VCP-CLIP: A Visual Context Prompting Model for Zero-Shot Anomaly Segmentation

Zhen Qu[1,2], Xian Tao[1,2,3](✉), Mukesh Prasad[4], Fei Shen[1,2,3], Zhengtao Zhang[1,2,3], Xinyi Gong[5], and Guiguang Ding[6]

[1] CAS Engineering Laboratory for Intelligent Industrial Vision, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{quzhen2022,taoxian2013,fei.shen,zhengtao.zhang}@ia.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] CASI Vision Technology Co., Ltd., Luoyang, China
[4] University of Technology Sydney, Sydney, Australia
mukesh.prasad@uts.edu.au
[5] Hangzhou Dianzi University, Hangzhou, China
gongxinyi@hdu.edu.cn
[6] Tsinghua University, Beijing, China
dinggg@tsinghua.edu.cn

Abstract. Recently, large-scale vision-language models such as CLIP have demonstrated immense potential in zero-shot anomaly segmentation (ZSAS) task, utilizing a unified model to directly detect anomalies on any unseen product with painstakingly crafted text prompts. However, existing methods often assume that the product category to be inspected is known, thus setting product-specific text prompts, which is difficult to achieve in the data privacy scenarios. Moreover, even the same type of product exhibits significant differences due to specific components and variations in the production process, posing significant challenges to the design of text prompts. In this end, we propose a visual context prompting model (VCP-CLIP) for ZSAS task based on CLIP. The insight behind VCP-CLIP is to employ visual context prompting to activate CLIP's anomalous semantic perception ability. In specific, we first design a Pre-VCP module to embed global visual information into the text prompt, thus eliminating the necessity for product-specific prompts. Then, we propose a novel Post-VCP module, that adjusts the text embeddings utilizing the fine-grained features of the images. In extensive experiments conducted on 10 real-world industrial anomaly segmentation datasets, VCP-CLIP achieved state-of-the-art performance in ZSAS task. The code is available at https://github.com/xiaozhen228/VCP-CLIP.

Keywords: Zero-shot · Anomaly segmentation · CLIP

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-72890-7_18.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15127, pp. 301–317, 2025.
https://doi.org/10.1007/978-3-031-72890-7_18

1 Introduction

In the field of industrial visual inspection, zero-shot anomaly segmentation (ZSAS) endeavors to accurately localize and segment anomalous regions within novel products, without relying on any pre-customized training data. Due to its significant potential applications in scenarios with data privacy concerns or a scarcity of annotated data, ZSAS has garnered increasing attention from researchers [5,8,10,31]. Unlike traditional anomaly segmentation methods [26], ZSAS requires strong generalization ability to adapt to significant variations in visual appearance, anomalous objects, and background features across different industrial inspection tasks.

In recent, CLIP [22] has emerged as a vision-language foundation model for addressing the ZSAS task. As shown in Fig. 1(a), existing CLIP-based methods map images and their corresponding two-class text into a joint space and compute cosine similarity. Image regions that have high similarity with the defect-related text are considered as anomalies. For example, WinCLIP [10], AnVoL [8], and APRIL-GAN [5] extract dense visual features by applying multi-scale windowing or patching to images and align normal and abnormal image regions separately through a two-class text prompt design. However, the existing CLIP-based methods [5,8,10,31] present significant challenges in practical applications. On the one hand, previous methods [5,8,10] assume that the product category (e.g., wood) of inspected images is known in advance and utilize this information to design product-specific textual prompts (e.g., a photo of a normal wood). However, the product categories are unattainable or unpredictable in data privacy scenarios, rendering these methods unusable. Furthermore, we conducted an experiment in which we replaced the product categories (names) in the text prompts with semantically similar terms in WinCLIP, such as substituting *bottle* with *container* or *vessel.* We observed fluctuations in segmentation performance of up to ±8% in terms of Average Precision (AP) metric. This motivates us to reconsider the importance of product names in text prompts, especially since some product names are ambiguous (e.g., *pcb1*, *pcb2*, *pcb3* in the VisA [33] dataset). Even within the same product category, significant differences arise due to specific components and differences in the production process, such as variations in appearance color, size, and manufacturing materials, among others. Recently, AnomalyCLIP [31] attempted to design object-agnostic text prompts, but they replaced all product name with a uniform description "object", leading to challenges in adapting to complex industrial scenarios. On the other hand, mapping images and text separately into a joint space [5,10,31] without any interaction does not facilitate mutual understanding of various modalities, and easily leads to image overfitting to certain text prompts. As illustrated in Fig. 1(a), where the output image and text embeddings are directly aligned, this approach results in a limited grasp of diverse modalities, thereby affecting anomaly segmentation performance.

To address the aforementioned problems, a straightforward and effective visual context prompting (VCP) model based on CLIP is proposed for ZSAS task. As shown in Fig. 2(a), we aim to perform anomaly segmentation on novel (unseen) products (such as bottle and hazelnut) after training on limited seen

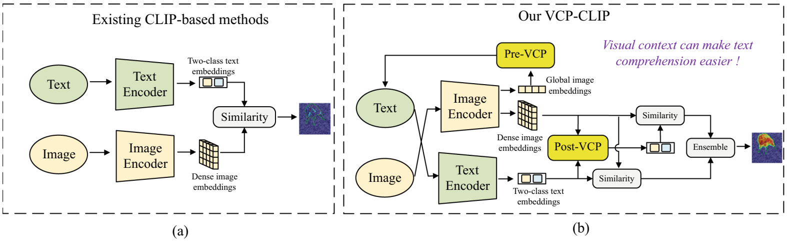

Fig. 1. A comparison between existing CLIP-based methods and VCP-CLIP. VCP-CLIP introduces a Pre-VCP module and a Post-VCP module, offering a distinct enhancement over existing CLIP-based methods. (a) Existing CLIP-based methods. (b) VCP-CLIP

products (such as cashews and pcb1) in auxiliary datasets. Existing methods [5,10] rely on manually defined text prompts as shown in Fig. 2(b). The unified text prompts are used as the baseline as shown in Fig. 2(c) in this paper, where the product categories are set as continuous learnable tokens. The proposed Pre-VCP module, depicted in Fig. 2(d), is an upgraded version of the baseline. It incorporates global image features to more accurately encode the product category semantics in the text space. To facilitate understanding of global image features, a deep text prompting (DTP) technique is introduced to refine the text space. Compared to the baseline, Pre-VCP enables the transition from uniform prompts to image-specific prompts, significantly reducing the cost of prompt designs. To enhance the mutual understanding of features from different modalities, the Post-VCP module is further proposed, which adjusts the output text embeddings based on fine-grained visual features. This approach further strengthens CLIP's ability to accurately segment anomalous regions.

