods.

Anchor-based Methods	Sliding Window	InstanceFCN [125]
		FCIS [93]
		TensorMask [101]
	Linear Combination	YOLACT [48]

3.3.1.2. Linear combination. Although the above two single-stage instance segmentation methods can obtain better performances than MNC [122], their inference speed is slow. To reduce post-processing time consumption, Bolya et al. present a fully-convolutional model, which adds a mask generation branch to the existing one-stage object detection model [48,49]. Dubbed YOLACT (You Only Look At CoefficienTs). The main architecture of YOLACT is demonstrated in Fig. 15. This method splits the instance segmentation procedure into two parallel sub-architecture: (1) Protonet architecture: extracting spatial information by generating a certain number of prototype masks, (2) Head architecture: generating the mask coefficients and object locations. By linearly combining the prototype masks with the corresponding coefficients, the instance segmentation task is performed. In addition, it employs Fast NMS rather than traditional NMS to reduce post-processing time. The Fast NMS first computes a pairwise IoU matrix for the top n detections sorted descending by the classification score. In this IoU matrix, the lower triangular and diagonal parts are set to zero and then the maximum IoU values are calculated in the column-wise. Finally, it determine which instances are retained by thresholding the maximum values of each column. Compared to traditional NMS, Fast NMS only need to calculate IoU matrix once which can is 11.8 ms faster than a Cython implementation of traditional NMS [48]. That fast NMS allows already-removed detections to suppress other detections causes more candidate bounding boxes are removed, which reduces performance by 0.1 *mAP*. However, YOLACT does not directly provide location information, which is completely learned by the network itself. Although the *mAP* value of YOLACT is lower than Mask R-CNN [21] and FCIS [93], its AP_L value on larger objects are better than these two methods.

3.3.1.3. Anchor-based methods' advantage. Rather than predicting masks on each location, anchor-based single-stage methods can save more inference time by regression top n detections. Similar to the two-stage top-down methods, it is easier to realize that anchor-based single-stage methods segment each instance in their corresponding positive bounding box. Thanks to the previous anchor-based detection methods, this kind of methods can also achieve great performance.

3.3.1.4. Anchor-based methods' disadvantage. Definitely, the performance of anchor-based instance segmentation methods are highly relied on the results of detection. In line with two-stage top-down instance segmentation methods, this category of methods inherit the weakness of detection, *e.g.* bad performance on overlapping objects.

3.3.2. Anchor-free methods

Instance segmentation can locate an instance in the pixel-level. Without anchors, location information should be provided in other ways. Before discussing anchor-free single-stage instance segmentation, we first briefly introduce a representative anchor-free single-

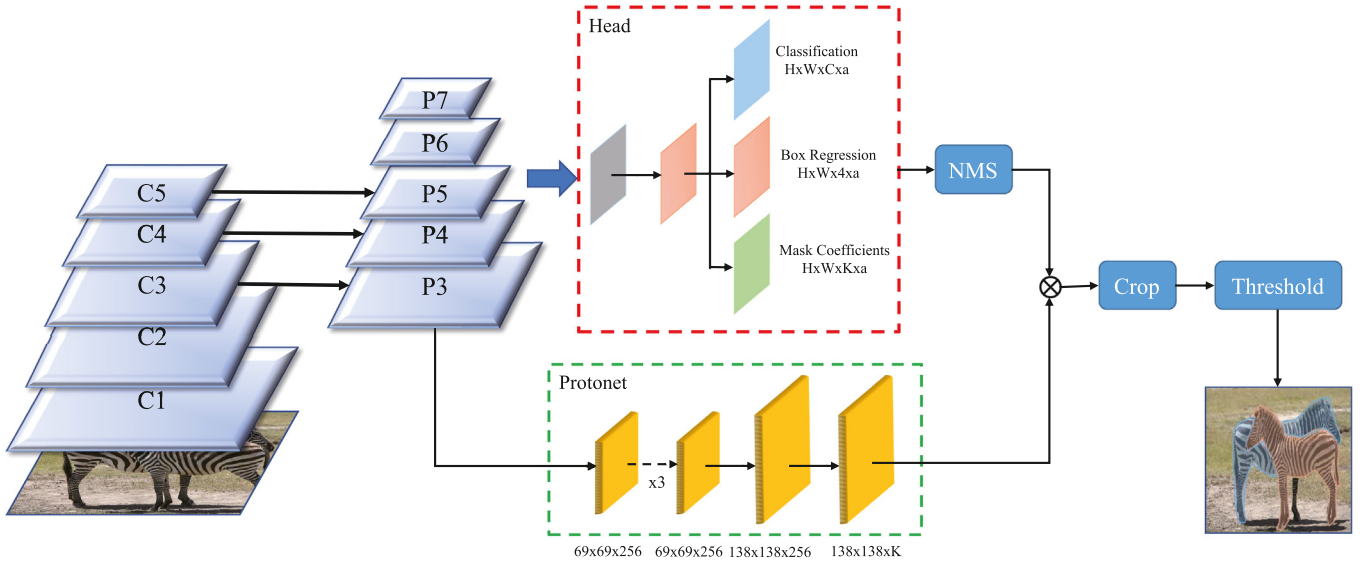


Fig. 15. The architecture of YOLACT. The head architecture and the protonet architecture are shown in red dashed box and green dashed box respectively. The labels denote feature size and channels for an image size of 550 × 550. In this figure, C is the number of object class, a denotes the number of anchors and K means the number of each instance coefficients. Figure is reproduced from [48].

stage object detection: Fully Convolutional One-Stage Object Detection (FCOS).

Different from anchor-based object detection (each location of feature maps supervise several scale bounding box prediction), FCOS only predicts a 4D vectors on each location to encode the lengths from the location to the four sides of the bounding boxes that will release the ambiguous supervision. And then a center-ness branch to indicate the possibility of this location as a relative central point, which can reduce those low-quality object proposals. The simple but robust architecture of FCOS is shown in Fig. 16. It is worth mentioning that FCOS can enjoy the FPN architecture [104] to achieve better *best possible recall* (BPR) and relieve intractable ambiguity of overlapping objects. Thanks to the simple and flexible detection framework, anchor-free instance segmentation methods are explored. And the representative anchor-free methods are listed in Table 6.

3.3.2.1. FCOS extension. Similar to two-stage bottom-up instance segmentation methods, Ying et al. propose embedding-based single-stage

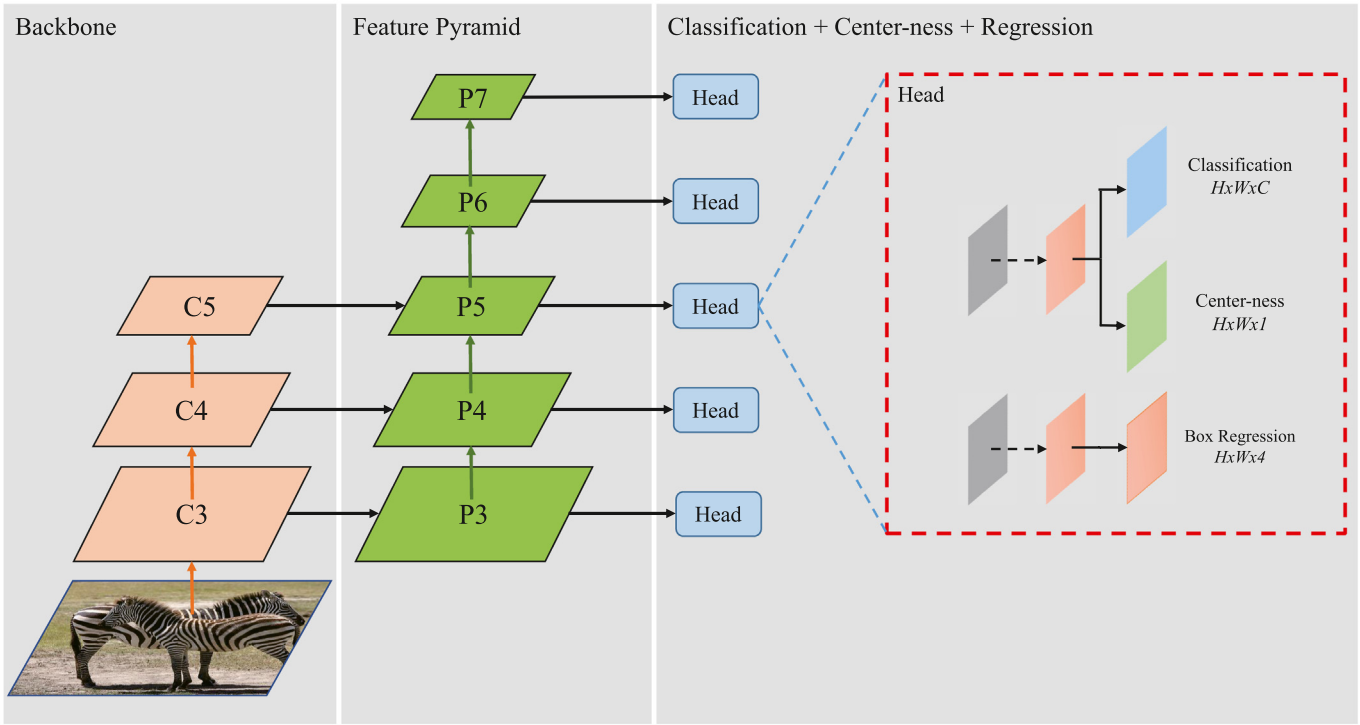


Fig. 16. The detection architecture of FCOS. The box regression branch predict a 4D vectors (l^*, t^*, r^*, b^*) to present the bounding box of each location. The center-ness sub-branch can reduce more low-quality predicted bounding boxes by locations far away from the center of an object. Figure is reproduced from [139].

Table 6
The representative anchor-free instance segmentation methods.

Anchor-free Methods	FCOS Extension	EmbedMask [140]
		BlendMask [138]
		CondInst [136]
		PolarMask [137]
		CenterMask [133]
		CenterMask [166]
	Segment Objects by Locations	SOLO [126]
		SOLOv2 [127]

instance segmentation (named *EmbedMask*), which groups semantic segmentation results into different instance. In line with YOLACT [48], EmbedMask uses the largest feature maps of FPN to generate pixel embedding. The pixel embedding represents the pixel-level context features for each location on the image, which encodes the relation between each pixel with the corresponding instance. Based on the FCOS architecture, EmbedMask predicts a proposal embedding to represent the object-level context features for the object instance. Extending the box regression branch, EmbedMask predicts a learnable proposal margin to assign each pixel to the corresponding object instance by comparing with the distance between the pixel embedding and the proposal embedding [140]. Wang et al. simultaneously predicted a rough shape-aware local mask and a rough global saliency map focusing on the overall situation. Similarly, based on the idea of YOLACT, the final instance mask is generated by assembling these two heatmaps linearly [166].

