

CAC: Confidence-Aware Co-Training for Weakly Supervised Crack Segmentation

Fengjiao Liang ¹, Qingyong Li ¹, Xiaobao Li ², Yang Liu ¹ and Wen Wang ^{1,*}

¹ Key Laboratory of Big Data Artificial Intelligence in Transportation (Beijing Jiaotong University), Ministry of Education, Beijing 100044, China; liangfj@bjtu.edu.cn (F.L.); liqy@bjtu.edu.cn (Q.L.); yliucit@bjtu.edu.cn (Y.L.)

² School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China; 6020230055@jsnu.edu.cn

* Correspondence: wangwen@bjtu.edu.cn

Abstract: Automatic crack segmentation plays an essential role in maintaining the structural health of buildings and infrastructure. Despite the success in fully supervised crack segmentation, the costly pixel-level annotation restricts its application, leading to increased exploration in weakly supervised crack segmentation (WSCS). However, WSCS methods inevitably bring in noisy pseudo-labels, which results in large fluctuations. To address this problem, we propose a novel confidence-aware co-training (CAC) framework for WSCS. This framework aims to iteratively refine pseudo-labels, facilitating the learning of a more robust segmentation model. Specifically, a co-training mechanism is designed and constructs two collaborative networks to learn uncertain crack pixels, from easy to hard. Moreover, the dynamic division strategy is designed to divide the pseudo-labels based on the crack confidence score. Among them, the high-confidence pseudo-labels are utilized to optimize the initialization parameters for the collaborative network, while low-confidence pseudo-labels enrich the diversity of crack samples. Extensive experiments conducted on the Crack500, DeepCrack, and CFD datasets demonstrate that the proposed CAC significantly outperforms other WSCS methods.

Keywords: weakly supervised learning; crack segmentation; co-training; confidence aware; pseudo-label dynamic division



Citation: Liang, F.; Li, Q.; Li, X.; Liu, Y.; Wang, W. CAC: Confidence-Aware Co-Training for Weakly Supervised Crack Segmentation. *Entropy* **2024**, *26*, 328. <https://doi.org/10.3390/e26040328>

Academic Editor: Amelia Carolina Sparavigna

Received: 12 January 2024

Revised: 29 March 2024

Accepted: 7 April 2024

Published: 12 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Crack is one of the common defects in infrastructure such as roads, bridges, and tunnels. Regular maintenance contributes to effectively extending the service life of these infrastructures [1–3]. Automatic crack detection based on computer vision has become an instantly efficient and widely adopted method due to its non-contact and cost-effectiveness [4–7]. This task is essentially treated as a binary image segmentation problem, where each pixel in an image is classified as either “crack” or “non-crack”. Fully supervised crack image segmentation methods have demonstrated outstanding performance [8,9]. However, its effectiveness is contingent upon accurate pixel-level annotations. Pixel-level annotations are not only costly but also demand specialized expertise. To address this issue, weakly supervised crack segmentation (WSCS) attracts increasing attention [10]. The weakly supervised segmentation only needs coarse-grained information such as the bounding boxes [11], and patch-level labels [12–14]. Among them, the patch level labels greatly reduce annotation difficulty by only giving the category information of image patches, which are thus widely researched in the existing literature of WSCS.

To learn the accurate position and boundary information with merely the coarse-grained labels, the existing approach typically comprises two key stages: pseudo-label generation and segmentation model training. During the pseudo-labels generation stage, pixel-level pseudo-labels are derived from the crack detector using weak labels. During the segmentation model training stage, these pixel-level pseudo-labels are treated as ground

truth for training the crack segmentation model. The majority of existing WSCS methods primarily concentrate on improving the accuracy of pixel pseudo-label generation. König et al. [12] introduced a WSCS method that produces high-quality pixel pseudo-labels by integrating class activation maps (CAMs) [15] with classifier localization and threshold segmentation. Similarly, Dong et al. [13] proposed a WSCS model rooted in patch-based techniques. They applied Conditional Random Field (CRF) post-processing to the crack CAM to generate initial crack pixel pseudo-labels, subsequently using these pseudo-labels to train the segmentation model. It is worth noting that both these methods rely exclusively on image class labels to generate pseudo-labels from CAMs, but CAMs often highlight the most discriminative part of the region and potentially compromise the overall integrity of crack regions. Al-Huda et al. [10] incorporated multi-scale CAMs as the initial crack pixel pseudo-labels. From the model training perspective, they introduced an Incremental Annotation Refinement (IAR) strategy, meticulously designed to iteratively improve the segmentation model. Moreover, the challenge arises from the inherent thin topology and low contrast of cracks, causing the activated crack pixels to appear coarser than the ground truth. Consequently, the generated pixel-level pseudo-labels contain a significant amount of noisy pixels, making them less reliable for model training. Nevertheless, this deep neural network model is sensitive to noisy labels, which can lead to model over-fitting in a mislabeled feature space [16].

To overcome the challenge of noisy labels in crack segmentation, the WSCS task can be defined as a robust learning problem of pixel-wise noisy labels. Notably, it can be observed that the center pixels of activated crack objects tend to have higher confidence scores in pseudo-labels, while the surrounding pixels have lower scores. This observation has inspired us to introduce a novel framework for co-training crack segmentation models, which leverages pixel pseudo-labels with varying confidence scores. In addition, we have incorporated a dynamic correction mechanism during the model training process to ensure that pixel-level pseudo-labels with different confidence scores are iteratively refined. Our approach aims to exploit the properties of noisy pseudo-labels and provide valuable insights for crack segmentation tasks.

The confidence-aware co-training (CAC) framework has been proposed to effectively address the issue of crack segmentation. Firstly, the CAC framework trains the crack classifier to generate class activation maps (CAMs) [15] using only patch-level class labels. These CAMs are subsequently refined to serve as initial crack pixel pseudo-labels. Secondly, the crack pixel pseudo-labels are categorized into two groups, high confidence and low confidence, based on the confidence scores within the CAM. Subsequently, a co-training crack segmentation model is introduced, where two collaborative networks are fed successively with both the high-confidence and low-confidence pseudo-label sample sets. The high-confidence pseudo-labels offer the model more reliable crack features and superior initialization parameters, while the low-confidence pseudo-labels provide a diverse range of crack features to enrich understanding. This strategy of co-training pseudo-labels with varying confidence scores harnesses the inherent characteristics of noisy pseudo-labels, enhancing the generalization of the model. Finally, to elevate the quality of the pseudo-labels, the intermediate results predicted by the segmentation model during the training process are used as weighting parameters to dynamically re-weight the pseudo-labels. This iterative optimization process ensures that both high-confidence and low-confidence pseudo-labels receive continual updates, ultimately contributing to a more accurate and robust segmentation model.

The contributions of this work are summarized as follows:

- A novel confidence-aware co-training framework is introduced for weakly supervised crack segmentation.
- Aiming at mitigating the effect of noisy pseudo-labels, a co-training mechanism is designed to iteratively refine the predicted pseudo-labels and accordingly learn a more robust crack segmentation model.

- A dynamic division strategy is proposed to handle the noisy pseudo-labels. Among them, the high-confidence pseudo-labels are utilized to optimize the initialization parameters and those with low-confidence enrich the diversity of crack samples.
- The effectiveness of the proposed CAC is demonstrated through extensive validation on three crack datasets: Crack500, DeepCrack, and CFD. The results showcase the superior performance of this approach compared to other state-of-the-art models.

The subsequent structure is as follows: Section 2 reviews existing work related to crack segmentation methods. Section 3 outlines the proposed CAC framework in detail. Section 4 presents the experimental setup and results pertaining to crack segmentation. Finally, Section 5 concludes the paper and offers insights for future research.

2. Related Works

This section provides an overview of crack segmentation research, covering both fully supervised and weakly supervised crack segmentation methods.