In conclusion, we propose a visual context prompting model based on CLIP (VCP-CLIP) for the ZSAS task. As depicted in Fig. 1(b), we extract the global and dense image embeddings from the image encoder. The former is integrated into the input text prompts after passing through the Pre-VCP module, while the latter is utilized for fine-grained image features in anomaly segmentation. A Post-VCP module is further designed to update the text embeddings based on fine-grained visual features, effectively facilitating mutual understanding between different modalities and further enhancing the model's generalization ability to novel products. The final anomaly maps simultaneously integrate segmentation results aligned from the original text embeddings and dense image embeddings, which helps further enhance the segmentation performance.

The main contributions of this work are as follows:

1. We propose a novel visual context prompting model based on CLIP, namely VCP-CLIP, to tackle the ZSAS problem. By training on a limited set of seen products, VCP-CLIP can localize anomalies in any unseen product, even when the product category is unknown. Compared to current text prompting approaches [5,8,10,31], our approach utilizes visual context prompting to fully activate CLIP's anomalous semantic perception ability.

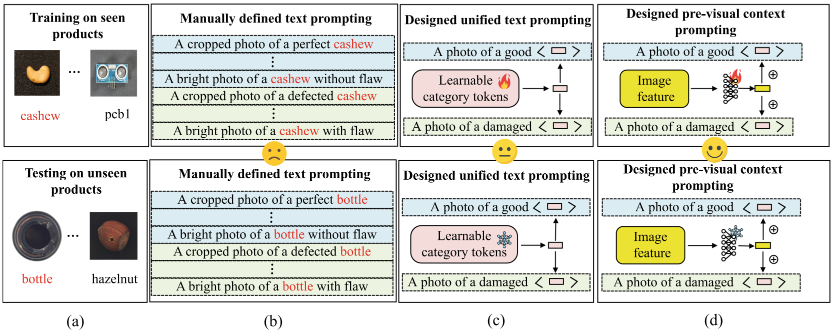
Training on seen products
cashew
pcb1
Manually defined text prompting
A cropped photo of a perfect cashew
A bright photo of a cashew without flaw
A cropped photo of a defected cashew
A bright photo of a cashew with flaw
Designed unified text prompting
A photo of a good < ▭ >
Learnable category tokens
A photo of a damaged < ▭ >
Designed pre-visual context prompting
A photo of a good < ▭ >
Image feature
⊕
A photo of a damaged < ▭ >
Testing on unseen products
bottle
hazelnut
Manually defined text prompting
A cropped photo of a perfect bottle
A bright photo of a bottle without flaw
A cropped photo of a defected bottle
A bright photo of a bottle with flaw
Designed unified text prompting
A photo of a good < ▭ >
Learnable category tokens
A photo of a damaged < ▭ >
Designed pre-visual context prompting
A photo of a good < ▭ >
Image feature
A photo of a damaged < ▭ >
(a)
(b)
(c)
(d)

Fig. 2. Comparison of different text prompting methods. (a) Task setting. (b) Manually defined text prompting. (c) Designed unified text prompting. (d) Designed pre-visual context prompting.

2. We reveal for the first time that visual context provides additional information for text prompts in the ZSAS task. Specifically, the Pre-VCP and Post-VCP modules are designed to utilize global and fine-grained image features for text prompting, respectively. In doing so, VCP-CLIP avoids extensive manually defined text prompting engineering, thus alleviating the overfitting issue arising from pre-training on specific text prompts.
3. In extensive experiments conducted on 10 real-world industrial anomaly segmentation datasets, VCP-CLIP exhibits superior zero-shot performance in segmenting anomalies on unseen products.

2 Related Work

Prompt Learning. Prompt learning is initially applied in the field of NLP, aiming to utilize affordable annotated data to automatically generate prompts, thereby enhancing the capabilities of foundation models, such as CLIP [22], GPT-3.5 [21], and LLaMA [27] in downstream tasks. CoOp [22] first introduces prompt learning in the CLIP model, utilizing learnable prompt tokens in the textual space. VPT [11] and ZegCLIP [32] insert trainable embeddings in each layer of the image encoder, allowing refinement of the image space to better adapt to downstream semantic segmentation task. These methods aim to enable the pretrained backbone to adapt to the target domain using prompt learning. In recent works, CoCoOp [30] and DenseCLIP [23] guide the pretrained backbone to adapt to the target domain through the visual context prompting. Related to our VCP module is CoCoOp, which incorporates visual contexts into text prompts to improve the classification performance on novel categories. However, our VCP replaces product categories within the text prompts rather than the entire sentence, in contrast to CoCoOp. The proposed approach has been validated as more effective than CoCoOp in ZSAS, which does not necessitate prior knowledge of product categories.

