Received: 17 November 2022 | Revised: 9 March 2023 | Accepted: 18 April 2023

DOI: 10.1049/cvi2.12202

IET Computer Vision

ORIGINAL RESEARCH

An encoder-decoder framework with dynamic convolution for weakly supervised instance segmentation

Liangjun Zhu[1] | Li Peng[1] | Shuchen Ding[2] | Zhongren Liu[1]

[1]Engineering Research Center of Internet of Things Applied Technology, Jiangnan University, Wuxi, China

[2]School of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou, China

Correspondence

Li Peng, Jiangnan University, Lihu Campus, No. 1800, Lihu Avenue, Binhu District, Wuxi, Jiangsu, China.
Email: jnpengli@outlook.com

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 61873112, 52005124

Abstract

In the systems of industrial robotics and autonomous vehicles, instance segmentation is widely employed. However, manually labelling an object outline is time-consuming. In order to reduce annotation costs, we present a weakly supervised instance segmentation method in this article. A deeply convolutional network is first used to construct multi-scale feature maps for each object in the input image. After that, the encoder-decoder framework with dynamic convolution is utilised to enhance model capacity and efficiency, while avoiding the issues of anchor design, proposal selection, and RoIAlign implementation. In particular, Dynamic Heads are used in the encoder to create dynamic convolution kernels, while Instance Heads are used in the decoder to provide the global feature map. With dynamic convolution, each instance can be segmented independently, reducing interference with other instances and improving segmentation accuracy. Under the supervision of projection loss and pixel point colour pairing loss, the contours of each object are finally outlined. On the PASCAL VOC and MS COCO datasets, the proposed method is competitive with more sophisticated approaches. In the VOC dataset, segmentation performance achieved 37.6% average precision with ResNet-101 and FPN networks. The extensively visualised results demonstrate the effectiveness of the proposed encoder-decoder framework with dynamic convolution.

KEYWORDS

image segmentation, object detection

1 | INTRODUCTION

In computer vision, instance segmentation is one of the fundamental tasks for separating special objects from the overall image [1]. Based on this attribute, there have been numerous applications of state-of-the-art algorithms in manufacturing, such as autonomous harvesters [2], unmanned vehicles [3, 4], and medical image analysis [5]. However, according to ref. [6], manually annotating an object instance takes 79 s on average, and a bounding box usually takes only 10 s. Therefore, weakly supervised instance segmentation (WSIS) based on the bounding box [7, 8] has been of interest to researchers since it was proposed.

To deal with weakly supervised annotations, most recent WSIS models use pseudo-labels generated from external datasets [9, 10] to train segmentation networks. Specifically, a relatively small saliency detection dataset [11] is first constructed to serve as a guide pseudo-label for box-based segmentation. After multiple training tasks [12] are used to optimise the feature maps extracted from the segmentation network by using pseudo-label guides, a series of iterations is performed to determine the optimal instance masks. Despite reaching high accuracy, these methods [13] are incredibly expensive, especially if the segmentation network is not trained by end-to-end fashion. In contrast to these methods, bounding box tightness prior (BBTP) [14] proposed a multiple instance learning (MIL) architecture in post-processing stage to overcome these limitations. The core idea of MIL is that, for each object box, at least one pixel within its horizontal and vertical direction belongs to the corresponding instance. As a result, end-to-end training can be used to implement WSIS based on box annotations. However, BBTP [14] uses an anchor-based Mask R-CNN framework that cannot take full advantage of the information outside the box due to the introduction of RoIAlign [15]. The rest of the input image regarded as background information is discarded when only cropped proposal regions are used. With increased proposals, the speed of training and inference will also slow down.

In this paper, we propose an encoder-decoder framework with dynamic convolution for WSIS. It is an anchor-free, proposal-free and RoIAlign-free framework by using fully convolutional layers, which reduces the complexity of module design and achieves a good balance between performance and cost. By introducing the single-stage fully convolutional network [16, 17], our model avoids the dependency of anchors and elaborate designs associated with the corresponding hyperparameters. Thanks to dynamic convolution [18, 19] and the global fast segmentation method [20], we design the encoder-decoder structure with dynamic convolution to achieve high quality segmentation. The encoder generates dynamically convolutional kernels that encode the characteristics of each object, and the decoder receives a global feature map and the dynamic kernels to decode instance masks. In order to supervise each detected mask, we apply the projection and pairwise loss from Boxinst [21], which fully exploits the relationship between box annotation and rest background. Experimental results on PASCAL VOC [22] and MS COCO [23] datasets demonstrate the remarkable efficiency of our proposed framework. The major contributions of the work are summarised as follows:

1. We develop an end-to-end encoder-decoder framework coupled with dynamic convolution for WSIS, termed encoder-decoder framework with dynamic convolution (EDDC). It does not require an anchor, proposal, and RoIAlign, which reduces the choice of related hyperparameters, storage in device memory, and cost of training or inference. Moreover, EDDC only uses the box annotations for supervision, making it accessible for other datasets and possibly improving its performance.
2. To alleviate dependence on a single instance, EDDC reconstructs the encoder-decoder structure to exploit both global information and fine-grained features. The encoder in EDDC creates dynamic convolution kernels to encode the characteristics and contextual connections of each instance. Taking the global feature map as input, the decoder improves segmentation accuracy based on box annotations.
3. In experimental comparisons with the PASCAL VOC and large-scale MS COCO datasets, our model outperforms recent techniques and is even competitive with some fully supervised networks without external datasets. Comprehensive ablation experiments on PASCAL VOC demonstrate the advantages of our proposed EDDC. The results of some segmentation of EDDC and state-of-the-art have also been examined, which provides an additional insight into EDDC efficiency. Some qualitative examples are shown in Figure 1.

The rest of the paper is organised as follows: Section 2 discusses the previous work on instance segmentation and the implementation of Encoder-Decoder based segmentation. Section 3 gives a detailed explanation of the proposed method. Section 4 describes the datasets, experimental setup, experimental results and qualitative analysis. Finally, the conclusion is presented in Section 5.

2 | RELATED WORK

2.1 | Fully supervised instance segmentation

With deep learning-based segmentation, significant performance gains have been achieved over the past few years. Long et al. [16] first proposed fully convolutional network with the fully convolutional network to generate a segmentation map of the same size as the input image. After that, He et al. [15] proposed the baseline Mask R-CNN, which is an anchor-based two-stage instance segmentation model that detects first, then segments. The RoIAlign operation is used to crop every possible object from the feature maps of backbone network. Additional works [24] have been proposed to make the inference more accurate and faster. For example, the PANet [25] combined Mask R-CNN [15] with FPN [26] and used an enhanced path to incorporate the underlying features into the high-level semantic layer. Similarly, Chen et al. [27] proposed a series of DeepLab [28–30], and the recently released DeepV3+ uses an encoder-decoder architecture to generate fine-grained masks. Based on the development of attentional mechanisms, Fu et al. [31] developed a dual attention model that focuses on the contextual associations. In addition, BlendMask [32] constructs box attention to areas where objects may be located on the image. However, Mask-based annotation is time-consuming and costly for fully supervised instance segmentation [6], which has led researchers to develop weakly supervised segmentation.

