Multimedia Systems
https://doi.org/10.1007/s00530-022-00944-4

SPECIAL ISSUE ARTICLE

Asymmetric exponential loss function for crack segmentation

Fan Liu[1,2,3] · **Junfeng Wang**[1,3] · **Delong Chen**[1,3] · **Chunmei Shen**[1,3] · **Feng Xu**[1,3]

Received: 9 January 2022 / Accepted: 11 April 2022

Abstract

Real-time and effective crack detection on public facilities is significant in maintaining the facilities even saving lives. Recent methods mostly explore the impact of model structures but neglect the impact from the loss functions. In this paper, we concentrate on analyzing the loss functions during the training process of crack segmentation tasks and propose an Asymmetric Exponential Loss Function (AELF) that addresses two key challenges, sample biases and data set biases. For the sample biases, AELF adopts an exponential loss function, thus can assign higher weights to the 'hard' samples, making the models concentrate on the crack details. For the data set biases, AELF leverages asymmetric protocol to balance the inevitable False Positive and False Negative samples. We conduct extensive experiments on three data sets of road, dam, and wall collected from real scenes. The impressive performances reveal the effectiveness of our proposed Asymmetric Exponential Loss Function.

Keywords Crack detection · Semantic segmentation

1 Introduction

Under continuous heavy pressure, cracks may appear on the structures, e.g., roads, bridges, and dams. In most cases, the appearance of cracks implies material fatigue and potential danger. Detecting and localizing the cracks in these structures in time is critical for maintaining public safety and avoiding economical losses. However, the cracks are always very small compared with the entire structure. This makes manual examination extremely costly and inefficient. Therefore, efficient and reliable crack segmentation methods are required to make early warnings. A formal procedure composed of collecting data with sensors installed on cars or unmanned aerial vehicles, analyzing them automatically, and alerting if cracks are detected, can severely reduce the disasters caused by cracks. Over years, researchers in the computer vision domain have made many efforts to develop efficient image processing techniques and improve the performance of crack detection [21, 37, 63]. The major difficulty of crack detection is caused by the presence of noise, e.g., textural patterns, road lanes, shadows, etc. Researchers have applied various image processing techniques to solve this issue, including multi-scale filtering [19, 59], histogram equalization [9, 10], adaptive thresholding [17, 26], wavelet transform [28, 30], anisotropy measurement [38, 57] and Canny operator [46, 56], etc. However, these crack detection methods focus on the traditional image processing techniques, hardly analyze images in a semantic level, thus usually failing to handle the complex noises.

In recent years, due to the strong learning ability, deep learning models [12–15, 22, 24, 29, 33–35, 43, 50, 51] has shown great success in various artificial intelligence applications. Convolutional Neural Networks (CNN) [27] have outperformed traditional methods by a large margin in a series of computer vision tasks, including image classification [11], object detection [66], and semantic segmentation [36]. Some of these methods that were originally designed for general semantic segmentation tasks can be directly applied to perform crack detection. For example, Fully Convolutional Network (FCN)-based [16, 60], SegNet-based [45], Mask-RCNN-based [4], YOLO-based [39], and U-Net-based [53] crack detection models have been validated to

✉ Fan Liu
fanliu@hhu.edu.cn

1 Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, Jiangsu, China

2 Science and Technology on Underwater Vehicle Technology Laboratory, Harbin, Heilongjiang, China

3 College of Computer and Information, Hohai University, Nanjing, Jiangsu, China

Published online: 07 May 2022

be effective. Despite directly applying existing models to perform crack detection, researchers have also attempted to design adaptive modifications, such as from the perspective of receptive field [8, 31, 54, 58], integrating Bayesian fusion model [18], developing multi-task learning scheme [1, 41, 49, 52, 62, 65], etc.

Deep learning learns knowledge from data under the guidance of loss functions. Hence, the choice of the loss function impacts the effectiveness of the models. Most existing methods utilize the classic loss functions in the process of use, but there are no further explorations on how the loss functions work. Intuitively, the loss functions should fit the distribution of the training data, and fix the biases and noises of them, thus can capture the perfect model. However, there are two obvious observations in the recent training process. First, from the data point of view, some data are easier to be understood by the model, while others are difficult to understand, but most of the existing loss functions are given the same weight; second, from the data set point of view, the characteristics of different data sets themselves Different, some data set cracks are easy to be detected, and some are easily misjudged by noise, but most of the existing loss functions do not consider the characteristics of the data set. Therefore, we term these observations as 'sample biases' and 'data set biases', and propose an Asymmetric Exponential Loss Function (AELF) to address these biases. For the sample biases, AELF adopts an exponential loss function, thus can assign higher weights to the 'hard' samples, making the models concentrate on the crack details. For the data set biases, AELF leverages asymmetric protocol to balance the inevitable False Positive and False Negative samples. We conduct extensive experiments on three data sets of road, dam, and wall collected from real scenes. The impressive performances reveal the effectiveness of our proposed Asymmetric Exponential Loss Function.

Summarily, the contributions of this paper are:

1. We point out the existence of sample biases during the training process and propose an exponential loss function to make the models concentrate on the hard details.
2. We note the different characteristics of different data sets and propose an Asymmetric Exponential Loss Function (AELF) to balance the inevitable False Positive and False Negative samples.
3. We conduct extensive experiments on five data sets. The results demonstrate the superiority of the proposed AELF loss function.

The rest of this paper is organized as follows. Section 2 reviews related work. Our proposed AELF loss function is described in Sect. 3. Comparative experiments and ablation studies are presented in Sect. 4. We finally conclude our paper in Sect. 5.

2 Related works

Zakeri et al. [63], Hsieh et al. [21], and Munawar et al. [37] have presented comprehensive reviews summarizing different approaches for crack detection. In general, crack detection can be done in three different ways: (1) patch classification, (2) object detection, and (3) image segmentation. Patch classification is the most straightforward way to apply a CNN to crack detection. This type of method first crops the camera captured images into many small patches, then feeds these patches to CNN and gets binary predictions indicating whether there is a crack instance within the patch. For this type of method, one common issue is that the contextual information between neighbourhood patches is ignored. Moreover, fixed patch size cannot handle multi-scale inputs. It is hard to decide a unified patch size when the input images are captured from different distances. Object detection-based crack detection can solve this problem, for example, using Feature Pyramid Network (FPN). However, it cannot provide quantified information on crack (e.g., the width of a crack instance). Comparatively, image segmentation [2, 20, 32, 40, 48, 55, 64]-based crack detection can not only utilize both local and global contextual information but also provide pixel-wise localization of the crack, which would be useful for further analysis.

Early crack detection methods are based on traditional image processing methods [9, 10, 17, 19, 26, 28, 30, 38, 46, 56, 57, 59], but their learning abilities are limited. With the rise of deep learning, the research community began to pay more attention to deep learning-based feature extraction approaches. Researchers successfully implemented AlexNet-based [25], FCN-based [16, 60], SegNet-based [45], Mask-RCNN-based [4], YOLO-based [39], and U-Net-based [53] crack detection models. It is worth noting that crack detection has its unique characteristics against generic computer vision tasks, so adaptive modifications on these deep models can further improve crack detection performances. Wang et al. [54] used dilated convolutions to enlarge the receptive field and utilize more contextual information. Similarly, Yang et al. [58] proposed Feature Pyramid and Hierarchical Boosting Network (FPHBN) to integrate semantic information of multi-level features. Liu et al. [31] proposed a DeepCrack model that can aggregate multi-scale and multi-level features, and Asadi et al. [3] found that assembling multiple branches of the DeepCrack model can further improve the performance.

Another line of research is combining multiple learning tasks to improve the crack detection models. Qu et al. [41] used an image classification task to improve the feature learned by CNN. Cha et al. [8] and Yao et al.

[61] detected cracks in a patch classification manner. Based on the patch classification framework, Ali et al. [54] added a sliding window-based crack localizing module after the classifier to provide additional information about the crack size, Yusef et al. [62] further combined a classification module to tell the type of the crack (transverse or longitudinal). Zhang et al. [65] proposed an APLCNet, which integrates classification, semantic segmentation, and instance segmentation into a single model.

3 Methods

3.1 Problem formulation

Suppose we have a data set $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$ with N samples, where X_i and Y_i are the ith input image and the corresponding binary crack annotation. For each pixel of Y_i, zero represents background, while one represents crack instance. For visualization purpose, in this paper, zero and one will be, respectively, plotted as black and white in the figures. Our goal is to learn a function f, which is a Convolutional Neural Network (CNN), that predicts Y_i from X_i accurately. In other words, the crack detection can be written as $\hat{Y}_i = f(X_i)$ and we expect low prediction error $d_i = \hat{Y}_i - Y_i$ for each i. Going forward, the subscripts i are omitted for simplicity. In the following, we will first introduce the architectures of the crack prediction network in detail (Sect. 3.2), then describe the learning objectives and the optimization procedure (Sect. 3.3) .

