Attack-inspired Calibration Loss for Calibrating Crack Recognition

Zhuangzhuang Chen, Qiangyu Chen, Jiahao Zhang, Zhiliang lin, Xingyu Feng, Jie Chen, Jianqiang Li *

The National Engineering Laboratory of Big Data System Computing Technology
Shenzhen University, Shenzhen 518060, China
{chenzhuangzhuang2016,2021152010, 2021152005, linzhiliang2022,fengxingyu2017}@email.szu.edu.cn,
chenjie@szu.edu.cn, lijq@szu.edu.cn

Abstract

Deep neural networks (DNNs) have substantially achieved high predictive accuracy in many vision tasks. However, we find that they are poorly calibrated for crack recognition tasks, as these DNNs tend to produce both under-confident and over-confident predictions in such safety-critical applications, thereby limiting their practical use in real-world scenarios. To address this issue, we propose a novel attack-inspired calibration loss (AICL) that explicitly regularizes class probabilities to be better confidence estimation. Specifically, we first propose the attack-inspired correctness estimation method (ACE) that aims to estimate the correctness degree of each sample via adversarial attacks. Then, we propose Correctness-aware Distribution Guidance, which starts from a distribution perspective that enforces the ordinal ranking of the predicted confidence referring to the estimated correctness degree. The proposed method can be conveniently implemented on top of any DNNs-based crack recognition model by serving as a plug-and-play loss function. To address the limited availability of related benchmarks, we collect a fully annotated dataset, namely, Bridge2024, which involves inconsistent cracks and noisy backgrounds in real-world bridges. Our AICL outperforms the state-of-art calibration methods on various benchmark datasets including CRACK2019, SDNET2018, and our BRIDGE2024.

Datasets — https://github.com/cheny124800/AICL

Introduction

Crack recognition has proved to be the critical stage in structure health monitoring (SHM) (Chen et al. 2023, 2024c,a). Recurring bridge collapses underline the importance of regular and thorough structural inspection (Crawford 2023). In this regard, cracks have a significant impact on the mechanical behavior of structures and their identification can reveal structural stress mechanisms, which is an important part of SHM to ensure structural safety and durability (Koch et al. 2015). Moreover, recognizing and repairing cracks before the onset of serious deterioration can ease the heavy maintenance cost. However, due to the low signal-to-noise ratios and inconsistent shapes (Yuan et al. 2020), crack recognition remains challenging in realistic environments (Nguyen et al. 2023). To address this, inspired by the great success of deep learning technology in general image and audio recognition tasks (Li et al. 2022; Mi et al. 2023, 2024; Li et al. 2024b), previous works mainly try to achieve high crack recognition accuracy by leveraging powerful Deep neural networks (DNNs) (Nguyen et al. 2023; Li et al. 2024a; Chen et al. 2024d). For example, (Li et al. 2023) adopt ResNet50 to achieve high accuracy in recognizing cracks. Moreover, a new Transformer-based Road crack identification system (Han et al. 2022) is designed to classify cracks and non-cracks images.

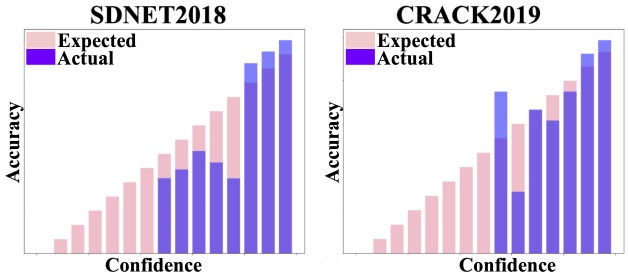

Figure 1: Reliability diagrams: crack recognition without calibration on SDNET2018 and CRACK2019 datasets. The diagram is expected to plot an identity function of accuracy with respect to confidence. It is worth noting that the deviation from a perfect diagonal represents the miscalibration, such as the gap between the blue and pink histogram.

Unlike these works, we argue that, as a safety-critical application, crack recognition should not only care about high accuracy but also accurate model confidence. For instance, a trustworthy crack recognition model should safely reject predictions with low confidence, however, if it mistakenly skips reviewing an incorrect prediction (i.e., a crack sample has been wrongly classified as the non-crack class with high confidence), it may lead to serious deterioration as those cracks can not be repaired in time. Hence, a high-quality confidence estimate from DNNs is required for practical crack recognition applications. However, it has not been studied in the crack recognition scenario, which gives rise to one fundamental question: ***Are the current crack recognition methods well-calibrated, i.e., providing a high-quality confidence estimation?***

*Corresponding Author (email: lijq@szu.edu.cn)

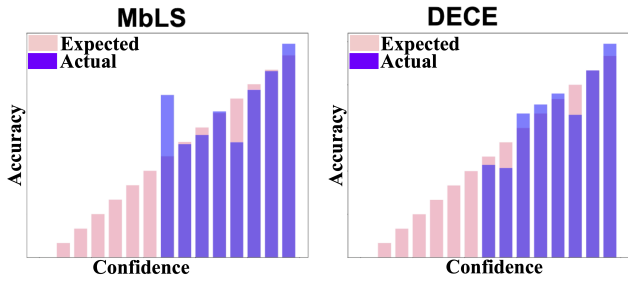

Figure 2: Reliability diagrams: crack recognition with the existing calibration methods on SDNET2018 dataset.

To answer the above question, we present experiments assessing the relationship between confidence and accuracy on crack recognition tasks. Surprisingly, as shown in Figure 1, we discover that existing DNNs are far distant from being well-calibrated in crack recognition scenarios, and more importantly, DNNs tend to be both under-confident and over-confident in their predictions, which is different from other general classification tasks that are often over-confident (Kumar, Sarawagi, and Jain 2018). DNNs being under-confident means that many predictions are distributed in the low-confidence range, and therefore, some cracks may be fail to recognized. Meanwhile, DNNs being over-confident indicates that many predictions are distributed in the high confidence range, and therefore, some non-crack samples may be classified into crack classes, thereby increasing the manual checking workload. This leads to another natural question: ***Are the existing confidence calibration methods flexible for crack recognition tasks?***