2.2 | Weakly supervised instance segmentation

Despite their high performance in instance segmentation [33–37], fully supervised approaches require a large amount of annotated data, which limits their applicability. As a result, some researchers have begun to pay attention to the method of weaker supervision. For example, based on bounding boxes, Anna et al. [38] proposed a pseudo mask label derived from a GrabCut algorithm [39] to train the segmentation network. Souly et al. [40] employ the GANs [41] framework to generate additional training samples. In another work, Xue et al. [42] proposed a novel adversarial network to learn both global and local features that capture long- and short-range contextual associations between pixels. Using top-level features from a trained object detector, Lee et al. [43] developed a bounding box attribute map that serves as a pseudo-ground-truth mask to provide pixel-level location information. Wang et al. [12] proposed a class-agnostic segmentation model that is trained by box annotations and salient images. Although these methods are appreciated, they require quite a

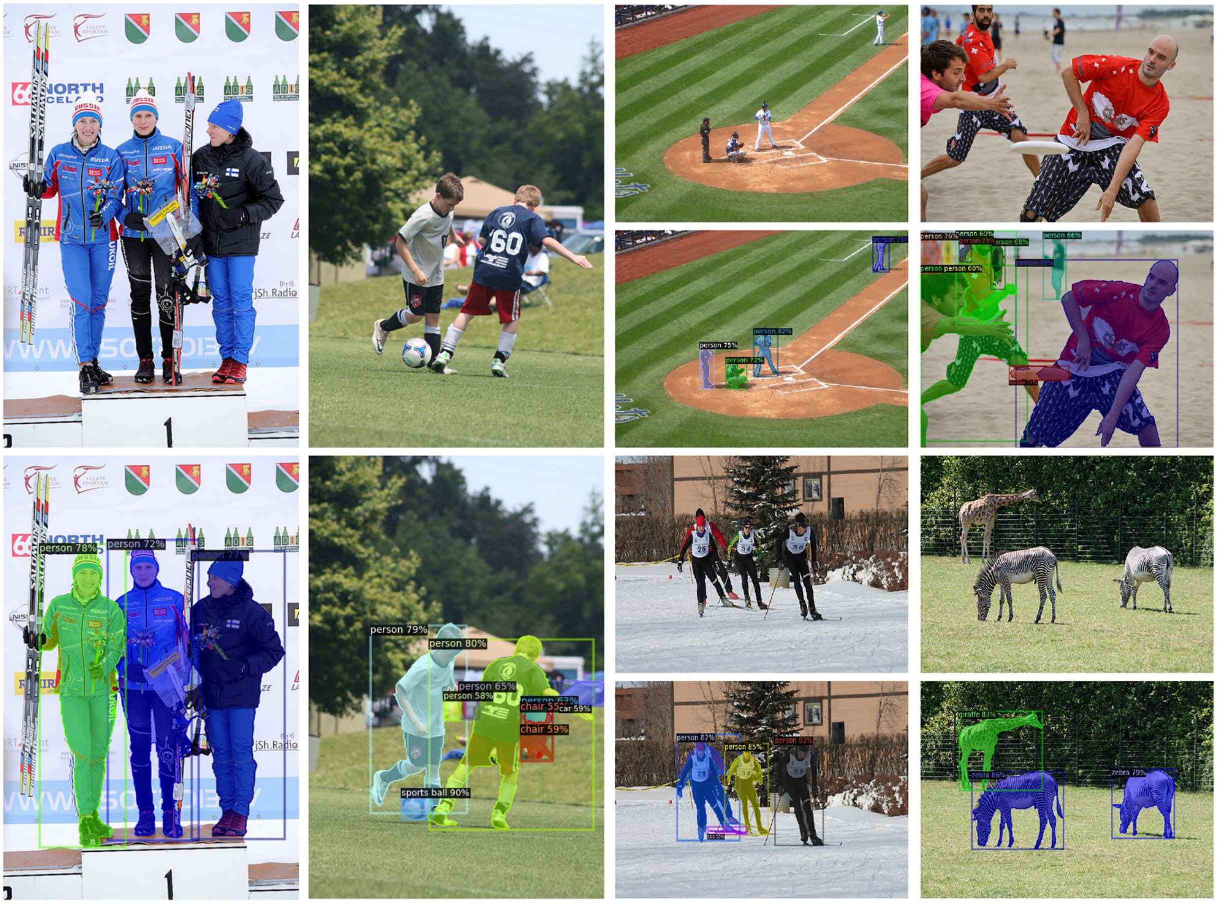

FIGURE 1 The qualitative results of encoder-decoder framework with dynamic convolution with the backbone of ResNet-101 achieving 33.0% mask average precision on COCO test-dev 2017. Better viewed in colour.

bit of time for pseudo-label generation and subsequent training.

2.3 | Dynamic convolution for instance segmentation

Since dynamic convolution was first proposed, researchers have been interested in its remarkable flexibility [18, 44–47]. By setting a weighted combination of multiple convolution kernels, Chen et al. [48] proposed a CondConv based on data dependence, which increases the capacity and size of neural networks. Similarly, dynamic convolution [19] based on attention allows the network to be more expressive by aggregating attention in a non-linear manner. Then Zhang et al. [49] proposed DyNet, which makes dynamic convolution networks lighter. Recently, SOLO, SOLOv2 [33, 34] proposed using dynamic convolution to segment instances without anchors and related hyperparameters. Additionally, Condinst [50] performs conditional dynamic convolution to filter the mask features for obtaining feature representations of individual instances. However, most of these methods tend to focus only on high-level semantic features of a single object, which may lead to incorrect segmentation or poor segmentation quality. In this work, we integrate the core of these dynamic convolution approaches and reconstruct the encoder-decoder structure to generate instance masks from the global feature map. The proposed framework alleviates the focus on a single object and refines the edges of instances.

2.4 | Encoder-decoder for image segmentation

Due to its compatibility and practicality, the encoder-decoder structure is popular for computer vision tasks. Specifically, the encoder encodes image spatial information as well as global features, and the decoder converts the encoder output into a specific shape for further classification. Early encoder usually consisted of a convolutional network similar to visual geometry group [51], without fully connected layers. For example, Segnet [52] proposed a symmetric encoder-decoder network with 13 hierarchically convolutional layers in each component for semantic segmentation. In ref. [53], U-net proposed cropping the features from the encoder output to concatenate with the decoder input, and it has made marked advances in general segmentation networks. Further, Ilyas et al. [2] added a parallel dilated convolution at the end of encoder to enlarge the effectively receptive field, and this operation aggregates global and local context to preserve all semantic. Recently, ref. [54]

proposed a multi-scale residential encoding and decoding architecture for segmenting unclear skin lesions. These works follow the same principle as traditional encoder-decoder modules, in that they extract hierarchical semantic features and transform them into required forms. In this paper, the encoder-decoder structure is quite distinct from those discussed above. Firstly, it is not a symmetric structure due to the employ of dynamic convolution and parameters can be significantly reduced. Secondly, the output of the encoder is directly sent to the decoder, avoiding the need for intermediate processing. Lastly, the encoder-decoder structure with dynamic convolution can effectively extract contextual features without auxiliary operations.

