Domain Adaptation Strategies for Cross-Domain Concrete Crack Classification

by Taoyuan Zhu

Master By Research

Supervised by: Dr Mukesh Prasad, Dr Ali Braytee, Dr Xian Tao

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney

August 2025

Certificate of Original Authorship

I, Taoyuan Zhu, declare that this thesis is submitted in fulfilment of the requirements for the award of Master by Research, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date: 18/08/2025

Abstract

The rapid development of modern infrastructure has brought unprecedented challenges to structural health monitoring, particularly in the domain of concrete crack detection. While deep learning has shown promising results in automating this process, its practical implementation faces significant challenges when models are deployed across different materials, environments, and operational conditions. The traditional approach of collecting and annotating extensive datasets for each new deployment scenario is not only resource-intensive but also increasingly problematic due to data privacy concerns and regulatory requirements.

This thesis presents a comprehensive investigation into solving these challenges through two interconnected research directions. Our first approach challenges the conventional wisdom of relying solely on supervised learning by exploring the potential of self-supervised learning in understanding crack features. By leveraging recent advances in vision foundation models, particularly DINOv2, we demonstrate how models can learn robust and transferable representations without the need for extensive labeled data. This research reveals fascinating insights into how self-supervised models develop a more comprehensive understanding of crack characteristics compared to traditional supervised approaches, offering new perspectives on feature learning in structural defect detection.

Building upon these insights, our second research direction addresses the critical challenge of domain adaptation in real-world deployments. We develop an innovative source-free domain adaptation framework that leverages the power of cross-modal understanding through CLIP (Contrastive Language-Image Pre-training). This approach not only eliminates the need for source domain data during adaptation but also introduces novel mechanisms for efficient knowledge transfer across different material types. By incorporating careful consideration of data efficiency and computational resources, our framework provides a practical solution that balances performance with real-world constraints.

The experimental validation of our approaches spans multiple datasets and deployment scenarios, demonstrating significant improvements in both accuracy and efficiency compared to existing methods. More importantly, our research provides a scalable framework for practical implementations, addressing key challenges such as data privacy, computational efficiency, and deployment flexibility. The success of these approaches in laboratory testing and initial field trials suggests promising potential for widespread adoption in automated structural inspection systems.

This thesis contributes to both the theoretical understanding of domain adaptation in structural health monitoring and the practical implementation of automated crack detection systems. The methodologies developed here not only advance the state-of-the-art in computer vision applications for infrastructure inspection but also provide valuable insights for researchers and practitioners working on similar challenges in other domains of structural health monitoring.

Acknowledgement

Throughout this academic journey, I would first like to express my deepest gratitude to my supervisor, Dr. Mukesh Prasad. His expertise and insightful feedback have been invaluable in refining my research direction and methodologies. I would also like to thank my co-supervisors, Dr. Ali Braytee and Dr. Xian Tao. Their encouragement and rigorous scientific approach have contributed not only to my personal growth but also to my professional development. Thank you for their patient guidance, dedicated mentorship, and trust in my research potential.

I would also like to extend my sincere appreciation to my laboratory colleagues, especially Xing Zi. Their collaborative academic support has enriched my research experience. Their in-depth discussions and technical assistance have helped me overcome numerous challenges throughout my research process.

Finally, I would like to express my profound gratitude to my family. Throughout my academic journey, they have consistently provided me with steadfast support and understanding. Their continuous encouragement and emotional support have given me the strength and motivation to remain resilient during difficult times and stay focused on achieving my academic goals.

List of Publications

Papers related to thesis

1. Zhu, T., Braytee, A., Thiyagarajan, K., Zi, X., Mustapha, S., Tao, X., & Prasad, M. (2025). Autonomous Detection of Concrete Cracks Using Self-Supervised DinoV2. Machine Intelligence Research. (Published)(chapter 3)

2. Zhu, T.,Zi, X., Braytee, A., Tao, X., & Prasad, M. Source-Free Domain Adaptation for Concrete Crack Classification Using CLIP. Science China Information Sciences, Apr. 2025, (Submitted)(chapter 4)

Other papers

1. Zi, X., Shi, Y., Zhu, T., Jin, K., Tao, X., Li, J., Thiyagarajan, K., & Prasad, M. (2024, December). BDC Dataset: A Comprehensive Dataset for Automated Build Damage Classification. In International Conference on Advanced Data Mining and Applications (pp. 91-104). Singapore: Springer Nature Singapore. (Published)

2. Fan, X., Zhu, T., Zi, X., Tao, X., & Prasad, M. (2024). Wild Fire Classification using Learning Robust Visual Features. Scientific reports. (Under review)

3. Zi, X., Zhu, T., Shi, Y., Tao, X., Li, J., & Prasad, M. Assessing Property Damage from Natural Disasters: Self-Supervised and Supervised Learning Models for Remote Sensing Imagery Analysis. Journal of Geography & Natural Disasters. Oct. 2024. (Under review)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Crack inspection is a critical task in civil engineering, as it directly impacts the safety and longevity of infrastructure such as bridges, roads, and buildings [1]. With over 70% of infrastructure worldwide exceeding its designed service life, the importance of effective crack detection has become increasingly crucial [2]. Infrastructure serves as the backbone of modern society, and its degradation due to aging, environmental factors, and increased usage poses significant risks to public safety and economic stability [2]. Timely detection and classification of cracks are essential to prevent catastrophic failures, reduce maintenance costs, and extend the lifespan of critical structures [3].

To address these infrastructure monitoring challenges, automated systems have emerged as a promising solution. With the increasing reliance on automated systems for structural health monitoring, the development of accurate and reliable crack classification systems has become a priority [4]. These systems leverage advancements in machine learning and computer vision to identify and classify cracks in images, including the advantages of deep learning models in feature extraction [5], the capabilities of computer vision algorithms in image preprocessing and enhancement [6], and breakthroughs in end-to-end learning systems for real-time detection [7]. Together, these technological advances provide unprecedented analytical capabilities for crack detection, offering significant advantages over traditional manual inspection methods [8]. While manual inspections are widely used,

they are labor-intensive, time-consuming, and prone to human error, making them unsuitable for large-scale or high-frequency monitoring. In contrast, automated systems promise improved efficiency, scalability, and accuracy, enabling more proactive maintenance strategies.

However, despite their potential, automated crack classification systems face two fundamental challenges in real-world applications [9]: domain shift and data scarcity. Domain shift occurs when models trained on a specific dataset—referred to as the source domain are deployed in a different environment or target domain, where the data distribution differs due to variations in factors such as lighting, surface textures, imaging equipment, and environmental conditions [10]. These shifts can significantly degrade model performance, limiting their reliability and generalization capabilities [2]. Furthermore, the scarcity of labeled data in the target domain exacerbates this problem, manifesting in multiple aspects: first, acquiring large-scale crack image datasets is inherently challenging as it requires professional field inspections and photography; second, even when images are obtained, the annotation process by professional engineers is extremely time-consuming and costly; finally, image data from critical infrastructure may be subject to security and confidentiality requirements, preventing free sharing and utilization [11]. These factors collectively make it impractical to collect and annotate sufficient data for each new deployment scenario, severely constraining the scalable implementation of crack detection systems in real-world engineering applications [12].

To address these challenges, this paper explores the application of domain adaptation techniques, with a particular focus on source-free domain adaptation(SFDA) [13]. As a significant advancement over traditional domain adaptation methods, source-free domain adaptation eliminates the need for direct access to source domain data during the adaptation process [14]. This characteristic makes it a practical solution in scenarios where data privacy concerns, storage limitations, or regulatory restrictions prevent the sharing of source data. Unlike traditional domain adaptation approaches that require simultaneous access to both source and target domain data, source-free domain adaptation accomplishes the adaptation process through innovative knowledge transfer strategies, relying solely on a pre-trained source model and unlabeled target domain data [15]. This approach not only offers significant advantages in protecting data privacy but also substantially reduces data

transmission and storage costs. Through the design of efficient feature alignment strategies and adaptive learning mechanisms, source-free domain adaptation aims to bridge the gap between source and target domains, achieving robust and efficient model adaptation [16]. Recent research has demonstrated that source-free domain adaptation methods incorporating contrastive learning and self-supervised techniques have shown exceptional performance in multiple real-world application scenarios [17].

Furthermore, this research investigates integrating few-shot learning techniques to enhance domain adaptation in scenarios with limited labelled data in the target domain. Few-shot learning leverages advanced algorithms to adapt models using minimal labelled examples, addressing the data scarcity problem while maintaining adaptability across diverse environments. By combining source-free domain adaptation with few-shot learning, this thesis aims to develop innovative methods that overcome the limitations of existing approaches, contributing to the advancement of automated crack classification and structural health monitoring systems.

1.1 Background and Motivation

Infrastructure crack classification involves multiple technical aspects and challenges, which constitute the motivation for this research. In civil infrastructure, crack classification requires consideration of various characteristics, including crack morphology and environmental factors. From a morphological perspective, cracks can be classified based on their patterns (such as longitudinal, transverse, alligator), width (ranging from hairline to severe), and depth, with different types of cracks indicating distinct structural issues requiring specific maintenance approaches [18]. Regarding environmental factors, the detection and classification process is significantly influenced by surface conditions (wet, dry, weathered), lighting variations (natural light, artificial light, shadows), material variations (concrete, asphalt, composite materials), and weather conditions during inspection, all of which pose complex challenges for automated detection systems [19].

In the implementation of automated detection technology, systems face multiple technical barriers. Firstly, there are challenges in image quality and preprocessing, including

resolution limitations under field conditions, requirements for noise and artifact removal, image enhancement necessary for accurate detection, and the standardization of input data across different acquisition devices. Secondly, there are challenges in feature extraction, where systems need to accurately distinguish between cracks and surface textures, handle complex backgrounds, deal with incomplete or interrupted crack patterns, and manage cracks of varying orientations and scales. Furthermore, the challenges brought by data distribution variations cannot be ignored, including specification differences among imaging devices, varying inspection protocols between organizations, regional differences in building materials and practices, as well as seasonal and temporal changes in environmental conditions [20, 21].

In the context of domain adaptation, crack classification faces unique challenges. Data privacy and security concerns are particularly prominent, as infrastructure data often contains sensitive information, and data sharing between organizations is restricted by legal requirements, necessitating secure knowledge transfer methods that comply with infrastructure security protocols [22]. Resource constraints also present significant challenges, manifesting in the limited availability of professional annotators, high costs of data collection and labeling, time constraints in deployment scenarios, and limitations in storage and computational resources. Additionally, model performance requirements pose challenges, as systems need to possess real-time processing capabilities, meet high accuracy requirements for safety-critical applications, demonstrate robustness to environmental changes, and adapt to new scenarios with minimal retraining [23].

These challenges collectively drive the demand for advanced solutions that can operate effectively with limited labeled data, adapt to new environments without requiring source data, maintain high accuracy under varying conditions, and meet practical deployment constraints. The comprehensive consideration of these factors pushes research directions toward integrating source-free domain adaptation and few-shot learning, aiming to address these challenges while maintaining feasibility in practical scenarios.

1.2 Problem Statement

The primary challenge in crack classification lies in addressing the domain shift that arises when models are deployed in diverse real-world scenarios. Domain shift refers to the differences in data distributions between the source domain (used for training) and the target domain (real-world deployment), which can significantly degrade model performance.

Traditional domain adaptation methods often require simultaneous access to both source and target domain data. However, this is not always feasible due to data privacy concerns, storage limitations, or regulatory restrictions. Source-free domain adaptation, which relies solely on a pre-trained source model and unlabeled target domain data, offers a practical solution to this problem. However, it introduces new challenges, such as:

- The lack of supervision from source domain data makes it difficult to align the feature distributions of the source and target domains.
- The risk of overfitting to the target domain, especially when the target domain data is limited or noisy.

Additionally, in scenarios where labelled data in the target domain is scarce or unavailable, few-shot learning techniques provide a complementary approach. These methods aim to adapt models using minimal labelled examples, but their effectiveness in addressing domain shifts and maintaining robustness across varying image qualities remains a key challenge. This research seeks to address these challenges by developing and evaluating novel methods for source-free domain adaptation and few-shot learning in crack classification tasks.

1.3 Research Objectives

The primary objective of this research is to address the challenges posed by domain shifts in crack detection models and propose effective domain adaptation methods to enhance their generalization capabilities in real-world scenarios. To achieve this, the study focuses on two key objectives: source-free domain adaptation and few-shot learning.

Source-Free Domain Adaptation

This objective explores using pre-trained large models or self-supervised models as source models, which are trained on diverse, large-scale datasets but do not provide access to their original source domain data. The focus is on designing strategies to utilize the knowledge embedded in these pre-trained models for domain transfer, optimizing the domain adaptation capability of the target model through progressively enhanced supervision signals. These signals range from weak supervision methods, such as pseudo-labeling, to strong supervision techniques, like fine-tuning with limited labeled data. This approach aims to adapt crack detection models across different surface types, such as transitioning from concrete to asphalt, without requiring extensive datasets for each new domain, thereby addressing practical constraints such as data scarcity and privacy concerns.

Generative Model-based Domain Adaptation

This objective explores the application of generative models in domain adaptation, particularly in scenarios with limited or no labeled data in the target domain. The research leverages advanced generative model techniques such as Diffusion Models and Generative Adversarial Networks (GANs) to enhance the domain adaptation capabilities of crack detection models. This approach can bridge domain gaps either by generating synthetic data that aligns with target domain characteristics or by employing style transfer techniques to convert source domain images into target domain styles. This methodology is particularly crucial for handling domain adaptation across significant image quality variations, such as converting high-quality images to different lighting conditions, weather conditions, or simulating various levels of image noise and degradation. Through these efforts, the research aims to provide a comprehensive evaluation of the proposed methods, explore the potential and limitations of generative models in domain adaptation, identify the trade-offs between source-free domain adaptation and generative model-based approaches, and offer insights into their practical applications, contributing to future advancements in automated structural health monitoring.

1.4 Research Goals

- How to enhance the domain adaptation capability of the target model when source domain data is inaccessible?
- How to develop effective domain adaptation methods when there is only a small amount or no labelled data in the target domain?

1.5 Scope and Limitations

The scope of this research is focused on the application of domain adaptation techniques, particularly source-free domain adaptation and few-shot learning, in crack classification tasks. The study is designed to address the following scenarios:

- Cases where source domain data cannot be directly accessed due to privacy, storage, or regulatory constraints.
- Scenarios with limited or no labelled data in the target domain necessitating the use of generative model techniques.
- Adaptation across diverse real-world conditions, such as variations in lighting, surface textures, and imaging quality.

This research is limited to datasets representing cracks in infrastructure, such as bridges, roads, and buildings [18]. Specifically, the study focuses on concrete and asphalt surfaces under varying environmental conditions [19], with image resolutions ranging from standard (640×480) to high-definition (1920×1080) captures. While the findings aim to provide general insights into domain adaptation, the specific results may be influenced by the datasets' characteristics, such as the types of cracks and imaging conditions. Additionally, the proposed methods are evaluated within the context of crack classification and may require further validation for other types of structural health monitoring tasks.

Chapter 2

Literature Review

2.1 Overview of Crack Detection Methods

Automated crack classification systems have been a significant focus in civil engineering research, evolving substantially over the past decade. The field initially relied on traditional image processing techniques, with early approaches primarily focusing on edge detection and texture analysis methods. In 2012, Koch et al. conducted a comprehensive review of computer vision-based defect detection methods, highlighting the limitations of these conventional approaches in handling variable environmental conditions and complex surface textures [18]. These challenges were further elaborated by Jahanshahi et al., who specifically examined the difficulties in crack detection across diverse concrete structures using traditional computer vision methods [24]. Their work particularly emphasized the impact of lighting variations, surface irregularities, and environmental factors on detection accuracy.