3.2 Crack prediction network

The network structure of this article is shown in Fig. 1, which consists of three parts: a U-Net backbone with an encoder network E, a decoder network D, and a prediction head g. The input image sequentially goes through E, D and g. Therefore, the procedure of the crack prediction network can be written as

$$\hat{Y} = f(X) = g(D(E(X))) \tag{1}$$

Encoder network The encoder network E has four downsample stages with convolution and pooling layers, which analyzes the contextual pixel information in the image to obtain the semantic feature. In each stage, the input tensor is first go through two 3×3 convolutional layers with ReLU activation. Same padding is applied in convolutional layers to preserve the spatial resolution. The output of convolutional layers is max-pooled with a kernel size of 2×2, therefore, reducing the spatial resolution by a half.

As shown in Fig. 1, the resolution of the input image is 512×512. The encoder network sequentially downsample the inputs into 256×256, 128×128, 64×64, and 32×32. At the same time, the channel of feature maps grows from 64 to 128, 256, 512, and 1024. Let conv(·) denote the two 3×3 convolutional layers, and let pool(·) represent the max-pooling layer. The L is the total number of stages (in the case of this method, the L is set to 5). The output feature maps of each stage of the encoder E can be formulated as follows:

$$X_E^l = \begin{cases} \text{conv}(X), & l = 1 \\ \text{conv}(\text{pool}(X_{\text{en}}^{l-1})), & l = 2, \ldots, L \end{cases} \tag{2}$$

Decoder network The decoder network D consists of four upsample stages. It recovers the same resolution of the given

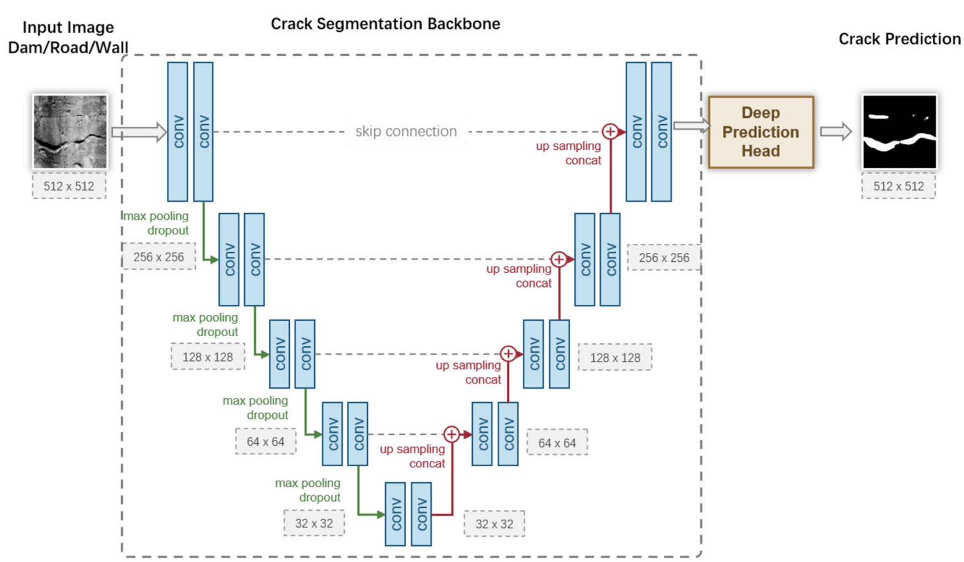


Fig. 1 Structure of crack prediction network

input image. In each stage, a skip connection is built with the corresponding stage in the encoder. The output tensor of the former stage of the decoder are upsampled and concatenated along the channel axis with the output tensor of the same stage of the encoder. The upsampling is performed by doing nearest interpolation. Then, two 3×3 convolutions with ReLU activation and same padding is applied. Let upsample$(\cdot)$ denote the upsampling operation and let the $\oplus$ represent the concatenate operation, the output tensor of each stage of the decoder D can be formulated as follows:

$$X_D^l = \begin{cases} X_E^l, & l = L \\ \text{conv}(X_E^l \oplus \text{upsample}(X_D^{l+1})), & l = 1, \dots, L-1. \end{cases} \tag{3}$$

Deep prediction head Conventionally, the prediction head g is a single 1×1 convolutional layer, as shown in the upper of Fig. 2. We argue that such shallow structure is not sufficient to capture complex semantic information in the decoder's feature map. As in the bottom of Fig. 2, here we modify the shallow prediction head into a deep prediction head structure. The deep prediction head consists of four 3×3 convolutional layers with ReLU activation and an 1×1 layer with sigmoid activation. The channel of feature maps are, respectively, 64, 64, 64, 2, and 1. After the sigmoid activation of the final 1×1 convolutional layer, the feature map is converted to a single channel 512×512 tensor $\hat{Y}$ in range of (0, 1), which matches the crack annotation Y.

The deep prediction head benefits this network, since it extracts and fuses richer semantic information from the feature maps of the decoder. We will demonstrate this point empirically in the experiment section. Formally, the prediction head g reads the feature maps from the decoder and gives the crack prediction $\hat{Y}$:

$$\hat{Y} = g(X_D^1). \tag{4}$$

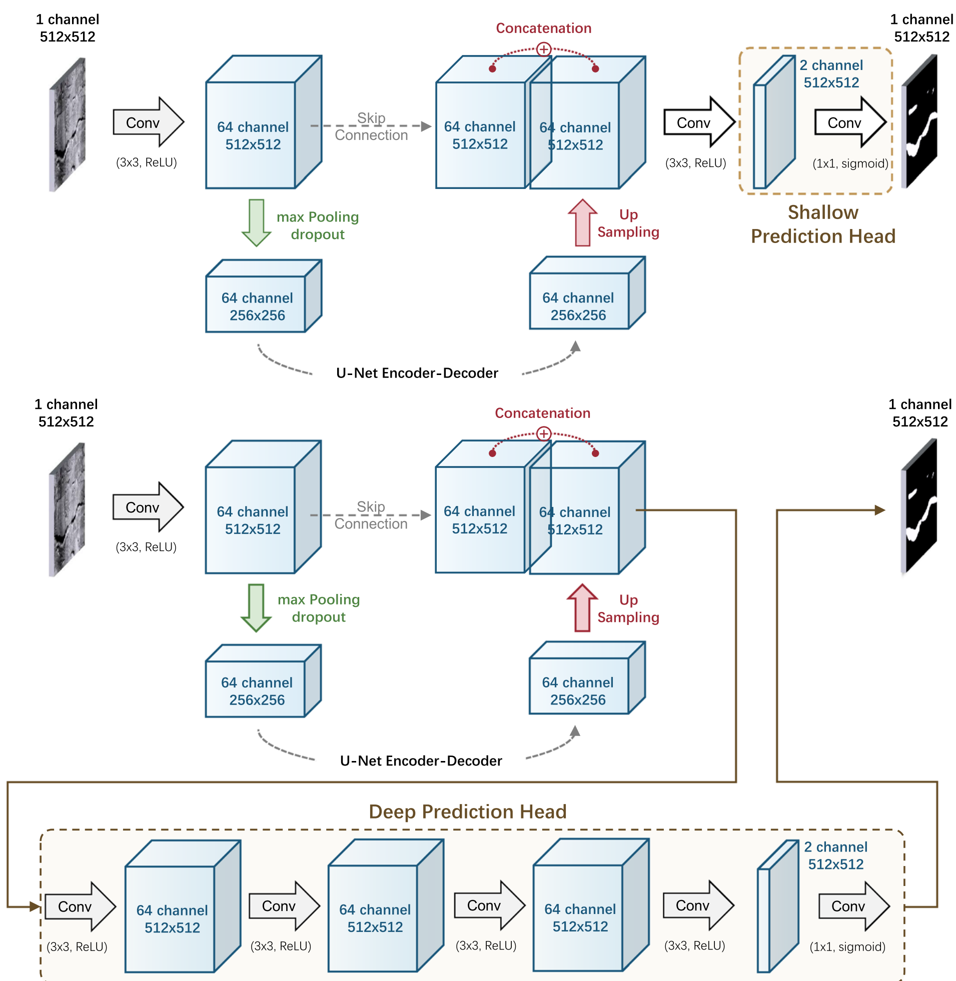


Fig. 2 Detailed structure of shallow (upper) and deep (bottom) prediction heads

3.3 Asymmetric exponential loss function

In this section, we introduce the learning objective of our crack detection model. Recall that our goal is to minimize the prediction error:

