Boosting Low-Data Instance Segmentation by Unsupervised Pre-training with Saliency Prompt

Hao Li[1,3], Dingwen Zhang[1]*, Nian Liu[2], Lechao Cheng[3],Yalun Dai[4]
Chao Zhang[5], Xinggang Wang[6], Junwei Han[1]
[1] Brain and Artificial Intelligence Lab, Northwestern Polytechnical University
[2] Mohamed bin Zayed University of Artificial Intelligence [3] Zhejiang Lab
[4] University of Chinese Academy of Sciences [5] NetEase [6] Huazhong University of Science and Technology

Abstract

Inspired by DETR variants, query-based end-to-end instance segmentation (QEIS) methods have recently outperformed CNN-based models on large-scale datasets. Yet they would lose efficacy when only a small amount of training data is available since it's hard for the crucial queries/kernels to learn localization and shape priors. To this end, this work offers a novel unsupervised pre-training solution for low-data regimes. Inspired by the recent success of the Prompting technique, we introduce a new pre-training method that boosts QEIS models by giving ***Saliency Prompt*** *for queries/kernels. Our method contains three parts: 1)* ***Saliency Masks Proposal*** *is responsible for generating pseudo masks from unlabeled images based on the saliency mechanism. 2)* ***Prompt-Kernel Matching*** *transfers pseudo masks into prompts and injects the corresponding localization and shape priors to the best-matched kernels. 3)* ***Kernel Supervision*** *is applied to supply supervision at the kernel level for robust learning. From a practical perspective, our pre-training method helps QEIS models achieve a similar convergence speed and comparable performance with CNN-based models in low-data regimes. Experimental results show that our method significantly boosts several QEIS models on three datasets.*[1]

1. Introduction

Modern CNN models address the instance segmentation task in an indirect way, by defining the localization problem on a large set of proposals [16], window centers [6, 10], or location-based masks [30, 33, 35]. A typical example is Mask-RCNN [16], which generates candidate bounding boxes using a well-designed region proposal network. Although this paradigm makes localization learning easily optimized, it still relies on the manually-designed non-maximum suppression (NMS) as post-processing to remove duplicated predictions.

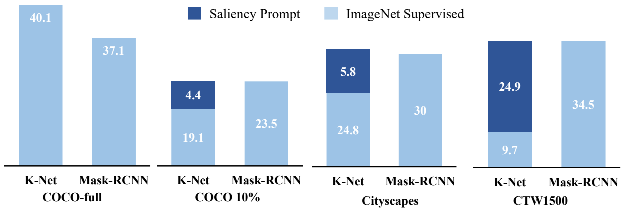


Figure 1. Performance comparison between K-Net and Mask-RCNN. K-Net can outperform Mask-RCNN on large-scale datasets (COCO-full). However, on small datasets (the right three), it can not perform as well as Mask-RCNN since it's hard to learn localization and shape priors. Our proposed unsupervised pre-training method based on saliency prompt not only boosts the vanilla K-Net significantly, but also helps to achieve comparable performance compared with Mask-RCNN.

Based on a state-of-the-art object detection model, DETR [27], a few Query-based End-to-end Instance Segmentation (QEIS) models [7, 8, 15, 18, 32, 41] have been proposed to perform instance segmentation in a new way. Unlike CNN-based methods which usually require a large set of proposals, QEIS models use dynamic queries/kernels to automatically encode object localization knowledge with different locations and object shapes. This design effectively eliminates hand-crafted anchors and post-processing like NMS. However, due to the intrinsic dynamic attribute, the kernels are forced to learn general object spatial distribution and shape priors in a data-driven manner so that they can fit any input image. This makes QEIS models require a much larger amount of training data and a much longer training time to achieve competitive performance with CNN-based methods. Once in low-data regimes [1], QEIS models will encounter much more significant performance drops than CNN-based methods, as shown in Fig-

*Corresponding author.

[1] Code: https://github.com/lifuguan/saliency_prompt

ure 1. Here we take K-Net [41] as the typical example of QEIS models and compare it with Mask-RCNN.

That being said, the potential of QEIS models is still enormous since once good localization and shape priors can be learned, they can perform on par with or even outperform CNN-based methods with a much more concise pipeline. This makes us think about how we can help QEIS models learn localization and shape priors quickly, especially for low-data regimes.

A promising solution is to adopt unsupervised pre-training, which requires no extra data and any modification to existing model architectures. However, most existing unsupervised pre-training methods [1, 3, 5, 11, 36] are only used for the backbone and can not benefit instance segmentation prediction heads, where localization and shape priors are exactly encoded. In the object detection field, some works [1, 11] do pre-train a full detection architecture. However, they use pseudo bounding boxes for training, many of which do not contain any object inside hence can not generate pseudo instance masks for instance segmentation. FreeSOLO [34] is the first method specifically designed for instance segmentation. Yet it mainly focuses on generating pseudo masks and directly using them to supervise the model training. Such a way still learns the object localization and shape priors in a data-driven manner, hence requiring tedious steps to generate high-quality pseudo masks. To address these problems, we present a novel unsupervised pre-training method for QEIS models. Inspired by the recent advances in Prompting in NLP and vision tasks [12, 19, 28, 43, 44], we propose to directly inject localization and shape priors into the kernels using our proposed **Saliency Prompt (SP)**. The prompts are generated by saliency masks which indicate potential objects, and then are used to decorate the kernels for injecting location and shape knowledge.