Zero-Shot Anomaly Segmentation. With the advancements of foundation models such as CLIP [22] and SAM [13], ZSAS has increasingly captured the attention of researchers. According to whether auxiliary data for training is required, existing methods can be broadly categorized into two groups. 1) Training-free methods. Building upon CLIP, WinCLIP [10] and AnVoL [8] carefully craft text prompts to identify anomalies without training on auxiliary datasets. The former proposes a window-based approach, aggregating classification results from images within different scale windows using harmonic aggregation. The latter utilizes V-V attention instead of the original Q-K-V attention in the image encoder to extract fine-grained features and adaptively adjusts for each image during testing in a self-supervised manner. SAA/SAA+ [4] utilizes language to guide the Grounding DINO [16] for detection of anomalous regions and then employs SAM for finely segmenting the detection results. However, these existing methods not only require more complex prompt designs or post-processing but also introduce additional computational and storage burdens during inference. 2) Training-required methods. APRIL-GAN [5], CLIP-AD [6], and AnomalyCLIP [31] utilize seen products with annotations as auxiliary data to fine-tune CLIP for ZSAS on unseen products. These approaches employ linear layers to map patch-level image features to a joint space of text and vision, facilitating alignment between different modalities. AnomalyGPT [9] is another seminal work that utilizes the large language model Vicuna [7] to guide the model in locating anomalies. Through supervised pretraining on synthesized anomaly images, AnomalyGPT can support multi-turn dialogues and locate anomalies in unseen products. However, existing methods all overlook the role of visual context in fine-grained multimodal alignment, and they may struggle when confronted with complex industrial anomaly segmentation scenes. Recently, ClipSAM [14], an integration of CLIP and SAM, has been employed for cross-modal interaction in ZSAS task. However, the two-stage prediction has increased the complexity of the model.

3 Our Method

3.1 Problem Definition

Our approach follows the generalized ZSAS methods adopted in works [5,31], which requires segmenting the anomalies in unseen products C^u after training on seen products C^s with pixel-annotations. During the training stage, the model generates pixel-wise classification results based on two categories of textual descriptions: normal and abnormal. During the testing stage, the model is expected to directly segment anomalies in unseen products. It is worth noting that $C^u \cap C^s = \emptyset$ and the products used in the training and testing stages come from different datasets. This undoubtedly poses a significant challenge to the model's domain generalization capability.

3.2 The Design of Baseline

Existing CLIP-based improvement methods have three main drawbacks: 1) manually designing text prompts is time-consuming and labor-intensive, 2) product-

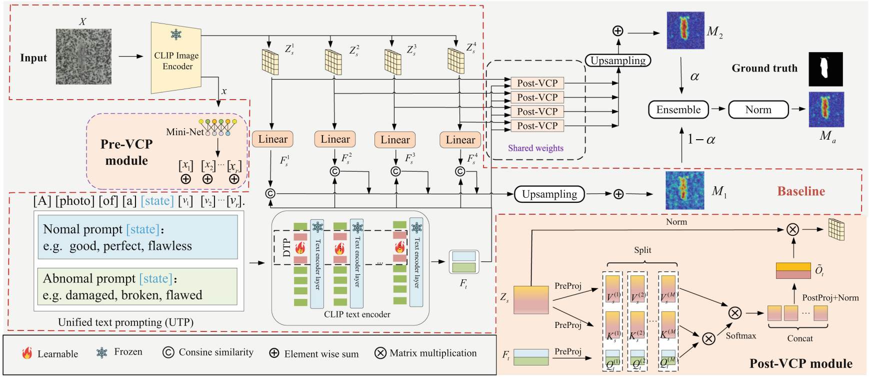

Fig. 3. Framework of VCP-CLIP. Our approach incorporates richer visual knowledge into the textual space, and cross-modal interaction between textual and visual features by using a Pre-VCP module and a Post-VCP module.

specific text prompts cannot adapt to data privacy scenarios, and 3) the localization results are easily influenced by the semantics of product categories in the text prompts [31]. To address the aforementioned issues, we propose a baseline that incorporates two main designs: unified text prompting (UTP) and deep text prompting (DPT). As shown in Fig. 3, given an input image $X \in \mathbb{R}^{h \times w \times 3}$ and two-class text prompts, the designed baseline (marked in red dashed) first extracts patch-level image features and text features separately. Then, the patch-level image features are mapped to a joint space, where the similarity between image features and text features is computed to generate anomaly maps. Finally, anomaly maps from multiple intermediate layers of the image encoder are fused after upsampling to obtain the final results.

Unified Text Prompting (UTP). A unified template for generating normal and abnormal text prompts is designed as follows:

$$H = [a][photo][of][a][state][v_1][v_2]\cdots[v_r]$$

where $v_i, i \in \{1, 2, \cdots r\}$ is a C-dimensional learnable vector embedded into the word embedding space, used to learn the unified textual context of the product categories. A pair of opposing [state] words, such as “good/damaged” and “perfect/flawed”, is utilized to generate normal and abnormal text prompts, respectively. H represents the word embedding matrix corresponding to specific prompts in the textual space. In this paper, we choose a common state word pair, i.e. “good/damaged”.

Deep Text Prompting (DTP). Before statement, let us first review the inference process of the CLIP text encoder briefly. Before being fed into the text encoder, [SOS] and [EOS] are respectively added to the front and back of the text prompt, indicating the beginning and end of the sentence. Afterwards, these tokens are mapped to a discrete word embedding space, capped to a fixed length of 77 in CLIP. Let us denote the word embeddings as $[s, H, e, J] \in \mathbb{R}^{77 \times C}$, where s and e are C-dimensional word embeddings corresponding to [SOS] and [EOS] tokens, respectively. J is a placeholder matrix initialized to zero to ensure a fixed length of the word embeddings. The final output text embedding at the position of the [EOS] token is aligned with the image features after passing through a linear projection layer.