3 | PROPOSED METHOD

The proposed network is composed of fully convolutional layers for WSIS, and it developed an encoder-decoder framework with dynamic convolution, which we named EDDC is shown in Figure 2. First, we introduce the backbone & neck framework to extract the feature pyramid from the input image. Next, the Dynamical Heads are used to detect each object and output the instance kernels that encode the characteristics of the detected objects. Meanwhile, the Instance Head provides the global mask feature map as the input of instance kernels to generate the object masks. Finally, we apply projection and pairwise loss from ref. [21] to supervise the generated masks. Each step is explained in detail below.

3.1 | Backbone and neck

For an image, the backbone module is used to extract rough feature maps. We build the backbone using a deep residual network (ResNet) [55], which is composed of a sequence of residual layers and the classification layer. The ResNet are divided into five stages, each stage containing a different number of residual blocks, except for stage 1. In order to generate various scale maps, the final classification layer is removed, and only the residual structure is used as the feature extractor. As shown in Figure 2, the feature maps C2–C5 are derived from stage 2–5, and their channel dimensions are 256, 512, 1024, and 2048 respectively. At stage 1, it extracts only the shallow texture of the image, which is not suitable for use. To refine these feature maps, the FPN [26] is introduced as the neck to build multi-scale feature maps. Through a 1 × 1 convolution, all channel dimensions of C2–C5 are aligned to 256, and a 3 × 3 convolution with stride-1 is used to obtain P2–P5. Note that the feature map sizes are not resized between C2–C5 and P2–P5 by padding operation. According to fully convolutional one-stage object detection (FCOS) [17], P6–P7 are derived by downsampling P5 with two 3 × 3 convolution layers of stride-2 to realise adequate recall.

3.2 | Dynamic Head

As shown in Figure 3, each Dynamic Head contains four branches: a class branch for judging every object category, an object-regression branch for predicting corresponding box regions, a center-ness branch for depressing low-quality positive samples, and an instance branch for generating instance kernels. The detailed architecture is elaborated below.

3.2.1 | Class branch

The Class branch consists of four convolution blocks and one classification layer with kernel size 3. In each convolutional

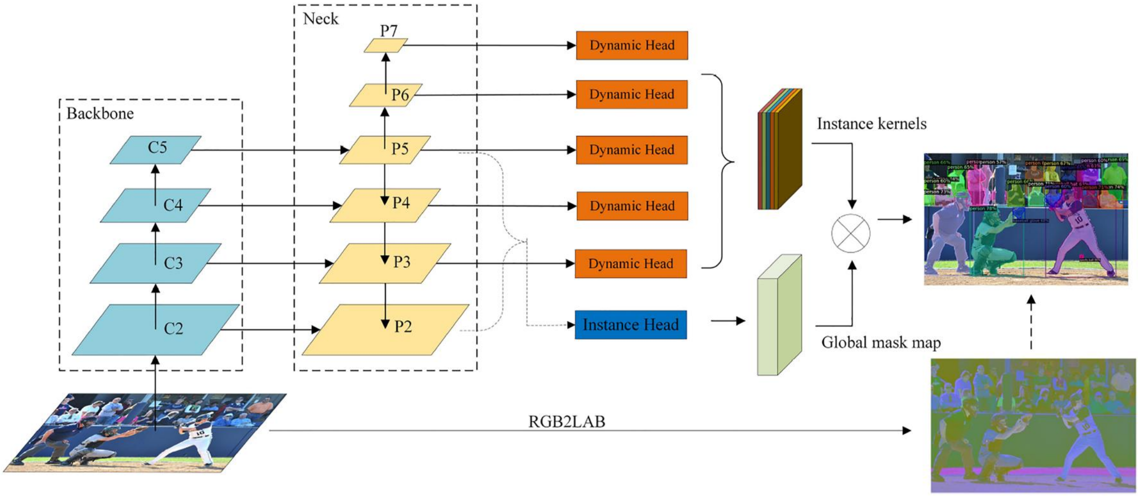

FIGURE 2 The pipeline of encoder-decoder framework with dynamic convolution. The input image is passed through the backbone and neck module to extract the pyramid feature maps P2–P5. Using two stride-2 convolutions, additional feature maps P6 and P7 are created from downsampling P5 and P6. Based on P3–P7 feature maps, the Dynamic Heads generate dynamic instance kernels to encode the characteristics of each object. Meanwhile, the Instance Head takes P2–P5 as input to output the global mask map which is decoded by the instance kernels to generate final masks. With box annotations and CIELAB colour space, the instance masks are supervised by projection and pairwise losses.

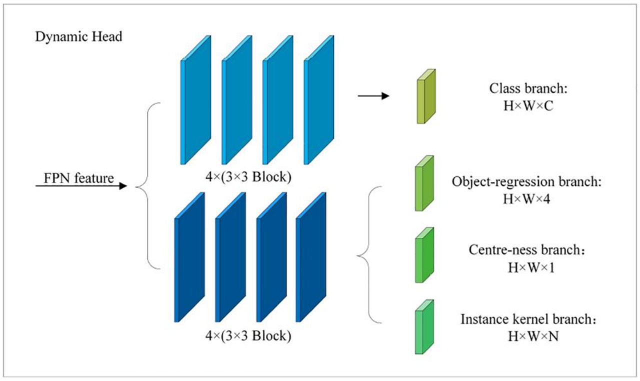

FIGURE 3 Details of the Dynamic Head. For one thing, each FPN feature map is used to determine the categories of objects by the class branch. This branch is composed of four convolution blocks and a classification layer. For another thing, the feature map passes through four parallel convolutional blocks and then is fed into the object-regression, center-ness and instance kernel branches respectively.

block, they have the same convolution layer with stride-1 and padding size-1, as well as a Group Norm with 32 channel dimension and ReLU activation function. The input and output shapes for the class branch are $H \times W \times 256$ and $H \times W \times C$ respectively, where H and W are the height and width of the corresponding feature level. Their channel dimensions remain 256 before they are fed to the classification layer. C is the number of dataset classes. For the instance segmentation benchmark PASCAL VOC [22] or COCO [23] datasets, $C = 20$ or 80. Similar to other segmentation networks, the Focal loss [56] L_{Class} is applied to optimise the Class branch.

In order to decouple from the Class branch and improve performance, the parallel four convolution blocks are utilised to extract the object characteristics. The structure of these blocks is consistent with blocks in the Class branch. After that, the next three branches share the same feature map with size $H \times W \times 256$ as their individual input.