Generally speaking, the confidence calibration is to calibrate the outputs (also known as logits) of DNNs. Naturally, a typical way is to re-scale the logits of a trained model via a single temperature scaling parameter (Guo et al. 2017). However, these methods are architecture-dependent and heavily rely on a representative held-out validation set, which may have no access in complex real-world scenarios (Liu et al. 2022). Moreover, to achieve implicit model calibration, MbLS (Liu et al. 2022) is proposed to design margin constraint logit distances. By considering the differentiable surrogate for expected calibration error, DECE (Bohdal, Yang, and Hospedales 2023) is proposed to serve as a novel loss function and show its effectiveness in supervised classification tasks. Now, to answer the previous question, we present experiments on the BRIDGE2024 dataset with the above calibration methods including MbLS and DECE. Figure 2 shows that the current crack recognition model with the help of these confidence calibration methods is still far from well-calibrated. The reason is that the inconsistent cracks in varying sizes and shapes make DNNs tend to be under-confidence in the prediction, even if the prediction is right. Meanwhile, the noisy texture patterns of the surrounding background tend to make DNNs make an over-confident wrong prediction. Thus, a crucial question remains: ***How to calibrate the confidence predictions given by DNNs so as to make them more trustworthy for crack recognition tasks?***

To answer the above question, we first revisit the definition of a well-calibrated DNN: ***The predicted confidence is well-aligned with the likelihood of the sample being correctly classified.*** Thus, the key is to align the increase/ decrease trend of confidence and correctness. To achieve this, the essential question is how to estimate the correctness of the current prediction. Generally speaking, a high correctness of the current prediction indicates that the corresponding features are not vulnerable to attacks. Meanwhile, the features of inputs that are vulnerable to attacks have a small likelihood of being correctly classified. As shown in Figure 3, to find the mis-classified adversarial variants of input features, the features with low correctness require less rounds than those with high correctness. By exploiting this effect, the proposed attack-inspired correctness estimation method (ACE) is allowed to estimate the correctness degree of current predictions by performing adversarial attacks on feature embeddings.

Then, with the help of the ACE, one plausible guidance for confidence calibration is to enforce a sample that involves a high correctness degree (i.e., needs large adversarial attack rounds) to have larger predicted confidence than a sample that involves a small correctness degree (i.e., needs small adversarial attack rounds). However, such a ***sample-level guidance***, though simple, is overly restrictive as it relies on the success of the estimated correctness for each sample, leading to inferior crack calibration performance (see details in the experiment section). Instead, we propose ***Correctness-aware Distribution Guidance*** that consider the above guidance at a distribution level. Specifically, suppose that we have two distributions: (1) where samples in the first distribution involve high correctness; (2) where samples in the second distribution involve low correctness, then the expectation of predicted confidence with respect to the first distribution is supposed to be larger than that of the second distribution. Such guidance provides more flexibility for each sample during training, as it is not strictly dependent on the well-estimated correctness degree for each sample.

The contributions of this paper are four-fold:

- To the best of our knowledge, this is the first work to study the trustworthy problem of DNNs in crack recognition scenario, and discover one unique characteristic of this scenario, i.e., the predictions made by DNNs are usually both over-confident and under-confident.
- Investigate the relationship between the adversarial attack and the correctness of predictions: the features of inputs vulnerable to attacks are likely to be correctly classified. Then, we propose the attack-inspired correctness estimation method (ACE) that aims to estimate the correctness degree of current predictions.
- Propose attack-inspired calibration loss (AICL) to calibrate crack recognition models via our ***Correctness-aware Distribution Guidance***. It allows us to provide guidance at a distribution level, contributing more flexibility for the correctness estimation from ACE.
- Propose a labeled bridge dataset, namely, BRIDGE2024, which contains accurate annotations, to facilitate the research about trustworthy crack recognition for the real-world applications.

Related Work

Crack Recognition

According to previous research (Nguyen et al. 2022), image-based crack recognition via classification approaches can be clustered into two groups: traditional approaches and deep learning approaches. In the traditional approaches, there are two stages of the method. In the first stage, hand-craft features are extracted by different types of descriptors, e.g., HOG (Kapela et al. 2015) and LBP (Varadharajan et al. 2014). And then, a pre-trained classifier is applied to extract potential crack patches (Fang et al. 2020). With the development of deep learning (DL) technology, many existing works integrate deep learning techniques for crack recognition tasks (Fang et al. 2020). The typical DL methods first train the Convolution Neural Network (CNN) (Gopalakrishnan et al. 2017; Zhang et al. 2016) on sub-images, then use the trained model to scan the high-resolution image with a sliding window (Cha, Choi, and Büyüköztürk 2017) to coarsely locate the crack by classifying each sub-image. These works focus on how to design and train a powerful CNN model for identifying crack regions on sub-images of 99×99 pixels (Eisenbach et al. 2017; Zhang et al. 2016), 120×120 pixels (Chen and Jahanshahi 2016), 200×200 pixels (Zhang, Cheng, and Zhang 2018), 224×224 pixels (Chen et al. 2022), and 256×256 pixels (Cha, Choi, and Büyüköztürk 2017; Dorafshan, Thomas, and Maguire 2018).

In this paper, we reveal that the existing DNNs are far distant from being well-calibrated in crack recognition scenario. To address the limitations of existing research, we focus on studying the trustworthy problem of DNNs in crack recognition, an area that has been overlooked thus far.

Adversarial Attack

Essentially, adversarial attacks can be divided into two types: (i) black-box setting and (2) white-box setting. Generally, the gradient-based attack methods under the white-box setting are more practical and powerful on real tasks (Zhou et al. 2020). Typically, the fast gradient sign method (FGSM) changes the clean seed image by taking the gradient sign of the model loss function in one step (Goodfellow, Shlens, and Szegedy 2014). Then, κ-step projected gradient descent method (Madry et al. 2018) (PGD-κ) works iteratively and can be viewed as an iterative version of FGSM, which produces adversarial examples to fool the model by directly increasing the loss of the model.

The most recent work that is close to ours, (Qin et al. 2021) leverages adversarial robustness as an indicator to smooth the training labels, and then improve model calibration adaptively. Our work, instead, explores the relationship between the adversarial attack and the correctness degree of predictions, to calibrate crack recognition.