To better align fine-grained normal and anomalous visual semantics with text, deep text prompting is designed to further refine the textual space as shown in Fig. 3. In specific, continuous trainable embeddings are inserted at the beginning of text embedding in each transformer layer of the text encoder. Assuming the text encoder's (i+1)-th layer is represented as $Layer_{i+1}^{text}$, the inserted embeddings are $P_i \in \mathbb{R}^{n \times C}$ and the output text embedding is g. The process is formulated as follows:

$$[s_i, \_, H_i, e_i, J_i] = Layer_i^{text}([s_{i-1}, P_{i-1}, H_{i-1}, e_{i-1}, J_{i-1}]) \tag{1}$$

$$g = TextProj(Norm(e_{N_t})) \tag{2}$$

where $i = 1, 2, \cdots N_t$, $s_0 = s$, $H_0 = H$, $e_0 = e$. N_t is the number of text encoder layers. $TextProj(\cdot)$ and $Norm(\cdot)$ respectively denote final text projection and LayerNorm [1] layers. For normal and abnormal text prompts, we denote the embeddings after DTP as g_n and g_a, respectively. Since the masked self-attention is employed in the text encoder, $[s_i, P_i, H_i, e_i, J_i]$ and $[s_i, H_i, P_i, e_i, J_i]$ are not mathematically equivalent. We adopted the former because the model can only attend to tokens before itself, thus placing the learnable embeddings at the beginning of the sentence leads to a greater degree of refinement in the textual space. More details are shown in the Appendix B.2.

How to Acquire the Anomaly Map? For an input image $X \in \mathbb{R}^{h \times w \times 3}$, patch-level visual feature map $Z_s^l \in \mathbb{R}^{H \times W \times d_I}, l = 1, 2, \cdots, B$ are extracted from the image encoder layers, where $H = h/patchsize, W = w/patchsize$, d_I is the size of image embeddings and B is the number of extracted intermediate patch-level feature layers. Then, the feature maps are mapped to a joint space and align with text embeddings using a single linear layer by calculating the cosine similarity. Let us respectively denote the visual and textual features in the joint space as $F_s^l \in \mathbb{R}^{HW \times C}$ and $F_t = [g_n, g_a] \in \mathbb{R}^{2 \times C}$, where C is the embedding size in the joint space. The process of acquiring the anomaly map can be formulated as:

$$M_1^l = softmax(Up(\widetilde{F}_s^l \widetilde{F}_t^T)/\tau_1), l = 1, 2, \cdots B \tag{3}$$

where τ_1 denotes the temperature coefficient, which is set as a learnable parameter. $Up(\cdot)$ is an upsampling operation with bilinear interpolation. $\widetilde{(\cdot)}$ represents the L_2-normalized version along the embedding dimension.

3.3 The Design of VCP-CLIP

The baseline has made some progress, but still faces the following three main problems: 1) The unified text prompt does not consider specific visual contexts.

2) Overfitting phenomena may occur in the unified text prompt. 3) Insufficient interaction between information from different modalities limits further improvement in segmentation performance. In this end, we further designed two novel visual context prompting modules, namely Pre-VCP and Post-VCP as shown in Fig. 3. In contrast to the baseline, the global features of the image are encoded into the text prompt using the Pre-VCP module. The Post-VCP module receives patch-level features from the image encoder and text features from the text encoder as inputs to generate the anomaly map.

Pre-VCP Module. We designed a Pre-VCP module to introduce global image features into the text prompts of the baseline. Due to the extensive alignment of image-text pairs during the pretraining process of CLIP, the embedding at the [CLS] token position of the image encoder encompasses rich global image features. We combine the global image features with learnable vectors in the baseline to facilitate the fusion with the unified category contexts. Specifically, the global image features are initially mapped to the word embedding space through a small neural network, namely *Mini-Net*. This can be expressed as $\{x_i\}_{i=1}^{r} = h(x)$, where $x_i \in \mathbb{R}^{1\times C}, i = 1, 2, \cdots r$ represents the mapping results, which are combined with embeddings corresponding to the product category:

$$z(x, v) = [z_1(x_1, v_1), z_2(x_2, v_2), \cdots, z_r(x_r, v_r)] \tag{4}$$

where $z_i = x_i + v_i$. For the *Mini-Net* $h(\cdot)$, a parameter-efficient design utilizing only a one-dimensional convolutional layer with (r, 1×3) kernels is employed. The final text prompt based on Pre-VCP can be expressed as follows:

$$H_v = [a][photo][of][a][state][[z_1(x_1, v_1)][z_2(x_2, v_2)] \cdots [z_r(x_r, v_r)]$$

For convenience in the subsequent text, we refer to the text prompt template as “a photo of a [state] $[z(x, v)]$”.

Post-VCP Module. To further enable the text embedding to adapt based on fine-grained image features, we devised a Post-VCP module, as illustrated in Fig. 3. The text embedding $F_t \in \mathbb{R}^{2\times C}$ and flattened visual embedding $Z_s^l \in \mathbb{R}^{HW\times d_I}$ from each layer are projected into a latent space with C-dimension. Then the learnable queries Q_t, keys K_s^l, and values V_s^l can be obtained:

$$Q_t = F_t W_t^q, K_s^l = Z_s^l W_s^k, V_s^l = Z_s^l W_s^v \tag{5}$$

where $W_t^q \in \mathbb{R}^{C\times C}, W_s^k \in \mathbb{R}^{d_I\times C}, W_s^v \in \mathbb{R}^{d_I\times C}$ are linear projection matrices in the *PreProj* layer. To capture richer visual features for fine-tuning text, a multi-head structure is adopted for computing attention maps to update text features within each head using matrix multiplication:

$$\{Q_t^{(m)}\}\{K_s^{l(m)}\}\{V_s^{l(m)}\} = Split(Q_t, K_s^l, V_s^l) \tag{6}$$

$$A_t^{l(m)} = SoftMax(Q_t^{(m)} K_s^{l(m)T}), \quad O_t^{l(m)} = A_t^{l(m)} V_s^{l(m)} \tag{7}$$