$$d = \hat{Y} - Y. \tag{5}$$

This goal can be achieved by adopting L_1 loss, L_2 loss, or Binary Cross Entropy (BCE) loss to train the crack detection model with back-propagation. However, these methods will encounter two key challenges. The first challenge is related to the noise in the input images, which is usually caused by textural patterns, road lanes, shadows, etc. These pixels are very hard to classified correctly, and thus significantly increase the difficulty of crack detection. Second, L_1 loss, L_2 loss, and BCE loss treat false positive (false prediction of crack instance for background pixels) and false negative (false prediction of background for crack instances pixels) equally, since they are symmetrical. Importantly, we found that the crack detection data set is highly imbalanced. As shown in Fig. 3, we compare the data set statistics of our constructed data set with other published data sets. We count the total number of pixel that is annotated as crack instance (one) and background (zero). The number of crack instance pixel is normalized to 1.0 across different data sets for the ease of comparison. As we can see, three crack detection data sets are significantly more imbalanced than other data sets. For HHU-crack-*road* data set, the ratio of crack instance v.s. background reaches even 1:64. When train a crack detection model on such imbalanced data set with traditional symmetrical losses (L_1, L_2, or BCE), huge amount of background pixels may overwhelm the model, leading to overly conservative prediction result.

To address the above two challenges, we first propose a novel Exponential loss function $\mathscr{L}_{\mathrm{Exp}}$. Subsequently, it has been improved on this basis, introducing asymmetry, and proposes an Asymmetric Exponential (AELF) loss function $\mathscr{L}_{\mathrm{AELF}}$, which is defined as

$$\mathscr{L}_{\mathrm{Exp}} = \begin{cases} \exp(\alpha \mid d \mid^3 + \beta d^2), & d > 0 \\ \exp(\alpha \mid d \mid^3 + \beta d^2), & d \leq 0 \end{cases} \tag{6}$$

$$\mathscr{L}_{\mathrm{AELF}} = \begin{cases} \exp(\alpha \mid d \mid^3 + \beta d^2 + \gamma_1 d), & d > 0 \\ \exp(\alpha \mid d \mid^3 + \beta d^2 + \gamma_2 d), & d \leq 0. \end{cases} \tag{7}$$

To give an intuitive understanding of $\mathscr{L}_{\mathrm{Exp}}$ loss function, we plot the loss curve of $\mathscr{L}_{\mathrm{Exp}}$ and its derivative w.r.t a certain pixel of the model prediction $\hat{Y}$ across different prediction error d in Fig. 4, comparing it with traditional L_1, L_2, and BCE losses. The derivative of $\mathscr{L}_{\mathrm{Exp}}$ is *exponentially* growing with d, while the derivative of L_2 loss is *linearly* growing, and the derivative of BCE loss grows *too late*. These properties (early growing + exponentially growing) leads to the following advantage of the $\mathscr{L}_{\mathrm{Exp}}$. When the difference between the prediction and groundtruth is small, e.g., $d < 0.5$, the model's prediction can be binarized to the same as groundtruth, so the sample can be considered to be an *easy* sample. Relatively smaller volume of gradient is produced in this case, so that the model can pay more attention to the harder samples. When $d > 0.5$, the model prediction and groundtruth belong to different categories. Under such situation, the sample can be considered as *hard* sample. The $\mathscr{L}_{\mathrm{Exp}}$ produces significantly larger value of loss, so that the model will pay the most attention to these cases during training.

$\mathscr{L}_{\mathrm{Exp}}$ effectively solves the performance deviation caused by hard samples, but still cannot solve data set bias due to data imbalance. Therefore, in this case, we introduce asymmetry into $\mathscr{L}_{\mathrm{Exp}}$ and propose $\mathscr{L}_{\mathrm{AELF}}$. In $\mathscr{L}_{\mathrm{AELF}}$, we use γ_1 and γ_2 to specify the degree of asymmetry. Larger γ_1 and γ_2 leads to more punishment to false positive, while smaller γ_1 and γ_2 leads to more punishment to false negative. We plot the curve of $\mathscr{L}_{\mathrm{AELF}}$ under different d in Fig. 5. For simplicity, we tried $\gamma_1, \gamma_2 \in \{-0.07, 0.03, 0, 0.03, 0.07\}$. As we can see, positive values of γ_1 and γ_2 tilt the curve to the left, while negative values tilt the curve to the right.

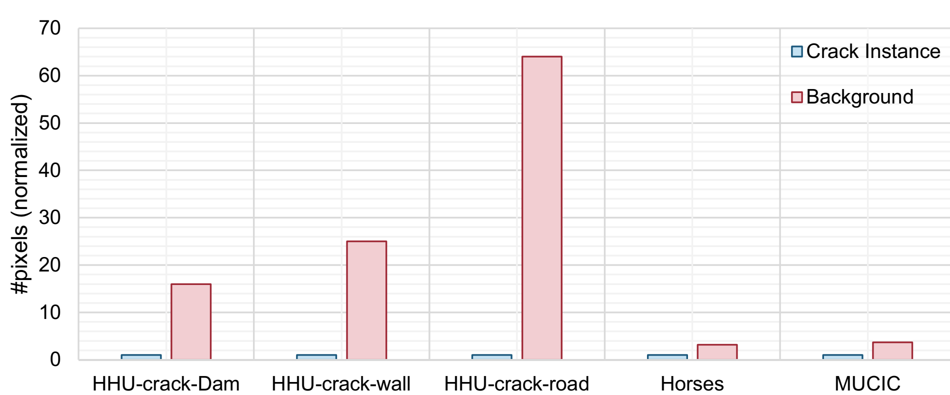


Fig. 3 Imbalance ratio (total number of pixels of crack instance vs. background) of our constructed data sets and other public data set

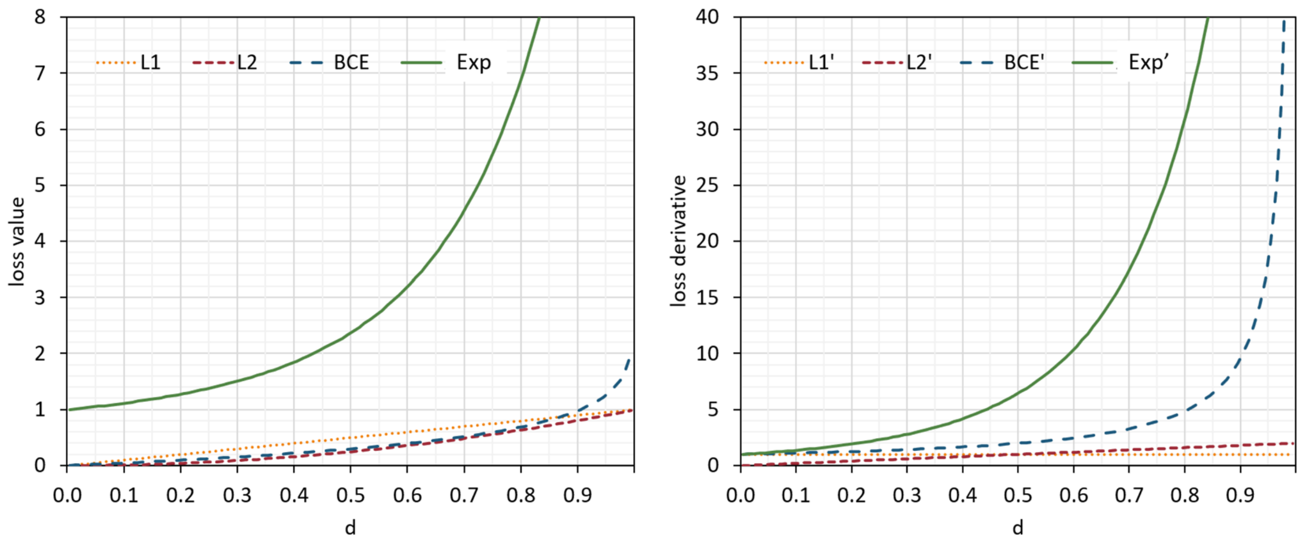


Fig. 4 Loss curve of $\mathscr{L}_{\mathrm{Exp}}$ and its derivative w.r.t. the model prediction across different d, compared with traditional L_1, L_2, and BCE losses. Here in $\mathscr{L}_{\mathrm{Exp}}$, α is set to 1 and β is set to 0.7