2.1. Fully Supervised Crack Segmentation Method

In recent years, deep neural network techniques have exhibited remarkable performance in the realm of image semantic segmentation [17]. These techniques have also found widespread application in the detection of cracks on infrastructure surfaces such as pavements and bridges [4]. Unlike natural images, cracks present a distinct linear topological structure with discontinuous, low-contrast crack pixels. Consequently, fully supervised crack segmentation methods have traditionally amalgamated principles of edge detection [18,19] with semantic segmentation frameworks. These methods fuse multi-scale features to distinguish cracks from the background. A noteworthy example is DeepCrack [20], which integrates multi-level features within the fully convolutional networks (FCNs) for semantic segmentation [21] to achieve superior crack recognition. Zou et al. [22] introduced DeepCrack, which is a crack segmentation framework that incorporates multi-scale features of encoder and decoder in the SegNet for image segmentation [23] to capture crack structures. Both of these approaches propose the strategy of multi-scale feature fusion, which effectively addresses the challenge of crack continuity.

Building upon multi-scale fusion, several methods introduce an attention mechanism to emphasize crack pixels within the segmentation. Chen et al. [24] proposed a crack segmentation method that integrates an attention mechanism into the U-Net [25] framework to efficiently focus on crack pixel information. Sun et al. [26] introduced a pavement crack segmentation framework. This framework incorporates a novel multi-scale attention module within the decoder of DeepLabv3+ for semantic segmentation [27]. The primary function of this multi-scale attention module is to generate an attention mask. This mask dynamically assigns weights to features in different layers, thereby enhancing multi-scale feature fusion. Wang et al. [28] developed a saliency detection method, RENet, for pavement cracks. It incorporates a rectangular convolutional pyramid module to fuse contrast information between crack and background at different scales. Furthermore, the method introduces a crack edge enhancement network to filter out background noise and refine crack boundaries globally.

The advent of Transformer [29] has brought about a paradigm shift in vision tasks. In the field of crack segmentation [30–32], Transformer overcomes the limitation of fully exploiting contextual information in convolutional neural networks (CNNs) by capturing long-range dependencies in images through a self-attentive mechanism [33]. Presently, fully supervised crack segmentation methods that rely on data-driven approaches have achieved optimal performance in the field. However, these methods are contingent on high-precision manual annotation, incurring substantial costs.

2.2. Weakly Supervised Crack Segmentation Methods

To mitigate the cost of pixel-level annotation, weakly supervised crack segmentation (WSCS) methods have garnered increasing interest [34–37]. Presently, common weak labels

in WSCS methods encompass category labels, bounding boxes, scribble lines, and other variants. Zhang et al. [11] introduced a WSCS method that leverages the bounding box labels of cracks to train an object detection network for the generation of initial pseudo-labels. These initial pseudo-labels are subsequently refined into precise pixel-level crack labels through the application of region-growing and GrabCut algorithms [38]. Zhang et al. [39] developed CrackGAN, a crack segmentation network based on generative adversarial networks [40] using scribble lines as weak labels. This method employs an asymmetric U-shape generator to address the “all-black” problem caused by the imbalance between crack and background pixels.

Among the various weak labels mentioned, patch-level category labels prove to be the most accessible and widely adopted. Fan et al. [41] presented a patch-based crack segmentation method. It combines traditional image processing methods with a deep classification model. The deep classification model is responsible for locating the cracks, while traditional image processing techniques are employed for bilateral filtering and threshold segmentation on the crack image patches. Nevertheless, the prevalent WSCS method, which relies exclusively on image-level category labels, follows a two-stage approach. This process entails the generation of crack pixel labels and the subsequent training of the segmentation model. Image-level category labels are utilized to train the crack classifier, generating CAMs [15] as pixel-level crack pseudo-labels. These pseudo-labels are then used to train the crack segmentation model. König et al. [12] proposed a WSCS method that employed location with a classifier and threshold segmentation to generate crack pixel-level pseudo-labels. These pseudo-labels serve as training data for the crack segmentation model. This approach enables the approximate identification of crack locations by the crack classifier while mitigating the impact of noisy background pixels. However, this method may overlook tiny cracks. Dong et al. [13] introduced a weakly supervised patch-based crack segmentation method. It obtains CAMs as the initial pixel pseudo-labels from a trained classification network and subsequently refines these labels using CRF. Wang et al. [42] developed Crack-CAM, a pixel-level WSCS method that incorporates clustering within the CNN classifier to enhance crack features and improve the quality of the pseudo-labels assigned to crack pixels.

While several WSCS methods have prioritized the generation of higher-quality crack pixel pseudo-labels, the most recent approaches have extended their focus to enhancing the training process of the segmentation model. Al-Huda et al. [10] proposed a weakly supervised pavement crack segmentation method. This approach employs the strategy of multi-scale CAM fusion to enhance the completeness of pseudo-labels and improve the quality of the initial pseudo-labels. Furthermore, the method incorporates an incremental annotation refinement (IAR) module to progressively enhance the pseudo-labels and iteratively optimize the crack segmentation model. Al-Huda et al. [14] also introduced a hybrid deep learning approach for WSCS. This method combines CAMs from the CNN classifier with encoder-extracted features. These fused features are then input into the decoder of the segmentation model to improve the quality of crack segmentation. However, our proposed approach not only concentrates on enhancing pseudo-labels during the model training process but also capitalizes on the inherent characteristics of noisy pseudo-labels. This approach allows the crack segmentation model to be iteratively optimized for better performance.

3. Methods

3.1. Overview

This paper proposes a novel method named confidence-aware co-training (CAC) for weakly supervised crack segmentation. The whole framework of CAC is shown in Figure 1. As aforementioned, CAC is designed to iteratively refine pixel-level pseudo-labels by co-training data with high confidence and low confidence. The proposed CAC method consists of three main modules: (a) crack pseudo-labels generation, (b) dynamic division of confidence pseudo-labels, and (c) co-training of segmentation models.

The first module, the crack pseudo-label generation module, involves training a crack image classifier using solely patch-level class labels to generate crack class activation maps (CAMs), serving as initial crack pseudo-labels. During the dynamic refining process of the pixel-level pseudo-labels, the confidence measures of the pseudo-labels also change accordingly. Among them, the pixels with high-confidence pseudo-labels are utilized to mine intrinsic discriminative patterns, while those with low-confidence pseudo-labels carry some partially activated crack pixels and boundary information. This motivates the idea of confidence-aware co-training to enable a more robust crack segmentation model. Specifically, in the second stage, the crack pseudo-label is divided into two sets of pseudo-labels, each assigning different confidence scores. In the third stage, two collaborative crack segmentation networks are co-trained by both the high-confidence and low-confidence pseudo-label sample sets, leading to the optimization of the pseudo-label quality during the training process. These main modules are respectively detailed in the following three subsections.

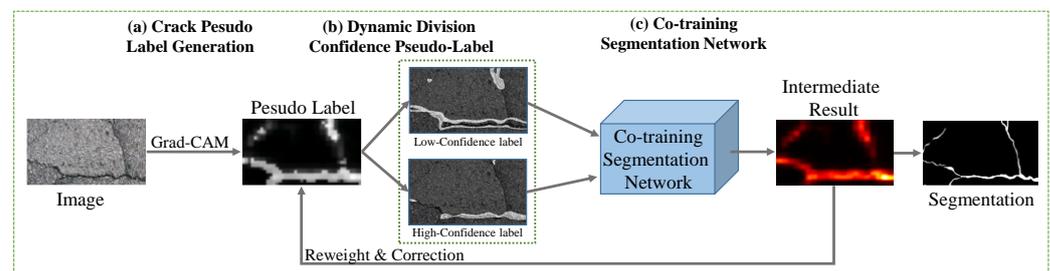


Figure 1. Overview of the proposed CAC framework. (a) Pixel-level crack pseudo-labels are produced using the patch-based Grad-CAM method. (b) The prediction results generated by the segmentation model at each training epoch are utilized as weights for updating the pseudo-labels. The pseudo-labels are dynamically divided into high-confidence and low-confidence sets based on the confidence scores associated with the crack pixels. (c) A co-training network is designed to train the segmentation model by collaborating high-confidence pseudo-label samples with low-confidence ones. The purpose is to train the segmentation model progressively and iteratively optimize its performance.

