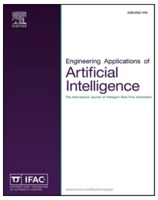
Engineering Applications of Artificial Intelligence 126 (2023) 107078

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Few-shot learning for image-based bridge damage detection

Yan Gao[a], Haijiang Li[a,*], Weiqi Fu[b]

[a] *BIM for Smart Engineering Centre, School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK*
[b] *Institute of Material Medica, Chinese Academy of Medical Sciences, Beijing, 10050, China*

ARTICLE INFO

Keywords:
Few-shot learning
Damage detection
Cross-domain transfer learning
Prototypical networks
Transduction

ABSTRACT

Autonomous bridge visual inspection is a real-world challenge due to various materials, surface coatings, and changing light and weather conditions. Traditional supervised learning relies on massive annotated data to establish a robust model, which requires a time-consuming data acquisition process. This work proposes a few-shot learning (FSL) approach based on improved ProtoNet for damage detection with just a few labeled examples. Feature embedding is achieved through cross-domain transfer learning from ImageNet instead of episodic training. The ProtoNet is improved with embedding normalization to enhance transduction performance based on Euclidean distance and a linear classifier for classification. The approach is explored on a public dataset through different ablation experiments and achieves over 94% mean accuracy for 2-way 5-shot classification via the pre-trained GoogleNet after fine-tuning. Moreover, the proposed fine-tuning methods based on a fully connected layer (FCN) and Hadamard product are demonstrated with better performance than the previous method. Finally, the approach is validated using real bridge inspection images, demonstrating its capability of fast implementation for practical damage inspection with weakly supervised information.

1. Introduction

Autonomous bridge visual inspection has become a real-world challenge due to various materials, surface coatings, changing light and weather conditions, and possible overlapping of different damages (Mundt et al., 2019a). Traditional supervised learning approaches for damage detection require a large number of labeled examples to establish a model, which results in a time-consuming and labor-intensive process for image acquisition (Nuthalapati and Tunga, 2021). It is also impractical to always collect sufficient defects from various damage scenarios. Furthermore, the supervised model can only identify specific defects and needs further training with new examples for novel classes. Transfer learning was expected to solve this issue, but conventional supervised transfer learning tends to be overfitting or challenging to converge with just a few annotated examples (Gidaris et al., 2018). However, humans can recognize novel classes with just a little supervised information, e.g., only one or a few examples, and generalize the knowledge to new images, which differs from inductive supervised learning, i.e., the capability of few-shot learning (FSL). Hence, many efforts have been made in this field currently. A typical FSL problem is few-shot classification, which aims to identify objects with very few examples (Lake et al., 2011), which can compensate for the deficiency of supervised learning in many fields. Therefore, developing an FSL approach for vision-based bridge damage detection with weakly supervised information, such as changing light, different materials, and novel defects, is significant. It should be available for fast implementation in real-world bridge inspection with drones or robots under complex circumstances.

This work proposes an approach based on improved ProtoNet (prototypical network) (Snell et al., 2017) for few-shot damage detection. Firstly, the inspection image is split into multiple patches. Feature embedding is achieved through cross-domain transfer learning from ImageNet. It enables the embedding function to be exempt from episodic training and become “training-free” (no need to be trained from scratch). Then, normalization is integrated after feature embedding to reduce domain variation and enhance the ProtoNet performance based on Euclidean distance by bridging the gap between Euclidean distance and cosine similarity as the metric for transduction inference. Secondly, the mean embedding vector is computed as the prototype for each class. Then, the transductive inference can be taken on each patch to show the initial performance by determining if the patch has the specific defect of the support set. The transduction alleviates the issue encountered by conventional transfer learning with only a few examples, such as overfitting or difficulty in convergence. Furthermore, a linear classifier

[*] Corresponding author.
E-mail address: lih@cardiff.ac.uk (H. Li).

https://doi.org/10.1016/j.engappai.2023.107078
Received 21 January 2023; Received in revised form 17 July 2023; Accepted 28 August 2023

$W^T x + b$ is added at the end of the ProtoNet for classification, and fine-tuning is taken based on the support set because the model aims to be trained before seeing query items in the practical inspection. Finally, the obtained prototypes and the fine-tuned classifier can be applied to a new inspection image, which is also split into multiple patches.

The proposed approach and architecture are explored in a public dataset for autonomous bridge crack detection (Dhillon et al., 2020). The dedicated CNN with the atrous spatial pyramid pooling (ASPP) module and depth-wise separable convolution for this dataset based on supervised learning can reach 96.37% accuracy in the test set. Extensive ablation studies are conducted to explore the approach performance, including hardcoded transformation, embedding normalization, various supervised or unsupervised DNN (deep neural network) backbones, and different fine-tuning methods. It achieves over 94% mean accuracy via GoogleNet after fine-tuning for 2-way 5-shot classification in the test set, which is already close to the performance of supervised learning (Xu et al., 2019a). Moreover, three different fine-tuning methods are compared in the experiment, including the transductive fine-tuning (i.e., Baseline) based on embedding vectors in the previous research (Dhillon et al., 2020), (Chen et al., 2019a), and the proposed methods based on Euclidean distance using a fully connected network (FCN) and the Hadamard product, respectively. It demonstrates that the proposed FCN-based and Hadamard-product fine-tuning methods can perform better than the previous method. Early stopping should be taken at the epoch number where the query accuracy reaches its peak and can be determined empirically for real damage detection. It also demonstrates that entropy regularization will slow down the fine-tuning. The entropy is calculated based on the support set because the model aims to be trained and fine-tuned before seeing query items. Hence, fine-tuning without entropy regularization is suggested for practical application.

The approach is also validated using real bridge inspection images, demonstrating its capability of fast implementation for practical damage detection without a time-consuming and labor-intensive process for data acquisition. The time cost of the approach for damage detection on each patch (84 × 84) can be 0.08s through the embedding functions of pre-trained VGG neural networks based on ImageNet, which demonstrates the approach's potential for damage detection in near real-time. Although the approach has the above advantages, it still has a few limitations, such as the robustness for noise (like stains and marks) and similar defects (but different kinds). Meanwhile, because different support sets will result in different performance in few-shot damage detection, how to determine support examples need further study.

The contribution of this work is four-fold.

1) This work proposes an approach for few-shot damage detection based on improved ProtoNet, wherein feature embedding is achieved by cross-domain transfer learning from ImageNet instead of episodic training.
2) The ProtoNet is improved with embedding normalization to reduce domain variation and enhance transduction performance based on Euclidean distance and a linear classifier for classification.
3) By comparison, the proposed classifier based on Euclidean distance and fine-tuning using FCN and the Hadamard product is recommended for practical application. The early-stopping time can be determined empirically in the experiment.
4) The approach is validated using real bridge inspection images, demonstrating its capability of fast implementation for damage detection with just a few annotated examples and its potential for practical inspection in near real-time.

The rest of this paper is organized as follows: Section 2 introduces the related work about damage detection and few-shot learning for images; Section 3 presents the proposed approach and architecture as well as the theoretical foundation; Section 4 conducts the ablation studies and validation for the approach; Section 5 concludes the work.

2. Related work

2.1. Damage classification and detection

For bridge visual inspection, a fundamental task is to determine if there are certain kinds of damage in an image, such as surface cracking, spalling, or rebar corrosion, i.e., damage classification (König et al., 2022). The task can be defined as the binary classification for each defect or a multi-defect classification. It can also be extended to determine whether damages exist and deduce the exact damage type, such as longitudinal crack, transverse crack, and alligator crack (König et al., 2022). Furthermore, damage detection aims to provide more information about the damage, such as location, area, skeleton, and direction, which is helpful because classification only indicates the existence of defects in an image but leaves the task of finding the actual defect to inspectors (König et al., 2022). A typical damage detection approach can be achieved by sliding the window or splitting the image into patches and then applying classification on each window or patch, followed by stitching them back, as shown in Fig. 1. Another type of damage detection utilizes bounding boxes to indicate defects, like object detection tasks in many competition datasets, such as COCO (Lin et al., 2014) and Pascal VOC (Everingham et al., 2010). However, this method is not always the best option to locate damage because defects have various shapes. The created bounding box can include many undefective sub-regions, e.g., an oblique crack is marked by a sizeable bounding box determined by its diagonal points.

The image-processing methods for damage detection underperform on practical inspection images due to the interference of surface textures, changing light, stains, etc. (Fu et al., 2021). Therefore, many data-driven approaches have been developed based on artificial intelligence (AI) for damage classification and detection to assist visual inspection. They can be categorized based on feature extraction, i.e., traditional machine learning (with handcrafted features) and deep learning (without handcrafted features). The former include support vector machine (SVM) (Wang et al., 2017a)– (Chen et al., 2017a), Random Forest (Wang et al., 2018a)– (Frias and Hidalgo, 2021), Adaptive boosting (Adaboost) (Wang et al., 2018b), (Cord and Chambon, 2012), artificial neural network (ANN) (Wang et al., 2019)– (Cheng et al., 2001), etc. In traditional ML-based approaches, image processing is still required to implement pre-defined feature extraction. Various features have been utilized in research, such as statistical information, feature map projection, and defined defects' characteristics (Hsieh and Tsai, 2020). For example, Chen et al. (2017b) utilized local binary patterns (LBP), SVM, and Bayesian decision theory to detect cracks; Wang et al. (2017b) employed crack characteristics (i.e., density and connectivity) and SVM to discriminate alligator and transverse cracking. Meanwhile, ML can also be used to find optimal parameters for feature extraction, such as threshold values (Cheng et al., 2001), (Prasanna et al., 2016). The major problem with traditional ML approaches is that they still require handcrafted features and contain shallow learned information (or representation) (Hsieh and Tsai, 2020).

