International Journal of Multimedia Information Retrieval
https://doi.org/10.1007/s13735-020-00195-x

TRENDS AND SURVEYS

A survey on instance segmentation: state of the art

Abdul Mueed Hafiz[1] · **Ghulam Mohiuddin Bhat**[1]

Received: 10 December 2019 / Revised: 13 May 2020 / Accepted: 19 June 2020

Abstract
Object detection or localization is an incremental step in progression from coarse to fine digital image inference. It not only provides the classes of the image objects, but also provides the location of the image objects which have been classified. The location is given in the form of bounding boxes or centroids. Semantic segmentation gives fine inference by predicting labels for every pixel in the input image. Each pixel is labelled according to the object class within which it is enclosed. Furthering this evolution, instance segmentation gives different labels for separate instances of objects belonging to the same class. Hence, instance segmentation may be defined as the technique of simultaneously solving the problem of object detection as well as that of semantic segmentation. In this survey paper on instance segmentation, its background, issues, techniques, evolution, popular datasets, related work up to the state of the art and future scope have been discussed. The paper provides valuable information for those who want to do research in the field of instance segmentation.

Keywords Instance segmentation · Object detection · Convolutional neural networks · Deep learning

1 Introduction

1.1 Background

Semantic segmentation [1], vis-à-vis its relation to deep learning [2–7], can be understood by consideration of the fact that the former is not an isolated research area, rather an incremental research approach in the continuation of coarse inference towards fine inference. Its origins can be traced to automated classification approaches [8–21]. Classification in turn may be defined as the process of predicting a complete input, i.e. prediction of class of an object in the image or providing a list of classes of objects in an image according to their classification scores. Detection or object localization is an incremental step from coarse to fine inference, which provides not only classes of the image objects but also gives the location of the classified image objects in the form of bounding boxes or centroids. The goal of semantic segmentation is to obtain fine inference by predicting labels for each image pixel. Every pixel is class labelled according to the object or region within which it is enclosed. Going in this direction, instance segmentation provides different labels for separate instances of objects belonging to the same object class. Thus, instance segmentation can be defined as the task of finding simultaneous solution to object detection as well as semantic segmentation [22]. Part-based segmentation continues the evolution of this research by decomposing each of the segmented objects into their respective sub-components. Figure 1 depicts this image segmentation evolution. In this survey paper, we will focus on instance segmentation, its techniques, its frequently used datasets, related work and its potential scope for the future.

1.2 Issues

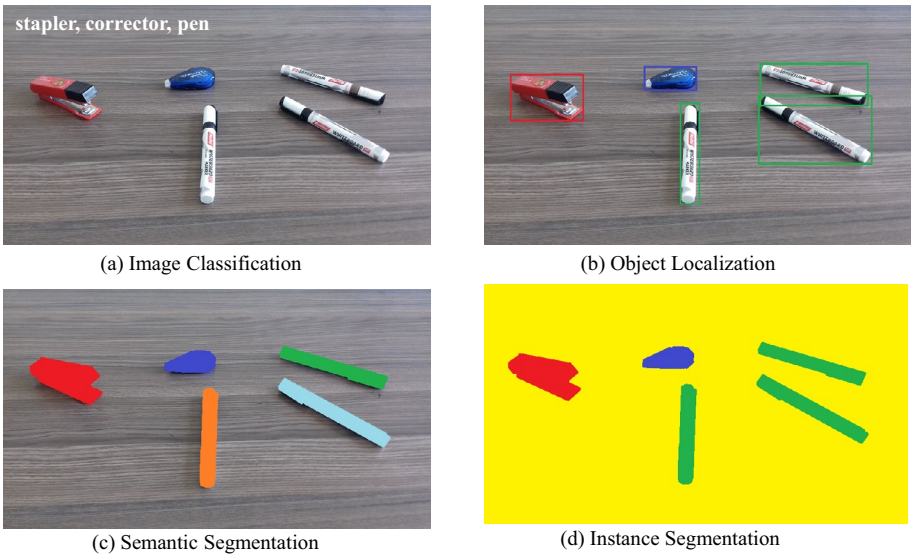
The idea of semantic segmentation is developing a technique/algorithm that performs well in the two domains of *better segmentation accuracy* and *better segmentation efficiency*. Better segmentation accuracy encompasses accurate localization and recognition of objects in images/frames, with the result that a large variety of object-related categories in real scenario can be distinguished (i.e. better distinctiveness), and that instances of objects belonging to same class which are subject to intra-class appearance variation

✉ Abdul Mueed Hafiz
mueedhafiz@uok.edu.in

Ghulam Mohiuddin Bhat
drgmbhat@uok.edu.in

[1] Department of Electronics and Communication Engineering, Institute of Technology, University of Kashmir, Srinagar, J&K 190006, India

Published online: 03 July 2020



(a) Image Classification (b) Object Localization

(c) Semantic Segmentation (d) Instance Segmentation

Fig. 1 Object recognition evolution: from coarse- to fine-grained inference: **a** image classification, **b** object detection or localization, **c** semantic segmentation, **d** instance segmentation

may be localized and recognized (i.e. better robustness). Better segmentation efficiency refers to computational cost of the segmentation algorithm. It refers efficient real-time computational costs like acceptable memory/storage requirements and lesser burden on the processor(s).

One of the important components in an object detector for segmentation is good feature representation which is of primary importance in object detection [23–27]. Previously, a lot of effort was put in designing local descriptors (like SIFT [28] and HOG [29]) and in exploring approaches (like Bag of Words ([30] and Fisher Vector [31]) in order to group and to abstract descriptors into high-level representations for emerging the discriminative parts. The downside was that these feature representation methods needed handcrafted fine engineering and a large amount of domain expertise. As against this, methods based on deep learning (like Deep CNNs) are able to learn powerful representations of features with various abstraction levels from images [4, 32]. Subsequently, the problem of feature representation has been transferred to the development of better performing network architectures and more optimized training procedures.

The trend in evolution of network architecture is of increasing depth: AlexNet [33] had eight layers, VGGNet [34] had 16 layers, and more recently ResNet [35] and DenseNet [36] both have more than 100 layers. In fact, it was VGGNet and GoogLeNet [37] which showed that with increasing the network depth, it is possible to increase the network's power of representation. Deep networks like AlexNet, OverFeat [38], ZFNet [39], and VGGNet have an extremely large number of parameters although they have few layers. This can be attributed to enormous number of parameters coming from the fully connected layers. Newer networks like Inception [40], ResNet, and DenseNet, although having a great depth, have far fewer parameters by avoiding the use of fully connected layers.

Deep CNN-based detectors like RCNN [24], Fast RCNN [41], Faster RCNN [42], and YOLO [43] usually use the deep CNN architectures and subsequently use features from the topmost CNN layer for object representation. But there is a problem. Detection of objects across various scales is a big challenge. In order to address this issue, the detector is run over a pyramid of images [24, 44, 45]. Though this approach typically leads to more accurate detection, however, it suffers from limitations of inference time and also that of computational resources like memory.

Instance segmentation for small objects remains an issue. CNNs compute features in a layer-by-layer hierarchy; hence, the sub-sampling layers in the hierarchy of features by default lead to an inbuilt multi-scale pyramid and in turn produce maps of features at various resolutions. This behaviour leads to issues [46–48]. For example, higher CNN layers have a broad receptive field with more robustness to variations in pose, deformation, and illumination, but resolution is lower and detail is lost. As against this, lower CNN layers have a narrow receptive field with richer detail, but resolution is higher and sensitivity to semantics is much lesser. Semantic attributes of objects emerge in various layers, which in turn depend on size of the objects. Hence, if an object is small, its detail in earlier CNN layers is less, and the same can almost disappear in higher layers. This issue makes small object detection quite challenging. Various techniques have been proposed to tackle this issue, e.g. dilated convolution [49–51], and increasing resolution of features. However, these techniques lead to higher computational complexity. Also, if the object is large, then its semantic concept will be reflected in higher layers. Many techniques [48, 52–54] have been developed for improving the detection accuracy by using various CNN layers.

Another issue is the handling of geometric transformations. DCNNs by nature cannot be spatially invariant with

regard to geometric transformation [55–57]. Local max pooling in DCNN layers allows the networks to have some degree of translational invariance. In spite of this, the intermediate maps of features are not actually transformation invariant [55].

Handling of occlusions is also an issue. In real images, occlusions commonly occur which result in loss of information from instances of objects. For this problem, deformable ROI pooling [58–60] and deformable convolution [58] were proposed. These techniques alleviate occlusions by making fixed geometric structures more flexible. Wang et al. [61] also proposed training of an adversarial network [62]. In spite of these efforts, the problem of occlusions has not been even merely solved. Using GANs in order to address this problem looks promising.

Handling of image degradations is also an issue. Noise in real-world images is a problem. This is usually caused by problems in lighting, low quality in cameras, compression in images, etc. Although low quality in images tends to degrade their recognition, most contemporary techniques are benchmarked in a degradation free environment. This is justified by the fact that image databases like ImageNet [63], Microsoft COCO [64], PASCAL VOC [65], etc. all use high-quality images. As of now, it is observed that there is a miniscule body of work to address this issue.

1.3 Instance segmentation

Instance segmentation has come to be one of the relatively important, complex, and challenging areas in machine vision research. Aimed at predicting the object class label and the pixel-specific object instance mask, it localizes different classes of object instances present in various images. Instance segmentation aims to help largely robotics, autonomous driving, surveillance, etc.

With the advent of deep learning [4, 5], more specifically convolutional neural networks (CNNs) [33, 39, 66], many instance segmentation frameworks were proposed, for example [6, 67–69], in which the segmentation accuracy grew rapidly [65]. Mask R-CNN [67] is a straightforward and efficient instance segmentation approach. Taking a lead from Fast/Faster R-CNN [41, 42], a fully convolutional network (FCN) has been used to predict segmentation masks, side by side with box regression and object classification. For high performance, feature pyramid network (FPN) [53] has been used to extract stage-wise network features, in which a top-down network path having lateral connections has been used to obtain features which are semantically strong.