Confidence Calibration

The goal of model confidence calibration is to well align the model's predictive confidence with the actual likelihood of its correctness (Munir et al. 2024). To achieve this, the current model calibration methods can be grouped into two categories: post-hoc and train-time methods. The difference between the two categories lies in that the former require hold-out validation set and involve a few parameters, whereas the latter do not require validation data and involve all model parameters. The first category methods rely on an ideal hold-out validation data, which may be hard to satisfy in many real-world applications. In this regard, we mainly focus on the second category. Train-time calibration methods aim to design the auxiliary loss functions. Then, the DNN-based model can be jointly supervised by the above auxiliary loss function and the task-specific loss function, thereby achieving model calibration (Hebbalaguppe et al. 2022). For example, DCA (Liang et al. 2020) is proposed for the medical field in which neural network miscalibration has the potential to lead to significant treatment errors. Motivated by the Hilbert space (Gretton 2013), (Kumar, Sarawagi, and Jain 2018) design an auxiliary loss via a customized reproducing kernel. Furthermore, AVUC (Krishnan and Tickoo 2020) is proposed to allow a model to learn to provide well-calibrated uncertainties while improving accuracy. By considering the multi-class difference in the confidence and accuracy, (Hebbalaguppe et al. 2022) propose a novel loss function that aims to calibrate the predicted confidence of all classes. To achieve implicit model calibration, (Liu et al. 2022) propose a margin constraint logit distances based on the label smoothing techniques (Müller, Kornblith, and Hinton 2019). Notably, (Bohdal, Yang, and Hospedales 2023) introduce a novel differentiable surrogate for expected calibration error and show its effectiveness in supervised classification tasks. Moreover, AR-AdaLS (Qin et al. 2021) improves label smoothing by explicitly teaching the model to differentiate the training data according to their adversarial robustness and then adaptively smooth their labels. ACE (Chen et al. 2024b) is proposed to maximize the predicted information entropy by leveraging adversarial examples.

However, even if some works have explored the adversarial attack for confidence calibration, they generally degrade the classification accuracy of the original classifier, while good accuracy is still a basic requirement for real-world crack recognition applications. Notably, DNNs exhibit both under-confidence and over-confidence in crack recognition tasks, distinct from general image classification tasks. For this reason, DNNs are still far from being well-calibrated in crack recognition tasks with the above calibration methods.

Method

In this section, we introduce the proposed AICL. We start with the motivation of the AICL, and provide the descriptions of the attack-inspired correctness estimation method (ACE) and the ***Correctness-aware Distribution Guidance*** of AICL.

Motivation of AICL

Generally speaking, DNNs are more robust when they require more adversarial attack rounds to be successfully attacked, thereby enjoying stronger generalization ability (Pedraza, Deniz, and Bueno 2021). For this cause, we have the motivation that the sample is more likely to be successfully

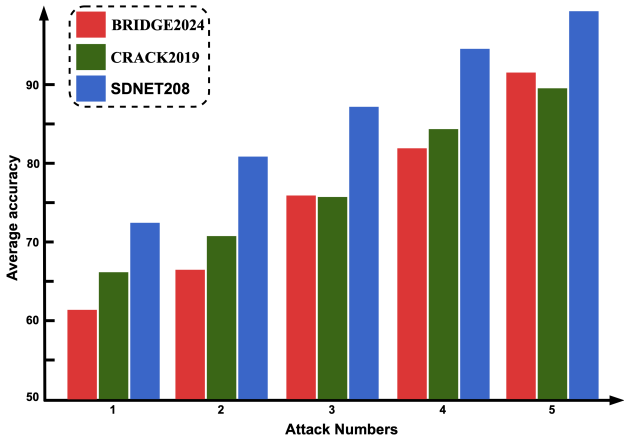

Figure 3: The relationship between the least number of iterations required for samples to be successfully attacked and their prediction correctness. All experiments on three datasets verify that attack numbers are associated with the correctness of their corresponding prediction.

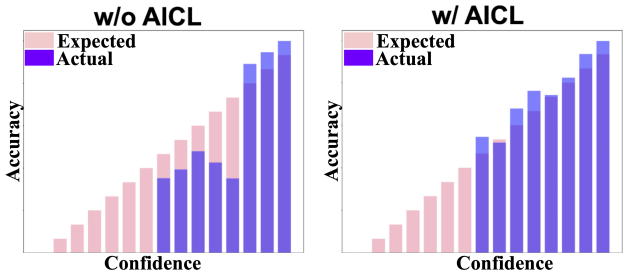

Figure 4: Reliability diagrams: crack recognition with (w/) and without (w/o) AICL on SDNET2018 dataset.

classified when it needs more adversarial attack rounds. Accordingly, our AICL is based on an intuitive idea: ***The least attack numbers required to change the initial predicted pseudo label are closely related to the correctness degree of initial prediction.***

To verify this intuition, we provide a series of experiments on CRACK2019, SDNET2018, and our Bridge2024 datasets. More specifically, we first train the commonly used ResNet-50 (He et al. 2016) with the supervision of cross-entropy loss on the above dataset, respectively. Then, we extract the features of the samples for each dataset and perform PGD attacks on these features to get the least number of iterations that are required to change the initial pseudo label. After that, we calculate the mean accuracy of the samples that need different attack numbers. Figure 3 shows that the features involving small/large κ are less/more likely to be correctly classified. Thus, the correctness degree of current predictions can be reflected by the least number of attacks. Moreover, we also train the ResNet50 with and without the supervision of AICL, and show the calibration performance in Figure 4. We can observe that the model's predictive confidence and correctness are well aligned when AICL is adopted for supervision. This further verifies that the attack number can reflect the correctness degree of current predictions, and then we are allowed to calibrate crack recognition models by regularizing class probabilities regarding the ordinal ranking of the estimated correctness degree.

Realization of ACE

Based on the previous discussions, the correctness degree of initial predictions can be reflected by the least number of iterations required to generate its adversarial variant that can change the initial pseudo label. Herein, we propose an efficient attack-inspired correctness estimation method (ACE), that aims to estimate the correctness degree of the current prediction. Inspired by the existing method (Zhang et al. 2020), our method is distinct in that (1) we focus on exploring the relationship between the least attack numbers (required to change the initial pseudo label) and the correctness degree of initial predictions, and (2) we enjoy the efficiency as it directly attacks the features, and does not require back-propagation to the backbone during the attack process. The corresponding pseudocode is presented in Algorithm 1.

