# Research on Instance Segmentation via Transformer

**Lei Jin 31520211154008**
**Songlin Yu 31520211154004**
**Hong Yang 31520211154105**
**HaoWei Wang 31520211154084**
[1]School of information, Xiamen University
422 Siming South Road, Siming District
Xia Men, China 361000

## Abstract

Instance segmentation, as one of the most challenging tasks in the field of computer vision, requires the generation of a pixel-level segmentation mask for each object based on image classification. The mainstream solutions in the industry can be divided into top-down and bottom-up paradigms. The top-down paradigm can be divided into two-stage segmentation and one-stage segmentation. In order to improve the speed of inference in one-stage segmentation schemes, full-image convolution operations are often used instead of the strategy of first detection and then segmentation in the two-stage segmentation scheme. However, the translation invariance of the convolution network makes the features extracted by different instances of the same type similar, and it is difficult to distinguish them only by the whole image convolution, which leads to the reduction of the accuracy of the one-stage segmentation scheme. Aiming at the problem of one-stage segmentation accuracy reduction, an attention mechanism is proposed. This mechanism adds self-attention to the feature map. Since the position coding of self-attention can distinguish the feature vectors at different positions of the feature map, it is easier to distinguish the instances of different positions on the feature map after adding the self-attention. Through the attention mechanism, the full-image convolution operation in the one-stage segmentation scheme can better distinguish different instances of the same type, and generate high-quality segmentation masks.

## 1. Introduction

Computer vision(CV) is a research hotspot in the field of artificial intelligence. Since the invention of the AlexNet[8] network in 2012, CV has experienced the development of tasks such as simple image classification to more refined semantic segmentation and instance segmentation. With the development of deep learning and convolutional neural networks(CNN), CV has made great progress in recognition accuracy and speed. In the past ten years, a large number of theories and methods have emerged, and fruitful results have been achieved in many fields. The definition of instance segmentation is to input an image, output the category of each object in the image, and generate a pixel-level instance mask for each object at the same time[15]. Image classification, object detection, semantic segmentation, and instance segmentation are four types of CV tasks which increasing in difficulty. Image classification only needs to point out the category of the image. Object detection needs to output the bounding box of the object on the basis of image classification. Semantic segmentation needs to predict which category each pixel on the image belongs to. Instance segmentation needs to distinguish different instances of the same category on the basis of semantic segmentation.

Mask R-CNN[5], the two-stage instance segmentation scheme representation, follows the formula of first detection and then segmentation. First, the region proposal network(RPN) proposes some candidates[14], then conduct alignment and pooling operations on candidates. Second, Candidates will be send to the subsequent network for classification and mask generation. The mask generation of the two-stage segmentation scheme is based on the features of the candidates, which avoids the bad effect of background and other instances, generally has a higher segmentation accuracy. However, there are a large number of negative candidates, so the calculation is time-consuming. In order to solve this problem, one-stage segmentation scheme like YOLACT[2], SOLO[17], etc. are produced. Compared with two-stage segmentation, this kind of scheme abandons the operation of RPN and uses convolution in the whole feature map. One-stage segmentation realizes category prediction and mask generation at the same time and increases the speed. However, the full-image convolution is inevitably affected by the background in the mask generation process. Moreover, it is difficult to distinguish different instances of the same type, so the accuracy is reduced.

In order to improve the one-stage segmentation accuracy, We produces a method of adding an attention mechanism to the feature map. This can improve the discrimination between different instances, facilitate mask generation, and improve segmentation accuracy.

## 2. Related Work

### 2.1 Instance Segmentation

Most methods perform instance segmentation at the pixel level in the region proposal, which is especially effective for standard CNNs. A representative example is Mask R-CNN. It first detects the object, and then uses the mask predictor to segment the instance within the suggested box. In order to make better use of the spatial information in the frame, PANet integrates the mask prediction from the fully connected layer and the convolutional layer. This proposal-based approach achieves state-of-the-art performance. One limitation of these methods is that they cannot resolve errors in positioning, such as boxes that are too small or shifted.