3.2. Crack Pseudo-Label Generation

The CAM method proposed as a visualization tool for convolutional neural networks is widely applied to weakly supervised image segmentation tasks [43–45]. Image-level labels are employed to train an image classification model, and the CAM method generates pixel-level pseudo-labels based on up-sampling the regions of interest of the classification model. The CAM operates by applying global average pooling (GAP) to the final convolutional layer features in the classifier and then reversely mapping the weight of the GAP output layer back to the output feature layer. Building on the CAM method, the Grad-CAM method [46] omits the GAP layer and directly propagates the gradient information of the target object into the final convolutional layer, thereby providing a more direct focus on the target pixels. Following this principle, this work employs the ResNet network [47] to train a classification model and applies the Grad-CAM method to generate an initial pseudo-label for each pixel. Since crack pixels typically represent a relatively small proportion of the entire image and tend to be overlooked during feature extraction due to their linear topology, this work constructs an image patch dataset with class labels (crack or non-crack) for training the classification model. The Grad-CAM method based on local image patches aims to maximize the activation of the crack area within the entire image.

Given an image X , it is divided into a crack image patch $x = \{x_i \mid i = 1, 2, \dots, n\}$ with overlapping regions using a step size of d . Each crack image patch x_i is fed into the classification model, resulting in the generation of the corresponding CAM s_i . These s_i are then synthesized to produce the crack CAM S corresponding to the original image X . The crack pseudo-labels are generated using two distinct methods for comparison. The first method employs location information with the crack classifier and threshold segmentation

as proposed by König et al. [12]. The second method utilizes CRF post-processing to refine the crack CAMs, following the approach by Dong et al. [13]. Both of these methods produce pseudo-labels with varying degrees of quality. According to Equation (1), the initial pseudo-label M is formulated as follows: if P_{ij} is greater than τ_1 , then M_{ij} is set to P_{ij} , and otherwise, M_{ij} is set to 0:

$$M_{ij} = \begin{cases} P_{ij}, & P_{ij} \geq \tau_1; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Here, P_{ij} denotes the pixel values of the i -th row and j -th column in the pseudo-label image P . The threshold τ_1 corresponds to the value selected from the top k elements in the histogram of pseudo-label P . Based on the crack pseudo-label pixels statistics, k is set to a larger value of 15% to retain as many crack pixels as possible. This thresholding process is used to create the gray-scale image M , which serves as the initial pseudo-label for crack segmentation.

3.3. Dynamic Division of Confidence Pseudo-Labels

This section addresses the issue of noisy pseudo-labels generated by CAM methods and focuses on utilizing these labels to train a robust crack segmentation model. Since CAM methods tend to activate only the most discriminative regions of the crack, the proposed method divides the noisy pseudo-labels into two sets: high-confidence and low-confidence pseudo-labels. These two sets are used to co-train the segmentation network.

In the previous stage, the initial noisy pseudo-labels are denoted as M , where $M_{ij} \in [0, 255]$. At the current stage, these M are divided into high-confidence pseudo-labels H and low-confidence pseudo-labels L by threshold processing. The division of pseudo-labels is described as follows:

$$H_{ij} = \begin{cases} 255, & M_{ij} \geq \tau_2; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$$L_{ij} = \begin{cases} 255, & \tau_1 \leq M_{ij} < \tau_2; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Here, τ_1 is defined in Equation (1), and τ_2 is the threshold corresponding to the top $C\%$ in the gray-scale histogram of M . More details about the configuration of C can be found in Section 4.9. The high-confidence sample set D_H provides accurate supervision for the segmentation network, while the low-confidence sample set D_L offers richer information to the network. During the training process, the semantic information of uncertain crack pixels is corrected by dynamically updating the noisy pseudo-labels M . The correction weights are defined as α , the size of α is the same as that of M , $\alpha_{ij} \in [0, 1]$, and α is initialized as a unit matrix. During the model training process, after each training epoch, the result map output from the segmentation model is used as the correction weights α and represents the probability value of each pixel belonging to the crack in the input image. The correction process for the pseudo label M is described as:

$$M_{t+1} = \alpha \cdot M_t. \quad (4)$$

This approach progressively refines the pixel pseudo-labels as the model is iteratively optimized.

3.4. Co-Training of Segmentation Models

Instead of directly training a segmentation model using noisy labels, the CAC framework is inspired by the optimization methodology of Model-Agnostic Meta-Learning (MAML) [48] to overcome the challenges posed by noisy pseudo-labels. Although our problem setting differs significantly from MAML, we are inspired by the optimization methodology of MAML and propose a co-training strategy. The CAC framework employs two collaborative networks that share the same architecture. This design allows high-confidence and low-confidence pseudo-labels to co-train the segmentation model

effectively. High-confidence pseudo-labels offer accurate supervision information, allowing for better initialization parameters for the segmentation model. In contrast, low-confidence pseudo-labels contain some partially activated crack pixels and background pixels around the cracks, providing richer data features.

In the CAC framework, after collecting the required image and pseudo-labels in the previous stage, the semantic segmentation dataset $D = \{(X^i, H^i, L^i)\}_{i=1}^N$ is obtained, and X^i, H^i, L^i represent the original image, high-confidence pseudo-labels, and low-confidence pseudo-labels, respectively. Two collaborative networks, denoted as f_θ and $f_{\theta'}$, are constructed. The parameters of the two networks are θ and θ' , respectively. The details of the co-training strategy are shown in Figure 2.

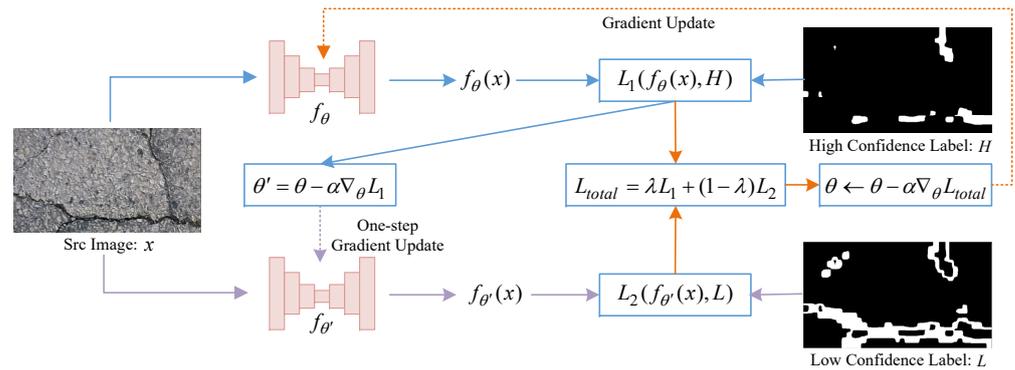


Figure 2. Co-training segmentation networks with pseudo-labels of different confidence.

Firstly, the high-confidence pseudo-label samples set $D_H = \{(X^i, H^i)\}_{i=1}^N$ is fed into the crack segmentation network f_θ , resulting in prediction results P_1^i , where $P_1^i = f_\theta(X^i)$. The segmentation loss $L_1(P_1^i, H^i)$ is calculated using the focal loss [49]. L_1 is defined as:

$$L_1(P_1^i, H^i) = - \sum_{i=1}^N \beta (1 - P_1^i)^\gamma H^i \log P_1^i + (1 - \beta) (P_1^i)^\gamma (1 - H^i) \log (1 - P_1^i). \quad (5)$$

Here, low-confidence crack pixels are not involved in training f_θ to avoid the interference of low-confidence pixels, as background pixels, with the training of f_θ . So P_1^i and H^i are masked to hide the low-confidence crack pixels portion of pseudo-labels as shown in Figure 3a. Moreover, β is used to balance the number of positive and negative samples, and γ regulates the imbalance between easy-to-discriminate and hard-to-discriminate samples. A one-step gradient update θ' is performed: $\theta' \leftarrow \theta - \nabla_\theta L_1$.