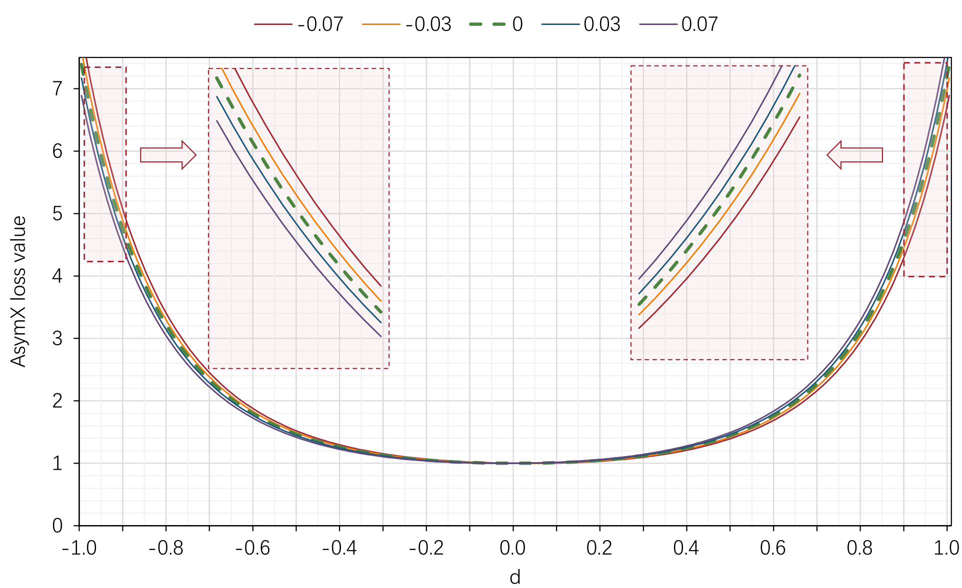


Fig. 5 Loss curve of $\mathscr{L}_{\mathrm{AELF}}$ under different d, γ_1 and γ_2. We tried $\gamma_1, \gamma_2 \in \{-0.07, 0.03, 0, 0.03, 0.07\}$

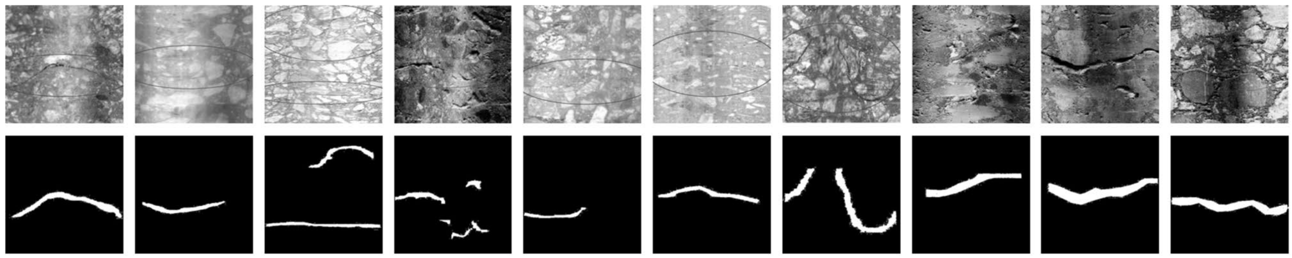
Since different data set has different ratio of imbalance, it is impossible to decide a universal value of γ_1 and γ_2. Therefore, on different data set, we first perform hyperparameter grid search on a separated validation set, and select the best γ_1 and γ_2 to train the model with training set.

4 Experiments

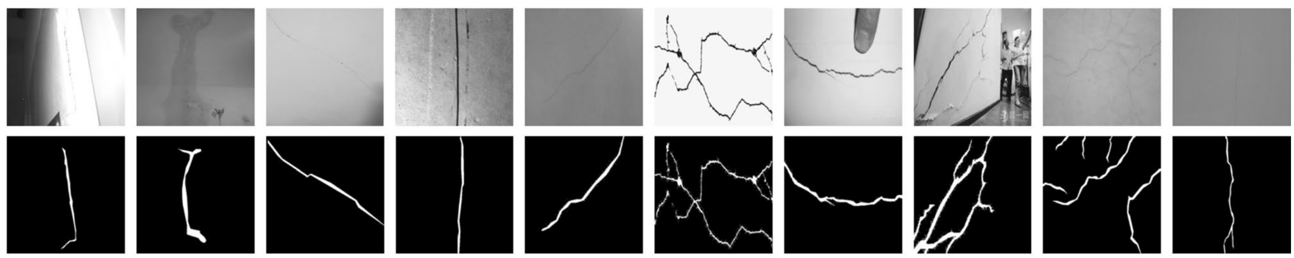
4.1 Data preparation

To evaluate the performance of our proposed method, five data sets (HHU-crack-*road*, HHU-crack-*dam*, HHU-crack-*wall*, Weizmann Horses [6] and MUCIC [47]) are used for experiment. Among them, HHU-crack-*road*, HHU-crack-*dam*, HHU-crack-*wall* are newly collected and expert-annotated data sets. Following are their brief description:

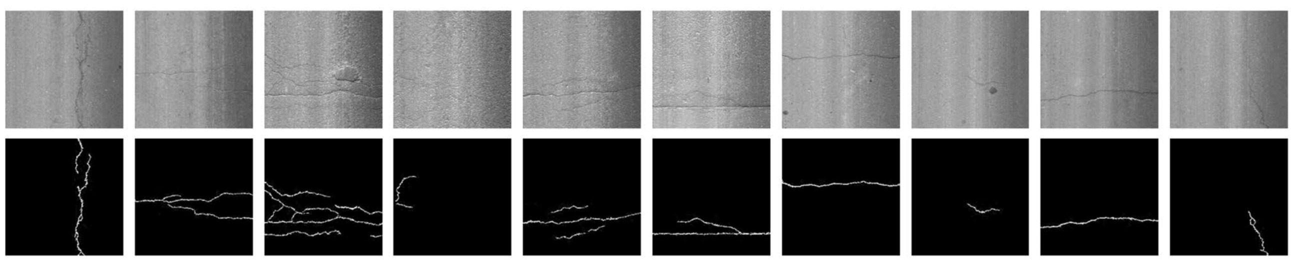
1. HHU-crack-*dam* includes 288 training images and 83 testing images. As shown in Fig. 6a, the images are collected from the daily dam monitoring of high arch dams. This data set is very challenging, since there are lots of complex noise in the image. The difference between cracks and background is small, making it more difficult to distinguish. To our best knowledge,



(a) HHU-crack-*dam*



(b) HHU-crack-*wall*



(c) HHU-crack-*road*

Fig. 6 Annotations of our constructed crack detection data sets

HHU-Crack-*dam* data set is the first crack detection data set for dams.

2. HHU-crack-*wall* includes 124 training images and 43 testing images. As shown in Fig. 6b, some images have less interference and cleaner environments, while some others are in noisy background.
3. As shown in Fig. 6c, HHU-crack-*road* includes 114 training images and 81 testing images. Though there are fewer noise interference, it is more imbalanced compared to HHU-crack-*dam* and HHU-crack-*wall*, as previously discussed in Sect. 3.3.

Except the these crack data sets, we also conducted experiments on the other public data sets to evaluate the generalization of our proposed method, including Weizmann Horses [6] (237 training images and 91 testing images) and MUCIC [47] (706 training images and 291 testing images).

4.2 Implementation details

We implement our proposed approach with Keras. The weights of the U-Net are initialized by He normal. To prevent over-fitting, various data augmentations are used, including random rotation, shifting shearing, and flipping. Nearest interpolation is used for all data augmentations. The model is trained for 30 epochs with a batch size of 1 and a learning rate of 1e−4. Adam optimizer is used for optimization. A NVIDIA GTX 1080 GPU is used to train the model. For each experiment, we independently train the model for 5 times, and report the averaged metric value.