Some relatively new datasets provide adequate room for improvement in proposed techniques. Microsoft's Common Objects in Context or COCO dataset [64] has 200 k images. Many instances having complicated spatial layout have been captured in the images of this dataset. Also, the Cityscapes dataset [70] and the Mapillary Vistas Dataset or MVD [71] have street scene images containing a large number of traffic objects per image. Blurring, occlusion, and minute instances are found in the images of these datasets.

Many principles have been proposed for network design for classifying images. The same are also substantially useful for object recognition. Some examples in this regard are shortening of information path [35, 72], using dense connections [36], increasing information path flexibility and diversity by creation of parallel paths [73, 74], etc.

From the 100 + papers covered in this survey, there are 4 top-level research clusters which we describe next.

2 Instance segmentation techniques: a taxonomy

2.1 Classification of mask proposals

Figure 2 shows the general framework for this class of techniques.

2.1.1 Bottom-up mask proposals

Before being popularized by COCO [64], instance segmentation, in its modern sense, was introduced by Hariharan et al. [75]. The proposed technique involved generation of mask proposals [76, 77], followed by classification of the generated proposals [75]. Earlier, this classification of mask proposals approach was used elsewhere. As a example, selective search [76] used this technique for obtaining box detections and performing semantic segmentation. These techniques could conveniently be used for instance segmentation.

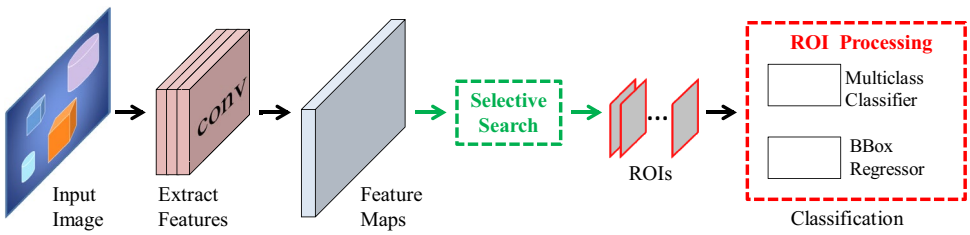


Fig. 2 General framework for *classification of mask proposals* techniques

2.1.2 Deep learning

The prior techniques depended on doing bottom-up mask proposal generation, before deep learning became popular [76, 77]. Subsequently, the former were replaced by new techniques having a more efficient structure, which came along with such as RCNN [24]. In spite if their better segmentation accuracy, RCNN and the other techniques inside this band suffered from some issues. For example, training was based on a multistage pipeline, which was slow and was difficult to optimize, due to the need to train each stage separately. Features had to be extracted for each proposal in every image from the CNN, leading to storage, time and detection scale issues, respectively. Testing was also slow due to the need to extract CNN features. Subsequently, RCNN was followed by Fast RCNN [41] and Faster RCNN [42], which addressed its problems.

2.2 Detection followed by segmentation

The popular approach for instance segmentation involves object detection using a box followed by object box segmentation [67, 68, 72, 78]. Figure 3 shows the general framework for this class of techniques.

2.2.1 Mask-based techniques

One of the most successful techniques in this regard can be Mask RCNN [67]. Mask RCNN extends the Faster R-CNN [42] detection algorithm using a relatively simple mask predictor. Mask RCNN is simple to train, has better generalization, and only adds a small computational overhead to Faster R-CNN. The former runs at 5 FPS. Instance segmentation approaches that bank on Mask R-CNN [79–81] have shown promising results in recent instance segmentation challenges [64, 70, 71].

2.2.2 Other techniques

For the purpose of detecting object bounding boxes, the following techniques have been used.

- sliding window techniques [82–84]
- region-based techniques [41, 42]

In spite of their strengths of the techniques discussed in Sects. 2.2.1 and 2.2.2, they depend on pipelining leading partially to some of the drawbacks discussed in Sect. 2.1.

2.3 Labelling pixels followed by clustering

Another approach to instance segmentation, e.g. [69, 85, 86], uses techniques created for the task of semantic segmentation [7, 87]. This approach involves categorical labelling of every image pixel. This is followed by grouping pixels into object instances using a clustering algorithm. Figure 4 shows the general framework.

The approach benefits from recent positive developments in semantic segmentation which can predict a high-resolution object mask. In comparison with detection-followed-by-segmentation techniques, labelling-pixels-followed-by-clustering methods have lesser accuracy on frequently used benchmarks [64, 70, 71]. Due to intense computation necessitated by pixel labelling, more computational power is generally required.

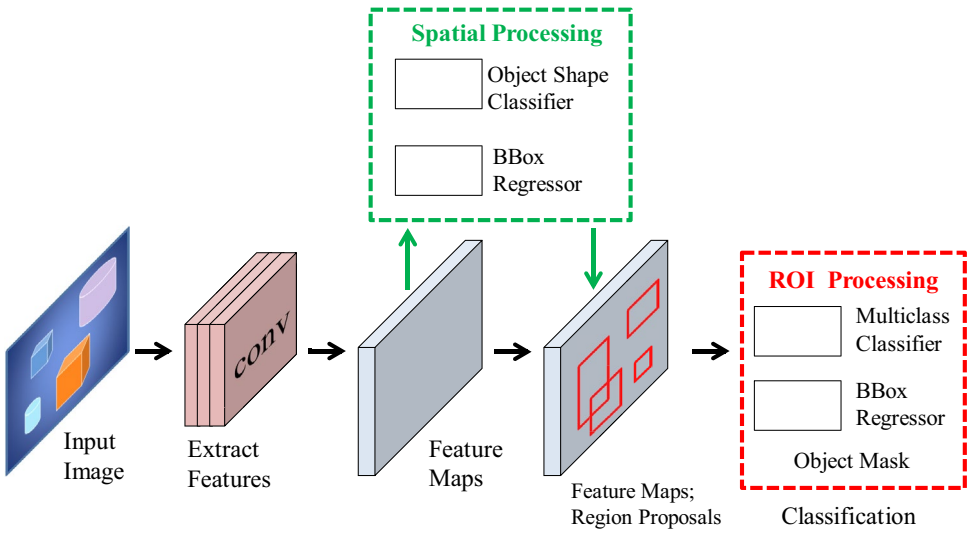


Fig. 3 General framework for *detection followed by segmentation* techniques

2.4 Dense sliding window methods

The general framework for this class of techniques is shown in Fig. 5.

2.4.1 Class agnostic mask generation techniques

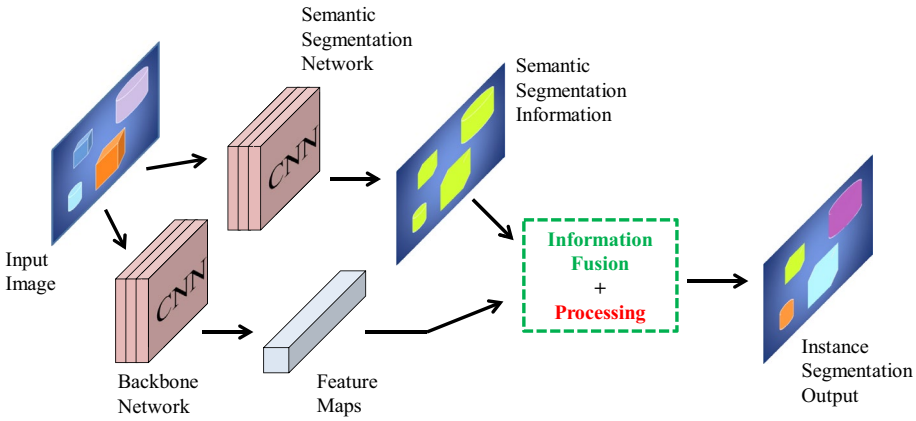
These techniques, e.g. DeepMask [88, 89], InstanceFCN [90], etc. use CNNs for mask proposal generation by dense sliding window techniques.

2.4.2 TensorMask

TensorMask [91] uses a different architecture as compared to the techniques discussed in Sect. 2.4.1 leading to better performance. Also, unlike DeepMask and InstanceFCN, TensorMask involves classification for multiple classes, which is done in parallel with predicting masks. This feature makes it useful for instance segmentation.

The techniques (discussed in Sects. 2.4.1 and 2.4.2), e.g. TensorMask, have a decent performance on benchmarking datasets like COCO; however, the algorithmic complexity is an issue.

Tabular taxonomy of the notable methods discussed in Sect. 2 is given in Table 1.



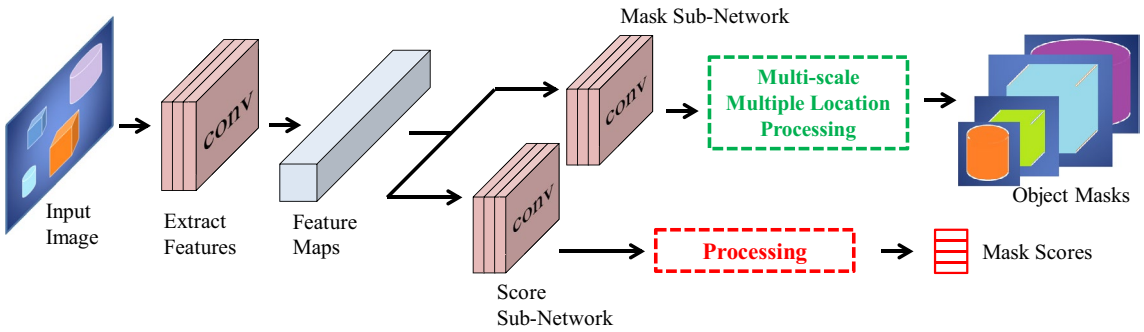


Fig. 4 General framework for *labelling pixels followed by clustering* techniques

Fig. 5 General framework for *dense sliding window methods*

Table 1 Tabular taxonomy of the notable techniques mentioned in Sect. 2

Group	Technique
Classification of mask proposals	RCNN, Fast RCNN, Faster RCNN
Detection followed by segmentation	HTC, PANet, Mask RCNN, Mask Scoring RCNN, MPN, YOLACT
Labelling pixels followed by clustering	Deep Watershed Transform, Instance Cut
Dense sliding window methods	Deep Mask, Instance FCN, Tensor Mask

3 The evolution of instance segmentation

Although there are a large number of techniques applied to instance segmentation, the notable methods in light of the discussion from Sect. 2 are discussed below which are also shown in the timeline in Fig. 6.