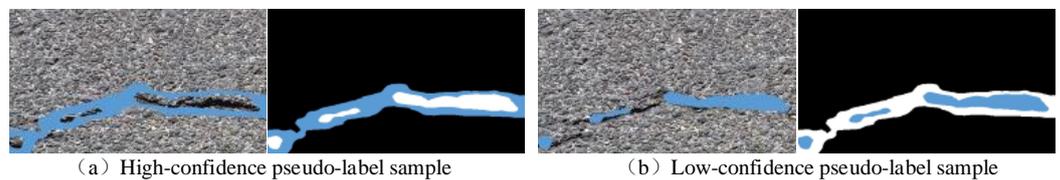


Figure 3. The graphic shows the high-confidence and low-confidence pseudo-label cracking samples. The blue parts are the masked pixels at the current confidence level.

Then, the low-confidence pseudo-label samples set $D_L = \{(X^i, L^i)\}_{i=1}^N$ is fed into the crack segmentation network $f_{\theta'}$, resulting in prediction results P_2^i , where $P_2^i = f_{\theta'}(X^i)$. The segmentation loss $L_2(P_2^i, L^i)$ is calculated as:

$$L_2(P_2^i, L^i) = - \sum_{i=1}^N \beta (1 - P_2^i)^\gamma L^i \log P_2^i + (1 - \beta) (P_2^i)^\gamma (1 - L^i) \log (1 - P_2^i). \quad (6)$$

Here, P_2^i and L^i are masked to hide the high-confidence crack pixel portion as shown in Figure 3b. High-confidence crack pixels are not involved in training $f_{\theta'}$ to avoid the high-confidence pixels, as background pixels, interfering with the training of $f_{\theta'}$.

Finally, the total loss L_{total} is calculated and is formulated as:

$$L_{total} = \lambda L_1 + (1 - \lambda)L_2. \quad (7)$$

Here, λ is a parameter used to adjust the balance of the contribution of high-confidence and low-confidence sample sets. The gradient $\theta \leftarrow \theta - \nabla_{\theta} L_{total}$ is updated for the segmentation network f_{θ} , which is used as the final test model. This process ensures that the model learns from both high-confidence and low-confidence pseudo-labels, improving the robustness of the segmentation model while exploiting the properties of the noisy pseudo-labels.

Algorithm 1 describes the overall flow of the proposed CAC method.

Algorithm 1: CAC algorithm.

Input: $D_H = \{(X^i, M^i)\}_{i=1}^N$: crack pixel pseudo-label dataset,

Epochs: training rounds,

Iterations: numbers of iterations each round,

α : step size hyperparameter

Output: θ : crack segmentation model parameters

```

1 Initialize: randomly initialize  $\theta$ ;
2 for Epochs do
3   Dividing  $M^i$  into  $H^i$  and  $L^i$ , then constructing high-confidence sample set
    $D_H = \{(X^i, H^i)\}_{i=1}^N$  and low-confidence sample set  $D_L = \{(X^i, L^i)\}_{i=1}^N$ 
   based on Equations (2) and (3);
4   for Iterations do
5     Sample a batch from the high-confidence sample set  $D_H$ ;
6     Compute loss  $L_1(f_{\theta})$  by the high-confidence pseudo-label based on
     Equation (5);
7     One-step gradient update model parameters  $\theta' \leftarrow \theta - \nabla_{\theta} L_1$ ;
8     Sample a batch from the low-confidence sample set  $D_L$ ;
9     Compute loss  $L_2(f_{\theta'})$  by the low-confidence pseudo-label based on
     Equation (6);
10    Compute total loss  $L_{total}$  based on Equation (7);
11  end
12  Update model parameters  $\theta \leftarrow \theta - \nabla_{\theta} L_{total}$ ;
13  Update crack pseudo-label  $M^i$  based on Equation (4);
14 end

```

4. Experimental Results, Comparisons, and Analysis

This section discusses the implementation details of the proposed CAC and compares its effectiveness to other state-of-the-art WSCS methods presented in recent years using three crack image datasets.

4.1. Datasets

Following the literature [12], the Crack500 dataset [50] is employed to train the classification network for computing grad-CAM and generating initial pseudo-labels. This dataset consists of 1896 crack images, each with a resolution of 648×484 pixels. To augment the data, these images are sliced into patches of 128×128 pixels, and augmentation techniques like rotation and flipping are applied. In total, 556,448 images are used for training the classification network. The dataset contains 238,820 images with cracks and 317,628 images without cracks.

The crack segmentation network is trained on the original images and the generated pseudo-labels for crack pixels in the Crack500 training set. The segmentation model is then tested on three different datasets:

- Crack500 testing dataset [50]: This dataset consists of 1124 crack images. Crack500 is a pavement cracking dataset that is collected with a mobile phone on the campus of Temple University.
- CFD dataset [51]: It contains 118 crack images, each with a resolution of 320×480 pixels, which reflect urban road surface conditions in Beijing, China. This dataset includes various types of noise such as shadows, oil spots, and water stains.
- DeepCrack dataset [20]: This dataset comprises a total of 537 images, each with a resolution of 544×384 pixels. It includes crack data with multiple textures, scenes, and scales.

These datasets serve as the foundation for training and testing the CAC framework. The diversity in dataset sources and characteristics allows for a comprehensive evaluation of the performance and its ability to handle different types of crack data.

4.2. Evaluation Metrics

Following the original literature [12,22], three different F1-based metrics are employed for the evaluation of the proposed CAC: optimal dataset scale (ODS), optimal image scale (OIS), and average precision (AP). The ODS represents the best F1 scores on the entire dataset for a fixed threshold. The OIS denotes the aggregate F1 scores calculated for each image in the dataset using the best threshold for each image. The AP denotes the area under the precision–recall curve [52]. ODS and OIS are represented as follows:

$$Pr = \frac{TP}{TP + FN'} \quad (8)$$

$$Re = \frac{TP}{TP + FP'} \quad (9)$$

$$F1 = 2 \times \frac{Re \times Pr}{Re + Pr'} \quad (10)$$

$$OIS = \frac{1}{N} \sum_{i=1}^N \max \{ F1_t^i : \forall t \in \{0.01, 0.02, \dots, 0.99\} \}, \quad (11)$$

$$ODS = \max \left\{ \left\{ \frac{1}{N} \sum_{i=1}^N F1_t^i \right\} : \forall t \in \{0.01, 0.02, \dots, 0.99\} \right\}. \quad (12)$$

Here, Pr , Re , $F1$, TP , FP , TN , and FN denote precision, recall, F1-score, true positive, false positive, true negative, and false negative, sequentially. AP is represented as follows:

$$AP = \sum_{t=0.01}^{t=1} \frac{1}{T} (Re_t - Re_{t-0.01}) Pr_t, \quad (13)$$

where t is the selected threshold and T is set to 100.

4.3. Implementation Details

4.3.1. Environment

Our experiments are conducted on a deep learning workstation running Ubuntu 16.04 LTS containing a CPU of Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10 GHz and a Nvidia Titan XP GPU with 8 GB of RAM. The framework used for the experiments is pytorch1.2.

4.3.2. Experimental Setting

This section provides detailed information about the experimental settings for the crack pixel-level pseudo-label generation and the training and testing phases of the crack segmentation model. During the crack pixel-level pseudo-label generation, a binary classification

model based on ResNet50 architecture is used for generating pixel-level pseudo-labels for crack images. The classification model is trained for 10 epochs with a batch size of 16. The initial learning rate is set to 1×10^{-3} , and the learning rate is reduced by a factor of 10 per epoch. Stochastic Gradient Descent (SGD) is employed as the optimizer with a momentum value of 0.9. Grad-CAM is generated for crack images by training the binary classification network (ResNet50). Two different processes are used to obtain the initial pseudo-label from the Grad-CAM. One process involves fusing the CAMs and crack locations obtained from the classification network, followed by threshold segmentation to generate the initial crack pseudo-labels [12]. The other process refines the CAMs using CRF post-processing to produce initial crack pseudo-labels [13]. During the division phase of different confidence pseudo-labels, a threshold segmentation method P-Tile [53] is employed due to its anti-noise capabilities compared to other threshold segmentation methods. P-Tile adapts the threshold dynamically based on the gray-scale histogram statistics. The division threshold values τ_1 and τ_2 are selected based on prior information from DeepCrack [22] to match the statistics of crack pixels. During the phase of crack pseudo-label generation, k is set to 15% by P-Tile.