4.3 Evaluation metrics

To evaluate the and compare the performances of different approaches, we use the precision (P_r), recall (R_e), and F-Measure [7, 57] (F_{score})as the measurements:

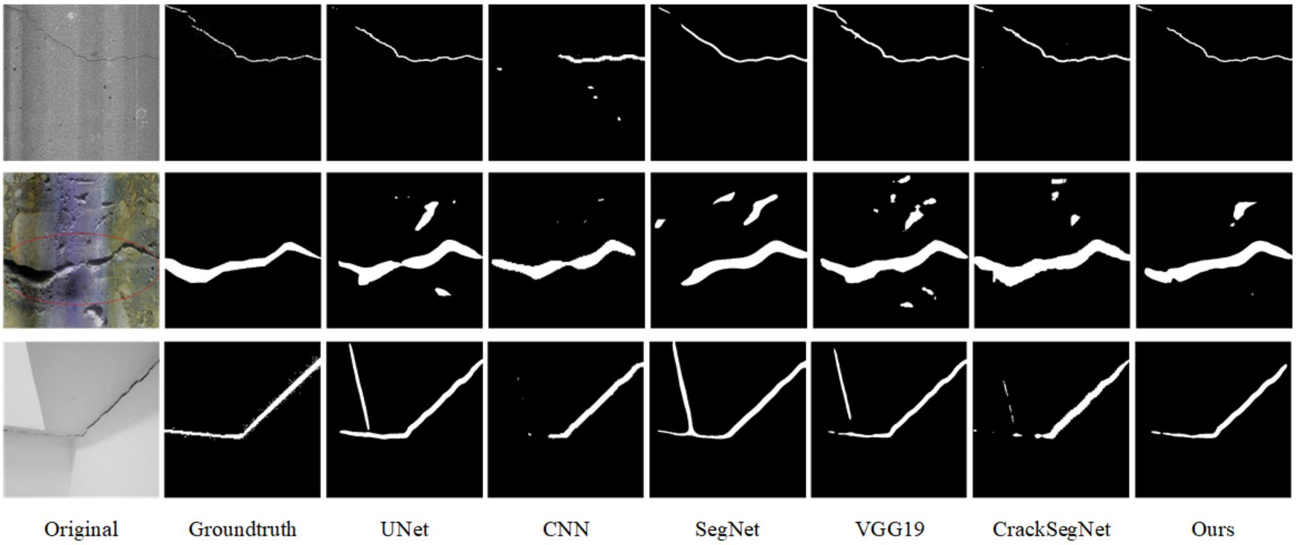


Fig. 7 Crack detection results of different methods on three crack data sets

$$P_r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{8}$$

$$R_e = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{9}$$

$$F_{\mathrm{score}} = \frac{(1+\beta^2) \times P_r \times R_e}{\beta^2 \times P_r + R_e} \tag{10}$$

where TP, FP, FN represent the number of true positives, false positives, and false negatives, the hyperparameter β^2 is set to 0.3 following common experiment setting. In addition, we also use precision–recall curve to evaluate the performance.

4.4 Performance comparison

A total of Five segmentation models (CNN [25], SegNet [5], VGG19 [44], CrackSegNet [42], and Semantic FPN [23] with a ResNet-50 backbone) are selected for experiments in this paper. We first present the comparison results on our constructed crack data sets HHU-crack-*road*, HHU-crack-*dam*, HHU-crack-*wall*, then report the comparison results on the public data sets Weizmann Horses [6] and MUCIC [47].

4.4.1 Crack detection

We first show qualitative comparison of crack prediction results in Fig. 7. It can be seen that our method is the most robust to noise inference. Tables 1, 2, and 3 show the quantitative results on HHU-crack-*road*, HHU-crack-*dam*, and HHU-crack-*wall*. We also show the percentage of improvement on F_{score} for comparison. It can be seen that, on HHU-crack-*road*, the F_{score} of our method reaches the best result of 80.72%. The F_{score} of UNet, CNN, SegNet, VGG19 and CrackSegNet are, respectively, 5.53%, 32.24%, 5.53%, 14.07%, and 11.11% lower than ours. On HHU-crack-*dam* data set, our method still achieves the best result, with an

Table 1 Results of P_r, R_e and F_{score} on HHU-Crack-*road* data set

Method	P_r	R_e	F_{score}	Our improvement
CNN	0.4921	0.6051	0.4848	+ 32.24%
SegNet	0.7577	0.7445	0.7519	+ 5.53%
VGG19	0.6420	0.8347	0.6665	+ 14.07%
CrackSegNet	0.6553	0.9079	0.6961	+ 11.11%
UNet	0.7479	0.8122	0.7519	+ 5.53%
FPN-R50	0.8846	0.5735	0.7862	+ 2.6%
Ours	0.8561	0.7207	**0.8072**	–

Bold value indicates the best performing method

Table 2 Results of P_r, R_e and F_{score} on HHU-Crack-*dam* data set

Method	P_r	R_e	F_{score}	Our improvement
CNN	0.5799	0.5341	0.5410	+ 5.41%
SegNet	0.5253	0.6642	0.5390	+ 5.61%
VGG19	0.6097	0.5051	0.5546	+ 4.05%
CrackSegNet	0.5387	0.6598	0.5481	+ 4.70%
UNet	0.6138	0.5434	0.5739	+ 2.12%
FPN-R50	0.5656	0.5141	0.5528	+ 7.10%
Ours	0.6971	0.4809	**0.5951**	–

Bold value indicates the best performing method

Table 3 Results of P_r, R_e and F_{score} on HHU-Crack-*wall* data set

Method	P_r	R_e	F_{score}	Our improvement
CNN	0.6329	0.4993	0.5238	+ 8.69%
SegNet	0.5296	0.6930	0.5259	+ 8.48%
VGG19	0.6205	0.6267	0.5649	+ 4.58%
CrackSegNet	0.4746	0.7822	0.4981	+ 11.26%
UNet	0.5409	0.7584	0.5396	+ 7.11%
FPN-R50	0.6805	0.4599	0.6127	–
Ours	0.6820	0.5987	**0.6107**	–

Bold value indicates the best performing method

F_{score} value of 59.51%. Compared to UNet, CNN, SegNet, VGG19 and CrackSegNet, there are, respectively, 2.12%, 5.41%, 5.61%, 4.05% and 4.70% performance improvement on F_{score}. On the HHU-crack-*wall* data set, our method outperforms other methods with an F_{score} value of 61.07%. Compared to UNet, CNN, SegNet, VGG19, and CrackSegNet, F_{score} of ours increases by 7.11%, 8.69%, 8.48%, 4.58% and 11.26%, respectively. In sum, the experimental results indicate that our method outperforms the other compared crack detection methods.

Figure 8 shows the Precision–Recall (PR) curves of the all the six methods on the these crack data sets. It can be seen that, on HHU-crack-*road*, our method holds a curve most close to the up-right corner, and achieves the best precision and recall values. On HHU-crack-*dam* data set, the performance of our method is slightly worse than SegNet, but still better than other methods. On HHU-crack-*wall* data set, the performance of our method is better than all the other methods. The comparative advantage on PR curves further demonstrate that the effectiveness of the proposed method.

4.4.2 Semantic segmentation

This group of experiments is to verify the performance of the proposed method on public data sets. Weizmann Horses [6] and MUCIC [47] are selected to test the generalization ability of our method. The visualized crack prediction results are shown in Fig. 9. On Weizmann Horses data set, our method is much robust to noise compared with other methods. The output segmentation result is complete and accurate. On MUCIC data set, our result is similar to other method results but can provide clearer boundaries.

Tables 4 and 5 show the quantitative results. It can be seen that, the F_{score} of our method has a 8.34% improvement compared to that of UNet on Weizmann Horses. Moreover, our method increases F_{score} by 7.95%, 5.41%, 10.68%, and 12.57% compared to CNN, SegNet, VGG19, and CrackSegNet, respectively. On MUCIC, our method increases F_{score} by 0.33%, 9.11%, 6.92%, 5.30%, and 4.73% compared to