3.2.2 | Object-regression branch

For the one-stage fully convolutional network, the most significant difference is how accurately the object is detected in comparison to two-stage anchor-based image segmentation. Without anchors to locate, the model requires other labels to detect objects. According to FCOS [17], the point (x_{i0}, y_{i0}) on the input feature map is considered as a positive sample only if it falls within the projection of ground truth boxes. Therefore, for an annotated box $B_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})$, the regression branch aims to infer the corresponding top-left and bottom-right coordinates $\{x^*_{i1}, y^*_{i1}, x^*_{i2}, y^*_{i2}\}$ based on the sample point. Similarly to Mask R-CNN [15], we do not directly infer the object coordinates, but rather predict the quadrilateral distances $l^*_i, t^*_i, r^*_i, b^*_i$ between the sample point and the projected annotated box on the input feature map. As illustrated in Figure 4, the ground truth labels are calculated as:

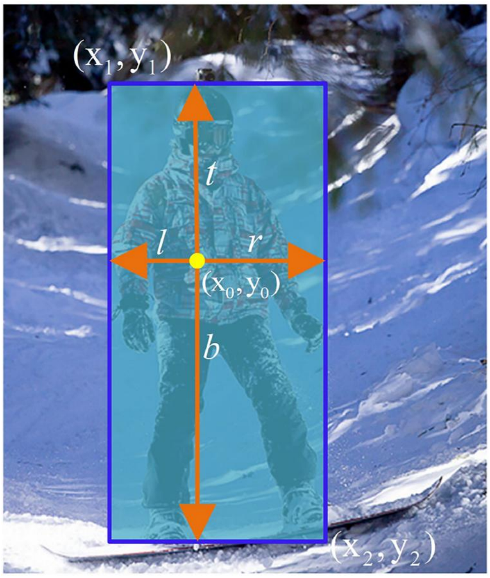

FIGURE 4 Illustrations of the box regression target. (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates of the annotated box. l, t, r, b represent the distances from the current point (x_{i0}, y_{i0}) to the box sides.

$$\begin{aligned} l_i &= x_{i0} - x_{i1}, \quad & t_i &= y_{i0} - y_{i1} \\ r_i &= x_{i2} - x_{i0}, \quad & b_i &= y_{i2} - y_{i0} \end{aligned} \tag{1}$$

From this perspective, the Object-regression branch consists of a 3×3 convolution layer and outputs the distances with shape $H \times W \times 4$. Based on this design, the GIoU loss L_{GIoU} is used to train the Object-regression branch.

3.2.3 | Center-ness branch

To better recognise the different sizes of objects and achieve a high recall rate, we use FPN [26] as the neck module to further improve detection performance. However, as described in FCOS [17] and LPSNet [20], the FPN affects the regression of high-quality boxes by generating low-quantity positive sample points on multi-scale feature maps. Thus, a single convolutional layer, known as the Center-ness branch, is used to suppress low-quality boxes before non-maximum suppression (NMS). This branch aims to output the weights centerness* that represent the distances between each point on the input feature map and the corresponding projection edges of the ground box. In order to reduce the decay rate of the Center-ness branch, we tested several power functions on the distance centerness* and found that the square root was the most suitable choice. From this assumption, the Center-ness branch consists of a 3×3 convolution layer and outputs the centerness* with shape $H \times W \times 1$. As shown in Figure 4, given a sample point and an annotated box, the ground truth centerness is defined as:

$$\text{centerness} = \sqrt{\frac{\min(l_i, r_i)}{\max(l_i, r_i)} \times \frac{\min(t_i, b_i)}{\max(t_i, b_i)}} \tag{2}$$

where l_i, r_i, t_i, b_i are available from Equation (1). Since the centerness ranges from 0 to 1, the binary cross-entropy (BCE) loss $L_{\text{BCE\_Centerness}}$ is used to train the Center-ness branch. During inference, the NMS is exploited to filter the lower products of classification scores and centerness* weights.

3.2.4 | Instance kernel branch

Dynamic convolution, in contrast to standard and local convolution, preserves translation invariance and reduces computational complexity more than the latter. Inspired by refs. [18, 19, 57], we reconstruct the encoder and decoder with dynamic convolution. Specifically, the Instance kernel branch is constructed as the encoder to capture object characteristics. The decoder is constructed by parsing the encoder output, that is, dynamic instance kernels, in a non-linear combination manner. Note that the decoder input is the Instance Head output, which will be discussed in the next subsection.

As the encoder, the Instance kernel branch generates the dynamic convolution kernels for each object. Different from refs. [18, 19, 57], the Instance kernel branch consists of a 3 × 3 standard convolution with N filters to replace the original attention weight $G = \{G_i\}^N \in \mathbb{R}^{H \times W \times N}, 0 \leq G_i \leq 1$, and it outputs the dynamic convolution kernels with shape $H \times W \times N$. For the decoder, our design principles are that its parameters should match the encoder output and its input should correspond to the Instance Head output. Within these rules, we construct the decoder by using the non-linear combination of dynamic convolution kernels to maximise its capacity as much as possible. The decoding process can be represented mathematically as follows:

$$\begin{gathered} M = \text{Relu}\left(\tilde{W}^T(x) \otimes X + \tilde{B}(x)\right) \\ \tilde{W}(x) = \sum_{i=1}^{N} G_i \tilde{W}_i(x), \tilde{B}(x) = \sum_{i=1}^{N} G_i \tilde{B}_i(x) \\ \text{s.t.} \quad \sum_{i=1}^{N} G_i = 1 \end{gathered} \tag{3}$$

where $X \in \mathbb{R}^{H \times W \times 16}$ represents the global mask map provided by the Instance Head, and $\otimes$ represents the 2-d convolution. Weight matrices $\tilde{W}_i(x) \in \mathbb{R}^{3 \times 3}$ and bias vectors are trainable parameters in relation to dynamic convolution kernels $G_i \in \mathbb{R}^{H \times W \times 1}$. N is the hyperparameter to control the number of the decoder layers, and we discuss it in Section 4.4.1. Activated by Relu, the output $M \in \mathbb{R}^{H \times W}$ predicts the instance mask for each object.

3.3 | Instance Head

The Instance Head aims to provide the global mask map as input to the decoder, that is, the dynamic convolution kernels generated by the Instance kernel branch. Most previous studies used Backbone features to extract the mask map [55, 58], while few used FPN features as the mask map to predict the instance masks. For example, in ref. [20], the Segmentation Head outputs the mask map by concatenating on the FPN feature maps, but this operation does not fuse the multi-scale information. Likewise, in ref. [50], the Mask Head considers the high-level semantic feature map as the mask map, while the low-level detailed information is neglected. Motivated by these works, we reduce the computational complexity by cutting the number of FPN feature channels while fusing these features by Upsample & Adding. As shown in Figure 5, the 256-d FPN feature maps P2–P5 are fed separately into a Conv block, which includes a 3 × 3 convolution layer with stride-1, BatchNorm2d function and ReLU activation to downscale the channels to 128-d. Subsequently, we upsample the feature map of the upper layer to the lower layer by bilinear interpolation and then sum these feature maps bit-wise. After four Conv blocks and a 1 × 1 convolution, the channel dimension of the global mask map is reduced to 16-d for better efficiency. Finally, the instance masks are obtained by feeding the global mask map $X \in \mathbb{R}^{H \times W \times 16}$ into the decoder based on Equation (3).

3.4 | Weakly supervised mask loss

With only bounding boxes as supervision, one may consider a parameter-free method to fit the edges of the objects, but it is difficult to sketch the contour features when the objects are occluded from each other. From another perspective, if we look at the colour information in the same instance mask area, it is obvious that some colours are similar. As shown in Figure 1, some colours are almost identical, such as the blue in the hats and clothes of the skating athletes. Apparently, it is not a coincidence. By comparing pixels inside and outside the instance mask, the colour prior information can be used to constrain the mask.