There are some pixel-based methods without region proposal. In these methods, each pixel generates auxiliary information, and then a clustering algorithm groups the pixels into object instances based on their information. The auxiliary information can be diverse or grouping algorithms. [1] Predict the perceptual energy of the boundary of each pixel and use the watershed transform algorithm for grouping. [13] Distinguish instances by learning instance-level embedding. The input image is treated as a graph and returned to pixel affinity, and then processed by a graph merging algorithm. Since the mask is composed of dense pixels, post-clustering algorithms tend to be time-consuming.

Contour-based method. In these methods, the object shape includes a series of vertices along the boundary of the object. The traditional snake algorithm first introduced the contour-based image segmentation representation. They deform the initial contour into the object boundary by optimizing the manual energy relative to the contour coordinates. In order to improve the robustness of these methods, it is recommended to learn the energy function in a data-driven manner. [11] Some recent learning-based methods are not to iteratively optimize the contour, but try to regress the coordinates of the contour points from the RGB image, which is much faster. However, they are not accurate compared to the most advanced pixel-based methods.

In the field of semi-automatic labeling, try to use other networks instead of standard CNNs for contour labeling. Use recurrent neural network to predict contour points in sequence. In order to avoid sequential reasoning[10], the pipeline of the serpentine algorithm is followed and the graph convolutional network is used to predict the vertex offset of the contour deformation. This strategy significantly increases the speed of annotation while being as accurate as pixel-based methods. However, there is a lack of pipelines for instance segmentation and the special topology of contours is not fully utilized. Different from the contour as a general graph, Deep Snake uses the cyclic graph topology and introduces cyclic convolution to perform effective feature learning on the contour.

## 2.2 Transformer

Transformer[16] was proposed by the Google team in 2017. It was originally applied to machine translation tasks in natural language processing (NLP) and brought significant performance improvements. Its motivation lies in the fact that due to the inherent chronological nature of the Recurrent Neural Network (RNN), the network must process the input words serially, which brings about the problems of long-distance dependence and inability to parallelize processing. The Transformer's proposal realizes parallel processing, effectively improves training efficiency, and solves the problem of long-distance dependence with the help of self-attention mechanism.

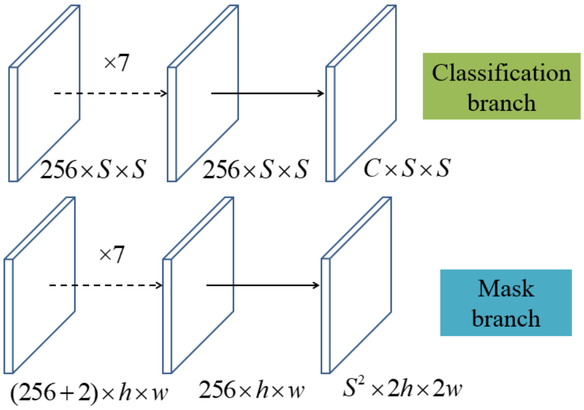

Figure 1: Classification branch and mask branch output space.

CNN has been widely used as a basic part of the visual field before. But the emergence of Transformer has challenged it. Inspired by Transformer-related ideas, scholars have applied Transformer models to classification, detection, segmentation, and multi-modal tasks, such as ViT[4], DETR[3], SETR[19], and ViLT[7]. And it has achieved competitive effects and achievements. The emergence of these results has pushed the status of Transformer to a new height and inspired scholars to carry out further research.

# 3. Method

In general, our method is based on the SOLO algorithm for improvement. Mentioned above, the SOLO algorithm divides the image into $S \times S$ grids. For each grid, SOLO predicts the category probability for the instance whose geometric center falls on the grid. In the classification network, the output space corresponding to each grid is a C-dimensional vector, where C is the number of categories in the data set. Since the image is divided into $S \times S$ grids, the output space of the classification network is $C \times S \times S$. At the same time, each grid will be responsible for the mask generation of the instances where the geometric center falls into the grid. The output space of the mask generation network is $S^2 \times 2h \times 2w$, which represents the number of output channels as $S^2$. The k-th channel corresponds to the generated mask of the i-th row and j-th column grid, which satisfies the condition: $k = i \times S + j$. Figure 1 shows the network output space.