In detail, our saliency prompt involves two essential parts: *saliency* and *prompt*: First, a **Saliency Mask Proposal** generation method is responsible for generating saliency-level pseudo masks from unlabeled images. Instead of directly learning from noisy pseudo masks, we use them to generate corresponding region features and then achieve prompts from them. Next, a **Prompt-Kernel Matching** module matches the saliency prompts to the kernels and then injects the prior knowledge encoded in the prompts into the best-matched kernels. Furthermore, we also propose a **Kernel Supervision** scheme to supervise the model learning at the kernel level to gain kernel robustness. See Figure 2 for overview.

In our experiments, our method surpasses all the existing unsupervised pre-training algorithms on low-data regimes on four datasets. It can be used as a plug-and-play pre-training step for most QEIS methods and enables faster convergence speed and better performance without any increase in parameters or memory. Most importantly, our method achieves two *desiderata* on downstream tasks a) it leads to the same convergence speed as CNN-based methods. (b) it gains comparable or even better performance than CNN-based methods on most downstream datasets. In ablations, we find that our method shows big tolerance to the quality of pseudo masks. As such, we can easily achieve performance improvement without a sophisticated and time-consuming pseudo mask generation method as in FreeSOLO [34].

Meanwhile, it is essential that our approach has the following significant differences from the currently popular semi-supervised methods [37]: (1) Our model is a self-supervised method that works in ”pre-training+down-stream task finetuning” fashion, where the domains of the down-stream tasks are not constrained, which, in most cases, differ from the pre-training domain. However, the semi-supervised setting constrains all training data residing in a single domain. Otherwise, the semi-supervised model cannot converge based on our experiments. (2) Semi-supervised methods usually use auxiliary loss (like pseudo-label supervision) which we don’t use. Based on the above two-fold reasons, the semi-supervised works are incomparable to ours.

2. Related Work

2.1. Query-Based End-to-End Instance Segmentation

With the development of Transformers, a brand new object detector based on object-queries is proposed by DETR [2]. It considers the detection task as a set prediction problem, which makes DETR become the first end-to-end model without any human-crafted anchors or NMS. Subsequently, many works [7, 8, 15, 18, 32, 41] follow this paradigm to tackle the instance segmentation task, which we call Query-Based End-to-End Instance Segmentation (QEIS) methods. In the prediction head, instead of proposing dense object proposals, QEIS models use queries/tokens/kernels to capture individual instance features on the global scale, hence are more flexible and enable end-to-end learning. Meanwhile, various improvements have been made by different QEIS models. Inspired by the idea of SOLO-v2 [35], K-Net [41] generates convolution kernels to predict masks directly. This kernel-mask paradigm enables K-Net to segment both semantic and instance categories consistently by a group of learnable kernels. QueryInst [15] builds upon Sparse-RCNN [29] and adopts parallel supervision on dynamic mask heads. Mask2Former [7] improves the efficiency and accuracy of the prediction head by using masked-cross-attention and multi-scale feature fusion. In this work, we take K-Net as a typical example of QEIS models and develop our unsupervised pre-training method upon it. How-

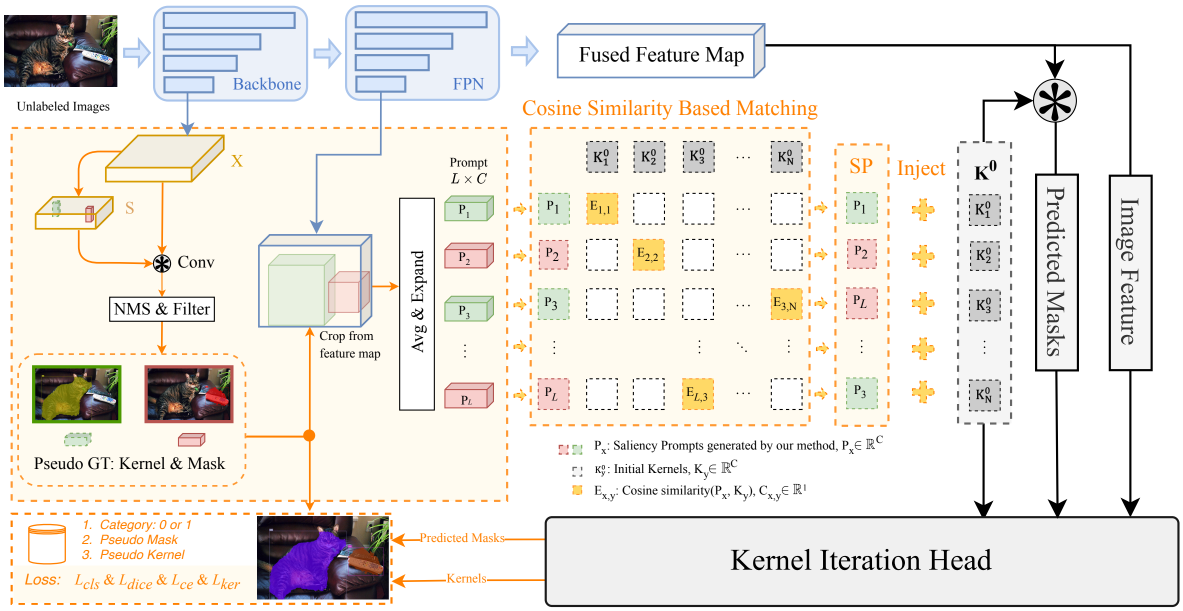


Figure 2. Overview of our proposed pre-training framework. Modules with orange colors denotes our pre-training method with the corresponding supervision. As can be seen,our method is parameter-free. Blue and gray modules denote a vanilla QEIS model, here we use K-Net for example.

ever, our method can also be deployed on other QEIS-style models freely and improve their performance on low-data regimes, as proved in the experiments part.