Following the idea of YOLACT [48] and Mask R-CNN [21], Chen et al. propose a single-stage dense instance segmentation method, named *BlendMask* [138]. The overall pipeline of BlendMask is demonstrated in Fig. 17. BlendMask replaces the protonet of YOLACT with the decoder of DeepLabV3+ [95] in the bottom module. Rather than predicting 1-D mask coefficients in YOLACT, BlendMask predicts a set of 3-D attention coefficients with shape of $K \times M \times M$, where $M \times M$ is the resolution of attention maps. Compared with 1-D mask coefficients, the 3-D attention coefficients can grasp more instance information, *e.g.* the coarse shape and the pose of the object. And then the outputs of towers (*predict bounding boxes and attention maps*) and bottom module (*predict prototypes*) are sent to *Blender* module. The *Blender* module first use RoIPooler to crop bases with each proposal and then resize the region to a fixed size feature map, and then interpolate the attention coefficients to the size of fixed feature maps. Finally, a simple element-wise product between attention coefficients and fixed feature maps is adopted to generate each instance masks.

For further enhancing the pixel-level information at the detected box, Lee et al. add a spatial attention-guided mask branch (modified mask branch in Mask R-CNN) to FCOS. Similar to Mask R-CNN [21], Lee et al. use the detected box to crop the feature maps and send the cropped features to an adaptive RoI assignment function. Then the spatial attention-guided mask branch forward RoI features to generate final masks [133]. Since spatial attention can focus where is an interesting region [131,132], the spatial attention-guided mask branch can segment the foreground from the background better.

However, the RoI-based methods may have the following disadvantages [136]. (1) The traditional RoIs always are axis-aligned features which may cause supernumerary computation when input image contains irregular shaped objects. (2) A stacks of convolutional layers are always needed to forward aligned RoI features, which may result that the inference time is varied in the number of instances. (3) Different RoIs, Different sizes. In order to use effective batched computation in modern deep learning frameworks [134,135], a resizing operation is often required to resize the cropped regions into patches of the same size. Considering these problems, Tian et al. propose a conditional convolutions for instance segmentation (*CondInst*). The main idea of CondInst is that for an image with K instances, K different mask heads will be dynamically generated, and each mask head will contain the characteristics of its target instance in their filters. By modifying the box head of FCOS, CondInst can predict the corresponding number of parameters which are stacked as mask FCN head. Consistent with YOLACT, CondInst forward the *P3* layer of FPN in the mask branch to provide the global mask information. To increase the relative location information of different instances, the outputs of mask branch are used to concatenate the relative coordinates information. Then for each positive location at feature maps, instance masks can be performed with the global mask information and generating mask head parameters. Without RoI features, CondInst can grasp the global image information rather than aligned RoI features. No resizing operation, no image information loss. Experimental results show that it can outperform Mask R-CNN [21] both in accuracy and speed.

In polar coordinates, FCOS can be seemed as polar diameter prediction of four direction (0°, 90°, 180°, 270°). Extended by this idea, Xie et al. view instance segmentation as an extending detection task in polar coordinates. In general, Xie et al. alter the box regression branch in FCOS as a mask regression branch by only modifying the number of polar diameters (*e.g.* 4 to 36). In the same line with FCOS, a mask classification branch and a polar centerness prediction branch are adopted in this architecture. The polar centerness prediction branch is designed to suppress these low-quality detected objects without introducing any hyperparameters. Given a set $\{d_1, d_2, \ldots, d_n\}$ for the length of n rays of one instance, the position of each corresponding contour point can be calculated by transforming from Polar Coordinate System to Cartesian Coordinate System. Then we connect the points in different positions

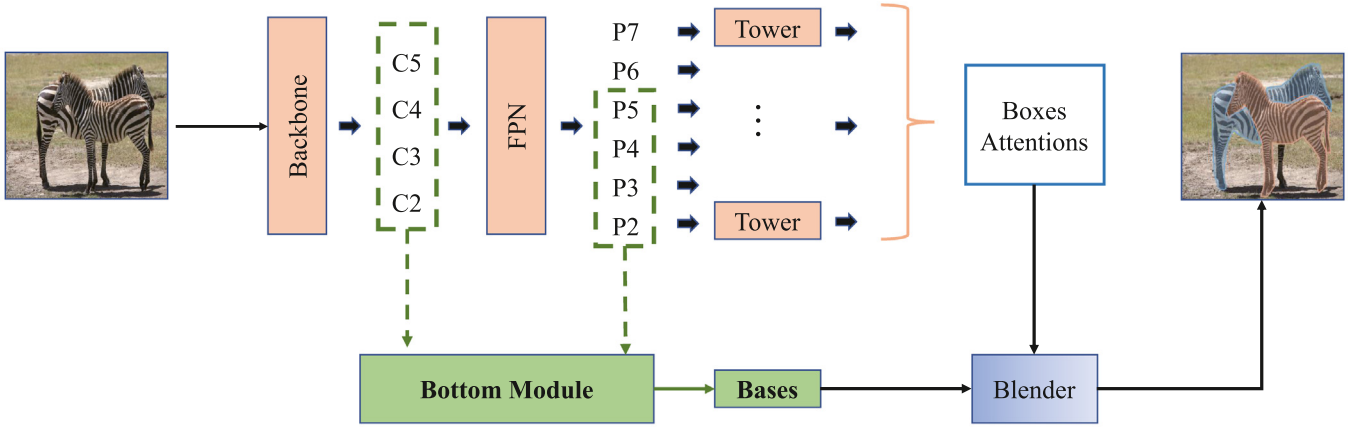


Fig. 17. The architecture of BlendMask. Figure is reproduced from [138].

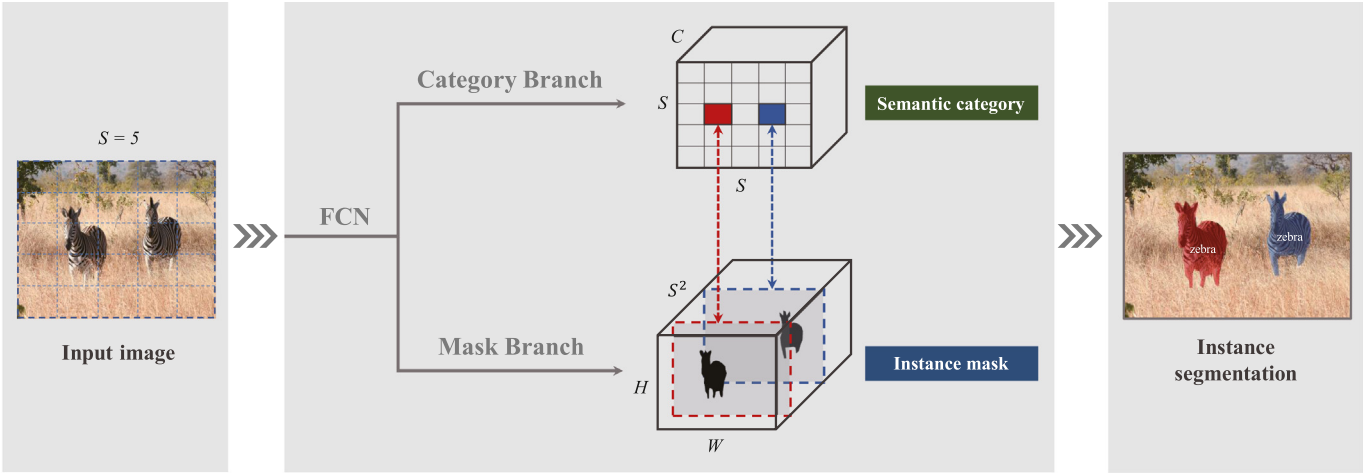


Fig. 18. The framework of SOLO. The SOLO head consists of the category branch and mask branch. Similar to YOLO [46], an image is divided into a uniform grids (*e.g.* 5×5). Each grid is responsible to predict the semantic category (top) and masks of instances (bottom) for the positive objects. Figure is extracted from [126].

in a clockwise order to get a rough instance shape. And the rough shapes can be seemed as the proposed instance masks [137]. Experiments demonstrate that such a sample transformation can also worked for instance segmentation.

3.3.2.2. Segment objects by locations. If the location and shape of the object are introduced, the corresponding instance can be distinguished. Inspired by this idea, Wang et al. present a method to segment objects by locations (SOLO). To provide location information, SOLO divides an input image into several grids, which encodes the relative position of object instances. If the instance center locate in a grid cell, that grid cell is responsible for (1) predicting the semantic category and (2) segmenting that object instance [126]. The framework of SOLO is illustrated in Fig. 18. For predicting semantic category, the category branch output a $C \times S \times S$ dimensional (C is the number of classes) tensor to denote the semantic category in each grid. Sharing features with the category branch, the mask branch is designed to predict instance masks for each grid in the global view. To distinguish instances with different sizes, SOLO employs FPN to assign objects of different sizes to different levels feature maps. Given an certain grid number, *e.g.* $S = 20$, the mask branch needs to predict $S^2 = 400$ dimensional maps. It exits somewhat redundance as the number of instances is small and the objects are located sparsely in the image. Therefore, SOLO decouples the S^2 classifier of mask branch into two groups of S classifiers: S horizontal and S vertical location categories. Thus, the output space is decreased from $H \times W \times S^2$ to $H \times W \times 2S$. And the instance masks are defined as the element-wise multiplication of two channel maps. Experiments have shown that with the spatially variant convolution module (CoordConv) [128] and deformable convolutional networks (DCN) backbone [129,130], SOLO achieves 40.4 *mAP* on the COCO dataset.

That dynamic convolutional kernels can extract more category-specific information than traditional convolutional kernels inspires Wang et al. design a mask kernel branch to predicted the parameters of mask prediction head. To enhance the relationship between mask kernel branch and mask feature branch, they use an unified mask feature branch to integrate features of different scales into a uniform size. Then the learned mask kernels are utilized to forward the integrated features to generate final instance masks [127]. Owing to that no bounding boxes generating in SOLO architecture to perform box-level NMS, SOLO should utilize mask-level NMS to reduce the proposal masks. Unfortunately, mask-level NMS is much slower than box-level NMS. To lessen the inference computation, this work also proposes the matrix NMS that considers how a predicted mask can be eliminated by assigning a decay factor to each predicted mask. For a certain m_j, the decay factor is proportional to the penalty of each prediction m_i on m_j (the mask score of m_i is larger than m_j) and inversely proportional to the probability of m_i being suppressed. Experimental results show that this dynamic, faster and stronger method can achieve better trade-off between speed and accuracy.

3.3.2.3. Summary. All the anchor-free instance segmentation methods are following a main idea: *One Location, One Mask*. For the FCOS extension methods, it is simple and straightforward to extend a mask branch on anchor-free object detection architecture. In another view, the FCOS extension methods are limited by the detection performance of FCOS. For example, the AP values of BlendMask and CondInst are lower than FCOS. SOLO is a promising pipeline to differentiate object instances by using position and scale information. With the sophisticated designs, SOLO can get better speed-accuracy trade-off than current anchor-free instance segmentation methods. However, when two objects fall in the center of the same grid, the prediction will be ambiguous, which may cause very serious problems when applying the model to real life (*e.g.* autonomous driving).

3.4. Semi and weakly supervised methods

Since annotating training data is a bottleneck for segmentation tasks [141]. Compared with instance mask annotations, box-, image- and point-level annotations are relatively easy to obtain. There are some scholars considering how to use the existing weak annotations to perform instance segmentation. Therefore, instance segmentation methods based on *semi-* and *weakly-supervised* are proposed. Table 7 shows several representative *semi- and weakly-supervised instance segmentation methods*.

Table 7
The representative semi- and weakly supervised instance segmentation methods.