Our method proposes a mask generation network structure with an attention mechanism, as shown in Figure 2. The proposed method is mainly divided into five parts: feature extraction, prediction head, attention mechanism, new network structure with attention mechanism, and loss function. We will describe them in detail as follow.

## 3.1 Feature Extraction

Feature extraction uses the ResNet-FPN structure. Specifically, the input image is forward by ResNet-50 [6], and outputs 4 layers of feature. This is the bottom-up calculation

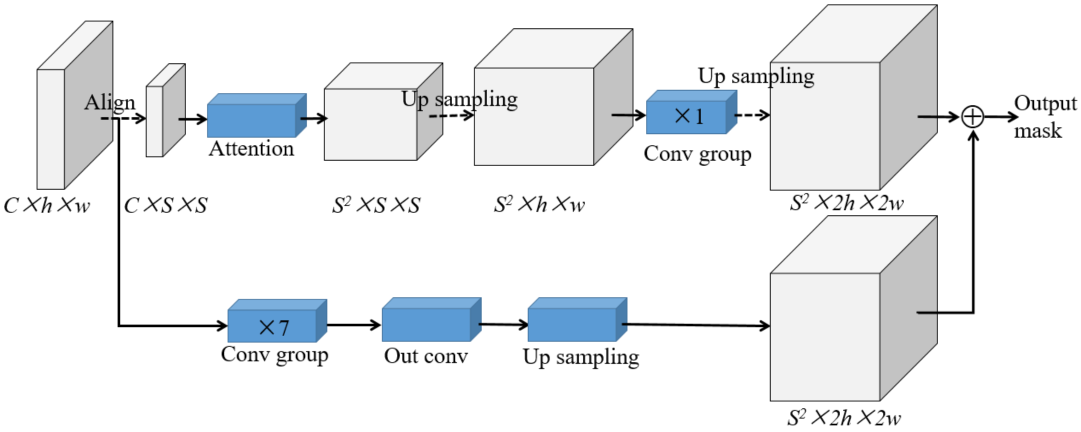

Figure 2: Mask generation network structure with attention mechanism.

Table 1: ResNet-FPN extracts detailed information of each feature layer.

| Name | Width | Height | Channels |
|---|---|---|---|
| P2 | 336 | 168 | 256 |
| P3 | 168 | 84 | 256 |
| P4 | 84 | 42 | 256 |
| P5 | 42 | 21 | 256 |
| P6 | 21 | 11 | 256 |

Table 2: The S value of each feature layer.

| Name | Value |
|---|---|
| P2 | 40 |
| P3 | 36 |
| P4 | 24 |
| P5 | 16 |
| P6 | 12 |

process in the corresponding FPN structure. Then each layer of feature is connected horizontally by a $1 \times 1$ convolution, and the feature channel is regularized to 256. At the same time, the top-down calculation process upsampling the feature layer. Then add the horizontally connected feature layer and the top-down feature layer correspondingly. Finally, the feature outputs through a $3 \times 3$ convolution. The resolution of the extracted features is lowered layer by layer from bottom to top, but the semantic information is enriched layer by layer. Low-level features are used for small object segmentation, and high-level features are used for large object segmentation. The resolution and channel number of each layer are shown in Table 1.

### 3.2 Prediction Head

The prediction head is composed of 2 branches, one branch is responsible for category prediction, and the other branch is responsible for mask generation. The classification branch first divides the feature map into $S \times S$ grids. The values of S for different feature layers are shown in Table 2.

