CoGenSAM: Codebook-Interactive Generative Labeling for Adapting SAM to Crack Segmentation

Zhuangzhuang Chen, Nuo Chen, Dachong Li, Zhiliang Lin, Xingyu Feng, Yifan Zhang, Jie Chen*, Jianqiang Li*

College of Artificial Intelligence, Shenzhen University, Shenzhen, China
{chenzhuangzhuang2016, 2022150064, 2350273001, linzhiliang2022,fengxingyu2017,zhangyifan2020}@email.szu.edu.cn, chenjie@szu.edu.cn, lijq@szu.edu.cn

Abstract

The goal of this work is to adapt Segment Anything Models (SAM) into crack segmentation tasks via automatic label generation, thus eliminating manual annotation cost. In this regard, an intuitive approach is to extract edges of crack samples and generate labels via the dilation and erosion processes for fine-tuning SAM. However, this simple solution cannot guarantee the quality of generated labels, as crack regions will be corrupted due to the imperfect edge detection. To this end, this paper proposes CoGenSAM, a novel Codebook-interactive Generative Labeling framework that enables an annotation-free SAM fine-tuning. To achieve this, in the first stage, we pre-train a vector-quantized variational auto-encoder (VQVAE) by reconstructing the synthesized crack-like structures for learning crack-aware priors within the codebook. In the second stage, these priors help another VQVAE serve as the restoration model to restore the randomly corrupted structures into uncorrupted ones. Specifically, we propose the crack-aware contrastive-interaction to maximize the mutual information with the above priors via codebook interaction. Then, high-quality labels can be generated by restoring corrupted labels from edge detection, contributing to an annotation-free SAM fine-tuning. We collect a new dataset, Bridge2025, to address the limited availability of related bridge-oriented benchmarks. Experiments show that our performance is close to fully-supervised methods.

Introduction

Crack segmentation plays a significant role in structural health monitoring for maintaining the structural health and reliability of many infrastructures, including concrete pavement (Lei, Zhong, and Wang 2024; Lei et al. 2025), bridges (Jiang et al. 2020; Inoue and Nagayoshi 2023), and nuclear power plants (Chen et al. 2025c,b). Manual inspection of crack regions by human experts is time-consuming and suffers from variability across different inter-observers. Thus, the advancement of automated crack segmentation methods has received extensive attention from both the industry and academia (Chen et al. 2022, 2024).

The significant advancements lie in the rise of deep learning methods (Xu, Yang, and Zhang 2023a,b; Li et al. 2025a). Concretely, convolutional neural networks (CNN) are the first to prove their potential (Chen et al. 2023). Later, Crackformer (Liu et al. 2021a) successfully leverages Vision Transformer (ViT) to capture long-range interactions for crack segmentation. Later, SCSegamba (Liu et al. 2025) proposes a lightweight structure-aware vision mamba network to achieve efficient crack segmentation. Motivated by the great success of the Segment Anything Model (SAM) (Kirillov et al. 2023), a series of works make great attempts to fine-tune SAM via a low-rank mechanism in a supervised manner (Ge et al. 2024; Wan et al. 2025). For example, FlexiCrackNet (Wan et al. 2025) modifies the SAM architecture by adding an information-interaction gated attention mechanism. Meanwhile, (Rostami, Chen, and Hosseini 2025) focuses on improving the fine-tuning strategy and proposes an automatic selection mechanism. Despite these works demonstrating the effectiveness of SAM on crack segmentation tasks compared with CNN-based, ViT-based, and Mamba-based methods, they still heavily rely on pixel-level crack annotations, which are difficult to acquire due to the high demand for experts and labeling cost.

To avoid labeling cost, a simple idea is to directly use

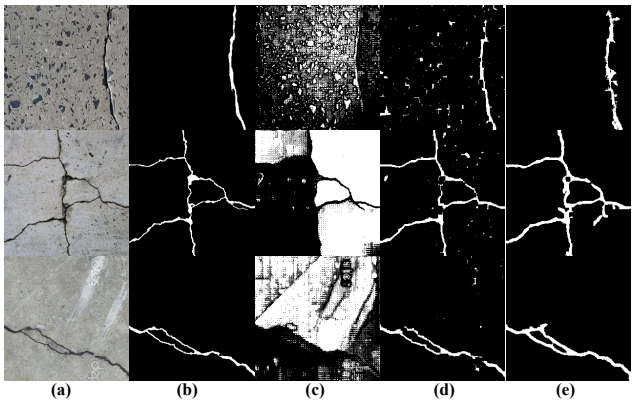


Figure 1: (a) Crack images. (b) Real labels. (c) Segmentation results of SAM. (d) Generated labels via edge detection: extracting edge images of crack samples by the Sobel operator, and obtaining labels via the dilation and erosion process. (e) Our generated labels: herein, we propose the codebook-interactive generative labeling to improve the quality of generated labels from the edge detection process.

*Corresponding Author

SAM to get crack labels and then serve as pseudo labels by following the existing methods (Li et al. 2024; Zhang et al. 2025). However, such approaches fail to produce reliable crack labels, as they either over-segment cracks into numerous fractions or under-segment the crack regions by misclassifying many of them as the background, see Fig. 1 (c). Considering this, another intuitive approach is to extract edges of crack samples, and then obtain labels via the dilation and erosion process. Then, these generated labels can act as real labels to help SAM adapt to crack segmentation. Interestingly, Fig. 1 (d) shows that although generated labels can reflect crack regions. However, they still involve corrupted regions, due to imperfect edge detection.

In this paper, we propose a novel Codebook-interactive Generative Labeling framework, called CoGenSAM, that automatically generates labels for fine-tuning SAM. ***Our key idea lies in that we learn crack-aware priors within the codebook by training on synthesized crack-like structures, which can provide guidance for a restoration model to restore corrupted regions via codebook-level interaction***. In return, the trained restoration model can help to improve the quality of generated labels from imperfect edge detection. To achieve this, in the first stage, a VQVAE is pre-trained on the synthesized crack-like structures to enforce a codebook to learn crack-aware priors. In the second stage, we simulate corrupted labels by generating multi-scale square masks and randomly selecting them to corrupt the synthesized crack-like structures by removing certain regions. Then, we propose the crack-aware contrastive-interaction to enforce another VQVAE as the restoration model to restore corrupted regions by maximizing the crack-aware information via contrastive interaction with the pre-trained codebook. Afterward, high-quality labels can be generated by restoring the corrupted labels from the edge detection process, see Fig. 1 (e), thus contributing to a reliable SAM fine-tuning.