3.1 RCNN

After being inspired by the breakthroughs in image classification, obtained by use of CNNs and the success of selective search technique in region proposals for manually generated features [76], Girshick et al. [24] were one of the first to explore CNNs for instance segmentation [27]. They developed the RCNN technique which integrated AlexNet [33] along with a region proposal using the selective search technique [76]. Training an RCNN model consists of the following steps. The first step involves computing class agnostic region proposals obtained using selective search. The next step is CNN model fine-tuning which consists of using the region proposals for fine-tuning a pre-trained CNN model like AlexNet. Next, a set of class specific support vector machine (SVM) classifiers are trained on features extracted from the CNN which replace the softmax classifier learned by fine-tuning. This is followed by class specific bounding box regressor training for each object class using features obtained from the CNN.

Although RCNN achieved high object detection quality, it has some notable drawbacks. For example, training in a multistage pipeline is slow and difficult because each individual stage has to be trained separately. Also, for training the SVM classifier and BBox regressor, respectively, more resources and time are needed. Finally, testing is slow, because features from CNN have to be extracted for each object proposal in every testing image sans shared computation.

These problems with RCNN inspired development of other techniques, which led to birth of improved detection frameworks, e.g. Fast RCNN, and Faster RCNN.

3.2 Fast RCNN

Fast RCNN [41] addressed some of the issues of RCNN and subsequently improved its object detection ability [27]. Fast RCNN uses end-to-end training of the detector. It does this by streamlining the training process by simultaneous learning of the softmax classifier and of the class specific BBox regression, rather than individually training various components of the model as done in RCNN. Fast RCNN shares the computation of convolution among region proposals and subsequently adds an ROI pooling layer between the last convolution layer and the first fully connected layer in order to extract features for every region proposal. ROI pooling uses the concept of feature-level warping in order to achieve image-level warping. The ROI pooling layer features are given into a sequence of fully connected layers which finally branch into 2 layers, viz. object category prediction softmax probability and class proposal refinement offsets. In comparison with RCNN, Fast RCNN improves the efficiency to a large extent, e.g. by 3 times training speed and by 10 times in testing speed.

3.3 MultiPath Network

In the MultiPath Network technique [72], three modifications have been done to the standard Fast R-CNN model. First, skip connections [47, 92, 93] have been incorporated which give the object detector access to features of different network layers. Second, a foveal element has been added for exploitation of context of objects at different resolutions. And finally, a loss function of integral nature has been added. The network has been adjusted in order to improve localization of the instance segmentation masks. As a result

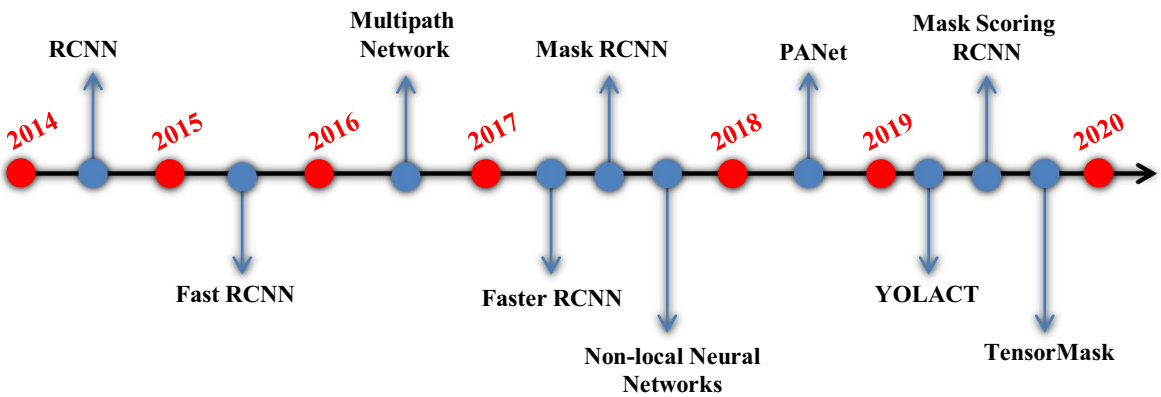


Fig. 6 Timeline for notable techniques in instance segmentation

of the modifications mentioned, a multiple-path information flow gets created in the network, hence the name "MultiPath" network. The MultiPath Network has been coupled with DeepMask model, the latter being quite convenient for localizing object even if they are small. The pipeline has also been adapted for predicting masks as well as bounding boxes. The new model demonstrated improvement over Fast R-CNN model [41] in overall by 66% and by four times on small object instances. It ranked second in both 2015 detection and segmentation challenges for the COCO dataset.

3.4 Faster RCNN

In spite of the fact that Fast RCNN led to significant speed up in detection, it still relied on external region proposals, for which computation was the speed bottleneck in Fast RCNN [27]. At that point in time, work [94, 95] showed that CNNs have the ability of object localization in convolutional layers, an ability which is weakened in fully connected layers. Hence, it was found to be feasible that selective search could be replaced by a CNN for production of region proposals. Subsequently, Faster RCNN model was proposed by Ren et al. [42] which had a Region Proposal Network (RPN) for generation of region proposals and this was efficient and accurate. The same backbone network was used, taking features from the last shared convolutional layer in order to accomplish region proposal by RPN and region classification by Fast RCNN.

3.5 Mask R-CNN

In [67], the authors present Mask R-CNN, a relatively simple and flexible model for instance segmentation. The model conveniently performs instance segmentation by object detection with simultaneous generation of high-quality masks. Mask R-CNN furthers the Faster R-CNN [42]. Normally, Faster R-CNN has a branch for object bounding box recognition. Mask R-CNN adds an object mask prediction branch in parallel as an improvement. The head architecture using an FPN backbone is shown in Fig. 7.

The proposed model is relatively easy to train, while adding a small computational load to the Faster R-CNN model which runs at 5 fps. Another advantage of Mask R-CNN is ease of generalization with regard to other related tasks. For example, Mask R-CNN allows estimation of human poses in a similar environment. Mask R-CNN achieved first position in all the 3 COCO suite 2016 challenges, viz. instance segmentation, detection of bounding boxes, detection of person key points. The model, as demonstrated by the authors, outperformed other state-of-the-art models on each COCO task for the 2016 challenge of the dataset.

3.6 MaskLab

MaskLab [22] improves Faster R-CNN [42] and produces 2 additional outputs, viz. semantic segmentation and instance centre direction [96]. The prediction boxes given by Faster R-CNN bring the instances of the objects having different scales to a canonical scale, and then, MaskLab does foreground and background segmentation inside each prediction box by using both semantic segmentation and direction prediction. The semantic segmentation prediction by encoding the pixel-wise classification data along with background class has been adopted in order to distinguish between objects of various semantic classes. This technique removes the duplicate background encoding in [68]. In addition to this, the direction prediction has been used for separation of instances of an object with common semantic label. MaskLab employs same assembling technique as used in [68, 90] for collection of the direction

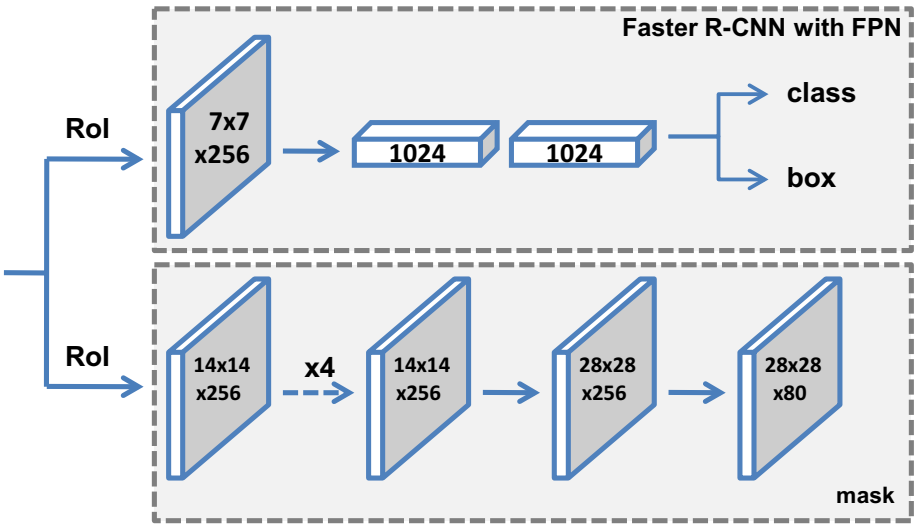


Fig. 7 Head architecture: extension of existing faster RCNN head with FPN backbone to which a mask branch is added. Numbers denote spatial resolutions and channels. Arrows denote convolution, deconvolution, or fully connected layers as can be inferred from context [convolution preserves spatial dimension while deconvolution increases it) [67]

information to get rid of the complicated template matching technique used in [96]. In addition to this, by taking motivation from the recent advances in both segmentation and detection, the proposed model further incorporates atrous convolution [97] for extraction of denser features maps, hypercolumn features [92] for the purpose of refining mask segmentation [98], multi-grid technique [58, 99, 100] for capturing various context scales, and a new TensorFlow technique [101], deformable cropping and resize, having been inspired by deformable pooling operation [58]. The performance of their model is comparable to other state-of-the-art models.