After the feature maps of each layer are divided into different grids, they are feed into to the classification network for category prediction. The classification network consists of 7 convolution groups and 1 output convolution. Each convolution group is composed of 1 $3 \times 3$ convolution layer, 1 GN (Group Normalization) layer [18] and 1 ReLU [12] activation layer. The size of the output convolution kernel is $3 \times 3$, the step size and padding are both 1, and the output channel is the number of categories. Similarly, the mask generation network is also composed of 7 convolution groups and 1 output convolution. Different from the classification branch, the mask generation network does not need to divide the feature into $S \times S$ grids, but adds two coordinate channels to the feature layer to distinguish different instances of the same type. Position (i, j) added x, y coordinate information calculation method is as follows, where w and h represent the width and height of the feature map, respectively.

$$x = -1 + 2 \times i/w, y = -1 + 2 \times j/h \qquad (1)$$

Then the feature with the added coordinate information is sent to the mask branch for mask generation.

### 3.3 Attention Mechanism

As mentioned before, for different instances of the same type, the classification network requires the output of the same object category while the mask generation network requires the output of different instance masks. In order to achieve this goal, the feature map needs to be processed. We propose an attention mechanism. After the ResNet-FPN feature extraction, a c-dimensional vector is composed of c channels at each location of the output feature map. The c-dimensional vectors represent the features of the location instance. For different instances of the same type, they should have the distinguishing ability after attention operation. The attention mechanism proposed in this paper is an operation based on vector dot product. This mechanism performs dot product operation on the feature vectors of each position of the feature map, and uses the result of the operation as a new feature map. The result of the dot product at the same position is maximized and that at the different location is minimized, so that to enrich the difference information of differ-

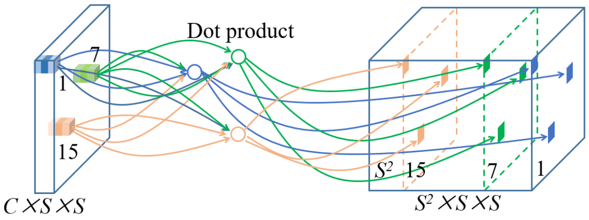

Figure 3: Illustration of attention mechanism.

ent instances in the feature map. Specifically, the positions of the feature map are embedded first, and the embedding is as follows:

$$num = i \times S + j \tag{2}$$

The corresponding relationship between the new feature and the original feature is as follows

$$F_{new}(c, i, j) = V_{num=c} \otimes V_{num=i \times S+j} \tag{3}$$

$F_{new}(c, i, j)$ represents the value of the position of the c-th channel i-th row j-th column of the new feature. $V_{num} = c$ represents the feature vector whose embedding is $c$. $V_{num} = i \times S + j$ represents the feature vector whose embedding is $i \times S + j$. Figure 3 illustrates this process in detail.

### 3.4 Network Structure with Attention Mechanism

The mask branch of the SOLO algorithm retains the size of the feature map $h \times w$, but the attention mechanism needs to divide the feature map into the grids of S × S. Different grids correspond to different instances in the image. After the attention mechanism, the feature space becomes $S^2 \times S \times S$. Inspired by the ResNet network structure, we use the attention mechanism as a separate branch. The output of this branch is directly added to the original SOLO algorithm mask generation branch. Therefore, the final mask will consist of two parts: the original SOLO mask branch and the attention branch.

### 3.5 Loss Function

We uses the SOLO algorithm loss function, which is defined as follows:

$$L = L_{cate} + \lambda L_{mask} \tag{4}$$

Among them, $L_{cate}$ is Focal Loss; $L_{mask}$ is the mask loss, please refer to [6] for details.

## 4.Experiment

### 4.1 Dataset

Instance common object segmentation on Microsoft released environment of common data sets (Common Objects in Context, COCO) data set [9], contains 80 categories. However, COCO data set is large and requires high computational power. Limited to the experimental equipment, the data set used in the experiment was part of CityScapes data set, which was annotated into COCO format. The constructed dataset consists of 450 training images and 50 verification images in 5 categories, namely person, Rider, CAR, Truck and bus.