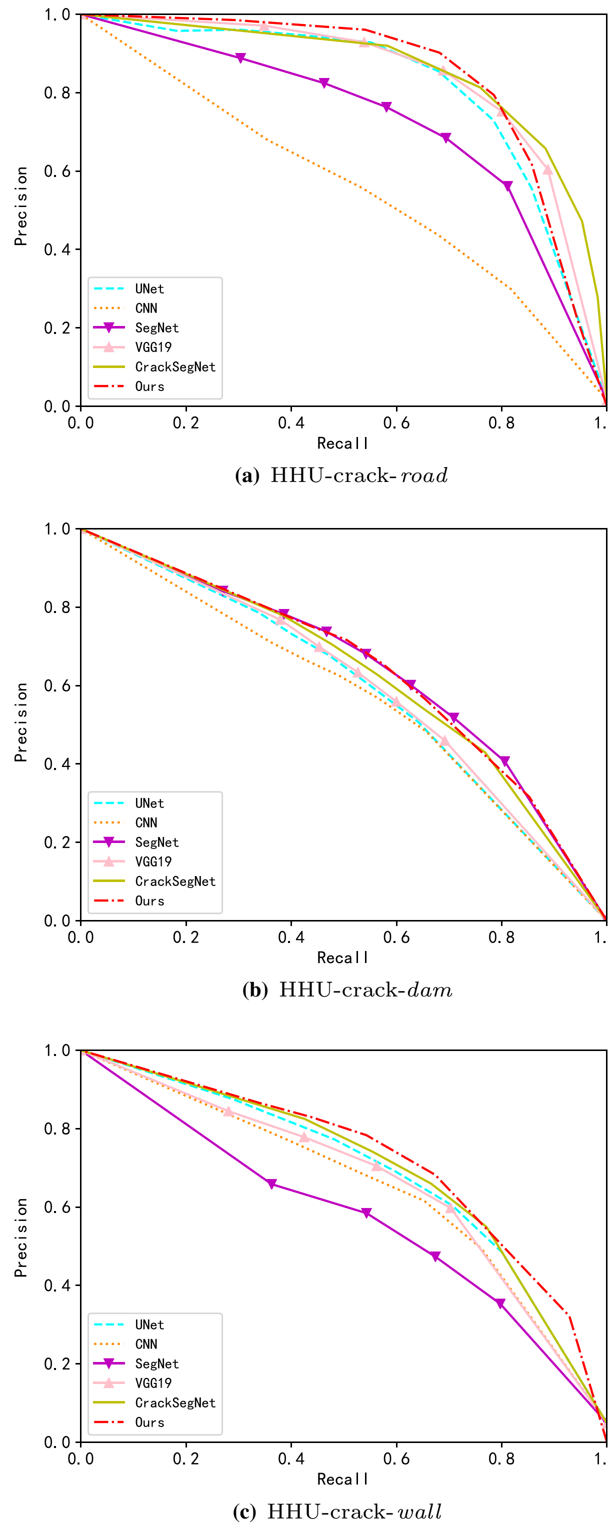


Fig. 8 PR curves on three crack data sets

UNet, CNN, SegNet, VGG19, and CrackSegNet, respectively. These results comprehensively show that our method can also obtain the best results on public data sets, indicating good generalization ability of the proposed method.

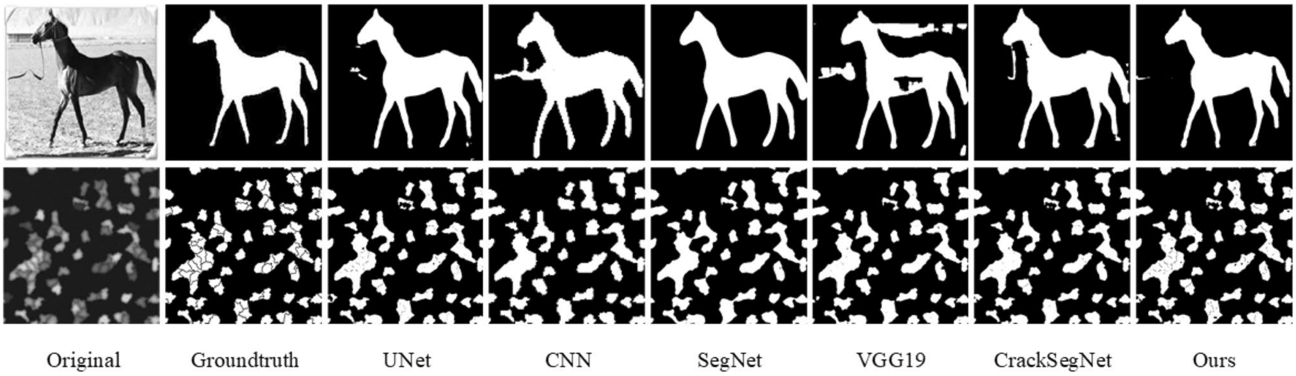


Fig. 9 Crack detection results of different methods on Weizmann Horses and MUCIC

Table 4 Results of P_r, R_e and F_{score} on Weizmann Horses data set

Method	P_r	R_e	F_{score}	Our improvement
CNN	0.7899	0.9411	0.8171	+ 7.95%
SegNet	0.8128	0.9693	0.8425	+ 5.41%
VGG19	0.7586	0.9422	0.7898	+ 10.68%
CrackSegNet	0.7266	0.9916	0.7709	+ 12.57%
UNet	0.7794	0.9745	0.8132	+ 8.34%
Ours	0.8848	0.9449	**0.8966**	–

Bold value indicates the best performing method

Table 5 Results of P_r, R_e and F_{score} on MUCIC data set

Method	P_r	R_e	F_{score}	Our improvement
CNN	0.8325	0.9572	0.8582	+ 9.11%
SegNet	0.8614	0.9487	0.8801	+ 6.92%
VGG19	0.8811	0.9504	0.8963	+ 5.30%
CrackSegNet	0.8948	0.9275	0.9020	+ 4.73%
UNet	0.9372	0.9768	0.9460	+ 0.33%
Ours	0.9472	0.9568	**0.9493**	–

Bold value indicates the best performing method

Figure 10 shows the PR curves of this group of experiments. It can be seen that our method is the best along with SegNet and CrackSegNet on Horse data set. Our method has the best performance compared with other methods on MUCIC. Which once again prove the superiority of the proposed method.

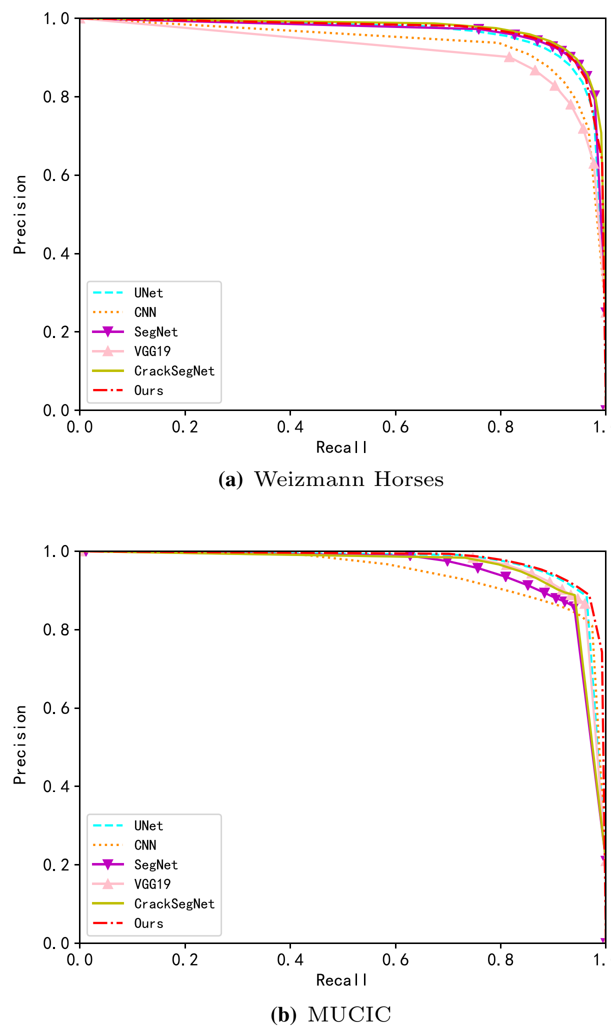


Fig. 10 PR curves on Weizmann Horses and MUCIC data set

4.5 Ablation study

To test the effectiveness of each module in our proposed method, ablation experiments are conducted. It can be observed from Table 6 that, compared with using shallow prediction head (Shallow Head), the F_{score} of using deep prediction head (Deep Head) has, respectively, increased by 1.31%, 0.28%, 2.50%, 5.24%, and 0.05%. When the exponential loss is used, the F_{score} increased by around 1% across all the five data set. However, they

Table 6 F_{score} results of ablation study

	HHU-crack-*dam*	HHU-crack-*wall*	HHU-crack-*road*	Horses	MUCIC
Shallow Head	0.5739	0.5396	0.7519	0.8132	0.9460
Deep Head	0.5814	0.5411	0.7707	0.8558	0.9465
Deep Head + Exp loss ($\gamma = 0$)	0.5887	0.5531	0.7878	0.8556	0.9492
Deep Head + AELF loss (Final)	0.5951	0.6107	0.8072	0.8966	0.9493

are, respectively, 0.64%, 5.76%, 1.94%, 4.10%, and 0.01% lower than our final model (Deep Head + AELF loss). The result indicates that the exponential loss can significantly improve the model's ability to identify hard samples, increase the model robustness against the data set bias, and consequently improve the model performance. Meanwhile, the asymmetry in the loss function can further enhance the model performance by further solving the sample bias. We also visualize the grid search process of γ_1 and γ_2 in Fig. 11. It shows that varying γ_1 and γ_2 significantly affect model performance. Most trails achieve better F_{score} than the default setting ($\mathscr{L}_{\text{Exp}}$) or $\gamma_1 = 0$ and $\gamma_2 = 0$, demonstrating the necessity of asymmetry in loss function and the effectiveness of hyperparameter grid search.