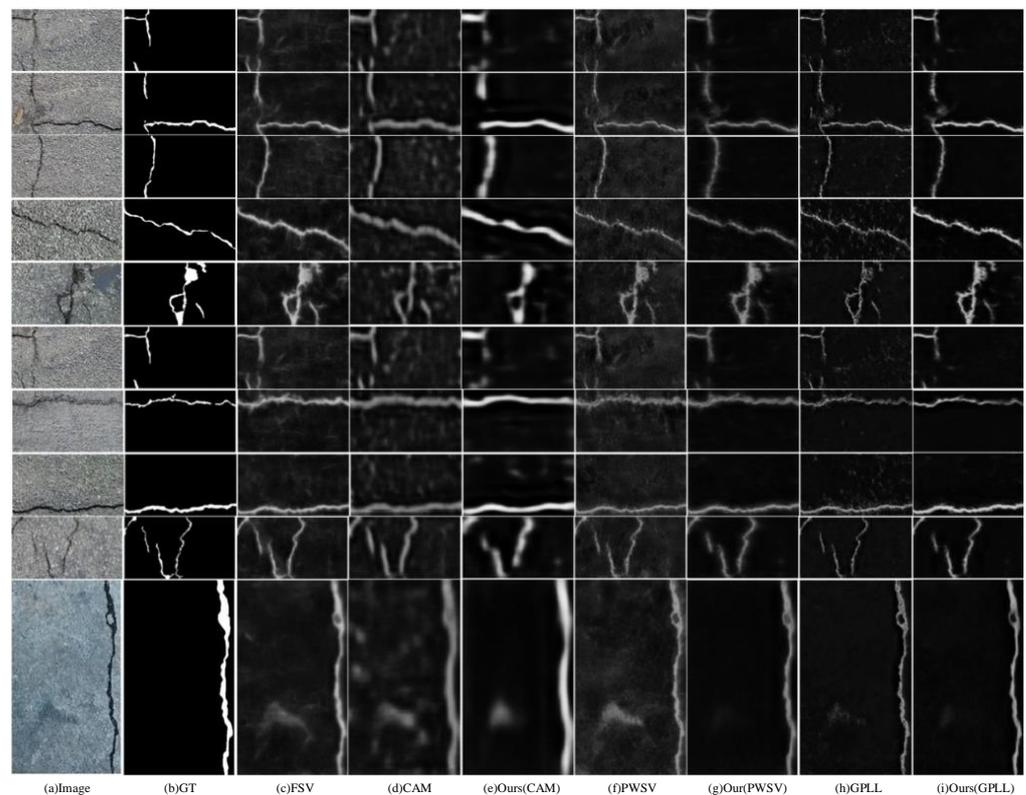
During the training phase of the crack segmentation model, DeepCrack, a specific segmentation framework, is chosen for training the crack segmentation model. The model is trained on the Crack500 training set, which includes original images and pseudo-labels of crack pixels with different confidence scores. Training takes place for 30 epochs with a batch size of 4. The initial learning rate is set to 1×10^{-3} and is reduced by a factor of 10 every 10 epochs. The used optimizer is SGD with a momentum value of 0.9. The input image size for DeepCrack is set to 256×256 pixels. Focal loss, a type of binary cross-entropy loss, is employed to calculate the error between the predicted output and the corresponding pseudo-labels. During segmentation model testing, three datasets including the Crack500 test set, CFD, and DeepCrack are involved in the inference process.

4.4. Evaluation on Crack500

As shown in Table 1, the results of the Crack500 testing dataset demonstrate the effectiveness of the proposed CAC method compared to the state-of-the-art methods. Compared to using CAM as pseudo-labels for direct segmentation model training, CAC achieves improvements in terms of the following metrics: *ODS* increases by 0.76% to reach a value of 53.88%; *OIS* increases by 1.58% to reach a value of 58.44%; and *AP* improves by 7.18% to reach a value of 57.07%. Compared to PWSV methods, refining CAM with CRF post-processing as pseudo-labels for direct segmentation model training [13], CAC achieves improvements in the following metrics: *ODS* increases by 4.68% to reach a value of 61.22%; *OIS* improves slightly, with a 0.34% increase to reach a value of 64.07%; and the *AP* value is closer, with a value of 65.10%. Compared to the GPLL method that employed location with a classifier and thresholding as the initial crack pseudo-labels for direct segmentation model training [12], CAC achieves significant improvements in the following metrics: *ODS* increases substantially by 15.39% to reach a value of 60.43%; *OIS* improves by 7.81%, reaching a value of 64.50%; and *AP* sees a remarkable improvement of 18.19% to reach a value of 63.65%. It is important to note that the fully supervised method (FSV), which uses pixel labels with precise annotations, serves as an upper bound on the performance of the segmentation model. These results indicate that CAC outperforms other methods when it comes to optimizing the training of models. The model is more resistant to noise interference, resulting in a refined crack segmentation map with less noise and more visible cracks, particularly with a higher recall of crack pixels as demonstrated in Figure 4.

Table 1. Evaluation of the segmentation results *ODS*, *OIS*, and *AP* of different methods on the Crack500 test set (%).

Methods	<i>ODS</i>	<i>OIS</i>	<i>AP</i>
FSV	66.20	71.97	76.70
CAM [46]	53.12	56.86	49.89
PWSV [13]	56.54	63.73	65.13
GPLL [12]	45.04	56.69	45.46
Ours (CAM)	53.88	58.44	57.07
Ours (PWSV)	61.22	64.07	65.10
Ours (GPLL)	60.43	64.50	63.65

**Figure 4.** Visualization of the crack segmentation results of the different methods on the Crack500 testing dataset.

4.5. Evaluation on CFD

As shown in Table 2, the evaluation of the CFD dataset demonstrates the performance of CAC compared to other state-of-the-art methods. The CAC performs better than existing methods when using pseudo-labels of the same quality. Specifically, when GPLL is used as the initial pseudo label [12], CAC achieves the best segmentation results, which hold on an *ODS* value of 25.31%, an *OIS* value of 31.55%, and an *AP* value of 18.55%. However, when using CAMs as pseudo-labels, the pseudo-label refinement strategy is less effective due to the roughness of the intermediate results α . Compared to the Crack500 dataset, the cracks in the CFD dataset are thinner and more challenging to detect, resulting in lower performance for the segmentation results. Note that the FSV in Table 2 refers to the segmentation model trained on Crack500 rather than CFD, and it does not show the best performance because of its poor generalization ability. Similar results can be observed in Table 3. The visualizations provided in Figure 5 indicate that CAC has less noise compared to other methods, but it may still miss thinner cracks. These results suggest that CAC

outperforms other methods on the CFD dataset in terms of handling noisy pseudo-labels, although it still faces challenges with very thin cracks. The exploration of trying to use the output results of different layers (shallow feature or deep feature) in the DeepCrack framework is helpful to further enhance the pseudo-label correction strategy.

Table 2. Evaluation of the segmentation results *ODS*, *OIS*, and *AP* of different methods on the CFD dataset (%).