Deep learning (DL) can extract features automatically with multilayer neural networks. Cha et al. (2017) proposed a convolutional neural network (CNN) to identify cracks without calculating handcrafted features for the first time. The model was trained on 40 k images (256 × 256), including crack and non-crack, and then combined with the sliding window to scan any image larger than 256 × 256 for crack detection, which shows better performance and can detect concrete cracks in practical scenarios. Subsequently, a few datasets and DL approaches were created for damage detection based on supervised learning, including CNN (Fu et al., 2021), (Mohammed et al., 2020; Nie and Wang, 2019; Xu et al., 2019b), transformer (Wu et al., 2019; Liu et al., 2021; Fang et al., 2022; Pan et al., 2020; Xiang et al., 2022), etc. For example, Xu et al. (2019b) created an image set for automatic bridge crack detection. They proposed a CNN architecture by leveraging the atrous spatial pyramid pooling (ASPP) module and depth-wise separable

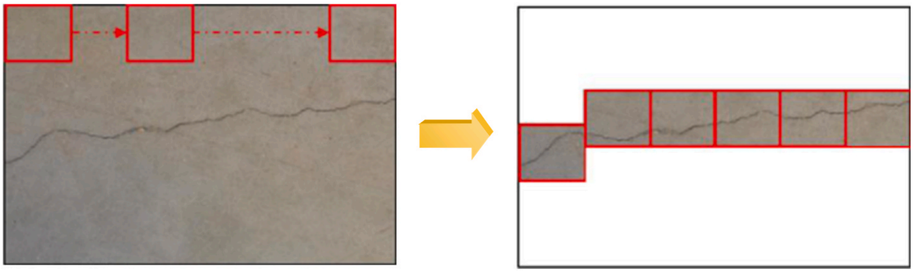


Fig. 1. Crack detection by patch splitting and classification (Cha et al., 2017).

convolution, which can achieve 96.37% accuracy on the test set. Xiang et al. (2022) integrated a transformer module in YOLOv5 for road crack detection. Cha et al. (2018) created a dataset including five typical defects – concrete rack, steel corrosion with two levels (medium and high), bolt corrosion, and steel delamination. Then, they employed the faster region-based convolutional neural network (Faster-RCNN) and the region proposal network (RPN) for multiple damage detection. Furthermore, Mundt et al. (2019b) developed a concrete defect bridge image dataset (CODEBRIM) of five commonly appearing concrete defects. They employed two meta-learning approaches based on reinforcement learning, i.e., MetaQNN and efficient neural architecture search, to find suitable CNN architectures for multi-class and multi-target damage detection.

The above ML and DL approaches are all based on inductive supervised learning, in which the performance relies on the pre-collected annotated examples before the inspection. They must work with pre-trained models to detect specific types of damage and cannot adapt themselves to novel defects quickly. However, annotation is usually time-consuming and tedious, and collecting sufficient defect images from various damage scenarios is not always practical. Traditional supervised transfer learning was expected to solve this issue, but it tends to be overfitting or challenging in convergence with only a few labeled examples. To our best knowledge, little research exists about weakly supervised learning for few-shot image-based bridge damage classification and detection. The only related one is an attribute-based approach (Xu et al., 2021) for structural damage identification through meta-learning, which relies on episodic training through a series of pre-collected tasks and is not developed to the level of damage detection.

In summary, the previous research about damage detection and their approaches are illustrated in Table 1. As can be seen, proposing an efficient transductive FSL approach, which can be exempt from episodic training, is beneficial to assist vision-based bridge damage detection without a tedious data acquisition process before the inspection. It will also promise fast implementation for damage detection under complex circumstances with weakly supervised information.

2.2. Few-shot learning for images

The time-consuming and labor-intensive data acquisition process is the bottleneck for applying supervised ML in many fields. FSL aims to solve this issue by learning from a limited number of annotated images, including few-shot classification and segmentation, which is essentially related to the data-efficiency problem. This work focuses on the few-shot classification, which is usually taken as an example of meta-learning. A meta-learner is trained through a series of related works (episodic training) to perform well to unseen but related tasks with just a few examples. Meanwhile, transduction has been widely adopted for FSL tasks in learning and inference because it is more effective at using only a few labeled examples than induction with supervised models (Vapnik, 1999).

Many great efforts have been made in this field, including a few specific image datasets (Lake et al., 2019; Bertinetto et al., 2019; Triantafillou et al., 2019; Wah et al., 2011) (such as Omniglot, CIFAR-FS, CUB, and mini-ImageNet) and various approaches. For example, a few works (Hu et al., 2019; Hariharan and Girshick, 2017; Chen et al., 2019b, 2019c; Zhou et al., 2022) aim to use data augmentation based on different methods to solve the few-shot classification with limited training samples, such as self-augmentation (Chen et al., 2019b), deformation (Chen et al., 2019c), and generation from DCGAN (Hu et al., 2019). Some other works aim to learn good model initialization (Rusu et al., 2019a), (Nichol and Schulman, 2018) or an optimizer (Ravi and Larochelle, 2017), (Finn et al., 2017a) to achieve rapid adaption with a limited number of training examples for new classes. In contrast, the other approaches aim to learn latent embeddings that can be used to compare (Chopra et al., 2005) or cluster (Laenen and Bertinetto, 2021) query items using appropriate metrics. It includes creating the exemplar for each class from the support set and selecting a metric for evaluation (Dhillon et al., 2019). For example, ProtoNet (Snell et al., 2017) calculates the mean vector of feature embedding as the prototype for each class in the support set and classifies query items as the nearest prototype based on the Euclidean distance because its case study fits Bregman divergence (Chen et al., 2021a) some other approaches prefer cosine similarity (Gidaris et al., 2018), (Chen et al., 2019a). Relation Network further developed the ProtoNet using a relation module as a learning metric in training (Sung et al., 2018).

However, the sophisticated meta-learning FSL approaches are based on episodic training through an intentionally collected series of related works, which is still time-consuming. Recently, a few works (Dhillon et al., 2020), (Chen et al., 2019a) have challenged the efficiency and effectiveness of this way by replacing episodic training with inter-class transfer learning (except the classes in the target FSL tasks). They can achieve similar state-of-the-art performance as the meta-learning approaches in the CUB and mini-ImageNet datasets. Furthermore, they

Table 1
Related works for image-based structural damage detection.

Names	Approaches	Research	Advantage/Disadvantage
Supervised Learning (Inductive)	Traditional ML	(Wang et al., 2017a, 2018a, 2018b, 2019; Fujita et al., 2017; Chen et al., 2017a; Shi et al., 2016; Luo et al., 2019; Frias and Hidalgo, 2021; Cord and Chambon, 2012; Moon and Kim, 2011; Hoang, 2018; Cheng et al., 2001)	Fast with good interpretability but require handcrafted features
	DL	(Fu et al., 2021), (Mohammed et al., 2020; Nie and Wang, 2019; Xu et al., 2019b; Wu et al., 2019; Liu et al., 2021; Fang et al., 2022; Pan et al., 2020; Xiang et al., 2022)	No need for handcrafted features but heavy and require time-consuming image acquisition
Few-shot Learning (Weakly Supervised)	Meta-learning	Xu et al. (2021)	Transductive inference with only a few examples but requires episodic training

have also indicated that the proper feature embeddings learned from cross-domain transfer learning (e.g., CUB → mini-ImageNet) can achieve competitive performance for FSL to the sophisticated meta-learning approaches (Dhillon et al., 2020), (Chen et al., 2019a). Moreover, the latest work (Cheng et al., 2022) has demonstrated the availability of cross-domain transfer learning (i.e., ImageNet → MSCOCO and PASCAL VOC) for few-shot segmentation. It is achieved by leveraging a "training-tree" module (i.e., a pre-trained CNN backbone from ImageNet) to learn the feature representation.

Therefore, leveraging cross-domain transfer learning for few-shot damage detection is promising. However, the domain differences in the previous studies (Dhillon et al., 2020), (Chen et al., 2019a), (Cheng et al., 2022) are not distinct enough compared to the domain difference from a public dataset to a specific engineering scenario, such as ImageNet → bridge structural defects (e.g., cracks, spalling, and corrosion). Therefore, this work aims to develop a transductive FSL approach for bridge damage detection using cross-domain transfer learning from a public dataset. It should be available for fast implementation under practical scenarios without episodic training and supervised learning, i.e., achieve similar "training-free" (Cheng et al., 2022). Hence, it is necessary to find a reliable source domain to perform effective feature embedding for few-shot damage detection and compare the performance of different pre-trained DNN backbones derived from supervised or unsupervised learning. Based on the transduction in the ProtoNet, it is also helpful to explore the performance of different metrics (i.e., Euclidean distance and cosine similarity) and propose a proper fine-tuning method for practical application.

3. Proposed approach and architecture

3.1. Theoretical foundation

3.1.1. Few-shot problem definition

Machine learning is said to learn from experience E to some classes of task T, and the performance is measured by P (Mitchell, 1997), e.g., E – ImageNet dataset, T – object recognition, and P – classification accuracy. Few-shot learning is a specific type of machine learning problem where E contains only a little supervised information for the task T. In the few-shot setting, the dataset D is separated into $D_{sup\ port}$ and D_{query}, as shown in Eq. (1) and Eq. (2). I is a very small integer, commonly from 1 to 5. In a standard N-way K-shot classification task, $D_{support}$ comes from N categories (N-way) with K samples (K-shot) per category, so there are $\mathrm{I} = N \times K$ support examples. D_{query} contains samples from the same N categories with Q samples per category. The goal is to classify the $N\times\ Q$ images into N categories based on the limited supervised information from $D_{support}$ (Chen et al.).

$$D_{support} = \{(x_i, y_i)\}_{i=1}^{I=N\times K} \tag{1}$$

$$D_{query} = \{x_j\}_{j=1}^{J=N\times Q} \tag{2}$$

Where N is the number of categories; K is the number of samples (i.e., the support items); x_i is the support item; y_i is the corresponding category for the support item; x_j is the query item.

Let $p(x, y)$ as the joint probability distribution of input x and label y. h is the hypothesis model mapping from x to y. Few-shot classification aims to learn h from $D_{support}$ for prediction and then test it in D_{query}. Here, h is parameterized as $h(\theta)$. The algorithm aims to find the optimal θ for $D_{support}$ in the vector space H. The model h performance is evaluated through the loss function $L(\widehat{y}, y)$ between the prediction value $\widehat{y} = h(x;\theta)$ and the actual value y.