Semi- and weakly supervised methods	Box-level Labels	Simple does it [53]
		Bayesian semantic instance segmentation [144]
		Learning to segment every thing [146]
		Siamese Mask R-CNN [147]
		Budget-aware semi-supervised [145]
	Image-level Labels	Peak Response Maps (PRMs) [54]
		Weakly-supervised Instance SEgmentation (WISE) [148]
	Data Augmentation	Instaboost [149]

3.4.1. Box-level labels

By using bounding box annotations, Khoreva et al. propose a detection-based *weakly-supervised* instance segmentation [53]. They modifies Grubcut [142] and MCG [97] to generate object instance masks and strengthen shape information.

Based on the Bayesian framework, Pham et al. formulate instance segmentation as boundary estimation, bounding box detection and mask evaluation. Boundary results (obtained from a specialized method, *e.g.* COB [143]) can be used to describe unknown class objects, and the boxes information and masks information are considered to segment known class objects. And then the simulated annealing approach is applied to merge a partition of the image domain and obtain approximate optimal segmentation [144]. It is worthy noting that this work [144] can be seemed as the rudiment of panoptic segmentation [22].

Since object detection and instance segmentation are linked, Hu et al. propose a novel transfer learning approach on the basis of Mask R-CNN. In this method, a learned weight transfer function is used to enable training instance segmentation models on a large set of categories all of which have box annotations, but only a small fraction of which have pixel-level annotations. For a given category c, this learned weight transfer function can transform the class-specific object detection weights into the class-specific mask weights. In the training time, this method trains the bounding box head using the standard box detection losses on all classes, but trains the mask head and the transfer function on the certain classes [146].

Based on Mask R-CNN, Michaelis et al. build a Siamese Mask R-CNN architecture that can detect and segment object instances based on a single visual example of some object category [147]. In the Siamese Mask R-CNN, the Siamese backbone applies the same backbone with shared weights to compute a similarity metric to the reference at each possible location.

To verify how different types of annotations effect on the experimental results, Bellver et al. propose to learn the internal rule of labeled data. And they also strengthen the information of unlabeled and weak-labeled data on the basis of learned rule. By modifying RSIS [124], this method demonstrates that at the same annotation cost, few strong instance labels are better than a large amount of weak labels [145].

3.4.2. Image-level labels

Compared with box annotations and instance masks, image- and point-level labels are relatively easy to obtain. With image-level annotations, Zhou et al. use local maximum values obtained from CNNs to estimate object locations and pseudo masks [54]. And then the network generates a query to select the best pseudo mask among a set of candidate object proposals. Taking a step further, Laradji et al. use point-level annotations to represent instance locations. Based on the idea of bottom-up instance segmentation, an embedding network is used to cluster different instances in an embedding space [148].

3.4.3. Data augmentation

If there are already some mask annotations, data augmentation is also a feasible way to provide more training data for instance segmentation. In the lack of annotations, Fang et al. study data augmentation techniques to tackle the issue of insufficient training data in instance segmentation [149]. By applying a random transformation to generate new images, this method can augment the training set efficiently. Experimental results demonstrate that this method can achieve 1.7 *mAP* improvement over Mask R-CNN on the COCO instance segmentation benchmark.

3.4.4. Summary

Although experiments show that current weakly- and semi-supervised methods can hardly achieve equivalent instance segmentation performance, it is worth noting that weakly- and semi-supervised instance segmentation methods greatly reduce the workload of labeling.

3.5. Specific instance segmentation methods

Under certain conditions, we only need to focus on the segmentation of instances of in some specific categories. As mentioned in Section 1, salient instance segmentation, amodal instance segmentation, and human instance segmentation are getting more and more attention because of the demand of practical applications. In this subsection, we will give a brief review of existing methods of these three specific tasks.

3.5.1. Human instance segmentation

Those *human-related* computer vision tasks are very practical, such as human pose estimation [150–152], pedestrian detection [153,154], and person re-identification [155,156]. In particular, many computer vision tasks related to human can be treated as detection and grouping jointly [96]. For instance, human pose estimation consists of key point detection and grouping them into person; human instance segmentation consists of relevant pixels detection and clustering them into different instances.

Human-related research is usually of greater interest, and likewise human instance segmentation has been well-studied. Based on the two-stage bottom-up idea, Newell et al. combine associative embedding with a stacked hourglass network [157] to perform instance segmentation and multi-person pose estimation. This stacked hourglass network outputs two heatmaps: a detection heatmap to assign each pixel a semantic label and a labeling heatmap to group pixels to instances [96]. Papandreou et al. develop a model that jointly addresses the problems of person detection, pose estimation, and human instance segmentation using a unified part-based modeling approach. By reasoning about relationships between pixels and key points in images, the semantic segmentation results can be obtained. After that, semantic segmentation results are associated with object instances *via* geometric embedding to achieve instance segmentation [158]. Tripathi et al. propose a simple cascade network to accomplish human keypoint detection and human instance segmentation. They modify VGG [9] with atrous convolution [95] to predict keypoints and map pose estimation results to segmentation [159]. To tackle the overlapping problem, Zhang et al. propose an Affine-Align module (similar to RoIAlign) to align RoIs to a uniform size, which can straighten human postures to separate overlapping person [59].

3.5.2. Amodal instance segmentation

Amodal instance segmentation is designed to predict instance masks covering both visible and invisible parts of each object [58]. Li et al. present the first amodal instance segmentation method by extending their instance segmentation method [160]. This amodal instance segmentation method iteratively expands the bounding box of each object and then refines the corresponding instance mask. Besides, a data augmentation method is provided to enlarge the amodal training set by adding object occlusion and rescaling modal boxes [58]. Zhu et al. also provide a new amodal instance segmentation dataset by annotating about 5000 images from the MS COCO dataset. They train the SharpMask [100] model by using the amodal ground truth to output the class-agnostic masks. And a semantic label is assigned to each mask through an additional ResNet-50 network [161]. Follmann et al. add an amodal mask prediction branch on Mask R-CNN [21] to produce both invisible and visible mask simultaneously. An occlusion mask prediction branch is followed to obtain the invisible masks [57].

Qi et al. add two branches to Mask R-CNN: an occlusion classification branch and a multi-level coding (MLC) branch. The occlusion classification branch is to predict existence of occlusion regions in a candidate RoI. If there exists an occluded area, the candidate RoI is considered a positive sample. The MLC branch merges all the features of the box branch, the occlusion prediction branch and the mask branch to

Table 8
Experimental results of single-stage instance segmentation methods on the test set of COCO.

Method	Backbone	AP	AP50	AP75	APS	APM	APL
CenterMask-Lite [133]	MobileNet-v2-FPN	26.7	*	*	9.0	27.0	40.9
YOLACT700 [48]	ResNet-101-FPN	31.2	50.6	32.8	12.1	33.3	47.1
PolarMask [137]	ResNeXt-101-FPN [11]	32.9	55.4	33.8	15.5	35.1	46.3
TensorMask [101]	ResNet-101-FPN	37.3	59.5	39.5	17.5	39.3	51.6
EmbedMask [140]	ResNet-101-FPN	37.7	59.1	40.3	17.9	40.4	53.0
CenterMask [166]	ResNeXt-101-DCN-FPN	38.5	61.5	41.0	18.7	40.5	54.8
CondInst [136]	ResNet-101-FPN	40.1	62.1	43.1	21.8	42.7	52.6
SOLO [126]	ResNet-101-DCN-FPN	40.4	62.7	43.3	17.6	43.3	58.9
BlendMask [138]	ResNet-101-FPN	41.3	63.1	44.6	22.7	44.1	54.5
SOLOv2 [127]]	ResNet-101-DCN-FPN	41.7	63.2	45.1	18.0	45.0	61.6
CenterMask [133]]	VoVNetV2-FPN [133]	41.8	*	*	24.4	44.4	54.3

Note: * means that no result is reported.

perform amodal instance segmentation. And the mask branch is divided into two parallel sub-branches to output amodal segmentation and modal segmentation, respectively [77].

3.5.3. Salient instance segmentation

Salient instance segmentation task aims to segment instances in detected salient bounding boxes. Li et al. design a multi-scale refinement network (MSRNet) for salient instance segmentation. This architecture can produce salient object maps and contour maps in parallel. Finally, a CRF module takes as input the refined object proposals and the saliency object map to output final salient instance segmentation masks [162]. Obviously, the performance of MSRNet heavily depends on the optimization subset and the CRF module. Fan et al. propose a real-time single-stage salient-instance segmentation (S4Net) method [56]. Similar to Mask R-CNN, S4Net consists of a box branch and a mask branch. In Mask R-CNN, RoIAlign only focus on the proposed region, which will lose the local coherence information [101]. Therefore, Barnes et al. present a new quantization-free layer for preserving local coherence, named RoIMasking. The RoIMasking layer expands the original mask obtained from each candidate box to a ternary case, which can make better use of the background information around the regions of interest.

4. Experimental evaluation

A summary of the performance of different methods can help researchers to grasp the current status of instance segmentation and select the most suitable method to complete tasks related to instance segmentation. In this section, we are going to demonstrate the performances of different instance segmentation methods with respect to different metrics mentioned in the Section 1. Considering the limited number of instance segmentation methods for the specific datasets and newer datasets, we will mainly summarize performances of instance segmentation methods on the following benchmarks: MS COCO [50], PASCAL VOC [70], Cityscapes [69], KITTI [1,71], CVPPP [74], Pascal 2012 SBD [73], KINS [77] and MVD [75].

Although most of the methods are evaluated on standard benchmarks and provide detailed descriptions to reproduce their result, some methods did not do this. Besides, the experimental settings and hardware platforms of different instance segmentation methods are not completely consistent. Therefore, the quantitative results shown in this section are mainly cited from the corresponding papers. There are also some methods that only focus on accuracy of the instance segmentation task without considering real-time and light-weight properties, hence the inference time will not be reflected in this section. Based on the discussion of Section 3, we will display the quantitative results of different instance segmentation methods on different datasets.

4.1. MS COCO

The first is the most commonly used and well investigated dataset: COCO dataset [50]. In this subsection, we will discuss the different types of instance segmentation methods performance on the basis of the number of stages.

4.1.1. Single-stage methods

In Table 8, we can see that CenterMask has the highest *AP* value of 41.8 with the VoVNetV2-FPN backbone. According to the paper reports, CenterMask-Lite with the MobileNet-v2-FPN backbone enjoys the fastest inference speed of 50 *fps* in the listed single-stage instance segmentation methods.

4.1.2. Two-stage methods

As shown in Table 9 of two-stage instance segmentation methods, the ensemble PANet+++ [103] architecture achieves the best performance with 46.7 *AP*. The ensemble architecture contains 3 ResNeXt-101, 2 SE-ResNeXt-101 [131], 1 ResNet-269 and 1 SENet. In these instance segmentation methods, the fastest method is RDSNet with a processing speed of 8.8 *fps*, which is far away from real-time inference. We can conclude that the two-stage methods can achieve better accuracy, but slower inference speed, compared with the single-stage instance segmentation methods.

4.1.3. Multi-stage methods

The Table 10 shows that the well-designed HTC++ with SwinV2-G backbone can achieve the best performance on the COCO dataset in all of the displayed methods. In contrast to the single-stage methods and two-stage methods, the multi-stage methods need longer inference time. Taking HTC [45] as an example, with the ResNet-50-FPN backbone, it only achieves 2.5 *fps* when producing 1333×800 images on a single TITAN Xp GPU.