Our contributions can be summarized as four-fold:

- To the best of our knowledge, we are the first one that automatically generates crack labels for adapting SAM to crack segmentation without any manual annotation cost. Notably, our annotation-free SAM achieves satisfactory performance even compared with supervised methods.
- Propose the codebook-interactive generative labeling by starting from a corrupted-to-uncorrupted labels restoration perspective, which aims to learn crack-aware priors from the synthesized crack-like structures and then provide guidance for the restoration model.
- Propose the crack-aware contrastive-interaction that enables explicitly cross-codebook learning via a contrastive interaction that maximizes the mutual information between the restoration and reconstruction model with a theoretical guarantee.
- A new crack dataset, named Bridge2025, is collected from real-world bridges, to promote research on bridge-oriented crack segmentation tasks by serving as a new bridge-oriented crack segmentation benchmark.

Related Work

Annotation-free Crack Segmentation. Early unsupervised methods act in a clustering manner (Li et al. 2021). Later, inspired by anomaly detection, Zhang et al. leverage an autoregressive model to learn the distribution of these discrete representations from non-crack samples (Zhang et al. 2024). To improve the restoration of the non-crack region, UP-CrackNet (Ma, Fan, and Xie 2024) designs a generative adversarial network to restore the corrupted regions. However, the existing VQVAE-based unsupervised segmentation methods (Cheng, Qu, and Lee 2024) still require human efforts to filter out those crack samples, as these methods assume that crack areas cannot be reconstructed by training with only normal samples (Lei et al. 2025). Meanwhile, self-supervised methods aim to leverage large-scale unsupervised data to perform pre-training by using contrastive learning losses (Liu et al. 2021b). Although (Kim, Oh, and Ye 2022) and (Ma et al. 2021) achieve self-supervised segmentation via adversarial learning and fractals, they require background-only images as input for generating synthesis data. This limits its real-world applications. To this end, FreeCOS (Shi et al. 2023) explicitly encodes geometric and photometric characteristics, as well as some observed varieties of curvilinear objects in the target application.

Herein, our method is distinct from existing FreeCOS and VQVAE-based methods in two-fold: (1) Instead of using synthetic data for training segmentation models, we focus on automatically generating labels for real samples via our corrupted-to-uncorrupted labels restoration. (2) Instead of learning a crack-free codebook from crack-free images, we learn crack-aware priors from the synthesized crack-like structures, and then they can guide the restoration model via codebook interaction for generating high-quality labels for an annotation-free yet reliable SAM fine-tuning.

Vision Foundation Models. Foundation models are pre-trained, large-scale models that allow for fast customization through fine-tuning (Xu, Ye, and Su 2025). For example, (Ye et al. 2024) and (Ge et al. 2024) utilize low-rank adaptation (LoRA) to apply SAM for crack segmentation. (Wan et al. 2025) proposes FlexiCrackNet by integrating an information-interaction gated attention mechanism. (Wang, He, and Yu 2024) and (Rakshitha et al. 2024) aim to train an object detector to serve as a prompter for enhancing SAM's segment capacity. To improve the fine-tuning strategy, (Rostami, Chen, and Hosseini 2025) propose a selective fine-tuning strategy. (Jiang et al. 2024) leverages distribution-aware domain-specific semantic knowledge to guide the learning process. CrossDiff (Shi et al. 2025) aim to generating synthetic crack images from segmentation masks.

Unlike existing works that generate synthetic images for training segmentation models, our key difference lies in two-fold: (1) We aim to achieve annotation-free SAM fine-tuning via generated labels without any real-world crack labels. (2) Our work starts from a corrupted-to-uncorrupted labels restoration perspective, which aims to learn crack-aware priors from the synthesized crack-like structures and then provide guidance for the restoration model.

Contrastive Learning. Contrastive learning has proved its advantage in learning feature representations for various

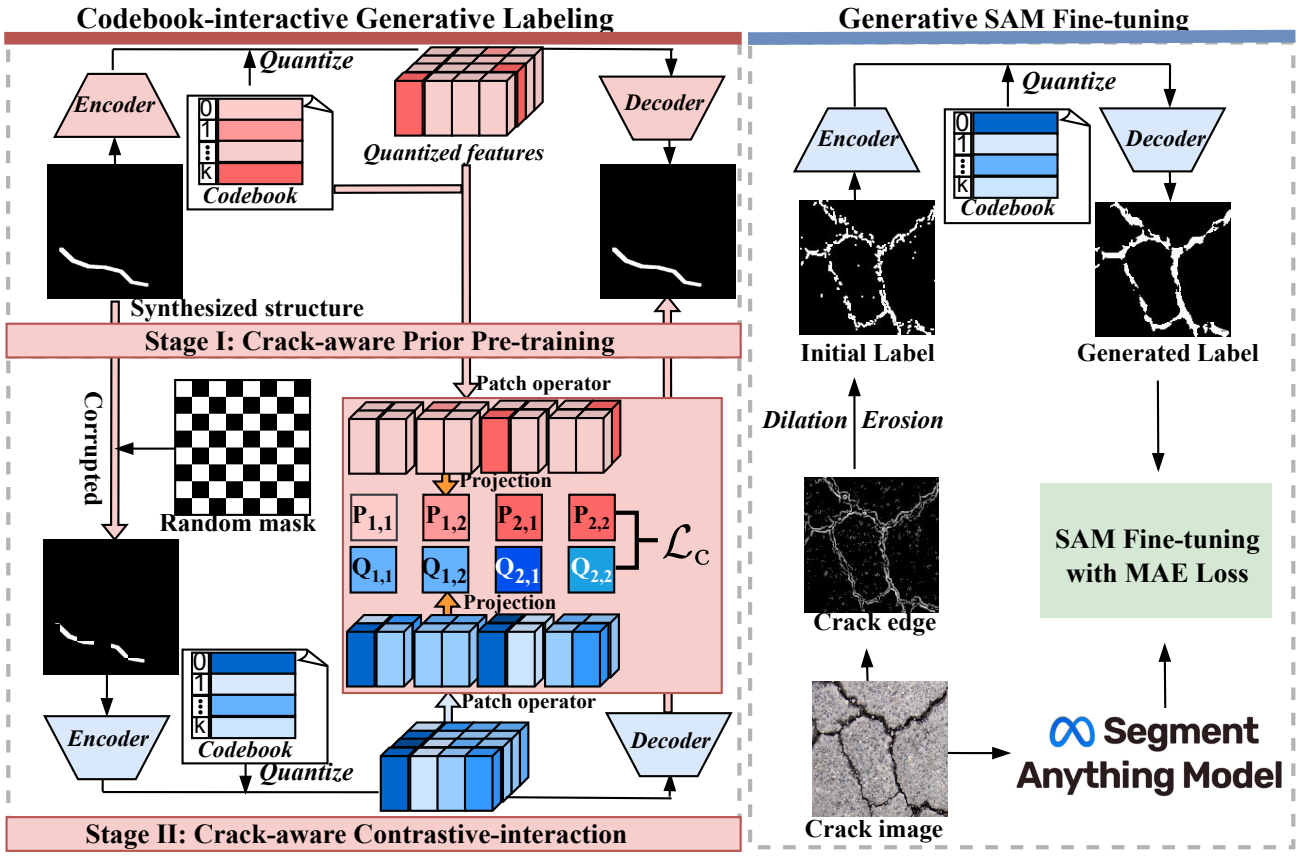


Figure 2: The overview of CoGenSAM. Our codebook-interactive generative labeling employs a VQVAE pre-trained on the synthesized crack-like structure images in Stage I, to learn a codebook for providing informative guidance for the next stage. Then, we utilize another VQVAE that takes randomly corrupted synthesized crack-like structures as input, to restore the uncorrupted one via our proposed crack-aware contrastive-interaction. Finally, an annotation-free yet reliable SAM fine-tuning process can be achieved with automatically generated labels by restoring the generated labels from the edge detection.