Assuming vector space H, task T, and distribution $p(x, y)$, to minimize the loss function $L(\widehat{y}, y)$ equals to minimize the expected risk $R(h)$ with appropriate θ, which can be indicated in Eq. (3).

$$min\ R(h) = \min \int L(h(x;\theta), y)dp(x, y) = \min\ \mathbb{E}[L(h(x;\theta), y)] \tag{3}$$

In practice, posterior distribution from data sampling is utilized to approach $p(x, y)$ through machine learning. However, as $p(x, y)$ is unable to know, the empirical risk $R_I(h)$ is used to estimate $R(h)$, as indicated in Eq. (4).

$$R(h) \approx R_I(h) = \frac{1}{n}\sum L(h(x_i;\theta), y_i) \tag{4}$$

Hence, there will be three different optimal solutions (Wang et al., 2020), which are: 1) $\widehat{h} = argmin R(h)$ – global optimal solution; 2) $h^* = argmin_{h\in\mathrm{H}} R(h)$ – optimal solution in hypothesis space H; 3) $h_I = argmin_{h\in\mathrm{H}} R_I(h)$ – optimal solution in H for $R_I(h)$. Moreover, with model h trained from a random set for a task, its total error consists of two parts: 1) approximation error $\varepsilon_{app}(H)$ caused by the difference between the hypothesis space H and the global space; 2) estimation error $\varepsilon_{est}(H, I)$ is the impact of using empirical risk $R_I(h)$ instead of expected risk $R(h)$. Here, I is the set of training data. In theory, as the training set increases, $\varepsilon_{est}(H, I)$ converges to zero, as shown in Eq. (5).

$$\lim_{I\to\infty} \varepsilon_{est}(H, I) = \lim_{I\to\infty} \mathbb{E}[R(h_I) - R(h^*)] = 0 \tag{5}$$

However, as few-shot learning lacks plenty of training data, it becomes difficult to use $R_I(h)$ approaching $R(h)$ accurately. Therefore, the most difficulty of few-shot learning is the gap between the empirical best $h_I(I)$ and hypothesis best $h^*(H)$.

3.1.2. Meta-learning and feature embedding

Meta-learning approaches aim to learn prior knowledge from a series of training tasks to solve a new task. It includes hallucination-based (learning to augment), initialization-based (learning to fine-tune), and metric-based (learning to compare) approaches. The hallucination-based approaches (Hariharan and Girshick, 2017)– (Zhou et al., 2022) aim to generate more training examples of novel classes through data augmentation to alleviate the issue of insufficient data. The initialization-based approaches, e.g., MAML (Finn et al., 2017b), Reptiles (Nichol and Schulman, 2018), and LEO (Rusu et al., 2019b), aim to learn the optimal hyperparameter initialization to reach convergence with only a small number of data samples. The metric-based approaches, e.g., MatchingNet (Vinyals et al., 2016), ProtoNet (Snell et al., 2017), and RelationNet (Sung et al., 2018), aim to project data into an embedding space in which similar objects are close to each other and vice versa. The transductive inference process is to calculate the distance (or similarity) between $x_i \in D_{support}$ and $x_j \in D_{query}$, then the label y_i with the closest distance (or highest similarity) in $D_{support}$ is assigned as y_j in D_{query}. In detail, MatchingNet uses attention calculated from the cosine similarity of extracted features for classification; ProtoNet uses the mean vector of each class as the cluster center and Euclidean distance as the metric for classification; RelationNet employs relation module instead of Cosine similarity and Euclidean distance, generating a non-linear classifier based on relation score. These sophisticated meta-learning approaches are usually based on episodic training through a series of related tasks (episodes) sampled from the base dataset to simulate reasoning scenarios (Cheng et al., 2022).

Feature embedding (representation) is used to represent a data point $x_i \in X \subset \mathbb{R}^d$ in a low-dimension space $z_i \in Z \subset \mathbb{R}^m$ $(m < d)$, which is supposed to have three essential assumptions (Devgan et al., 2020), i.e., smoothness, clustering, and manifold. Feature embedding must retain consistent similarities or differences among data points in the original space. Embedding functions are usually in the form of DNN architectures. Note that feature representations through different embedding functions can have different properties, even if they are from the same data point, which can significantly impact the performance of downstream tasks. The hyper-parameters of the embedding function can be learned from prior knowledge or task-specific information, e.g., multiple sophisticated tasks or a related source domain.

The support embedding function and query embedding function are usually the same. The most straightforward way to learn the embedding function is training a model in the support set through supervised learning, but its parameters are prone to overfitting or difficult to converge under few-shot conditions. Hence, many existing few-shot learning works tackle this problem based on meta-learning, i.e., trained on a series of invariant tasks and then generalized to the target task. However, cross-domain transfer learning has been recently demonstrated as an effective way to initialize the feature embedding functions for few-shot classification (Dhillon et al., 2020), (Chen et al., 2019a) instead of meta-learning.

3.1.3. Transfer learning and fine-tuning

Transfer learning focuses on storing the knowledge learned while solving one task T_S in a source domain $\mathbb{R}_S$ and applying it to a different but related task T_T in a target domain $\mathbb{R}_T$. The correlative research problems, such as multi-task learning and domain adaption, are also related to few-shot learning and meta-learning (Panigrahi et al., 2021). In multi-task learning, the hypothesis space of each task strongly correlates with each other. This correlation (i.e., prior knowledge) can be represented through sharing hyperparameters of DNNs. According to explicit or implicit constraints in parameter space, the sharing methods can be classified into soft parameter sharing, which does not place a strong constraint on parameters but encourages them to meet some requirements, such as regulation function L_1 or L_2, and hard parameter sharing, such as freezing specific layers in DNN. The frozen layers can be part of the embedding function or just the classifier, which solidifies the prior knowledge learned from the source task T_S. At the same time, the rest of the network will be updated (i.e., fine-tuning) to adapt the target task T_T in the target domain.

Some meta-learning works have been developed to leverage transfer learning by learning the scaling and shifting functions of DNN weights through episodic training for each task, such as meta-transfer learning (Sun and Chua, 2019). Research (Chen et al., 2019a) has recently demonstrated that cross-domain transfer learning can achieve the comparable performance of (or even overperform) many state-of-the-art meta-learning approaches in few-shot classification. Moreover, fine-tuning can enhance average accuracy by 1%–2% on the CUB and ImageNet datasets (Dhillon et al., 2020), (Chen et al., 2019a). This progress enables few-shot classification to be exempt from episodic training and become “training-free” like (Cheng et al., 2022) by using pre-trained DNN backbones from a large-scale public dataset (e.g., ImageNet) for feature embedding.

3.2. Few-shot damage detection approach

3.2.1. Proposed architecture

The proposed approach for bridge damage detection is derived from the ProtoNet (Snell et al., 2017), which consists of episodic training through a series of related tasks and prototypical transduction based on Euclidean distance for few-shot classification. Its improvement includes three aspects: (1) previous episodic training is replaced with cross-domain transfer learning from ImageNet for “training-free” feature embedding; (2) embedding normalization is integrated to reduce domain variation and enhance the original ProtoNet performance based on Euclidean distance; (3) the fine-tuning methods based on fully connected network (FCN) and the Hadamard product can achieve better performance in fewer epochs compared to the previous transductive fine-tuning (Dhillon et al., 2020).

The approach architecture is shown in Fig. 2 with an example of 2-way 3-shot crack detection, and the steps are shown below.

1) **Image splitting into support and query sets** – an inspection image is split into multiple patches, in which the support and the query items are selected, respectively. Here, the patches marked with the blue boundary are picked up as the support set, while the rest patches are taken as the query set.
2) **Feature embedding** (cross-domain transfer learning) – the pre-trained DNN backbones from ImageNet are applied on both support and query items for feature embedding, which not only enables the feature embedding to be exempt from episodic training but also makes the process become “training-free” (no need to be trained from scratch).
3) **Feature normalization** – normalization is employed after feature embedding to reduce domain variation.
4) **Calculating prototypes** – the mean vector of the support feature embeddings is calculated as the prototype for each class, and the initial transductive inference can be taken based on Euclidean distance.
5) **Fine-tuning** – fine-tuning is employed to improve the linear classifier further using the support examples and the derived prototypes.
6) **Inference** – finally, the damage type, location, and skeleton can be obtained based on the inference for each patch. Meanwhile, the obtained prototypes and fine-tuned classifier can be applied to a new image to detect the specific defect.

The pseudocode of the algorithm is shown below.

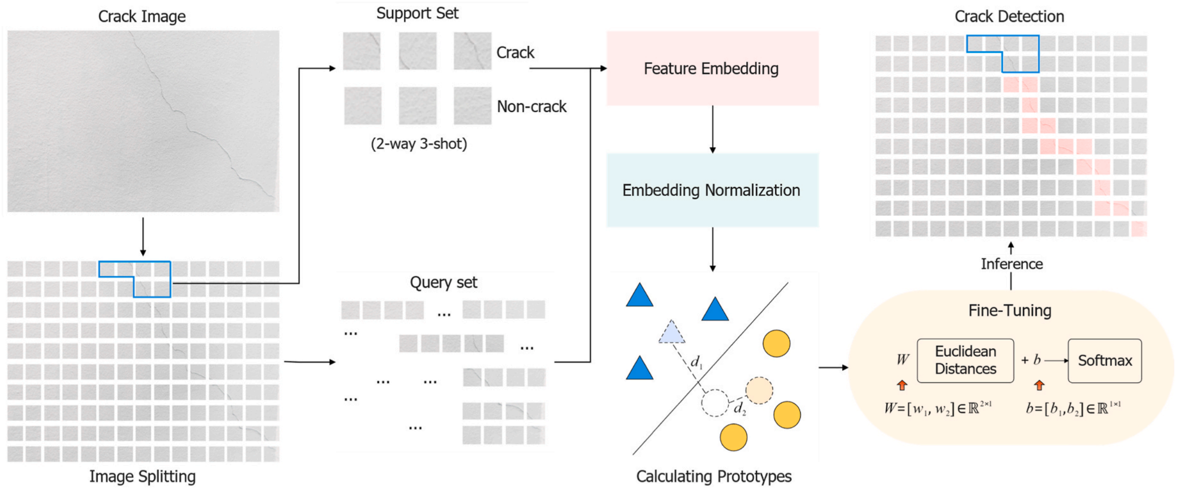


Fig. 2. Proposed approach for few-shot damage detection.