5 Conclusions

This paper proposed an AELF loss function for crack detection. The Asym-UNet contains a deep prediction head instead of a traditional shallow prediction head. The AELF loss encourages the model to focus more on hard samples. At the same time, it helps the model to handle data imbalance based on its asymmetry. We also presented three challenging crack detection data sets, namely, HHU-Crack-*road*, HHU-Crack-*wall*, and HHU-Crack-*dam*. They are more closed to real-world application scenarios. Experiments showed the effectiveness of our method. The performance of our AELF loss approach outperforms various comparisons.

Acknowledgements This work was partially funded by Natural Science Foundation of Jiangsu Province under Grant no. BK20191298, Research Fund from Science and Technology on Underwater Vehicle

HHU-crack-*road*

γ_2 \ γ_1	-0.07	-0.03	0	0.03	0.07
-0.07	0.7538	0.7839	0.7907	0.7987	0.8054
-0.03	**0.8072**	0.7807	0.7786	0.7926	0.7785
0	0.8066	0.7587	0.7878	0.7942	0.8047
0.03	0.7989	0.7413	0.7708	0.7829	0.8045
0.07	0.8053	0.797	0.7783	0.7866	0.7888

HHU-crack-*dam*

γ_2 \ γ_1	-0.07	-0.03	0	0.03	0.07
-0.07	0.5701	0.5729	0.5901	**0.5951**	0.5726
-0.03	0.5804	0.58	0.5832	0.5802	0.5733
0	0.5835	0.5741	0.5887	0.5853	0.5944
0.03	0.5676	0.5908	0.5646	0.5814	0.5937
0.07	0.5754	0.5791	0.5833	0.5897	0.5859

HHU-crack-*wall*

γ_2 \ γ_1	-0.07	-0.03	0	0.03	0.07
-0.07	0.607	0.5821	0.5637	0.4167	0.4309
-0.03	0.5448	0.5651	0.4127	0.548	0.5359
0	0.5368	0.5126	0.5531	0.5833	0.5524
0.03	0.5371	0.5148	0.5927	0.5027	0.5998
0.07	**0.6107**	0.5187	0.5971	0.5962	0.544

Horses

γ_2 \ γ_1	-0.07	-0.03	0	0.03	0.07
-0.07	0.8801	0.8811	0.8723	0.875	0.8587
-0.03	0.8875	0.8851	0.883	0.8863	0.8359
0	0.8896	**0.8966**	0.8556	0.876	0.8526
0.03	0.8216	0.8807	0.8729	0.8849	0.8861
0.07	0.8622	0.8442	0.8904	0.8628	0.8835

MUCIC

γ_2 \ γ_1	-0.07	-0.03	0	0.03	0.07
-0.07	0.9487	0.9463	0.9483	0.9481	0.9443
-0.03	0.9473	0.9478	0.9488	0.9473	0.9467
0	0.9473	0.9485	0.9492	0.9465	**0.9493**
0.03	0.949	0.9476	0.9432	0.9455	0.9489
0.07	0.9479	0.9463	0.9475	0.9483	0.9486

Fig. 11 Results of F_{score} using different γ_1 and γ_2. Blue/red represents higher/lower F_{score}, respectively (colour figure online)

Technology Laboratory (2021JCJQ-SYSJJ-LB06905),Water Science and Technology Project of Jiangsu Province under Grant no. 2021072 and National Natural Science Foundation of China under Grant nos. 61602150 and no. 61902110.

References

1. Ali, L., Valappil, N.K., Kareem, D.N.A., John, M.J., Al Jassmi, H.: Pavement crack detection and localization using convolutional neural networks (CNNs). In: 2019 International Conference on Digitization (ICD), pp. 217–221. IEEE, Piscataway (2019)
2. Araújo, R.L., de Araújo, F.H.D., Silva, R.R.V.: Automatic segmentation of melanoma skin cancer using transfer learning and fine-tuning. Multimedia Syst. **27**, 1–12 (2021)
3. Asadi, M., Sadeghi, M.T., Bafghi, A.Y.: A multi-classifier system for rock mass crack segmentation based on convolutional neural networks. In: 2021 26th International Computer Conference, Computer Society of Iran (CSICC), pp. 1–6. IEEE, Piscataway (2021)
4. Attard, L., Debono, C.J., Valentino, G., Di Castro, M., Masi, A., Scibile, L.: Automatic crack detection using mask R-CNN. In: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 152–157. IEEE, Piscataway (2019)
5. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
6. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop, pp. 46–46. IEEE, Piscataway (2004)
7. Boyd, K., Eng, K.H., Page, C.D.: Area under the precision-recall curve: point estimates and confidence intervals. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 451–466. Springer, Berlin (2013)
8. Cha, Y.-J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Comput. Aided Civ. Infrastruct. Eng. **32**(5), 361–378 (2017)
9. Chang, Y., Jung, C., Ke, P., Song, H., Hwang, J.: Automatic contrast-limited adaptive histogram equalization with dual gamma correction. IEEE Access **6**, 11782–11792 (2018)
10. Chen, C., Seo, H.S., Zhao, Y., Chen, B., Kim, J.W., Choi, Y., Bang, M.: Automatic pavement crack detection based on image recognition. In: International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving Data-Informed Decision-Making, pp. 361–369. ICE Publishing, London (2019)
11. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE, Piscataway (2009)
12. Du, X.-Y., Tang, J., Li, Z., Qin, Z.: Wheel: accelerating CNNs with distributed GPUs via hybrid parallelism and alternate strategy. In: Liu, Q., Lienhart, R., Wang, H., "Kuan-Ta" Chen, S.-W., Boll, S., Chen, Y.-P.P., Friedland, G., Li, J., Yan, S. (eds.) Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23–27, 2017, pp. 393–401. ACM, New York (2017)
13. Du, X.-Y., Yang, Y., Yang, L., Shen, F., Qin, Z., Tang, J.: Captioning videos using large-scale image corpus. J. Comput. Sci. Technol. **32**(3), 480–493 (2017)
14. Du, X., Liu, Q., Li, Z., Qin, Z., Tang, J.: Cauchy matrix factorization for tag-based social image retrieval. IEEE Access **7**, 132302–132310 (2019)
15. Du, X., Yang, X., Qin, Z., Tang, J.: Progressive image enhancement under aesthetic guidance. In: El-Saddik, A., Del Bimbo, A., Zhang, Z., Hauptmann, A.G., Candan, K.S., Bertini, M., Xie, L., Wei, X.-Y. (eds.) Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10–13, 2019, pp. 349–353. ACM, New York (2019)
16. Dung, C.V., et al.: Autonomous concrete crack detection using deep fully convolutional neural network. Autom. Constr. **99**, 52–58 (2019)
17. 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: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 474–479. IEEE, New York (2019)
18. Fang, F., Li, L., Rice, M., Lim, J.-H.: Towards real-time crack detection using a deep neural network with a Bayesian fusion algorithm. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 2976–2980. IEEE, New York (2019)
19. Guo, Y., Xiao, D.-H., Huang, Y.-G.: Mathematical morphology and multi-scale analysis for pavement crack detection. Highway. **63**(01), 31–34 (2018)
20. He, D., Xie, C.: Semantic image segmentation algorithm in a deep learning computer network. Multimedia Syst. **26**, 1–13 (2020)
21. Hsieh, Y.-A., Tsai, Y.J.: Machine learning for crack detection: review and model performance comparison. J. Comput. Civ. Eng. **34**(5), 04020038 (2020)
22. Hu, Q., He, Q., Huang, H., Chiew, K., Liu, Z.: Learning from crowds under experts' supervision. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 200–211. Springer, Cham (2014)
23. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6399–6408. IEEE, Piscataway (2019)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
25. Li, S., Zhao, X.: Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Adv. Civ. Eng. **2019**, 1–12 (2019)
26. Li, X., Jiao, S., Lu, X.: An improved Otsu method for image segmentation. Infrared Technology **2**(3), 164–166 (2007)
27. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans. Neural Netw. Learn. Syst. pp. 1–21 (2021)
28. Liang, S., Sun, B.: Using wavelet technology for pavement crack detection. In: ICLEM 2010: Logistics For Sustained Economic Development: Infrastructure, Information, Integration, pp. 2479–2484. ASCE, Fort Collins (2010)
29. Liu, A.-A., Zhou, H., Nie, W., Liu, Z., Liu, W., Xie, H., Mao, Z., Li, X., Song, D.: Hierarchical multi-view context modelling for 3D object classification and retrieval. Inf. Sci. **547**, 984–995 (2021)
30. Liu, T., Li, A., Ding, Y.: Application of wavelet analysis to structural damage identification. Earthquake Engineering and Engineering Vibration. **28**(2), 29–35 (2008)
31. Liu, Y., Yao, J., Lu, X., Xie, R., Li, L.: Deepcrack: a deep hierarchical feature learning architecture for crack segmentation. Neurocomputing **338**, 139–153 (2019)
32. Liu, Z., Han, K., Wang, Z., Zhang, J., Song, Y., Yao, X., Yuan, D., Sheng, V.S.: Automatic liver segmentation from abdominal ct volumes using improved convolution neural networks. Multimedia Syst. **27**(1), 111–124 (2021)
33. Liu, Z., Qian, P., Wang, X., Zhu, L., He, Q., Ji, S.: Smart contract vulnerability detection: from pure neural network to interpretable graph feature and expert pattern fusion. In: International Joint

