Proceedings of CONF-CDS 2025 Symposium: Application of Machine Learning in Engineering
DOI: 10.54254/2755-2721/2025.AST26037

Introduction to Target Instance Segmentation Method Based on Deep Learning

Peisen Liu

School of Information Science and Engineering, Lanzhou University, Lanzhou, China
2028162790@qq.com

Abstract: With the rapid development of high-tech technologies in the 21st century, society is increasingly moving toward intelligentization. Intelligentization refers to endowing machines with human-like capabilities, and among these, the foremost is the ability to perceive the external environment—what is known as computer vision. Possessing perceptual capabilities means being able to distinguish and categorize elements within an environment, identifying those with similar attributes while differentiating those with distinct attributes. This involves organizing objects in the environment into cohesive units under unified concepts, enabling recognition of diverse entities. This process requires target instance segmentation methods based on deep learning. This paper primarily summarizes and analyzes early and mainstream approaches to target instance segmentation using deep learning, thoroughly examining existing research to guide future work in this field. Through analysis, this paper finds that diverse innovative strategies, such as parallel mask assembly (YOLACT), instance classification via spatial grids (SOLO), polar coordinate representation (PolarMask), and unified mask classification (MaskFormer), can effectively combine instance distinction with pixel-level classification.

Keywords: Target Instance Segmentation, Convolution, Two-Stage, Single-Stage, Transformer

1. Introduction

In recent years, research into areas such as autonomous driving, robot vision, virtual reality, and smart transportation has been steadily advancing. Commonly used technologies such as object detection and semantic segmentation have gradually faded from view due to their inherent limitations—object detection cannot provide precise shape information, and semantic segmentation cannot distinguish between different individuals of the same category. In this context, driven by the need to enhance accuracy, improve efficiency, and strengthen robustness and generalization capabilities, achieving pixel-level instance segmentation has become essential. This is why we must continuously explore and optimize instance segmentation techniques. Instance segmentation addresses both semantic segmentation (pixel-level classification) and instance separation (distinguishing different objects), making it highly suitable for handling complex scenarios involving occlusions or overlapping objects of the same category. In recent years, driven by the continuous development and breakthroughs in deep learning technology, instance segmentation

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.AST26037

methods have undergone multiple iterations and updates: from early two-stage paradigms to efficient single-stage frameworks, and then to Transformer-based models. This paper will systematically introduce the various methods established based on these concepts. By systematically organizing these typical methods, it provides researchers entering this field with a comprehensive knowledge framework, facilitating further in-depth development and innovation.

2. Basic methods

2.1. Fully Convolutional Networks

Let's first take a look at Convolutional Neural Networks (CNN). Convolution, as the name suggests, involves the accumulation of multiple products. At its core is a set of data windows with predefined weights, known as convolution kernels. Following a specified stride, the input layer is continuously divided into blocks of the same size as the convolution kernel. The parameters within each block are multiplied by the corresponding weights of the convolution kernel and then summed until all input parameters have been processed. This method effectively captures local features of an image without being influenced by their position.

Fully Convolutional Networks (FCN) architecture is based on CNN, modifying the fully connected layers of CNN by replacing them with 1×1 convolutional layers, resulting in a purely convolutional structure. This change effectively addresses the limitation of traditional classification networks (such as AlexNet and VGG), which can only accept fixed-size inputs, enabling them to accept inputs of any size and support end-to-end dense prediction. In previous methods, downsampling was commonly used, but downsampling has limitations, namely, low output resolution. To address this issue, FCN employs transposed convolutional layers for upsampling, which can relatively well restore spatial resolution. When observing an external object, people typically first seek to understand two questions: what it is and where it is. These two questions correspond to semantic information and detail information, respectively. When extracting features, semantic information is often reflected in deep features, while detail information is often reflected in shallow features. To address both questions simultaneously, FCN uses skip connections to fuse deep and shallow features before outputting the results. The integrated design of "fully convolutional + upsampling + skip connections" laid the foundation for subsequent developments in the field of semantic segmentation, and many of the current state-of-the-art models are optimized based on this framework.

The FCN's network architecture can be simply summarized as a single encoder for feature extraction (e.g., VGG) followed by a decoder for upsampling and restoring resolution. It takes an entire image (of any size) as input and outputs a semantic label map with the same resolution as the input [1].