4.2. PASCAL VOC

Next, we introduce another common segmentation benchmark: PASCAL VOC [70]. Table 11 shows the experimental results of fully-supervised and weakly-supervised methods on the PASCAL VOC 2012 validation set. The ESE method [108] can achieve a better trade-off between performance and speed. And the DIN method [114] enjoys the

Table 9
Experimental results of two-stage instance segmentation methods on the test set of COCO.

Method	Backbone	AP	AP50	AP75	APS	APM	APL
FCIS [93]	ResNet-152	29.5	49.8	*	*	*	*
Mask SSD [165]	ResNet-101	35.3	56.1	37.6	15.3	37.2	50.5
MaskLab [107]	ResNet-101	35.4	57.4	37.4	16.9	38.3	49.2
RDSNet [89]	ResNet-101	36.4	57.9	39.0	16.4	39.5	51.6
Mask R-CNN [21]	ResNeXt-101-FPN	37.1	60.0	39.4	16.9	39.9	53.5
InstaBoost [149]	ResNet-101-FPN	39.5	61.4	42.9	21.2	42.5	52.1
MS R-CNN [87]	ResNet-101-FPN-DCN	39.6	60.7	43.1	18.8	41.5	56.2
PANet [103]	ResNeXt-101-FPN	40.0	62.8	43.1	18.8	42.3	57.2
PANet+++[103]	Ensemble	**46.7**	**69.5**	**51.3**	**26.0**	**49.1**	**64.0**

Note: * means that no result is reported.

Table 10
Experimental results of multi-stage instance segmentation methods on the test set of COCO.

Method	Backbone	AP	AP50	AP75	APS	APM	APL
MNC [122]	ResNet-101	24.6	44.3	*	4.7	25.9	43.6
Cascade R-CNN [88]	ResNeXt-101-FPN	38.6	60.6	41.5	18.5	41.3	57.2
ISTR [173]	ResNet101-FPN	39.9	*	*	22.8	41.9	52.3
K-Net [183]	ResNet101-FPN	40.6	63.3	43.7	18.8	43.3	59.0
HTC [45]	ResNeXt-101-FPN	41.2	63.9	44.7	22.8	43.9	54.6
SOLQ [172]	Swin-L	46.7	*	*	29.2	50.1	60.9
QueryInst [171]	Swin-L	49.1	*	*	**31.5**	51.8	63.2
Mask2Former [176]	Swin-L	50.5	**74.9**	**54.9**	29.1	**53.8**	**71.2**
HTC++ [177]	Swin-L	51.1	*	*	*	*	*
HTC++ [178]	SwinV2-G	**54.4**	*	*	*	*	*

Note: * means that no result is reported.

Table 11
Experimental results of fully-supervised methods on the validation set of PASCAL VOC.

Method	$mAP^r_{0.5}$	$mAP^r_{0.6}$	$mAP^r_{0.7}$	$mAP^r_{0.8}$	$mAP^r_{0.9}$	AP^r_{vol}
PRM [54]	26.8	*	*	*	*	*
AssoEmbed [96]	35.1	*	26.0	*	*	*
SDS [20]	43.8	34.5	21.3	8.7	0.9	41.4
Simple does it [53]	46.4	*	*	*	*	*
Hypercolumns [102]	52.8	*	33.7	*	*	*
RSIS [124]	57.0	51.8	41.5	37.8	*	*
Deep CRFs [43]	58.3	52.4	45.4	34.9	20.1	53.1
PFN [113]	58.7	51.3	42.5	31.2	15.7	52.3
SGN [121]	61.4	**55.9**	49.9	**42.1**	**26.9**	47.2
InstanceFCN [125]	61.5	*	43.0	*	*	*
DIN [114]	61.7	55.5	48.6	39.5	25.1	**57.5**
DML [27]	62.1	53.3	41.5	*	*	*
MNC [122]	63.5	*	41.5	*	*	*
Iterative [160]	63.6	*	43.3	*	*	*
RPE [44]	64.5	*	*	*	*	*
FCIS [93]	65.7	*	**52.1**	*	*	*
ESE [108]	**69.3**	*	36.7	*	*	54.2

Note: * means that no result is reported.

highest accuracy of 57.5 AP^r_{vol}. And the weakly-supervised methods (*PRM* [54] and *Simple does it* [53]) still has a lot of room for improvement when comparing with fully-supervised methods.

4.3. Cityscapes

Table 12 lists the experimental results of instance segmentation methods on another important dataset: Cityscapes. Among these methods, WISE [148] and RSIS [124] are weakly-supervised instance segmentation methods. From Table 12, we can conclude that SSAP [120] is the top method with 31.8 *AP* and the performance of weakly-supervised methods is far lower than the performance of fully-supervised methods.

Table 13
Experimental results of different methods on the KITTI.

Method	Set	MWCov	MUCov	AvgFP	AvgFN	InsPr	InsRe	InsF1
ETE [85]	Vehicle test	**80.0**	**66.9**	**0.764**	**0.201**	*	*	*
ETE [85]	Validation	75.1	64.6	0.375	0.283	*	*	*
PEDL [106]		**80.7**	**76.3**	**0.1**	**0.1**	**91.8**	**82.3**	**86.8**
Monocular [118]	Test	70.3	55.4	0.411	0.675	59.2	59.0	59.1
Deep MRFs [2]		74.1	55.2	0.417	0.833	70.9	53.7	61.1
WISE [148]		74.2	58.9	*	*	*	*	*
PEDL [106]		79.7	**75.8**	**0.201**	**0.159**	**86.3**	74.1	79.7
Box2Pix [117]		**89.0**	*	*	*	84.6	**80.7**	**82.6**

Note: * means that no result is reported for the corresponding method.

4.4. KITTI

Then, Table 13 presents instance segmentation results on the KITTI set. It is obvious that PEDL [106] and Box2Pix [117] are the two best performing models on the KITTI validation dataset and test dataset respectively.

4.5. CVPPP

Table 14 shows the experimental results on the CVPPP dataset. The winner, ETE method [85], achieves the top accuracy of 84.9 SBD and 0.8 |*DiC*|. From the statistical data, the RNN-based approach performs instance segmentation better on the CVPPP dataset compared to the CNN-based approach.

4.6. Pascal 2012 SBD

In addition, Table 15 shows the performances of different instance segmentation methods on the SBD test set. Experimental results present that the DIN [114] method is the best one on the SBD dataset.

Table 14
Experimental results of different methods on the CVPPP test set.

Method	*SBD*	|*DiC*|
RIS [92]	56.8	1.1
RSIS [124]	74.7	1.1
DLF [28]	84.2	1.0
ETE [85]	**84.9**	**0.8**

Table 12
Experimental results of instance segmentation methods on the test set of Cityscapes.

Method	Person	Rider	Car	Truck	Bus	Train	Mcycle	Bicycle	AP50	AP
WISE [148]	*	*	*	*	*	*	*	*	*	7.8
RSIS [124]	*	*	25.8	*	*	*	*	*	17.0	7.8
PEDL [106]	*	*	22.5	*	*	*	*	*	21.1	8.9
ETE [85]	*	*	27.5	*	*	*	*	*	18.9	9.5
Box2Pix [117]	*	*	*	*	*	*	*	*	27.2	13.1
DLF [28]	*	*	*	*	*	*	*	*	35.9	17.5
DWT [115]	15.5	14.1	31.5	22.5	27.0	22.9	13.9	8.0	35.3	19.4
DIN [114]	*	*	*	*	*	*	*	*	38.8	20.0
SGN [121]	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4	44.9	25.0
Mask R-CNN [21]	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0	49.9	26.2
ADGM [119]	31.5	25.2	42.3	21.8	37.2	28.9	18.8	12.8	45.6	27.3
SE [116]	34.5	26.1	52.4	21.7	31.2	16.4	20.1	18.9	50.9	27.6
DeepSnake [110]	**37.2**	27.0	**56.0**	29.5	40.5	28.2	19.0	16.4	**58.4**	31.7
PANet [103]	36.8	**30.4**	54.8	27.0	36.3	25.5	**22.6**	**20.8**	51.8	31.8
SSAP [120]	35.4	25.5	55.9	**33.2**	**43.9**	**31.9**	19.5	16.2	51.8	**32.7**

Note: * means that no result is reported for the corresponding method.

Table 15
Experimental results of different methods on the SBD test set.

Method	mAP^r_{vol}	$mAP^r_{0.5}$	$mAP^r_{0.7}$
ESE-50 [108]	32.6	39.1	10.5
ESE-20 [108]	35.3	40.7	12.1
SDS [20]	41.4	49.7	25.3
Hypercolumns [102]	*	56.5	37.0
DeepSnake [110]	54.4	62.1	**48.3**
MNC [122]	*	**63.5**	41.5
DIN [114]	**55.4**	52.0	44.8

Note: * means that no result is reported.

Table 16
Experimental results of different methods on the KINS test set.

Method	Amodel Seg	Inmodal Seg
MNC [122]	18.5	16.1
FCIS [93]	23.5	20.8
ORCNN [57]	29.0	26.4
Mask R-CNN [21]	29.3	*
PANet [103]	30.4	**27.6**
DeepSnake [110]	**31.3**	*

Note: * means that no result is reported.

4.7. KINS

Results of different instance segmentation methods on the KINS test set in terms of the AP metric is illustrated in Table 16. All data shown in the table are from DeepSnake [110]. And the DeepSnake method is the best work on the KINS dataset currently.

4.8. MVD

Table 17 shows the result of PANet [103] on the MVD dataset. The PANet method can achieve 26.3 *AP* on the test set.

4.9. Summary

In previous subsections, we review the accuracy and speed of different instance segmentation methods on several common datasets. For some of the instance segmentation works, we fail to reproduce them because the source codes are not provided or the standard metrics are not used. Therefore, this section only presents some of the existing works.

From all the data listed above, we can see that the well-designed multi-stage instance segmentation methods achieve solid results on COCO, Cityscapes, and MVD. The success of the multi-stage instance segmentation approach is due to the fact that the framework of Mask R-CNN can be conveniently extended. And the unified framework Mask2Former also achieves brilliant instance segmentation performance, which can direct researchers to employ the relationship of semantic segmentation and instance segmentation. The two-stage instance segmentation methods account for the main part of the existing methods. However, the newly proposed single-stage methods can achieve a better precision-speed trade-off with the comparative performance as the two-stage methods. As the actual situation requires, the performance of single-stage instance segmentation methods will be further improved.

To the best of our knowledge, the backbone used in most of the off-the-shelf instance segmentation is ResNet. Take YOLACT as an example, the model file size is about 120 M when the backbone network adopts Resnet50. That is, the light-weight instance segmentation method has not been sufficiently researched. Limited by the hardware equipment in actual production, we expect to get a lightweight, real-time and good performance model. We think that in order to facilitate the actual deployment in the future, it is also very meaningful to study the light-weight instance segmentation model in the future works.

Table 17
Experimental results of different methods on the MVD test set.