2.2. Unsupervised Pre-training

Unsupervised pre-training aims to pre-train deep models with carefully designed pretext tasks for boosting the model performance in downstream tasks. Most state-of-the-art unsupervised pre-training methods, such as DenseCL [36], SwAV [3], and MoCo-v2 [5], are only used to pre-train backbones and ignore the prediction head, thus can not solve the data-hungry problem of QEIS models.

UP-DETR [11] and DETReg [1] creatively build end-to-end unsupervised learning frameworks based on DETR's query-object mechanism. Specifically, UP-DETR [11] randomly crops image patches from images as pseudo labels. DETReg [1] uses region proposals generated by ResNet and SwAV as pseudo labels. However, none of these methods work for segmentation tasks as their pseudo labels would only contain backgrounds that cannot provide object and shape information. Such pseudo labels can largely mislead the training of the segmentation models.

Recent work FreeSOLO [34] concentrates on generating good-quality pseudo masks inspired by SOLOv2 [35], making the first time, pseudo labels can be used for training instance segmentation models. However, FreeSOLO requires multiple steps, such as pre-training and self-training, to generate pseudo masks. It only considers using the pseudo masks as labels to supervise the model training, without exploring how to use them in a more efficient way. Our experiments indicate that explicitly injecting object localization and shape priors contained in the pseudo labels can bring further improvements compared with solely using pseudo labels as the supervision.

2.3. Prompting

The prompting technique originates from NLP [12] and is soon transferred to the multi-modal domain [28, 43, 44]. It formulates downstream tasks as a "fill-in-the-bank" problem, such as "A photo of a {object}" in CLIP [28]. Here "A photo of a" stands for prompt templates, which guide the language model to elicit useful information from the pretrained models and predict the "{object}". CoOp [44] replaces man-defined language prompts with a set of learnable vectors. Based on CoOp, CoCoOp [43] is further purposed to generate input-conditional prompts for each image and combine them with existing language dynamic prompts. To conclude, prompting has shown its performance in the language and vision-language domain. Most recently, VPT [19] first integrates prompting into pure vision tasks. It prepends several learnable prompt tokens to the patch tokens of the frozen ViT [13] model to fit it for a variety of downstream tasks and datasets without the need of finetuning the whole ViT model. We find that most previous works use prompts to improve the performance of pre-trained models on downstream tasks. That is to say, they do not use prompts in the pre-training stage and only utilize them in the finetuning stage. Contrary to them, we ingeniously tailor the prompting mechanism for our pre-training task. Our saliency prompts are only used in the pre-training stage to help the QEIS model learn a better prediction head and then are removed in the finetuning stage.

3. Methodology

In this section, we take K-Net [41] as a typical example of QEIS models and present our proposed unsupervised pre-training method upon it. We first briefly review the K-Net model, and then show how to use our proposed saliency prompt for pre-training K-Net.

3.1. K-Net Review

In most instance segmentation scenarios, the number of instances to segment is usually assumed to be unknown (average 7.7 in COCO). K-Net dynamically encodes the instance-level information into N kernels[2], each of which is responsible for finding the pixels belonging to its corresponding instance. In particular, given the feature maps $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ of the input image and the generated kernels $\mathbf{K} \in \mathbb{R}^{N\times C}$, the instance segmentation masks $\mathbf{M} \in \mathbb{R}^{N\times H\times W}$ can be obtained by performing convolution on $\mathbf{F}$ with $\mathbf{K}$, denoted as

$$\mathbf{M} = \sigma(\mathbf{K} * \mathbf{F}), \tag{1}$$

where σ means the sigmoid activation function and $*$ denotes the convolution operation along the channel dimension.

For kernel generation, K-Net designs a dynamic kernel update mechanism that uses the segmented masks and the features $\mathbf{F}$ to enhance the kernels $\mathbf{K}$ in an iterative way. At each iteration step i, the kernel updation is formulated as

$$\mathbf{K}^i, \mathbf{M}^i = \mathcal{F}^i(\mathbf{M}^{i-1}, \mathbf{K}^{i-1}, \mathbf{F}). \tag{2}$$

The initial kernels $\mathbf{K}^0$ are randomly initialized and then learned during training, encoding general image-agnostic localization and shape priors. The kernels are then updated by $\mathcal{F}^i$ to receive image-specific information. Therefore, the learning of $\mathbf{K}^0$ can only be driven by the final instance segmentation loss and the localization loss, hence needs lots of training data and time to learn general priors.

3.2. Saliency Mask Proposal Generation

Most previous deep unsupervised segmentation methods generate pseudo masks with unsupervised algorithms for supervising the model training. Here we also follow the same pipeline and additionally use the pseudo masks for generating prompts. There exist various unsupervised algorithms for generating pseudo masks, such as selective search [31], random proposal [11], and FreeSOLO [34]. Here we adopt the saliency mechanism [14,23,24,45] since its simplicity. Our main idea is to generate dense saliency maps through foreground-background separation modeling. We first use a self-supervised pre-trained model, such as ResNet-50 [17] trained with the DenseCL [36] algorithm, as the backbone to extract image features. Then, dense feature similarity is calculated upon the output of the backbone network to generate dense saliency maps. Specifically, given the feature maps $\mathbf{X} \in \mathbb{R}^{H\times W\times D}$, we first uniformly sample $H' \times W'$ foreground seeds and generate the seed features $\mathbf{S} \in \mathbb{R}^{H'\times W'\times D}$ using average pooling on $\mathbf{X}$. Next, dense saliency is computed by using the feature of each seed $\mathbf{S}_{i,j} \in \mathbb{R}^D$ as the convolution weights to convolve the feature $\mathbf{X}$. The operation for generating a saliency map $\mathbf{Y}_{i,j}$ for the seed (i, j) is formulated as

$$\mathbf{Y}_{i,j} = \mathrm{Conv}\left(\mathbf{S}_{i,j}, \mathbf{X}\right) \in \mathbb{R}^{H\times W}. \tag{3}$$