tasks (Li et al. 2025b). Jeong et al. (Jeong et al. 2021) enhance the discrimination ability of memory with the feature contrastive loss. However, it involves a huge computation burden as the memory bank gets outdated quickly in a few passes. Later, (Gou et al. 2023) design multi-feature contrastive learning by constructing a patch-wise contrastive loss using the feature information. Recently, Zhao et al. (Zhao, Cai, and Yuan 2024) propose the dual contrastive regularization.

Notably, different from the above methods that aim to learn class-relevant features, our fundamental advancement lies in that our crack-aware contrastive-interaction constructs contrastive pairs between quantized features from the codebook, which enables explicit cross-codebook learning via contrastive interaction with a theoretical guarantee.

Method

Framework Overview

In this paper, we propose Codebook-interactive Generative Labeling (CoGenSAM), to achieve annotation-free SAM fine-tuning for crack segmentation. As illustrated in Fig. 2, in the first stage, we pre-train a VQVAE by reconstructing the synthesized crack-like structures, which allow the codebook to learn crack-aware priors. Then, in the second stage, we propose the crack-aware contrastive-interaction to train another VQVAE as a restoration model to restore corrupted regions via contrastive interaction with a pre-trained codebook. Finally, we enable generative SAM fine-tuning by automatically generating high-quality labels via restoring corrupted labels from the edge detection. In the following section, we will discuss more details of CoGenSAM.

Codebook-Interactive Generative Labeling

Notably, existing methods (Seibold et al. 2022; Chen et al. 2025a) obtain generated labels via a well-trained segmentation model, which is difficult to acquire due to the substantial effort required to develop well-annotated crack segmentation datasets. In contrast, we aim to learn crack-aware priors from the synthesized crack-like structures, and then provide crack-aware contrastive-interaction for the restoration model to improve the quality of generated labels from the imperfect edge detection process.

Stage I: Crack-aware Prior Pre-training: Since frac-

tals show similar patterns with cracks and can be rendered by mathematical formulas, herein, we propose crack-aware prior pre-training that leverages parametric Fractal L-systems (Zamir 2001) to generate fractal tree structures by following physiological rules of cracks with bifurcations:

$$rule : \mathcal{A} \rightarrow \mathcal{A}[-\mathcal{A}][+\mathcal{A}], \tag{1}$$

where $rule$ denotes the production rule and $\mathcal{A}$ denotes an example of an axiom, e.g., a line of unit length in the horizontal direction. "[" and "]" denote the departure and return to a branch point. The symbols "-" and "+" denote a certain rotation angle in anti-clockwise and clockwise directions. Please see implement details in Appendix A.

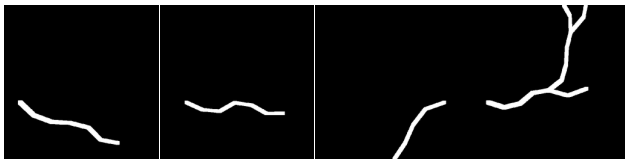


Figure 3: The synthesized crack-like structure images

Given the synthesized crack-like structure image X_{ori} based on the basic Fractal system (see Fig. 3), the convolutional encoder $\mathbb{M}_e^p$ maps the input image X_{ori} to the latent feature vector $\mathbf{z_e^p}$ as following :

$$\mathbf{z_e^p} = \mathbb{M}_e^p(X_{\text{ori}}), \text{ and } \mathbf{z_e^p} \in \mathbb{R}^{W \times H \times C}, \tag{2}$$

where W, H, and C indicate the feature's width, height, and the number of channels, respectively. Next, we built a codebook $\mathbb{P} \in \mathbb{R}^{N \times c}$ that contains N discrete latent vectors to model the distributions of synthesized crack-like structure images, where N is the number of entries in the codebook. And, c is the dimension of each entry, which is equal to C. Then, we can discretize the distribution of the latent feature vector to get the quantized features $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}$ by the following vector quantization $\text{VQ}_\mathbb{P}(\cdot)$:

$$\text{VQ}_\mathbb{P}(\mathbf{z}) := \text{argmin}_{\mathbf{z}_k \in \mathbb{P}} \left\| \mathbf{z} - \mathbf{z}_k \right\|_2, \tag{3}$$

where $\mathbf{z}_k$ is the k-th entry in the codebook $\mathbb{P}$. Then, each vector in $\mathbf{z_e^p}$ is replaced with its nearest neighbor entry in the codebook $\mathbb{P}$ via $\text{VQ}_\mathbb{P}(\cdot)$, resulting in the corresponding quantized feature vector $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p} \in \mathbb{R}^{W \times H \times c}$:

$$\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p} = \text{VQ}_\mathbb{P}(\mathbf{z_e^p}), \text{ and } \hat{\mathbf{z}}_\mathbf{e}^\mathbf{p} \in \mathbb{R}^{W \times H \times c}. \tag{4}$$

The convolutional decoder $\mathbb{M}_d^p$ reconstructs the quantized vector $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}$ back into the image $\hat{X}_{\text{recon}}^p$ by the Eq. 5.

$$\hat{X}_{\text{recon}}^p = \mathbb{M}_d^p(\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}). \tag{5}$$

Finally, codebook $\mathbb{P}$, encoder $\mathbb{M}_e^p$ and decoder $\mathbb{M}_d^p$ are jointly optimized with the following reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{VQVAE}}^p(\mathbb{P}, \mathbb{M}_e^p, \mathbb{M}_d^p) = \left\| X_{\text{ori}} - \hat{X}_{\text{recon}}^p \right\|_2 + \\ \left\| \mathbf{sg}[\mathbf{z_e^p}] - \hat{\mathbf{z}}_\mathbf{e}^\mathbf{p} \right\|_2 + \left\| \mathbf{sg}[\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}] - \mathbf{z_e^p} \right\|_2,\end{aligned} \tag{6}$$

where $\mathbf{sg}[\cdot]$ indicates the stop gradient operator. The first item in Eq. 6 optimizes the encoder and decoder to enforce the reconstructed image to be close to the original synthesized crack-like structure images. The second item in Eq. 6 provides gradients to the codebook $\mathbb{P}$. To incentivize the encoder $\mathbb{M}_e^p$ to commit to the codebook, a third term is added to update the encoder. In other words, it enforces the latent feature $\mathbf{z_e^p}$ to be close to the nearest neighbor entry in $\mathbb{P}$.