Methods	<i>ODS</i>	<i>OIS</i>	<i>AP</i>
FSV	16.67	24.35	6.31
CAM [46]	23.16	17.52	14.07
PWSV [13]	8.56	14.46	7.72
GPLL [12]	18.74	19.41	14.88
Ours (CAM)	22.87	15.11	12.88
Ours (PWSV)	25.82	14.96	18.36
Ours (GPLL)	25.31	31.55	18.55

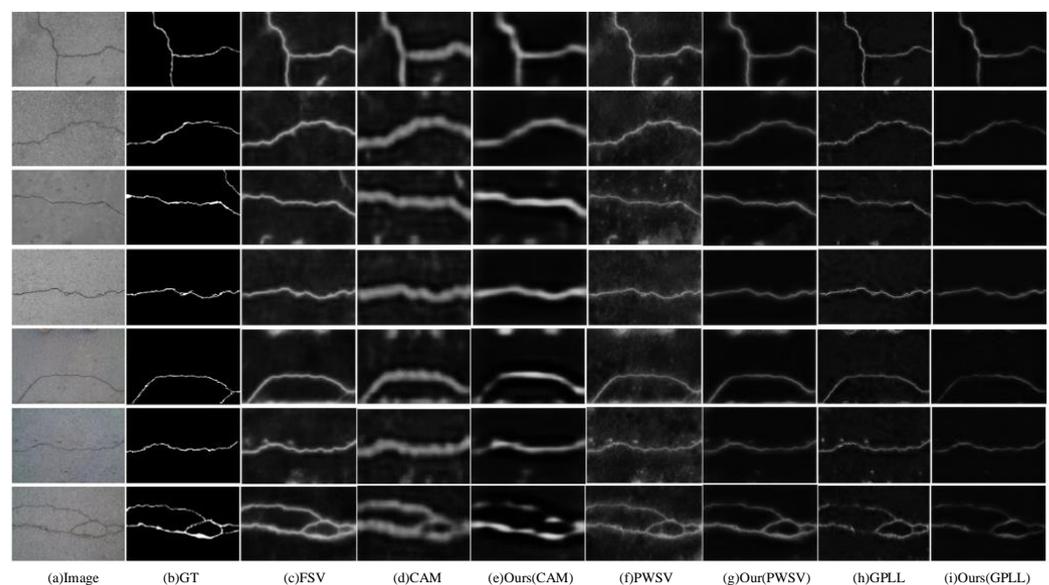


Figure 5. Visualization of the crack segmentation results of the different methods on the CFD dataset.

Table 3. Evaluation of the segmentation results *ODS*, *OIS*, and *AP* of different methods on the DeepCrack dataset (%).

Methods	<i>ODS</i>	<i>OIS</i>	<i>AP</i>
FSV	46.43	54.97	30.95
CAM [46]	44.88	52.43	37.33
PWSV [13]	37.05	43.95	44.31
GPLL [12]	65.97	73.19	72.28
Ours (CAM)	49.66	53.18	47.22
Ours (PWSV)	69.47	63.31	73.91
Ours (GPLL)	71.01	77.98	75.51

4.6. Evaluation on DeepCrack

As shown in Table 3, the evaluation of the DeepCrack dataset shows that CAC achieves superior performance compared to other state-of-the-art methods. The CAC outperforms other methods on the DeepCrack dataset when using the same quality of pseudo-labels. Specifically, when GPLL is used as the initial pseudo label [12], the CAC achieves state-of-the-art segmentation results, which hold on an *ODS* value of 71.01%, an *OIS* value of

77.98%, and an AP value of 75.51%. In Figure 6, the visualized crack segmentation results of CAC and existing methods on the DeepCrack dataset with three different pseudo-labels demonstrate that CAC produces segmentation results with fewer noisy pixels compared to other methods. The CAC displays excellent generalization performance on the DeepCrack dataset, showcasing its robustness and superior performance. These results highlight that CAC is highly effective in handling noisy pseudo-labels and demonstrates excellent generalization performance, making it a state-of-the-art method for crack segmentation on the DeepCrack dataset.

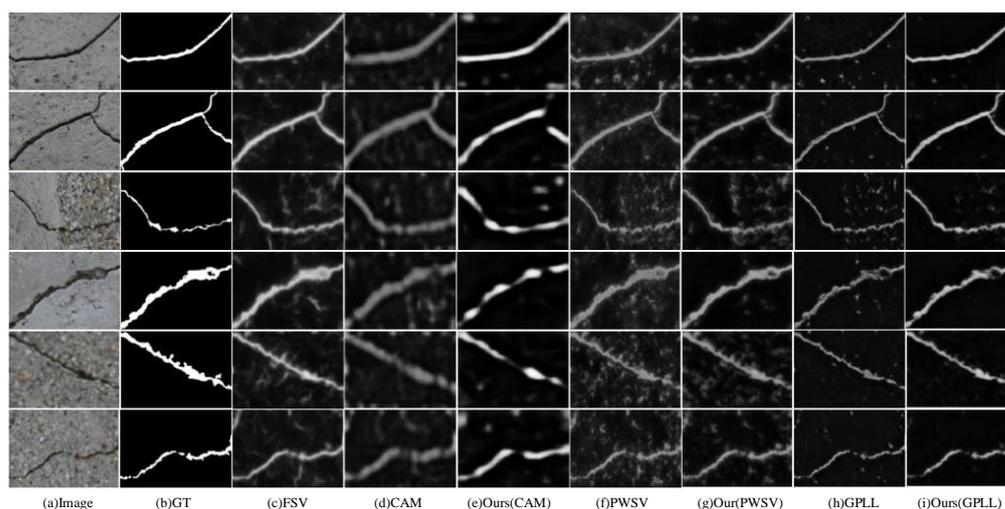


Figure 6. Visualization of the crack segmentation results of the different methods on the DeepCrack dataset.

4.7. Model Performance Discussion and Summary

From our experimental results, we observed that the performance of segmentation results significantly improves when the segmentation model is trained using crack pseudo-labels of comparable quality. Surprisingly, the noise inherent in these crack pseudo-labels has minimal impact on the performance of our model. Consequently, our approach demonstrates remarkable resilience to noise interference from pseudo-labels. The segmentation results on the CFD dataset are relatively poor compared to the Crack500 dataset. This can be attributed to the thinner and lower-contrast nature of the cracks in the CFD dataset, which can be more challenging to detect and segment accurately. The DeepCrack dataset, with a data distribution more similar to the crack distribution in the training set, yields better segmentation results with CAC.

The results presented in Figure 7 and the provided explanation indicate that the proposed CAC method faces challenges in detecting thin and low-contrast cracks. All WSCS methods for detecting thin cracks, including our proposed method, may experience instances of missed detection.

The experimental results presented in Tables 2 and 3 reveal that the weakly supervised crack segmentation method exhibits superior generalization performance compared to the fully supervised method on both the CFD and DeepCrack datasets. Analyzing the training data perspective, the fully supervised approach in this study utilizes the Crack500 dataset, potentially leading to model over-fitting. In contrast, the weakly supervised approach incorporates pseudo-labels with varying degrees of noise, which contributes to enhanced model generalization. Consequently, the proposed CAC offers a balancing mechanism that effectively mitigates the influence of dataset-specific features and noise during the data-fitting process.

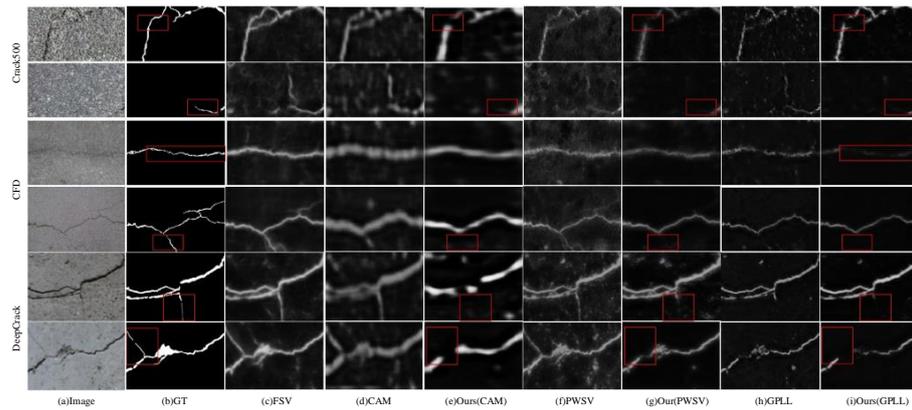


Figure 7. The graphic shows some example images with poor segmentation results under different datasets. The red box is the region where the detection error occurs.