$$O_t^l = Concat(O_t^{l(1)}, O_t^{l(2)}, \cdots, O_t^{l(M)}) W_t^o \tag{8}$$

where $m = 1, 2, \cdots, M$. M is the number of heads, $Q_t^{(m)} \in \mathbb{R}^{2\times(C/M)}, K_s^{l(m)} \in \mathbb{R}^{HW\times(C/M)}, V_s^{l(m)} \in \mathbb{R}^{HW\times(C/M)}$ represent the features within each head after the $Split(\cdot)$ operation for partitioning along the embedding dimension. $A_t^{l(m)} \in \mathbb{R}^{2\times HW}$ and $O_t^{l(m)} \in \mathbb{R}^{2\times(C/M)}$ respectively refer to the attention maps and the text features updated through the image feature within each head. After concatenating all features along the embedding dimension using the $Concat(\cdot)$ operation, a *PostProj* layer with weight matrix $W_t^o \in \mathbb{R}^{C\times d_I}$ is employed to obtain the final updated text embedding $O_t^l \in \mathbb{R}^{2\times d_I}$ from F_t. Then, the updated anomaly map is calculated as:

$$M_2^l = softmax(Up(\widetilde{Z}_s^l \widetilde{O}_t^{lT})/\tau_2), l = 1, 2, \cdots B \tag{9}$$

where τ_2 is a temperature coefficient set as a learnable parameter.

To visually validate the effectiveness of the Post-VCP module, we show the attention maps $A_t^{l(m)}$ under different heads corresponding to normal and abnormal text embeddings. These maps reveal that abnormal text embeddings concentrate more on defective regions of the image compared to normal text embeddings. This clear differentiation stems from employing fine-grained visual contexts in the Post-VCP module to update text embeddings from F_t to O_t^l (Fig. 4).

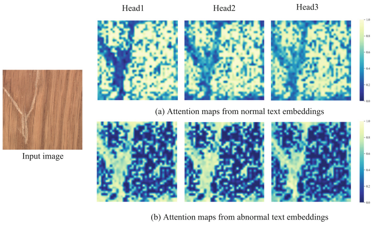

Fig. 4. The visualization result of the attention maps from the Post-VCP module.

3.4 Training and Inference

Loss Function. In this work, we employed focal loss [15] and dice loss [18] to supervise the learning of VCP-CLIP. The total loss function of VCP-CLIP is calculated as:

$$\begin{aligned} L_{\text{total}} = & \underbrace{\sum_l \text{Focal}(M_1^l, S) + \sum_l \text{Dice}(M_1^l, S)}_{\text{Baseline}} \\ & + \underbrace{\sum_l \text{Focal}(M_2^l, S) + \sum_l \text{Dice}(M_2^l, S)}_{\text{Additional VCP modules}} \end{aligned} \tag{10}$$

where the loss function consists of two components, one for the baseline and the other for additional VCP module. M_1^l and $M_2^l, l = 1, 2, \cdots B$ are anomaly maps generated from the two branches mentioned above. $S \in \mathbb{R}^{h\times w}$ is the ground truth corresponding to the input image.

Inference. The ultimate anomaly maps come from different layers of the image encoder by summation. The anomaly maps generated from the two branches are represented as M_1 and M_2. To further enhance the ZSAS capability, we introduced a weighted fusion policy to generate the final anomaly map, $M_a = (1 - \alpha)M_1 + \alpha M_2$, where $\alpha \in [0, 1]$ is a fusion weight designed as a hyperparameter to balance the importance of different anomaly maps.

Table 1. Comparison with existing state-of-the-art methods. The (a, b, c) represents the pixel-level AUROC (%), PRO (%) and AP (%), respectively. The methods denoted by † are training-free, while the others are training-required.

Datasets	WinCLIP † [10]	AnVoL † [8]	CoCoOp [30]	AnomalyGPT [9]	APRIL-GAN [5]	Baseline(ours)	VCP-CLIP(ours)
MVTec-AD	(85.1, 64.6, 18.2)	(90.6, 77.8, 28.1)	(88.2, 83.2, 40.4)	(79.5, 45.9, 23.7)	(87.6, 44.0, 40.8)	(89.2, 85.8, 45.2)	(**92.0**, **87.3**, **49.4**)
VisA	(79.6, 56.8, 5.4)	(91.4, 75.0, 12.7)	(94.9, 88.0, 24.8)	(90.3, 61.5, 13.3)	(94.2, 86.8, 25.7)	(95.5, 89.6, 27.3)	(**95.7**, **90.7**, **30.1**)
BSD	(87.7, 56.8, 4.4)	(96.3, 72.6, 13.3)	(98.7, 85.3, 55.5)	(87.8, 54.0, 37.9)	(98.8, 61.6, 59.7)	(99.1, 86.4, 58.5)	(**99.3**, **87.0**, **70.2**)
GC	(71.9, 44.2, 8.6)	(92.1, 66.5, 14.1)	(96.1, 81.6, 41.5)	(60.0, 11.6, 2.3)	(94.0, 21.5, 34.4)	(97.5, 81.2, 39.6)	(**97.8**, **83.8**, **42.6**)
KSDD2	(89.4, 65.9, 17.5)	(95.9, 80.4, 33.9)	(96.1, 90.9, 69.6)	(91.5, 61.9, 29.7)	(97.5, 49.6, 67.2)	(99.4, 95.4, 71.6)	(**99.5**, **98.0**, **75.2**)
MSD	(47.0, 41.7, 1.5)	(95.0, 68.6, 9.4)	(96.1, 82.3, 27.0)	(67.9, 22.7, 1.8)	(98.1, 36.8, 36.0)	(98.5, 91.0, 54.9)	(**99.0**, **91.1**, **61.0**)
Road	(78.1, 37.9, 11.0)	(85.8, 39.9, 18.3)	(91.0, 56.0, 29.4)	(67.6, 15.5, 9.2)	(89.0, 6.1, 30.4)	(92.7, 62.9, 30.2)	(**93.6**, **66.4**, **32.1**)
RSDD	(91.4, 63.6, 3.7)	(94.7, 75.5, 3.5)	(99.1, 94.4, 37.4)	(93.2, 58.4, 16.0)	(99.1, 62.9, 35.9)	(99.3, 95.9, 35.0)	(**99.5**, **97.5**, **44.1**)
BTech	(63.2, 22.8, 11.4)	(85.6, 45.4, 32.1)	(90.8, 70.1, 44.4)	(75.9, 29.3, 17.6)	(90.8, 18.8, 43.6)	(91.2, 68.9, 43.7)	(**94.1**, **74.6**, **51.4**)
DAGM	(75.1, 43.1, 3.2)	(83.4, 64.7, 10.7)	(98.0, 94.7, 42.9)	(81.9, 35.7, 4.7)	(99.0, 44.1, 50.5)	(99.1, 97.2, 48.9)	(**99.4**, **98.3**, **52.0**)