The convolution operation calculates the similarity between the weights and the feature at each location. The locations obtain large convolution activation means they are similar to the foreground seed (i, j) hence belonging to the salient foreground. While those that have small convolution activation belong to the background of the seed. Then, we linearly normalize the saliency map $\mathbf{Y}_{i,j}$ to the range [0,1] and separate the foreground and background by using a threshold to binarize it, thus obtaining the saliency mask for the seed (i, j). Finally, each foreground seed in $\mathbf{S}$ has a saliency mask, which usually highlights the coarse region of the foreground object this seed belongs to.[3] However, different saliency masks may indicate the same object since their seeds may all belong to this object. Hence, we further use the mask NMS to filter out overlapping masks. The whole process can be formulated as

$$\mathbf{Z} = \mathrm{NMS}\left(\mathrm{Thres}\left(\mathrm{Norm}\left(\mathbf{Y}\right)\right)\right), \tag{4}$$

where $\mathbf{Z}$ denotes the final saliency mask proposals and Norm means linear normalization. The process of generating saliency mask proposals is also shown in Figure 2.

3.3. Prompt-Kernel Matching

Figure 2 shows the details of our proposed Prompt-Kernel Matching, which has two key steps: Prompt Generation and Cosine Similarity Based Matching.

Prompt generation. Given the saliency mask proposals $\mathbf{Z}$, we use their tightest bounding boxes to crop the image feature maps output by an FPN [20]. We denote the cropped features as $\mathbf{f} = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_L\}$, where L is the number of the masks in $\mathbf{Z}$ (L can vary from different images). Then, we use average pooling to convert the features $\mathbf{f}$ into prompts:

$$\mathbf{P} = \mathrm{Avg}(\mathbf{f}) \in \mathbb{R}^{L\times C}, \tag{5}$$

where C is also the channel number of the FPN feature and Avg means average pooling along the spatial dimension.

[2] N is defined to be larger than the maximum instance number in images.

[3] If one seed does not belong to any object, we usually get a null saliency mask.

Table 1. Instance segmentation fine-tune results on COCO with 5% and 10% annotated images based on K-Net.

	Pre-train	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
5% images	Img. Sup.	14.8	29.1	13.7	4.3	15.5	24.4
	DenseCL	16.7	31.2	15.9	5.1	17.5	27.7
	SwAV	15.7	30.3	14.7	4.6	25.9	16.6
	MoCo-v2	17	32	16.2	5.3	18.3	27.1
	SP(ours)	**19.9**	**35.7**	**19.9**	**6.0**	**21.0**	**32.6**
10% images	Img. Sup.	19.1	35.7	18.2	6.7	20	31.6
	DenseCL	20.3	36.4	20.3	6.6	21.8	33.6
	SwAV	18.9	34.8	18.3	6.8	20.8	30.6
	MoCo-v2	20.7	37.7	20.4	6.4	22.1	34.2
	SP(ours)	**23.5**	**41.4**	**23.7**	**7.9**	**24.8**	**38.6**

Cosine similarity based matching. Each prompt encodes the localization and shape priors for an individual object and can be injected into one of the initial kernels $\mathbf{K}^0$ of K-Net. This raises an interesting question: which prompt should be injected into which kernel? A straightforward way is to randomly or sequentially assign the L prompts to N tokens. However, as found in DETR [2] and K-Net [41], under the dynamic learning training scheme, different kernels/queries encode the localization and shape priors of different image regions and objects with different shapes, while each kernel mainly learns a specific pattern of similar object shapes and locations. As a result, using random or sequential assignments of the prompts may easily inject totally different object localization and shape information into the same token in different training samples, hence making the learning of the initial tokens very unstable. To this end, we propose a novel prompt-kernel matching scheme based on cosine similarity to assign the best-matched prompt to each token. Specifically, given the L prompts $\mathbf{P} = \{\mathbf{P}_l \in \mathbb{R}^C\}_{l=1}^{L}$ and the N initial kernels $\mathbf{K}^0 = \{\mathbf{K}_n^0 \in \mathbb{R}^C\}_{n=1}^{N}$, we compute the cosine similarity between them to build the similarity matrix $\mathbf{E} \in \mathbb{R}^{N\times L}$:

$$\mathbf{E}_{n,l} = \frac{\mathbf{K}_n^0}{\|\mathbf{K}_n^0\|_2} \cdot \frac{\mathbf{P}_l}{\|\mathbf{P}_l\|_2}. \tag{6}$$

Then, for each kernel n, we select the best-matched prompt index $\delta(n)$ with the largest similarity score:

$$\delta(n) = \mathop{\arg\max}_{l\in[1,\ldots,L]} \mathbf{E}_{n,l}. \tag{7}$$

Next, the best-matched prompt $\mathbf{P}_{\delta(n)}$ is injected into the kernel n via summation:

$$\mathbf{K}_n^{0'} = \mathbf{K}_n^0 + \mathbf{P}_{\delta(n)}. \tag{8}$$

Finally, the decorated initial kernels $\mathbf{K}^{0'}$ are fed into the prediction head of K-Net. As such, each kernel can get the best-matched localization and shape awareness to ease its learning.

3.4. Loss Function and Kernel Supervision

We use the saliency mask proposals $\mathbf{Z}$ as pseudo labels to perform bipartite matching with the N predictions of the tokens and then use the set prediction loss to supervise the pre-training, which is the same as the original K-Net [41]. The overall loss function $\mathcal{L}_K$ of K-Net is assembled by three components: the focal loss [21] $\mathcal{L}_{cls}$ for classification, the Dice loss $\mathcal{L}_{dice}$ and cross-entropy loss $\mathcal{L}_{ce}$ for segmentation, which only consider supervising the predictions. Since the predictions are mainly generated by the kernels, we argue that we can directly supervise the kernels as a supplementary loss.