Then, the overall objective $\mathcal{L}_{\text{stage1}}$ is defined as $\mathcal{L}_{\text{VQVAE}}^p(\mathbb{P}, \mathbb{M}_e^p, \mathbb{M}_d^p)$. By exploiting this, we are allowed to obtain a codebook $\mathbb{P}$ that contains crack-aware priors corresponding to those synthesized crack-like structure images. In the following, we discuss how to leverage a pre-trained codebook to promote the corrupted region restoration task.

Stage II: Crack-aware Contrastive-interaction : The success of our annotation-free SAM fine-tuning process lies in the quality of the generated labels. However, due to the imperfect edge detection result, there still exist corrupted crack regions in the generated crack labels via the vanilla process (i.e., edge detection, dilation, and erosion). Considering this, we aim to simulate such effects by randomly corrupting synthesized crack-like structures. We then train a restoration model to restore the corrupted regions. In return, the restoration model can help us to improve the quality of the generated labels from the edge detection process.

Herein, we first divide an image into $\frac{W}{S} \times \frac{H}{S}$ patches, where H and W represent the height and width of input synthesized crack-like structure image X_{ori} respectively, and S denotes the patch size. Then, masks can be generated via a Boolean logic strategy, where pixel values are set to 0 or 1 to indicate the regions that should be removed or retained. Notably, the ratio between the removed and retained regions is $1:1$. Given the uncorrupted X_{ori}, the random corrupted image X_{cor} can be obtained by: $X_{\text{cor}} = X_{\text{ori}} * M$, where M denotes the random generated mask. Now, we train a VQ-VAE that takes the corrupted image X_{cor} as input, and then restores the corresponding uncorrupted image X_{ori}. However, due to the domain gap between uncorrupted and corrupted synthesized crack-like structure images, vanilla VQ-VAE suffers from the model collapse problem, where the model is likely to get stuck in local optima and fails to optimize the codebook effectively. Thus, we propose the crack-aware contrastive-interaction that aims to maximize mutual information with the pre-trained codebook, to encourage the restoration model to explore more new codewords.

More specifically, the restoration model contains encoder $\mathbb{M}_e^r$, codebook $\mathbb{R}$, and decoder $\mathbb{M}_d^r$. We input X_{cor} to encoder $\mathbb{M}_e^r$ and obtain the latent feature vector $\mathbf{z_e^r}$. Similar to Eq. 3, the quantized features $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{r}$ can be obtained via codebook $\mathbb{R}$. Afterward, the restored image $\hat{X}_{\text{recon}}^r$ is obtained via decoder $\mathbb{M}_d^r$. Besides the vanilla VQVAE loss, i.e., $\mathcal{L}_{\text{VQVAE}}^r(\mathbb{R}, \mathbb{M}_e^r, \mathbb{M}_d^r)$ (See Eq. 6), the restoration model adopts our crack-aware contrastive-interaction for joint supervision, as discussed below.

For computation efficiency, our patch operator first divides $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{r}$ and $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}$ into non-overlapping square patch features $\hat{\mathbf{z}}^\mathbf{r}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$ and $\hat{\mathbf{z}}^\mathbf{p}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$, where H and W represent the height and width of input features $\hat{\mathbf{z}}_\mathbf{e}^\mathbf{p}$ respectively, and S denotes the patch size. Moreover, we add two additional projection modules $\phi^r(\cdot)$ and $\phi^p(\cdot)$,

each of which includes a global average pooling layer and a fully connected layer. $\phi^r(\cdot)$ and $\phi^p(\cdot)$ are used to transform the patch features into $\mathbf{P}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$ and $\mathbf{R}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$, respectively. The transformed embeddings are used for contrastive learning. However, previous contrastive learning methods do not consider leveraging crack-aware priors within the codebook. In contrast, we aim to effectively leverage informative quantized features from the pre-trained reconstruction model, to promote codebook learning for corrupted mask to uncorrupted mask restoration tasks. Given an embedding $\mathbf{R}_{(\mathbf{i},\mathbf{j})} \in \mathbf{R}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$, we denote it as an anchor embedding. Then, we construct the positive contrastive embedding $\mathbf{P}_{(\mathbf{i},\mathbf{j})}$ and negative embeddings $\mathbf{P}_{\{(\mathbf{1},\mathbf{1}),(\mathbf{1},\mathbf{2}),\cdots,(\mathbf{m},\mathbf{n}),(\frac{\mathbf{W}}{\mathbf{S}},\frac{\mathbf{H}}{\mathbf{S}})\}}$, where $(\mathbf{m},\mathbf{n}) \neq (i,j)$. Note that, the above feature embeddings are preprocessed by l_2-normalization for numerical stability. Then, the contrastive probability distribution $\mathcal{P}_{\mathbb{R}\to\mathbb{P}}$ between codebook $\mathbb{R}$ and $\mathbb{P}$ can be formulated as:

$$\mathcal{P}_{\mathbb{R}\to\mathbb{P}} = \operatorname{softmax}([\left(\mathbf{R}_{(\mathbf{i},\mathbf{j})} \cdot \mathbf{P}_{(\mathbf{i},\mathbf{j})}/\tau\right), \cdots, \left(\mathbf{R}_{(\mathbf{i},\mathbf{j})} \cdot \mathbf{P}_{(\mathbf{m},\mathbf{n})}/\tau\right), \cdots]) \tag{7}$$

where $\operatorname{softmax}(x_i) = e^{x_i}/\sum_{j=1}^{n} e^{x_j}$, τ is a constant temperature, and $\mathcal{P}_{\mathbb{R}\to\mathbb{P}} \in \mathbb{R}^{\frac{W}{S}\cdot\frac{H}{S}}$. Notably, the quantized features from the pre-trained codebook are supposed to contain crack-like priors, so as to reconstruct crack-like structures. Thus, the restoration model can benefit from the pre-trained codebook by enforcing a large similarity at the same location (i,j). For this purpose, we follow the existing work and adopt cross-entropy loss to let positive pair to have a larger similarity:

$$\mathcal{L}_{\text{c}} = -\log \frac{\exp(\mathbf{R}_{(\mathbf{i},\mathbf{j})} \cdot \mathbf{P}_{(\mathbf{i},\mathbf{j})}/\tau)}{\sum_{m=1,n=1}^{\frac{W}{S},\frac{H}{S}} \exp(\mathbf{R}_{(\mathbf{i},\mathbf{j})} \cdot \mathbf{P}_{(\mathbf{m},\mathbf{n})}/\tau)}. \tag{8}$$