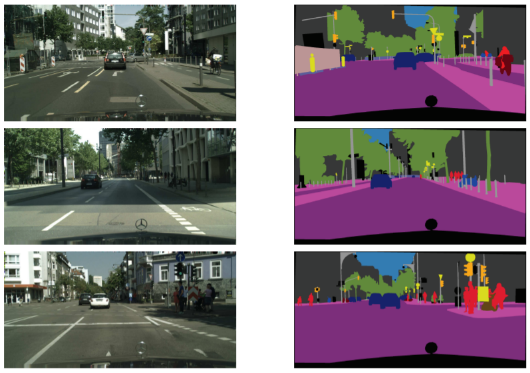

Figure 4: Training datae xamples.

Table 3: Comparison of AP with different algorithms.

| algorithms | AP | $AP_{IoU=0.5}$ | $AP_{IoU=0.75}$ |
|---|---|---|---|
| SOLO (no coords) | 11.0 | 20.1 | 11.0 |
| SOLO | 12.8 | 21.0 | 13.7 |
| Attention mechanism | 14.7 | 26.6 | 14.0 |

### 4.2 Experiment Results

In the experiment, the SOLO algorithm removing coordinate channels was taken as the benchmark, and the segmentation effects of the three methods including the SOLO algorithm removing coordinate information, the original SOLO algorithm and the addition of attention mechanism were compared. Figure 5 shows the segmentation effect of three samples.

In sample 1, the mask generated by the SOLO algorithm for the car in the center of the screen is inaccurate, and the SOLO algorithm that removes coordinate information does not generate a mask for the car in the depth of the screen. However, the attention mechanism improves these problems somewhat. In sample 2, there are multiple pedestrians overlapping on the right side of the screen. SOLO algorithm and SOLO algorithm that removes coordinate information cannot distinguish overlapping pedestrians, while the discrimination effect of attention mechanism is obviously better. In Sample 3, due to the relatively empty image and small instance area, the SOLO algorithm and the SOLO algorithm that removes coordinate information can hardly generate masks, while the attention mechanism can generate masks for pedestrians. This paper compares the accuracy of the three methods, and the results are shown in Table 3.

By comparing the SOLO algorithm with the original SOLO algorithm, it is found that the average accuracy (AP) can be improved by 1.8% only by adding coordinate information of two channels, indicating that differentiation information plays a pivotal role in instance segmentation. Compared with the SOLO algorithm and the attention mecha-

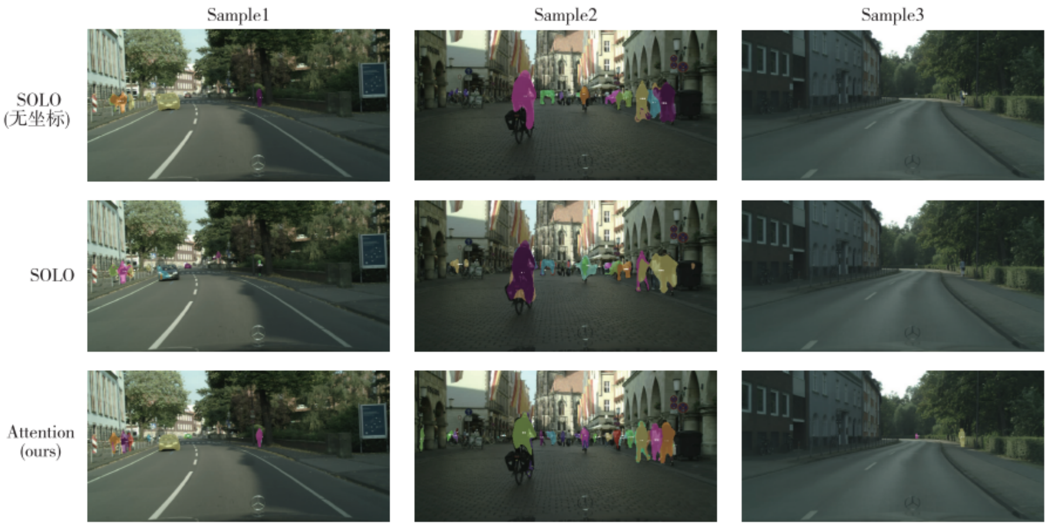

Figure 5: Segmentation example.