3.7 Non-local neural networks

Non-local means [4] is a filtering technique which computes a weighted mean of all pixels in an image. In doing so, it allows distant pixels to contribute to the filtering response at a location which is based on path appearance similarity. This idea was successively developed by Block-matching 3D (BM3D) [102–105]. Long-range dependencies have been modelled by graphical models, e.g. conditional random fields (CRF) [106, 107]. The mean field inference in a CRF can be converted into a recurrent network and subsequently can be trained [108–112]. The authors of this work [113] claim that their technique is simpler and the collective work of theirs and others is related to graph neural networks [114]. They further claim that their work is related to the *self-attention* [115] method used in machine translation. A self-attention capsule calculates the response at a position in a sequence, e.g. a sentence, by looking at all positions and subsequently taking their weighted average inside an embedding space. Self-attention can be viewed as a non-local mean [116] and hence can thus their work connect self-attention in machine translation with the general class of non-local filtering operations applicable to image and video problems in machine vision. The authors claim that they address non-local modelling, which is a long-time important component of image processing [116, 117] and has been overlooked to a large extent in recent neural networks for machine vision, and that they address this issue.

It is notable that convolution and recurrent operations, both, are the building blocks which process a single spatially local neighbourhood at a given time. The authors of this approach [113] propose a family of non-locally operating building blocks for the purpose of capturing long-range pixel dependencies. The non-local mean operation [116] can be defined as a generic non-local operation in deep neural networks as:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{6}$$

Here i is the index of the output position whose response is to be calculated and j is the index which gives all possible positions. x is the input signal, and y is the output signal which has same size as x. f is a pair-wise function which computes a scalar value like affinity between i and all values of j. g is a unary operation which computes the representation of the signal at the input at position j. The response is normalized by $C(x)$. The authors claim that a non-local operation is a flexible building block which can be conveniently used with convolutional and recurrent neural networks, unlike fully connected layers that are usually used in the end of these networks. They have used this operation and built a rich architecture which is able to combine non-local and local information. Figure 8 shows a space–time non-local block.

For the purpose of classification of video frames, the authors claim that the non-locally operating models are competent or even capable of outperforming the state-of-the-art approaches used on the Kinetics and the Charades datasets. The technique has been demonstrated to improve instance segmentation on COCO dataset.

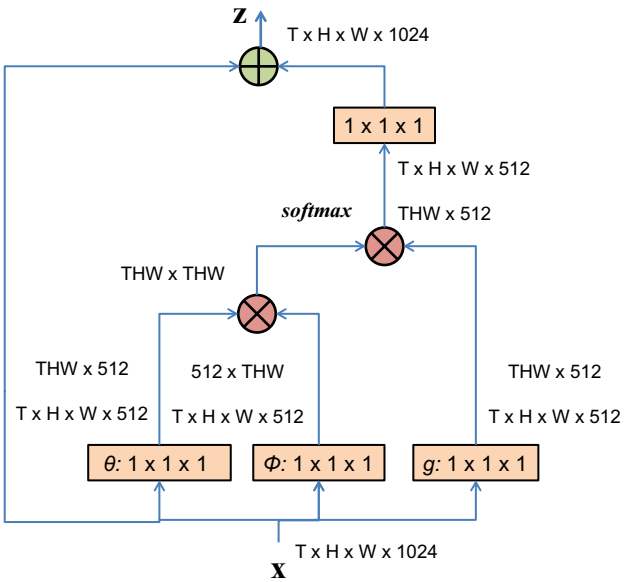


Fig. 8 A space–time non-local block. Feature maps are denominated by the shape of the tensors; for example, $T \times H \times W \times 1024$ is for 1024 channels. Matrix multiplication and element-wise sum are denoted by red circles and green circles, respectively. The softmax operation is performed on every row. The orange boxes denote $1 \times 1 \times 1$ convolution. An embedded Gaussian version with a bottleneck of 512 channels is shown above [113]

3.8 Path aggregation network (PANet)

The authors of [79] discuss that, however, the practice of using features from different layers in image recognition is well established [47, 48, 53, 82, 89, 118–123]. Accordingly, they take FPN [53] as baseline and enhance it significantly. They also discuss that some research groups [92, 93, 124, 125] concatenated feature grids from various layers for higher efficiency. But they note that a sequence of operations like normalization, concatenation, and dimension reduction are required to get useful new features. They claim that their design is much simpler while giving good results. Feature grid feature fusion has been used previously with feature maps on input with different scales [126]. The authors note that their technique uses single-scale input. End-to-end training is used. The authors note that some research groups [25, 124, 127] have used techniques of pooling features for every proposal with a foveal structure for exploitation of contextual information from regions with different resolutions, e.g. global pooling [118, 128, 129]. The authors note that they have used the mask prediction branch which also supports accessing global information in a novel way.

The framework of the proposed technique is shown in Fig. 9. The authors of PANet note that manner in which information flows or propagates in deep neural networks is very important. The authors of PANet Model [79] aim at boosting the flow of information in the proposal-based framework used for instance segmentation task. They improve the deep network hierarchy of features with specific signals related to localization in the lower layers. This process is referred to as bottom-up path augmentation. It leads to shorter information paths between the lower layers and the features at the top of the deep network. They also propose a technique referred to as adaptive feature pooling which relates the grid of features and features at all levels. Due to this technique, relevant information in every level of features flows to the subsequent sub-networks used for generating proposals. An alternate branched segment captures various proposal views in order to enhance the prediction of the generated masks.

The techniques proposed are relatively easy to implement and have relatively small overhead with regard to computational load. PANet achieved the first position in the 2017 Instance Segmentation Challenge Task for COCO and also achieved the second position in the task of object detection sans batch training for a large number of images.

3.9 Hybrid task cascade

Cascade combination of models has proved to be a useful approach for boosting performance of different tasks. Introducing cascading to instance segmentation is a challenging area. In this direction, the authors introduce Hybrid Task Cascade [81]. The authors begin with direct combination of Mask R-CNN and Cascade R-CNN which they call Cascade Mask R-CNN. Specifically, a mask branch on lines of the Mask R-CNN is added to each stage of Cascade R-CNN. The pipeline [81] is given by:

$$\begin{aligned} x_t^{\text{box}} &= P\left(x, r_{t-1}\right),\ r_t = B_t\left(x_t^{\text{box}}\right), \\ x_t^{\text{mask}} &= P\left(x, r_{t-1}\right),\ m_t = M_t\left(x_t^{\text{mask}}\right) \end{aligned} \tag{1}$$

In the above equation, x is the CNN feature of the network; x_t^{box} and x_t^{mask} indicate box and mask features obtained from x and the input region of interests (ROIs). P(·) is the pooling operator; for example, ROI align. B_t and M_t refer to the box head and mask head at the t^{th} stage; r_t and m_t refer to the corresponding box and mask predictions, respectively. By combination of the advantages of cascaded refinement, and the benefits from bounding boxes

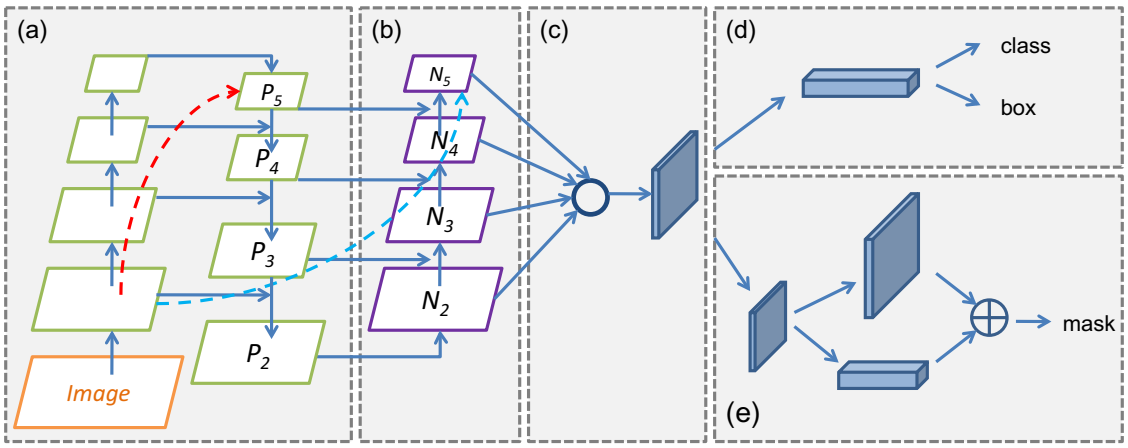


Fig. 9 PANet framework. **a** FPN backbone. **b** Bottom-up path augmentation. **c** Adaptive feature pooling. **d** Box branch. **e** Fully connected fusion [79]

and mask predictions, the technique improves the box AP, in comparison with Mask R-CNN and Cascade R-CNN. But the authors note that the performance is unsatisfying. The authors propose interleaving of the box branch and mask branch. This execution is expressed as:

$$\begin{aligned} x_t^{\text{box}} &= P\left(x, r_{t-1}\right), \; r_t = B_t\left(x_t^{\text{box}}\right), \\ x_t^{\text{mask}} &= P\left(x, r_t\right), \; m_t = M_t\left(x_t^{\text{mask}}\right) \end{aligned} \tag{2}$$

The authors note that by doing the above, the mask branch is able to take advantage of the updated predictions of the bounding boxes. The authors next introduce information flow among the mask branches by manner of feeding the masks of the previous stages to the new stages.

$$\begin{aligned} x_t^{\text{box}} &= P\left(x, r_{t-1}\right), \quad r_t = B_t\left(x_t^{\text{box}}\right), \\ x_t^{\text{mask}} &= P\left(x, r_t\right), \quad m_t = M_t\left(F(x_t^{\text{mask}}, m_{t-1}^{-})\right) \end{aligned} \tag{3}$$

where m_{t-1}^{-} refers to the intermediate features of M_{t-1} which is used as the representation of the mask at stage $t-1$. F is the combination function. The authors propose the following implementation, wherein they adopt ROI features prior to the deconvolution layer as the mask representation m_{t-1}^{-} with spatial size 14×14.

$$F\left(x_t^{\text{mask}}, m_{t-1}\right) = x_t^{\text{mask}} + G_t\left(m_{t-1}^{-}\right) \tag{4}$$

A notable contribution of this work is the use of spatial contexts by the addition of a new branch for prediction of per-pixel semantic segmentation for the entire image. Figure 10 shows the architecture of this branch.