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. To assess the performance of the model, ten real industrial anomaly segmentation datasets are used, including MVTec-AD [2], VisA [33], BSD [24], GC [17], KSDD2 [3], MSD [29], Road [25], RSDD [28], BTech [19], DAGM [20]. Since the products in VisA do not overlap with those in other datasets, we use VisA as the training dataset for evaluation on other datasets. For VisA itself, we assess it after training on MVTec-AD. Please refer to the Appendix C for more details. To ensure a fair comparison, pixel-level AUROC (Area Under the Receiver Operating Characteristic), PRO (Per-Region Overlap), and AP (Average Precision) are employed as the evaluation metrics, following the recent works [5,6].

Implementation Details. In the experiments, we adopt the CLIP model with ViT-L-14-336 pretrained by OpenAI [22] by default. Specifically, we set the number of layers B for extracting patch-level features to 4. Since the image encoder comprises 24 transformer layers, we evenly extract image features from layers {6, 12, 18, 24}. All images are resized to a resolution of 518×518, and then fed into the image encoder. The length of the learnable category vectors r and the length of the learnable text embeddings n in each text encoder layer are set to 2 and 1, respectively, by default. The number of attention heads M in the Post-VCP module is set to 8. The fusion weight α for different anomaly maps is set to 0.75 as the default value. The Adam optimizer [12] with an initial learning rate of 4e-5 is used, and the model is trained for continuous 10 epochs with a batch size of 32. All experiments are conducted on a single NVIDIA GeForce RTX 3090. We conducted three runs using different random seeds and then averaged the results. More details can be found in Appendix A.

4.2 Comparison with the State-of-the-Art

Two kinds of state-of-the-art approaches are used to compare with ours: training-free approaches and training-required approaches. The training-free approaches include WinCLIP [10] and AnVoL [8], which do not require auxiliary datasets for fine-tuning the model but necessitate more complex manual prompt designs and inference processes. The training-required approaches comprise CoCoOp [30], AnomalyGPT [9] and APRIL-GAN [5], which adhere to the protocol of training on the seen products and testing on the unseen products.

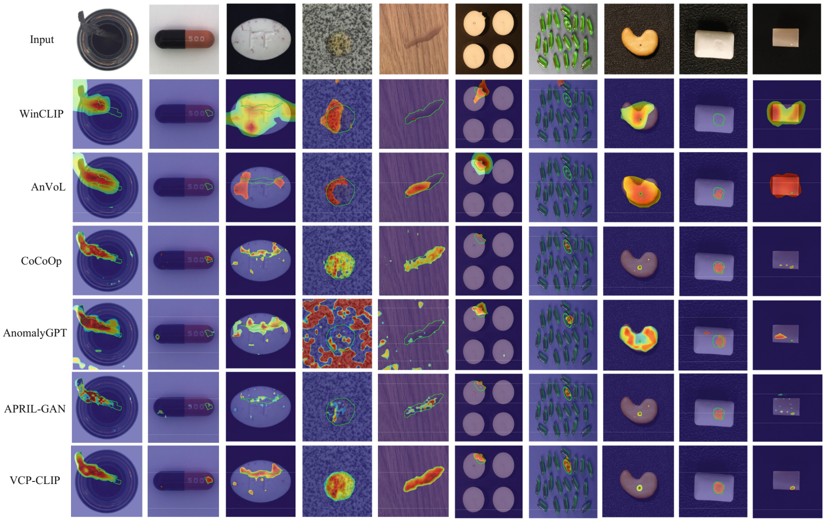

Fig. 5. Qualitative segmentation results. The first five columns use images from the MVTec-AD dataset, and the last five are from the VisA dataset.

Quantitative Comparison. Table 1 shows the quantitative performance comparison with other state-of-the-art methods on ZSAS. The best results are shown in bold, and the second best results are underlined. It can be observed that the proposed VCP-CLIP outperforms all other methods across all metrics, particularly in terms of AP. Due to the tiny anomaly regions on the Visa dataset, its anomaly segmentation is more challenging. However, VCP-CLIP still maintains its advantage compared to other methods. Notably, it achieves state-of-the-art results on VisA dataset, with AUROC score of 95.7%, PRO score of 90.7% and AP score of 30.1%. It is noteworthy that our baseline approach has already achieved nearly superior performance compared to existing methods such as CoCoOp, which similarly introduces global image information in the text prompts. This is because our method simultaneously adjusts text embeddings using fine-grained image features.

Qualitative Comparison. For a more intuitive understanding of the results, we visualized the anomaly segmentation results of our VCP-CLIP alongside another five methods: WinCLIP [10], AnVoL [8], CoCoOp [30], AnomalyGPT [9], and APRIL-GAN [5] on the MVTec-AD and VisA datasets in Fig. 5. The visualization results clearly indicate that the compared approaches have a tendency to generate incomplete or false-positive results, which can negatively impact the performance of anomaly localization. In contrast, our VCP-CLIP effectively mitigates these issues, providing a more accurate and reliable approach to ZSAS. More quantitative and qualitative comparisons are provided in the Appendix D.