Algorithm 1 Few-shot damage detection (n-way, k-shot) based on cross-domain transfer learning. n is the number of classes. k is the number of support items for each class

```
Input: Support set $S = \{s_{ij}\}$, Query set $Q = \{q_{ij}\}$, $i \in [1, n], j \in [1, k]$, and pre-trained
       embedding function $f_\theta$
Output: Predicted query labels $Y_Q$
$v = f_\theta(s) and v = \frac{v}{max(\|v\|)}$        ▷ Support Feature embedding and normalization
$w = mean(v)$        ▷ Compute prototype for each class
$d_s = (\sum(v - w)^2)^{\frac{1}{2}}$        ▷ Calculate support Euclidean distance
$p_s = softmax(Wd_s + b)$        ▷ Establish a linear classifier
$\theta^* = argmin_\theta(-\frac{1}{n}\sum y_s log(p_s) + Regulation)$        ▷ Finetuning with target function
$u = f_\theta(q)$ and $u = \frac{u}{max(\|u\|)}$        ▷ Query feature embedding and normalization
$d_q = (\sum(u - w)^2)^{\frac{1}{2}}$        ▷ Calculate query Euclidean distance
return $Y_Q \leftarrow p_q = softmax(Wd_q + b)$        ▷ Prediction for query set
```

3.2.2. Domain adaption and transduction

In principle, the pre-trained DNN backbones and weights based on prior knowledge (e.g., from the related source domain) can help to constrain the hypothesis space into a smaller one for few-shot classification, as shown in Fig. 3, thereby achieving less ε_{est} quickly and better $R_I(h)$ (see Eq. (4) and Eq. (5)). The left ellipse in Fig. 3 shows the normal ε_{est} based on a large dataset, which is the goal to pursue. The middle one shows a bigger ε_{est} based on a small dataset (i.e., under FSL conditions), while the right one shows a decreased ε_{est} in a constrained hypothesis space by prior knowledge.

In the embedding module, the pre-trained DNN backbone (feature extractor) learned from 1000-class ImageNet of 12 million images is employed as the embedding function $f_\theta(x_i)$ for both support and query sets. Note that the object classes of ImageNet do not include the specific defects for detection, i.e., the source domain has a vast difference from the target domain. The embedding function $f_\theta(x_i)$ can be derived from supervised learning or self-supervised learning, as shown in Fig. 4. The former includes different DCNNs and vision transformers. The latter mainly involves masked image modeling (MIM) approaches, such as masked autoencoder (MAE) (He et al., 2022) or BEiT (Bao et al., 2021).

Although the hardcoded mean and the standard deviation obtained statistically from ImageNet, i.e., $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ can be employed for image transformation, it cannot guarantee the normalization in the target domain. Hence, normalization according to Eq. (6) (v is the embedding vector) is required for the obtained feature embeddings to minimize domain variation.

$$v_{norm} = \frac{v}{max(\|v\|_2)} \tag{6}$$

In the transductive inference, the mean vector of the support embeddings is computed as the prototype for each class. Then, the distances from the query embedding to each prototype are calculated. Consequently, the query item can be predicted as the closest prototype. The commonly used metrics include Euclidean distance and cosine similarity, as indicated in Eq. (7) and Eq. (8). Here, v and w are the query and prototype embedding vectors, respectively. As seen, embedding normalization enables the transduction based on Euclidean distance and cosine similarity to start from the same circumstance.

$$d = dist(v, w) = \left(\sum |v - w|^2\right)^{\frac{1}{2}} \tag{7}$$

$$s = cos\,\theta = \frac{v^T w}{\|v\|_2 \cdot \|w\|_2} \tag{8}$$

3.2.3. Loss function and fine-tuning

In the proposed architecture, the linear classifier $W^T x + b$ is utilized for few-shot classification. x can be either the query embedding vector v or the distances d between the query item and the prototypes. The softmax function is utilized as the output layer to convert the result to a probability distribution $p_i \in [0, 1]$ for each class, as shown in Eq. (9).

$$p_i = soft\,max(x_i) = \frac{e^{x_i}}{\sum_{n=1}^{N} e^{x_n}} \tag{9}$$

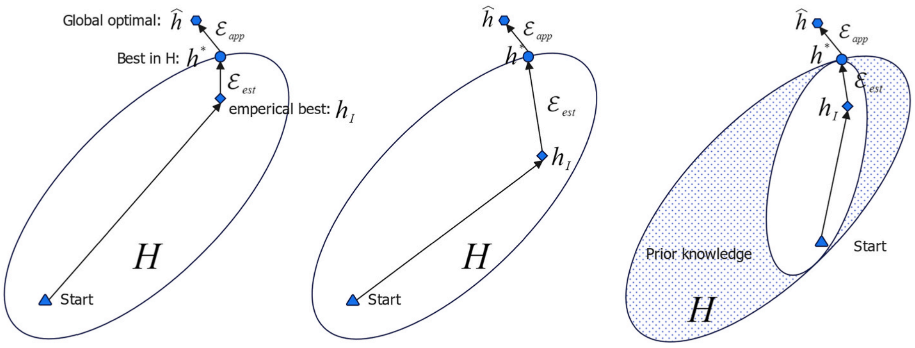


Fig. 3. Decreased ε_{est} in constrained hypothesis space by prior knowledge (Wang et al., 2020).

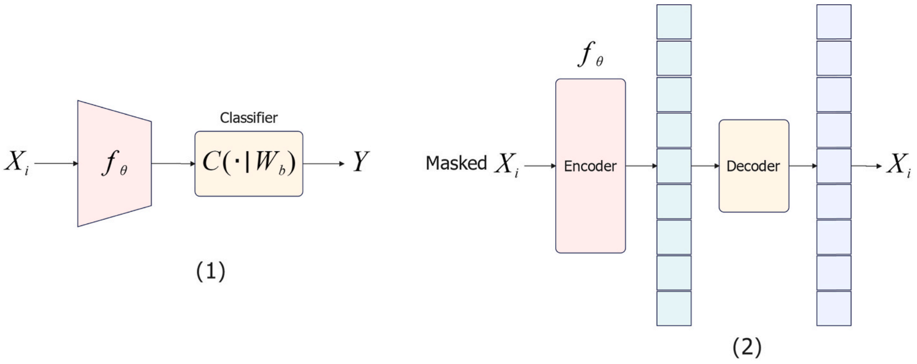


Fig. 4. (1) $f_\theta(x_i)$ from supervised learning; (2) $f_\theta(x_i)$ from self-supervised learning.

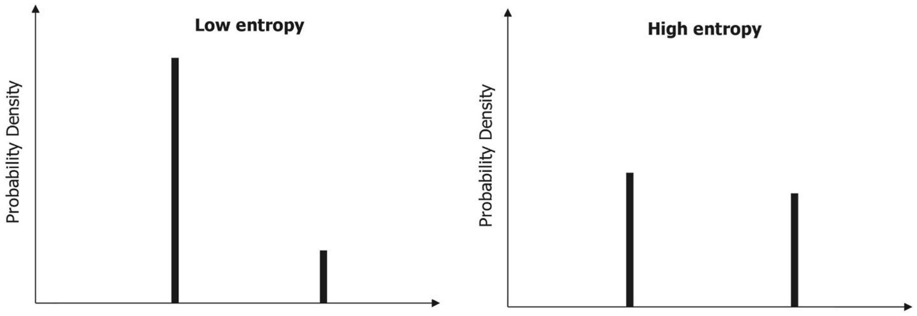


Fig. 5. Entropy increases along with uncertainty increasing in binary classification.

Then, the loss function L is defined based on binary cross-entropy, as indicated in Eq. (10).

$$L = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_i [y_i \, log_2(p_i) + (1-y_i) log_2(1-p_i)] \tag{10}$$

Where, y_i is the example label (0 or 1); p_i is the probability of y_i for the example i.

As the support set is quite small under few-shot conditions, the Shannon Entropy (Eq. (11)) is introduced as the regularization item to alleviate overfitting due to increased uncertainty in classification, as shown in Fig. 5. This is similar to the transductive fine-tuning method in (Dhillon et al., 2020), but the entropy $H(x)$ is calculated based on the support set rather than the query set because the model aims to be trained and fine-tuned before seeing all the query items in the practical inspection.

$$H(x) = -\sum_i p_i \cdot log_2 \frac{1}{p_i} \tag{11}$$

Hence, the fine-tuning step solves Θ^* to minimize the target function indicated in Eq. (12).

$$\Theta^* = \underset{\Theta}{argmin}\left(-\frac{1}{N}\sum_i [y_i \, log_2(p_i) + (1-y_i) log_2(1-p_i)] - \frac{1}{N}\sum_i p_i \, log_2(p_i) \right) \tag{12}$$

4. Experiments and approach validation

4.1. Experiment preparation

An image dataset created for automatic bridge crack detection in (Xu et al., 2019b) is employed for ablation studies using the proposed architecture for few-shot crack classification. The images were collected from real concrete bridges, including the 2014 background and 4055 crack images (224 × 224). The dedicated CNN in the previous research (Xu et al., 2019b) can reach 96.37% accuracy on the test set (train-test split of 80%:20%) based on supervised learning. Here, the experiment aims to explore the performance of the proposed approach for few-shot crack classification (2-way 1-shot or 2-way 5-shot) on the test set, i.e., with no access to the training set for supervised learning. It can mimic the situation for crack identification without a pre-trained supervised model. The query accuracy is illustrated in a boxplot based on 5000 samplings, recommended to compare FSL performance by (Dhillon et al., 2020). The random state remains unchanged to guarantee the reliability of ablation experiments. The experiments are taken on Google CodeLabs. The code is generated based on the original ProtoNet from a public GitHub project (https://github.com/sicara/easy-few-shot-learning).

4.2. Ablation studies

4.2.1. Domain adaption and normalization

The experiment starts with 2-way 1-shot and 2-way 5-shot crack identification. The ResNet18 backbone, popular in previous few-shot learning research (Dhillon et al., 2020), (Chen et al., 2021b), (Chen et al., 2019d), is employed as the feature embedding function. Its parameters are pre-trained on ImageNet, and the hardcoded mean $\mu = [0.485, 0.456, 0.406]$ and standard deviation $\sigma = [0.229, 0.224, 0.225]$, derived from ImageNet statistically, are utilized for image transformation. The raw and hardcoded-transformed images can be shown in Fig. 6. The image size is 224 × 224.

The performance of architecture with and without embedding normalization is explored in the experiment. Moreover, Euclidean distance and cosine similarity are tested as the evaluation metric. The results are shown in Fig. 7. Here, the annotation with raw and hard represents raw and hardcoded-transformed images, respectively; Eu indicates that the result is based on Euclidean distance of raw embedding vectors, while Eu_norm stands for Euclidean distances of embedding vectors after normalization; Cosine means using cosine similarity of raw embedding vectors.