2.2. DeepMask

Unlike FCN, which performs semantic segmentation tasks with pixel-level output granularity, DeepMask generates instance-aware candidate regions with instance-level output granularity. At the time this method was proposed, the mainstream object detection framework was a two-stage process such as R-CNN: the first stage was to generate candidate regions for objects, and the second stage was to classify the candidate regions. It is clear that metrics such as recall rate, accuracy, and quantity of candidate regions are critical to the performance of detection throughout the entire process. To better ensure the quality of candidate regions, DeepMask attempts to bypass the

DOI: 10.54254/2755-2721/2025.AST26037

underlying segmentation step, allowing the deep network to directly learn and generate segmentation masks from raw pixels. DeepMask adopts a dual-branch design, where the mask branch generates class-independent candidate object masks, and the scoring branch predicts the probability of object existence. The key innovation in its network architecture lies in replacing the large fully connected output layer with two fully connected layers (without intermediate nonlinearities), which is referred to by the academic community as the low-rank decomposition layer. The low-rank decomposition layer effectively reduces parameters while enabling each pixel classifier to detect the entire feature map.

As the first end-to-end learning segmentation candidate, DeepMask's key features can be briefly summarized as: joint learning, low-rank decomposition, and full-image inference. It typically takes image patches as input and outputs a set of corresponding candidate object masks and their corresponding confidence scores [2].

3. Mainstream methods

3.1. Two-stage method

The main workflow of a two-stage object detection algorithm consists of two steps: first, generating high-quality candidate boxes, and then fine-tuning the boxes. Object positive and negative labels are defined using an Intersection over Union (IoU) threshold. Regions with CNN features (R-CNN), as a classic two-stage object detection algorithm, processes input images through four main steps: a. Region proposal generation; b. CNN feature extraction; c. Object classification; d. Bounding box regression. However, R-CNN has some limitations, with three core issues: a. Increasing the threshold leads to an exponential decrease in positive samples, causing overfitting; b. A single detection head cannot accommodate multiple quality proposals; c. Repeated optimization by a single regressor harms high IoU boxes. Cascade R-CNN, as a major improvement in the CNN series, provides solutions to these issues one by one. The improvements in Cascade R-CNN primarily involve replacing the single detection head of R-CNN with a sequence of specialized detection heads and gradually increasing the fixed IoU threshold of R-CNN. The core idea of Cascade R-CNN can be summarized as: cascaded gradual optimization and hypothesis box resampling. The resampling mechanism (where the output of the previous stage serves as the input for the next stage) addresses the issue of insufficient high IoU samples; quality matching (where higher-level detection heads only process high-quality proposals) resolves the mismatch between detector quality and proposal quality; and cascaded specialized regressors (where each regressor is optimized for the current distribution) tackle the degradation of localization accuracy. Cascade R-CNN demonstrates significant advantages in high-precision detection, making it suitable for scenarios with dense small objects and tasks sensitive to false positives (such as security surveillance) [3].

3.2. Single-stage method

3.2.1. YOLACT

YOLACT is a classic single-stage real-time instance segmentation model that addresses the issue of slow speed in two-stage algorithms, achieving both high efficiency and high accuracy. Its core innovation lies in parallelized mask generation, which decomposes instance segmentation into two parallel sub-tasks: a. Using an FCN to generate k category-independent prototype masks covering the entire image. b. Adding a branch to the detection head to predict k coefficients for each anchor.

DOI: 10.54254/2755-2721/2025.AST26037

Subsequently, mask assembly is performed via a single matrix multiplication. YOLACT employs the innovative Fast NMS component to replace the serial processing of traditional NMS, specifically implemented as follows: a. Calculate the IoU matrix for the top n detection boxes of each class; b. Mask the lower triangle of the matrix; c. Take the maximum IoU value for each column, and suppress the box if it exceeds the threshold [4].

3.2.2. SOLO

Traditional implementation methods for instance segmentation can be divided into two categories: a. Top-down: first detect bounding boxes, then perform segmentation within the boxes; b. Bottom-up: learn pixel embedding vectors and cluster them into groups. The former relies on detection accuracy, while the latter relies on post-processing and has low accuracy. SOLO innovates on this by introducing the concept of instance categories, transforming instance segmentation into a location-aware classification task. The innovations of SOLO are primarily reflected in the following aspects: spatial grid partitioning; replacing traditional convolutions with spatial-sensitive convolutions (CoordConv); using multi-scale processing (FPN) for size-based classification; and adopting a dual-branch output (including a semantic category branch and an instance mask branch). The synergistic use of these innovations enables SOLO to achieve both high performance and flexibility [5].

3.2.3. PolarMask

PolarMask is a single-stage, anchor-free instance segmentation method that simplifies the entire instance segmentation process into two main parts: a. instance center classification, which aims to determine whether a pixel belongs to the object center; b. dense distance regression in polar coordinates, which aims to predict the ray length from the center to the contour. Its most critical innovation lies in the polar coordinate mask representation, specifically: using the object's center of mass as the origin, n rays are emitted at uniform angles, and the length of each ray is predicted. Center sample sampling is employed to calculate the polar coordinate centroid and polar coordinate IoU loss. The model's network architecture is as follows: a. The backbone uses ResNet/ResNeXt and FPN. b. It has two task heads: one is the classification branch, responsible for predicting the category and polar coordinate centroid; the other is the regression branch, responsible for predicting the lengths of the n rays. c. The post-processing workflow primarily involves filtering center points based on a score threshold, assembling the contour using ray lengths, and finally performing Non-maximum suppression (NMS) based on the mask's minimum bounding box [6].