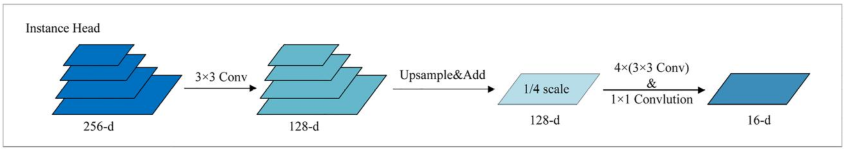

FIGURE 5 Illustration of the Instance Head. The Conv block contains a 3 × 3 convolution layer with stride-1, BatchNorm2d function and ReLU activation. The 1/4 scale means that the feature map size is one quarter of the original image.

In contrast to the pre-defined pseudo label approach, we introduce the box projection loss L_{Proj} and pairwise loss L_{Pair} from Boxinst [21], which efficiently explore the box annotations and colour priors. Specifically, in the absence of bounding boxes as weak supervision, the maximum values $n_{ix} = \max\left(x_{i1},\ x_{i2}\right), n_{iy} = \max\left(y_{i1}, y_{i2}\right)$ of grounding box B_i are extracted in each row and column with x and y directions. Similarly, maximum values n^*_{ix}, n^*_{iy} are extracted from the corresponding predicted box B^*_i. The Dice loss [59] L_{Dice} is applied to evaluate each projection pixel value, which is formulated as follows:

$$\begin{gathered} L_{\mathrm{proj}} = L_{\mathrm{Dice}}\left(n_{ix}, n^*_{ix}\right) + L_{\mathrm{Dice}}\left(n_{iy}, n^*_{iy}\right) \\ L_{\mathrm{Dice}}\left(n_{ix}, n^*_{ix}\right) = 1 - \frac{2\sum_{i=1}^{T} n_{ix} \cdot n^*_{ix} + \epsilon}{\sum_{i=1}^{T} n_{ix} + \sum_{i=1}^{T} n^*_{ix} + \epsilon} \end{gathered} \tag{4}$$

where T is the total number of predicted boxes, and ϵ is a constant to avoid division by zero.

By establishing the side association between two pixels, the pairwise loss assigns the side value of the same CIELAB colour space to 1. It can be summarised as follows:

$$P(l = 1) = \tilde{m}_{i1,i2} \times \tilde{m}_{j1,j2} + \left(1 - \tilde{m}_{i1,i2}\right) \times \left(1 - \tilde{m}_{j1,j2}\right) \tag{5}$$

where $\tilde{m}_{i1,i2}$ and $\tilde{m}_{j1,j2}$ represent the probability that points $p(i1, i2)$ and $p(j1, j2)$ are foreground in the mask M. On the contrary, the probability that colours are different is $P(l = 0) = 1 - P(l = 1)$. The probability distributions of identical or different colours are always 0 or 1, which can be supervised using BCE. Because the loss function generally leads the network equipped with an optimal set of parameters, the pairwise loss may cause the model to output directly that all pixels with the same colour are foreground or background within the predicted bounding boxes. However, the foreground output is our desired result, and the background is also almost impossible under the supervision of projection loss. As there are some pixels inside the box such that $P(l = 1)$, and all the pixels outside the control of projection loss are background. More details about the pairwise loss refer to [21]. It can be described as follows:

$$L_{\mathrm{Pair}} = -\frac{1}{N} \sum_{\tilde{m} \in M_i} \mathbb{1} \log P(l = 1) \tag{6}$$

where $\mathbb{1}$ denotes the colour similarity matrix. The segmentation loss is calculated as follows:

$$L_{\mathrm{Mask}} = L_{\mathrm{Proj}} + L_{\mathrm{Pair}} \tag{7}$$

Overall, we use the Dynamic Head to generate the dynamic convolution kernels, and the Instance Head to provide the global mask map. The projection and pairwise losses are applied to train the segmentation masks. Formally, the entire loss function to train the EDDC model can be formulated as follows:

$$L = L_{\mathrm{Class}} + L_{\mathrm{GIoU}} + L_{\mathrm{BCE\_Centerness}} + \lambda L_{\mathrm{Mask}} \tag{8}$$

where L_{Class}, L_{GIoU}, $L_{\mathrm{BCE\_Centerness}}$ respectively represent the supervision for the Class branch, the Object-regression branch, and the Center-ness branch. λ is the coefficient used to balance the mask loss for different datasets. In this paper, λ is set to 1.

4 | EXPERIMENT

4.1 | Datasets and evaluation metric

We conduct experiments on the benchmark PASCAL VOC [22] and MS COCO [23] datasets for WSIS. The 2012 version of PASCAL VOC contains 1442 images for training and 1499 images for validation, covering 20 categories of objects such as bicycles, dogs etc. Due to the relatively small size of the dataset, we follow refs [14, 21] to adopt the augmented version [60] with 10,582 training samples and leave the validation set unchanged. MS COCO contains 80 object classes with a total of 330K samples, including train2017 with 11,827 images, val2017 with 5000 images, and test-dev2017 with 20,288 images for server validation.

Based on previous studies, the standard COCO metrics, that is, mask average precision AP, AP_{50}, AP_{75}, AP_S, AP_M, AP_L are used to evaluate the segmentation performance. Because of the different metrics between PASCAL VOC and COCO, we adopt the COCO-style mask provided by ref. [14] to align the evaluation criteria in PASCAL VOC.

4.2 | Experiment detail

Following previous works, we use the ResNet-50 or ResNet-101 [55] as the backbone network, and the general FPN [26] as the neck to strengthen multi-scale features. All the models are trained on four NVIDIA 1080 Ti GPUs each with 11GB of RAM. We use stochastic gradient descent as the training optimiser with the momentum and weight decay set to 0.9 and 10^{-4} respectively. On PASCAL VOC, the mini-batch size, learning rate, and the number of iterations are set to 8, 10^{-2}, 20K. The learning rate is decreased to 10^{-3} at step 15K. On COCO, due to resource limitations, we set a smaller batch size of 4 instead of 16, and the training schedule is also adjusted accordingly. Note that the initial learning rate is changed from 10^{-2} to 2.5×10^{-3}, which decreases by a factor of 10 at iteration 240K and 320K respectively. The number of iterations is also changed from 90K to 360K. For data augment, following [50], the resize and left-right operations are used. Our models are configured by Detectron2 [61], and the backbone is initialised with the parameters of ResNet-50 or ResNet-101 pretrained on ImageNet [62]. The calculation of inference time on the whole network is conducted on a single GeForce GTX 1080 Ti GPU with batch 1. Following previous methods, the short side of the input image is resized to 800 and the long side is less than or equal to 1333.

4.3 | Comparisons with state-of-art approaches

Table 1 represents the instance segmentation results on COCO test-dev2017, including fully supervised methods with mask annotations, weakly supervised methods with pseudo labels or box annotations. These models also include with/without region proposals and RoIAlign approaches, for example, BBTP [14] and Boxinst [21]. From this table, it can be seen that EDDC achieves a 7% higher average precision (AP) than Boxinst [21] with the same backbone and training batch. Note the result about Boxinst is our replication in the second and fifth penultimate rows for a fair comparison. Our EDDC outperforms recent state-of-the-art methods such as bounding box attribution map [43], BoxCaseg [12], and even fully supervised approaches such as PolarMask [63] and YOLACT-700 [64]. DISCOBOX [13] was proposed recently, which uses a self-ensembling framework based on the teacher-student model, but the training process is not end-to-end. Our model shows comparable performance with DISCOBOX.