Method	AP_{test}	$AP50_{test}$	AP_{val}	$AP50_{val}$
PANet [103]	26.3	45.8	24.9	44.7

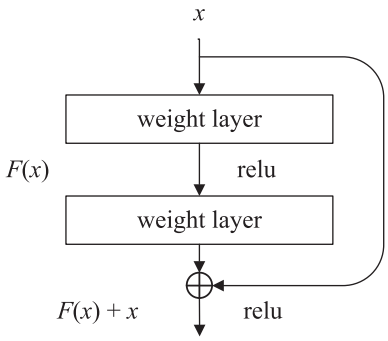


Fig. 19. A residual block [10].

5. Backbones in instance segmentation models

From Section 4, we can see that ResNet [10], FPN [104], DCN [129,130] and Swin Transformer [177] are commonly used backbones. Therefore, we are going to give a brief overview of these three architectures.

5.1. ResNet

The residual neural network is one of the most widely used networks recently. Based on a deep residual learning framework, ResNet can address the degradation problem and extract more information from the original data [10]. As a result, the depth of ResNet can reach 152 layers. The main contribution of this deep residual learning framework is using the identity mapping module, which learns to fit a residual mapping by using skipping connections or shortcuts to jump over these layers. As shown in Fig. 19, the representative ResNet block has a double-layer skipping connection, which contains nonlinearities and batch normalization [10].

5.2. FPN

It is well-known that the features of different levels extract different information in convolution neural networks. For further utilizing high-level semantic information and low-level localization information, Feature Pyramid Networks (FPN) uses a bottom-up pathway to complete the forward propagation and a top-down pathway to generate higher resolution features by upsampling spatially coarser but semantically stronger feature maps from higher pyramid levels. Then a simple lateral connection fuses the top-down features and bottom-up features by element-wise addition. The fusion process is shown in Fig. 20. Combined with ResNet, FPN is often used as a module in the instance segmentation methods [104].

5.3. DCN

The standard convolutional neural networks are composed of shape-fixed geometric convolution module, whose geometric transformation modeling ability is essentially limited. Since the instance mask is usually irregular, the standard convolution is a suboptimal choice. To apply convolutional operations to the dis-connected regions, the DCN uses two deformable modules: deformable convolution and deformable RoI

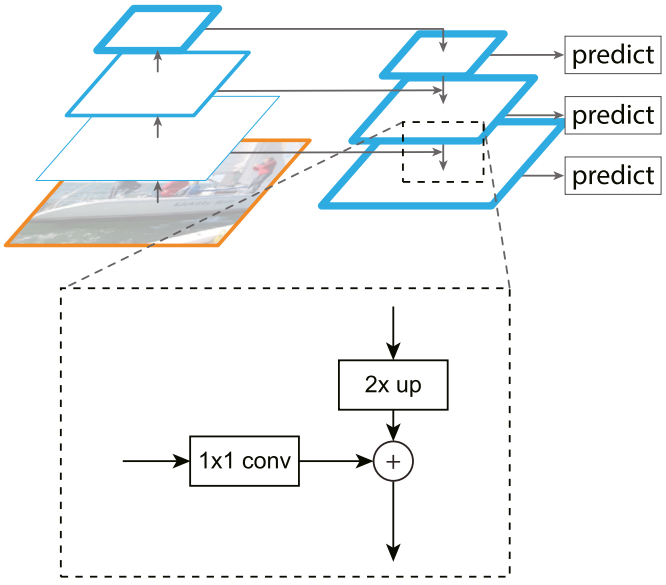


Fig. 20. The simple architecture of FPN. Figure is extracted from [104].

pooling [129]. The deformable convolution module can operate on irregular regions by adding a 2D offset to the regular grid sampling locations in the standard convolution. Similarly, the deformable RoI pooling achieves non-constant position mapping by adding an offset to each bin position in the bin partition of the previous RoI pooling module [94]. In particular, all of the offsets are learnable. As shown in Fig. 21, the deformable convolution operation can improve the ability to localize the non-grid objects.

5.4. Swin transformer

Compared with processing in language Transformers, the scale of object in computer vision tasks is usually not fixed, which makes it impossible to directly constitute the fixed-dimensional embedding in Transformer. Another difference is that the resolution of the pixels in the image is much higher than the words in the text passage. There are many vision tasks such as semantic segmentation that require dense prediction at the pixel level, which is difficult for Transformers on high-resolution images because the computational complexity of its own attention is quadratic of the image size.

With the Transformer being verified to perform well in computer vision, a unified architecture across computer vision and natural language processing has been extensively studied. Among them, the partition-based Swin Transformer is the most representative network structure [177,178]. The demonstration of the main ideas of Swin Transformer is illustrated in Fig. 22. In order to obtain an initial embedding of uniform size, the Swin Transformer first divides the input RGB image into non-overlapping fixed-size slices through a slicing module. For reducing quadratic computational cost to linear computational cost, Swin Transformer proposes to compute self-attention within a local window. These local windows are arranged to evenly segment the image in a non-overlapping manner. Swin Transformer constructs a hierarchical feature representation by merging adjacent patches in deeper transformer layers.

The Fig. 22 (b) illustrate the shift window approach between consecutive self-attention layers, which can providing more global image information to enhance modeling power. However, one problem with moving window partitions is that it creates more unequal partitions. If self-attention computation is simply performed on these unequal partitions after the padding operation, the computational cost within the same window will increase several times. To save computational cost, Liu et al. proposed a more efficient batch computation method by cyclic shift to the upper left, as shown in Fig. 22 (c). After this shift, the batch window may consist of multiple sub-windows that are not adjacent to each other in the feature map, so a masking mechanism is used to restrict the self-attentive computation to each sub-window. Using an efficient circular shift method, the number of batch windows is the same as the number of regular window partitions.

The Swin Transformer is constructed by replacing the standard Multihead Self-Attention (MSA) module in the Transformer block with a shift window-based module, leaving the other layers unchanged. A Swin Transformer block consists of a shift window-based MSA module followed by a 2-layer MLP with GELU nonlinearity between them. A LayerNorm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module.

As a result, experiments show that Swin Transformer achieves significant improvements in image classification, target detection, semantic segmentation, and instance segmentation tasks. At the same time, Swin Transformer offers more possibilities to realize a unified architecture across computer vision and natural language processing.

6. Discussion

Through the above sections, we summarized the existing 2D instance segmentation methods from a quantitative and qualitative

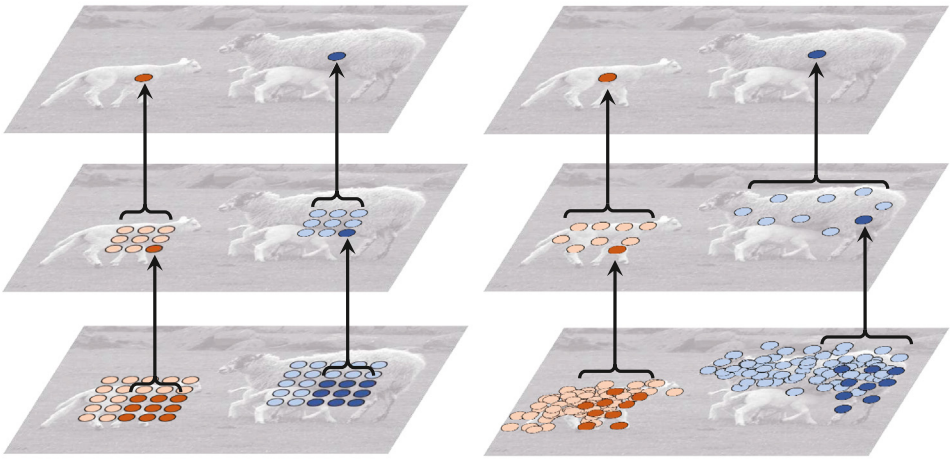
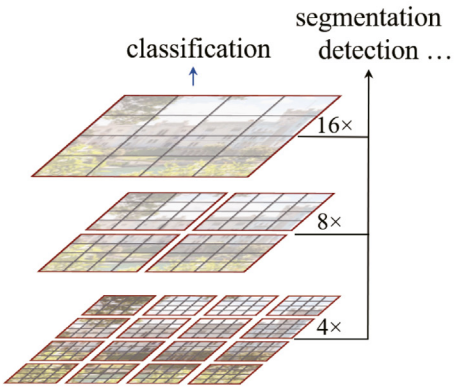
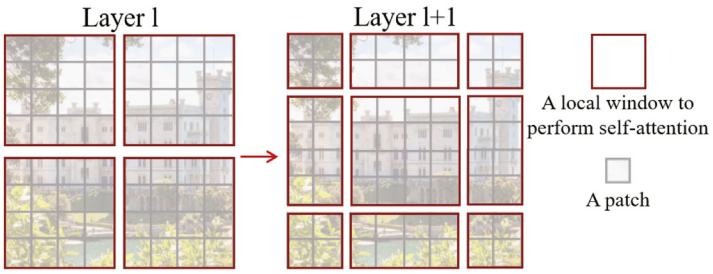


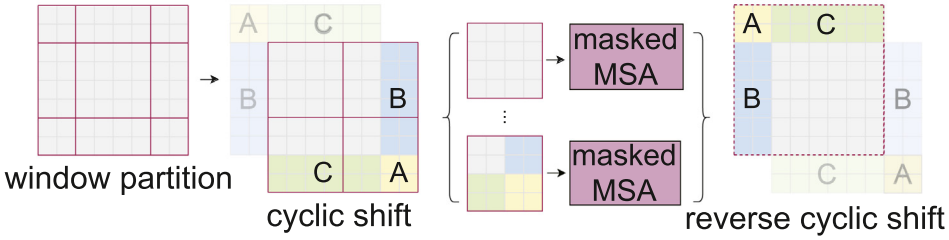
Fig. 21. Illustration of the effect of standard convolution (left) and deformable convolution (right). Figures are extracted from [129].



(a) Illustration of the patching image and hierarchical feature maps.



(b) Illustration of the *shifted window* approach.



(c) Illustration of the efficient *batch computation* approach.

Fig. 22. Demonstration of the main ideas of Swin Transformer. These figures are extracted from [177].

perspective. In this section, we will discuss the 2D instance segmentation potential future directions and applications.

6.1. Potential future direction

Instance segmentation requires pixel-level segmentation that will be used in the fine-grained scenario understanding. Therefore, higher accuracy is an essential condition for applying a model to practical applications. Taking the largest natural scene benchmark COCO as an example, the performance of the listed methods can reach up to 49.0 AP, especially the indicator of small objects is only 33.9 AP. Besides, the model storage and computation are not meeting the needs of practical applications yet. It can be said that there is still a large research untapped land in the instance segmentation task. Next, we are going to discuss some potential future directions of instance segmentation task.

6.1.1. One location, one mask

Instance segmentation can distinguish different object instances by using location information. Since anchor-free FCOS [139] can nicely perform object detection based on the idea of *One Location, Four Direction, One Bounding-box*, the idea of *One Location, One Mask* can be easily extended on instance segmentation task. We consider that the idea of *One Location, One Mask* is likely to be the next big leap. And there are two categories to provide the location information of the object: (1) relying on the object detection framework, (2) dividing the instance into a predefined grid.

We suppose that all of the existing FCOS extension instance segmentation methods can be classified into the first category. And as shown in Fig. 23, most FCOS extension methods based on this idea still have problems such as semantic ambiguity and miss-detection. And SOLO [126,127] is the representative work of the second category, whose experimental results show its solid performance. However, the performance of this method will decrease when processing occlusion problem. Therefore, there is still much room for improvement in this promising direction.