Herein, our Eq. 8 employs contrastive embeddings from the quantized features within two codebook under different tasks. It can model an explicit relationship between two codebooks among different tasks, facilitating mutual information to promote codebook learning of the restoration model. Our ablation studies further show the necessity of crack-aware contrastive interaction. The results show that our restoration model can learn the codebook effectively with the help of an informative codebook from the pre-trained reconstruction model. Notably, we provide an information perspective to prove that minimizing Eq. 8 is equal to maximizing the upper bound of the mutual information $I(\mathbf{P},\mathbf{R})$ between codebook $\mathbb{P}$ and $\mathbb{R}$ as follows:

$$I(\mathbf{P},\mathbf{R}) \geq \log(\frac{W}{S}\cdot\frac{H}{S} - 1) - \mathbb{E}_{(\mathbf{P},\mathbf{R})}\mathcal{L}_{\text{c}} \tag{9}$$

Notably, the mutual information $I(\mathbf{P},\mathbf{R})$ measures the information overlap degree of quantized features between the pre-trained reconstruction model and restoration model, corresponding to codebook $\mathbb{R}$ and $\mathbb{P}$. In other words, the cookbook of the restoration model could gain extra contrastive knowledge from the cookbook of the pre-trained reconstruction model by using Eq. 8. Finally, the overall objective $\mathcal{L}_{\text{stage2}}$ is formulated as:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}^r_{\text{VQVAE}}(\mathbb{R},\mathbb{M}^r_e,\mathbb{M}^r_d) + \lambda \sum_{i=1,j=1}^{\frac{W}{S},\frac{H}{S}} \mathcal{L}_{\text{c}}, \tag{10}$$

where λ is used to balance the above two loss functions.

Generative SAM Fine-tuning: As shown in Fig. 2 (c), we first obtain the gray image of a given crack sample X_{crack}. Then, we leverage the Sobel operator (Vairalkar and Nimbhorkar 2012) to obtain the corresponding edge image X_{edge} from the above gray image and perform dilation and erosion operations in Opencv, shown as follows:

$$X_{\text{cor}} = erosion(dilate(X_{\text{edge}})), \tag{11}$$

where X_{cor} denotes the generated corrupted crack label. $dilate(\cdot)$ and $erosion(\cdot)$ denotes dilation and erosion operations with the kernel size of 8×8. Afterward, we improve the quality of the above corrupted crack label by inputting X_{cor} to the restoration model, and obtain the restored label X_{recon} via vector quantization by using codebook $\mathbb{R}$. Finally, we use the following sigmoid function as the activation function to limit each pixel value to (0,1):

$$Y_{\text{gen}} = \text{Sigmoid}(X_{\text{recon}}) = \frac{1}{1+e^{-X_{\text{recon}}}}. \tag{12}$$

Now, the generated label Y_{gen} can enable an annotation-free SAM fine-tuning process (See more details in Appendix C). Since mean absolute error (MAE) has been identified as a robust loss function for reducing the impact of incorrect labels (Ghosh, Kumar, and Sastry 2017), we adopt MAE loss for the supervision:

$$\mathcal{L}_{\text{CoGenSAM}} = \text{MAE}(\text{SAM}(X_{\text{crack}}), Y_{\text{gen}}). \tag{13}$$

Experiment

Datasets

Since we focus on crack segmentation, we adopt the following commonly used DeepCrack and Crack500 datasets for evaluation. Meanwhile, to address the limited availability of related benchmarks, we propose the Bridge2025 dataset to provide bridge-oriented crack segmentation benchmarks. The DeepCrack dataset (Liu et al. 2019) contains 537 RGB color images with manually annotated segmentations. The Crack500 dataset (Yang et al. 2019) contains 3368 crack samples and is more challenging with various shapes and cluttered backgrounds. Moreover, the widths and shapes of cracks in Crack500 vary over a large range, making crack segmentation challenging. Our Bridge2025 dataset is collected from real-world bridges via a handheld camera, comprising 500 images with the size of 4096×3072. The collected images contain noise, such as uneven illumination and blurred backgrounds. To augment the dataset without compromising its resolution, we slice the captured images into image patches of 256×256 pixels, composing a final dataset with 1002 samples with manually annotated segmentations. We provide some examples in Appendix D. Herein, we consider a more challenging setting that splits 100 samples for training and validation, respectively. The remaining are test samples for different datasets.

Method	Training	DeepCrack		Bridge2025		Crack500	
		mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑
SCSegamba	**Fully Annotation**	61.07 ± 1.25	74.88 ± 0.85	50.05 ± 3.18	64.75 ± 3.10	49.19 ± 1.06	65.13 ± 2.01
FlexiCrackNet		62.25 ± 3.16	75.31 ± 1.41	51.32 ± 2.11	65.11 ± 2.09	52.11 ± 0.86	68.10 ± 1.21
CrackSAM		65.62 ± 1.65	77.55 ± 1.56	54.89 ± 2.11	68.75 ±1.90	54.22 ± 1.54	70.58 ± 1.56
SECrackSeg		64.99 ± 1.21	77.02 ± 0.97	54.31 ± 1.89	68.33 ±1.78	53.04 ± 2.31	69.11 ± 2.35
Patch-VQVAE	**Annotation-free**	29.22 ±. 3.21	40.20 ± 3.43	15.02 ± 1.87	21.77 ± 2.30	10.15 ± 2.91	18.33 ± 2.18
RIAD		26.76 ±. 3.07	38.31 ± 3.79	11.36 ± 2.06	18.34 ± 2.88	9.63 ± 3.73	16.50 ± 5.63
Up-CrackNet		48.09 ± 2.99	61.63 ± 3.20	21.02 ± 1.05	31.42 ± 2.03	21.27 ±1.61	29.80 ± 2.33
FreeCOS		55.21 ± 3.15	68.11 ± 2.87	41.05 ± 4.10	57.09 ± 3.93	39.15 ± 2.40	55.34 ± 2.78
CoGenSAM		61.57 ± 2.33	74.28 ± 1.05	48.12 ± 3.23	63.34 ± 2.29	46.33 ± 1.34	62.22 ± 1.32

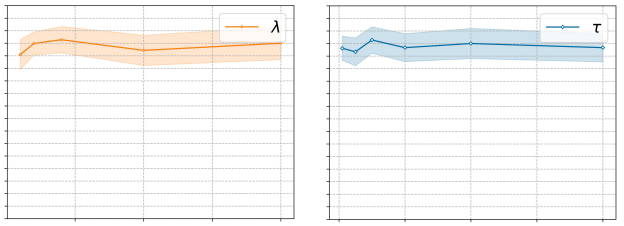
Table 1: Comparisons with different methods, including fully-supervised, weakly-supervised, and annotation-free methods, on three crack datasets. Each experiment is repeated three times and reports the mean values and standard deviations.