The authors of the work discuss that they found the key to optimum cascading for instance segmentation was to take maximum advantage of the inverse relationship between object detection and object instance segmentation. Hybrid Task Cascade or HTC differs from conventional cascading in 2 important ways. First, instead of using refined cascading on the two tasks, HTC processes them in multiple stages in a combined manner. Second, it uses a fully convolutional segment for providing spatial context. This helps in distinguishing foreground from noisy background. The authors claim that HTC is able to learn more useful features by integration of features which are complementary, progressively with every stage. Without any fine tuning, an HTC model obtained 38.4% mask AP with 1.5% improvement w.r.t. a cascaded Mask R-CNN for the COCO dataset. Further, the proposed model achieved a score of 48.6 Mask AP on the test challenge subset of COCO, achieving the first position for the COCO 2018 Challenge in object detection.

3.10 GCNet

The authors of Global Context Network (GCNet) [130] note that the Non-Local Networks [113] (discussed in the original work given in Sect. 3.7) present a novel approach for capturing long-range dependencies by aggregation of query-specific global context for every query location in the image. In spite of this, they found through a thorough mathematical analysis that global contexts which are modelled by non-local networks are almost the same for various query positions throughout the image. The authors claim to have taken advantage of this finding in order to create a simple network which is based on query-independent formulation. They claim that the proposed network maintains the accuracy of Non-local Networks but with much lesser computational expenditure. The authors of GCNet note that their design is similar in structure to Squeeze-Excitation Network (SENet) [131]. They propose unification into a three-step general model for modelling of global context. Inside the general model, a more efficient instantiation, referred to as the Global Context (GC) Block, has been designed. The block is lightweight and is able to efficiently model the global

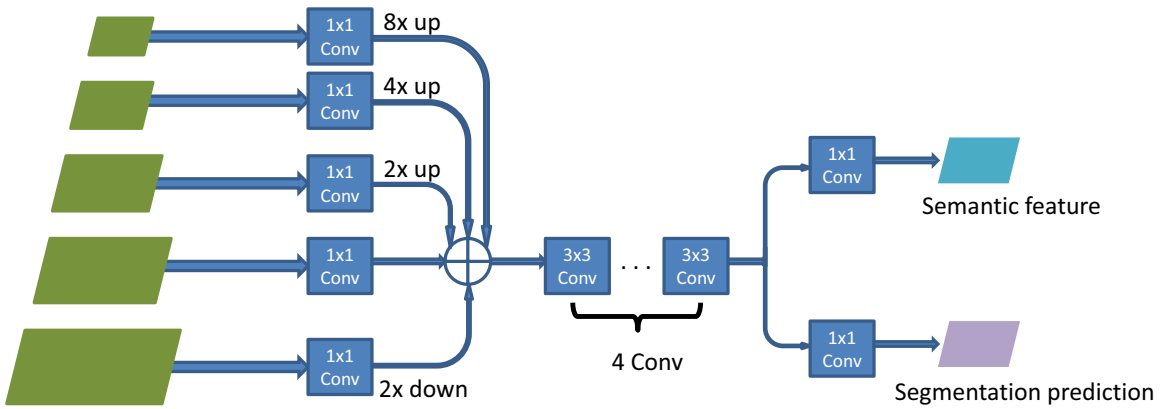


Fig. 10 Architecture of the semantic segmentation branch [81]

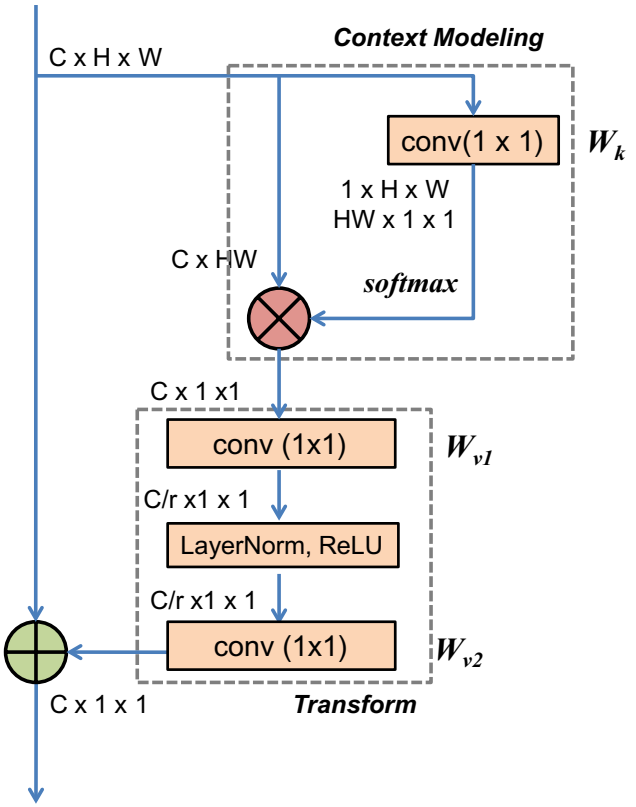


Fig. 11 Architecture of the global context block. feature maps have been shown as feature dimensions. For example, C×H×W denotes the feature map with channel number C, with height H and with width W, respectively. Matrix multiplication and element-wise sum are denoted by red circles and green circles, respectively. (The architecture draws inspiration from the non-local block given in Fig. 8, whose original work is discussed in Sect. 3.7.) '*r*' is bottleneck ratio, and *C*/*r* represents hidden representation dimension of the bottleneck. Default reduction ratio is set to $r=16$ [130]

context. The fact that it is lightweight allows the designers to apply it among multiple layers in the network, thus constructing a Global Context Network or GCNet. The GC block is shown in Fig. 11. It is formulated as:

$$z_i = \mathrm{x}_i + W_{v2}\mathrm{ReLU}\left(\mathrm{LN}\left(W_{v1}\sum_{j=1}^{N_p}\frac{e^{W_k x_j}}{\sum_{m=1}^{N_p} e^{W_k x_m}}x_j\right)\right) \quad (5)$$

where x_i = input for each query position i,W_k and W_v denote linear transformation matrices, $\alpha_j = \frac{e^{W_k x_j}}{\sum_m e^{W_k x_j}}$ is the weight for *global attention* (mentioned in Sect. 3.7) *pooling,* and $\delta(\cdot) = W_{v2}\mathrm{ReLU}\left(\mathrm{LN}\left(W_{v1}(\cdot)\right)\right)$ gives the bottleneck transform.

GCNet outperforms both Non-Local Networks and SENet on MS COCO.

3.11 Yolact

YOLACT [132] is a fast and simple instance segmentation model with fully convolutional topology. It is used for instance segmentation in real time and had the distinction of being the fastest real-time instance segmentation technique when it was introduced. It achieved a segmentation score of 29.8 Mask AP on the COCO dataset at 33 frames per second when one Titan XP GPU was used. This was faster than other state-of-the-art approaches at that time. The experimentation used a single GPU for training. This result was achieved by first bifurcation of image segmentation in parallel into 2 subtasks: (1) generation of prototype masks and (2) prediction of mask coefficients for each instance mask. Next, instance masks are produced by linear combination of the prototype masks with the coefficients of the masks. The authors also analysed the behaviour of emerging prototype masks and demonstrated that the network learnt localization of the object instances automatically with translational variance. This was in spite of the fact that the model was fully convolutional.

3.12 Mask scoring R-CNN

An important task in the context of image segmentation is allowing a deep neural network to be aware of its prediction quality. For the purpose of instance segmentation, the confidence estimate of classification of the instance has been used as quality score of the mask by a majority of instance segmentation approaches. In contradiction to this approach, the quality of the mask, as quantified as the Intersection over Union or IoU between the mask of the instance and the ground truth, has not been related properly to the classification score. In [133], the authors have studied this problem and subsequently proposed Mask Scoring R-CNN. The authors of this work note that without losing generality, they have worked on Mask R-CNN [67] and added an additional MaskIoU head module that learns the Mask-IoU aligned mask score. They claim that their technique is conceptually simple. Mask R-CNN has been combined with MaskIoU Head, which feeds on the instance features and predicted mask in combination. This arrangement is used to predict the IoU between the input mask and the ground truth mask. Figure 12 shows the Mask Scoring R-CNN architecture.