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be a dataset consisting of n labeled samples. x_i denotes the input sample and $y_i \in \{1, 2, 3, .., C\}$ indicates the ground-truth label of x_i, where C denotes the class num. Given the feature extractor $\mathbb{F}$ and the linear classifier $\mathbb{C}$, the adversarial attacks in ACE can be interpreted as a multi-step scheme for maximizing the cross-entropy (CE) loss function $\mathcal{L}_{\text{CE}}$ with respect to the feature $\mathcal{F}_{x_i}$ and the initial pseudo label $\hat{y}^0$ of the sample x_i, where $\mathcal{F}_{x_i}$ can be obtained by the feature extractor: $\mathcal{F}_{x_i} = \mathbb{F}(x_i)$. Then, $\hat{y}^0$ can be obtained as follows:

$$\hat{y}^0 = \underset{k \in \mathcal{Y}}{argmax}\, \mathbb{C}(\mathcal{F}_{x_i}) \tag{1}$$

Then, the $(t)^{th}$ adversarial variant $\mathcal{F}_{x_i}^{(t)}$ can be updated along the gradient $\nabla_{\mathcal{F}_{x_i}^{(t)}} \mathcal{L}(\cdot)$ of $\mathcal{L}_{\text{CE}}$ with respect to $\mathcal{F}_{x_i}^{(t)}$. As a consequence, $\mathcal{F}_{x_i}^{(t+1)}$ can be obtained by the follows:

$$\begin{aligned} &\mathcal{F}_{x_i}^{(t+1)} = \mathcal{F}_{x_i}^{(t)} + \\ &\Pi_{\mathcal{B}[\mathcal{F}_{x_i}^{(t)}, \epsilon]} \left(\gamma \operatorname{sign} \left(\nabla_{\mathcal{F}_{x_i}^{(t)}} \mathcal{L}_{\text{CE}} \left(\mathbb{C} \left(\mathcal{F}_{x_i}^{(t)} \right), \hat{y}^0 \right) \right) \right), \end{aligned} \tag{2}$$

where 0^{th} adversarial variant $\mathcal{F}_{x_i}^{(0)}$ is initialized by $\mathcal{F}_{x_i}$. The step size γ is used to control the magnitude of adversarial variant features change, and the project function $\Pi_{\mathcal{B}[\mathcal{F}_{x_i}^{(t)}, \epsilon]}$ is used to project $\mathcal{F}_{x_i}^{(t)}$ back into the center of $\mathcal{F}_{x_i}^{(t)}$, where the metric ϵ-ball controls its perturbation bound.

After that, the least number of adversarial attacks, i.e., required to change the initial pseudo label, can be obtained by starting from $\mathcal{F}_{x_i}^{(0)}$ until t_{th} adversarial variants that can fool the current network to have different predicted label with $\hat{y}^0$. Thus, in Algorithm 1, the certain stopping criterion is that the predicted label of the $\mathcal{F}_{x_i}^{(t)}$ is unequal to the initial pseudo label $\hat{y}_0$, or reaches the maximum attack number K. Finally, the correctness degree $\kappa(x_i)$ of the sample x_i can be approximated by the t.

Algorithm 1: Attack-inspired correctness estimation method (ACE)

Input: sample x_i, feature extractor $\mathbb{F}$, linear classifier $\mathbb{C}$, cross-entropy (CE) loss $\mathcal{L}_{\text{CE}}$, class num C, label set $\mathcal{Y} = \{1, 2, 3, .., C\}$, maximum attack number K, step size γ, perturbation bound ϵ

Feature extraction: $\mathcal{F}_{x_i} = \mathbb{F}(x_i)$
Initial pseudo label: $\hat{y}^0 = argmax_{k \in \mathcal{Y}} \mathbb{C}(\mathcal{F}_{x_i})$
Adversarial feature initialize: $\mathcal{F}_{x_i}^{(0)} \leftarrow \mathcal{F}_{x_i}$
Attack number initialization : $t \leftarrow 0$
while $K > 0$ **do**
  $\nabla = \Pi_{\mathcal{B}[\mathcal{F}_{x_i}^{(t)}, \epsilon]}(\gamma \operatorname{sign}(\nabla_{\mathcal{F}_{x_i}^{(t)}} \mathcal{L}_{\text{CE}}(\mathbb{C}(\mathcal{F}_{x_i}^{(t)}), \hat{y}^0)))$
  $\mathcal{F}_{x_i}^{(t+1)} = \mathcal{F}_{x_i}^{(t)} + \nabla$
  if $argmax_{k \in \mathcal{Y}} \mathbb{C}(\mathcal{F}_{x_i}^{(t+1)}) == \hat{y}^0$ **then**
    $t \leftarrow t + 1$
  else
    $\kappa(x_i) \leftarrow t + 1, K = 0$
  end if
  $K \leftarrow K - 1$
end while
Output: Correctness degree $\kappa(x_i)$

Learning Objective of AICL

Now, given two samples x_i and x_j, we are allowed to obtain the ordinal ranking of their correctness degree with the help of ACE. Then, as mentioned before, the goal of confidence calibration is to align the model's confidence with the correctness of current predictions. To achieve this, we are encouraged to enforce the estimated confidence of x_i and x_j to have a consistent ordinal ranking with their estimated correctness degree by ACE:

$$\begin{aligned} &\kappa(x_i) \leq \kappa(x_j) \\ &\Downarrow \\ &conf(\mathbf{p}_i|x_i, \theta_{\mathbb{F}}, \theta_{\mathbb{C}}) \leq conf(\mathbf{p}_j|x_j, \theta_{\mathbb{F}}, \theta_{\mathbb{C}}) \end{aligned} \tag{3}$$

where $conf(\cdot)$ denotes a confidence function (e.g., the maximum class probability and margin) and $\kappa(\cdot)$ denotes the correctness function of ACE. Herein, we adopt maximum class probability as the confidence function. Note that $\mathbf{p}_i$ and $\mathbf{p}_j$ represents the predicted class probabilities of sample x_i and x_j by feature extractor $\mathbb{F}$ and classifier $\mathbb{C}$. Meanwhile, $\theta_{\mathbb{F}}$ and $\theta_{\mathbb{C}}$ denotes the parameters of $\mathbb{F}$ and $\mathbb{C}$.