6.1.2. Multi-level feature integration

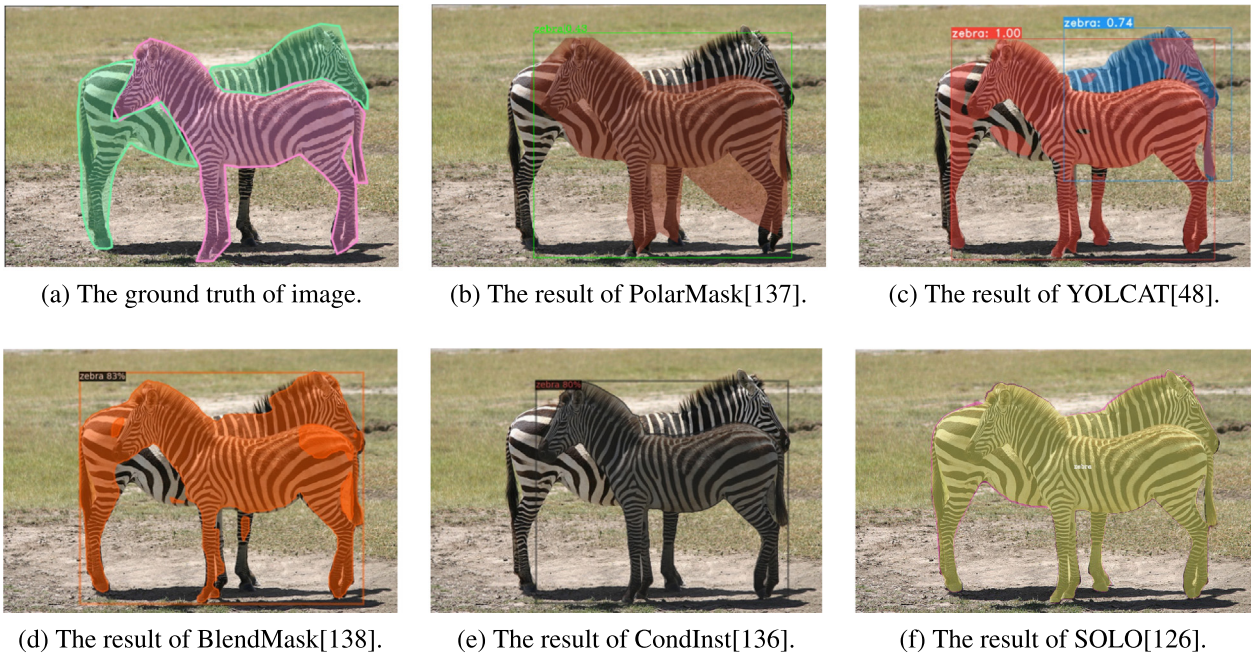
The architectures of single-stage instance segmentation methods are commonly composed of ResNet and FPN, which can extract adequate features and integrate different level features. Compared with multi-stage instance segmentation and PANet [103], single-stage instance segmentation methods are failed to further strengthen the integration of different level features. Taking YOLACT as an example, the original mask obtained from the P3 layer of FPN may exist misalignment with the bounding boxes obtained from the P7 layer of FPN. It is advisable to design the integration architecture of different level features to enhance features alignment.

6.1.3. Real time

Taking the autonomous driving scene as an example, real-time instance segmentation of the image can ensure driving safety. Based on the experimental results shown in Section 4, we can conclude that only a small part of the existing instance segmentation methods can meet real-time requirements (inference speed exceeds 30 *fps*). Therefore, we expect that in the next few years there will be more researchers working on the direction of real-time instance segmentation.

6.1.4. Memory

For most platforms, the storage memory and graphics card memory are relatively limited to store the model parameters and forward images. To the best of our knowledge, most instance segmentation methods adopt ResNet and FPN as the feature extractor. Unfortunately, the number of parameters in these two frameworks is very large. In addition to the limited storage space, the instance segmentation network



(a) The ground truth of image. (b) The result of PolarMask[137]. (c) The result of YOLCAT[48].

(d) The result of BlendMask[138]. (e) The result of CondInst[136]. (f) The result of SOLO[126].

Fig. 23. Demonstration of the results of different anchor-free instance segmentation methods. We can see that PolarMask loses too much boundary information comparing the other methods. That BlendMask and YOLACT are box-based instance segmentation methods causing semantic ambiguity in some regions. SOLO and CondInst can perform instance segmentation in the global perspective that may release the semantic ambiguity.

requires a large amount of graphics card memory during calculation. Thus it is advisable to focus the usage of storage memory and graphic memory.

6.1.5. Occlusion and disconnection

As denoted in Fig. 23, we can deduce that it is hard for current instance segmentation methods to resolve occlusion and disconnection issues. Regardless of whether the instance segmentation method is box-based or box-free, the main reason for its poor performance on occluded objects is that the current method of instance segmentation is limited by how to let the network recognize the fragmentary object and how to judge whether the disconnected areas belong to the same object. Some methods solve the disconnection problem by detecting whether there are multiple inner boxes in the outer object box, but this method relies heavily on the detection result. In general, there is still a lot of research frontier left in disconnection and occlusion scenes.

6.1.6. Small object

Similar to small object detection [164], there is still a significant gap in the performance between the segmentation of small and large objects. The existing methods always use FPN to extract the features of small objects as much as possible. As shown in Fig. 24, current anchor-free instance segmentation methods can not handle small object segmentation well. Thus, we suggest that future research can focus on the segmentation of small objects and the improvement of ground-truth masks.

6.1.7. Unified segmentation framework

K-Net [183] and Mask2Former [176] show us the possibility and effectiveness of a unified framework to implement three segmentation tasks. By designing a unified segmentation structure, it allows us to understand essentially the three segmentation tasks as region-splitting tasks. However, the relationship between semantic partitioning and instance partitioning has not been fully exploited in the unified framework. Therefore, a more concise and convenient unified structure is waiting to be further investigated.

6.1.8. Fine annotations

It is well known that supervised deep neural network models rely heavily on ground-truth data. However, the ground-truth masks of the COCO dataset are very rough. As shown in Fig. 23, the ground-truth mask of the back zebra do not contain all the object. And in the Fig. 24, there are many pigeons that are not annotated. We also wish that the quality of annotations in the commonly used datasets can be further improved.

6.1.9. Weakly- and semi-supervised

As mentioned in Sec 3.4, fully-supervised instance segmentation methods need more detailed pixel-level annotation which costs too much time and simple labor. Since weakly- and semi-supervised methods can reduce the time spent on instance segmentation annotations, we recommend that relevant researchers can further improve the performance of the weakly- and semi-supervised instance segmentation method.

6.2. Applications

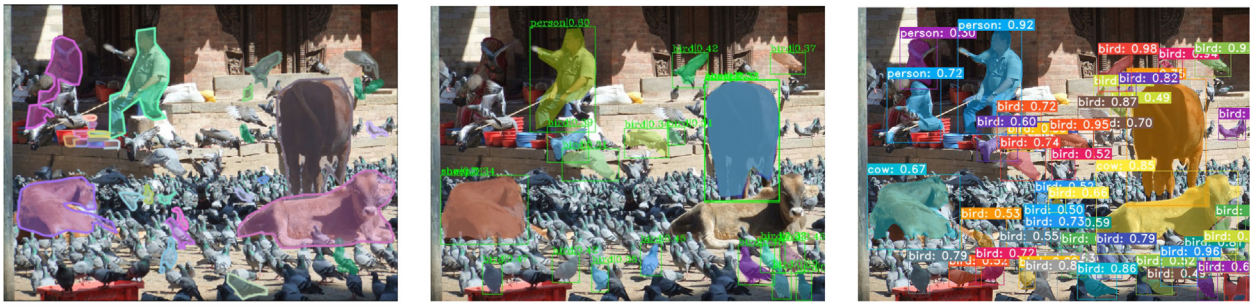
Because of the ability to segment each instance in a given image, instance segmentation can be applied to different applications. Now, we are going to simply talk about the potential applications of the instance segmentation.

6.2.1. Image editing

With the development of network technology, images and videos have gradually replaced text as the main means of expression. Therefore, image editing is a growing demand for internet users. Since instance segmentation can segment and distinguish different object instances, we can edit images more easier. For example, it is essential to segment the foreground when we want to resize the designated person (*e.g.* ant-man) in the film production.

6.2.2. Scene text detection

Most state-of-the-art methods of scene text detection are based on object detection. However, since text region detection can also be



(a) The ground truth of image. (b) The result of PolarMask[137]. (c) The result of YOLCAT[48].

(d) The result of BlendMask[138]. (e) The result of CondInst[136]. (f) The result of SOLO[126].

Fig. 24. The performance of different anchor-free instance segmentation methods on small objects in the COCO dataset. It is obvious that the quality of the ground-truth mask in the COCO dataset is also very rough and insufficient.

considered as a semantic segmentation containing complete location information, the prediction of bounding boxes is not essential. Also, text instances in scene images are often very close to each other, and it is difficult to separate them by semantic segmentation. Therefore, instance segmentation is needed to address this problem [163].

6.2.3. Autonomous driving

A fundamental point for autonomous driving is that the vehicle can analyze and determine the location of major categories of objects. Compared with only using object boxes to understand the actual scene, instance segmentation can better distinguish irregularly shaped objects (*e.g.* street lights, running people). Instance segmentation can be well determined to the boundary of the object, which reduces the problem of highly overlapping boxes(*e.g.* two people in parallel). We expect that more research on datasets, methods, and applications of autonomous driving based on instance segmentation will be available in the future.

6.2.4. Robots

Robots can greatly liberate human labor in industrial production. Most robots at this stage are still hard-coded for their functionality based on human experience, making their scope of use fixed to a particular function. Taking the robot hand grasping objects as an example, it is not enough to use only the results of semantic segmentation and object detection. Semantic segmentation can not tell the number and relative position of objects while object detection fails to grasp the detailed object outline information. Therefore, we expect that the robot's behavior can be freed from hard-coded forms when lightweight real-time instance segmentation methods are well-studied.