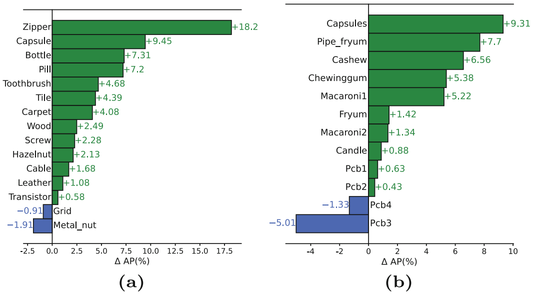

Fig. 6. The AP improvement of VCP-CLIP over the baseline for each product. (a) MVTec-AD (b) VisA

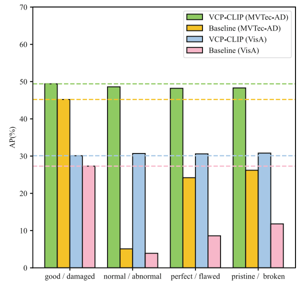

Fig. 7. Performance comparison for different prompts during training and testing.

4.3 Unified Text Prompting Vs. Visual Context Prompting

Same Prompts During Training and Testing. To better validate the effectiveness of VCP-CLIP, we compared it with the proposed baseline on MVTec-AD and VisA. Figure 6 illustrates the AP improvement of VCP-CLIP over the baseline for each product. In specific, VCP-CLIP demonstrates varying degrees of improvement among 13 out of the 15 products and 10 out of the 12 products on the MVTec-AD and VisA datasets, respectively. This affirms the robust generalization capability of VCP-CLIP, which is attributed to both the global visual context in Pre-VCP and the fine-grained local visual context in Post-VCP.

Different Prompts During Training and Testing. To validate the robustness of VCP-CLIP during the test process with different text prompts, we employed text prompts different from those used during training on the MVTec-AD and VisA datasets. Specifically, during training, the default state words "good/damaged" were used. During testing, we reported the metric AP when the state words were respectively "normal/abnormal", "perfect/flawed", and "pristine/broken". As shown in Fig. 7, our baseline performance sharply declined on two datasets, while the performance of VCP-CLIP remained relatively stable. This indicates that after incorporating VCP, the model can adaptively adjust the output text embeddings based on input images, thereby avoiding dependence on the specific text prompts used during training.

Table 2. Ablation on different components.

DTP	VCP		AUROC	PRO	AP
	Pre	Post			
√		√	91.4	86.6	47.5
√	√		90.4	86.0	46.1
	√	√	91.7	86.7	48.2
√			89.2	85.8	45.2
√	√	√	**92.0**	**87.3**	**49.4**

Table 3. Ablation on ensemble of different patch-level image layers.

Image layers	AUROC	PRO	AP
{6}	79.6	65.6	22.5
{12}	91.4	84.8	44.1
{18}	91.2	84.4	44.5
{24}	90.1	80.2	38.2
{6, 12}	91.1	85.9	46.2
{6, 12, 18}	91.8	87.1	49.2
{6, 12, 18, 24}	**92.0**	**87.3**	**49.4**

Table 4. Ablation on different template and state words in text prompts.

Template	State words	AUROC	PRO	AP
this is a [state] photo of [$z(x, v)$]	perfect/flawed	91.9	**87.3**	48.5
	normal/abnormal	90.1	86.1	48.7
	flawless/imperfect	91.3	87.0	49.0
	pristine/broken	91.2	86.5	48.8
	good/damaged	91.5	86.9	49.1
a photo of a [state] [$z(x, v)$]	perfect/flawed	91.8	87.2	48.8
	normal/abnormal	91.7	86.6	48.7
	flawless/imperfect	**92.1**	87.2	49.2
	pristine/broken	92.0	87.1	49.3
	good/damaged	92.0	**87.3**	**49.4**

4.4 Ablation Studies

Influence of Different Components. To assess the impact of different components on VCP, experiments were conducted on MVTec-AD. Results in Table 2 indicate performance when using DTP, Pre-VCP or Post-VCP individually. Notably, the optimal performance for VCP is achieved when all combined. It can been seen that the performance decline is more pronounced after removing Post-VCP compared to Pre-VCP. We also attempted to remove the learnable text embeddings from each layer of the text encoder (without DTP), which resulted in a decrease of 0.3% in AUROC, 0.6% in PRO, and 1.2% in AP. This is because the original text space cannot directly comprehend the global features of images, while DTP ensures deep fine-tuning of each text encoder layer, thereby fostering mutual understanding and fusion of different modalities.

Influence of Ensemble of Different Patch-Level Image Layers. In Table 3, we explore the impact of patch-level features from different image encoder layers on VCP-CLIP's performance. The experiments were conducted on the MVTec-AD dataset. An intuitive observation is that image features from intermediate layers (i.e. the 12th and 18th layers), contribute more to the final segmentation result. Image features from lower layers (i.e., the 6th layer) are too low-level, while those from higher layers (i.e., the 24th layer) are overly abstract. Their effectiveness is not as pronounced as those from intermediate layers. However, We observed a positive correlation between incorporated layer numbers and improved segmentation results. To maintain high performance, we adopted all patch-level features from {6, 12, 18, 24} layers in VCP-CLIP.

Ablation on Text Prompt Design. As demonstrated in Table 4, we considered two commonly used text prompt templates and explored the impact of different prompting state words in the proposed VCP-CLIP on MVTec-AD. Specifically, we designed the following two text prompt templates: 1) this is a [state] photo of [$z(x, v)$]; 2) a photo of a [state] [$z(x, v)$]. The state words (e.g. “perfect/flawed”) are respectively inserted into the template to generate normal and abnormal text prompts. It can be observed that for the same template with different state words, our VCP-CLIP model consistently maintains similar performance, validating the robustness towards the state words. Furthermore, the second type of template, default employed in VCP-CLIP, outperforms the first type overall, which may be attributed to the repeated usage of similar template during the pre-training process of the vanilla CLIP.

Table 5. Ablation on different input resolutions upon VCP-CLIP.

Input resolution	AUROC	PRO	AP	Time (ms)
224^2	91	84.6	38.1	**101.6**
336^2	90.7	87.6	44.9	104.5
518^2	**92.0**	**87.3**	**49.4**	127.9
546^2	91.2	85.3	45.1	134.9
798^2	90.8	85	38.4	265.3

Table 6. Ablation on different Pre-trained backbone upon VCP-CLIP.