The model contains a network block which has the ability to learn predicted mask quality. This network block proposed by the authors combines the features of the instance with the predicted mask, in order to regress the predicted mask IoU. This mask scoring approach calculates the alignment error between quality of the mask and the score of the mask, thus improving the performance of the instance segmentation task by prioritizing better predictions of masks in COCO AP evaluation. Following extensive experimentation using

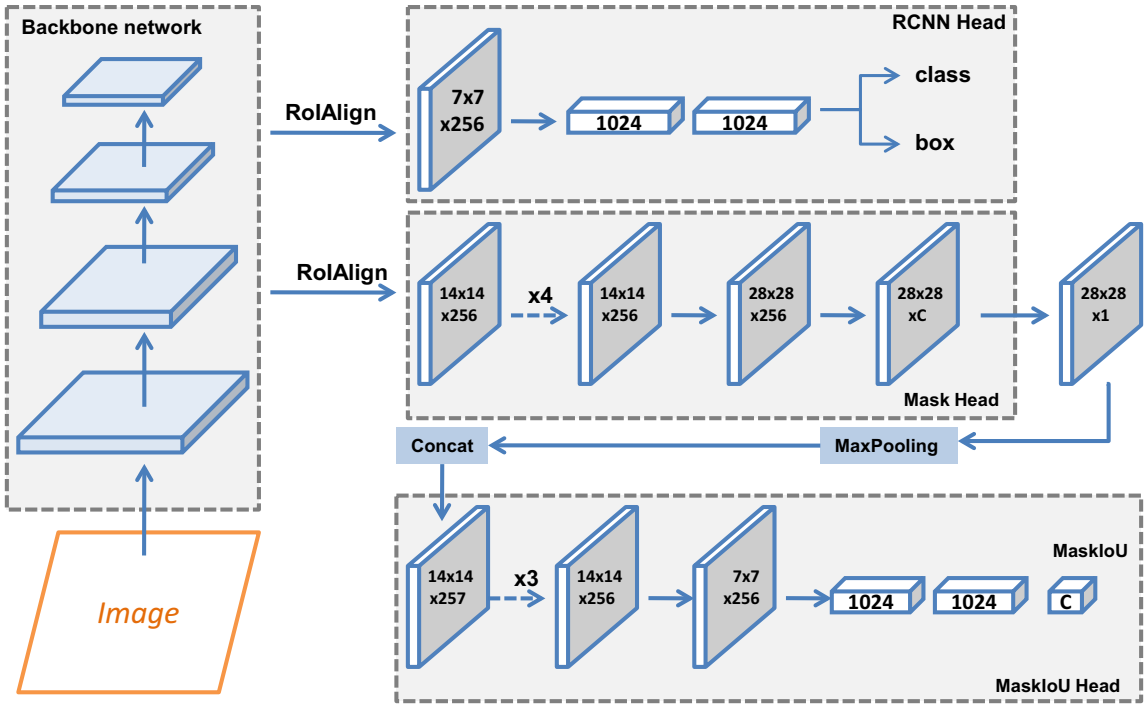


Fig. 12 Architecture of Mask Scoring R-CNN model. Image is fed to the backbone network to generate the ROIs via R.P.N. and the ROI features via ROIAlign. RCNN Head and Mask Head are components of Mask R-CNN model. For predicting MaskIoU, they use the predicted mask and ROI features as input. MaskIoU Head has four convolutional layers and has three fully connected layers. The last fully connected layer gives outputs for C classes MaskIoU [133]

the COCO dataset, the proposed approach consistently and noticeably improves various models. It even outperforms the efficient Mask R-CNN approach.

3.13 TensorMask

Object detection models using sliding window technique which generate predictions of bounding box with the help of a dense and regularly spaces grid have shown rapid advancement besides garnering significant popularity. In [91], the authors propose a model referred to as TensorMask, wherein they perform instance segmentation using dense sliding windows. This is an area which is relatively unexplored. The output for every pixel location tends to be a geometric structure having its own dimensions w.r.t. spatial aspects. In this work, a dense form of instance segmentation is performed by prediction over four-dimensional tensors. The geometry of the same is captured, and novel operators over four-dimensional tensors are used. The key concept of the TensorMask representation is using structured 4-D tensors for representation of masks over a spatial domain. This perspective is different from earlier work on the task of instance segmentation such as DeepMask [88] as well as InstanceFCN [90] which used unstructured 3-D tensors, wherein the segmentation mask has been packed in the third channel axis. Using a simple channel representation leads to loss of an opportunity to benefit from using structural arrays for representation of masks as 2-D entities. To overcome this problem, the authors of TensorMask propose to use 4-D tensors with shape (V, U, H, W), wherein both (H, W) for object position, and (V, U) for representing relative mask position, are subtensors. By virtue of the TensorMask framework, the authors have developed a pyramidal structure on top of a scale-indexed list of the 4-D tensors called a tensor bipyramid. This structure has a pyramid shape in both (H, W) and (V, U) sub-tensors; however, they grow in oppositely. They have combined these components into a network backbone and have followed the training procedure of RetinaNet [53] closely in which their dense mask predictors extend the original dense bounding box (BBox) predictors. The authors through their work on TensorMask stress on the importance of specifically capturing the geometric structure of this task. They show that TensorMask gives similar results to its Mask R-CNN counterpart. They claim that their results are promising and that the results suggest that the proposed framework/model can pave the way for future work on dense sliding window instance-based segmentation.

4 Datasets

A 2D image dataset contains gray-scale or RGB images. The information content is thus limited to spatial location of pixels and their intensity values. 2D image datasets are abundant for instance segmentation research. In the following section, we describe some of the most popular large 2D image datasets for instance segmentation.

4.1 Microsoft Common Objects in Context (COCO) dataset

The Microsoft Common Objects in Context or COCO dataset [64] is a large-scale image dataset for the purpose of recognizing, segmenting, and captioning images. It has many challenges. The detection challenge is most relevant for instance segmentation. The detection challenge features above 80 object classes while providing above 82,783 training images, 40,504 validation images, and above 80,000 testing images. The testing image set is divided amongst 4 subsets. These are: (1) test-dev with 20,000 images for the purpose of additional validation and debugging, (2) test-standard with 20,000 images which is the default testing image data subset for different competitions and for state-of-the-art benchmarking, (3) test-challenge with 20,000 images for evaluation server challenge submission, and (4) test-reserve with 20,000 images used to avoid possible challenge overfitting. The importance and popularity of the COCO dataset have increased substantially since it first appeared, mainly due to its large size. The challenge results are made available on a yearly basis in a combined workshop organized on lines of the European Conference on Computer Vision (ECCV) alongside those of ImageNet. Table 2 shows the notable benchmarking techniques on MS COCO dataset.

4.2 Cityscapes dataset

Cityscapes dataset [70] is a large collection of urban street scene images. It focuses on semantic understanding of the street scene. The dataset provides semantic, instance-specific, and pixel-specific annotations; 30 object classes have been grouped into 8 categories relevant to urban scenes like flat surfaces, vehicles, people, sky, etc. Cityscapes dataset consists of about 5000 images with fine annotation and 20,000 with coarse annotation. The images for the dataset were captured in fifty cities in a span of several months during daytime with good weather. First, it was video recorded. As such, the video-frames had to be hand selected for having aspects like high number of object classes, different scenes, and different backgrounds.

Table 2 Notable instance segmentation work on the microsoft COCO dataset

Method	Average precision (AP) (%)	Year
Hybrid Task Cascade [81] (with extra training data)	43.9	2019
PANet [79]	42.0	2018
SOLOv2 [134]	41.7	2020
GCNet [130]	41.5	2019
BlendMask [135]	41.3	2020
SOLO [136]	40.4	2019
Non-local Neural Networks [113]	40.3	2017
Mask Scoring R-CNN [133]	39.6	2019
CenterMask [137]	38.3	2019
MaskLab + [22]	38.1	2017
TensorMask [91]	37.3	2019
Mask R-CNN [67]	37.1	2017
PolarMask [138]	32.9	2019
YOLACT [132]	29.8	2019
MultiPath Network [72]	25.0	2016

4.3 The Mapillary Vistas Dataset (MVD)

The Mapillary Vistas Dataset (MVD) [71] is another large street scene image dataset. It contains 25,000 annotated images with 66 classes. The annotation has been done using a dense, fine-grained manual style with the help of polygonal delineation for individually demarcating different objects. The dataset is five times bigger than that of Cityscapes for fine annotation. It features images from across the world which have been captured during different weathers, seasons, and daytimes. The dataset images have been captured using different devices like cell phones, cameras, etc. with different photographers. The aim of developing the database is to further develop the state-of-the-art research in understanding street scenes.

5 Summary and discussion

In this section, we would like to discuss the key factors and issues which have emerged in instance segmentation based on deep learning [27].

5.1 Detection frameworks: two stage versus single stage

Using the number of stages as a means of classification of the framework, two major categories emerge for detection

frameworks, viz. region-based (2-stage) and unified framework (single stage):

- For platforms having rich computational resources, two-stage frameworks produce better accuracies than their single-stage counterparts. In fact, most winning techniques in famous challenges are usually two-stage frameworks. This is because their framework is flexible and more suitable for region-based detection, e.g. Mask RCNN [67].
- Single-stage detectors, e.g. YOLO [83], are usually faster than their two-stage counterparts due to lack of pre-processing, light backbone network, lesser number of candidate regions, and use of fully convolutional detection sub-network. However, single stage frameworks poorly detect small objects, which is not the case for two-stage frameworks.

Many attempts have been made to increase accuracy and efficiency of detectors leading to convergence towards some crucial design choices:

- Fully Convolutional Framework
- Exploration of complementary information from correlated tasks, e.g. in Mask R-CNN.
- Use of sliding windows
- Information fusing from various layers of the backbone

5.2 Backbone networks

Backbone networks are one of the important factors for performance due to their role as discriminators of object feature representations. Though deep backbones like ResNet [35], ResNeXt [73], etc. have been used successfully, these are computationally expensive.

5.3 Improvement in robustness of representation of objects

Various factors like object size, lighting, background, blur, resolution, noise, etc. contribute to challenges in object recognition. The important techniques used to handle these challenges are discussed below.

5.3.1 Object size/scale

Variation in scale of objects especially small ones, poses challenges. The main strategies used are mentioned below.

- Use of image pyramids: This simple and efficient technique helps to enlarge small objects and also shrinks larger ones. Though computational expensiveness is an issue, the use of such techniques is common for obtaining better accuracy.
- Use of features from different convolutional layers for different resolutions [53].
- Up-scaling to better resolution in the network, for detection of small objects [139].

5.3.2 Occlusion, deformation and other factors

There are techniques to handle transformation, occlusions, and deformation, e.g. by the use of a spatial transformer network. This technique uses regression to obtain a deformation area and then warps the features according to the deformation area [58]. Rotation invariance is important in real-world scenarios but is lesser attended to here due to the fact that popular benchmark segmentation-based datasets (like MS COCO) do not have large variations in rotation. Handling occlusion is well researched in other areas; however, it is lesser attended to in the current area.

5.4 Detection proposals

Detection proposals significantly reduce the search space for instance segmentation candidates. After the success of RPN [42], which was able to integrate generation of proposals and detection into a single framework, CNN-based detection proposal frameworks have dominated region proposal.