Traditionally, crack detection relied heavily on visual inspection, which depended on the inspector's patience, skill, and experience [25]. However, this method proved to be insufficient when the I-35W highway bridge in the United States collapsed due to structural issues and inadequate inspection in 2007, causing a severe accident [26]. To prevent such disasters, continuous inspection and assessment of cracks are necessary, which could be inefficient if done manually. With the advent of machine learning, which can mimic human

behavior and avoid human interference, it has found applications in many fields [27]. In the context of Structural Health Monitoring (SHM), machine learning has increasingly been utilized [28]. Li et al. [29] used Support Vector Classification (SVC) for crack image classification. Kaseko et al. [30] used Neural Networks for the same purpose, while Shi et al. [31] proposed the CrackForest, a Random Structure Forest for crack detection and classification. Nevertheless, traditional machine learning methods often do not consider deep features and can be influenced by many disturbance factors due to the complex environment where cracks often occur, thus affecting the classification results.

The emergence of deep learning marked a pivotal shift in crack classification methodology, revolutionizing both accuracy and robustness in detection systems. Zhang et al. pioneered the application of Convolutional Neural Networks (CNNs) for crack detection in 2016 [32], demonstrating superior performance compared to traditional methods through their novel network architecture. This breakthrough was further advanced by Cha et al. in 2017, who developed a more sophisticated CNN architecture specifically for structural crack identification, achieving 98% accuracy in controlled environments and introducing innovative data augmentation strategies [33]. In the same year, Kim and Cho [34] proposed an enhanced CNN model that could effectively handle multi-scale crack features through a hierarchical feature extraction approach, further improving detection accuracy in real-world scenarios and addressing the challenges of varying crack sizes and orientations.

The challenge of domain shift in crack classification systems was first systematically addressed by Li et al., who identified significant performance degradation when models were deployed across different infrastructure materials and environmental conditions [35]. Their comprehensive analysis quantified how variations in image quality, lighting conditions, and surface textures could lead to up to 40% reduction in model reliability. This critical finding was later reinforced and expanded by Yang et al., who conducted extensive experiments across multiple infrastructure types including concrete bridges, asphalt pavements, and steel structures, proposing adaptive learning strategies to mitigate domain shift effects [19]. Their work specifically demonstrated how domain-specific feature extraction and adaptation techniques could help maintain model performance across diverse deployment scenarios.

2.2 Deep Learning in Crack Detection

Deep learning models, particularly Transformers, have achieved tremendous success in crack detection in recent years. Jang et al. [36] proposed a method to identify multiple damages by integrating attention mechanisms with neural networks, achieving 98% accuracy. Zhu et al. [37] built a context contrast network to capture multi-scale features and added an attention module to make the model more attentive to smaller foreground proportions, thereby improving sensitivity to cracks. Wang et al. [38] combined CNNs, Transformers, and MLP heads to automatically classify cracks. Ye et al. [39] proposed a novel CSWin transformer-skip link to expand the image feature range of CSW-S, achieving 96.2

With the further exploration of deep learning methods, self-supervised learning has emerged as a promising approach that generates tasks automatically, enabling models to learn useful feature representations without human-annotated data. Grill et al. [40] proposed bootstrap your own latent (BYOL), which was the first to introduce self-distillation through two networks and a predictor that maps the output of one network to another. Chen et al. [41] proposed SimCLR, which learns visual representations by encouraging similarity between two augmented views of an image. He et al. [42] proposed Moco, which uses a queue and a moving average encoder to build a dynamic dictionary for unsupervised contrastive learning. Moco features three core operations: queue dictionary, momentum update, and Shuffle BN. The queue dictionary avoids degeneration by satisfying both alignment and uniformity; momentum update ensures that key encoder parameters are the moving average of query encoder parameters, preventing parameter mutations in the key branch; Shuffle BN calculates the BN layer separately in multi-GPU training. MocoV2 incorporates tricks from SimCLR, achieving performance surpassing the then state-of-the-art. Lin et al. [43] used self-supervised MAE and ViT to design a new method for crack detection. In Eq. 2.1, the parameter update rule for the key encoder θ_k is defined. Here, θ_k represents the parameters of the key encoder, and θ_q represents the parameters of the query encoder. The momentum coefficient $m \in [0, 1)$ controls the update rate, ensuring that the key encoder parameters θ_k are a moving average of the query encoder parameters θ_q.

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{2.1}$$

2.3 Domain Adaptation: Principles and Evolution

2.3.1 Traditional Domain Adaptation

The foundation of modern domain adaptation was established with Deep Domain Confusion (DDC) by [44], introducing deep feature adaptation through a domain confusion mechanism. Maximum Mean Discrepancy (MMD) emerged as a crucial measure for domain distribution differences [45]. Long et al. advanced this through Deep Adaptation Networks (DAN) [46], which overcame the limitations of single adaptation layers by introducing multi-kernel maximum mean discrepancy (MK-MMD) metrics.

A significant breakthrough came with the introduction of Domain-Adversarial Neural Networks (DANN) by Ganin et al. [47], which pioneered the use of adversarial training for domain adaptation. This approach introduced a gradient reversal layer that enabled the joint optimization of feature extraction and domain confusion. Building upon this foundation, Sun and Saenko proposed Deep Correlation Alignment (CORAL) [48], which aligned the second-order statistics of source and target distributions, offering a computationally efficient alternative to MMD-based methods.

The field further evolved with Joint Adaptation Networks (JAN) by Long et al. [49], which extended the adaptation to the classifier layers through joint maximum mean discrepancy. This was complemented by Conditional Domain Adversarial Networks (CDAN) [50], which introduced conditional adversarial domain adaptation by leveraging the discriminative information from the classifier predictions. Meanwhile, Saito et al. proposed Maximum Classifier Discrepancy (MCD) [51], introducing a novel perspective by utilizing the disagreement between two classifiers to detect target samples that are far from the source domain.

2.3.2 Adversarial Domain Adaptation

A significant breakthrough came with Domain-Adversarial Neural Networks (DANN) by [47], which implemented an end-to-end adversarial training framework through gradient reversal layers. This approach inspired a series of influential adversarial adaptation methods. Deep CORAL by Sun et al. [48] performed feature alignment based on second-order statistics, while Adversarial Discriminative Domain Adaptation (ADDA) [52] further developed this approach with separate feature extractors for source and target domains.

The field of adversarial domain adaptation continued to evolve with significant contributions. Conditional Domain Adversarial Networks (CDAN) [50] enhanced the discriminative ability of adversarial adaptation by conditioning the domain discriminator on classifier predictions. Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [53] combined pixel-level and feature-level adaptation through cycle consistency losses. Meanwhile, Maximum Classifier Discrepancy (MCD) [51] introduced a novel perspective by leveraging the disagreement between task-specific classifiers as a domain alignment signal.

More recent advances include Virtual Adversarial Domain Adaptation (VADA) [54] which incorporated virtual adversarial training to improve robustness, and Symmetric and Asymmetric Multi-Domain Adversarial Learning (SAMDA) [55] which proposed a unified framework for handling both symmetric and asymmetric domain adaptation scenarios. These methods collectively demonstrate the power and versatility of adversarial approaches in domain adaptation.

2.3.3 Advanced Alignment Strategies

The field progressed with more sophisticated alignment approaches. [56] proposed joint distribution alignment using Joint MMD, while [57] developed the Maximum Classifier Discrepancy (MCD) method. Long et al. [58] introduced the Conditional Domain Adversarial Network (CDAN), incorporating multiple semantic information from features and predictions. Further advances included the work of [59] and [60], who developed intra-domain and inter-domain confrontation strategies.

Recent developments have introduced more nuanced alignment techniques. Wei et al. [61] proposed MetaAlign, which coordinates domain alignment and classification through meta-learning principles. Zhao et al. [62] introduced a Domain Adaptive Region Proposal Network (DA-RPN) that performs both global and local feature alignment. The Joint Adaptation Networks (JAN) framework by Long er al. [49] extended traditional MMD to align joint distributions of multiple domain-specific layers across domains. Kang et al. [63] proposed Contrastive Adaptation Network (CAN) that explicitly models the intra-class domain discrepancy through a contrastive domain discrepancy measure.

Advanced theoretical frameworks emerged with Zhao et al. [64]'s work on optimal transport-based alignment and Peng et al. [65]'s Moment Matching for Multi-Source Domain Adaptation (M3SDA). These methods provided stronger theoretical guarantees for domain adaptation success. Additionally, Chen et al. [66] developed Higher-Order Moment Matching (HOMM) to capture fine-grained statistical dependencies between domains.

2.3.4 Source-Free Domain Adaptation

Recent developments in source-free domain adaptation have addressed privacy concerns by enabling adaptation without source domain data. [67] established theoretical error bounds for source-free adaptation, while Ding et al. [68] proposed an effective two-stage adaptation approach. Significant advances include Yang et al.'s [69] unified framework, Tang et al. [70]'s CLIP-based approach, and Chopra et al. [71]'s application of diffusion models.

The field has seen remarkable progress with the integration of foundation models. Li et al. [72] introduced a novel framework leveraging frozen multimodal foundation models for SFDA, demonstrating superior performance across various visual tasks. Chopra et al. [73] proposed DM-SFDA, a pioneering approach that utilizes diffusion models' generalizability for source-free adaptation through guided source data generation. Wang et al. [74] developed a neighborhood-informed diffusion model that enables target-to-source generation, addressing the domain gap more effectively.

2.4 Domain Adaptation in Crack Detection

The application of domain adaptation in crack detection has become increasingly important due to the significant domain shifts encountered in real-world deployments. Zhang et al. [35] pioneered the use of adversarial domain adaptation for crack detection, demonstrating how domain-invariant features could be learned across different infrastructure materials. Their work showed a 15-20% improvement in cross-domain performance compared to non-adapted models. Liu et al. [75] further advanced this approach by introducing a multi-scale feature alignment strategy specifically designed for crack detection, addressing the challenge of varying crack sizes and orientations across domains.

In the context of concrete crack detection, Wang et al. [76] developed a novel adaptation framework that specifically addresses the challenges of varying surface textures and environmental conditions. Their method incorporated both global and local feature alignment strategies, achieving robust performance across different concrete types and aging conditions. The effectiveness of their approach was demonstrated through extensive experiments on the SDNET2018 dataset, showing particular strength in handling domain shifts between laboratory and field conditions.

Recent work by Chen et al. [77] has focused on source-free domain adaptation for crack detection, addressing the practical constraints of data privacy and storage limitations. Their approach enables model adaptation without requiring access to the original training data, while maintaining competitive performance. This is particularly relevant for infrastructure monitoring systems where data sharing might be restricted due to security concerns.Performance evaluation in cross-domain crack detection presents unique challenges. Standard metrics such as accuracy, precision, and recall often fail to capture the nuanced differences in crack detection performance across domains. To address this, Li et al. [78] proposed a comprehensive evaluation framework that considers three main aspects. First, detection accuracy metrics include overall classification accuracy, class-wise precision and recall, and F1-score for crack detection, providing a thorough assessment of the model's detection capabilities. Second, domain alignment quality is evaluated through Maximum Mean Discrepancy (MMD) between domains, A-distance as a measure of domain divergence, and feature visualization through t-SNE, offering insights into the effectiveness of

domain adaptation. Third, practical considerations encompass computational efficiency, memory requirements, and inference time, ensuring the model's feasibility in real-world applications.

2.5 Current Challenges and Future Directions

The field of crack detection using domain adaptation faces several significant challenges that present opportunities for future research:

2.5.1 Technical Challenges

One of the primary technical challenges lies in feature representation. Extracting domain-invariant features while preserving crack-specific characteristics remains difficult, particularly when handling multi-scale crack features across domains. There is an urgent need for more robust feature extraction methods that can effectively handle severe domain shifts while maintaining detection accuracy.

Model architecture presents another significant challenge. Researchers must carefully balance model complexity with computational efficiency, particularly in resource-constrained environments. The integration of attention mechanisms for better feature selection has shown promise, but optimizing these architectures for specific crack detection tasks requires further investigation.

Training stability continues to be a crucial concern in domain adaptation for crack detection. Addressing the instability issues inherent in adversarial training remains challenging. There is a need for more robust optimization strategies that can effectively manage the trade-off between adaptation performance and task-specific accuracy while maintaining consistent training behavior.

2.5.2 Practical Application Issues

Real-world deployment presents numerous challenges in crack detection systems. Environmental conditions can vary dramatically, affecting system performance through changes in lighting conditions and surface textures. Managing computational resources in field deployments requires careful consideration of hardware limitations and processing requirements.

Data collection and annotation represent significant practical hurdles. The high cost of collecting and annotating crack data, combined with the difficulty in obtaining representative samples across different domains, creates substantial barriers to system development. More efficient data collection protocols are needed to address these challenges while maintaining data quality and representation.

System integration challenges arise when implementing crack detection systems in existing infrastructure. Real-time processing requirements must be balanced with system accuracy, while ensuring compatibility with different sensor types and data formats. These integration challenges require careful consideration of both technical and practical constraints.

Chapter 3

Autonomous Detection of Concrete Cracks Using Self-Supervised DinoV2

3.1 Introduction

Crack detection plays a crucial role in extending the lifespan of facilities and ensuring public safety. With the development of artificial intelligence, neural networks and deep learning technologies have been introduced into the field of crack detection, with many supervised learning methods achieving excellent performance in this task. Traditionally, manual inspection has been the primary method for detecting structural cracks[79, 80]. However, this approach is not only costly but also time-consuming, labor-intensive, and prone to human error. The inefficiency of manual inspections can have severe consequences, as potentially exemplified by the collapse of the I-35W highway bridge, where inadequate inspection might have been a contributing factor[81]. Moreover, manual inspections can be dangerous in certain environments, putting inspectors at risk[82]. These limitations of manual inspection not only pose financial burdens but also raise significant safety concerns, highlighting the urgent need for more efficient and reliable methods.

To advance capabilities in this field, Structural Health Monitoring (SHM) has become a key research focus and even a subject of international project competitions[83]. The integration of AI-powered technologies promises to overcome the inefficiencies of manual inspections by offering faster, more accurate, and safer alternatives. However, the application of these technologies has also brought a series of new challenges, particularly in terms of data requirements and model generalization capabilities. As deep learning technologies emerge and continue to develop, researchers have gradually adopted these techniques to achieve more efficient and accurate crack detection. However, for supervised models, this has brought about a series of significant challenges. First, the issues of data dependency and generalization capability are particularly prominent. Most existing crack detection techniques heavily rely on labelled data, which has become a major limitation. Cracks can appear on various materials and may even be hidden in different environments, such as underwater stains or within asphalt[84]. This diversity of data requires models to automatically adapt to the specific characteristics of each dataset. However, existing models typically require manual parameter adjustment to optimize performance. This approach of individually adjusting models based on different datasets may weaken their generalization ability and robustness when facing new, unknown datasets. Specifically, the diversity of cracks requires training data to have broad representatives, while most models need to manually optimize parameters for specific datasets, limiting their universality. As a result, individually adjusted models may struggle to adapt to new, unseen datasets, affecting their practical application effectiveness [85]. Secondly, the data imbalance problem is also a severe challenge. In actual crack detection tasks, non-crack areas are usually far more common than crack areas, leading to a serious data imbalance problem. For supervised learning models, this imbalance may cause the model to be biased towards predicting the more common class, i.e., non-crack areas, thereby reducing the detection sensitivity to actual cracks. This not only may lead to bias in class prediction but may also reduce the ability to detect actual cracks, increasing the risk of missed detections. Furthermore, this imbalance may cause traditional evaluation metrics to inaccurately reflect the model's true performance on the minority class (cracks).