Ideally, a promising approach is to learn the relationship in Eq. 3 directly during training. Thus, it is intuitive to design a loss function to reflect the desirable ordinal ranking of confidence estimates in Eq. 3. This loss function should be affected by whether the ranking of two samples is right or not, and the loss will be incurred when the relationship in Eq. 3 is violated. For this purpose, a ***Sample-level Guidance*** can be defined as:

$$\begin{aligned} \mathcal{L}_s = &\max\{0, -g(\kappa(x_i), \kappa(x_j)) \cdot \\ &(conf(\mathbf{p}_i|x_i, \theta_{\mathbb{F}}, \theta_{\mathbb{C}}) - conf(\mathbf{p}_j|x_j, \theta_{\mathbb{F}}, \theta_{\mathbb{C}}))\} \end{aligned} \tag{4}$$

where

$$g\left(\kappa(x_i), \kappa(x_j)\right) = \begin{cases} 1, & \text{if } \kappa(x_i) > \kappa(x_j) \\ 0, & \text{if } \kappa(x_i) = \kappa(x_j) \\ -1, & \text{otherwise} \end{cases} \tag{5}$$

In practice, such a sample-level guidance follow the relationship in Eq. 3, but results in inferior performance. The reason is that it requires a well-estimated likelihood of each sample's correctness, which is too overly restrictive for ACE.

Based on the above discussion, our AICL proposes an alternate guidance strategy termed, ***Correctness-aware Distribution Guidance*** for calibrating crack recognition, which loosens the reliance on an ideal ACE. The key idea is to consider the relationship in Eq. 3 from a distribution perspective. To achieve this, we first define the input data x such that $x \sim \mathcal{P}(x)$, where $\mathcal{P}(x)$ is the true data distribution. Then, we define correctness-conditional distribution $\mathcal{P}(x|\kappa(x) = t)$ where $t \in \{1, \cdots, K\}$ denotes the estimated correctness degree by ACE. Next, suppose that $1 \leq t_m \leq t_n \leq K$, Eq. 3 can be reformulated as:

$$\begin{aligned} &t_m \leq t_n \\ &\Downarrow \\ &\mathbb{E}_{x \sim \mathcal{P}(x|\kappa(x)=t_m)} \left(conf(\mathbf{p}|x, \theta_{\mathbb{F}}, \theta_{\mathbb{C}})\right) \leq \\ &\qquad \mathbb{E}_{x \sim \mathcal{P}(x|\kappa(x)=t_n)} \left(conf(\mathbf{p}|x, \theta_{\mathbb{F}}, \theta_{\mathbb{C}})\right) \end{aligned} \tag{6}$$

where $\mathbf{p}$ represents the predicted class probabilities of sample x by feature extractor $\mathbb{F}$ and classifier $\mathbb{C}$. Now, we are able to design a loss function to reflect the desirable ordinal ranking of confidence estimation at the distribution level. Such a loss function should be affected by whether the ranking of the expectation of two distributions is right or not, and the loss will be incurred when the relationship in Eq. 6 is violated. Then, our ***Correctness-aware Distribution Guidance*** can be derived as:

$$\begin{aligned} \mathcal{L}_d = \sum_{1 \leq t_m \leq t_n \leq K} &[\mathbb{E}_{x \sim \mathcal{P}(x|\kappa(x)=t_m)} \left(conf(\mathbf{p}|x, \theta_{\mathbb{F}}, \theta_{\mathbb{C}})\right) \\ &- \mathbb{E}_{x \sim \mathcal{P}(x|\kappa(x)=t_n)} \left(conf(\mathbf{p}|x, \theta_{\mathbb{F}}, \theta_{\mathbb{C}})\right)] \end{aligned} \tag{7}$$

In summary, the learning objective of AICL can be derived as $\mathcal{L}_d$ and is easily plugged into deep models.

Overall loss function. Herein, we adopt the CE loss $\mathcal{L}_{CE}$ and $\mathcal{L}_d$ as our final loss function $\mathcal{L}$ to calibrate deep models in crack recognition tasks.

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_d, \tag{8}$$

where λ is used to balance the relative contributions of these two loss terms. The sensitivity study of λ is provided in the experiment section.

Experiments

In this section, we present the essential experimental setup. Then, to find the most suitable parameters K and λ in AICL, we carry out some experiments on different parameters. Furthermore, to verify the effectiveness of our proposed method, we not only compared it with other crack classification approaches, but also compared it with other state of the existing calibration methods. At last, we present ablation study and visualization on the proposed method.

Method	SDNET2018						BRIDGE2024					
	AUC ↑	**ACC ↑**	**SEN ↑**	**SPEC ↑**	**F1 ↑**	**ECE ↓**	**AUC ↑**	**ACC ↑**	**SEN ↑**	**SPEC ↑**	**F1 ↑**	**ECE ↓**
SDNET	86.32	87.19	87.20	72.95	82.13	10.4	96.04	95.75	96.32	88,97	92.18	6.87
SilvaNet	89.17	90.43	88.19	77.32	83.32	8.78	99.01	96.73	95.18	89.43	93.07	7.05
AliNet	87.43	89.32	86.71	74.68	82.96	9.88	98.73	96.90	96.03	91.18	93.94	7.21
DCrack	88.92	90.14	85.49	73.94	83.35	9.04	97.42	**97.93**	97.12	90.97	94.83	5.98
IVGG	91.45	92.02	**87.82**	75.64	83.17	9.56	98.90	98.14	97.58	91.61	95.03	6.78
ResNet50 + AICL	90.18	91.63	82.67	73.31	82.81	7.80	99.25	97.77	97.36	**92.07**	**95.88**	**2.10**
VGG16 + AICL	**92.96**	**92.40**	87.70	**77.23**	**84.61**	**6.79**	**99.65**	97.49	**98.16**	91.57	95.24	2.19

Table 1: Comparisons with the state-of-the-art crack recognition methods on SDNET2018 and BRIDGE2024 datasets. "AUC", "ACC", "SEN" and "SPEC" are ROC curve, Accuracy, Sensitivity, and Specificity, respectively. Expected calibration error (ECE) is adopted to evaluate calibration performance.

Experimental Setup

Here, we present detailed information regarding the crack classification datasets and implementation details.