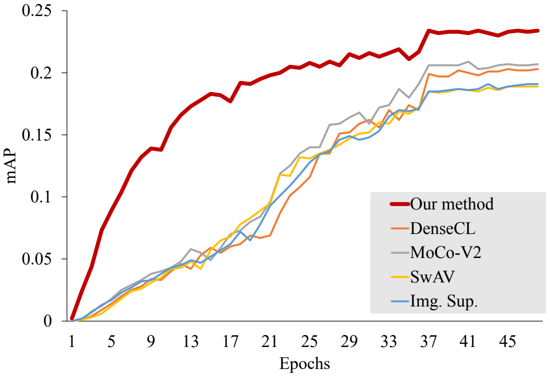


Figure 3. AP learning curves of K-Net with different pre-training methods on COCO with 10% annotated images.

Specifically, for each mask proposal $\mathbf{Z}_l$ where $l \in [1, ..., L]$, we can find its corresponding saliency seed and feature $\mathbf{S}_l \in \mathbb{R}^D$, which encodes the representative object information of this mask. Then, we transform its channel number to C for supervising the embedding of the token whose prediction is matched with the proposal $\mathbf{Z}_l$ after bipartite matching. We denote the index of the token matched with $\mathbf{Z}_l$ as n_l. Then, the kernel supervision loss can be formulated as

$$\mathcal{L}_{ker} = \sum_{l=0}^{L}\sum_{i}(1 - \mathrm{Cos}(\mathrm{Linear}(\mathbf{S}_l), \mathbf{K}_{n_l}^i)), \tag{9}$$

where the supervision is adopted for every K-Net kernel update iteration step i and summed over all mask proposals. Linear means a linear transformation to reduce the channel number to C and Cos denotes the cosine similarity.

Our final loss can be defined as

$$\mathcal{L}_K = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{dice}\mathcal{L}_{dice} + \lambda_{ce}\mathcal{L}_{ce} + \lambda_{ker}\mathcal{L}_{ker}. \tag{10}$$

where $\lambda_{(\cdot)}$ are corresponding loss weights. Since our pseudo labels are class-agnostic, we use binary classification (foreground *v.s.* background) for $\mathcal{L}_{cls}$.

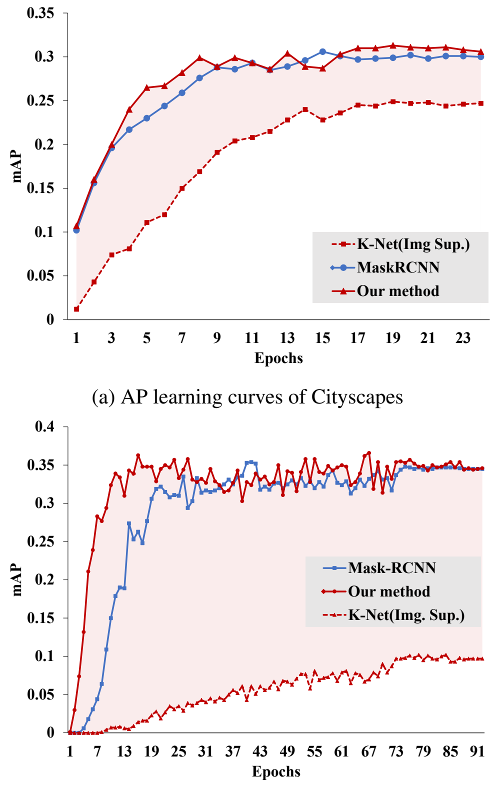
4. Experiments

4.1. Implementation Details

Pre-training setting. ResNet-50 [17] is applied for all models as the backbone and pre-trained with the DenseCL algorithm. We adopt the AdamW optimizer with 0.05 weight decay and 1,000 steps of linear step warmup. As for data augmentation, we simply apply random flipping. The

Table 2. Instance segmentation fine-tune results on Cityscapes and CTW1500.

Model	Pre-train	Cityscapes							CTW1500						
		Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask RCNN [16]	Img. Sup.	24	30	57.4	-	8.3	27.9	49	96	34.5	69.8	32.1	25.9	40.4	36.8
SOLO-v2 [35]	Img. Sup.	24	24.9	44.4	-	1.8	20.1	50.4	96	27.9	59.6	23.3	8.7	30.2	41
K-Net	Img. Sup.	24	24.8	47.4	-	4.8	19.9	43.1	96	9.7	26.5	6.1	3.0	9.2	19.2
	DenseCL	24	28	52.2	-	6.6	25.2	55.2	96	18.9	42.6	15.1	7.0	18.8	32.5
	SwAV	24	27.4	52.1	-	5.3	22.7	49.2	96	9.1	25.8	4.8	2.7	9	19.8
	MoCo-v2	24	28.2	51.2	-	5.9	26.4	52.7	96	13.3	32.2	10	4.3	13.3	24.2
	SP(ours)	**24**	**30.6**	**55.4**	**-**	5.8	**27**	**54**	**96**	**34.6**	**71.1**	**31**	18.0	**36.1**	**45.7**



(a) AP learning curves of Cityscapes

(b) AP learning curves of CTW1500

Figure 4. AP learning curves of Mask-RCNN, vanilla K-Net and our method on Cityscapes and CTW1500.

model is trained with a batch size of 96 for 12 epochs on 8 A100 GPUs. The initial learning rate is set to 1×10^{-4} and decreased by 0.1 after 8 and 11 epochs. As for the hyperparameters of our model, we set the number of kernels/queries N as 100, $\mathcal{L}_{cls} = 2$, $\mathcal{L}_{dice} = 4$, $\mathcal{L}_{ce} = 1$ and $\mathcal{L}_{ker} = 1$.