5.5 Strengths and weaknesses with various instance segmentation techniques

Table 3 lists the strengths and weaknesses with various techniques discussed earlier (in Sect. 2).

6 Scope for future work

Instance segmentation remains a challenging task. For example, on the popular MS COCO dataset, the overall average precision is around 50%, leaving plenty of room for improvement. Even as of now, researchers are occupied with the *hardware required v/s algorithmic simplicity,* as well as *speed v/s accuracy* tradeoffs, respectively. As an example, in [113] they train the model on an 8-GPU machine where each GPU has 8 clips in a mini-batch (in total a mini-batch size of 64 clips). The models are trained for 400 k iterations in total, starting with a learning rate of 0.01 and reducing it 10 times for every 150 k iterations. Hardware constraints limit the scope of research. As can be seen, tasks like instance segmentation are computationally expensive. In spite of this, real-time instance segmentation (speed optimization) remains an issue. This has potential applications in

Table 3 Strengths and weaknesses of groups (of techniques) mentioned in Sect. 2

Group	Strengths	Weaknesses
Classification of mask proposals	Relatively simple to implement Modest segmentation accuracy	Slow and difficult to optimize training Storage, time, and detection scale issues during training Slow testing Not suited for real-time applications
Detection followed by segmentation	Relatively simple to train Better generalization Relatively faster (e.g. YOLACT) Good segmentation accuracy	Depend on a complicated training pipeline which is difficult to train, and to optimize
Labelling pixels followed by clustering	Use some recently investigated techniques Relatively simpler techniques	Lesser segmentation accuracy Intense computation necessitates high computational power Not suited for real-time applications
Dense sliding window methods	Relatively unexplored area Modest segmentation accuracy	Use complex algorithms Difficult to train and optimize Not suited for real time applications

intelligent systems like autonomous vehicle systems, security applications, biometrics, etc.

At the model design level, efficient management of the feature *flood* (due to complex architectures) at large, and self-automation of fine-grained convolution metrics like stride for better results, is still issues. Small object detection remains quite challenging. End-to-end-based systems design and their training remain issues.

Body part detection research had also generated interest. Datasets on human pose estimation and human parsing (MHPv1.0 [140], MHPv2.0 [141], and Pascal Person Part Database [142]) have been made available, preliminary results [67, 140, 141, 143] are available, and new discoveries are around the corner.

7 Conclusion

In this paper, an overview of instance segmentation is given. The evolution of image segmentation is from coarse to fine inference. This evolution has come up to instance segmentation and is continuing further, with advancements in computing power and research prowess. In this paper, important instance segmentation issues have been discussed. Various techniques used for instance segmentation have been discussed, from both holistic and individual perspectives. Their taxonomy, strengths, and weaknesses have been discussed. The popular datasets used for instance segmentation have been discussed. Major issues which open the scope for future research have been discussed. The survey is an attempt to provide information about the state of the art in the field of instance segmentation, with regard to its purpose, emergence, techniques and related work, datasets, and scope for future work.

References

1. Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J (2018) A survey on deep learning techniques for image and video semantic segmentation. Appl Soft Comput 70:41–65. https://doi.org/10.1016/j.asoc.2018.05.018
2. Tang Y (2013) Deep learning using linear support vector machines. arXiv preprint arXiv:13060239
3. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
4. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
5. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
6. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26. https://doi.org/10.1016/j.neucom.2016.12.038
7. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651. https://doi.org/10.1109/TPAMI.2016.2572683
8. Kirsch RA, Cahn L, Ray C, Urban GH (1957) Experiments in processing pictorial information with a digital computer. In: Eastern joint computer conference, pp 221–229
9. Earnest LD (1963) Machine reading of cursive script. In: IFIP congress, Amsterdam. pp 462–466
10. Moore GA (1968) Automatic scanning and computer processes for the quantitative analysis of micrographs and equivalent subjects. In: Cheng GC (ed) Pictorial Pattern Recognition. Thompson, Washington DC, pp 275–326
11. Rumelhart DE, Hinton GE, McClelland JL (1986) A general framework for parallel distributed processing. Parallel distributed processing. Explor Microstruct Cogn 1:45–76
12. Rabiner L (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286. https://doi.org/10.1109/5.18626
13. Nouboud F, Plamondon R (1990) On-line recognition of handprinted characters: survey and beta tests. Pattern Recogn 23(9):1031–1044. https://doi.org/10.1016/0031-3203(90)90111-W

14. Mori S, Suen CY, Yamamoto K (1992) Historical review of OCR research and development. Proc IEEE 80(7):1029–1058. https://doi.org/10.1109/5.156468
15. Bunke H, Wang PS-P (1994) HandBook of Character Recognition and Document Image Analysis. World Scientific, Singapore
16. Cortes C, Vapnik V (1995) Support vector machine. Mach Learn 20(3):273–297
17. O'Gorman L, Kasturi R (1995) Document Image Analysis. IEEE Computer Society Press, New York
18. Tang YY, Lee S-W, Suen CY (1996) Automatic document processing: a survey. Pattern Recogn 29(12):1931–1952. https://doi.org/10.1016/S0031-3203(96)00044-1
19. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp 2278–2324
20. Nagy G (2000) Twenty years of document image analysis in PAMI. IEEE Trans Pattern Anal Mach Intell 22(1):38–62. https://doi.org/10.1109/34.824820
21. Ahmed P, Al-Ohali Y (2000) Arabic character recognition: progress and challenges. J King Saud Univ Comput Inf Sci 12:85–116. https://doi.org/10.1016/S1319-1578(00)80004-X
22. Chen L, Hermans A, Papandreou G, Schroff F, Wang P, Adam H (2018) MaskLab: instance segmentation by refining object detection with semantic and direction features. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, 18–23 June 2018, pp 4013–4022. https://doi.org/10.1109/cvpr.2018.00422
23. Dickinson SJ, Leonardis A, Schiele B, Tarr MJ (2009) Object categorization: computer and human vision perspectives. Cambridge University Press, Cambridge
24. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
25. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: ICCV
26. Zhu X, Vondrick C, Fowlkes CC, Ramanan D (2016) Do we need more training data? Int J Comput Vis 119(1):76–92
27. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. Int J Comput Vis 128(2):261–318. https://doi.org/10.1007/s11263-019-01247-4
28. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. IEEE, pp 1150–1157
29. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005. IEEE, pp 886–893
30. Sivic (2003) Zisserman Video Google: a text retrieval approach to object matching in videos. In: Proceedings ninth IEEE international conference on computer vision, 13–16 Oct 2003, vol 1472, pp 1470–1477. https://doi.org/10.1109/iccv.2003.1238663
31. Perronnin F, Sánchez J, Mensink T (2010) Improving the fisher kernel for large-scale image classification. In: European conference on computer vision. Springer, pp 143–156
32. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828. https://doi.org/10.1109/TPAMI.2013.50
33. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp 1097–1105
34. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: ICLR
35. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
36. Huang G, Liu Z, Maaten Lvd, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 2261–2269. https://doi.org/10.1109/cvpr.2017.243
37. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
38. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2014) Overfeat: integrated recognition, localization and detection using convolutional networks. In: ICLR
39. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, pp 818–833
40. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariance shift. In: ICML, pp 448–456
41. Girshick R (2015) Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV), 7–13 Dec 2015, pp 1440–1448. https://doi.org/10.1109/iccv.2015.169
42. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
43. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
44. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645. https://doi.org/10.1109/TPAMI.2009.167
45. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824
46. Hariharan B, Arbeláez P, Girshick R, Malik J (2017) Object instance segmentation and fine-grained localization using hypercolumns. IEEE Trans Pattern Anal Mach Intell 39(4):627–639. https://doi.org/10.1109/TPAMI.2016.2578328
47. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR
48. Shrivastava A, Sukthankar R, Malik J, Gupta A (2017) Beyond skip connections: top-down modulation for object detection. In: CVPR. arXiv:1612.06851
49. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:151107122
50. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: NIPS, pp 379–387
51. Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848. https://doi.org/10.1109/TPAMI.2017.2699184
52. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2018) Graph neural networks: a review of methods and applications. arXiv preprint arXiv:181208434
53. Lin T, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 936–944. https://doi.org/10.1109/cvpr.2017.106