Implementation Details & Evaluation Metrics

Our CoGenSAM is implemented based on the PyTorch framework (Paszke et al. 2019) with NVIDIA RTX 3090. The encoder/decoder of VQVAE consists of four blocks, where each block contains two ResBlocks (He et al. 2016) and a downsampling/upsampling layer. Note that, crack images are resized to 1024×1024 pixels for training and testing. The intermediate feature has a spatial size of 18×18 and a feature dimension of 256. The patch size S is set as 3. The codebook has $N = 1024$ entries and an entry dimension of $c = 256$. The input and output dimensions of the fully connected layer in two additional projection modules are 256 and 1024, respectively. For three training stages, we use an Adam optimizer (Diederik 2014) with a batch size of 4, a learning rate of $1.0 \times e^{-3}$, and a training epoch of 20. According to our sensitivity studies, the weight of our $\mathcal{L}_c$ in Eq. 10 is 0.02 and the constant temperature τ is set as 0.1. By following existing works (Cheng et al. 2021), we first calculate the F1 score and mIoU for each image. Then, the average of each metric across all images is set as our metric.

Sensitivity Study

There are two hyper-parameters in this paper. λ is used to balance the loss functions, and τ is the constant temperature in contrastive probability distribution. The hyper-parameter sensitivity study is conducted on the DeepCrack dataset with the F1 metric. Fig. 4 shows that our method is not sensitive to the choice of λ and τ, and we set $\lambda = 0.02$ and $\tau = 0.1$.



(a) Sensitivity study on λ (b) Sensitivity study on τ

Figure 4: Hyper-parameter sensitivity study on DeepCrack dataset. Each experiment is repeated three times.

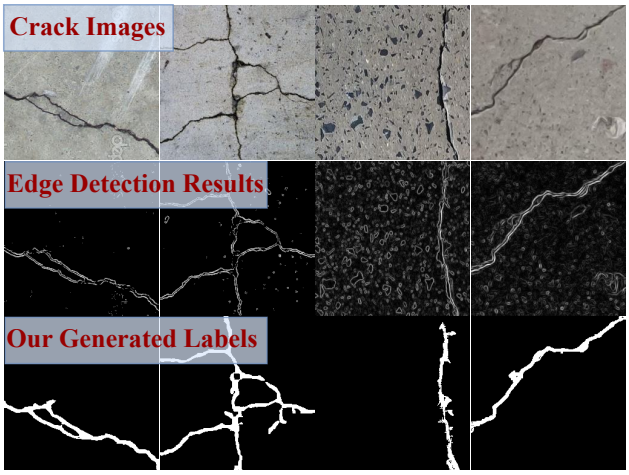


Figure 5: The first row: crack images. The second row: edge detection results. The third row: our generated labels.

Quantitative and Qualitative Results

Herein, we compare CoGenSAM with existing methods including: (i) Fully annotation methods, e.g., SCSegamba (Liu et al. 2025), FlexiCrackNet (Wan et al. 2025), CrackSAM (Ge et al. 2024) and SECrackSeg (Chen, Shi, and Pang 2025); (ii) Annotation-free methods, e.g., Patch-VQVAE (Cheng, Qu, and Lee 2024), RIAD (Zavrtanik, Kristan, and Skočaj 2021), Up-CrackNet (Ma, Fan, and Xie 2024) and FreeCOS (Shi et al. 2023); (iii) Weakly-supervised methods, e.g., CrackCLIP (Liang et al. 2025) on DeepCrack, Bridge2005, and Crack500 datasets. Table 1 shows that our CoGenSAM achieves **61.57%** on mIoU and **74.28%** on F1, which are close to the supervised methods, e.g., 64.99 and 77.02 of CrackSAM. The reason behind this effect is that, as shown in Fig. 5, our method can produce high-quality labels from imperfect edge detection results.

These results are attributed to that our synthesized crack-like structure pre-training enables the pre-trained codebook to capture crack-aware priors. Then, the restoration model can be guided to overcome noisy edge images with the help of the crack-aware priors within the codebook by explicitly performing cross-codebook learning in a contrastive interaction manner. As expected, our method achieves a noticeable

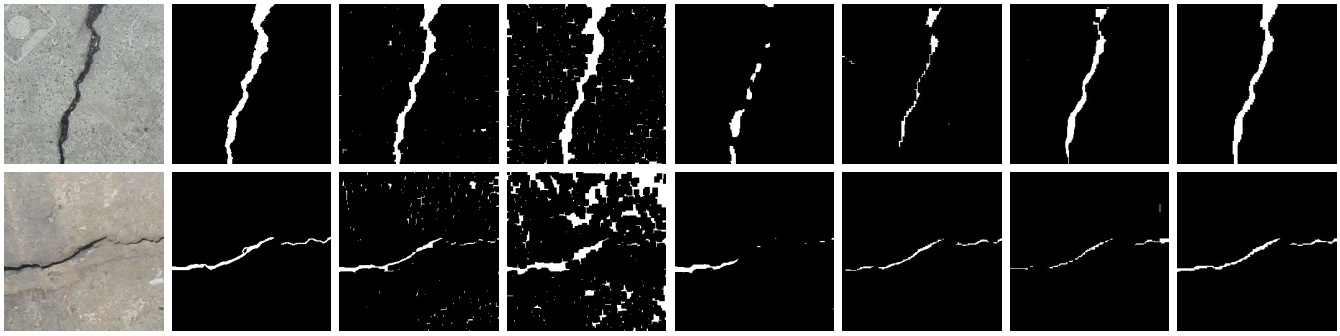


Figure 6: Results of various methods, including Patch-VQVAE, RIAD, Up-CrackNet, CrackCLIP, FreeCOS, and CoGenSAM.

performance gap with existing annotation-free methods, see Fig. 6. Visualization results can be found in Appendix E.

Ablation Study

To examine each component: codebook, crack-aware contrastive-interaction, and edge detection methods, a series of ablation experiments is performed on both the DeepCrack and Bridge2025 datasets.

Evaluation on Restoration Model. Table 2 shows that CoGenSAM involves a huge performance drop without the restoration model or codebook. The reason behind this effect is that both restoration and codebook are essential to improve the quality of generated labels from the edge detection process. As we can see from Fig. 1, the vanilla labels from edge detection suffer from substantial noise and a discontinuity problem of cracks. In contrast, the generated labels from the restoration model involve high quality, thus contributing to a reliable SAM fine-tuning.