Table 2 compares the performance on the PASCAL VOC validation set. These methods include Boxinst [21], BBTP [14] with conditional random field, Simple does it [38] and GrabCut [39]. Our EDDC improves the state-of-the-art baseline (DISCOBOX) by 4.3 AP_{50}, 0.2 AP_{75}, which demonstrates the efficiency of the proposed method. When the backbone is replaced by ResNet-101 with stronger feature extraction capability, EDDC even achieves mask APs of 37.6%, 66.5%, and 37.7% respectively.

In addition to the performance comparison, we also compare the number of parameters and inference speed of the representative methods on the COCO val set. As the pseudo label-based method requires the use of pseudo labels generated from external datasets to train the segmentation network, the prohibitive cost makes it difficult to apply, and therefore not considered. Table 3 shows the number of parameters and inference time of different methods. All experimental settings are the same, including input batch and image size. Our proposed method is just behind Boxinst [21] in terms of parameters and inference speed, surpassing DISCOBOX [13] and PolarMask [63] due to the framework of anchor-free, proposal-free, and RoIAlign-free. While in terms of segmentation performance, our method outperforms Boxinst thanks to the introduction of global information.

TABLE 2 Results on PASCAL VOC val set.

Methods	Backbone	AP	AP_{50}	AP_{75}
GrabCut [39]	ResNet-101-FPN	19.0	38.8	17.0
SDI [38]	VGG-16	-	44.8	16.3
BBTP [14]	ResNet-101-FPN	23.1	54.1	17.1
BBTP w/CRF	ResNet-101-FPN	27.5	59.1	21.9
BoxInst [21]	ResNet-50-FPN	34.3	59.1	34.2
BoxInst	ResNet-101-FPN	36.5	61.4	37.0
DISCOBOX [13]	ResNet-50-FPN	-	59.8	35.5
DISCOBOX	ResNet-101-FPN	-	62.2	37.5
EDDC(Ours)	ResNet-50-FPN	36.2	64.1	35.9
EDDC(Ours)	ResNet-101-FPN	**37.6**	**66.5**	**37.7**

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.

Abbreviations: AP, average precision; BBTP, bounding box tightness prior; CRF, conditional random field; EDDC, encoder-decoder framework with dynamic convolution; SDI, Simple does it; VGG, visual geometry group.

TABLE 1 Results on COCO test-dev2017 set.

Methods	Supervision	Backbone	Iterations	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [15]	Mask annotations	ResNet-50-FPN	270K	**37.5**	**59.3**	**40.22**	**21.1**	**39.6**	**48.3**
PolarMask [63]	Mask annotations	ResNet-101-FPN	180K	32.1	53.7	33.1	14.7	33.8	45.3
YOLACT-700 [64]	Mask annotations	ResNet-101-FPN	405K	31.2	50.6	32.8	12.1	33.3	47.1
BBAM [43]	Pseudo labels	Deep V2-R-101	300K	25.7	50.0	23.3	-	-	-
BoxCaseg [12]	Pseudo labels	ResNet-101-FPN	405K	30.9	54.3	30.8	12.1	32.8	46.3
DISCOBOX [13]	Box annotations	ResNet-50-FPN	90K	32.0	53.6	32.6	11.7	33.7	48.4
BBTP [14]	Box annotations	ResNet-101-FPN	90K	20.8	45.1	17.0	10.4	21.7	30.30
Boxinst [21]	Box annotations	ResNet-50-FPN	270K	32.1	55.1	32.4	15.6	34.3	43.5
Boxinst[a]	Box annotations	ResNet-50-FPN	360K	30.9	53.2	31.3	15.2	33.0	41.7
Boxinst	Box annotations	ResNet-101-FPN	90K	32.5	55.3	33.0	15.6	35.1	44.1
Boxinst[a]	Box annotations	ResNet-101-FPN	360K	32.3	55.2	32.7	15.7	34.8	43.8
EDDC(Ours)	Box annotations	ResNet-50-FPN	360K	31.8	53.8	32.7	16.4	34.0	42.4
EDDC(Ours)	Box annotations	ResNet-101-FPN	360K	**33.0**	**55.9**	**33.7**	**17.2**	**35.3**	**44.0**

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.

Abbreviations: AP, average precision; BBAM, bounding box attribution map; BBTP, bounding box tightness prior; EDDC, encoder-decoder framework with dynamic convolution.

[a] Indicates the result of replication based on the official code.

4.4 | Ablation experiments

To validate and explore the effect of dynamic convolution and Instance Head, extensive ablation experiments are performed on the augmented PASCAL VOC dataset. Unless specified, our models use ResNet-50 as the backbone, and remove the neck module to avoid unnecessary interference. Other parameters are set as closely as possible to Boxinst [21]. The inference time is calculated using the code provided by Detectron2 [61] to measure the efficiency of each component. The number of parameters in each module is also presented.

4.4.1 | The influence of dynamic convolution at different layers

Table 4 shows the results for different numbers N of dynamic convolution layers with feature maps C2–C5 as input to the Instance Head. Adding the number of convolution layers improves segmentation performance, and when the number of layers is set to 3, the model accuracy tends to be saturated. Meanwhile, the inference time increases with the accumulation of convolution layers. Hereinafter, we keep the dynamic convolution layers fixed at 3 to make a better trade-off between accuracy and inference time.

4.4.2 | The effectiveness of different strategies for the encoder-decoder structure

Table 5 reports the segmentation accuracy with different convolution strategies for the encoder-decoder structure. The first line shows the results of conventional 2-d convolution with a single layer as the baseline, and the second line shows the results of CondConv [48] with parameters set by Boxinst for optimal performance. The Dynamic Conv is our proposed method that achieves the highest accuracy of 30.7%. Although Dynamic Conv consumes more trainable parameters, it can be acceptable when compared to the entire network. In Table 5, Dynamic Convolution improves the baseline by 11.7% to Conv2d on the mask AP. This indicates that our model encodes mask features more efficiently. Besides, with single Conv2d, it means our model does not fully use the encoder-decoder framework, and the results demonstrate the effectiveness of our proposed framework.

TABLE 3 Parameters and speed analysis with other methods on COCO val set.

Methods	Supervision	Backbone	Parameters (*M*)	Device	Inference (FPS)	Time (ms)
Mask R-CNN [15]	Mask annotations	ResNet-50-FPN	44.13	1080 Ti	7.17	139.50
PolarMask [63]	Mask annotations	ResNet-50-FPN	34.46	1080 Ti	5.80	172.41
DISCOBOX [13]	Box annotations	ResNet-50-FPN	46.37	1080 Ti	7.20	138.89
BBTP [14]	Box annotations	ResNet-50-FPN	44.13	1080 Ti	7.10	140.85
Boxinst [21]	Box annotations	ResNet-50-FPN	**34.04**	1080 Ti	**8.47**	**118.06**
EDDC(Ours)	Box annotations	ResNet-50-FPN	34.98	1080 Ti	7.82	127.89

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.

Abbreviations: BBTP, bounding box tightness prior; EDDC, encoder-decoder framework with dynamic convolution.