7. Conclusion

In this paper, we discuss the instance segmentation task in the following aspects: (1) We elaborate on the existing evaluation tools, including 11 datasets and 3 evaluation metrics. For each dataset, we enumerate the resolution composition, number of categories, applicable scenarios, and image composition statistics. For each metric, we also provide detailed formulas to illustrate its use and purpose. (2) We review the instance segmentation methods for the taxonomy listed in Table 4. The methods were investigated from three perspectives: method universality, supervision methods, and number of stages. Based on our survey, we conclude that two-stage methods have dominated general instance segmentation research in the last few years; multi-stage methods have higher accuracy with well-designed architectures; and in recent years, single-stage methods are also achieving better performance and speed tradeoffs. From our perspective, Transformer-based instance segmentation methods and single-phase instance segmentation methods will be the main directions for future research. (3) We briefly introduce several common backbone networks for instance segmentation methods: ResNet combined with FPN networks are always used to accomplish feature extraction at different levels; DCN networks can improve the feature extraction ability of deep nerual networks for irregularly shaped objects; Swin Transformer can extract better image features. (4) We summarize the performance of different methods on different datasets to facilitate comparative experiments for related researchers. The experimental results show that the existing instance segmentation methods are far from reaching the upper limit of performance, especially the single-stage instance segmentation methods. Meanwhile, Transformer-based instance segmentation methods will dominate the performance of instance segmentation for some time in the future. (5) Based on the current development of deep learning and the current status of instance segmentation research, we propose several meaningful future directions and potential applications of instance segmentation. In general, we expect a series of more innovative instance segmentation methods to emerge in the next few years.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE 2012, June, pp. 3354–3361.
[2] Z. Zhang, S. Fidler, R. Urtasun, Instance-level segmentation for autonomous driving with deep densely connected mrfs, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 669–677.
[3] B. De Brabandere, D. Neven, L. Van Gool, Semantic instance segmentation for autonomous driving, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2017, pp. 7–9.
[4] Y. Xu, Y. Li, M. Liu, Y. Wang, M. Lai, I. Eric, C. Chang, Gland instance segmentation by deep multichannel side supervision, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Cham 2016, October, pp. 496–504.
[5] Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, ... C. Chang, Gland instance segmentation using deep multichannel neural networks, IEEE Trans. Biomed. Eng. 64 (12) (2017) 2901–2912.
[6] H. Scharr, M. Minervini, A. Fischbach, S.A. Tsaftaris, Annotated image datasets of rosette plants, European Conference on Computer Vision. Zrich, Suisse 2014, July, pp. 6–12.
[7] M. Minervini, A. Fischbach, H. Scharr, S.A. Tsaftaris, Finely-grained annotated datasets for image-based plant phenotyping, Pattern Recogn. Lett. 81 (2016) 80–89.
[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems 2012, pp. 1097–1105.
[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv Preprint (2014) arXiv:1409.1556.
[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 770–778.
[11] S. Xie, R. Girshick, P. Dollr, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 1492–1500.
[12] P. Latorre-Carmona, V.J. Traver, J.S. Snchez, E. Tajahuerce, Online reconstruction-free single-pixel image classification, Image Vis. Comput. 86 (2019) 28–37.
[13] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[14] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollr, Focal loss for dense object detection, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 2980–2988.
[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 580–587.
[16] K. Tong, Y. Wu, F. Zhou, Recent advances in small object detection based on deep learning: a review, Image Vis. Comput. 97 (2020), 103910.
[17] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 801–818.
[18] Y. Zhang, D. Sidib, O. Morel, F. Mriaudeau, Deep multimodal fusion for semantic image segmentation: a survey, Image Vis. Comput. 105 (2021), 104042.
[19] Q. Tang, F. Liu, T. Zhang, J. Jiang, Y. Zhang, Attention-guided chained context aggregation for semantic segmentation, Image Vis. Comput. 115 (2021), 104309.
[20] B. Hariharan, P. Arbelez, R. Girshick, J. Malik, Simultaneous detection and segmentation, European Conference on Computer Vision, Springer, Cham 2014, September, pp. 297–312.
[21] K. He, G. Gkioxari, P. Dollr, R. Girshick, Mask r-cnn, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 2961–2969.
[22] A. Kirillov, K. He, R. Girshick, C. Rother, P. Dollr, Panoptic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 9404–9413.
[23] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, R. Urtasun, Upsnet: A unified panoptic segmentation network, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 8818–8826.
[24] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[25] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[26] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[27] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H.O. Song, S. Guadarrama, K.P. Murphy, Semantic instance segmentation via deep metric learning, arXiv Preprint (2017) arXiv:1703.10277.
[28] B. De Brabandere, D. Neven, L. Van Gool, Semantic instance segmentation with a discriminative loss function, arXiv Preprint (2017) arXiv:1708.02551.
[29] I. Armeni, S. Sax, A.R. Zamir, S. Savarese, Joint 2d-3d-semantic data for indoor scene understanding, arXiv Preprint (2017) arXiv:1702.01105.
[30] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, ... Y. Zhang, Matterport3d: learning from rgb-d data in indoor environments, arXiv Preprint (2017) arXiv:1709.06158.
[32] L. Yang, Y. Fan, N. Xu, Video instance segmentation, Proceedings of the IEEE/CVF International Conference on Computer Vision 2019, pp. 5188–5197.
[33] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021, pp. 8741–8750.
[37] W. Wang, R. Yu, Q. Huang, U. Neumann, Sgpn: Similarity group proposal network for 3d point cloud instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 2569–2578.
[38] X. Wang, S. Liu, X. Shen, C. Shen, J. Jia, Associatively segmenting instances and semantics in point clouds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 4096–4105.
[39] Q.H. Pham, T. Nguyen, B.S. Hua, G. Roig, S.K. Yeung, JSIS3D: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 8827–8836.
[40] J. Tan, K. Wang, L. Chen, G. Zhang, J. Li, X. Zhang, HCFS3D: hierarchical coupled feature selection network for 3D semantic and instance segmentation, Image Vis. Comput. 109 (2021), 104129.
[41] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, N. Trigoni, Learning object bounding boxes for 3d instance segmentation on point clouds, Advances in Neural Information Processing Systems 2019, pp. 6740–6749.
[42] P.O. Pinheiro, R. Collobert, P. Dollar, Learning to segment object candidates, Neural Information Processing Systems, 2015.
[43] A. Arnab, P.H. Torr, Bottom-up instance segmentation using deep higher-order crfs, arXiv Preprint (2016) arXiv:1609.02583.
[44] S. Kong, C.C. Fowlkes, Recurrent pixel embedding for instance grouping, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 9018–9028.
[45] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, ... C.C. Loy, Hybrid task cascade for instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 4974–4983.
[46] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv Preprint (2018) arXiv:1804.02767.
[47] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, Ssd: single shot multibox detector, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 21–37.
[48] D. Bolya, C. Zhou, F. Xiao, Y.J. Lee, Yolact: Real-time instance segmentation, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 9157–9166.
[49] D. Bolya, C. Zhou, F. Xiao, Y.J. Lee, Yolact++: better real-time instance segmentation, arXiv Preprint (2019) arXiv:1912.06218.
[50] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, ... C.L. Zitnick, Microsoft COCO: common objects in context, European Conference on Computer Vision, 2014.
[51] Z.H. Zhou, A brief introduction to weakly supervised learning, Natl. Sci. Rev. 5 (1) (2018) 44–53.
[52] X.J. Zhu, Semi-Supervised Learning Literature Survey, University of Wisconsin-Madison Department of Computer Sciences, 2005.
[53] A. Khoreva, R. Benenson, J. Hosang, M. Hein, B. Schiele, Simple does it: Weakly supervised instance and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 876–885.
[54] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, J. Jiao, Weakly supervised instance segmentation using class peak response, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 3791–3800.
[55] A. Bearman, O. Russakovsky, V. Ferrari, L. Fei-Fei, Whats the point: semantic segmentation with point supervision, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 549–565.
[56] R. Fan, M.M. Cheng, Q. Hou, T.J. Mu, J. Wang, S.M. Hu, S4net: single stage salient-instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 6103–6112.
[57] P. Follmann, R.K. Nig, P.H. Rtinger, M. Klostermann, T.B. Ttger, Learning to see the invisible: End-to-end trainable amodal instance segmentation, 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE 2019, January, pp. 1328–1336.
[58] K. Li, J. Malik, Amodal instance segmentation, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 677–693.
[59] S.H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, ... S.M. Hu, Pose2seg: Detection free human instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 889–898.
[60] G. Li, Y. Yu, Visual saliency based on multiscale deep features, Computer Vision and Pattern Recognition, 2015.
[61] G. Li, Y. Yu, Deep contrast learning for salient object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 478–487.
[62] N. Liu, J. Han, Dhsnet: deep hierarchical saliency network for salient object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 678–686.
[63] M. Cheng, F. Zhang, N.J. Mitra, X. Huang, S. Hu, RepFinder: finding approximately repeated scene elements for image editing, International Conference on Computer Graphics and Interactive Techniques, 2010.
[64] Y. Ma, L. Lu, H. Zhang, M. Li, A user attention model for video summarization, ACM Multimedia, 2002.
[65] A.M. Hafiz, G.M. Bhat, A survey on instance segmentation: state of the art, Int. J. Multim. Inform. Retriev. (2020) 1–19.
[66] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: a brief review, Comput. Intell. Neurosci. 2018 (2018).
[67] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 843–852.
[68] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, ... L. Feifei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[69] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, ... B. Schiele, The cityscapes dataset for semantic urban scene understanding, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 3213–3223.
[70] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[71] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the KITTI dataset, Int. J. Robot. Res. 32 (11) (2013) 1231–1237.
[72] L. Chen, S. Fidler, R. Urtasun, Beat the MTurkers: automatic image labeling from weak 3D supervision, Computer Vision and Pattern Recognition, 2014.
[73] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, International Conference on Computer Vision, 2011.
[74] M. Minervini, A. Fischbach, H. Scharr, S.A. Tsaftaris, Finely-grained annotated datasets for image-based plant phenotyping, Pattern Recogn. Lett. 81 (2016) 80–89.
[75] G. Neuhold, T. Ollmann, S. Rota Bulo, P. Kontschieder, The mapillary vistas dataset for semantic understanding of street scenes, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 4990–4999.
[76] K. Sirinukunwattana, J.P. Pluim, H. Chen, X. Qi, P.A. Heng, Y.B. Guo, ... A. Bhm, Gland segmentation in colon histology images: the glas challenge contest, Med. Image Anal. 35 (2017) 489–502.
[77] L. Qi, L. Jiang, S. Liu, X. Shen, J. Jia, Amodal instance segmentation with kins dataset, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 3014–3023.
[78] A. Gupta, P. Dollar, R. Girshick, Lvis: A dataset for large vocabulary instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 5356–5364.
[79] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, ... V. Ferrari, The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale, arXiv Preprint (2018) arXiv:1811.00982.
[80] R. Benenson, S. Popov, V. Ferrari, Large-scale interactive object segmentation with human annotators, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 11700–11709.
[81] V. Iglovikov, S.S. Seferbekov, A. Buslaev, A. Shvets, TernausNetV2: fully convolutional network for instance segmentation, CVPR Workshops, vol. 233, 2018, June, p. 237.
[82] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, L. Van Gool, Towards end-to-end lane detection: an instance segmentation approach, 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE 2018, June, pp. 286–291.
[83] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K.I. Komatsu, ... K. Doi, Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules, Am. J. Roentgenol. 174 (1) (2000) 71–74.
[84] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, I. Sachs, Automatic portrait segmentation for image stylization, Computer Graphics Forum, vol. 35, 2016, May, pp. 93–102 , No. 2.
[85] M. Ren, R.S. Zemel, End-to-end instance segmentation with recurrent attention, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 6656–6664.
[86] H. Scharr, M. Minervini, A.P. French, C. Klukas, D.M. Kramer, X. Liu, ... X. Yin, Leaf segmentation in plant phenotyping: a collation study, Mach. Vis. Appl. 27 (4) (2016) 585–606.
[87] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring r-cnn, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 6409–6418.
[88] Z. Cai, N. Vasconcelos, Cascade R-CNN: high quality object detection and instance segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[89] S. Wang, Y. Gong, J. Xing, L. Huang, C. Huang, W. Hu, RDSNet: a new deep architecture for reciprocal object detection and instance segmentation, arXiv Preprint (2019) arXiv:1912.05070.
[90] K. Gregor, I. Danihelka, A. Graves, D.J. Rezende, D. Wierstra, Draw: a recurrent neural network for image generation, arXiv Preprint (2015) arXiv:1502.04623.
[91] S.H.I. Xingjian, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, Advances in Neural Information Processing Systems 2015, pp. 802–810.
[92] B. Romera-Paredes, P.H.S. Torr, Recurrent instance segmentation, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 312–329.
[93] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 2359–2367.
[94] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems 2015, pp. 91–99.
[95] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[96] A. Newell, Z. Huang, J. Deng, Associative embedding: end-to-end learning for joint detection and grouping, Advances in Neural Information Processing Systems 2017, pp. 2277–2287.
[97] P. Arbelez, J. Pont-Tuset, J.T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pp. 328–335.
[98] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, ... A. Rabinovich, Going deeper with convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 1–9.
[99] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, PatchMatch: a randomized correspondence algorithm for structural image editing, ACM Trans. Graph. 28 (3) (2009) 24.
[100] P.O. Pinheiro, T.Y. Lin, R. Collobert, P. Dollr, Learning to refine object segments, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 75–91.
[101] X. Chen, R. Girshick, K. He, P. Dollr, Tensormask: A foundation for dense object segmentation, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 2061–2069.
[102] B. Hariharan, P. Arbelez, R. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 447–456.
[103] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 8759–8768.
[104] T.Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 2117–2125.
[105] R. Girshick, Fast r-cnn, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 1440–1448.
[106] J. Uhrig, M. Cordts, U. Franke, T. Brox, Pixel-level encoding and depth layering for instance-level semantic labeling, German Conference on Pattern Recognition, Springer, Cham 2016, September, pp. 14–25.
[107] L.C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, H. Adam, Masklab: Instance segmentation by refining object detection with semantic and direction features, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 4013–4022.
[108] W. Xu, H. Wang, F. Qi, C. Lu, Explicit shape encoding for real-time instance segmentation, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 5168–5177.
[109] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vis. 1 (4) (1988) 321–331.
[110] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, X. Zhou, Deep snake for real-time instance segmentation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 8533–8542.
[111] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Appl. Soft Comput. 70 (2018) 41–65.
[112] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 3431–3440.
[113] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, S. Yan, Proposal-free network for instance-level object segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2978–2991.
[114] A. Arnab, P.H. Torr, Pixelwise instance segmentation with a dynamically instantiated network, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 441–450.
[115] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 5221–5229.
[116] D. Neven, B.D. Brabandere, M. Proesmans, L.V. Gool, Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 8837–8845.
[117] J. Uhrig, E. Rehder, B. Frhlich, U. Franke, T. Brox, Box2pix: single-shot instance segmentation by assigning pixels to object boxes, 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE 2018, June, pp. 292–299.
[118] Z. Zhang, A.G. Schwing, S. Fidler, R. Urtasun, Monocular object instance segmentation and depth ordering with cnns, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 2614–2622.
[119] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, Y. Lu, Affinity derivation and graph merge for instance segmentation, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 686–703.
[120] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, K. Huang, Ssap: Single-shot instance segmentation with affinity pyramid, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 642–651.
[121] S. Liu, J. Jia, S. Fidler, R. Urtasun, Sgn: Sequential grouping networks for instance segmentation, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 3496–3504.
[122] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 3150–3158.
[123] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 6154–6162.
[124] A. Salvador, M. Bellver, V. Campos, M. Baradad, F. Marques, J. Torres, X. Giro-i-Nieto, Recurrent neural networks for semantic instance segmentation, arXiv Preprint (2017) arXiv:1712.00617.
[125] J. Dai, K. He, Y. Li, S. Ren, J. Sun, Instance-sensitive fully convolutional networks, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 534–549.
[126] X. Wang, T. Kong, C. Shen, Y. Jiang, L. Li, Solo: segmenting objects by locations, arXiv Preprint (2019) arXiv:1912.04488.
[127] X. Wang, R. Zhang, T. Kong, L. Li, C. Shen, SOLOv2: dynamic, faster and stronger, arXiv Preprint (2020) arXiv:2003.10152.