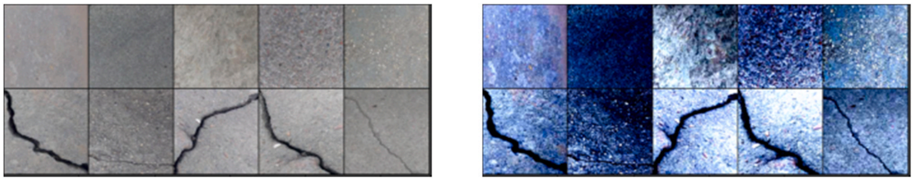


Fig. 6. Raw images (left) and hardcoded-transformed images (right).

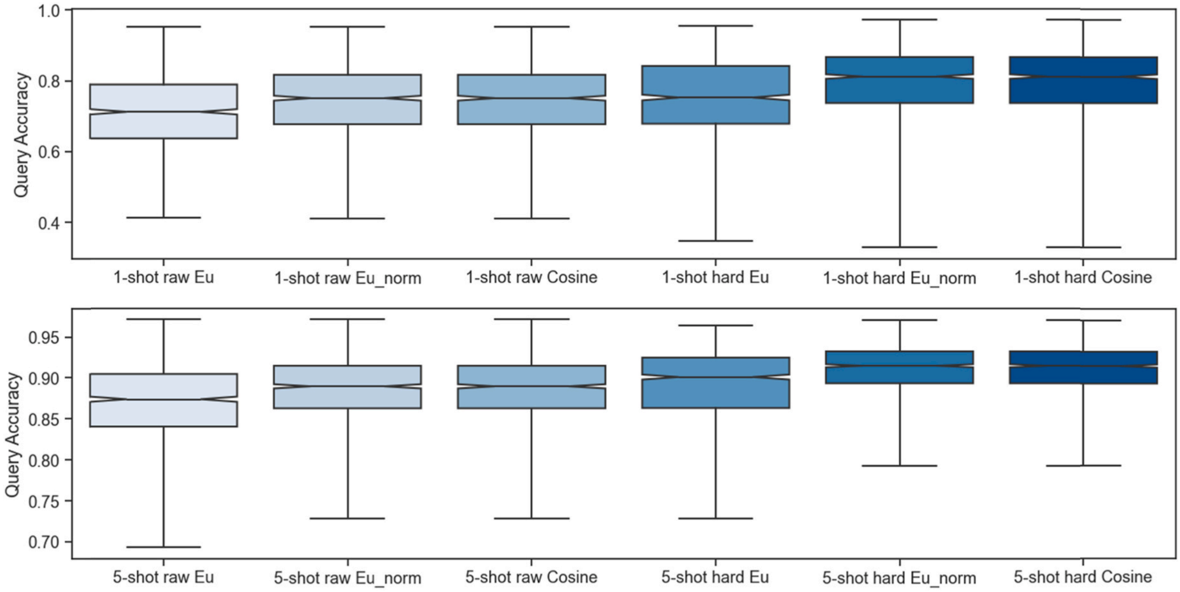


Fig. 7. 1-shot and 5-shot crack identification with pre-trained ResNet18 (224 × 224).

As can be seen, hard-coded transformation (i.e., hard) can significantly improve both 1-shot and 5-shot performance. After hard-coded transformation, it is shown with higher mean accuracy and narrower value distribution, i.e., interquartile range (IQR). IQR is calculated as $IQR = Q_3 - Q_1$ (Q_1 – the first quartile; Q_3 – the third quartile). Moreover, the Euclidean distance of normalized embeddings (i.e., Eu_norm) performs much better than the raw Euclidean distance (i.e., Eu_raw). The former has the equivalent performance as the cosine similarity, demonstrating that embedding normalization can bridge the gap between Euclidean distance and cosine similarity in the metric-based transduction for the few-shot classification in this dataset. Furthermore, 5-shot performs much better than 1-shot in both accuracy and IQR, which is promising to be enhanced as comparable to the dedicated supervised learning in previous research. Meanwhile, as the experiment aims to validate the proposed approach and figure out the appropriate conditions (such as feature embedding functions and fine-tuning methods) for practical application under weakly supervised scenarios, the 2-way 5-shot classification is adopted for the following experiment.

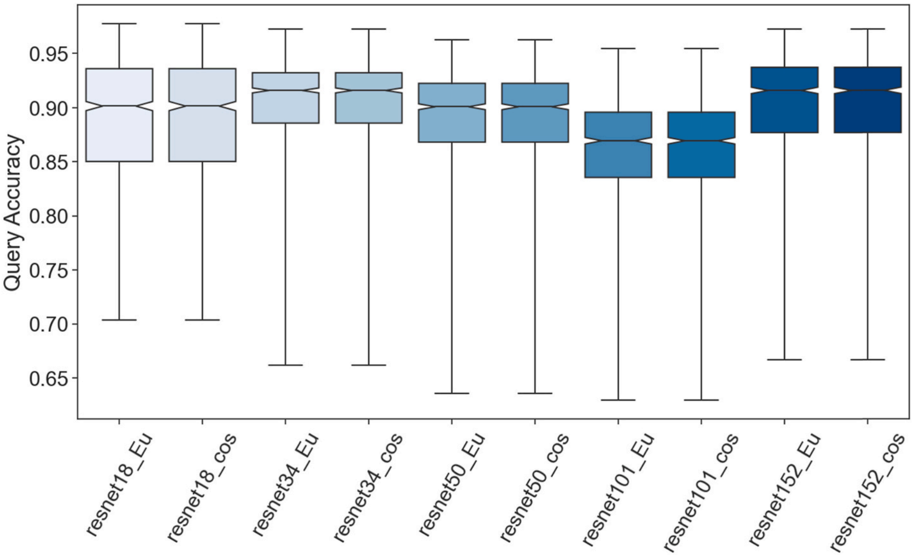


Fig. 8. 2-way 5-shot performance of ResNet backbones in different depths (84 × 84).

4.2.2. Different embedding functions

A series of ResNet backbones in different depths are employed in the experiment to explore the impact of DNN architecture depths on the few-shot performance. Their parameters are pre-trained on ImageNet. The experiment is conducted for 2-way 5-shot classification, and the approach integrates hard-coded transformation and embedding normalization. Euclidean distance and cosine similarity are tested as the evaluation metric in the experiment. The results are shown in Fig. 8.

Here, the images are resized to 84 × 84 to fit deep ResNets (such as ResNet152) due to CUDA memory limitation, so the ResNet18 performance differs from its previous result in Fig. 7 (224 × 224), i.e., the minimum accuracy drops to nearly 70%. Although the deeper ResNet has higher accuracy for image recognition in ImageNet, the experiment with different pre-trained ResNets for feature embedding cannot see a significant proportional relationship between the performance and the DNN depths for 2-way 5-shot classification, as shown in Fig. 8. Therefore, when using the pre-trained DNN backbones as embedding functions, their cross-domain few-shot performance does not necessarily correspond to their original performance in the source domain.

Moreover, different ResNets can perform diversely, even for the same sample. For example, ResNet18 has only 76.5% query accuracy for a sample (i.e., 5-shot for crack and 5-shot for non-crack), while ResNet152 can reach 91% for the same sample. Meanwhile, Euclidean distance and cosine similarity have the equivalent performance as the evaluation metric. Here, the pre-trained backbone ResNet34 has the best performance with the highest mean accuracy of 91.7% and narrower IQR in the series of ResNets for 2-way 5-shot classification in this dataset (images resized to 84 × 84).

Furthermore, the other prevalent DNN backbones are involved in the experiment, including multiple DCNN architectures and vision transformers (i.e., Swim Transformer and MAE). Their parameters are still pre-trained on ImageNet. The employed DNN models and their embedding dimensions are shown in Table 2.

The approach in the experiment is the same as the above for the ResNets, which integrates both hard-coded transformation and embedding normalization, and the experiment is taken under nearly the same condition. The only difference is that the pre-trained MAE can only be applied on the 224 × 224 images, which cannot take all the rest images (except the support images) as the query set due to CUDA limitation. Hence, the experiment with the pre-trained MAE for feature embedding is taken on the original 224 × 224 images with the randomly selected 50 images per class as the support set every time. In contrast, the experiment with the other pre-trained DNN backbones is taken under the same condition as the above, i.e., with resized images (84 × 84) and all the left images as the support set. Both Euclidean distance and cosine similarity are tested in the experiment. The results are shown in Fig. 9.

As can be seen, the pre-trained DNN backbones can achieve excellent performance for 2-way 5-shot classification. The improved ProtoNet can reach a mean accuracy of over 93% via GoogleNet and Swim Transformer, which proves that ImageNet is a reliable source domain for few-shot crack detection. Note that the pre-trained MAE encoder is derived from self-supervised learning, demonstrating the availability of a training embedding function without supervised information (i.e., labels). It also demonstrates that ImageNet is a reliable source domain for few-shot crack identification based on cross-domain transfer learning. Moreover, the Euclidean distance of the normalized embeddings can achieve the equivalent performance as cosine similarity for the transductive inference.

Table 2
Pre-trained embedding functions and embedding dimensions.

Embedding function	Pre-trained Models	Embedding dimensions	Input size
AlexNet	alexnet	9216	84 × 84
VGG	vgg16	25,088	84 × 84
DenseNet	densenet161	2208	84 × 84
EfficientNet	efficientnet_v2	1208	84 × 84
ResNet	resnet34	512	84 × 84
MobileNet	mobilenet_v3_large	960	84 × 84
GoogleNet	googlenet	1024	84 × 84
Swim Transformer	swim_t	768	84 × 84
MAE	mae_visualize_vit_base	768	224 × 224

4.2.3. Fine-tuning and comparison

Fine-tuning aims to improve the few-shot classification performance based on transduction after feature embedding through the pre-trained DNN backbones. Its target function can be seen in 3.2.3. Here, three different fine-tuning methods are compared in the experiment, including the Baseline and FCN-based (modified Baseline++) methods, which are inspired by previous research (Dhillon et al., 2020), (Chen et al., 2019a), and a proposed method based on Hadamard product (i.e., element-wise product). Meanwhile, fine-tuning with and without the Shannon Entropy regularization (see Eq. (11)) is also explored in the experiment. The entropy is calculated based on the support set rather than the query set because the model aims to be trained and fine-tuned before seeing all the query images. This process is different from the previous research (Dhillon et al., 2020).