BRIDGE2024. We collect this dataset from bridges via unmanned aerial vehicles equipped with high-resolution cameras, comprising 1000 images with the size of 4096×3072. Due to the diversity of cracks in bridges, the captured images of our dataset include cracks with different widths. Also, the collected images contain noises, such as uneven illumination and blurred backgrounds caused by the unstable flight stability. To augment the dataset without compromising the resolution, we slice the captured images into image patches of 400×400 pixels, composing a final dataset, which is carefully manually labeled in two classes: non-crack and crack. The numbers of image patches in each category are 10965 and 2407, respectively. This dataset is divided into training set, validation set, and test set at the ratio of $3 : 1 : 1$.

CRACK2019 (Zhang et al. 2016; Özgenel and Sorguç 2018)**.** This dataset is partially from crack500 (Zhang et al. 2016), which is collected from Temple University and various METU Campus Buildings. This dataset has 40000 images with the size of 227×227 pixels and involves two classes: non-crack and crack, which each class having 20000 images. It is worth noting that this dataset is generated from 458 high-resolution images (4032×3024 pixels) with high variance in surface finish and illumination condition. Thus, it is challenging to classify these images with well-estimated confidence. In our experiments, we divide it into training set, validation set, and test set at the ratio of $3 : 1 : 1$. This dataset is available at [1].

SDNET2018 (Dorafshan, Thomas, and Maguire 2018)**.** The total size of this dataset is 56000 with crack and non-crack samples collected from various scenarios including bridge decks, concrete walls, and pavements. More specifically, the SDNET2018 dataset contains cracks as narrow as 0.06 mm and as wide as 25 mm with various types of noise, such as shadow, scaling, edge, and hole (Dorafshan, Thomas, and Maguire 2018). Herein, we follow the same settings as the previous dataset that divide this dataset into the training set, validation set, and test set at the ratio of $3 : 1 : 1$.

Implementation details. For fairness, we select the commonly used ResNet50 (He et al. 2016) and VGG16 (Simonyan and Zisserman 2014) as the feature extractor. Herein, we resize all the images into the size of 224×224 and perform data augmentation via random horizontal flip. For all of these datasets, the total training epochs is set as 40 with an initial learning rate 0.01, and reduced by a factor of 10 after 15, 25 epochs. Meanwhile, we adopt mini-batch Adam as the optimizer with the mini-batch size 64. According to the sensitivity study, the maximum attack step K is set to 5. During the initial period of the training epochs, the attack numbers among different samples are less informative when the classifier is not properly learned. For this reason, the initial 10 epochs are burn-in period, in which we only adopt CE loss for the supervision. For a fair comparison, all methods are implemented with the same training configuration when it is possible.

Evaluation metrics. Since we aim to verify the proposed method can achieve good calibration performance while maintaining classification performance, we report a series of metrics that are concerned with real-world crack recognition applications, including area under the ROC curve (AUC), Sensitivity (SEN), Accuracy (ACC), Specificity (SPEC) and F1. Meanwhile, we adopt expected calibration error (ECE) with the number of bins set to 15 to evaluate model calibration performance (Naeini, Cooper, and Hauskrecht 2015).

Sensitivity Study on Hyper-parameters

There are two parameters that have been introduced in this paper. The parameter K is introduced to control the maximum attack number. λ is utilized to balance the proposed loss and CE loss. The sensitivity study is carried out on the BRIDGE2024 dataset with ResNet-50 as the backbone. Based on these observations from Figure 5, we set $K = 5$ and $\lambda = 2.0$ in our next experiments.

Comparison with Crack Classification Approaches

In this section, our method is compared to several baselines including state-of-the-art robust crack classification methods: SDNET (Dorafshan, Thomas, and Maguire 2018) uses the AlexNet architecture to detect the crack, SilvaNet (Silva and Lucena 2018) uses the VGG16 architecture, AliNet (Ali et al. 2021) proposes a customized ResNet50 architecture, DCrack (Zhou and Song 2021) proposes novel deep convolutional neural network–based crack classification model, IVGG (Que et al. 2023) proposes an improved VGG model

[1] https://data.mendeley.com/datasets/5y9wdsg2zt/2

Method	SDNET2018						BRIDGE2024					
	AUC ↑	**ACC ↑**	**SEN ↑**	**SPEC ↑**	**F1 ↑**	**ECE ↓**	**AUC ↑**	**ACC ↑**	**SEN ↑**	**SPEC ↑**	**F1 ↑**	**ECE ↓**
FLSD	86.67	88.46	84.18	68.90	77.09	11.7	97.45	96.89	97.01	88.17	93.16	6.78
AR-AdaLS	87.54	89.10	85.28	69.19	78.00	9.76	97.16	95.64	96.50	89.30	92.98	6.04
MbLS	88.93	89.79	**85.61**	67.13	80.38	10.9	98.88	97.06	97.22	87.41	95.29	5.89
DualFocalLoss	88.59	90.57	82.20	72.74	80.52	8.60	99.01	97.73	95.69	**93.61**	95.47	2.44
ACEnt	89.05	91.04	83.42	71.59	81.82	8.30	99.10	97.03	96.18	91.43	94.89	3.56
DECE	87.19	89.04	84.27	69.18	78.90	8.02	98.83	96.24	95.94	86.70	93.05	3.21
AICL	**90.18**	**91.63**	82.67	**73.31**	**82.81**	**7.80**	**99.25**	**97.77**	**97.36**	92.07	**95.88**	**2.10**

Table 2: Comparisons with the state-of-the-art calibration methods on SDNET2018 and BRIDGE2024 datasets. "AUC", "ACC", "SEN" and "SPEC" are ROC curve, Accuracy, Sensitivity, and Specificity, respectively. Again, expected calibration error (ECE) is adopted to evaluate calibration performance.

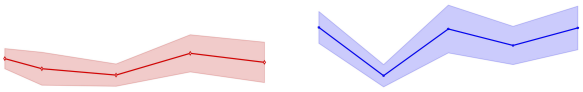

Figure 5: Hyper-parameter sensitivity study of K and λ with ECE as the metric on BRIDGE2024 dataset.

for pavement crack classification. Given that previous works used ResNet-50 and VGG16 as backbones, we also use these models as our backbones to ensure fairness.