Conference on Artificial Intelligence (IJCAI), pp. 2751–2759. Morgan Kaufmann, San Francisco (2021)
34. Liu, Z., Wang, Z., Zhang, L., Shah, R.R., Xia, Y., Yang, Y., Li, X.: Fastshrinkage: perceptually-aware retargeting toward mobile platforms. In: Proceedings of the 25th ACM international conference on Multimedia (ACM MM), pp. 501–509. ACM, New York (2017)
35. Liu, Z., Wu, S., Jin, S., Ji, S., Liu, Q., Lu, S., Cheng, L.: Investigating pose representations and motion contexts modeling for 3D motion prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 1–16 (2022)
36. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. IEEE, Piscataway (2015)
37. Munawar, H.S., Hammad, A.W.A., Haddad, A., Soares, C.A.P., Waller, S.T.: Image-based crack detection methods: a review. Infrastructures **6**(8), 115 (2021)
38. Nguyen, T.S., Begot, S., Duculty, F., Bardet, J.-C., Avila, M.: Pavement cracking detection using an anisotropy measurement. In: 11ème IASTED International Conference on Computer Graphics and Imaging (CGIM), pp. 80–87. Acta Press, Calgary (2010)
39. Nie, M., Wang, C.: Pavement crack detection based on yolo v3. In: 2019 2nd International Conference on Safety Produce Informatization (IICSPI), pp. 327–330. IEEE, New York (2019)
40. Olimov, B., Sanjar, K., Din, S., Ahmad, A., Paul, A., Kim, J.: FU-Net: fast biomedical image segmentation model based on bottleneck convolution layers. Multimedia Syst. **27**(4), 637–650 (2021)
41. Qu, Z., Mei, J., Liu, L., Zhou, D.-Y.: Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. IEEE Access **8**, 54564–54573 (2020)
42. Ren, Y., Huang, J., Hong, Z., Lu, W., Jun, Y., Zou, L., Shen, X.: Image-based concrete crack detection in tunnels using deep fully convolutional networks. Constr. Build. Mater. **234**, 117367 (2020)
43. Shu, X., Qi, G.-J., Tang, J., Wang, J.: Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In: Zhou, X., Smeaton, A.F., Tian, Q., Bulterman, D.C.A., Shen, H.T., Mayer-Patel, K., Yan, S. (eds.) Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM ’15, Brisbane, Australia, October 26–30, 2015, pp. 35–44. ACM, New York (2015)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR (2015)
45. Song, C., Wu, L., Chen, Z., Zhou, H., Lin, P., Cheng, S., Wu, Z.: Pixel-level crack detection in images using SegNet. In: International Conference on Multi-disciplinary Trends in Artificial Intelligence, pp. 247–254. Springer, Berlin (2019)
46. Sun, L., Xing, J., Xie, L., Wang, J.: An adaptive threshold based canny algorithm for crack detection. Microcomput. Appl. **36**(5), 35–37 (2017)
47. Svoboda, D., Ulman, V., Kováč, P., Šalingová, B., Tesařová, L., Koutná, I.K., Matula, P.: Vascular network formation in silico using the extended cellular potts model. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3180–3183. IEEE, New York (2016)
48. Ta, N., Chen, H., Lyu, Y., Wu, T.: BLE-Net: boundary learning and enhancement network for polyp segmentation. Multimedia Syst. **28**, 1–14 (2022)
49. Tang, J., Mao, Y., Wang, J., Wang, L.: Multi-task enhanced dam crack image detection based on faster R-CNN. In: 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), pp. 336–340. IEEE, New York (2019)
50. Tang, J., Shu, X., Li, Z., Jiang, Y.-G., Tian, Q.: Social anchor-unit graph regularized tensor completion for large-scale image retagging. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 2027–2034 (2019)
51. Tang, J., Shu, X., Qi, G.-J., Li, Z., Wang, M., Yan, S., Jain, R.C.: Tri-clustered tensor completion for social-aware image tag refinement. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1662–1674 (2017)
52. Wang, B., Zhao, W., Gao, P., Zhang, Y., Wang, Z.: Crack damage detection method via multiple visual features and efficient multi-task learning model. Sensors **18**(6), 1796 (2018)
53. Wang, J., Liu, F., Yang, W., Xu, G., Tao, Z.: Pavement crack detection using attention u-net with multiple sources. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), pp. 664–672. Springer, Berlin (2020)
54. Wang, L., Ye, Y., et al.: Computer vision-based road crack detection using an improved I-UNet convolutional networks. In: 2020 Chinese Control And Decision Conference (CCDC), pp. 539–543. IEEE, New York (2020)
55. Wang, Y., Zu, C., Ma, Z., Luo, Y., He, K., Wu, X., Zhou, J.: Patch-wise label propagation for MR brain segmentation based on multi-atlas images. Multimedia Syst. **25**(2), 73–81 (2019)
56. Xiao, Y., Li, J.: Crack detection algorithm based on the fusion of percolation theory and adaptive canny operator. In: 2018 37th Chinese Control Conference (CCC), pp. 4295–4299. IEEE, New York (2018)
57. Xu, W., Tang, Z., Zhou, J., Ding, J.: Pavement crack detection based on saliency and statistical features. In: 2013 IEEE International Conference on Image Processing, pp. 4093–4097. IEEE, New York (2013)
58. 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. **21**(4), 1525–1535 (2019)
59. Yang, H.-L., Cui, H.-E., Sun, H.-T., He, X.: Ordnance Test Center Baicheng. Method for pavement cracks detection based on multi-scale matrix filtering. Chin. J. Liq. Cryst. Disp. **31**(08), 778–783 (2016)
60. Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X.: Automatic pixel-level crack detection and measurement using fully convolutional network. Comput. Aided Civ. Infrastruct. Eng. **33**(12), 1090–1109 (2018)
61. Yao, G., Wei, F.-J., Qian, J.-Y., Wu, Z.-G.: Crack detection of concrete surface based on\newline convolutional neural networks. In: 2018 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 246–250. IEEE, New York (2018)
62. Yusof, N.A.M., Osman, M.K., Noor, M.H.M., Ibrahim, A., Tahir, N.M., Yusof, N.M.: Crack detection and classification in asphalt pavement images using deep convolution neural network. In: 2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 227–232. IEEE, New York (2018)
63. Zakeri, H., Nejad, F.M., Fahimifar, A.: Image based techniques for crack detection, classification and quantification in asphalt pavement: a review. Arch. Comput. Methods Eng. **24**(4), 935–977 (2017)
64. Zhang, J., Zhou, Y., Xia, K., Jiang, Y., Liu, Y.: A novel automatic image segmentation method for Chinese literati paintings using multi-view fuzzy clustering technology. Multimedia Syst. **26**(1), 37–51 (2020)
65. Zhang, Y., Chen, B., Wang, J., Li, J., Sun, X.: APLCNet: automatic pixel-level crack detection network based on instance segmentation. IEEE Access **8**, 199159–199170 (2020)
66. Zhao, Zhong-Qiu., Zheng, Peng, Shou-tao, Xu., Xindong, Wu.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)