1) The first linear classifier is implemented by adding a linear layer after the normalized feature embedding, similar to the Baseline in (Chen et al., 2019a) and transductive fine-tuning in (Dhillon et al., 2020). Its formula is indicated in Eq. (13), where n is the number of classes ($n = 2$), and m is the embedding dimension. $x_{m\times 1}$ is the normalized feature embedding of each support example. $W_{n\times m}$ is initialized with the prototype matrix $M_{n\times m}$ (i.e., the stack of prototype embedding vectors $[w_1, w_2] \in \mathbb{R}^{1\times m}$) because it can help hyperparameters converge quickly and achieve better performance, as suggested in (Dhillon et al., 2020). $b_{n\times 1}$ is the bias and initialized from 0.

$$y_{n\times 1} = soft\,max(W_{n\times m} \cdot x_{m\times 1} + b_{n\times 1}) \tag{13}$$

2) The second one is adding an FCN after Euclidean distance, as indicated in Eq. (14), which is similar to the Baseline++ in (Chen et al., 2019a) and taken as the modified Baseline++. $d_{n\times 1}$ represents Euclidean distances from a support example to each prototype. $W_{n\times n}$ and $b_{n\times 1}$ are initialized from an identity matrix and 0, respectively.

$$y_{n\times 1} = soft\,max(W_{n\times n} \cdot d_{n\times 1} + b_{n\times 1}) \tag{14}$$

3) The third one is based on the Hadamard product by adding a linear layer with fewer hyperparameters after Euclidean distance, as indicated in Eq. (15). $d_{n\times 1}$ represents Euclidean distances from a support example to each prototype. $W_{n\times 1}$ and $b_{n\times 1}$ are initialized from 1 and 0, respectively.

$$y_{n\times 1} = soft\,max(W_{n\times 1} \odot d_{n\times 1} + b_{n\times 1}) \tag{15}$$

Here, the experiment employs the GoogleNet result for fine-tuning as it performs well in both query accuracy and IQR. The experiment is taken using RMSProp optimizer at the learning rate of 0.01 until 2000 epochs. The mean query accuracies and 95% confidence interval of different fine-tuning methods with and without entropy regularization are shown in Figs. 10 and 11.

As can be seen, both the FCN-based (i.e., modified Baseline++) and the Hadamard product fine-tuning methods perform much better than the Baseline (i.e., transductive fine-tuning (Dhillon et al., 2020)), which can enhance the mean query accuracy from 93.4% to over 94%.

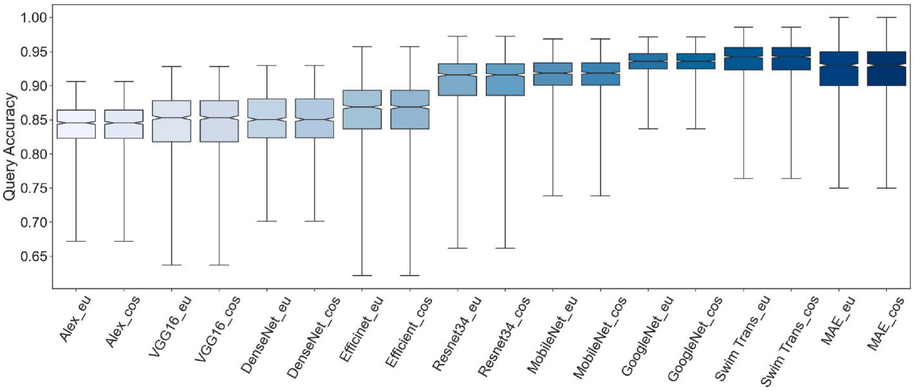


Fig. 9. Comparison of different pre-trained DNN embedding functions.

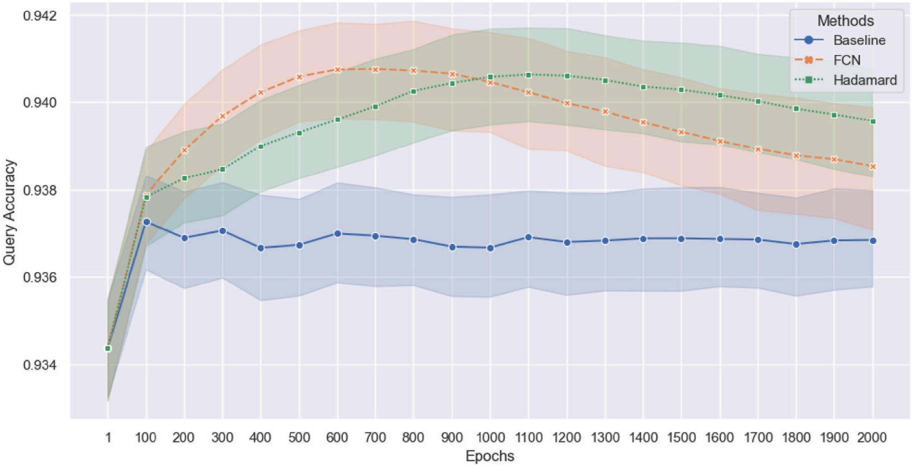


Fig. 10. Fine-tuning without entropy regularization.

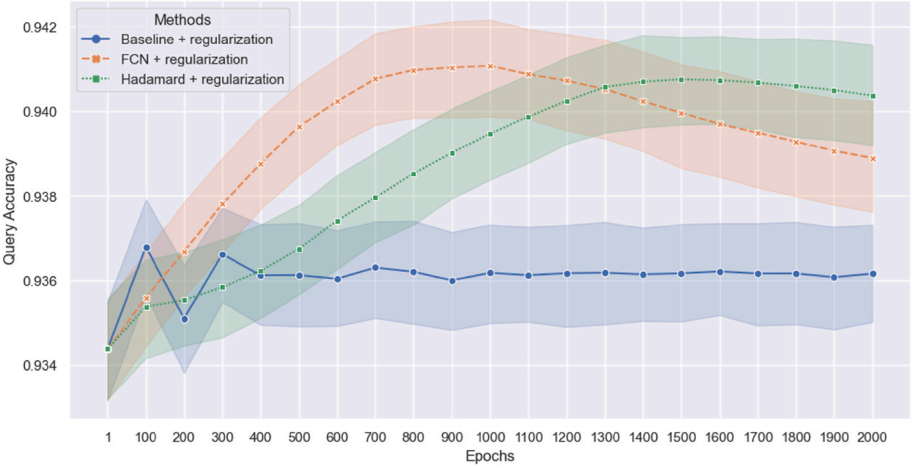


Fig. 11. Fine-tuning with entropy regularization.

Moreover, the FCN-based method can reach the peak faster than the Hadamard-product method in terms of accuracy during fine-tuning. Entropy regularization will slow down the fine-tuning of both methods and postpone their time to reach the peak. After the peak, there is overfitting for both methods. Hence, early stopping should be taken at the epoch number where query accuracy reaches the peak. As can be seen, early stopping can be determined empirically for few-shot crack detection as 600 epochs and 1000 epochs when using the proposed methods without regularization. Similarly, 1000 epochs and 1500 epochs are recommended for both methods with regularization.

In principle, it is difficult to avoid overfitting in few-shot classification because it is triggered by the discrepancy between the support examples and the overall items. If the support examples are representative, fine-tuning by fitting the model to the selected examples can enhance the query accuracy. On the contrary, fine-tuning will deteriorate the model and decrease its generalization capability if the support examples are unrepresentative. It can also be observed that the support set with increased accuracy after the Baseline fine-tuning can get more increment after the FCN and Hadamard-product fine-tuning. At the same time, the other two methods can also amplify the accuracy decrement after the Baseline fine-tuning.

4.3. Few-shot damage detection

The approach is also validated with the real bridge inspection images from the CODEBRIM dataset (Mundt et al., 2019a). The images are resized to 1260 × 840 and split into 150 patches (84× 84). A few patches with and without target defects are selected as the support set, while the others are taken as the query set. The embedding function is selected from the pre-trained DNN backbones based on ImageNet, and the classifier is fine-tuned with the support examples. Subsequently, the transductive inference is applied on each query patch using the obtained prototypes and fine-tuned classifier for damage detection. The pre-trained VGG16, VGG19, Swim Transformer, and MAE performed well as embedding functions in the experiment. Here, the results are shown based on the MAE encoder derived from self-supervised learning for feature embedding, in which each patch is resized to 224 × 224 for inference as required by Vision Transformer (i.e., ViT-Base). Moreover, the time cost is also tested for damage detection using different embedding functions.

An example of 2-way 2-shot crack detection on the real bridge inspection images is shown in Fig. 12. The support examples are from the first image in the top right, which is marked with a blue boundary, and the approach can recognize the crack skeleton with only two shots. The obtained prototypes and fine-tuned classifier can be applied on a new image directly for crack detection in the bottom right. As can be seen, most crack areas can be identified correctly, but a few crack patches were not recognized due to stains, which is related to the approach's robustness.

Spalling with rebar corrosion is another typical defect on the reinforced concrete bridge. An example of 2-way 5-shot spalling detection on the real bridge inspection images is shown in Fig. 13. The support patches are from the first image in the top right, marked with a blue boundary, and the approach can recognize the most spalling areas. Similarly, when applied on a new image in the bottom right through the identical prototypes and fine-tuned classifier, the spalling areas can also be identified well.

The time cost of the approach by using different embedding functions for each patch (84 × 84) is shown in Table 3. As seen, the time cost increases as the model complexity and input image size increase.

5. Conclusion

The current image-based approaches for drone-enabled bridge inspection still mainly rely on supervised learning, which requires time-consuming data acquisition and labor-intensive data annotation. These inductive approaches are inappropriate for practical damage detection under complex circumstances without enough supervised information, such as different materials, novel defects, and changing light. To solve this issue, this work proposes an approach based on improved ProtoNet for bridge damage detection under few-shot conditions (with only a few annotated examples).

In the approach, feature embedding is achieved by cross-domain transfer learning from ImageNet, which enables the embedding function to be not only exempt from episodic training but also become "training-free", i.e., no need to be trained from scratch. Moreover, after feature embedding, normalization is integrated into the ProtoNet to reduce the domain variation and enhance the transduction performance based on Euclidean distance. The linear classifier is added at the end of the ProtoNet for classification, and fine-tuning based on the support set can be further leveraged to improve the performance.