Fine-tuning setting. All models are trained with a batch size of 96 on 8 A100 GPUs. Random flipping and rotation are used as data augmentation. Referring to the open-source of MMDetection [4], we use the same hyperparameters for QEIS and CNN-based models. For QEIS, we apply the same training strategy as the pre-training stage except the training epoch, which will illustrate in the experiment tables. For CNN-based models, we apply SGD as the optimizer with weight decay and momentum. The learning rate is 0.02, momentum is set to 0.9 and weight decay is 0.0001.

Datasets. We pretrain our QEIS models on MS COCO [22] unlabeled 2017 split and fine-tune on multiple datasets, including MS COCO, Cityscapes [9] and CTW1500 [26]. MS COCO is a popular instance segmentation dataset that contains 164k labeled images, where objects from 80 object categories are annotated with dense pixel segmentation. Cityscapes is a popular instance segmentation dataset which focus on semantic understanding and urban street scenes. It contains 5000 fine annotated large-scale images. CTW1500 is a wild-scene dataset that focuses on text detection and segmentation. It contains 1,500 images with dense annotation, which is also a typical low-data regime.

4.2. Fine-tune Results on MS COCO

To evaluate our performance in low-data regimes, we split the MS COCO train2017 dataset into two different types of training subset: 1) COCO with 10% fully annotated images, which contains 12k+ images, 80k+ annotated masks; 2) COCO with 5% fully annotated images, which contains 5k+ images, 43k+ annotated masks.

Table 1 shows the comparison results on MS COCO with different pre-training methods. Img. Sup. denotes the ImageNet supervised pre-training. As can be seen, vanilla K-Net performs poorly in low-data regimes. However, our pre-training method significantly boosts its performance compared with the ImageNet supervised pre-training: up to **+5.1** AP on 5% COCO and **+4.4** AP on 10% COCO .

Moreover, from the AP learning curves on 10% COCO images shown in Figure 3, we can observe that our method converges much faster than other methods and gain much higher AP at the beginning of the fine-tuning stage. These evidences reflect that our method has probably learned shape and localization prior in the pre-training stage.

4.3. Fine-tune Results on Other Datasets

In this part, we test our method in wild scenes with unseen targets (CTW1500 [26]) and small objects (Cityscapes [9]). The comparison results are shown in Table 2. As can be seen, our method outperforms the ImageNet-supervised method by **+5.8**AP and achieves better performance(**+0.6**AP) compared with Mask-RCNN, a representative CNN-based model, on CItyscapes. On CTW1500, our method surpasses the supervised and unsupervised methods by **+24.9**AP and **+15.7**AP. It also achieves comparable performance(**+0.1**AP) with Mask-RCNN.

These experiments further demonstrate that our method

Table 3. Instance segmentation fine-tune results on QueryInst and Mask2Former.

Model	Pre-train	CTW1500							Cityscapes						
		Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask2Former [7]	Img. Sup.	80	38.8	67.6	41.6	15.6	41.2	57.4	24	29.1	52.4	-	5.6	23.9	55
	DenseCL	80	43.2	71.6	48.5	18.4	47.6	59.9	24	27.5	48.9	-	5.2	23.8	53.7
	SwAV	80	41.2	69.1	46.1	17.6	45.2	58.1	24	30.3	53.3	-	5.4	23.3	59
	MoCo-v2	80	43.3	71.2	49.2	18.9	47.6	59.4	24	30.7	54.3	-	5.4	25.5	56.4
	SP(ours)	**20**	**52.9**	**83.4**	**62.1**	**29.4**	**56.4**	**67.6**	**24**	**31.8**	**55.8**	**-**	5.1	**26.5**	**59.0**
QueryInst [15]	Img. Sup.	80	28.3	53.7	28.6	9.8	29	41.8	24	29.1	53.2	-	6.7	27.4	50.7
	DenseCL	80	31.6	56.7	33.4	10.4	32.5	46.6	24	30.8	54.7	-	8.6	28.9	54.5
	SwAV	80	24.6	50	23.1	8.1	25	36.3	24	30.7	54.4	-	7.9	28.5	53.9
	MoCo-v2	80	31.6	56.8	32.8	12.6	32	45.8	24	31.4	54.4	-	8.1	28.4	56.1
	SP(ours)	**20**	**39.2**	**66.8**	**43.1**	**16.7**	**42.2**	**51.9**	**24**	**32.8**	**57.3**	**-**	**8.8**	**29.2**	**57.0**

Table 4. Ablation of kernel supervised learning.

Model	L_{ker}?	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
K-Net	✓	**23.5**	**41.4**	**23.7**	**7.9**	**24.8**	**38.6**
K-Net	✗	23.1	41.4	23.0	7.5	24.6	37.8

could enable K-Net to learn localization and shape prior rather than simply remembering objects during the pre-train stage. When compared with Mask-RCNN in small object scenarios in terms of AP_S, our method did not show superior results. We conjecture there are two reasons: 1) the intrinsic deficiency of the QEIS paradigm—both vanilla SOLOv2 and K-Net perform extremely badly compared with Mask R-CNN; 2) Saliency Mask Proposals mainly provide large-scale pseudo labels, which makes our kernels pay more attention to big objects rather than small objects.