To address the aforementioned challenges, in this study, we not only analyzed the performance of various supervised classification models in crack classification, but also evaluated

the effectiveness of integrating a linear classification head with the self-supervised DinoV2 model in crack classification tasks. The main contributions of our research are as follows:

1. We adopt the self-supervised DinoV2 framework for feature extraction and combine it with a linear classification head for crack classification. This approach applies advanced self-supervised learning techniques to the field of structural integrity assessment, demonstrating its practicality in enhancing crack detection methods. Our main contribution lies in proving the strong capability of DinoV2 in crack feature extraction.

2. Following initial explorations, our research extended to a series of comprehensive experiments aimed at evaluating the effectiveness and generalization ability of supervised and self-supervised learning methods in classifying crack images across different datasets. We conducted cross-validation on four distinct datasets to showcase the generalization capability of DinoV2.

3. We provided a visual analysis of attention mechanisms, comparing the differences in attention between supervised models and DinoV2, demonstrating how DinoV2 focuses on important features of cracks.

3.2 Methodology

3.2.1 DinoV2 Architecture and Principles

The Dino framework utilizes the Vision Transformer (ViT) as its structural foundation [86]. This end-to-end architecture consists of a backbone, a multi-layer Transformer encoder, a multi-layer Transformer decoder, and several prediction heads. Images are partitioned into patches before being encoded by the network. During this process, techniques such as Masked Autoencoders (MAE) are applied to obscure and eliminate specific image patches, which are later reconstructed into a full image by the decoder. This approach has been effective for self-supervised learning. Building on Dino, DinoV2 integrates principles from the natural language processing field, synthesizing methods from DINO, iBOT [87], and

SwAV with the addition of regularization terms and a phase of high-resolution training. This facilitates the generation of versatile visual features for a wide array of computer vision tasks. Based on the ViT architecture, the model boasts one billion tunable parameters. A smaller, multi-purpose model is distilled from the expansive ViT-g model using unsupervised techniques. The ViT-g acts as a teacher model while the KoLeo regularizer ensures the student model's feature representation aligns with that of the teacher model [88], particularly within masked areas. The integration of a specifically curated dataset is shown to enhance DinoV2's performance significantly.

Moreover, DinoV2, building on Dino's teacher-student network strategy, incorporates the methodology from the iBOT approach. It employs self-distillation as an objective for generating self-supervised labels and carries out the Masked Language Model (MLM) through this self-distillation process. As depicted in Eq. 3.1 [87], the masked version is reconstructed after passing through the student network. The student network θ takes the masked version $\hat{u}$ as input, producing the corresponding patch tokens $\hat{u}_s^{\text{patch}} = P_\theta^{\text{patch}}(\hat{u})$. Conversely, the teacher network θ' processes the unmasked version u, generating the patch tokens $\hat{u}_t^{\text{patch}} = P_{\theta'}^{\text{patch}}(u)$.

Here, $\hat{u}$ represents the masked input image, and u represents the unmasked original input image. θ and θ' denote the student and teacher networks, respectively. P_θ^{patch} and $P_{\theta'}^{\text{patch}}$ refer to the patch token generation functions of the student and teacher networks. $\hat{u}_s^{\text{patch}}$ and $\hat{u}_t^{\text{patch}}$ represent the patch tokens generated by the student and teacher networks, respectively.

$$L_{\text{MIM}} = -\sum_{i=1}^{N} m_i \cdot P_{\theta'}^{\text{patch}}(u_i)^T \log P_\theta^{\text{patch}}(\hat{u}_i) \tag{3.1}$$

These features are guided through the student and teacher networks to maintain coherence in the masked regions. In addition, to cater to the increased resolution requirements for downstream tasks, the resolution is escalated to 518x518 towards the end of the training phase. The culmination of these processes inputs into a linear classification head or a KNN classification head, yielding the classification outcomes.

3.2.2 Crack detection framework based DinoV2

In this study, we utilized the DinoV2 framework: data input, data pre-processing, self-supervised DinoV2, and classification. As shown in Fig. 3.1, the first component is the input stage, where images are resized to a resolution of 224 x 224. In the data preprocessing stage, images are divided into an N x N grid (typically, N is 16 or 8), with each image patch subsequently converted into an embedding vector and combined with learnable CLS tokens and positional embeddings. These processed data are then input into the DinoV2 model, which is based on the Vision Transformer (ViT) architecture [86] and includes a backbone network (such as ResNet or ViT), multi-layer Transformer encoders, and decoders. During this process, DinoV2 applies various self-supervised learning techniques, such as Masked Autoencoders (MAE) and methods from iBOT [87]. The model adopts a teacher-student network strategy, with ViT-g serving as the teacher model, ensuring consistency between the student model's feature representation and the teacher model's through the KoLeo regularizer. The feature extraction stage obtains multi-scale features from the images, with CLS tokens summarizing information from the entire sequence [88]. Finally, these features are classified through linear classification heads. The linear classification head is then connected, which consists of two fully connected layers and a ReLU activation function. First, the input layer maps the 384-dimensional feature vector to a 256-dimensional space through a fully connected operation. Next, the ReLU activation function is applied to the outputs of these 256 neurons, introducing non-linearity and allowing the network to learn more complex feature patterns. The function of ReLU is to set all negative values to zero while keeping positive values unchanged, which helps to address the vanishing gradient problem in deep neural networks. Finally, the second fully connected layer further maps the 256-dimensional activation values to 2 output neurons, corresponding to the final prediction for a binary classification problem. Notably, in the later stages of training, the model increases the image resolution to 518x518 to meet the requirements of downstream tasks.

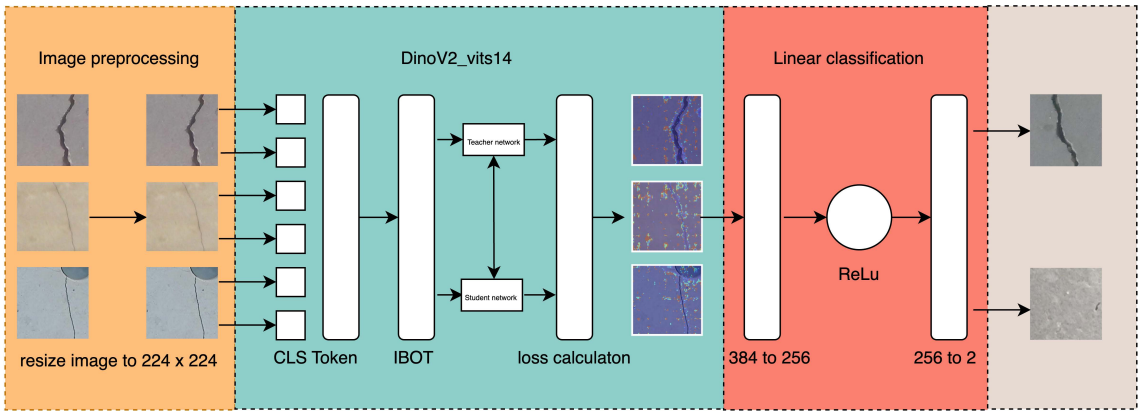


FIGURE 3.1: An overview of the proposed framework

3.3 Experiments

3.3.1 Dataset

Four publicly available concrete crack image datasets were utilized: Concrete Cracking Images for Classification (CCiC) provided by Özgenel et al. [89], Xu dataset provided by Xu et al. [90], SDNET2018 dataset provided by Maguire et al. [91] and Historical_Building_Crack_2019 (HBC2019) provided by Elhariri et al. [92]. The datasets are summarized in Table 3.1.

TABLE 3.1: Details of each dataset

Dataset	Classes name	Total	Total
CCiC	Negative	20,000	
	Positive	20,000	40,000
SDNET2018	Negative	47,608	
	Positive	8,484	56,092
Xu	Negative	2,014	
	Positive	4,055	6,069
HBC2019	Negative	3,139	
	Positive	757	3,896

3.3.1.1 CCiC

CCiC was collected from buildings on the METU campus [89], and it was derived from 458 high-resolution images (4032x3024 pixels) using the methodology proposed by Zhang

et al. [32]. The dataset consists of two categories, positive cracks and negative cracks, with 20,000 images in each category. These images are 227x227-pixel RGB channel images without any preprocessing, such as random rotations.

FIGURE 3.2: A sample of CCiC dataset [89]

3.3.1.2 Xu

Xu dataset [90] comprises 2,014 non-crack images (labeled as 'Negative') and 4,055 crack images (labeled as 'Positive'), all with a resolution of 224x224 pixels.

FIGURE 3.3: A sample of Xu dataset [90]

3.3.1.3 HBC2019

The HBC2019 dataset [92] represents a historical architectural surface crack collection obtained from Masjed using a Canon camera. This dataset captures many of the challenges

found in real-world settings, including cracks, blurriness, deep textures, and wood grain patterns. It comprises of 3,139 non-crack images (labeled as 'Negative') and 757 images displaying cracks (labeled as 'Positive'). All images have been resized to a resolution of 256x256 pixels.

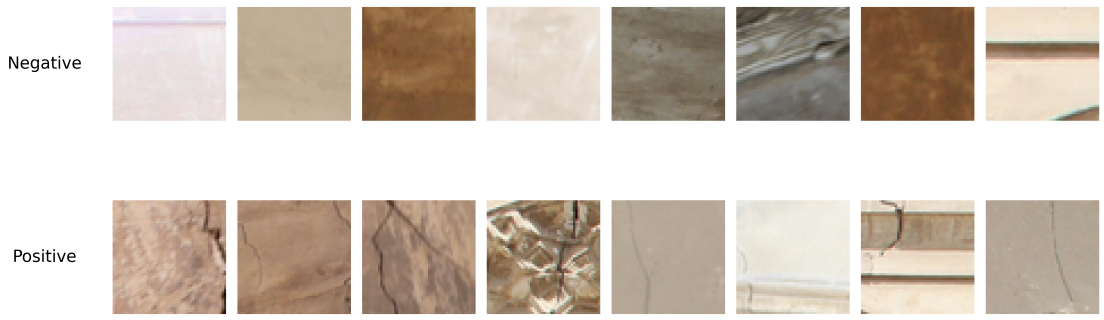


FIGURE 3.4: A sample of HBC2019 dataset [92]

3.3.1.4 SDNET2018

SDNET2018 dataset [91] comprises over 56,000 images with dimensions of 256x256 pixels, featuring cracks. The dataset encompasses three distinct material-based binary classification sets: concrete bridge decks, walls, and sidewalks. Images with cracks are labeled as 'Positive', while those without are labeled as 'Negative'. Each image conforms to the dimensions of 256x256 pixels.

3.3.2 Compared methods

We trained five supervised models: ResNet50[93], ResNet101[93], VGG16[94], MobileNet[95], DenseNet121[96] on four datasets. ResNet50 and ResNet101 introduced the concept of 'residual blocks' or 'skip connections' that allow for the training of deeper neural networks without encountering the vanishing gradient problem. VGG16 consistently employs a small 3x3 convolutional kernel and comprises a total of 16 layers, which include both convolutional and fully connected layers. MobileNet V2 is a deep neural network designed specifically for mobile and resource-constrained devices. It employs depthwise separable

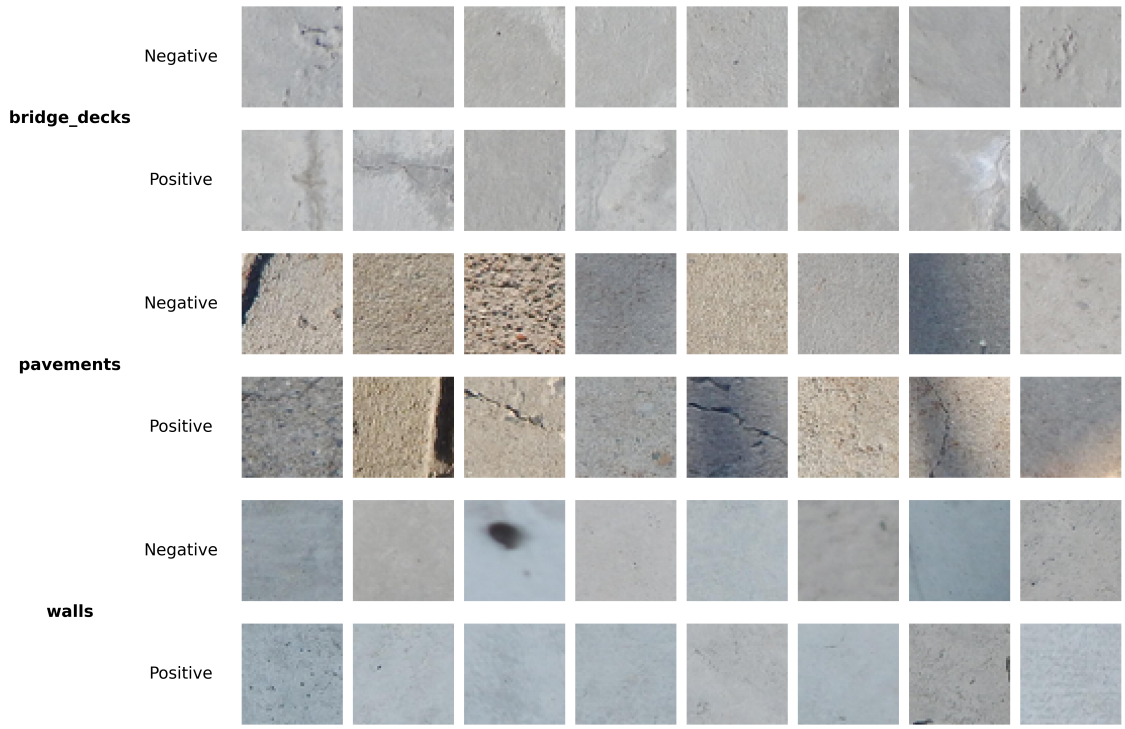


FIGURE 3.5: A sample of SDNET2018 dataset [91]

convolutions to minimize computation and model size. DenseNet121 is characterized by each of its layers being directly connected to all preceding layers. This dense connectivity enables DenseNet to outperform other networks in parameter efficiency. Additionally, it excels in feature reuse, which mitigates the vanishing gradient problem.

3.3.3 Experiment settings

During the data preprocessing phase, all images were resized to a resolution of 224x224 pixels to accommodate our models. Each dataset was then partitioned, with 60% designated for training, 20% for validation, and 20% for testing. Table 3.2 provides a detailed breakdown of the number of images in each dataset.

We utilized a pre-trained model built on the DinoV2 architecture, named "dinov2 vits14", to which we added a linear classification head for our classification task. For the classification head, we configured a fully connected layer from 384 to 256, followed by a ReLU activation function layer, and finally, a fully connected layer from 256 to 2 to output the

TABLE 3.2: Data set details

Dataset	Classes name	Train (60%)	Valid (20%)	Test (20%)
CCiC	Negative	12,000	4,000	4,000
	Positive	12,000	4,000	4,000
SDNET2018	Negative	28,564	9,521	9,523
	Positive	5,090	1,696	1,698
Xu	Negative	1,208	402	404
	Positive	2,433	811	811
HBC2019	Negative	1,883	627	629
	Positive	454	151	152

final classification results. Throughout the training phase, we ran the model for only 5 epochs using an initial learning rate of 0.000001. We selected CrossEntropyLoss as our loss function and employed the Adam optimizer for parameter tuning.