The approach is explored in a public automatic bridge crack detection dataset through extensive ablation studies. The experiment proves that ImageNet is a reliable source domain for few-shot damage detection and can achieve a mean accuracy of over 94% for 2-way 5-shot classification in the test set via the pre-trained GoogleNet after fine-tuning. The performance is already close to supervised learning using a dedicated CNN architecture. Moreover, the proposed fine-tuning methods based on the FCN and the Hadamard product demonstrated better performance than those in previous research (Dhillon et al., 2020), (Chen et al., 2019a). The time for early stopping can be determined empirically in the experiment. Furthermore, the approach is also validated using

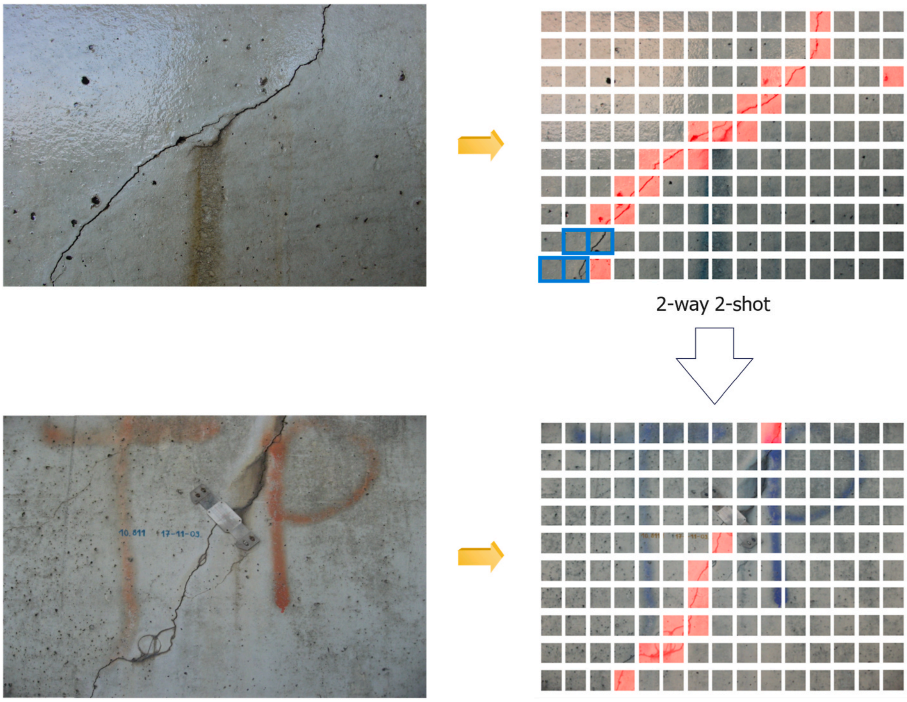


Fig. 12. Few-shot crack detection through the approach based on MAE.

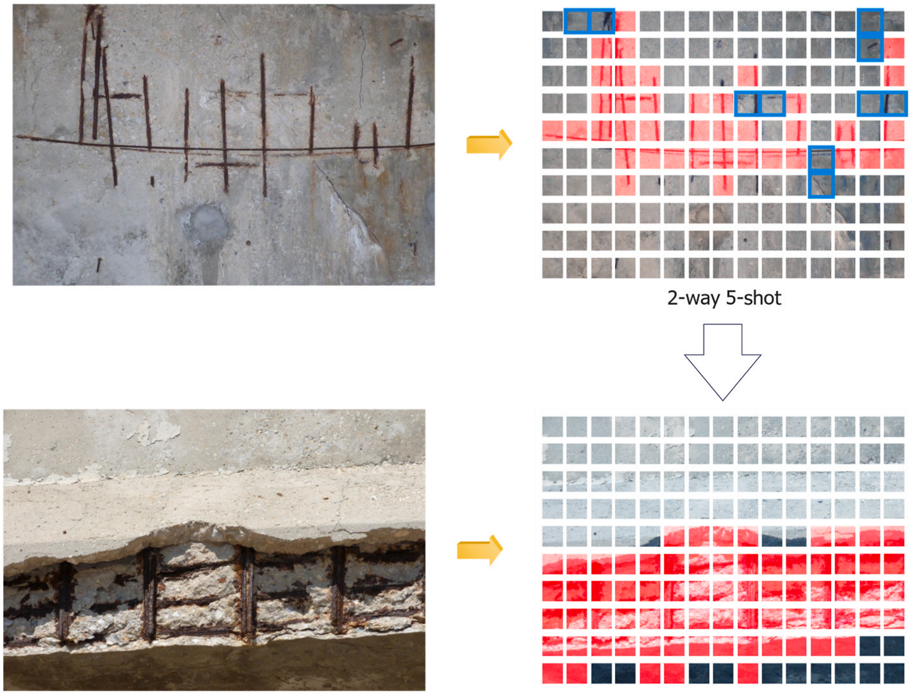


Fig. 13. Few-shot spalling detection through the approach based on MAE.

Table 3
Time costs of the approach using different embedding functions.

Embedding function	Pre-trained Models	Patch size	Time cost
VGG16	vgg16	84 × 84	0.08s/patch
VGG19	vgg19	84 × 84	0.08s/patch
Swim Transformer	swim_t	84 × 84	0.101s/ patch
MAE	mae_visualize_vit_base	224 × 224 (resized)	0.25s/patch

real bridge inspection images, demonstrating its capability of fast implementation for damage detection with weakly supervised information and the potential for practical application in near real-time.

Although the approach has the above advantages, it still has a few limitations. Firstly, the approach is sensitive to noise, such as oil stains, road marks, shadows, and bridge joints. Therefore, enhancing the approach's robustness in the next step would be helpful. Secondly, the current approach only focuses on binary classification in fixed patches. Hence, it is difficult to identify a specific defect in a step when different kinds of defects coexist in one image, especially for similar damage with different ROI (region of interest) sizes, such as potholes and cracks. The hierarchical ensemble learning and flexible region proposals are promising to solve this issue. Secondly, the support examples should be representative across the overall items because different support sets will result in different performances in damage detection. However, it requires a combination of machine learning and domain knowledge. Hence, how to select the support examples needs further study.

CRediT authorship contribution statement

Yan Gao: Conceptualization, Methodology, Software, Investigation, Writing. **Haijiang Li:** Conceptualization, Methodology, Supervision. **Weiqi Fu:** Methodology, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Cardiff University – China Scholarship Council (CSC) joint program.

References

Bao, H., Dong, L., Piao, S., Wei, F., 2021. BEiT: BERT Pre-training of Image Transformers. Mim, pp. 1–18.

Bertinetto, L., Torr, P.H.S., Henriques, J., Vedaldi, A., 2019. Meta-learning with differentiable closed-form solvers. In: 7th International Conference on Learning Representations, ICLR 2019, pp. 1–15.

Cha, Y.J., Choi, W., Büyüköztürk, O., 2017. Deep learning-based crack damage detection using convolutional neural networks. Comput. Aided Civ. Infrastruct. Eng. 32 (5), 361–378. https://doi.org/10.1111/mice.12263.

Cha, Y.J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O., 2018. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput. Aided Civ. Infrastruct. Eng. 33 (9), 731–747. https://doi.org/10.1111/mice.12334.

Y. Chen, T. Darrell, and X. Wang, 'Meta-Baseline : Exploring Simple Meta-Learning for Few-Shot Learning', pp. 9062–9071..

Chen, F.C., Jahanshahi, M.R., Wu, R.T., Joffe, C., 2017a. A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces. Comput. Aided Civ. Infrastruct. Eng. 32 (4), 271–287. https://doi.org/10.1111/mice.12256.

Chen, F.C., Jahanshahi, M.R., Wu, R.T., Joffe, C., 2017b. A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces. Comput. Aided Civ. Infrastruct. Eng. 32 (4), 271–287. https://doi.org/10.1111/mice.12256.

Chen, W.Y., Wang, Y.C.F., Liu, Y.C., Kira, Z., Bin Huang, J., 2019a. A closer look at few-shot classification. In: 7th International Conference on Learning Representations, ICLR 2019, pp. 1–17, 2018.

Chen, Z., Fu, Y., Chen, K., Jiang, Y.G., 2019b. Image block augmentation for one-shot learning. In: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st

Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, vol. 1, pp. 3379–3386. https://doi.org/10.1609/aaai.v33i01.33013379.

Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M., 2019c. Image deformation meta-networks for one-shot learning. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. 2019-June (c), 8672–8681. https://doi.org/10.1109/CVPR.2019.00888.

Chen, W.Y., Wang, Y.C.F., Liu, Y.C., Kira, Z., Bin Huang, J., 2019d. A closer look at few-shot classification. In: 7th International Conference on Learning Representations, ICLR 2019, pp. 1–17, 2018.

Chen, Y., Liu, Z., Xu, H., Darrell, T., Wang, X., 2021a. Meta-baseline: exploring simple meta-learning for few-shot learning. Proceedings of the IEEE International Conference on Computer Vision 9042–9051. https://doi.org/10.1109/ICCV48922.2021.00893.

Chen, Y., et al., 2021b. Meta-baseline: exploring simple meta-learning for few-shot learning. Proceedings of the IEEE International Conference on Computer Vision 9042–9051. https://doi.org/10.1109/ICCV48922.2021.00893.

Cheng, H.D., Wang, J., Hu, Y.G., Glazier, C., Shi, X.J., Chen, X.W., 2001. Novel approach to pavement cracking detection based on neural network. Transport. Res. Rec. 1764, 119–127. https://doi.org/10.3141/1764-13.

Cheng, G., Lang, C., Han, J., 2022. Holistic prototype activation for few-shot segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 45 (4), 4650–4666. https://doi.org/10.1109/TPAMI.2022.3193587.

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In: Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. I, pp. 539–546. https://doi.org/10.1109/CVPR.2005.202.

Cord, A., Chambon, S., 2012. Automatic road defect detection by textural pattern recognition based on AdaBoost. Comput. Aided Civ. Infrastruct. Eng. 27 (4), 244–259. https://doi.org/10.1111/j.1467-8667.2011.00736.x.

Devgan, M., Malik, G., Sharma, D.K., 2020. Semi-Supervised Learning. https://doi.org/10.1002/9781119654834.ch10.

Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S., 2019. A Baseline for Few-Shot Image Classification, pp. 1–20 [Online]. Available: http://arxiv.org/abs/1909.02729.

Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S., 2020. A baseline for few-shot image classification. In: 8th International Conference on Learning Representations, ICLR 2020, pp. 1–20.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Fang, J., Yang, C., Shi, Y., Wang, N., Zhao, Y., 2022. External attention based TransUNet and label expansion strategy for crack detection. IEEE Trans. Intell. Transport. Syst. 23 (10), 19054–19063. https://doi.org/10.1109/TITS.2022.3154407.