Table 4: Comparison of AP of instances with different size.

| algorithms | AP$_{Small}$ | AP$_{Medium}$ | AP$_{Large}$ |
|---|---|---|---|
| SOLO | 0.1 | 4.3 | 33.0 |
| Attention mechanism | 0.5 | 8.0 | 38.0 |

nism, the AP improved by 1.9% after the addition of the attention mechanism. It can be seen that the attention mechanism provides richer differentiation information, enhances the differentiation degree of different instances, and is more conducive to the generation of accurate masks.

At the same time, this paper compares the segmentation effects of different algorithms on small, medium and large objects. A closer look at Figure 5 shows that the masks generated by larger cars in sample 1 are more accurate than those generated by smaller pedestrians. In sample 2, a larger pedestrian in the front of the screen generated a more accurate mask than a smaller pedestrian in the rear of the screen. In sample 3, the SOLO algorithm could not even generate a mask for small pedestrians. This paper compares the segmentation accuracy of SOLO algorithm and attention mechanism for different area instances, and the results are shown in Table 4.

## 5. Conclusion

In this paper, an attentional mechanism is proposed for the inherent shortcoming of single-stage object detection full graph convolution, which can enhance the degree of discrimination between different instances and improve the segmentation accuracy.

The experimental results show that the attention mechanism can provide richer differentiation information than simple coordinate information, which also provides a way to improve the precision of single-stage instance segmentation in the future research. There are many different attention mechanisms in CV field, and the attention mechanism proposed in this paper has the following advantages:

1. Differences in basic ideas. Different from other attention mechanisms that focus on specific information, the purpose of the attention mechanism in this paper is to increase the degree of differentiation between different instances, so as to improve the segmentation accuracy of various instances.
2. Clear thinking and easy operation. The attention mechanism in this paper does not require similarity, but can be realized by directly adding vector dot product operation on the basis of the original algorithm.

However, the dot product needs to traverse all the positions of the feature graph, which makes the operation complicated and time-consuming. In addition, although instance segmentation has made a series of progress, there are also many challenges:

1. Small object segmentation accuracy is low. As the number of network layers increases, the receptive field becomes larger, but the resolution decreases, which is a disaster for the segmentation of small objects. The present segmentation schemes are generally better for large objects than for small objects.
2. Real-time and high-precision segmentation algorithms need to be studied. In the field of unmanned driving, not only the timeliness of recognition is highly required, but also the accuracy is highly required. At present, the case segmentation scheme is still difficult to deal with this field. How to fast segmentation with high accuracy is also an important direction of future research.
3. There are few researches on 3d image segmentation. At present, the mainstream schemes are aimed at plane image segmentation, but 3D point cloud is widely used and contains more information that plane images cannot express. Therefore, the realization of 3d point cloud segmentation will greatly enrich the application scene of instance segmentation.

## References

[1] Bai, M.; and Urtasun, R. 2017. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5221–5229.

[2] Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9157–9166.

[3] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229. Springer.

[4] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[5] He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

[6] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[7] Kim, W.; Son, B.; and Kim, I. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*.

[8] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

[9] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

[10] Ling, H.; Gao, J.; Kar, A.; Chen, W.; and Fidler, S. 2019. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5257–5266.

[11] Marcos, D.; Tuia, D.; Kellenberger, B.; Zhang, L.; Bai, M.; Liao, R.; and Urtasun, R. 2018. Learning deep structured active contours end-to-end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8877–8885.

[12] Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

[13] Neven, D.; Brabandere, B. D.; Proesmans, M.; and Gool, L. V. 2019. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8837–8845.

[14] Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99.

[15] Romera-Paredes, B.; and Torr, P. H. S. 2016. Recurrent instance segmentation. In *European conference on computer vision*, 312–329. Springer.

[16] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. 5998–6008.

[17] Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2020. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, 649–665. Springer.

[18] Wu, Y.; and He, K. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

[19] Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6881–6890.