[128] R. Liu, J. Lehman, P. Molino, F.P. Such, E. Frank, A. Sergeev, J. Yosinski, An intriguing failing of convolutional neural networks and the coordconv solution, Advances in Neural Information Processing Systems 2018, pp. 9605–9616.
[129] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 764–773.
[130] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 9308–9316.
[131] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 7132–7141.
[132] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, pp. 3146–3154.
[133] Y. Lee, J. Park, CenterMask: Real-time anchor-free instance segmentation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 13906–13915.
[134] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, ... A. Desmaison, Pytorch: an imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 2019, pp. 8026–8037.
[135] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, ... M. Kudlur, Tensorflow: A system for large-scale machine learning, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) 2016, pp. 265–283.
[136] Z. Tian, C. Shen, H. Chen, Conditional convolutions for instance segmentation, arXiv Preprint (2020) arXiv:2003.05664.
[137] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, ... P. Luo, Polarmask: Single shot instance segmentation with polar representation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 12193–12202.
[138] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, Y. Yan, BlendMask: Top-down meets bottom-up for instance segmentation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 8573–8581.
[139] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 9627–9636.
[140] H. Ying, Z. Huang, S. Liu, T. Shao, K. Zhou, Embedmask: embedding coupling for one-stage instance segmentation, arXiv Preprint (2019) arXiv:1912.01954.
[141] Q. Li, A. Arnab, P.H. Torr, Weakly-and semi-supervised panoptic segmentation, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 102–118.
[142] C. Rother, V. Kolmogorov, A. Blake, "GrabCut" interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.
[143] K.K. Maninis, J. Pont-Tuset, P. Arbelez, L. Van Gool, Convolutional oriented boundaries: from image segmentation to high-level tasks, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 819–833.
[144] T. Pham, V.B. Kumar, T.T. Do, G. Carneiro, I. Reid, Bayesian semantic instance segmentation in open set world, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 3–18.
[145] M. Bellver Bueno, A. Salvador Aguilera, J. Torres Vials, X. Gir Nieto, Budget-aware semi-supervised semantic and instance segmentation, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019 2019, pp. 93–102.
[146] R. Hu, P. Dollr, K. He, T. Darrell, R. Girshick, Learning to segment every thing, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 4233–4241.
[147] C. Michaelis, I. Ustyuzhaninov, M. Bethge, A.S. Ecker, One-shot instance segmentation, arXiv Preprint (2018) arXiv:1811.11507.
[148] I.H. Laradji, N. Rostamzadeh, P.O. Pinheiro, D. Vzquez, M. Schmidt, Instance segmentation with point supervision, arXiv Preprint (2019) arXiv:1906.06392.
[149] H.S. Fang, J. Sun, R. Wang, M. Gou, Y.L. Li, C. Lu, Instaboost: Boosting instance segmentation via probability map guided copy-pasting, Proceedings of the IEEE International Conference on Computer Vision 2019, pp. 682–691.
[150] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, K. Murphy, Towards accurate multi-person pose estimation in the wild, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 4903–4911.
[151] Z. Cao, T. Simon, S.E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 7291–7299.
[152] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 7103–7112.
[153] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.
[154] S. Zhang, J. Yang, B. Schiele, Occluded pedestrian detection through guided attention in cnns, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition 2018, pp. 6995–7003.
[155] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, Y. Yang, Improving person re-identification by attribute and identity learning, Pattern Recogn. 95 (2019) 151–161.
[156] L. Zheng, Y. Huang, H. Lu, Y. Yang, Pose-invariant embedding for deep person re-identification, IEEE Trans. Image Process. 28 (9) (2019) 4500–4509.
[157] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, European Conference on Computer Vision, Springer, Cham 2016, October, pp. 483–499.
[158] G. Papandreou, T. Zhu, L.C. Chen, S. Gidaris, J. Tompson, K. Murphy, Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model, Proceedings of the European Conference on Computer Vision (ECCV) 2018, pp. 269–286.
[159] S. Tripathi, M. Collins, M. Brown, S. Belongie, Pose2instance: harnessing keypoints for person instance segmentation, arXiv Preprint (2017) arXiv:1704.01152.
[160] K. Li, B. Hariharan, J. Malik, Iterative instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 3659–3667.
[161] Y. Zhu, Y. Tian, D. Metaxas, P. Dollr, Semantic amodal segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 1464–1472.
[162] G. Li, Y. Xie, L. Lin, Y. Yu, Instance-level salient object segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, pp. 2386–2395.
[163] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: detecting scene text via instance segmentation, arXiv Preprint (2018) arXiv:1801.01315.
[164] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, K. Cho, Augmentation for small object detection, arXiv Preprint (2019) arXiv:1902.07296.
[165] H. Zhang, Y. Tian, K. Wang, W. Zhang, F.Y. Wang, Mask SSD: an effective single-stage approach to object instance segmentation, IEEE Trans. Image Process. 29 (2019) 2078–2093.
[166] Y. Wang, Z. Xu, H. Shen, B. Cheng, L. Yang, Centermask: single shot instance segmentation with point representation, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, pp. 9313–9321.
[167] F. Sultana, A. Sufian, P. Dutta, Evolution of image segmentation using deep convolutional neural network: a survey, Knowl.-Based Syst. 201 (2020), 106062.
[168] S. Minaee, Y.Y. Boykov, F. Porikli, A.J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.
[169] D. Tian, Y. Han, B. Wang, T. Guan, H. Gu, W. Wei, Review of object instance segmentation based on deep learning, J. Electron. Imaging 31 (4) (2021), 041205.
[170] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, ... I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 2017, pp. 5998–6008.
[171] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang, Y. Shan, ... W. Liu, Instances as queries, Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, pp. 6910–6919.
[172] B. Dong, F. Zeng, T. Wang, X. Zhang, Y. Wei, SOLQ: segmenting objects by learning queries, arXiv Preprint (2021) arXiv:2106.02351.
[173] J. Hu, L. Cao, Y. Lu, S. Zhang, Y. Wang, K. Li, ... R. Ji, ISTR: end-to-end instance segmentation with transformers, arXiv Preprint (2021) arXiv:2105.00637.
[174] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdisc. Rev. 2 (4) (2010) 433–459.
[175] N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform, IEEE Trans. Comput. 100 (1) (1974) 90–93.
[176] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, arXiv Preprint (2021) arXiv:2112.01527.
[177] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, ... B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv Preprint (2021) arXiv:2103.14030.
[178] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, ... B. Guo, Swin transformer V2: scaling up capacity and resolution, arXiv Preprint (2021) arXiv:2111.09883.
[179] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, European Conference on Computer Vision, Springer, Cham 2020, August, pp. 213–229.
[180] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, ... N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, arXiv Preprint (2020) arXiv:2010.11929t_COMMENT>.
[181] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, ... P. Luo, Sparse r-cnn: end-to-end object detection with learnable proposals, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021, pp. 14454–14463.
[182] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, arXiv Preprint (2020) arXiv:2010.04159.
[183] W. Zhang, J. Pang, K. Chen, C.C. Loy, K-net: towards unified image segmentation, Adv. Neural Inf. Proces. Syst. 34 (2021).