To ensure a fair comparison, we selected five well-known supervised models for training: ResNet50, ResNet101, VGG16, MobileNet_v2, and DenseNet121. These models were all trained under the same standardized conditions. Instead of using pre-trained weights, we trained the models from scratch. For each model, we adjusted the final fully connected layer to match the number of classes in our dataset, taking into account the specific architecture of each model. We employed cross-entropy loss as our loss function, which is a widely accepted metric for measuring the difference between predicted and true labels. The optimization process utilized the Stochastic Gradient Descent (SGD) optimizer with a momentum value of 0.9 to enhance convergence. We set the initial learning rate to 0.0001 and trained for 30 epochs, which was chosen based on a search for the optimal number of epochs. Furthermore, we introduced a cosine annealing learning rate scheduler to optimize the learning process and stabilize model training. This strategy ensures gradual adjustment of the learning rate throughout the training process, contributing to better model convergence. After each training epoch, we evaluated the model's performance on the validation set to monitor training progress and prevent overfitting. Upon completion of training, we saved the final weights of each model for subsequent use or further evaluation. The rationale behind these parameter choices was to balance training convergence and the model's subsequent generalization ability, aiming to achieve a favourable balance between excellent training results and reliable performance evaluation. For our training setup, we

utilized the capabilities of a machine equipped with an Nvidia GeForce RTX 4090 graphics card. This was complemented by an Intel i9-13900KF processor and 64GB of DDR5 RAM clocked at 5200MHz. All training processes were executed on the Windows 11 operating system, using the PyTorch 2.0.0 framework within a CUDA 11.8 environment.

3.4 Results

In this study, we adopted a comprehensive evaluation approach, giving equal importance to four key metrics: precision, recall, F1-score, and accuracy. We thoroughly assess model performance, particularly when dealing with diverse types of datasets. We assessed the models by testing various CNN architectures and DinoV2 across the four datasets. Furthermore, we assess the generalization of the compared models by training on one dataset and testing them on others.

3.4.1 Comparison of Methods on same Training and Testing Datasets

Table 3.3 presents an evaluation of seven models: ResNet50, ResNet101, VGG16, MobileNet V2, DenseNet121, Moco V2 and DinoV2 tested on four different datasets: CCiC, SDNET2018, Xu, and HBC2019. The performance of these models is systematically assessed using five evaluation metrics: precision, recall, F1-score, specificity, and accuracy.

The results on the CCiC dataset indicate that MobileNet v2 performed the best among all models, achieving the highest precision, recall, F1-score, and accuracy (all 0.9965). Following closely was DenseNet121, which also demonstrated excellent performance with all metrics at 0.9962. ResNet50, ResNet101, and VGG16 also scored high but slightly lagged behind MobileNet v2 and DenseNet121 on these four critical metrics. DinoV2 also performed well on the CCiC dataset, with a precision of 0.9917, recall of 0.9885, F1-score of 0.9901, and accuracy of 0.9901. MoCo v2's performance remains accurately described. These outcomes highlight the efficiency and accuracy of MobileNet v2, DenseNet121, and DinoV2 in processing the CCiC dataset.

TABLE 3.3: Test results of different models in each dataset. Values in bold indicate the model's best performance.

Dataset	Model	Precision	Recall	F1-score	Accuracy
CCiC	ResNet50	0.9963	0.9963	0.9963	0.9963
	ResNet101	0.9955	0.9955	0.9955	0.9955
	VGG16	0.9898	0.9898	0.9898	0.9898
	MobileNet_v2	**0.9965**	**0.9965**	**0.9965**	**0.9965**
	DenseNet121	0.9962	0.9962	0.9962	0.9962
	MoCo_v2	0.9949	0.9790	0.9891	0.9892
	DinoV2	0.9917	0.9885	0.9901	0.9901
Xu	ResNet50	0.9770	0.9802	0.9787	0.9810
	ResNet101	0.9699	0.9728	0.9713	0.9744
	VGG16	0.8972	0.8335	0.8551	0.8806
	MobileNet_v2	0.3337	0.5000	0.4002	0.6674
	DenseNet121	0.9683	0.9666	0.9674	0.9711
	MoCo_v2	0.6000	0.0297	0.0566	0.5000
	DinoV2	**0.9771**	**1.0000**	**0.9884**	**0.9844**
HBC2019	ResNet50	0.7850	0.6799	0.7110	0.8501
	ResNet101	0.7805	0.6072	0.6304	0.8335
	VGG16	0.4026	0.5000	0.4460	0.8053
	MobileNet_v2	0.4026	0.5000	0.4460	0.8053
	DenseNet121	0.8915	0.8550	0.8715	0.9231
	MoCo_v2	0.7674	0.4400	0.5593	0.8659
	DinoV2	**0.8876**	**0.9868**	**0.9346**	**0.9731**
SDNET2018	ResNet50	0.6789	0.5087	0.4793	0.8487
	ResNet101	0.7065	0.5103	0.4822	0.8494
	VGG16	0.4243	0.5000	0.4590	0.8486
	MobileNet_v2	0.7137	0.5366	0.5332	0.8517
	DenseNet121	0.7838	0.6072	0.6376	0.8681
	MoCo_v2	0.2173	0.0059	0.0115	0.8464
	DinoV2	**0.8924**	**0.6985**	**0.7836**	**0.9416**

These outcomes highlight the efficiency and accuracy of DenseNet121, DinoV2, and MoCo_v2 in processing the CCiC dataset. The superior performance of DenseNet121 can be attributed to its dense connectivity pattern, which allows for more efficient feature reuse and improved information flow throughout the network. This architecture is particularly effective for the CCiC dataset. The test results on the Xu dataset indicate that DinoV2 outperformed all other models, achieving a perfect recall rate of 1.0000, the highest F1-score of 0.9884, and an accuracy of 0.9844. This exceptional performance can be attributed to DinoV2's self-supervised learning approach, which allows it to learn more robust and

generalizable features. The perfect recall rate is particularly noteworthy, as it indicates that DinoV2 was able to identify all crack instances in the dataset, a crucial factor in safety-critical applications where missing a crack could have severe consequences. DenseNet121's performance precision is 0.9683, recall 0.9666, F1-score 0.9674, and accuracy 0.9711. The slight drop in performance compared to the CCiC dataset suggests that the Xu dataset may have some unique characteristics or challenges that DenseNet121 found more difficult to handle. Interestingly, VGG16 achieved precision 0.8972, recall 0.8335, F1-score 0.8551, and accuracy 0.8806, while MobileNet v2 performed poorly with precision 0.3337, recall 0.5, F1-score 0.4002, and accuracy 0.6674. The low performance of MobileNet v2 is due to the class imbalance in the dataset, where the method struggles to predict the positive samples. Such behavior could be problematic in real-world applications, potentially leading to unnecessary inspections or maintenance. ResNet50 and ResNet101 performed solidly with F1-scores of 0.9787 and 0.97134, and accuracy of 0.98103 and 0.9744, respectively. While slightly behind DinoV2 and DenseNet121, they still displayed good performance on these key metrics. The residual learning framework of ResNet50 and ResNet101 performs well across these datasets but is not as well-suited to the Xu dataset as the dense connectivity of DenseNet121 or the self-supervised approach of DinoV2. MoCo v2 shows a significant decline in performance on the Xu dataset, with a precision of 0.6, a recall of only 0.0297, an F1-score of 0.0566, and an accuracy of 0.5. These data reflect severe performance issues of the model on this dataset. This stark contrast to its performance on the CCiC dataset (F1-score of 0.9891 and accuracy of 0.9892) suggests that MoCo v2 may be overfitting to specific characteristics of the CCiC dataset, limiting its ability to generalize to the Xu dataset. This highlights a potential limitation of the contrastive learning approach employed by MoCo v2 when faced with datasets that differ significantly from its training data. These results highlight the exceptional performance of DinoV2 on the Xu dataset, particularly in its ability to reduce false negatives. The perfect recall rate (1.0000) achieved by DinoV2 on the Xu dataset is especially significant in the context of crack detection, where missing a crack (false negative) could have more severe consequences than falsely identifying a crack (false positive).

Examining the results from the HBC2019 dataset, DinoV2 maintains the best performance, reporting the highest F1-score of 0.9346 and demonstrating an almost perfect recall rate

of 0.9868. Its precision was 0.8876, and its accuracy was 0.9731. Its performance underscores its strong capacity to minimize false negatives, thereby ensuring high-accuracy crack detection. The consistency of DinoV2's performance across different datasets is a strong indicator of its robustness and generalization capabilities. MoCo v2 also performed relatively well on this dataset, with an F1-score of 0.5593 and accuracy of 0.8659, although it lagged behind DinoV2 in precision and recall. This improved performance compared to the Xu dataset suggests that the HBC2019 dataset may share more similarities with the CCiC dataset, allowing MoCo v2 to leverage its learned features more effectively. Turning our attention to the SDNET2018 dataset, DinoV2 demonstrated excellent performance with a precision of 0.8924 and a recall rate of 0.6985, achieving the highest F1-score of 0.7836 and accuracy of 0.9416 among all models. The slightly lower recall rate on this dataset compared to others suggests that the SDNET2018 dataset may present unique challenges, possibly due to more subtle or complex crack patterns. The performance of DenseNet121 on the SDNET2018 dataset is better than previously described, with a precision of 0.7687, recall of 0.5551, F1-score of 0.5634, and accuracy of 0.8581. These metrics indicate that DenseNet121 performs relatively well in identifying true cracks, although there is still room for improvement. The performance of VGG16 and MobileNet v2 on the SDNET2018 dataset needs correction. Both models have identical performance metrics, with a precision of 0.4243, recall of 0.5, F1-score of 0.4590, and accuracy of 0.8486. These data suggest that while they perform decently in terms of accuracy, they still face challenges in precisely identifying cracks. ResNet50 and ResNet101 indeed show limited performance on this dataset, with F1-score of 0.4723 and 0.4641, respectively. MoCo v2 also performed poorly, with an F1-score of only 0.0115 and an accuracy of 0.8464. These results once again confirm DinoV2's efficiency and accuracy in handling datasets with complex backgrounds and varied material compositions. While other models like DenseNet121 and the ResNet series also demonstrate commendable performances across different datasets, they remain a step behind DinoV2, which consistently sets a higher standard in performance metrics, affirming its robustness and superior detection abilities in various scenarios. It is important to note the balanced approach of DinoV2 in maintaining high scores across all metrics, illustrating its all-rounded capabilities in crack detection tasks.

Although models such as DenseNet121 and the ResNet series have demonstrated excellent

performance across multiple datasets, DinoV2 consistently maintains a leading position in various scenarios, showcasing its superior robustness and detection capabilities. However, an in-depth analysis of performance differences across different datasets reveals some key insights. Firstly, the degree of data imbalance significantly impacts model performance, particularly affecting recall rates and F1-score. For instance, on the balanced CCiC dataset, all models performed exceptionally well, while on the highly imbalanced SDNET2018 (with only 15% positive samples), most models experienced a substantial performance decline. Although DinoV2 performed relatively best on SDNET2018 (F1-score 0.7836, accuracy 0.9416), its recall rate (0.6985) was lower than on other datasets, possibly due to the extreme class imbalance. Secondly, the importance of pre-training and model architecture is highlighted. DinoV2, DenseNet121, and the ResNet series demonstrated relatively stable performance when handling imbalanced data, which may be attributed to their effective feature extraction architectures and pre-training strategies. In particular, DinoV2's self-supervised learning method seems to enable it to better adapt to different data distributions and imbalance situations. In contrast, MoCo v2 performed poorly on imbalanced datasets, especially on SDNET2018, with an F1-score of only 0.0115, possibly due to challenges faced by its contrastive learning-based approach when dealing with skewed data. Furthermore, some models like VGG16 and MobileNet v2 performed well on CCiC but saw a sharp decline in performance on other imbalanced datasets, indicating their higher sensitivity to changes in data distribution and weaker generalization ability. Notably, even the best-performing DinoV2 faced challenges on the extremely imbalanced SDNET2018 dataset, highlighting the universal impact of class imbalance on model performance.

3.4.2 Cross-Dataset Evaluation of the Compared Methods

To further investigate the generalization capabilities of these models, we conducted cross-dataset evaluations. Tables 3.4, 3.5, 3.6, and 3.7 present the results of training on one dataset and testing on the others. This cross-dataset performance is crucial for assessing how well these models can generalize to new, unseen data, which is a critical factor in real-world applications where the test data may differ significantly from the training data. Table 3.4 shows the results of cross-testing on the remaining three datasets after training

on the CCiC dataset. In the cross-dataset evaluations, DinoV2 consistently demonstrated strong performance. When trained on the CCiC dataset (Table 3.4), DinoV2 achieved the highest accuracy on all three test datasets: SDNET2018 (0.8625), HBC2019 (0.9046), and Xu (0.9884). It also achieved the highest F1-score on HBC2019 (0.7836) and Xu (0.9913) datasets. However, on the SDNET2018 dataset, ResNet50 outperformed DinoV2 in terms of precision (0.6367 vs 0.5532), recall (0.5642 vs 0.4716), and F1-score (0.5726 vs 0.5092). However, the other compared methods show a degradation in performance when tested on SDNET2018 and HBC2019 datasets. This performance degradation is particularly noticeable for models like VGG16 and DenseNet121 on the SDNET2018 dataset. Despite some variations, DinoV2's overall strong performance across different datasets underscores its robust feature extraction and generalization capabilities, especially in terms of accuracy.

TABLE 3.4: Testing three other different models based on a model trained on the CCiC dataset. Values in bold indicate the model's best performance.

Dataset	Model	Precision	Recall	F1-score	Accuracy
SDNET2018	ResNet50	**0.6367**	**0.5642**	**0.5726**	0.8078
	ResNet101	0.5737	0.5610	0.5652	0.7624
	VGG16	0.4743	0.4570	0.4211	0.4872
	MobileNet_v2	0.4845	0.4741	0.4310	0.4938
	DenseNet121	0.4672	0.4459	0.3890	0.4295
	MoCo_v2	0.3461	0.1922	0.2471	0.8229
	DinoV2	0.5532	0.4716	0.5092	**0.8625**
HBC2019	ResNet50	**0.8597**	0.6461	0.6832	0.8554
	ResNet101	0.8360	0.6827	0.7216	0.8626
	VGG16	0.6761	0.6050	0.6214	0.8077
	MobileNet_v2	0.8180	0.7095	0.7440	0.8655
	DenseNet121	0.7114	0.6350	0.6562	0.8213
	MoCo_v2	0.6792	0.4800	0.5625	0.8557
	DinoV2	0.6876	**0.8933**	**0.7836**	**0.9046**
Xu	ResNet50	0.7435	0.7384	0.6504	0.6505
	ResNet101	0.7701	0.7886	0.7167	0.7175
	VGG16	0.7599	0.7926	0.7478	0.7535
	MobileNet_v2	0.7978	0.8345	0.7823	0.7864
	DenseNet121	0.8184	0.8554	0.8235	0.8319
	MoCo_v2	**1.0000**	0.6444	0.7837	0.7623
	DinoV2	0.9941	**0.9925**	**0.9913**	**0.9884**

Table 3.5 shows the results of cross-testing on the remaining three datasets after training on the Xu dataset. Overall, DinoV2 performed excellently on the CCiC dataset, achieving

the highest F1-score of 0.9567 and an accuracy of 0.9548. On the HBC2019 dataset, DinoV2 has the highest recall rate of 0.9709, but its accuracy of 0.7685 was lower than that of ResNet101 (0.8744), ResNet50 (0.8711), and MoCo_v2 (0.8686). However, on the SDNET2018 dataset, DinoV2's performance was not the best, with an accuracy of 0.7220, lower than MoCo_v2's 0.8157. It's worth noting that ResNet50 and ResNet101 also performed exceptionally well on the CCiC dataset, with accuracies of 0.9539 and 0.9538, respectively, which are very close to DinoV2's performance. These results demonstrate DinoV2's strong feature extraction and generalization capabilities across different datasets, especially its advantages in handling complex data, but also indicate that other models may perform better on certain specific datasets.