TABLE 4 Results of the numbers for the dynamic convolution layers.

Number	Backbone	Parameters	*AP*	AP_{50}	AP_{75}	Inference (FPS)	Time (ms)
1	ResNet-50	**145**	24.9	52.5	20.4	**9.72**	**102.89**
2	ResNet-50	290	27.9	54.3	25.6	9.04	110.62
3	ResNet-50	435	30.7	58.8	28.6	8.93	111.98
4	ResNet-50	580	30.5	58.4	27.7	8.86	112.87
5	ResNet-50	725	**30.9**	**59.0**	**29.1**	8.71	114.81

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.

Abbreviation: AP, average precision.

TABLE 5 Results of the different strategies for encoder-decoder structure.

Methods	Backbone	Parameters	*AP*	AP_{50}	AP_{75}	Inference (FPS)	Time (ms)
Conv2d	ResNet-50	**145**	19.0	38.8	17.0	**9.37**	**106.72**
CondConv	ResNet-50	169	30.0	57.5	27.7	9.24	108.23
Dynamic Conv	ResNet-50	435	**30.7**	**58.8**	**28.6**	8.93	111.98

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.

Abbreviation: AP, average precision.

4.4.3 | The significance of the instance head

To further explore the importance of different layers, we change the input feature maps in the Instance Head. As shown in Table 6, C5 indicates that we only use the fifth level of the feature pyramid as input for the Instance Head to generate the global mask feature map. There is no doubt that it achieves the lowest accuracy, because it is a highly compressed input for the instance kernels. Interestingly, after summing the entire feature map pyramid of C2–C5, the model achieves the highest performance without much impact on the inference time. Because the summation does not dramatically increase computational complexity, and the trainable parameters do not significantly magnify either. Additionally, the introduction of underlying features complements the detailed information for the global mask map. The results in Table 6 confirm the significance of the Instance Head to the decoder.

4.5 | Visualisation analysis

In this subsection, on the condition of the same mask loss, we visualised the results of Boxinst [21] and EDDC based on ResNet-101-FPN. As shown in Figure 6, compared with ground truth, we found that Boxinst failed to find some objects that were far from the camera, for example, sample 1 and 2. It is interesting to contrast the Boxinst results with EDDC, and the former does not segment the bottle and people in the remoter distances compared with the latter, which proves that the proposed encoder-decoder framework with dynamic convolution can focus on segmenting global objects as well as enhance the contextual connections of individual instances. Moreover, for sample 3 and 4, the segmentation quality of Boxinst is lower than EDDC. One possible reason is that the Instance Head contributes more detailed features to the segmentation masks, such as the legs of the bear and the feet of the chair. In other words, the Instance Head provides as many fine-grained features as

TABLE 6 Results of different layers for the Instance Head.

Layers	Backbone	Parameters (*M*)	*AP*	AP_{50}	AP_{75}	Inference (FPS)	Time (ms)
C5	ResNet-50	**0.89**	18.5	40.7	15.0	**12.48**	**80.13**
C4–C5	ResNet-50	1.18	26.6	52.0	24.2	10.87	92.00
C3–C5	ResNet-50	1.48	30.4	58.3	28.5	9.64	103.73
C2–C5	ResNet-50	1.77	**30.7**	**58.8**	**28.6**	8.93	111.98

Note: Bolded values indicate emphasis, meaning the highest performance compared to others.
Abbreviation: AP, average precision.

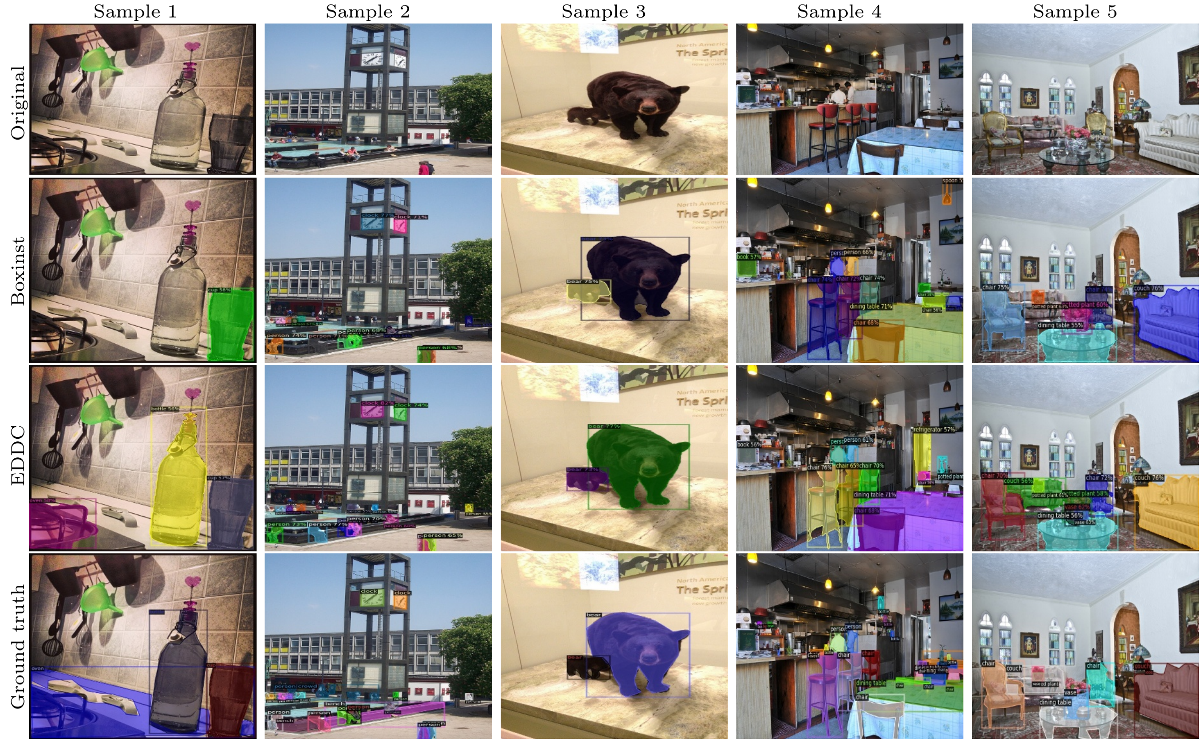

FIGURE 6 Visualization analysis of Boxinst and encoder-decoder framework with dynamic convolution (EDDC). The sample images are randomly selected from COCO val2017, and the first to fifth rows represent original, Boxinst, EDDC, and ground truth respectively. Best viewed on screen.

possible. The results of sample 5 further illustrate both of these advantages: EDDC not only segments the couch behind the potted plant, but also sufficiently preserves its edge information.

5 | CONCLUSION

To efficiently and effectively maximise utilisation of global image information, we propose a novel method named EDDC for weakly instance segmentation. It primarily consists of subnetworks of the backbone and neck, as well as the Dynamic Head and Instance Head. The former is used to extract the characteristics of each object from the input image, and the latter is used to analyse these characteristics and compose the segmentation masks. With only box-level supervision, EDDC significantly reduced the costs of annotations and produced high-quality segmentation results. As shown in experiments, the proposed EDDC with dynamic convolution and encoder-decoder structure outperforms previous approaches and achieves the state-of-the-art for WSIS.