Finn, C., Abbeel, P., Levine, S., 2017a. Model-agnostic meta-learning for fast adaptation of deep networks. In: 34th International Conference on Machine Learning, ICML 2017, vol. 3, pp. 1856–1868.

Finn, C., Abbeel, P., Levine, S., 2017b. Model-agnostic meta-learning for fast adaptation of deep networks. In: 34th International Conference on Machine Learning, ICML 2017, vol. 3, pp. 1856–1868.

Frias, D., Hidalgo, J., 2021. A High Accuracy Image Hashing and Random Forest Classifier for Crack Detection in Concrete Surface Images, pp. 1–13 [Online]. Available: http://arxiv.org/abs/2106.05755.

Fu, H., Meng, D., Li, W., Wang, Y., 2021. Bridge crack semantic segmentation based on improved deeplabv3+. J. Mar. Sci. Eng. 9 (6) https://doi.org/10.3390/jmse9060671.

Fujita, Y., Shimada, K., Ichihara, M., Hamamoto, Y., 2017. A method based on machine learning using hand-crafted features for crack detection from asphalt pavement surface images. In: Thirteenth International Conference on Quality Control by Artificial Vision 2017, vol. 10338, p. 103380I. https://doi.org/10.1117/12.2264075.

Gidaris, S., Komodakis, N., Paristech, P., Komodakis, N., 2018. Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4367–4375 [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/papers/Gidaris_Dynamic_Few-Shot_Visual_CVPR_2018_paper.pdf.

Hariharan, B., Girshick, R., 2017. Low-shot visual recognition by shrinking and hallucinating features. Proceedings of the IEEE International Conference on Computer Vision 2017-Octob, 3037–3046. https://doi.org/10.1109/ICCV.2017.328.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., Girshick, R., 2022. Masked Autoencoders Are Scalable Vision Learners, pp. 15979–15988. https://doi.org/10.1109/cvpr52688.2022.01553.

Hoang, N.D., 2018. An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction. Adv. Civ. Eng. 2018 https://doi.org/10.1155/2018/7419058.

Hsieh, Y.-A., Tsai, Y.J., 2020. Machine learning for crack detection: review and model performance comparison. J. Comput. Civ. Eng. 34 (5), 1–12. https://doi.org/10.1061/(asce)cp.1943-5487.0000918.

Hu, G., Wu, H., Zhang, Y., Wan, M., 2019. A low shot learning method for tea leaf's disease identification. Comput. Electron. Agric. 163 (May) https://doi.org/10.1016/j.compag.2019.104852.

König, J., Jenkins, M., Mannion, M., Barrie, P., Morison, G., 2022. 'What's Cracking? A Review and Analysis of Deep Learning Methods for Structural Crack Segmentation, Detection and Quantification', pp. 1–18.

Laenen, S., Bertinetto, L., 2021. On episodes, prototypical networks, and few-shot learning. Adv. Neural Inf. Process. Syst. 29 (NeurIPS), 24581–24592.

Lake, B.M., Salakhutdinov, R., Gross, J., Tenenbaum, J.B., 2011. One shot learning of simple visual concepts. In: Expanding the Space of Cognitive Science - Proceedings of the 33rd Annual Meeting of the Cognitive Science Society, pp. 2568–2573. CogSci 2011.

Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B., 2019. The Omniglot challenge: a 3-year progress report. Curr Opin Behav Sci 29, 97–104. https://doi.org/10.1016/j.cobeha.2019.04.007.

Lin, T.-Y., et al., 2014. Microsoft coco: common objects in context. In: European Conference on Computer Vision. Springer, pp. 740–755.

Liu, H., Miao, X., Mertz, C., Xu, C., Kong, H., 2021. CrackFormer: transformer network for fine-grained crack detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3763–3772. https://doi.org/10.1109/ICCV48922.2021.00376.

Luo, Q., Ge, B., Tian, Q., 2019. A fast adaptive crack detection algorithm based on a double-edge extraction operator of FSM. Construct. Build. Mater. 204, 244–254. https://doi.org/10.1016/j.conbuildmat.2019.01.150.

Mitchell, T., 1997. Machine Learning. McGraw-Hill Education.

Mohammed, Y., Uddin, N., Tan, C., Shi, Z., 2020. Crack Detection using Faster R-CNN and Point Feature Matching 10 (3). https://doi.org/10.19080/CERJ.2020.10.555790.

Moon, H.G., Kim, J.H., 2011. Inteligent crack detecting algorithm on the concrete crack image using neural network. In: Proceedings of the 28th International Symposium on Automation and Robotics in Construction, ISARC 2011, pp. 1461–1467. https://doi.org/10.22260/isarc2011/0279.

Mundt, M., Majumder, S., Murali, S., Panetsos, P., Ramesh, V., 2019a. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June, pp. 11188–11197. https://doi.org/10.1109/CVPR.2019.01145.

Mundt, M., Majumder, S., Murali, S., Panetsos, P., Ramesh, V., 2019b. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. 2019-June, 11188–11197. https://doi.org/10.1109/CVPR.2019.01145.

Nichol, A., Schulman, J., 2018. *arXiv preprint arXiv:1803*.02999. Reptile: a Scalable Metalearning Algorithm, vol. 2, p. 4, 3.

Nie, M., Wang, C., 2019. Pavement Crack Detection based on yolo v3. In: Proceedings - 2019 2nd International Conference on Safety Produce Informatization, IICSPI, pp. 327–330. https://doi.org/10.1109/IICSPI48186.2019.9095956, 2019.

Nuthalapati, S.V., Tunga, A., 2021. Multi-domain few-shot learning and dataset for agricultural applications. In: Proceedings of the IEEE International Conference on Computer Vision, 2021-Octob, pp. 1399–1408. https://doi.org/10.1109/ICCVW54120.2021.00161.

Pan, Y., Zhang, G., Zhang, L., 2020. A spatial-channel hierarchical deep learning network for pixel-level automated crack detection. Autom. ConStruct. 119 (July), 103357 https://doi.org/10.1016/j.autcon.2020.103357.

Panigrahi, S., Nanda, A., Swarnkar, T., 2021. 'A Survey on Transfer Learning', *Smart Innovation, Systems And Technologies*, vol. 194, pp. 781–789. https://doi.org/10.1007/978-981-15-5971-6_83, 10.

Prasanna, P., et al., 2016. Automated crack detection on concrete bridges. IEEE Trans. Autom. Sci. Eng. 13 (2), 591–599. https://doi.org/10.1109/TASE.2014.2354314.

Ravi, S., Larochelle, H., 2017. Optimization as a model for few-shot learning. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, pp. 1–11.

Rusu, A.A., et al., 2019a. Meta-learning with latent embedding optimization. In: 7th International Conference on Learning Representations, ICLR 2019, pp. 1–17.

Rusu, A.A., et al., 2019b. Meta-learning with latent embedding optimization. In: 7th International Conference on Learning Representations, ICLR 2019, pp. 1–17.

Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. IEEE Trans. Intell. Transport. Syst. 17 (12), 1–12. https://doi.org/10.1109/TITS.2016.2552248.

Snell, J., Swersky, K., Zemel, R., 2017. Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30.

Sun, Q., Chua, Y.L.T., 2019. Meta-Transfer Learning for Few-Shot Learning, pp. 403–412.

Sung, F., Yang, Y., Zhang, L., 2018. Relation Network for Few-Shot Learning, pp. 1199–1208. Cvpr.

Triantafillou, E., et al., 2019. Meta-dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *arXiv preprint arXiv:1903.03096*.

Vapnik, V., 1999. The Nature of Statistical Learning Theory. Springer science & business media.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D., 2016. Matching Networks for One Shot Learning. Adv Neural Inf Process Syst, pp. 3637–3645. Nips.

Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., 2011. The Caltech-Ucsd Birds-200-2011 Dataset.

Wang, S., Qiu, S., Wang, W., Xiao, D., Wang, K.C.P., 2017a. Cracking classification using minimum rectangular cover–based support vector machine. J. Comput. Civ. Eng. 31 (5), 1–9. https://doi.org/10.1061/(asce)cp.1943-5487.0000672.

Wang, S., Qiu, S., Wang, W., Xiao, D., Wang, K.C.P., 2017b. Cracking classification using minimum rectangular cover–based support vector machine. J. Comput. Civ. Eng. 31 (5), 1–9. https://doi.org/10.1061/(asce)cp.1943-5487.0000672.

Wang, S., Liu, X., Yang, T., Wu, X., 2018a. Panoramic crack detection for steel beam based on structured random forests. IEEE Access 6, 16432–16444. https://doi.org/10.1109/ACCESS.2018.2812141.

Wang, S., Yang, F., Cheng, Y., Yang, Y., Wang, Y., 2018b. Adaboost-based crack detection method for pavement. IOP Conf. Ser. Earth Environ. Sci. 189 (2) https://doi.org/10.1088/1755-1315/189/2/022005.

Wang, L., Zhuang, L., Zhang, Z., 2019. Automatic detection of rail surface cracks with a superpixel-based data-driven framework. J. Comput. Civ. Eng. 33 (1), 1–9. https://doi.org/10.1061/(asce)cp.1943-5487.0000799.

Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M., 2020. Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. 53 (3) https://doi.org/10.1145/3386252.

Wu, S., Fang, J., Zheng, X., Li, X., 2019. Sample and structure-guided network for road crack detection. IEEE Access 7, 130032–130043. https://doi.org/10.1109/ACCESS.2019.2940767.

Xiang, X., Wang, Z., Qiao, Y., 2022. An improved YOLOv5 crack detection method combined with transformer. IEEE Sensor. J. 22 (14), 14328–14335. https://doi.org/10.1109/JSEN.2022.3181003.

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., Chen, X., 2019a. Automatic bridge crack detection using a convolutional neural network. Appl. Sci. 9 (14) https://doi.org/10.3390/app9142867.

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., Chen, X., 2019b. Automatic bridge crack detection using a convolutional neural network. Appl. Sci. 9 (14) https://doi.org/10.3390/app9142867.

Xu, Y., Bao, Y., Zhang, Y., Li, H., 2021. Attribute-based structural damage identification by few-shot meta learning with inter-class knowledge transfer. Struct. Health Monit. 20 (4), 1494–1517. https://doi.org/10.1177/1475921720921135.

Zhou, J., Zheng, Y., Tang, J., Li, J., Yang, Z., 2022. FlipDA: effective and robust data augmentation for few-shot learning. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 8646–8665. https://doi.org/10.18653/v1/2022.acl-long.592.