TABLE 3.5: Testing three other different models based on a model trained on the Xu dataset. Values in bold indicate the model's best performance.

Dataset	Model	Precision	Recall	F1-score	Accuracy
CCiC	ResNet50	0.9576	0.9539	0.9538	0.9539
	ResNet101	0.9617	0.9588	0.9538	0.9538
	VGG16	0.8773	0.8379	0.8335	0.8379
	MobileNet_v2	0.2500	0.5000	0.3333	0.5000
	DenseNet121	0.9053	0.8841	0.8826	0.8841
	MoCo_v2	**0.9689**	0.1250	0.2214	0.5605
	DinoV2	0.9171	**0.9999**	**0.9567**	**0.9548**
SDNET2018	ResNet50	0.5259	0.5432	0.4599	0.5026
	ResNet101	0.5092	0.5155	**0.4571**	0.5150
	VGG16	0.4638	0.4553	0.3081	0.3107
	MobileNet_v2	0.0911	0.5000	0.1542	0.1823
	DenseNet121	**0.4873**	0.4816	0.3608	0.3723
	MoCo_v2	0.1285	0.0377	0.0583	**0.8157**
	DinoV2	0.2901	**0.5790**	0.3865	0.7220
HBC2019	ResNet50	**0.7956**	0.7872	0.7912	0.8711
	ResNet101	0.8020	0.7887	**0.7951**	**0.8744**
	VGG16	0.6193	0.6888	0.5597	0.5890
	MobileNet_v2	0.0971	0.5000	0.1626	0.1943
	DenseNet121	0.5902	0.6156	0.4324	0.4363
	MoCo_v2	0.7609	0.4667	0.5785	0.8686
	DinoV2	0.4551	**0.9709**	0.6197	0.7685

Table 3.6 shows the results of cross-testing on the remaining three datasets after training on the HBC2019 dataset. Overall, DinoV2 performed excellently across all datasets. On the CCiC dataset, DinoV2 achieved the highest precision of 0.9945, F1-score of 0.9104,

and accuracy of 0.9093. On the Xu dataset, DinoV2 also excelled, obtaining the highest precision of 0.9761, recall of 0.9887, F1-score of 0.9824, and accuracy of 0.9763. However, on the SDNET2018 dataset, DinoV2's accuracy of 0.6863 was lower than some other models, such as VGG16 and MobileNet_v2 both at 0.8176. Nevertheless, DinoV2 still achieved the highest recall rate of 0.5586 on this dataset. It's worth noting that ResNet101 performed well on the CCiC dataset with an accuracy of 0.8559, while MoCo_v2 achieved the highest recall rate of 0.9538 on the CCiC dataset. On the SDNET2018 dataset, ResNet101 achieved the highest F1-score of 0.5100, while VGG16 and MobileNet_v2 had the highest accuracy of 0.8176. These results demonstrate DinoV2's strong feature extraction and generalization capabilities across different datasets, especially on the CCiC and Xu datasets, but show relatively weaker performance on the SDNET2018 dataset. The varied performance of different models across datasets highlights the importance of model selection based on specific dataset characteristics and performance metrics of interest.

TABLE 3.6: Testing three other different models based on a model trained on the HBC2019 dataset. Values in bold indicate the model's best performance.

Dataset	Model	Precision	Recall	F1-score	Accuracy
CCiC	ResNet50	0.8313	0.8153	0.8130	0.8153
	ResNet101	0.8872	0.8559	0.8530	0.8559
	VGG16	0.2500	0.5000	0.3333	0.5000
	MobileNet_v2	0.2500	0.5000	0.3333	0.5000
	DenseNet121	0.5494	0.5437	0.5301	0.5437
	MoCo_v2	0.6008	**0.9538**	0.7372	0.6601
	DinoV2	**0.9945**	0.9215	**0.9104**	**0.9093**
SDNET2018	ResNet50	0.5161	**0.5268**	0.4745	0.5422
	ResNet101	**0.5204**	0.5130	**0.5100**	0.7575
	VGG16	0.4088	0.5000	0.4498	**0.8176**
	MobileNet_v2	0.4088	0.5000	0.4498	**0.8176**
	DenseNet121	0.4768	0.4835	0.4771	0.7312
	MoCo_v2	0.1179	0.0755	0.0920	0.7748
	DinoV2	0.2549	**0.5586**	0.3501	0.6863
Xu	ResNet50	0.6781	0.6800	0.6055	0.6055
	ResNet101	0.7348	0.7196	0.6247	0.6253
	VGG16	0.1659	0.5000	0.2491	0.3318
	MobileNet_v2	0.1659	0.5000	0.2491	0.3318
	DenseNet121	0.6512	0.5151	0.2845	0.3529
	MoCo_v2	0.6688	0.9721	0.7990	0.6665
	DinoV2	0.9761	**0.9887**	**0.9824**	**0.9763**

Finally, Table 3.7 shows the results of cross-testing on the remaining three datasets after training on the SDNET2018 dataset. Overall, DinoV2 performed excellently across all datasets. On the CCiC dataset, DinoV2 achieved the highest precision of 0.9767, F1-score of 0.8499, and accuracy of 0.8671. On the HBC2019 dataset, DinoV2's performance was equally impressive, reaching the highest precision of 0.9513, F1-score of 0.8536, and accuracy of 0.9484, although its recall rate of 0.7741 was significantly lower than that of Moco v2 of 0.9392. On the Xu dataset, DinoV2 once again demonstrated outstanding performance, obtaining the highest precision of 0.9917, recall of 0.9386, F1-score of 0.9644, and accuracy of 0.9537. It's worth noting that other models also performed well in certain metrics. For instance, on the CCiC dataset, Moco v2 achieved the highest recall rate of 0.8279, while MobileNet v2 showed good overall performance with precision, recall, F1-score, and accuracy all above 0.78. On the HBC2019 dataset, other models (such as ResNet50, ResNet101, VGG16, and MobileNet v2) all reached the same accuracy of 0.8056 apart from DinoV2, which is notably high considering DinoV2's accuracy of 0.9484. On the Xu dataset, Moco v2's recall rate of 0.7356 was second only to DinoV2, although its other metrics were considerably lower. The performance of other models on this dataset was generally poor, with accuracies ranging from 0.3318 to 0.4200. These results further demonstrate DinoV2's strong feature extraction and generalization capabilities in handling complex and imbalanced data, especially its advantages in tackling challenging datasets. However, the results also show that in certain specific cases, other models may perform better on some metrics, particularly in terms of recall rate where MoCo v2 excelled.

Through in-depth analysis of the CCiC, SDNET2018, Xu, and HBC2019 datasets, we discovered that DinoV2 exhibits exceptional performance in multiple aspects, particularly in key metrics such as accuracy, recall rate, and F1-score, significantly outperforming other models. Despite training DinoV2 for only 5 epochs, its accuracy did not show significant improvement with further training. However, its high sensitivity to complex features and outstanding performance in handling highly imbalanced data, especially achieving a 100% recall rate, an F1-score of 0.9884, and an accuracy of 0.9844 on the Xu dataset, not only demonstrate its strong capability in identifying complex features but also showcase its superior generalization performance. While DenseNet121 performed best on the CCiC dataset, a direct comparison with DinoV2 reveals the latter's more pronounced advantages

TABLE 3.7: Testing three other different models based on a model trained on the SDNET2018 dataset. Values in bold indicate the model's best performance.

Dataset	Model	Precision	Recall	F1-score	Accuracy
CCiC	ResNet50	0.7557	0.5224	0.3813	0.5224
	ResNet101	0.7558	0.5227	0.3820	0.5227
	VGG16	0.2500	0.5000	0.3333	0.5000
	MobileNet_v2	0.8514	0.7947	0.7861	0.7947
	DenseNet121	0.6970	0.5198	0.3805	0.5198
	MoCo_v2	0.4616	**0.8279**	0.5927	0.4311
	DinoV2	**0.9767**	0.7522	**0.8499**	**0.8671**
HBC2019	ResNet50	0.4028	0.5000	0.4461	0.8056
	ResNet101	0.4028	0.5000	0.4461	0.8056
	VGG16	0.4028	0.5000	0.4461	0.8056
	MobileNet_v2	0.4028	0.5000	0.4461	0.8056
	DenseNet121	0.5999	0.5045	0.4595	0.8041
	MoCo_v2	0.1888	**0.9392**	0.3145	0.2045
	DinoV2	**0.9513**	0.7741	**0.8536**	**0.9484**
Xu	ResNet50	0.6688	0.5173	0.2779	0.3493
	ResNet101	0.6686	0.5122	0.2760	0.3481
	VGG16	0.1659	0.5000	0.2491	0.3318
	MobileNet_v2	0.6819	0.5659	0.3833	0.4200
	DenseNet121	0.6722	0.5273	0.3081	0.3684
	MoCo_v2	0.6384	0.7356	0.6836	0.5450
	DinoV2	**0.9917**	**0.9386**	**0.9644**	**0.9537**

in overall performance and adaptability in specific scenarios. Moreover, although other models like the ResNet series and DenseNet121 also showed commendable performance across different datasets, they generally fell a step behind DinoV2 in comparisons. This further emphasizes the higher standard set by DinoV2 across all performance metrics, proving its robust detection capabilities and resilience in various scenarios. Notably, DinoV2 maintains high scores across all evaluation metrics, demonstrating its comprehensive abilities in crack detection tasks. Additionally, MoCo_v2, as another self-supervised learning method, also demonstrated good performance across multiple datasets. Although it slightly lagged behind DinoV2 in some metrics and its performance may not be as strong as DinoV2, MoCo_v2 still performed well in handling complex and imbalanced data, further proving the advantages of self-supervised learning methods in feature extraction and generalization capabilities. Overall, both DinoV2 and MoCo_v2 exhibited their potential and advantages in crack detection tasks, especially when dealing with challenging datasets.

The superior performance of DinoV2 can be attributed to its self-supervised learning approach. By learning to recognise and differentiate between various image transformations during pre-training, DinoV2 develops a rich, generalizable understanding of image features. This allows it to adapt more effectively to new datasets and crack patterns that it hasn't explicitly seen during training. In contrast, the performance of other models, including MoCo v2, varied more significantly across different cross-dataset scenarios. This variability highlights the challenges that traditional supervised learning and even some self-supervised approaches face when generalizing to new datasets. For instance, MoCo v2's performance was inconsistent across different cross-dataset evaluations. While it performed well in some scenarios, such as when trained on HBC2019 and tested on CCiC (Table 6), it struggled in others, like when trained on Xu and tested on SDNET2018 (Table 5). This inconsistency suggests that MoCo v2's contrastive learning approach, while effective in some cases, may not be as robust as DinoV2's approach when faced with significantly different datasets. The traditional CNN architectures (ResNet, VGG16, MobileNet v2, DenseNet121) also showed limitations in cross-dataset generalization. Their performance often dropped significantly when tested on datasets different from their training data. This observation underscores the importance of developing models with strong generalization capabilities, especially in domains like crack detection, where the test data may vary considerably from the training data due to differences in materials, lighting conditions, and crack patterns. In conclusion, our comprehensive analysis across multiple datasets and cross-dataset evaluations provide strong evidence for the superior performance and generalization capabilities of DinoV2 in crack detection tasks. Its consistent high performance, particularly in recall and accuracy, makes it a promising candidate for real-world applications where robustness and reliability are crucial. The self-supervised learning approach employed by DinoV2 appears to be a key factor in its ability to adapt to various datasets and crack patterns, outperforming both traditional CNN architectures and other self-supervised methods like MoCo v2.

3.4.3 The Impact of Class Imbalance on Features Extracted from Self-Supervised vs. Supervised Models

As demonstrated in our results section, features extracted through self-supervised learning methods, such as DinoV2, exhibit significant advantages in handling naturally imbalanced data. Particularly on the highly imbalanced datasets SDNET2018 and HBC2019, DinoV2 not only leads substantially in the F1-score (0.7836 for SDNET2018 and 0.8536 for HBC2019). This contrasts sharply with features extracted using traditional supervised learning methods, whose predictions often heavily favor the majority class, resulting in significantly diminished performance on highly imbalanced data (for example, the F1-score of supervised models like ResNet50, ResNet101, VGG16, MobileNetV2, and DenseNet121 are extremely low on the SDNET2018 dataset). The features extracted from supervised-based models are biased towards the majority class and fail to predict cracks effectively. This performance disparity becomes more pronounced as the level of class imbalance increases. In datasets with extreme imbalance ratios, such as SDNET2018, the superiority of DinoV2 in identifying the minority class is particularly evident. This suggests that self-supervised learning methods are more robust to class imbalance, maintaining their ability to extract meaningful features even when positive samples are scarce. Interestingly, features learned through the self-supervised method (DinoV2) consistently outperform their corresponding baselines. Significant improvements on extremely imbalanced datasets suggest that the biased label information can be greatly mitigated through self-supervision[97]. This mitigation of bias is crucial in real-world applications where class imbalance is common, such as crack detection. In these scenarios, the ability to accurately identify rare but critical instances (like cracks) is of paramount importance.

3.4.4 Attention Visualization

As shown in the model attention map in Fig.3.6, in datasets predominantly characterized by expansive viewpoints of cracks, conventional CNN architectures marginally outperformed DinoV2. Such performance nuances may be attributed to the rigorous training paradigms inherent to CNNs, which possibly equips them with a specialized prowess in recognizing cracks from expansive viewpoints. This depth of specialized learning might be

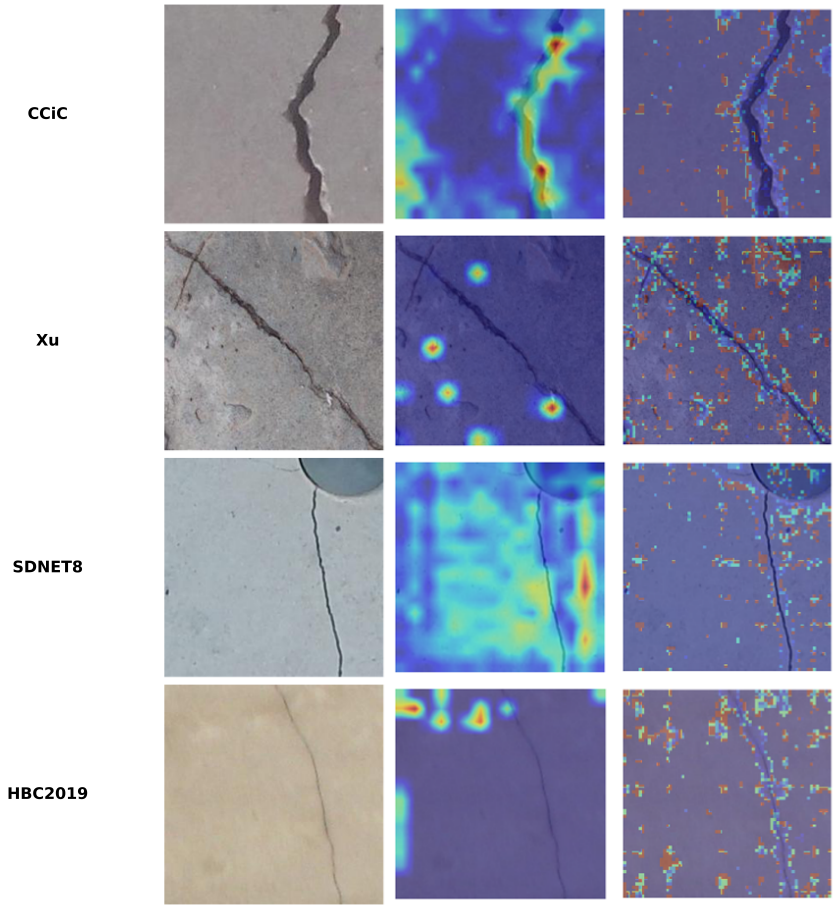


FIGURE 3.6: Attention visualization of the models across four datasets. The three columns represent Ground Truth, ResNet50, and DinoV2 respectively.

the impetus behind CNN's advantageous stance in this specific scenario. However, when the nature of the dataset diversifies, capturing cracks across diverse environments and materials, DinoV2 distinctly showcases its superior generalization capability. We must also acknowledge that the attention mechanism of our DinoV2 model occasionally fixates on areas that are not cracked, particularly in the Xu and HBC datasets. This observation suggests that, despite its general robustness, the DinoV2's attention could be influenced by features within the image that resemble cracks, potentially affecting the accuracy of the classification. In these heterogeneous datasets, The advantage of DinoV2 lies in its self-supervised learning approach, enabling it to decipher data structures without extensive labeled datasets. Nonetheless, we have observed instances where DinoV2's adaptability to environmental variances and material textures leads to some misallocation of attention. This reinforces the notion that self-supervised learning focuses on discerning fundamental

data features and structures rather than solely on annotations, which can be a double-edged sword in terms of model precision.