AUTHOR CONTRIBUTIONS

Liangjun Zhu: Conceptualisation; Data curation; Methodology; Software; Writing – original draft. **Li Peng**: Funding acquisition; Methodology; Supervision. **Shuchen Ding**: Funding acquisition; Supervision; Writing – original draft; Writing – review & editing. **Zhongren Liu**: Investigation; Project administration; Validation; Visualisation; Writing – review & editing.

ACKNOWLEDGEMENTS

The authors thank the peer reviewers for their careful reading and constructive comments. This research was funded by the National Natural Science Foundation of China under Grant 61873112, 52005124.

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the correspondence author.

ORCID

Liangjun Zhu https://orcid.org/0000-0002-5304-9276

REFERENCES

1. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6656–6664 (2017)
2. Ilyas, T., et al.: Dam: hierarchical adaptive feature selection using convolution encoder decoder network for strawberry segmentation. Front. Plant Sci. 12, 189 (2021). https://doi.org/10.3389/fpls.2021.591333
3. Chang, D., et al.: Multi-lane detection using instance segmentation and attentive voting. In: 2019 19th International Conference on Control, Automation and Systems (ICCAS), pp. 1538–1542. IEEE (2019)
4. Peng, S., et al.: Deep snake for real-time instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8533–8542 (2020)
5. Isensee, F., et al.: nnu-net: self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809.10486 (2018)
6. Bearman, A., et al.: What's the point: semantic segmentation with point supervision. In: European Conference on Computer Vision, pp. 549–565. Springer (2016)
7. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2209–2218 (2019)
8. Dai, J., He, K., Sun, J.: Boxsup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1635–1643 (2015)
9. Song, C., et al.: Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3136–3145 (2019)
10. Arun, A., Jawahar, C., Kumar, M.P.: Weakly supervised instance segmentation by learning annotation consistent instances. In: European Conference on Computer Vision, pp. 254–270. Springer (2020)
11. Zeng, Y., et al.: Joint learning of saliency detection and weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7223–7233 (2019)
12. Wang, X., et al.: Weakly-supervised instance segmentation via class-agnostic learning with salient images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10225–10235 (2021)
13. Lan, S., et al.: Discobox: weakly supervised instance segmentation and semantic correspondence from box supervision. arXiv preprint arXiv:2105.06464 (2021)
14. Hsu, C.-C., et al.: Weakly supervised instance segmentation using the bounding box tightness prior. Adv. Neural Inf. Process. Syst. 32, 6586–6597 (2019)
15. He, K., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
17. Tian, Z., et al.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
18. Jia, X., et al.: Dynamic filter networks. Adv. Neural Inf. Process. Syst. 29, 667–675 (2016)
19. Chen, Y., et al.: Dynamic convolution: attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11030–11039 (2020)
20. Hong, W., et al.: LPSNet: a lightweight solution for fast panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16746–16754 (2021)
21. Tian, Z., et al.: Boxinst: high-performance instance segmentation with box annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5443–5452 (2021)
22. Everingham, M., et al.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88(2), 303–338 (2010). https://doi.org/10.1007/s11263-009-0275-4
23. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer (2014)
24. Huang, Z., et al.: Mask scoring R-CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6409–6418 (2019)
25. Liu, S., et al.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)

26. Lin, T.-Y., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
27. Chen, L.-C., et al.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
28. Chen, L.-C., et al.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40(4), 834–848 (2017). https://doi.org/10.1109/tpami.2017.2699184
29. Chen, L.-C., et al.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
30. Chen, L.-C., et al.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
31. Fu, J., et al.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
32. Chen, H., et al.: Top-down meets bottom-up for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8573–8581 (2020)
33. Wang, X., et al.: Solo: segmenting objects by locations. In: European Conference on Computer Vision, pp. 649–665. Springer (2020)
34. Wang, X., et al.: Solov2: Dynamic and Fast Instance Segmentation. arXiv preprint arXiv:2003.10152 (2020)
35. Lee, Y., Park, J.: Centermask: real-time anchor-free instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13906–13915 (2020)
36. Ghiasi, G., et al.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2918–2928 (2021)
37. Fang, Y., et al.: Instances as queries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6910–6919 (2021)
38. Khoreva, A., et al.: Simple does it: weakly supervised instance and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 876–885 (2017)
39. Rother, C., Kolmogorov, V., Blake, A.: "grabcut" interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004). https://doi.org/10.1145/1015706.1015720
40. Souly, N., Spampinato, C., Shah, M.: Semi supervised semantic segmentation using generative adversarial network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5688–5696 (2017)
41. Goodfellow, I., et al.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. 27 (2014)
42. Xue, Y., et al.: Segan: adversarial network with multi-scale l 1 loss for medical image segmentation. Neuroinformatics 16(3), 383–392 (2018). https://doi.org/10.1007/s12021-018-9377-x
43. Lee, J., et al.: BBAM: bounding box attribution map for weakly supervised semantic and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2643–2652 (2021)
44. Liu, L., Deng, J.: Dynamic deep neural networks: optimizing accuracy-efficiency trade-offs by selective execution. In: Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, pp. 3675–3682 (2018)
45. Wang, X., et al.: Skipnet: learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 409–424 (2018)
46. Wu, F., et al.: Pay less attention with lightweight and dynamic convolutions. arXiv preprint arXiv:1901.10430 (2019)
47. Zhu, M., et al.: Dynamic feature pyramid networks for object detection. arXiv preprint arXiv:2012.00779 (2020)
48. Yang, B., et al.: Conditionally parameterized convolutions for efficient inference. arXiv preprint arXiv:1904.04971 (2019)
49. Zhang, Y., et al.: Dynet: dynamic convolution for accelerating convolutional neural networks. arXiv preprint arXiv:2004.10694 (2020)
50. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 282–298. Springer (2020)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
52. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(12), 2481–2495 (2017). https://doi.org/10.1109/tpami.2016.2644615
53. Ronneberger, O., et al.: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer (2015)
54. Dai, D., et al.: Ms red: a novel multi-scale residual encoding and decoding network for skin lesion segmentation. Med. Image Anal. 75, 102293 (2022). https://doi.org/10.1016/j.media.2021.102293
55. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
56. Lin, T.-Y., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
57. Chen, J., et al.: Dynamic region-aware convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8064–8073 (2021)
58. Sun, K., et al.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5693–5703 (2019)
59. Milletari, F., et al.: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571. IEEE (2016)
60. Hariharan, B., et al.: Semantic contours from inverse detectors. In: 2011 International Conference on Computer Vision, pp. 991–998. IEEE (2011)
61. Wu, Y., et al.: Detectron2. 2(3) https://github.com/facebookresearch/detectron2 (2019)
62. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
63. Xie, E., et al.: Polarmask: single shot instance segmentation with polar representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12193–12202 (2020)
64. Bolya, D., et al.: Yolact: real-time instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9157–9166 (2019)

How to cite this article: Zhu, L., et al.: An encoder-decoder framework with dynamic convolution for weakly supervised instance segmentation. IET Comput. Vis. 1–12 (2023). https://doi.org/10.1049/cvi2.12202