3.3. Transformer-based method

3.3.1. DETR

DETR is an end-to-end object detection framework that replaces the heuristic components relied upon in traditional methods with a Transformer architecture, treating object detection as a direct set prediction problem and directly outputting an unordered set of object predictions (bounding boxes + categories) without the need for post-processing. This change optimizes the issue in traditional methods where they rely on proxy tasks, introducing a large amount of prior knowledge and hyperparameters to predict bounding boxes and category labels. In addition to set prediction, another core innovation of DETR is the bipartite graph matching loss: it uses the Hungarian algorithm to match predictions with ground truth, assigning a unique predicted bounding box to each ground truth bounding box, resulting in the Hungarian loss. These two innovations successfully eliminate

DOI: 10.54254/2755-2721/2025.AST26037

the need for manually designed components, making the method highly suitable for large object detection and panoramic segmentation [7].

3.3.2. MaskFormer

As mentioned earlier, in traditional methods, semantic segmentation uses pixel-wise classification, predicting a category label for each pixel. Instance segmentation uses mask classification, with each mask associated with a global category label. This distinction poses a problem: although the two tasks are fundamentally similar, they use different models, loss functions, and training processes, hindering the development of a unified framework. To address this issue, MaskFormer treats all segmentation tasks as predicting a set of paired (category labels, binary masks). MaskFormer consists of three core modules: the pixel-level module is responsible for extracting low-resolution feature maps and generating high-resolution pixels; the Transformer module is responsible for outputting N segment embedding vectors, each encoding the global information of a predicted segment. The segmentation module is responsible for category prediction and mask prediction. Its simple architecture design (Transformer decoder + dot product mask generation using shared pixel embeddings) and training loss (single L_mask-cls) make it a landmark work in the field of segmentation [8].

4. Conclusion

This paper provides a comprehensive analysis and summary of deep learning-based object instance segmentation methods, tracing their evolution from foundational techniques (such as FCN and DeepMask) to contemporary mainstream approaches. These include two-stage methods (e.g., Cascade R-CNN), single-stage methods (e.g., YOLACT, SOLO, PolarMask), and Transformer-based paradigms (e.g., DETR, MaskFormer). The analysis yields the following conclusions: first, the field has made significant progress from computationally intensive multi-stage processes to more streamlined and efficient architectures, achieving significantly improved inference speeds without necessarily sacrificing accuracy; Second, through diverse innovative strategies such as parallel mask assembly (YOLACT), instance classification via spatial grids (SOLO), polar coordinate representation (PolarMask), and unified mask classification (MaskFormer), it is possible to effectively combine instance segmentation with pixel-level classification; Additionally, the emergence of Transformer architectures has enabled true end-to-end learning, eliminating the need for many manually designed components (such as NMS and anchor boxes), and providing a more unified framework for segmentation tasks. However, the scope of this study also has limitations: it does not delve into the application of emerging technologies such as few-shot learning or self-supervised pre-training in instance segmentation, nor does it address dynamic convolutions, lightweight methods, or contour-based approaches. Future research recommendations focus on the following promising directions: 1) Developing more computationally efficient models, particularly lightweight Transformers or neural architecture search techniques, to enable real-time deployment on edge devices; 2) Enhancing the robustness of models for segmentation in challenging scenarios, such as small targets, occluded targets, or irregularly shaped targets; 3) Reducing the heavy reliance on large-scale pixel-level annotated data through advanced semi-supervised, weakly supervised, or synthetic data generation methods.

DOI: 10.54254/2755-2721/2025.AST26037

References

[1] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[2] O Pinheiro, Pedro O., Ronan Collobert, and Piotr Dollár. "Learning to segment object candidates." Advances in neural information processing systems 28 (2015).
[3] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[4] Bolya, Daniel, et al. "Yolact: Real-time instance segmentation." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
[5] Wang, Xinlong, et al. "Solo: Segmenting objects by locations." European conference on computer vision. Cham: Springer International Publishing, 2020.
[6] Xie, Enze, et al. "Polarmask: Single shot instance segmentation with polar representation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[7] Carion, Nicolas, et al. "End-to-end object detection with transformers." European conference on computer vision. Cham: Springer International Publishing, 2020.
[8] Cheng, Bowen, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation." Advances in neural information processing systems 34 (2021): 17864-17875.