In practical applications, choosing the right model is crucial. CNN architectures may lead to large datasets with specific labeled data types. But for tasks requiring adaptability across different scenarios and materials for crack detection, DinoV2 stands out as a strong contender, despite the need for vigilance regarding its attention distribution. Furthermore, an astute observation reveals DinoV2's heightened adaptability to environmental variances and distinct material textures. This could be a testament to the fact that self-supervised learning leans more towards discerning fundamental features and structures within data rather than its annotations. Such innate focus on data's structural integrity potentially renders DinoV2 more adept at navigating unfamiliar terrains or novel material types. In practical implementations, judicious model selection remains pivotal. For voluminous datasets with a specific type of labeled data, CNN architectures might be the frontrunner. However, when the task at hand mandates a model's adaptability across a spectrum of scenarios and materials for crack detection, DinoV2 indisputably emerges as a formidable contender. These insights illuminate the profound potential of self-supervised learning, particularly the DinoV2 model, in crack detection and classification endeavors. Conversely, traditional supervised learning architectures might necessitate intricate fine-tuning and optimization when pitted against dynamically diverse datasets.

3.5 Summary

This study showcases the prowess of the self-supervised model, DinoV2, in crack classification over diverse datasets. We observed that while DinoV2 can identify cracks adeptly, its attention mechanism tends to cover the entire image rather than centring exclusively on the crack regions. This expansive attention span could sometimes lead to distractions in the model's output. Compared to traditional supervised models like CNNs, which sometimes show inconsistencies over different datasets, DinoV2 stands out with its persistent robustness and precision. A significant strength of this model is its ability to leverage the inherent data information without extensive annotations, enhancing both generalization and efficiency. In light of these findings, we are optimistic about DinoV2's potential to lead

the way in crack classification, paving a path for broader applications in crack detection and beyond. Recognizing this limitation, future studies will focus on refining DinoV2's feature extraction mechanism to minimize these distractions and improve its accuracy.

Chapter 4

Source-Free Domain Adaptation for Concrete Crack Classification Using CLIP

4.1 Introduction

Concrete structures currently serve as core components of modern buildings and infrastructure, and their structural health can be directly related to the safety and service life of engineering facilities[98]. As the most common cause of accidents in concrete structures, concrete cracks require timely detection, classification, and maintenance, which can play a significant role in preventing structural failure and guiding repairs. The traditional crack detection method is usually manual visual inspection, which is time-consuming and laborious. Meanwhile, inspection results are often affected by human subjective factors, which makes it difficult to meet the increasing inspection demands of large-scale engineering facilities[99].

With the development of computer vision and deep-learning technology, automated crack detection and classification based on computer vision has gradually become a popular topic. These methods not only address the limitations of manual detection but also provide more objective detection results. However, the application of machine learning

methods across different concrete datasets faces significant challenges due to inherent distribution shifts. These variations arise from multiple factors including surface texture differences (such as roughness and porosity), environmental conditions (lighting, moisture, and weathering effects), imaging settings (camera angles, resolutions, and exposure), and crack morphological characteristics. Such domain gaps often result in substantial performance degradation when models trained on one type of concrete surface are deployed on another, necessitating costly data collection and model retraining processes for each new deployment scenario[100, 101].

This challenge is widely recognized in the machine learning community as the domain adaptation problem, where the goal is to improve the generalization ability of the model by reducing the difference in feature distribution between the source and target domains[102]. Traditional domain adaptation methods mainly achieve knowledge migration through adversarial learning and feature alignment. These methods usually require simultaneous access to data in both the source and target domains, which tends to increase the storage costs and resource usage limitations[103]. Most also rely on task-specific labeling spaces, making it difficult to cope with new categories or complex semantic changes emerging in the target domain.

With the present evolution of large language models, the contrastive language image pretraining (CLIP) proposed by OpenAI has become the representative model at present[104]. It builds a powerful bridge for visual language understanding in multimodal domains by learning from 400 million images and text comparisons, breaking through the traditional visual dependence on fixed category labeling restrictions for the first time. It achieves visual task definition through natural language description and at the same time effectively handles the differences between different visual domains by its domain generalization capability, providing a new solution for domain adaptation tasks[105].

To address these challenges, we propose an approach that enables effective domain adaptation for crack detection without requiring source domain data during adaptation. Our method integrates three key components: (1) a CLIP-based pseudo-label generation strategy that leverages natural language descriptions to bridge domain gaps, (2) a three-layer

feature fusion network for comprehensive feature alignment, and (3) an efficient data selection mechanism inspired by Moderate Coreset. While our method is primarily validated on concrete crack detection scenarios, its underlying principles of cross-modal feature alignment and efficient data selection can be generalized to other computer vision tasks where domain adaptation is required, such as medical image analysis or industrial defect detection.

1. A CLIP-based pseudo-label generation strategy that aligns crack features with natural language descriptions, breaking through traditional fixed-category labelling limitations and improving cross-domain categorization accuracy.

2. A data selection module inspired by Moderate Coreset that identifies representative samples for efficient model training, effectively reducing computational overhead while maintaining domain adaptation performance.

3. A three-layer feature fusion network that adaptively integrates CLIP's high-level semantic features with CNN's low-level visual features, enhancing both local crack recognition and domain generalization capability.

4. Extensive experimental validation across different materials and data ratios demonstrates the effectiveness and superiority of our proposed approach in source-free domain adaptation for crack classification.

4.2 Methodology

In this section, we introduce a domain adaptation framework designed to address the challenges of crack classification under source-free scenarios. As illustrated in Figure 4.1, the proposed method leverages the powerful cross-modal understanding capabilities of the CLIP model to generate high-quality pseudo-labels, ensuring effective utilization of target domain data without relying on source domain samples. To further enhance the quality of pseudo-labels and optimize domain adaptation performance, we incorporate a moderate coreset selection strategy that identifies the most representative samples in the feature space. Additionally, we propose a three-layer feature fusion network that aligns features

across domains at multiple abstraction levels, supported by complementary alignment loss functions.

4.2.1 Clip

Owing to the powerful cross-modal understanding capability of the CLIP model obtained through pre-training on large-scale image-text pairs, it can effectively map images and text into the same feature space, as shown in Figure 1. We leverage this characteristic to address the issue of unavailable source-domain data by using the pre-trained CLIP model to generate high-quality pseudo-labels. First, given an input image f, we extract image features through CLIP's visual encoderf_v as shown in equation 4.1 [106].

$$h_l = f_v(I) \tag{4.1}$$

Meanwhile, we design two text prompts $\mathrm{T_c}$ (a photo of concrete surface with cracks) and $\mathrm{T_n}$ (a photo of concrete surface without cracks) , Through CLIP's text encoder f_t, we obtain the corresponding text features as shown in equation 4.2.

$$\begin{aligned} \mathrm{h_{T_c}} &= \mathrm{f_t}\,(\ \mathrm{T_c}) \\ \mathrm{h_{T_n}} &= \mathrm{f_t}\,(\ \mathrm{T_n}) \end{aligned} \tag{4.2}$$

To ensure feature comparability, we performed L2 normalization on all features as shown in equation 4.3, where $\mathrm{h_I}$ represents the image features extracted from the CLIP visual encoder, $\mathrm{h_{T_c}}$ denotes the text features of crack class descriptions, and $\mathrm{h_{T_n}}$ represents the text features of no-crack class descriptions. The normalized features $\widehat{\mathrm{h_I}}$, $\widehat{\mathrm{h_c}}$, and $\widehat{\mathrm{h_{T_n}}}$ are obtained by dividing each feature vector by its L2 norm, ensuring all features have unit length and lie on a hypersphere, which is essential for subsequent similarity computations.

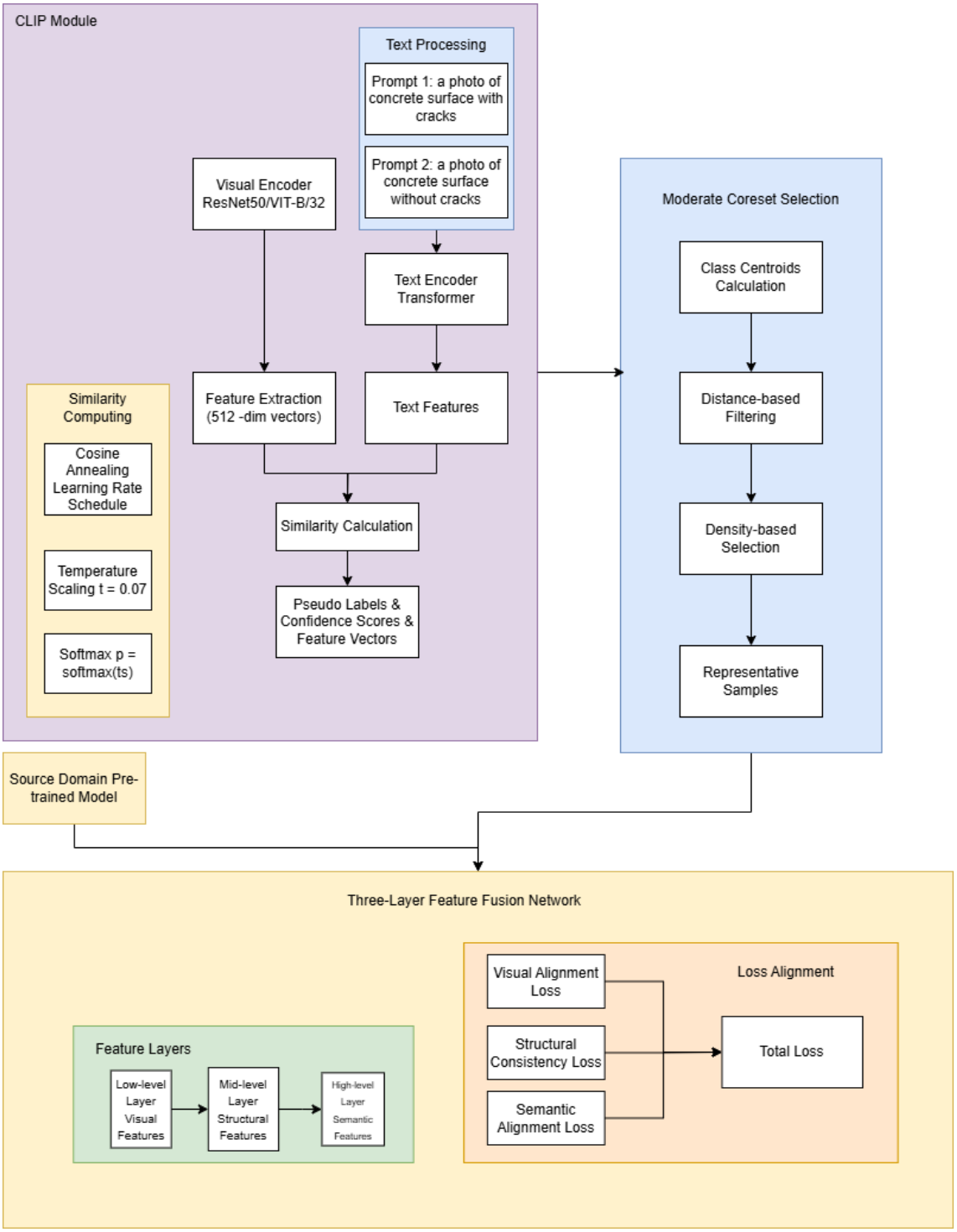


FIGURE 4.1: Overview of the proposed source-free domain adaptation framework for crack classification. The framework consists of three main components: (1) CLIP-based pseudo-label generation with confidence assessment, (2) moderate coreset selection for representative sample filtering, and (3) three-layer feature fusion network with hierarchical alignment strategies.

$$\begin{aligned} \widehat{\mathrm{h_I}} &= \frac{\mathrm{h_I}}{\| \mathrm{h_l} \|_2} \\ \widehat{\mathrm{h_c}} &= \frac{\mathrm{h_{T_c}}}{\|\mathrm{h_{T_e}}\|_2} \\ \widehat{\mathrm{h_{T_n}}} &= \frac{\mathrm{h_{T_n}}}{\| \mathrm{h_{T_n}} \|_2} \end{aligned} \tag{4.3}$$

4.2.2 Pseudo-label Generation and Confidence Assessment

Building upon the normalized features obtained from the previous section, we leveraged CLIP's cross-modal alignment capability to generate pseudo-labels. Specifically, we determined the image category by calculating the cosine similarity between the normalized image features $\widehat{\mathrm{h_I}}$ and the normalized text features ($\widehat{\mathrm{h_{T_c}}}$ for crack and $\widehat{\mathrm{h_{T_n}}}$ for no-crack), as shown in equation 4.4.

$$\mathrm{s} = \left[\widehat{\mathrm{h_I}} \cdot \widehat{\mathrm{h_{T_c}}}, \widehat{\mathrm{h_I}} \cdot \widehat{\mathrm{h_{T_n}}} \right] \tag{4.4}$$

The resulting similarity vector s contains two values representing the semantic similarity between the input image and each class description. To convert these similarity scores into probability distributions, temperature scaling and softmax functions were applied, as shown in equation 4.5.

$$\mathrm{p} = \mathrm{softmax}(\tau \mathrm{s}) \tag{4.5}$$

where τ is the temperature parameter that controls the sharpness of the probability distribution. The final pseudo-label and confidence level are determined by the maximum value in the probability distribution, as shown in equation 4.6.

$$\begin{aligned} y_{pseudo} &= \arg\max_i p_i \\ c &= \max_i p_i \end{aligned} \tag{4.6}$$

To facilitate subsequent domain adaptation training, we extracted and saved three key components for each image: a 512-dimensional CLIP image feature vector, binary classification pseudo-labels (where 0 indicates no cracks and 1 indicates cracks), and confidence scores for the model predictions. This information was systematically organized and stored in an array format, providing a structured database for subsequent model training and evaluation. Among them, the high-confidence pseudo-labels are used as reliable supervisory signals in the domain adaptation process, whereas the extracted feature vectors provide a good initial feature representation for the domain adaptation task, which helps to improve the performance of the model on the target domain.