54. Kong T, Sun F, Yao A, Liu H, Lu M, Chen Y (2017) RON: reverse connection with objectness prior networks for object detection. In: CVPR, pp 5936–5944
55. Lenc K, Vedaldi (2015) A understanding image representations by measuring their equivariance and equivalence. In: CVPR, pp 991–999
56. Liu L, Fieguth P, Guo Y, Wang X, Pietikäinen M (2017) Local binary features for texture classification: taxonomy and experimental study. Pattern Recogn 62:135–160. https://doi.org/10.1016/j.patcog.2016.08.032
57. Chellappa R (2016) The changing fortunes of pattern recognition and computer vision. Image Vis Comput 55:3–5. https://doi.org/10.1016/j.imavis.2016.04.005
58. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: ICCV
59. Mordan T, Thome N, Henaff G, Cord M (2019) End-to-end learning of latent deformable part-based representations for object detection. Int J Comput Vis 127(11):1659–1679. https://doi.org/10.1007/s11263-018-1109-z
60. Ouyang W, Wang X (2013) Joint deep learning for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision, pp 2056–2063
61. Wang X, Shrivastava A, Gupta A (2017) A-fast-RCNN: hard positive generation via adversary for object detection. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 3039–3048. https://doi.org/10.1109/cvpr.2017.324
62. Zhang S, Yang J, Schiele B (2018) Occluded pedestrian detection through guided attention in CNNs. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, 18–23 June 2018, pp 6995–7003. https://doi.org/10.1109/cvpr.2018.00731
63. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
64. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick L (2014) Microsoft COCO: common objects in context
65. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2009) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88:303–308
66. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
67. He K, Gkioxari G, Dollar P, Girshick R (2018) Mask R-CNN. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/tpami.2018.2844175
68. Li Y, Qi H, Dai J, Ji X, Wei Y (2017) Fully convolutional instance-aware semantic segmentation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 4438–4446. https://doi.org/10.1109/cvpr.2017.472
69. Bai M, Urtasun R (2017) Deep watershed transform for instance segmentation. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 2858–2866. https://doi.org/10.1109/cvpr.2017.305
70. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), 27–30 June 2016, pp 3213–3223. https://doi.org/10.1109/cvpr.2016.350
71. Neuhold G, Ollmann T, Bulò SR, Kontschieder P (2017) The Mapillary vistas dataset for semantic understanding of street scenes. In: 2017 IEEE international conference on computer vision (ICCV), 22–29 Oct 2017, pp 5000–5009. https://doi.org/10.1109/iccv.2017.534
72. Zagoruyko S, Lerer A, Lin T-Y, Pinheiro PO, Gross S, Chintala S, Dollár P (2016) A multipath network for object detection. arXiv preprint arXiv:160402135
73. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 5987–5995. https://doi.org/10.1109/cvpr.2017.634
74. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks. In: Advances in neural information processing systems, pp 4467–4475
75. Hariharan B, Arbeláez P, Girshick R, Malik J (2014) Simultaneous detection and segmentation. In: European conference on computer vision. Springer, pp 297–312
76. Sande KEAVD, Uijlings JRR, Gevers T, Smeulders AWM (2011) Segmentation as selective search for object recognition. In: 2011 international conference on computer vision, 6–13 Nov 2011, pp 1879–1886. https://doi.org/10.1109/iccv.2011.6126456
77. Arbeláez P, Pont-Tuset J, Barron J, Marques F, Malik J (2014) Multiscale combinatorial grouping. In: 2014 IEEE conference on computer vision and pattern recognition, 23–28 June 2014, pp 328–335. https://doi.org/10.1109/cvpr.2014.49
78. Dai J, He K, Sun J (2016) Instance-aware semantic segmentation via multi-task network cascades. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), 27–30 June 2016, pp 3150–3158. https://doi.org/10.1109/cvpr.2016.343
79. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759–8768
80. Peng C, Xiao T, Li Z, Jiang Y, Zhang X, Jia K, Yu G, Sun J (2018) MegDet: a large mini-batch object detector. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, 18–23 June 2018, pp 6181–6189. https://doi.org/10.1109/cvpr.2018.00647
81. Chen K, Pang J, Wang J, Xiong Y, Li X, Sun S, Feng W, Liu Z, Shi J, Ouyang W (2019) Hybrid task cascade for instance segmentation. arXiv preprint arXiv:190107518
82. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
83. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 6517–6525. https://doi.org/10.1109/cvpr.2017.690
84. Lin T, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: 2017 IEEE international conference on computer vision (ICCV), 22–29 Oct 2017, pp 2999–3007. https://doi.org/10.1109/iccv.2017.324
85. Kirillov A, Levinkov E, Andres B, Savchynskyy B, Rother C (2017) InstanceCut: from edges to instances with multicut. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 7322–7331. https://doi.org/10.1109/cvpr.2017.774
86. Arnab A, Torr PHS (2017) Pixelwise instance segmentation with a dynamically instantiated network. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), 21–26 July 2017, pp 879–888. https://doi.org/10.1109/cvpr.2017.100
87. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:14127062
88. Pinheiro PO, Collobert R, Dollar P (2015) Learning to segment object candidates 1990–1998
89. Pinheiro PO, Lin T-Y, Collobert R, Dollár P (2016) Learning to refine object segments. In: European conference on computer vision, 2016. Springer, pp 75–91

90. Dai J, He K, Li Y, Ren S, Sun J (2016) Instance-sensitive fully convolutional networks. In: European conference on computer vision. Springer, pp 534–549
91. Chen X, Girshick R, He K, Dollár P (2019) TensorMask: a foundation for dense object segmentation. arXiv preprint arXiv:190312174
92. Hariharan B, Arbelaez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: CVPR
93. Bell S, Zitnick CL, Bala K, Girshick RB (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: CVPR
94. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2015) Object detectors emerge in deep scene CNNs. In: ICLR
95. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
96. Uhrig J, Cordts M, Franke U, Brox T (2016) Pixel-level encoding and depth layering for instance-level semantic labeling. arXiv:1604.05096
97. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI
98. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV
99. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell G (2017) Understanding convolution for semantic segmentation. arXiv:1702.08502
100. Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587
101. Abadi M, Agarwal A (2016) Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467
102. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3-d transform-domain collaborative filtering. Trans Image Process (TIP) 16:2080–2095
103. Burger HC, Schuler CJ, Harmeling S (2012) Image denoising: can plain neural networks compete with BM3D? In: Computer vision and pattern recognition (CVPR)
104. Burger HC, Schuler CJ, Harmeling S (2012) Image denoising with multi-layer perceptrons, part 2: training trade-offs and analysis of their mechanisms. arXiv:1211.1552
105. Lefkimmiatis S (2017) Non-local color image denoising with convolutional neural networks. In: Computer vision and pattern recognition (CVPR)
106. Lafferty J, McCallum A, Pereira FC (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: International conference on machine learning (ICML)
107. Krahenbuhl P, Koltun V (2011) Efficient inference in fully connected crfs with gaussian edge potentials. In: Neural information processing systems (NIPS)
108. Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C, Torr PH (2015) Conditional random fields as recurrent neural networks. In: International conference on computer vision (ICCV)
109. Schwing AG, Urtasun R (2015) Fully connected deep structured networks. arXiv:1503.02351
110. Chandra S, Usunier N, Kokkinos I (2017) Dense and low-rank Gaussian CRFs using deep embeddings. In: International conference on computer vision (ICCV)
111. Harley A, Derpanis K, Kokkinos I (2017) Segmentation-aware convolutional networks using local attention masks. In: International conference on computer vision (ICCV)
112. Liu S, Mello SD, Gu J, Zhong G, Yang MH, Kautz J (2017) Learning affinity via spatial propagation networks. In: Neural information processing systems (NIPS)
113. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, 18–23 June 2018, pp 7794–7803. https://doi.org/10.1109/cvpr.2018.00813
114. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80
115. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Neural information processing systems (NIPS)
116. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. In: Computer vision and pattern recognition (CVPR)
117. Efros AA, Leung TK (1999) Texture synthesis by nonparametric sampling. In: International conference on computer vision (ICCV)
118. Peng C, Zhang X, Yu G, Luo G, Sun J (2017) Large kernel matters—improve semantic segmentation by global convolutional network. In: CVPR
119. Ghiasi G, Fowlkes CC (2016) Laplacian reconstruction and refinement for semantic segmentation. In: ECCV
120. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, 2015. Springer, pp 234–241
121. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: ICCV
122. Fu C, Liu W, Ranga A, Tyagi A, Berg AC (2017) DSSD: deconvolutional single shot detector. arXiv:1701.06659
123. Cai Z, Fan Q, Feris RS, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV
124. Zagoruyko S, Lerer A, Lin T, Pinheiro PHO, Gross S, Chintala S, Dollar P (2016) A multipath network for object detection. In: BMVC
125. Kong T, Yao A, Chen Y, Sun F (2016) Hypernet: towards accurate region proposal generation and joint object detection. In: CVPR
126. Ren S, He K, Girshick RB, Zhang X, Sun J (2017) Object detection networks on convolutional feature maps. PAMI
127. Zeng X, Ouyang W, Yan J, Li H, Xiao T, Wang K, Liu Y, Zhou Y, Yang B, Wang Z, Zhou H, Wang X (2016) Crafting GBD-net for object detection. arXiv:1610.02579
128. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: CVPR
129. Liu W, Rabinovich A, Berg AC (2015) Parsenet: looking wider to see better. arXiv:1506.04579
130. Cao Y, Xu J, Lin S, Wei F, Hu H (2019) GCNet: non-local networks meet squeeze-excitation networks and beyond. arXiv:1904.11492v1
131. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: IEEE conference on computer vision and pattern recognition
132. Bolya D, Zhou C, Xiao F, Lee YJ (2019) YOLACT: real-time instance segmentation. arXiv preprint arXiv:190402689
133. Huang Z, Huang L, Gong Y, Huang C, Wang X (2019) Mask scoring R-CNN. arXiv e-prints
134. Wang X, Zhang R, Kong T, Li L, Shen C (2020) SOLOv2: dynamic, faster and stronger. arXiv preprint arXiv:200310152

135. Chen H, Sun K, Tian Z, Shen C, Huang Y, Yan Y (2020) BlendMask: top-down meets bottom-up for instance segmentation. arXiv preprint arXiv:200100309
136. Wang X, Kong T, Shen C, Jiang Y, Li L (2019) SOLO: segmenting objects by locations. arXiv preprint arXiv:191204488
137. Lee Y, Park J (15 Nov 2019) CenterMask: real-time anchor-free instance segmentation. arXiv:1911.06667v1
138. Xie E, Sun P, Song X, Wang W, Liu X, Liang D, Shen C, Luo P (2019) PolarMask: single shot instance segmentation with polar representation. arXiv:1909.13226v2
139. Sun K, Xiao B, Liu D, Wang J (2019) Deep high resolution representation learning for hman pose estimation. In: CVPR
140. Li J, Zhao J, Wei Y, Lang C, Li Y, Sim T, Yan S, Feng J (2017) Multi-human parsing in the wild. arXiv:1705.07206
141. Zhao J, Li J, Cheng Y, Zhou L, Sim T, Yan S, Feng J (2018) Understanding humans in crowded scenes: deep nested adversarial learning and a new benchmark for multi-human parsing. arXiv:1804.03287v3
142. Chen X, Mottaghi R, Liu X, Fidler S, Urtasun R, Yuille A (2014) Detect what you can: detecting and representing objects using holistic models and body parts. In: CVPR, pp 1971–1978
143. Brabandere BD, Neven D, Gool LV (2017) Semantic instance segmentation with a discriminative loss function. arXiv:1708.02551v1