Table 2 shows the comparisons with the existing crack classification approaches evaluated on the test sets. In general, the proposed method with ResNet-50 and VGG16 as the backbone demonstrates its superior performance in both classification and calibration performance against the baselines with large gaps. Especially, compared to the baselines which also adopt VGG16 as the backbone, our approach achieves state-of-the-art calibration performance on both SDNET2018 and BRIDGE2024 datasets. This indicates that utilizing ***Distribution-level Guidance*** plays a pivotal role in calibrating crack recognition.

Comparison with Existing Calibration Methods

In this section, to further validate the superiority of the proposed method, we conduct a series of experiments that compare with the existing calibration methods on SDNET2018 and BRIDGE2024 dataset including: FLSD (Mukhoti et al. 2020), AR-AdaLS (Qin et al. 2021), MbLS (Liu et al. 2022), DualFocalLoss (Tao, Dong, and Xu 2023), ACEnt (Chen et al. 2024b), DECE (Bohdal, Yang, and Hospedales 2023). Regarding the existing calibration methods, our method also achieves superior performance. Compared with ACEnt which considers adversarial calibration entropy, our method achieves better performance by leveraging the relationship between the adversarial attack and the correctness degree of predictions. Meanwhile, the below table shows that the combination of our method and the existing post-hoc method, i.e., Temperature scaling (Guo et al. 2017), on BRIDGE2024 achieves the best competitive performance, as evidenced by reported results.

Methods	Our	Our+Temperature scaling
ECE ↓	2.10	**1.90**

Table 3: Performance with Temperature scaling.

Methods	Crack2019	
	F1 ↑	**ECE ↓**
w/ ***Sample-level Guidance***	96.54	4.19
w/ ***Distribution-level Guidance***	**99.84**	**0.20**

Table 4: Performance comparisons of with (w/) different level guidance on Crack2019 dataset.

Ablation Study

Evaluation on the *Correctness-aware Distribution Guidance*: As expected, AICL achieves better results by using ***Distribution-level Guidance***. Table 4 shows that it improves the F1 and ECE by 3.3% and 3.99%. The reason behind this effect is that ***Sample-level Guidance*** requires a well-estimated likelihood of each sample's correctness degree, which is too overly restrictive for ACE.

Conclusion

In this work, we are the first ones to reveal that the predictions made by DNNs are usually both over-confident and under-confident on crack recognition tasks. To address this problem, we first investigate the relationship between adversarial attack numbers and the correctness degree of predictions, and propose the attack-inspired correctness estimation method (ACE) that aims to estimate the correctness degree of current predictions. Then, we propose attack-inspired calibration loss (AICL) to calibrate crack recognition models via ***Correctness-aware Distribution Guidance*** from the distribution perspective. The experiments on both crack and general classification tasks demonstrate the superiority of our method over the existing calibration methods. In further work, we would like to consider leveraging detected information to help real-world robotic planning (Li et al. 2020).

Acknowledgements

This work is supported by the National Natural Science Funds for Distinguished Young Scholar under Grant 62325307, the National Natural Science Foundation of China under Grants 62473264, 62073225, 62203134, 62072315, the National Key R&D Program of China under Grants 2020YFA0908700, the Natural Science Foundation of Guangdong Province under Grants 2023B1515120038, Shenzhen Science and Technology Innovation Commission (20220809141216003, JCYJ20210324093808021, JCYJ20220531102817040, KJZD20230923113801004), the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603, the Guangdong "Pearl River Talent Plan" under Grant 2019JC01X235, the Scientific Instrument Developing Project of Shenzhen University under Grant 2023YQ019.

References

Ali, L.; Alnajjar, F.; Jassmi, H. A.; Gochoo, M.; Khan, W.; and Serhani, M. A. 2021. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*, 21(5): 1688.

Bohdal, O.; Yang, Y.; and Hospedales, T. 2023. Meta-Calibration: Learning of Model Calibration Using Differentiable Expected Calibration Error. *Transactions on Machine Learning Research*.

Cha, Y.-J.; Choi, W.; and Büyüköztürk, O. 2017. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5): 361–378.

Chen, F.-C.; and Jahanshahi, M. R. 2016. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion. *IEEE Transactions on Industrial Electronics*, 65(5): 4392–4400.

Chen, J.; Zhu, G.; Zhang, Y.; Chen, Z.; Huang, Q.; and Li, J. 2024a. UCAN: U-shaped context aggregation network for thin crack segmentation under topological constraints. *Robotic Intelligence and Automation*, 44(5): 637–647.

Chen, Y.; Hu, P.; Yuan, Z.; Peng, D.; and Wang, X. 2024b. Integrating confidence calibration and adversarial robustness via adversarial calibration entropy. *Information Sciences*, 668: 120532.

Chen, Z.; Lai, Z.; Chen, J.; and Li, J. 2024c. Mind Marginal Non-Crack Regions: Clustering-Inspired Representation Learning for Crack Segmentation. In *CVPR*, 12698–12708.

Chen, Z.; Lu, R.; Chen, J.; Song, H. H.; and Li, J. 2024d. Implicit Gradient-Modulated Semantic Data Augmentation for Deep Crack Recognition. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, Z.; Zhang, J.; Lai, Z.; Chen, J.; Liu, Z.; and Li, J. 2022. Geometry-Aware Guided Loss for Deep Crack Recognition. In *Proc. CVPR*, 4703–4712.

Chen, Z.; Zhang, J.; Lai, Z.; Zhu, G.; Liu, Z.; Chen, J.; and Li, J. 2023. The devil is in the crack orientation: A new perspective for crack detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6653–6663.

Crawford, K. C. 2023. Perspective Chapter: Bridge Deterioration and Failures. In *Failure Analysis-Structural Health Monitoring of Structure and Infrastructure Components*. IntechOpen.

Dorafshan, S.; Thomas, R. J.; and Maguire, M. 2018. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in brief*, 21: 1664–1668.

Eisenbach, M.; Stricker, R.; Seichter, D.; Amende, K.; Debes, K.; Sesselmann, M.; Ebersbach, D.; Stoeckert, U.; and Gross, H.-M. 2017. How to get pavement distress detection ready for deep learning? A systematic approach. In *Proc. IJCNN*, 2039–2047.