4.2.3 Moderate coreset selection

To further improve the quality of pseudo-labels and study the impact of different data sizes on domain adaptation performance, we employed the moderate coreset selection strategy [107] to identify and retain the most representative samples while filtering out potential noisy or redundant samples. This method operates in the feature space by analyzing the distribution characteristics of samples. Specifically, for each category (crack and no-crack), we first calculated its centroid in the feature space, as shown in equation 4.7. The selection process is based on two key criteria: distance-based filtering and density-based selection. In distance-based filtering, samples that are too far from their respective class centroids are removed as potential outliers or noisy samples. Through density-based selection, we select samples from the remaining set that best represent the data distribution, maintaining a balance between diversity and representativeness. This approach effectively removes three types of samples: those with unreliable pseudo-labels (samples far from class centroids), redundant samples in densely populated regions, and outliers that may negatively impact the domain adaptation process.

$$\mu_{\mathrm{c}} = \frac{1}{|\ \mathrm{S_c}|} \sum_{\mathrm{x_i} \in\ \mathrm{S_c}} \mathrm{f_i} \tag{4.7}$$

where S_c represents the sample set of class c and f_i is the CLIP feature vector of sample x_i. Then, we calculated the Euclidean distance from each sample to its corresponding class center, as shown in equation 4.8.

$$\mathrm{d_i} = |\mathrm{f_i} - \mu_{y_i}|_2 \tag{4.8}$$

where y_i is the pseudo-label of sample x_i. To select the most representative samples, we calculated the median d_{med} of all distances and ranked them according to their proximity to the median distance, as shown in equation 4.9.

$$\mathrm{s_i} = |\mathrm{d_i} - \mathrm{d_{med}}\,| \tag{4.9}$$

Finally, we select the top αN samples with the smallest s_i. Where α is the preset selection ratio set to 1.0, 0.8, 0.6, 0.4, and 0.2. This selection strategy ensures that the selected samples are neither too concentrated around nor too far from the class centre.

4.2.4 Three-Layer Feature Fusion Network

Inspired by the hierarchical feature alignment strategy [108] and correlation alignment techniques [109], we propose a domain adaptation method based on hierarchical feature alignment. Building upon the research findings from deep adaptation networks [110], we gradually reduce the discrepancy between source and target domains while maintaining the effectiveness of discriminative features through feature alignment strategies at three different levels.

To achieve this goal, we design a three-layer feature fusion network architecture, where each layer processes features at different levels of abstraction. The low-level feature extraction is formulated as shown in equation 4.10.

$$\mathrm{f_{low}} = \phi_{\mathrm{low}}\,(\mathrm{x}) = \mathrm{BN}\,(\mathrm{Dropout}\,(\mathrm{ReLU}\,(\,\mathrm{W_{low}\,x} + \mathrm{b_{low}}\,))) \tag{4.10}$$

This layer mainly maintains feature structure information and captures inter-domain structural similarity through covariance. The intermediate feature layer is as shown in equation 4.11.

$$\mathrm{f_{mid}} = \phi_{\mathrm{mid}}\ (\mathrm{f_{low}}) = \mathrm{BN}\left(\mathrm{Dropout}\left(\mathrm{ReLU}\left(\mathrm{W_{mid}}\left(\mathrm{f_{low}} + \mathrm{b_{mid}}\right)\right)\right)\right) \tag{4.11}$$

This layer mainly maintains the feature structure information and captures the inter-domain structural similarity through covariance. The high-level feature layer is as shown in equation 4.12.

$$\mathrm{f_{high}} = \phi_{\mathrm{hilh}}\ (\mathrm{f_{mid}}) = \mathrm{BN}\left(\mathrm{ReLU}\left(\mathrm{W_{high}}\,\mathrm{f_{mid}} + \mathrm{b_{high}}\right)\right) \tag{4.12}$$

This layer focuses on feature alignment at the semantic level using the confidence information of pseudo-labels to guide the migration of high-level semantic features.

To achieve full domain adaptation, we designed three complementary alignment loss functions. The visual alignment loss is as shown in equation 4.13.

$$\mathcal{L}_{\mathrm{visual}} = \left|E_{x_s \sim X_s}\left[f_{low}\left(x_s\right)\right] - E_{x_t \sim X_t}\left[f_{low}\left(x_t\right)\right]\right|_2^2 \tag{4.13}$$

The loss of structural consistency is as shown in equation 4.14.

$$\mathcal{L}_{\mathrm{struct}} = \left|\mathrm{Cov}\left(f_{mid}\left(X_s\right)\right) - \mathrm{Cov}\left(f_{mid}\left(X_t\right)\right)\right|_F^2 \tag{4.14}$$

where $(\mathrm{Cov}(\cdot))$ represents the feature covariance matrix, and $(|\cdot|_F)$ represents the Frobenius norm. The third is semantic alignment loss, as shown in equation 4.15.

$$\mathcal{L}_{\mathrm{semantic}} = \mathrm{MMD}\left(f_{high}\left(X_s\right) \odot c_s, f_{high}\left(X_t\right)\right) \tag{4.15}$$

where c_s represents the pseudo-label confidence of the source domain samples, and $\odot$ represents element-wise multiplication. The total domain-adaptation loss function is the weighted sum of these three losses, as shown in equation 4.16.

$$\mathcal{L}_{\text{align}} = \lambda_{visual}\mathcal{L}_{\text{visua}\ell} + \lambda_{struct}\mathcal{L}_{\text{struct}} + \lambda_{semantic}\mathcal{L}_{\text{semantic}} \tag{4.16}$$

The final optimization objective combines the classification loss and domain adaptation loss, as shown in equation 4.17.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\rfloor \ell \lfloor} + \lambda_{align}\mathcal{L}_{\neg \ell \rangle \} \backslash} \tag{4.17}$$

where (λ_{align}) is the weight coefficient used to balance the domain adaptation tasks.

4.3 Experiments

4.3.1 Dataset

4.3.1.1 SDNET2018

This study conducted experiments using the SDNET2018 dataset, a large-scale structural crack image dataset collected and labeled by Utah State University. As shown in Table 4.1, the dataset contained 11,200 high-resolution (4032 × 3024 pixels) images of concrete surfaces covering three different structural types: bridge decks (2,025 images), pavements (2,507 images), and walls (6,668 images). The images for each structure type contained both cracked and non-cracked samples, which maintained a balanced distribution, as illustrated in Figure 4.2. The images were acquired under different environmental conditions (e.g., light and weather) and contained various types of cracks (transverse, longitudinal, mesh, etc.) ranging from 0.06 mm to 25 mm, as well as various surface textures, stains, and non-crack defects.

Table 4.1: Distribution of SDNET2018 Dataset

Structure Type	**Total Images**	**Cracked**	**Non-cracked**
Bridge Decks	13,620	2,025	11,595
Pavements	24,334	2,608	21,726
Walls	18,138	3,851	14,287
Total	56,092	8,484	47,608

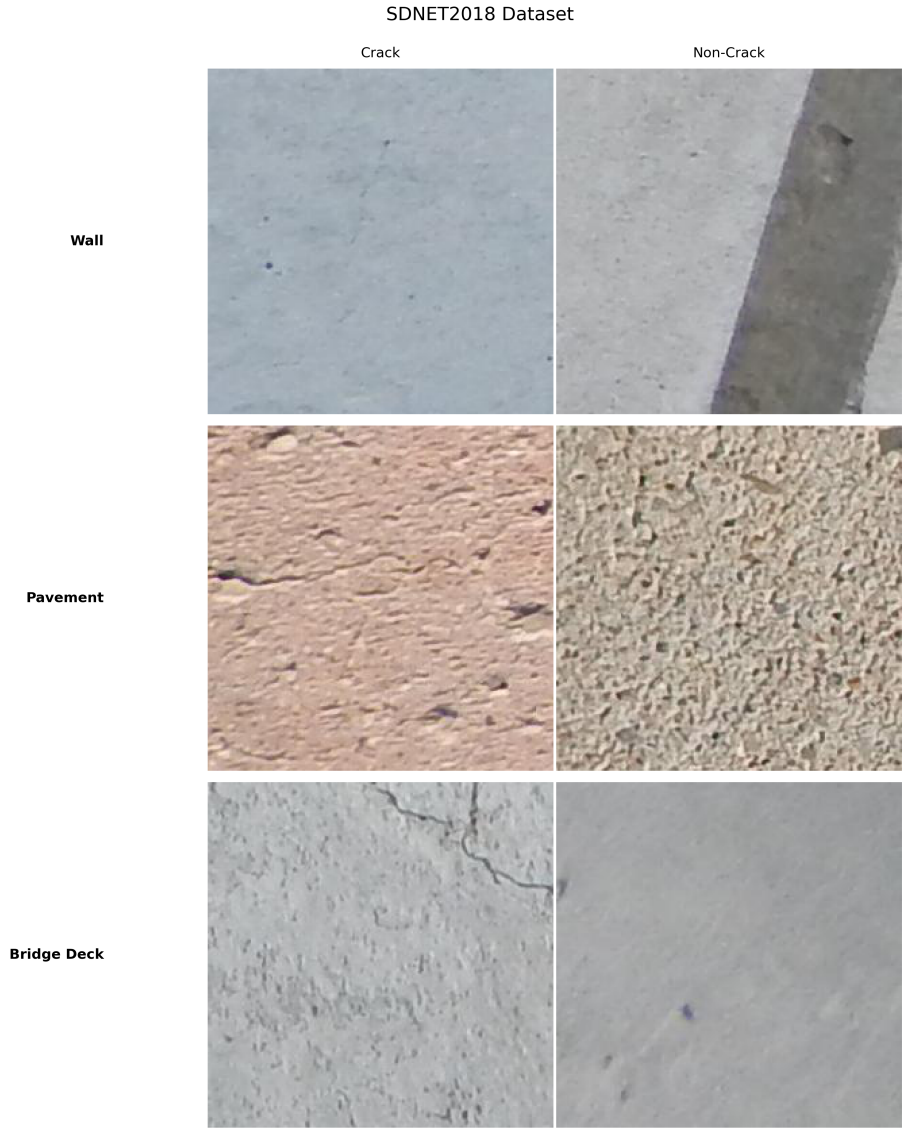


FIGURE 4.2: Distribution of images in SDNET2018 dataset across different structure types and their crack/non-crack categories.

4.3.1.2 Office-31 Dataset

The Office-31 dataset is a widely used benchmark dataset in domain adaptation research, consisting of 4,652 images across 31 categories collected from three distinct domains: Amazon (A), Webcam (W), and DSLR (D). The Amazon domain contains product images downloaded from amazon.com, the Webcam domain includes images captured by web cameras in office environments with various lighting conditions and poses, and the DSLR domain comprises high-resolution images taken by a digital SLR camera. This dataset is

particularly valuable for evaluating domain adaptation methods due to its realistic domain shifts and varying image qualities across different domains.

TABLE 4.2: Detailed Statistics of the Office-31 Dataset

Domain	**Total Images**	**Categories**
Amazon (A)	2,817	31
Webcam (W)	795	31
DSLR (D)	498	31
Total	4,110	31

Amazon (A) domain consists of images downloaded from amazon.com product pages
Webcam (W) domain contains images captured by a web camera in office environment
DSLR (D) domain includes high-resolution images taken by a digital SLR camera
All three domains contain the same 31 object categories but with varying image qualities and viewpoints

4.3.2 Experiment Setting

All experiments in this study were conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 (24GB of video memory). Model training was performed using a small-batch stochastic gradient descent method with a batch size of 32, parameter optimization using the Adam optimizer with an initial learning rate set to 0.001, and learning rate tuning using a cosine annealing strategy. The model training lasted for 100 epochs, and four work processes (num_workers) were used for data loading. For feature extraction, we used 512-dimensional CLIP features as inputs, which were processed through a three-layer feature fusion network with the hidden dimensions of each layer also maintained at 512 dimensions. To ensure the reproducibility of the experiments, the random seed was fixed at 42. In terms of the weight configuration of the loss function, the total feature alignment loss weight was 0.5, and the weights of the visual, structural, and semantic alignment losses were 0.2, 0.3, and 0.5, respectively. To comprehensively evaluate the performance of the model under different data sizes, we used source domain data with 100%, 80%, 60%, 40%, and 20% with five different sampling ratios for the experiments, while the complete dataset was always used for the target domain.

TABLE 4.3: Performance Comparison across Different Domain Adaptation Methods

Method	Tasks					
	B→P	B→W	P→B	P→W	W→B	W→P
ResNet50[111]	0.4816	0.5890	0.3701	0.3830	0.6230	0.4818
ViT[112]	0.6899	0.6704	0.5735	0.5513	0.7344	0.7184
DANN[113]	0.8911	0.8674	0.8942	0.8448	0.8516	0.8880
CDAN[114]	0.5892	0.7913	0.8491	0.5661	0.8513	0.8765
Our Method	**0.9488**	**0.9691**	**0.9878**	**0.9823**	**0.9842**	**0.9566**

B: Bridge, P: Pavement, W: Wall

4.4 Results

As shown in Table 4.3, our experimental results demonstrate significant performance advantages in the crack-detection migration task for different material types. In all six migration scenarios, our proposed method achieves a target domain test accuracy of more than 95%, which is significantly better than that of the baseline method. Specifically, in the bridge-to-pavement (B→P) migration task, the present method achieves 94.88% accuracy, which is 5.77 percentage points better than that of DANN (89.11%) and 35.96 percentage points better than that of CDAN (58.92%). The baseline models ResNet50 and ViT only achieved 48.16% and 68.99% accuracy, respectively, in this scenario, indicating that direct migration is less effective.

In the other two scenarios, bridge → walls and pavement → bridge, our method achieved 96.91% and 98.78% accuracy, respectively, which are the two best performers among all migration scenarios. Especially in the Pavement to Bridge migration, it improves 13.87 and 9.36 percentage points compared to CDAN (84.91%) and DANN (89.42%), respectively, showing a significant performance advantage. In pavement-to-wall (B→W) migration, the proposed method achieves 98.23% accuracy compared to only 56.61% for CDAN and 84.48% for DANN, indicating that the proposed method has a significant advantage in dealing with challenging cross-material migration tasks. Similarly, in wall-to-bridge (wall → bridge) migration, the 98.42% accuracy of the proposed method significantly outperforms the other methods.

Even in the relatively difficult wall-to-pavement (W→P) migration scenario, the present

method still maintains an accuracy of 95.66%, and although CDAN (87.65%) and DANN (88.80%) perform relatively better in this scenario, the present method still maintains a clear lead. It is worth noting that the ResNet50 and ViT models generally perform poorly in direct migration, with most accuracies being lower than 70%, which further proves the necessity of domain-adaptive methods in the cross-domain migration task. The method in this study maintains a consistently high performance in all migration scenarios, which fully demonstrates its effectiveness and robustness in handling the migration task of crack detection between different types of building materials.

TABLE 4.4: Performance Comparison across Different Data Ratios on SDNET2018 Dataset

Data Ratio	Tasks					
	B→P	B→W	P→B	P→W	W→B	W→P
100%	0.9488	0.9691	0.9878	0.9823	0.9842	0.9566
80%	0.9338	0.9493	0.9890	0.9702	0.9780	0.9316
60%	0.9195	0.9248	0.9838	0.9545	0.9659	0.9090
40%	0.9176	0.9159	0.9600	0.9372	0.9581	0.9264
20%	0.9003	0.9038	0.9203	0.8859	0.9490	0.8942

B: Bridge, P: Pavement, W: Wall

As shown in Table 4.4, in all six migration scenarios (bridge to roadway, bridge to wall, roadway to bridge, roadway to wall, wall to bridge, and wall to roadway), the model maintained a relatively stable performance even when the amount of source domain data was reduced to 20%.