Figure 4 shows the AP learning curves on Cityscapes and CTW1500. Although QEIS models like K-Net perform much worse than traditional CNN models like Mask-RCNN in both convergency speed and final accuracy, equipped with our method, K-Net converges much faster than the ImageNet supervised pre-training models and can gain considerable learning curves compared with Mask-RCNN.

4.4. Deployed on QueryInst and Mask2Former

Besides K-Net, here we further apply our pre-train method on another two QEIS methods:QueryInst [15] and Mask2Former [7]. As shown in table 3, with our pre-train method, QueryInst outperforms the state-of-the-art unsupervised pre-training method by **+7.6** AP on CTW1500 dataset and achieves gains of **1.4** AP on Cityscapes dataset. For Mask2Former, our method achieves significant gains of **+ 9.6** AP over the state-of-the-art unsupervised pre-training method on CTW1500. These results indicate that our pre-training method can help the kernels/queries of QEIS models to learn localization and shape prior effectively and help gain competitive performance improvement.

4.5. Ablation Study

We perform ablation analysis to understand the impact of each component of our pre-train method. In general, we pre-train K-Net with our method on COCO unlabeled2017 for 12 epochs and then fine-tune on COCO train2017 with 10% images for 48 epochs.

Loss function. Firstly, we investigate the contribution of the proposed kernel supervision loss. As shown in table 4, this loss function yields clear improvement, indicating that the kernel supervision plays a complementary role to prediction supervision—using the noisy prediction supervision alone may lead to over-fitting.

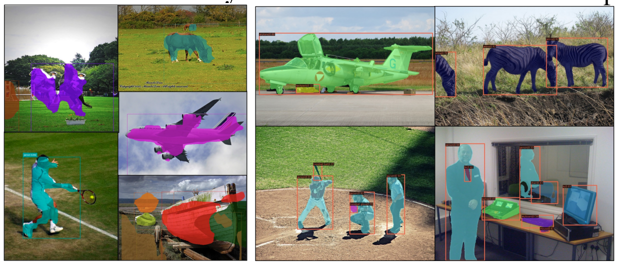
Table 5. Ablation of different prompt approaches. '✗' means the model only pre-trained by pseudo labels without prompt.

Prompt Approch	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✗	21.7	38.7	21.7	7.2	23.5	36.2
Random Assign	21.3	38.3	21.2	7.6	22.5	35.5
Seq. Assgin	23	40.6	23	8	24.3	37.6
Cosine Similarity	**23.5**	**41.4**	**23.7**	**7.9**	**24.8**	**38.6**

Cosine Similarity Based Matching. Table 5 shows the evaluation results of several prompt approaches, including Random Assignment, Sequential Assignment and Cosine Similarity based matching. Random Assignment is called 'shuffle' in UP-DETR [11], which leads to a performance drop of 0.4AP compared to the method even without using prompt. Sequential Assignment simply expands the number of saliency prompt and attaches them to the initial kernels of K-Net, which achieves 1.7 AP improvement compared to Random Assignment. Then, Cosine Similarity further surpasses Sequential Assignment by 0.5 AP.

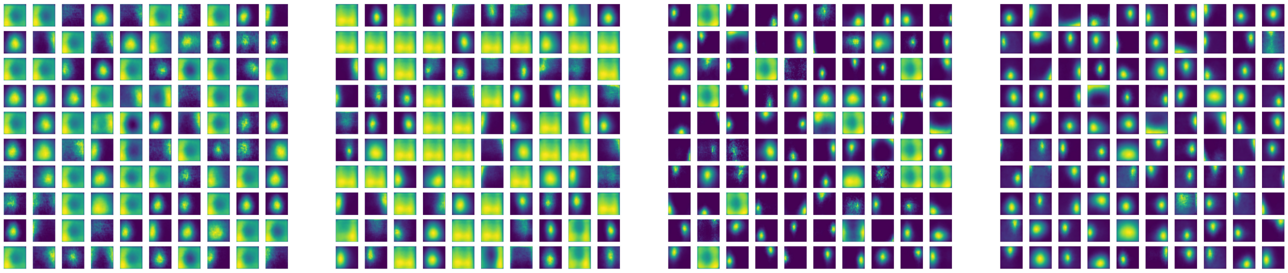
Class-agnostic Object Detection. We convert our proposed masks into bounding boxes like FreeSOLO [34], and compare with UP-DETR [11] and DETReg [1]. Table 6 shows the results of class-agnostic object detection on COCO val2017 benchmark. As can be seen, our method achieves better performance than other pre-training methods without self-training. Although FreeSOLO has better performance than other methods, its self-training process requires much longer training time (extra 14 hours) and larger memory cost.

Pseudo Mask Analysis. Here we evaluate our Prompt-



(a) Saliency Masks Proposal. (b) Fine-tune on COCO 10% images.

Figure 5. Examples of our Saliency Mask Proposals and fine-tuned results.



(a) Train from scratch. (b) Pre-trained by MoCo-v2. (c) Pre-trained by our method. (d) Trained on COCO train2017-full.

Figure 6. Average activation of the 100 kernels over 5000 images on COCO val2017. All masks are resized to 200 × 200 for analysis.

Table 6. Unsupervised class-agnostic object detection results.

Method	Self-train?	AP	AP_{50}	AP_{75}
FreeSOLO [34]	✓	5.5	12.2	4.2
UP-DETR [11]	✗	0	0	0
DETReg [1]	✗	1.0	3.1	0.6
K-Net *w* SP	✗	3.2	8.5	2.0

Table 7. Ablation of Pseudo Labels on COCO with 10% images. P means our Prompting method.