Pretrained backbone	AUROC	PRO	AP	Time (ms)
ViT-B-16-224	89.4	82.2	37.9	**84.4**
ViT-L-14-224	91.9	85.7	43.3	105.1
ViT-L-14-336	**92.0**	**87.3**	**49.4**	127.9

Ablation on Different Pretrained Models and Resolutions. In Table 5 and Table 6, we conducted a comprehensive analysis of the impact of varying input image resolution and pre-trained backbone on MVTec-AD. The former is tested using ViT-L-14-336, while the latter reports the optimal performance under different backbones pre-trained by OpenAI. The inference time was simultaneously tested for a single image (average of 200 images). We observe that a moderate increase in input image resolution contributes to more precise segmentation (higher AP). However, deviations from the original pre-training resolution (336^2 to 798^2), leading to model degradation. This outcome can be attributed to the model deviating from the original image space. The result in Table 6 shows that our VCP-CLIP achieves the optimal segmentation performance in ViT-L-14-336. Therefore, we have chosen it as the default backbone.

5 Conclusion

In this paper, we present VCP-CLIP, a novel zero-shot anomaly segmentation (ZSAS) method achieved through the integration of visual context prompting

(VCP). The core methodology involves incorporating richer visual knowledge into the textual space and cross-modal interaction between textual and visual features. Specifically, a Pre-VCP and a Post-VCP module are designed to respectively introduce global and fine-grained image features into the textual space. With this design, our model can directly segment anomalies in novel products without any prior knowledge. Extensive experiments conducted on 10 real-world industrial anomaly segmentation datasets showcase VCP-CLIP's state-of-the-art performance in ZSAS.

Acknowledgments. This work was supported in part by the National Science and Technology Major Project of China under Grant 2022ZD0119402; in part by the National Natural Science Foundation of China under Grant No. 62373350 and U21A20482; in part by the Youth Innovation Promotion Association CAS (2023145) and (2020139); in part by the Beijing Municipal Natural Science Foundation, China, under Grant L243018; in part by the Beijing Nova Program (20240484687).

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
3. Božič, J., Tabernik, D., Skočaj, D.: Mixed supervision for surface-defect detection: from weakly to fully supervised learning. Comput. Ind. **129**, 103459 (2021)
4. Cao, Y., et al.: Segment any anomaly without training via hybrid prompt regularization. arXiv preprint arXiv:2305.10724 (2023)
5. Chen, X., Han, Y., Zhang, J.: A zero-/few-shot anomaly classification and segmentation method for CVPR 2023 VAND workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. arXiv preprint arXiv:2305.17382 (2023)
6. Chen, X., et al.: Clip-ad: a language-guided staged dual-path model for zero-shot anomaly detection. arXiv preprint arXiv:2311.00453 (2023)
7. Chiang, W.L., et al.: Vicuna: an open-source chatbot impressing GPT-4 with 90%* chatgpt quality (2023). https://lmsys.org/blog/2023-03-30-vicuna/
8. Deng, H., Zhang, Z., Bao, J., Li, X.: AnoVL: adapting vision-language models for unified zero-shot anomaly localization. arXiv preprint arXiv:2308.15939 (2023)
9. Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., Wang, J.: Anomalygpt: detecting industrial anomalies using large vision-language models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 1932–1940 (2024)
10. Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A., Dabeer, O.: Winclip: zero-/few-shot anomaly classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19606–19616 (2023)
11. Jia, M., et al.: Visual prompt tuning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13693, pp. 709–727. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19827-4_41
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

13. Kirillov, A., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026 (2023)
14. Li, S., Cao, J., Ye, P., Ding, Y., Tu, C., Chen, T.: Clipsam: clip and sam collaboration for zero-shot anomaly segmentation. arXiv preprint arXiv:2401.12665 (2024)
15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
16. Liu, S., et al.: Grounding dino: marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
17. Lv, X., Duan, F., Jiang, J.J., Fu, X., Gan, L.: Deep metallic surface defect detection: the new benchmark and detection network. Sensors **20**(6), 1562 (2020)
18. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571. IEEE (2016)
19. Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L.: VT-ADL: a vision transformer network for image anomaly detection and localization. In: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), pp. 1–6. IEEE (2021)
20. für Mustererkennung, D.A.: Weakly supervised learning for industrial optical inspection (2007)
21. Ouyang, L., et al.: Training language models to follow instructions with human feedback. Adv. Neural. Inf. Process. Syst. **35**, 27730–27744 (2022)
22. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
23. Rao, Y., et al.: Denseclip: language-guided dense prediction with context-aware prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18082–18091 (2022)
24. Schlagenhauf, T., Landwehr, M.: Industrial machine tool component surface defect dataset. Data Brief **39**, 107643 (2021)
25. Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z.: Automatic road crack detection using random structured forests. IEEE Trans. Intell. Transp. Syst. **17**(12), 3434–3445 (2016)
26. Tao, X., Gong, X., Zhang, X., Yan, S., Adak, C.: Deep learning for unsupervised anomaly localization in industrial images: a survey. IEEE Trans. Instrum. Meas. **71**, 1–21 (2022)
27. Touvron, H., et al.: Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
28. Yu, H., et al.: A coarse-to-fine model for rail surface defect detection. IEEE Trans. Instrum. Meas. **68**(3), 656–666 (2018)
29. Zhang, J., Ding, R., Ban, M., Guo, T.: Fdsnet: an accurate real-time surface defect segmentation network. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3803–3807. IEEE (2022)
30. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16816–16825 (2022)
31. Zhou, Q., Pang, G., Tian, Y., He, S., Chen, J.: Anomalyclip: object-agnostic prompt learning for zero-shot anomaly detection. In: The Twelfth International Conference on Learning Representations (2023)

32. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: towards adapting clip for zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11175–11185 (2023)
33. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13690, pp. 392–408. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20056-4_23