4.8. Ablation Experiments

Table 4 presents the performance of different modules in the CAC framework using the pseudo-labels generated by the fusion of CAM and crack location [12] on the Crack500. The first row shows the segmentation performance of directly training the segmentation model using the initial pseudo-labels. This is the baseline method. The second row demonstrates that the co-training module provides a significant improvement in the segmentation performance of the model. Co-training with high-confidence and low-confidence pseudo-labels allows the model to better leverage the features of noisy labels, resulting in enhanced segmentation. The third row showcases the performance of the dynamical division module for pseudo-labels with different confidence scores. The results indicate that as the quality of pseudo-labels improves, the segmentation performance also improves. In summary, the experimental results suggest that the co-training module and the dynamical division module play crucial roles in enhancing the segmentation performance of the CAC model. These modules help the model make better use of pseudo-labels, leading to improved results.

Table 4. Performance of different modules in the model on Crack500 testing dataset (%).

Co-Training	Dynamical Division	ODS	OIS	AP
		45.04	56.69	45.46
✓		56.86	63.74	59.09
✓	✓	60.43	64.50	63.65

Figure 8 visually demonstrates the pixel pseudo-label correction during model training. It can be seen that the crack CAM is used as the initial coarse pseudo-label *Iter_0*, which contains a significant amount of noise. As the iterative number increases, it shows steady trends, and the pseudo-label is refined gradually. This supports the motivation of confidence-aware co-training experimentally and illustrates that the proposed CAC can effectively strengthen the ability to resist noises.

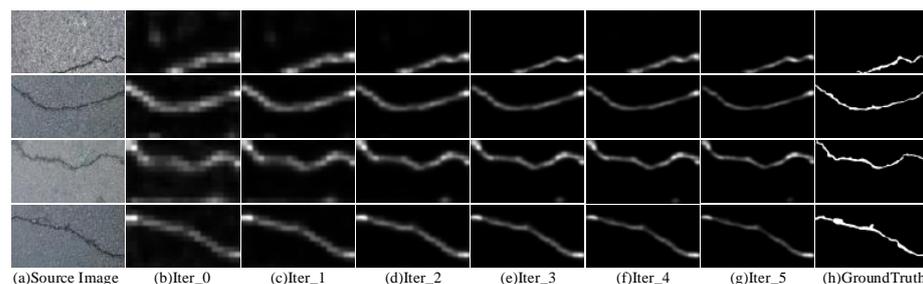


Figure 8. Visualization of the process of CAM pseudo-label correction during model iterations.

4.9. Parameter Experiments

Figure 9 illustrates the impact of different values of the parameter λ on the total loss of the segmentation model using the pseudo-labels generated by the fusion of CAM and crack location [12] on the Crack500. The graph shows that the crack segmentation model performs optimally when λ is set to 0.7. In this configuration, the contribution of the high-confidence sample set accounts for 70% of the total loss, while the contribution of the low-confidence sample set accounts for the remaining 30%. This balance between high-confidence and low-confidence sample sets yields the best segmentation results. As λ varies, the performance of the segmentation model changes. For values below 0.7, the model appears to under-utilize the high-confidence sample set, leading to sub-optimal performance. Conversely, for values above 0.7, the model starts to favor the high-confidence samples to a greater extent, which can also negatively affect the overall performance. The figure indicates that as the parameter λ changes, the performance of the model fluctuates within a relatively small range (around 10%).

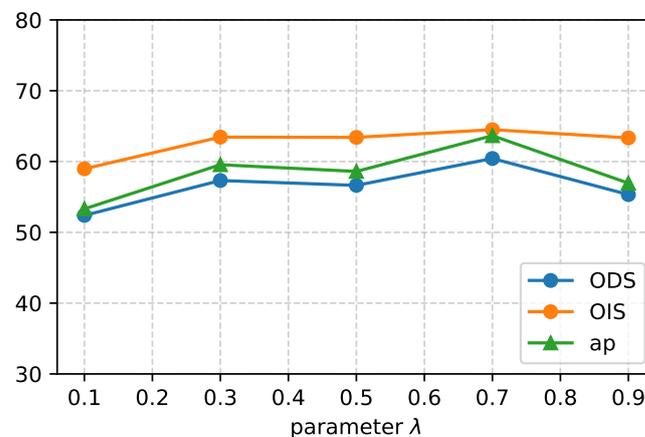


Figure 9. Performance of different λ in total loss for segmentation model on Crack500 testing dataset.

Figure 10 illustrates the impact of different division ratios, parameter C , on the performance of the crack segmentation model when using pseudo-labels generated by the fusion of CAM and crack location on the Crack500. It shows that the performance of the model is optimized when C is set to 5, which means that the top 5% of the initial noisy pseudo-labels are divided as a high-confidence sample set, and the remaining pixels are divided as a low-confidence sample set. These results indicate that selecting an appropriate division ratio is crucial for achieving the best segmentation performance with the CAC model.

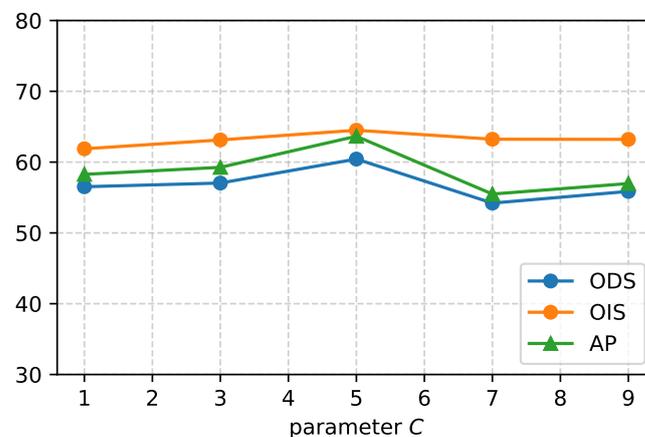


Figure 10. Performance of different division ratios C for the segmentation model on the Crack500 testing dataset.

5. Conclusions and Future Work

We contributed an effective strategy named confidence-aware co-training (CAC) for weakly supervised crack segmentation tasks. The proposed CAC employed the co-training of crack segmentation networks with pseudo-labels under different degrees of confidence to mitigate the impact of noisy labels and enhance pseudo-label accuracy. The approach leveraged high-confidence pseudo-labels for better initialization parameters and reduced over-fitting to noisy pixels while using low-confidence pseudo-labels to provide richer features for the model. Pseudo-labels were dynamically updated to improve their quality during model training by iterative optimization model. The experimental results have demonstrated the superiority of the proposed CAC approach over state-of-the-art methods on the Crack500, CFD, and DeepCrack datasets. The CAC framework offers a promising approach to weakly supervised crack segmentation and highlights the potential for further improvements in crack segmentation methodologies. For future work, more exploration will be performed to meet the challenges of CAC in preserving the connectivity of cracks.