Method	DeepCrack		Bridge2025	
	mIoU ↑	**F1** ↑	**mIoU** ↑	**F1** ↑
w/o Restoration	49.22	60.88	32.78	47.19
w/o Codebook	57.33	71.11	43.98	59.16
CoGenSAM	61.73	74.35	48.25	64.51

Table 2: Ablation study on codebook and restoration model.

Evaluation on codebook interaction. We study the effect of crack-aware contrastive-interaction ($\mathcal{L}_c$) in our proposed CoGenSAM. Table 3 shows that there is a significant drop without the supervision of $\mathcal{L}_c$. This is because such guidance could further enable our restoration model to benefit from a pre-trained codebook with a theoretical guarantee, i.e., maximizing the crack-aware information of the codebook between stage I and stage II. This will benefit the restoration model to restore corrupted labels into high-quality labels.

Evaluation on Edge Detection. We further study the effect of different edge detection methods to demonstrate our robustness, including Prewitt (Prewitt et al. 1970), Roberts (Boyle and Thomas 1988), Canny (Canny 1986), and Sobel (Sobel 2014). Table 3 shows results on DeepCrack. It can be observed that there is a little performance gap between different methods. The reason is that our method can restore those corrupted regions from edge detection, thus avoiding the instability from different edge detection methods.

Method	DeepCrack		Bridge2025	
	mIoU ↑	**F1** ↑	**mIoU** ↑	**F1** ↑
w/o $\mathcal{L}_c$	58.01	71.41	44.52	60.05
CoGenSAM	61.73	74.35	48.25	64.51

Table 3: Evaluation of codebook interaction.

Edge operator	**mIoU** ↑	**F1** ↑
Roberts	61.34 ± 2.98	74.01 ± 2.01
Prewitt	61.00 ± 2.76	73.65 ± 1.31
Canny	60.41 ± 1.92	73.10 ± 0.87
Sobel	61.57 ± 2.33	74.28 ± 1.05

Table 4: Ablation studies on edge detection methods. Each experiment is repeated three times on DeepCrack dataset.

Conclusion

This work aims to adapt Segment Anything Models into crack segmentation tasks via automatic label generation. To this end, we propose the codebook-interactive generative labeling to enable annotation-free SAM fine-tuning for crack segmentation. To achieve this, we first synthesize crack-like structure images for pre-training a reconstruction model to learn crack-aware priors within the codebook. Then, we propose the crack-aware contrastive-interaction that allows the restoration model to benefit from the above pre-trained codebook via contrastive interaction with a theoretical guarantee. In return, the restoration model could learn an informative codebook, and then restore corrupted labels from the edge detection. Finally, those generated labels can enjoy high quality and thus enable our annotation-free SAM fine-tuning process. Experiments on publicly available DeepCrack and Crack500 datasets, as well as our proposed Bridge2025 dataset, demonstrate the superiority of our method, even achieving comparable performance to supervised methods.

Acknowledgments

This work is supported in part by the National Natural Science Funds for Distinguished Young Scholar under Grant 62325307, in part by the National Natural Science Foundation of China under Grants 62527809, 62373257, 62473264, 62203134, in part by the Natural Science Foundation of Guangdong Province under Grants 2023B1515120038, in part by Shenzhen Science and Technology Innovation Commission (20220809141216003, KJZD20230923113801004), in part by the Scientific Instrument Developing Project of Shenzhen University under Grant 2023YQ019.

References

Boyle, R. D.; and Thomas, R. C. 1988. *Computer vision: A first course*. Blackwell Scientific Publications, Ltd.

Canny, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, (6): 679–698.

Chen, X.; Shi, Y.; and Pang, J. 2025. SECrackSeg: A High-Accuracy Crack Segmentation Network Based on Proposed UNet with SAM2 S-Adapter and Edge-Aware Attention. *Sensors*, 25(9): 2642.

Chen, Y.; Sun, R.; Li, W.; Mai, H.; Luo, N.; Pan, Y.; and Zhang, T. 2025a. Alleviate and mining: Rethinking unsupervised domain adaptation for mitochondria segmentation from pseudo-label perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2339–2347.

Chen, Z.; Chen, Q.; Zhang, J.; Lin, Z.; Feng, X.; Chen, J.; and Li, J. 2025b. Attack-inspired Calibration Loss for Calibrating Crack Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 15984–15992.

Chen, Z.; Hu, T.; Xu, C.; Chen, J.; Song, H. H.; Wang, L.; and Li, J. 2025c. Self-Adaptive Fourier Augmentation Framework for Crack Segmentation in Industrial Scenarios. *IEEE Transactions on Industrial Informatics*.

Chen, Z.; Lai, Z.; Chen, J.; and Li, J. 2024. Mind marginal non-crack regions: Clustering-inspired representation learning for crack segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 12698–12708.

Chen, Z.; Zhang, J.; Lai, Z.; Chen, J.; Liu, Z.; and Li, J. 2022. Geometry-aware guided loss for deep crack recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4703–4712.

Chen, Z.; Zhang, J.; Lai, Z.; Zhu, G.; Liu, Z.; Chen, J.; and Li, J. 2023. The devil is in the crack orientation: A new perspective for crack detection. In *Int. Conf. Comput. Vis.*, 6653–6663.

Cheng, M.; Zhao, K.; Guo, X.; Xu, Y.; and Guo, J. 2021. Joint topology-preserving and feature-refinement network for curvilinear structure segmentation. In *Int. Conf. Comput. Vis.*, 7147–7156.

Cheng, Q.; Qu, S.; and Lee, J. 2024. Patch-aware vector quantized codebook learning for unsupervised visual defect detection. In *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, 586–592. IEEE.

Diederik, P. K. 2014. Adam: A method for stochastic optimization. *(No Title)*.

Ge, K.; Wang, C.; Guo, Y.; Tang, Y.; Hu, Z.; and Chen, H. 2024. Fine-tuning vision foundation model for crack segmentation in civil infrastructures. *Construction and Building Materials*, 431: 136573.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Gou, Y.; Li, M.; Song, Y.; He, Y.; and Wang, L. 2023. Multi-feature contrastive learning for unpaired image-to-image translation. *Complex & Intelligent Systems*, 9(4): 4111–4122.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 770–778.

Inoue, Y.; and Nagayoshi, H. 2023. Weakly-supervised crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 24(11): 12050–12061.

Jeong, S.; Kim, Y.; Lee, E.; and Sohn, K. 2021. Memory-guided unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6558–6567.

Jiang, W.; Liu, M.; Peng, Y.; Wu, L.; and Wang, Y. 2020. HDCB-Net: A neural network with the hybrid dilated convolution for pixel-level crack detection on concrete bridges. *IEEE Transactions on Industrial Informatics*, 17(8): 5485–5494.