Fang, F.; Li, L.; Gu, Y.; Zhu, H.; and Lim, J.-H. 2020. A novel hybrid approach for crack detection. *Pattern Recognition*, 107: 107474.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gopalakrishnan, K.; Khaitan, S. K.; Choudhary, A.; and Agrawal, A. 2017. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157: 322–330.

Gretton, A. 2013. Introduction to rkhs, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 16(5-3): 2.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hebbalaguppe, R.; Prakash, J.; Madan, N.; and Arora, C. 2022. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *CVPR*, 16081–16090.

Kapela, R.; Śniatała, P.; Turkot, A.; Rybarczyk, A.; Pożarycki, A.; Rydzewski, P.; Wyczałek, M.; and Błoch, A. 2015. Asphalt surfaced pavement cracks detection based on histograms of oriented gradients. In *Proc. MIXDES*, 579–584.

Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; and Fieguth, P. 2015. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced engineering informatics*, 29(2): 196–210.

Krishnan, R.; and Tickoo, O. 2020. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33: 18237–18248.

Kumar, A.; Sarawagi, S.; and Jain, U. 2018. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, 2805–2814.

Li, D.; Duan, Z.; Hu, X.; Zhang, D.; and Zhang, Y. 2023. Automated classification and detection of multiple pavement distress images based on deep learning. *Journal of Traffic and Transportation Engineering (English Edition)*, 10(2): 276–290.

Li, D.; Li, L.; Chen, Z.; and Li, J. 2024a. Shift-ConvNets: Small convolutional kernel with large kernel effects. *arXiv preprint arXiv:2401.12736*.

Li, J.-q.; Zhang, Y.-f.; Chen, Z.-z.; Wang, J.; Fang, M.; Luo, C.-w.; and Wang, H. 2020. A novel edge-enabled SLAM solution using projected depth image information. *Neural Computing and Applications*, 32: 15369–15381.

Li, W.; Ma, Z.; Deng, L.-J.; Fan, X.; and Tian, Y. 2022. Neuron-based spiking transmission and reasoning network for robust image-text retrieval. *IEEE TCSVT*, 33(7): 3516–3528.

Li, W.; Wang, P.; Xiong, R.; and Fan, X. 2024b. Spiking tucker fusion transformer for audio-visual zero-shot learning. *IEEE TIP*.

Liang, G.; Zhang, Y.; Wang, X.; and Jacobs, N. 2020. Improved trainable calibration method for neural networks on medical imaging classification. In *British Machine Vision Conference*.

Liu, B.; Ben Ayed, I.; Galdran, A.; and Dolz, J. 2022. The devil is in the margin: Margin-based label smoothing for network calibration. In *CVPR*, 80–88.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Mi, Y.; Huang, Y.; Ji, J.; Zhao, M.; Wu, J.; Xu, X.; Ding, S.; and Zhou, S. 2023. Privacy-preserving face recognition using random frequency components. In *ICCV*, 19673–19684.

Mi, Y.; Zhong, Z.; Huang, Y.; Ji, J.; Xu, J.; Wang, J.; Wang, S.; Ding, S.; and Zhou, S. 2024. Privacy-preserving face recognition using trainable feature subtraction. In *CVPR*, 297–307.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P.; and Dokania, P. 2020. Calibrating deep neural networks using focal loss. *NeurIPS*, 33: 15288–15299.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In *Advances in neural information processing systems*.

Munir, M. A.; Khan, S. H.; Khan, M. H.; Ali, M.; and Shahbaz Khan, F. 2024. Cal-DETR: Calibrated Detection Transformer. In *Advances in Neural Information Processing Systems*, 1–12.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*.

Nguyen, S. D.; Tran, T. S.; Tran, V. P.; Lee, H. J.; Piran, M. J.; and Le, V. P. 2022. Deep Learning-Based Crack Detection: A Survey. *International Journal of Pavement Research and Technology*, 1–25.

Nguyen, S. D.; Tran, T. S.; Tran, V. P.; Lee, H. J.; Piran, M. J.; and Le, V. P. 2023. Deep learning-based crack detection: A survey. *International Journal of Pavement Research and Technology*, 16(4): 943–967.

Özgenel, Ç. F.; and Sorguç, A. G. 2018. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *ISARC*, volume 35, 1–8.

Pedraza, A.; Deniz, O.; and Bueno, G. 2021. On the relationship between generalization and robustness to adversarial examples. *Symmetry*, 13(5): 817.

Qin, Y.; Wang, X.; Beutel, A.; and Chi, E. 2021. Improving calibration through the relationship with adversarial robustness. *Advances in Neural Information Processing Systems*, 34: 14358–14369.

Que, Y.; Dai, Y.; Ji, X.; Leung, A. K.; Chen, Z.; Jiang, Z.; and Tang, Y. 2023. Automatic classification of asphalt pavement cracks using a novel integrated generative adversarial networks and improved VGG model. *Engineering Structures*, 277: 115406.

Silva, W. R. L. d.; and Lucena, D. S. d. 2018. Concrete cracks detection based on deep learning image classification. In *Multidisciplinary digital publishing institute proceedings*, 489.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tao, L.; Dong, M.; and Xu, C. 2023. Dual focal loss for calibration. In *ICML*, 33833–33849.

Varadharajan, S.; Jose, S.; Sharma, K.; Wander, L.; and Mertz, C. 2014. Vision for road inspection. In *Proc. WACV*, 115–122.

Yuan, X.; Li, W.; Yin, X.; Chen, G.; Zhao, J.; Jiang, W.; and Ge, J. 2020. Identification of tiny surface cracks in a rugged weld by signal gradient algorithm using the ACFM technique. *Sensors*, 20(2): 380.

Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. 2020. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*.

Zhang, K.; Cheng, H.; and Zhang, B. 2018. Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning. *Journal of Computing in Civil Engineering*, 32(2): 04018001.

Zhang, L.; Yang, F.; Zhang, Y. D.; and Zhu, Y. J. 2016. Road crack detection using deep convolutional neural network. In *Proc. ICIP*, 3708–3712.

Zhou, M.; Wu, J.; Liu, Y.; Liu, S.; and Zhu, C. 2020. Dast: Data-free substitute training for adversarial attacks. In *CVPR*, 234–243.

Zhou, S.; and Song, W. 2021. Deep learning–based roadway crack classification with heterogeneous image data fusion. *Structural Health Monitoring*, 20(3): 1274–1293.