Author Contributions: Conceptualization, F.L. and Q.L.; methodology, F.L. and W.W.; software, F.L. and X.L.; validation, F.L. and X.L.; formal analysis, Q.L.; investigation, F.L. and Y.L.; resources, Q.L.; data curation, F.L.; writing original draft preparation, F.L.; writing review and editing, Q.L. and W.W.; visualization, Y.L. and F.L.; supervision, Q.L.; project administration, F.L. and W.W.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2023JBZY037, 2022JBQY007, in part by the Beijing Natural Science Foundation under Grant L231019, in part by the Shanghai Industrial Development Project under Grant HCXBCY-2023-033.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Implementation of the proposed framework is publicly available on GitHub at the following link: <https://github.com/liangfengjiao/CAC>.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Munawar, H.S.; Hammad, A.W.A.; Haddad, A.; Soares, C.A.P.; Waller, S.T. Image-Based Crack Detection Methods: A Review. *Infrastructures* **2021**, *6*, 115. [CrossRef]
2. Yu, X.; Kuan, T.W.; Tseng, S.P.; Chen, Y.; Chen, S.; Wang, J.F.; Gu, Y.; Chen, T. EnRDeA U-net deep learning of semantic segmentation on intricate noise roads. *Entropy* **2023**, *25*, 1085. [CrossRef]
3. Zhong, J.; Huyan, J.; Zhang, W.; Cheng, H.; Zhang, J.; Tong, Z.; Jiang, X.; Huang, B. A deeper generative adversarial network for grooved cement concrete pavement crack detection. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105808. [CrossRef]
4. Ai, D.; Jiang, G.; Lam, S.K.; He, P.; Li, C. Computer vision framework for crack detection of civil infrastructure—A review. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105478. [CrossRef]
5. Wu, X.; Liu, X. Building crack identification and total quality management method based on deep learning. *Pattern Recognit. Lett.* **2021**, *145*, 225–231. [CrossRef]
6. Taheri, S. A review on five key sensors for monitoring of concrete structures. *Constr. Build. Mater.* **2019**, *204*, 492–509. [CrossRef]
7. Zhao, Y.; Yan, J.; Wang, Y.; Jing, Q.; Liu, T. Porcelain insulator crack location and surface states pattern recognition based on hyperspectral technology. *Entropy* **2021**, *23*, 486. [CrossRef] [PubMed]
8. Ali, R.; Chuah, J.H.; Talip, M.S.A.; Mokhtar, N.; Shoaib, M.A. Automatic pixel-level crack segmentation in images using fully convolutional neural network based on residual blocks and pixel local weights. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104391. [CrossRef]
9. Ali, R.; Chuah, J.H.; Talip, M.S.A.; Mokhtar, N.; Shoaib, M.A. Crack segmentation network using additive attention gate-CSN-II. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105130. [CrossRef]
10. Al-Huda, Z.; Peng, B.; Algburi, R.N.A.; Alfasly, S.; Li, T. Weakly supervised pavement crack semantic segmentation based on multi-scale object localization and incremental annotation refinement. *Appl. Intell.* **2023**, *53*, 14527–14546. [CrossRef]
11. Zhang, H.; Qian, Z.; Tan, Y.; Xie, Y.; Li, M. Investigation of pavement crack detection based on deep learning method using weakly supervised instance segmentation framework. *Constr. Build. Mater.* **2022**, *358*, 129117. [CrossRef]

12. König, J.; Jenkins, M.D.; Mannion, M.; Barrie, P.; Morison, G. Weakly-Supervised Surface Crack Segmentation by Generating Pseudo-Labels Using Localization with a Classifier and Thresholding. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24083–24094. [[CrossRef](#)]
13. Dong, Z.; Wang, J.; Cui, B.; Wang, D.; Wang, X. Patch-based weakly supervised semantic segmentation network for crack detection. *Constr. Build. Mater.* **2020**, *258*, 120291. [[CrossRef](#)]
14. Al-Huda, Z.; Peng, B.; Algburi, R.N.A.; Al-antari, M.A.; AL-Jarazi, R.; Zhai, D. A hybrid deep learning pavement crack semantic segmentation. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106142. [[CrossRef](#)]
15. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
16. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.G. Learning from Noisy Labels with Deep Neural Networks: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 8135–8153. [[CrossRef](#)] [[PubMed](#)]
17. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [[CrossRef](#)] [[PubMed](#)]
18. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
19. Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. Richer Convolutional Features for Edge Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
20. Liu, Y.; Yao, J.; Lu, X.; Xie, R.; Li, L. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing* **2019**, *338*, 139–153. [[CrossRef](#)]
21. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
22. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans. Image Process.* **2019**, *28*, 1498–1512. [[CrossRef](#)]
23. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)]
24. Chen, J.; He, Y. A novel U-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at pixel level. *Comput. -Aided Civ. Infrastruct. Eng.* **2022**, *37*, 1721–1736. [[CrossRef](#)]
25. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015.
26. Sun, X.; Xie, Y.; Jiang, L.; Cao, Y.; Liu, B. DMA-Net: DeepLab with Multi-Scale Attention for Pavement Crack Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18392–18403. [[CrossRef](#)]
27. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
28. Wang, Y.; Song, K.; Liu, J.; Dong, H.; Yan, Y.; Jiang, P. RENet: Rectangular convolution pyramid and edge enhancement network for salient object detection of pavement cracks. *Measurement* **2021**, *170*, 108698. [[CrossRef](#)]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
30. Wang, W.; Su, C. Automatic concrete crack segmentation model based on transformer. *Autom. Constr.* **2022**, *139*, 104275. [[CrossRef](#)]
31. Zhou, Z.; Zhang, J.; Gong, C. Hybrid semantic segmentation for tunnel lining cracks based on Swin Transformer and convolutional neural network. *Comput. -Aided Civ. Infrastruct. Eng.* **2023**, *38*, 2491–2510. [[CrossRef](#)]
32. Asadi Shamsabadi, E.; Xu, C.; Rao, A.S.; Nguyen, T.; Ngo, T.; da Costa, D.D. Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Autom. Constr.* **2022**, *140*, 104316. [[CrossRef](#)]
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
34. Hong, Y.; Lee, S.J.; Yoo, S.B. AugMoCrack: Augmented morphological attention network for weakly supervised crack detection. *Electron. Lett.* **2022**, *58*, 651–653. [[CrossRef](#)]
35. Inoue, Y.; Nagayoshi, H. Crack Detection as a Weakly-Supervised Problem: Towards Achieving Less Annotation-Intensive Crack Detectors. In Proceedings of the 2020 25th International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2021.
36. Rill-García, R.; Dokládlová, E.; Dokládál, P. Pixel-accurate road crack detection in presence of inaccurate annotations. *Neurocomputing* **2022**, *480*, 1–13. [[CrossRef](#)]
37. Rill-García, R.; Dokládlová, E.; Dokládál, P. Syncrack: Improving Pavement and Concrete Crack Detection Through Synthetic Data Generation. In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP'22), Virtual, 8–10 February 2022.
38. Rother, C.; Kolmogorov, V.; Blake, A. “GrabCut”: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Trans. Graph.* **2004**, *23*, 309–314. [[CrossRef](#)]
39. Zhang, K.; Zhang, Y.; Cheng, H.D. CrackGAN: Pavement Crack Detection Using Partially Accurate Ground Truths Based on Generative Adversarial Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1306–1319. [[CrossRef](#)]

40. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
41. Fan, R.; Bocus, M.J.; Zhu, Y.; Jiao, J.; Wang, L.; Ma, F.; Cheng, S.; Liu, M. Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium, Paris, France, 9–12 June 2019.
42. Wang, H.; Li, Y.; Dang, L.M.; Lee, S.; Moon, H. Pixel-level tunnel crack segmentation using a weakly supervised annotation approach. *Comput. Ind.* **2021**, *133*, 103545. [[CrossRef](#)]
43. Ahn, J.; Kwak, S. Learning Pixel-Level Semantic Affinity With Image-Level Supervision for Weakly Supervised Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
44. Ru, L.; Zhan, Y.; Yu, B.; Du, B. Learning Affinity From Attention: End-to-End Weakly-Supervised Semantic Segmentation With Transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.
45. Rong, S.; Tu, B.; Wang, Z.; Li, J. Boundary-Enhanced Co-Training for Weakly Supervised Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
46. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
48. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
49. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
50. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1525–1535. [[CrossRef](#)]
51. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [[CrossRef](#)]
52. Dollár, P.; Zitnick, C.L. Fast Edge Detection Using Structured Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1558–1570. [[CrossRef](#)] [[PubMed](#)]
53. Doyle, W. Operations useful for similarity-invariant pattern recognition. *J. ACM* **1962**, *9*, 259–267. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.