Jiang, X.; Wan, X.; Zhu, K.; Qiu, X.; and Fang, Z. 2024. Distribution-aware Noisy-label Crack Segmentation. *arXiv preprint arXiv:2410.09409*.

Kim, B.; Oh, Y.; and Ye, J. C. 2022. Diffusion adversarial representation learning for self-supervised vessel segmentation. *arXiv preprint arXiv:2209.14566*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Int. Conf. Comput. Vis.*, 4015–4026.

Lei, Q.; Zhong, J.; Dong, M.; and Ota, K. 2025. Faithful crack image synthesis from evolutionary pixel-level annotations via latent semantic diffusion model. *Expert Systems with Applications*, 275: 126986.

Lei, Q.; Zhong, J.; and Wang, C. 2024. Joint optimization of crack segmentation with an adaptive dynamic threshold module. *IEEE Transactions on Intelligent Transportation Systems*, 25(7): 6902–6916.

Li, J.; Fan, J.; Yang, Y.; Mei, S.; Xiao, J.; and Zhang, Z. 2024. Fully data-driven pseudo label estimation for pointly-supervised panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3127–3135.

Li, W.; Han, W.; Deng, L.-J.; Xiong, R.; and Fan, X. 2025a. Spiking Variational Graph Representation Inference for Video Summarization. *IEEE Transactions on Image Processing*.

Li, W.; Huyan, J.; Gao, R.; Hao, X.; Hu, Y.; and Zhang, Y. 2021. Unsupervised deep learning for road crack classification by fusing convolutional neural network and k_means clustering. *Journal of Transportation Engineering, Part B: Pavements*, 147(4): 04021066.

Li, W.; Yang, Z.; Han, W.; Man, H.; Wang, X.; and Fan, X. 2025b. Hyperbolic-constraint Point Cloud Reconstruction from Single RGB-D Images. In *AAAI*, volume 39, 4959–4967.

Liang, F.; Li, Q.; Yu, H.; and Wang, W. 2025. CrackCLIP: Adapting Vision-Language Models for Weakly Supervised Crack Segmentation. *Entropy*, 27(2): 127.

Liu, H.; Jia, C.; Shi, F.; Cheng, X.; and Chen, S. 2025. SCSegamba: lightweight structure-aware vision mamba for crack segmentation in structures. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 29406–29416.

Liu, H.; Miao, X.; Mertz, C.; Xu, C.; and Kong, H. 2021a. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3783–3792.

Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021b. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1): 857–876.

Liu, Y.; Yao, J.; Lu, X.; Xie, R.; and Li, L. 2019. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338: 139–153.

Ma, N.; Fan, R.; and Xie, L. 2024. UP-CrackNet: unsupervised pixel-wise road crack detection via adversarial image restoration. *IEEE Transactions on Intelligent Transportation Systems*.

Ma, Y.; Hua, Y.; Deng, H.; Song, T.; Wang, H.; Xue, Z.; Cao, H.; Ma, R.; and Guan, H. 2021. Self-supervised vessel segmentation via adversarial learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7536–7545.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*

Prewitt, J. M.; et al. 1970. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1): 15–19.

Rakshitha, R.; Srinath, S.; Kumar, N. V.; Rashmi, S.; and Poornima, B. 2024. Crack-SAM: Crack Segmentation Using a Foundation Model.

Rostami, G.; Chen, P.-H.; and Hosseini, M. S. 2025. Segment Any Crack: Deep Semantic Segmentation Adaptation for Crack Detection. *arXiv preprint arXiv:2504.14138*.

Seibold, C. M.; Reiß, S.; Kleesiek, J.; and Stiefelhagen, R. 2022. Reference-guided pseudo-label generation for medical semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 2171–2179.

Shi, T.; Ding, X.; Zhang, L.; and Yang, X. 2023. Freecos: self-supervised learning from fractals and unlabeled images for curvilinear object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 876–886.

Shi, X.; Jiang, Y.; Jiang, X.; Xu, M.; and Liu, Y. 2025. CrossDiff: Diffusion Probabilistic Model With Cross-conditional Encoder-Decoder for Crack Segmentation. *arXiv preprint arXiv:2501.12860*.

Sobel, I. 2014. An Isotropic 3x3 Image Gradient Operator. *Presentation at Stanford A.I. Project 1968*.

Vairalkar, M. K.; and Nimbhorkar, S. 2012. Edge detection of images using Sobel operator. *International Journal of Emerging Technology and Advanced Engineering*, 2(1): 291–293.

Wan, X.; Jiang, X.; Luo, G.; Sohel, F.; and Hwang, J. 2025. FlexiCrackNet: A Flexible Pipeline for Enhanced Crack Segmentation with General Features Transfered from SAM. *arXiv preprint arXiv:2501.18855*.

Wang, Y.; He, J.; and Yu, S. 2024. Crack-EdgeSAM Self-Prompting Crack Segmentation System for Edge Devices. *arXiv e-prints*, arXiv–2412.

Xu, Y.; Yang, Y.; and Zhang, L. 2023a. DeMT: Deformable mixer transformer for multi-task learning of dense prediction. In *AAAI*, volume 37, 3072–3080.

Xu, Y.; Yang, Y.; and Zhang, L. 2023b. Multi-task learning with knowledge distillation for dense prediction. In *Int. Conf. Comput. Vis.*, 21550–21559.

Xu, Y.; Ye, X.; and Su, D. 2025. Multi-Task Dense Prediction Fine-Tuning with Mixture of Fine-Grained Experts. In *ACM Int. Conf. Multimedia*, 4758–4767.

Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; and Ling, H. 2019. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Transactions on Intelligent Transportation Systems*.

Ye, Z.; Lovell, L.; Faramarzi, A.; and Ninić, J. 2024. Sam-based instance segmentation models for the automation of structural damage detection. *Advanced Engineering Informatics*, 62: 102826.

Zamir, M. 2001. Arterial branching within the confines of fractal L-system formalism. *The Journal of general physiology*, 118(3): 267–276.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112: 107706.

Zhang, P.; Ryu, H.; Miao, Y.; Jo, S.; and Park, G. 2024. Robust unsupervised-learning based crack detection for stamped metal products. *Journal of Manufacturing Systems*, 73: 65–74.

Zhang, Q.; Qi, Y.; Tang, X.; Yuan, R.; Lin, X.; Zhang, K.; and Yuan, C. 2025. Rethinking pseudo-label guided learning for weakly supervised temporal action localization from the perspective of noise correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 10085–10093.

Zhao, C.; Cai, W.-L.; and Yuan, Z. 2024. Spectral normalization and dual contrastive regularization for image-to-image translation. *The Visual Computer*, 1–12.