Approch	Pseudo Label (quality)	mAP
K-Net w/o P	Rand. Prop. (bad)	0.8
K-Net w P	Rand. Prop. (bad)	10.0
K-Net w/o P	Saliency (normal)	21.7
K-Net w P	Saliency (normal)	23.5
K-Net w/o P	FreeOLO (good)	23.3
K-Net w P	FreeSOLO (good)	24.0

ing method on three kinds of pseudo labels with different qualities. As illustrated in table 7, pseudo labels generated by 'Random Proposal' have a negative impact on the results of the fine-tuned model, yet our pre-training method achieves significant improvement by utilizing Saliency Prompt. Moreover, our Prompting method achieves competitive performance with both normal and good-quality pseudo masks, improving AP by 1.8 and 0.7, respectively. Besides, our method can also approach the performance of FreeSOLO which requires a much longer training time and many process steps. Notably, with the proposed Prompting approach, pre-train on normal pseudo masks gains comparable or even better performance than pre-train on good yet time-consuming pseudo masks like FreeSOLO. We visualize our pseudo masks in Figure 5.

4.6. Kernel Spatial Distribution Analysis

To further justify whether the proposed saliency prompt can assist the kernels of the prediction head to learn localization and shape priors, we visualize the average of mask activations of the 100 instance kernels over 5000 images on the 2017val split after 2 training epochs.

The best result (Figure 6d) is from the fully trained K-Net on COCO train2017. Those kernels have learned different shape and location priors. However, the priors from the supervised (Figure 6a) and the compared unsupervised (Figure 6b) method mainly focus on the central area. Surprisingly, the kernels learned by our pre-train method (Figure 6c) show positive trends in the diversity of shape and location priors, which is close to the fully trained kernels. These results demonstrate that the kernels pre-trained with Saliency Prompt have learned effective spatial distribution and shape discrimination ability.

5. Conclusion

This paper first points out that the QEIS models lack spatial distribution and shape awareness and perform poorly in low-data regimes. Hence we present **Saliency Prompt**, a novel unsupervised pre-train method using visual prompt, which can significantly boost the performance of QEIS models on low-data instance segmentation and achieve comparable or even better performance compared with CNN-based models. From perspective of technical, it is the first paper that explores the application of prompting in the instance segmentation field. We hope its novel design elements provide insights for future works on visual-based prompting mechanisms. In the future, we will follow more recent studies on visual saliency [24, 25, 39] for further promoting the prompt learning mechanism and apply the prompt learning mechanism to advance the weakly supervised learning community [38, 40, 42].

Limitations. Most of our saliency masks are large-scale and simple textures, which makes our pre-trained kernels/queries mostly focus on large objects rather than small objects. Compared with the accuracy improvements on large objects, our pre-train method achieves limited improvement on small ones. We believe there is plenty of room to further optimize our proposed method.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62293543, Grant U21B2048 and Grant 62106235, and in part by the Exploratory Research Project of Zhejiang Lab under Grant 2022PG0AN01.

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14605–14615, 2022. 1, 2, 3, 7, 8

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 5

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2, 3

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[6] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollar. Tensormask: A foundation for dense object segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2061–2069, 2019. 1

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 2, 7

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1, 2

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[10] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 534–549, Cham, 2016. Springer International Publishing. 1

[11] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1601–1610, 2021. 2, 3, 4, 7, 8

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] Chaowei Fang, Haibin Tian, Dingwen Zhang, Qiang Zhang, Jungong Han, and Junwei Han. Densely nested top-down flows for salient object detection. *arXiv preprint arXiv:2102.09133*, 2021. 4

[15] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6910–6919, 2021. 1, 2, 7

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[18] Jie Hu, Liujuan Cao, Yao Lu, ShengChuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. Istr: End-to-end instance segmentation with transformers. *arXiv preprint arXiv:2105.00637*, 2021. 1, 2

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 2, 3

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6

[23] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Pixel-wise contextual attention learning for accurate saliency detection. *IEEE TIP*, 29:6438–6451, 2020. 4

[24] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, pages 4722–4732, 2021. 4, 8

[25] Nian Liu, Wangbo Zhao, Ling Shao, and Junwei Han. Scg: Saliency and contour guided salient instance segmentation. *IEEE Transactions on Image Processing*, 30:5862–5874, 2021. 8

[26] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang. Detecting curve text in the wild: New dataset and new solution. *CoRR*, abs/1712.02170, 2017. 6

[27] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *arXiv preprint arXiv:2111.06091*, 2021. 1

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3

[29] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021. 2

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019. 1

[31] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 4

[32] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5463–5474, 2021. 1, 2

[33] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*, pages 649–665. Springer, 2020. 1

[34] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 2, 3, 4, 7, 8

[35] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, 33:17721–17732, 2020. 1, 2, 3, 6

[36] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 2, 3, 4

[37] Zhenyu Wang, Yali Li, and Shengjin Wang. Noisy boundaries: Lemon or lemonade for semi-supervised instance segmentation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16826–16835, 2022. 2

[38] Dingwen Zhang, Junwei Han, Gong Cheng, and Ming-Hsuan Yang. Weakly supervised object localization and detection: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5866–5885, 2021. 8

[39] Dingwen Zhang, Junwei Han, Yu Zhang, and Dong Xu. Synthesizing supervision for learning deep saliency network without human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 42(7):1755–1769, 2019. 8

[40] Dingwen Zhang, Wenyuan Zeng, Jieru Yao, and Junwei Han. Weakly supervised object detection using proposal-and semantic-level relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 8

[41] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338, 2021. 1, 2, 4, 5

[42] Wangbo Zhao, Jing Zhang, Long Li, Nick Barnes, Nian Liu, and Junwei Han. Weakly supervised video salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16826–16835, 2021. 8

[43] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2, 3

[44] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 3

[45] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *IEEE TPAMI*, 2022. 4