Specifically, when using the full amount of source domain data (100%), the target domain test accuracies for all migration scenarios exceeded 95%, with the most significant effect for pavement-to-bridge migration, which achieved a test accuracy of 98.79%. As the amount of source domain data decreased (80%, 60%, 40%, and 20%), the model performance exhibited a flat decline, but the decline was controlled within an acceptable range. It is particularly noteworthy that even in the extreme case of using only 20% of the source domain data, the accuracy of the migration scenarios can still be maintained above 90%, except for the relatively challenging wall-to-pavement (89.42%) and pavement-to-wall (88.59%) scenarios. This result indicates that the method proposed in this study has good data efficiency and can realize effective domain adaptation under the condition of

limited source domain data. In addition, the best model during training usually appeared within the first 50 epochs, indicating that the model had a fast convergence rate. The performance metrics (accuracy and F1 score) on the training set are generally close to 100%, whereas the performance on the test set only slightly decreases, indicating that the model has good generalization ability and does not show obvious overfitting.

4.5 compared dataset

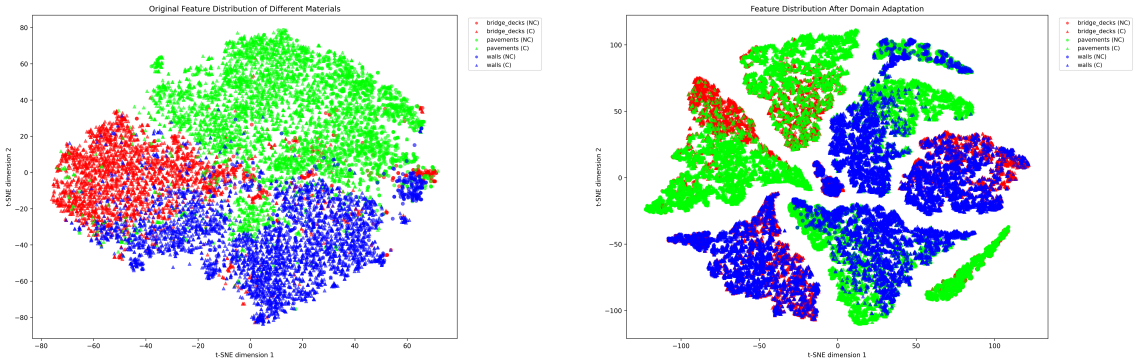
TABLE 4.5: Performance Comparison across Different Domain Adaptation Tasks on Office-31 Dataset

Source Domain	Target Domain	Best Target Accuracy
amazon	webcam	0.8931
amazon	dslr	0.8594
webcam	amazon	0.7820
webcam	dslr	0.9056
dslr	amazon	0.7696
dslr	webcam	0.8792

The table shows the best target accuracies for each domain adaptation task.

The Office-31 dataset is a classic benchmark for domain adaptation tasks and is widely used in research on cross-domain transfer learning algorithms. It consists of three distinct domains: Amazon (A), Webcam (W), and DSLR (D). Amazon contains product images with simple backgrounds, Webcam features images captured by webcams with complex backgrounds, and DSLR includes high-resolution images taken by professional DSLR cameras. The dataset covers 31 categories of images, with significant variations in data distribution across domains. In typical transfer tasks, models are required to transfer from one domain (source domain) to another (target domain), such as Amazon $\rightarrow$ Webcam (A $\rightarrow$ W) and Webcam $\rightarrow$ DSLR (W $\rightarrow$ D). Experimental results show significant differences in transfer performance across domains. For instance, in the A $\rightarrow$ W task, the best target accuracy is 89.31%, while in the D $\rightarrow$ A task, the best target accuracy is only 76.96%. Moreover, the W $\rightarrow$ D transfer task achieves the best performance with an accuracy of 90.56%. These results highlight the impact of domain data distribution differences on transfer learning performance. The Office-31 dataset provides a standardized benchmark for the development and evaluation of transfer learning algorithms.

4.5.1 Feature Distribution Analysis



(A) Original feature distribution

(B) Feature distribution after domain adaptation

FIGURE 4.3: t-SNE visualization of feature distributions before and after domain adaptation. Different colors and markers represent different material types and crack conditions (NC: Non-Cracked, C: Cracked).

We employ t-SNE visualization to demonstrate the effectiveness of our domain adaptation approach, with results presented in Fig.4.3. The visual comparison between pre- and post-adaptation feature spaces offers qualitative validation of our methodology.

The initial feature representation (Fig.4.3(a)) reveals considerable feature entanglement across material types and crack classifications, especially evident between bridge decks and wall surfaces, suggesting shared visual properties. Pavement features exhibit relative clustering but demonstrate classification ambiguity within crack categories. Such feature overlap underscores the complexity of cross-domain transfer and accounts for suboptimal baseline performance when models are directly transferred across material domains.

Following domain adaptation (Fig. 4.3(b)), the feature landscape shows marked organizational improvements consistent with our quantitative findings. Material-specific features achieve enhanced clustering with preserved crack/non-crack discriminability within each domain. This structural refinement confirms successful material-aware feature learning while maintaining detection capability. Interestingly, intra-material sub-clustering emerges, likely representing distinct crack morphologies or surface variations, indicating preserved domain-specific nuances during adaptation. The spatial arrangement reflects material relationships, positioning similar surfaces (bridge decks, pavements) in proximity while distancing distinct materials (walls).

The achieved feature separation validates our quantitative performance gains while demonstrating that our hierarchical adaptation preserves fine-grained discriminative information essential for reliable crack identification.

4.6 Discussion

Based on the analysis of the above experimental results, the hierarchical domain adaptation method proposed in this paper demonstrates significant performance advantages in the crack-detection migration task. This advantage mainly stems from three key designs: the hierarchical feature extraction strategy effectively captures the multi-scale features of the cracks, enabling the model to focus on both the local details and global structural information of the cracks; the improved domain-adversarial training mechanism effectively mitigates the differences in the feature distribution between the source and target domains; and the adaptive feature fusion mechanism dynamically adjusts the importance of the features according to different migration scenarios. The synergistic effect of these designs enables the model to maintain stable and high performance in various migration scenarios.

The experimental results highlight the excellent data-efficiency characteristics of this method. Even when the amount of data in the source domain drops to 20%, most migration scenarios can still maintain more than 90% accuracy with relatively flat performance decay curves. This high data efficiency characteristic is of great significance for practical applications, as it can significantly reduce the cost of data acquisition and labeling. At the same time, this method can effectively utilize limited labeled data to extract key domain-invariant features, which provides an effective solution to the data scarcity problem often encountered in practical engineering applications.

The experimental results revealed some interesting phenomena in terms of the performance differences between different migration scenarios. Road-to-bridge and wall-to-bridge migrations are the best, whereas wall-to-pavement migration is relatively poor. This difference may stem from the degree of variation in the surface properties of the materials: the surface texture characteristics of the bridge and roadway are relatively close to each

other, while the surface characteristics of the wall and roadway are more different. In addition, differences in environmental factors (e.g., lighting conditions and shooting angle) and crack morphology characteristics across scenes may also affect the migration effect. These findings provide an important reference for the further optimization of migration strategies.

It is worth exploring in depth that the success of this study is largely attributed to the inherent advantages of the CLIP model as a feature extractor, which provides a novel perspective for addressing Source-Free Domain Adaptation (SFDA) problems. Traditional domain adaptation methods often rely on aligning the underlying feature distributions between source and target domains, but under the SFDA setting, the absence of source domain data makes this approach challenging. Through its pre-training on billion-scale image-text pairs, CLIP has learned a highly generalizable visual feature space that is well-aligned with human semantics. This powerful prior knowledge implies that its feature representations inherently exhibit strong robustness to domain variations. More critically, CLIP's core advantage lies in its cross-modal alignment capability, which allows us to utilize text prompts as "semantic anchors" to guide the model in generating reliable pseudo-labels in the target domain. This achieves a paradigm shift from "distribution alignment" to "semantic alignment," enabling the model to perform self-calibration and optimization in the target domain solely through high-level semantic concepts (such as "crack" or "non-crack") without requiring access to source domain data. Therefore, CLIP's success lies not only in its role as a powerful feature extractor, but more importantly, in providing a novel, semantically-driven solution for SFDA problems.

Beyond academic performance, deploying this model in real-world scenarios requires careful consideration of fairness, reliability, and safety concerns. Regarding fairness, the model's performance heavily depends on the diversity of training data—if the training set lacks representation of certain surface materials or environmental conditions, the model may exhibit bias when encountering these scenarios. To enhance reliability in practical applications, we can leverage the confidence scores provided by the model: when the model lacks sufficient confidence in a prediction, the system should automatically flag it for human review, thereby ensuring overall system stability.

Despite the remarkable results of this method, there are still some limitations that need to be addressed in future work. First, hierarchical feature extraction and multi-scale fusion mechanisms enhance the performance but also increase the computational complexity of the model. Second, the relatively low performance in some specific migration scenarios (e.g., wall-to-pavement) suggests that the method still has room for improvement when dealing with widely varying domain migration tasks.

4.7 Summary

In this study, a CLIP-based passive domain adaptation method is proposed to solve the domain migration problem in concrete crack detection. By designing a CLIP-based pseudo-label generation strategy, cross-modal feature alignment was successfully achieved, which effectively solved the dependence of traditional methods on source domain data. The experimental results show that the method achieves more than 95% accuracy in the migration task between six different material types, which is significantly better than the existing methods. Meanwhile, the proposed three-layer feature fusion network achieves comprehensive feature alignment through loss-function design at three levels: visual alignment, structural consistency, and semantic alignment, which enables the model to demonstrate strong migration capabilities among different material types. Notably, the proposed method exhibits excellent data efficiency, maintaining more than 90% accuracy in most migration scenarios even when the source domain data is only 20%, which is an important feature for data collection and annotation cost control in practical engineering applications. This study provides a new solution to the domain adaptation problem of concrete crack detection, which not only makes a breakthrough in performance but also has obvious advantages in practicality. Future research will focus on several directions. Firstly, we will explore optimizing the model structure to reduce computational complexity. Secondly, we will investigate more effective feature alignment strategies to improve performance in cases of extreme domain differences. Furthermore, a highly promising direction is to integrate our SFDA framework with Few-Shot Learning. While our method performs effectively with unlabeled target data, incorporating a very small number of high-quality labeled target samples (e.g., 1-5 shots) for fine-tuning could further calibrate and refine the pseudo-labels

generated by CLIP. This could lead to enhanced accuracy and robustness, particularly in the most challenging adaptation scenarios. Finally, we aim to extend the present method to more types of structural defect detection tasks to promote the practical application of deep learning technology in the field of engineering structural health monitoring.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presents two innovative research projects that provide comprehensive and systematic solutions to the domain adaptation challenges in concrete crack detection. The DINOv2-based method and CLIP-based source-free domain adaptation approach address key challenges in practical applications from different perspectives, forming a complementary and mutually reinforcing technical framework. Specifically, our major contributions include: developing a novel DINOv2-based feature extraction method that effectively captures and characterizes crack features across different material surfaces, providing a solid technical foundation for cross-domain crack detection; proposing a CLIP-based source-free domain adaptation framework that completely eliminates the dependency on source domain data during adaptation, significantly improving the practicality and scalability of the method; and introducing an innovative moderate core set selection strategy that significantly enhances pseudo-label quality and adaptation efficiency through intelligent sample selection and optimization. Extensive experimental results thoroughly demonstrate the effectiveness and superiority of our proposed methods across various complex scenarios, achieving substantial improvements in cross-domain crack detection performance compared to existing state-of-the-art approaches. Our methods not only maintain stable high accuracy in the target domain but also demonstrate remarkable robustness when dealing

with different types of domain shifts and environmental variations, making them particularly valuable and promising for practical engineering applications.

5.2 Future Work

Although this research has achieved significant results in domain adaptation tasks for cross crack detection, there are still several important directions worthy of further exploration and improvement. First and foremost, a key area for future investigation is the integration of Few-Shot Learning (FSL) techniques. As mentioned in the introduction, FSL is critical for scenarios where a small number of labeled examples from the target domain are available. While this thesis focused on source-free and self-supervised methods, FSL presents a complementary approach. Future work could explore hybrid models that use the source-free adapted model as a strong starting point, which is then fine-tuned using few-shot techniques to achieve even higher accuracy and robustness with minimal labeling cost. The primary task is to optimize the existing model structure through in-depth research on lightweight network architectures and advanced model compression techniques to reduce computational complexity and achieve efficient model deployment on various edge devices. This includes developing more efficient model quantization strategies and exploring hardware-specific acceleration methods for different platforms to further enhance the practicality and universality of our approach. Additionally, to better address performance issues in scenarios with extreme domain differences, more advanced and robust feature alignment strategies need to be developed, including designing adaptive feature weight dynamic adjustment mechanisms and exploring innovative multimodal information fusion methods to enhance model robustness and adaptability in complex and varying environments. The incorporation of uncertainty estimation mechanisms in the feature alignment process and the development of more sophisticated and efficient domain-invariant feature learning approaches are also key areas that require focused attention in future research. Meanwhile, we recognize that in practical applications, model interpretability and traceability are equally important, necessitating the development of more transparent and understandable feature extraction and decision-making mechanisms.

Furthermore, future work will deeply explore the use of advanced generative models (such as Stable Diffusion combined with ControlNet technology) to generate high-quality synthetic crack images. By fully utilizing these advanced generative models, we can create more diverse and realistic crack image datasets that accurately mimic the characteristics and variations of different material domains, thereby significantly enhancing the overall performance of domain adaptation algorithms. This generative model-based approach aims to enhance model generalization ability and adaptability by systematically enriching training data and significantly reducing dependence on real-world labeled datasets. In-depth development of domain-specific data augmentation strategies and systematic research into the optimal balance between real and synthetic data during training will play a crucial role in maximizing the practical benefits of this approach. Meanwhile, we also plan to explore more diverse data generation strategies, including incorporating physical model constraints to ensure that generated images are not only visually realistic but also conform to the physical characteristics and formation mechanisms of actual cracks. These future research directions are interconnected and mutually supportive, collectively dedicated to further improving the practicality, reliability, and overall performance of our domain adaptation framework, ultimately providing more reliable, efficient, and universally applicable crack detection systems and solutions for various industrial application scenarios. Additionally, we will also focus on developing more comprehensive evaluation metrics and testing methods to more thoroughly measure model performance across different application scenarios, and explore the possibility of extending our methods to other types of defect detection tasks.

Bibliography

[1] Hadi Salehi, Rigoberto Burgueño, Shantanu Chakrabartty, Nizar Lajnef, and Amir H Alavi. "A comprehensive review of self-powered sensors in civil infrastructure: State-of-the-art and future research trends". In: *Engineering Structures* 234 (2021), p. 111963.

[2] Younes Hamishebahar, Hong Guan, Stephen So, and Jun Jo. "A comprehensive review of deep learning-based crack detection approaches". In: *Applied Sciences* 12.3 (2022), p. 1374.

[3] Saeed Mirza. "Durability and sustainability of infrastructure—A state-of-the-art report". In: *Canadian journal of civil engineering* 33.6 (2006), pp. 639–649.

[4] KC Laxman, Nishat Tabassum, Li Ai, Casey Cole, and Paul Ziehl. "Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning". In: *Construction and Building Materials* 370 (2023), p. 130709.

[5] Vaughn Peter Golding, Zahra Gharineiat, Hafiz Suliman Munawar, and Fahim Ullah. "Crack detection in concrete structures using deep learning". In: *Sustainability* 14.13 (2022), p. 8117.

[6] Tarutal Ghosh Mondal and Mohammad Reza Jahanshahi. "Applications of computer vision-based structural health monitoring and condition assessment in future smart cities". In: *The rise of smart cities* (2022), pp. 193–221.

[7] Wenjun Wang and Chao Su. "Deep learning-based real-time crack segmentation for pavement images". In: *KSCE Journal of Civil Engineering* 25.12 (2021), pp. 4495–4506.

[8] Jianghua Deng, Amardeep Singh, Yiyi Zhou, Ye Lu, and Vincent Cheng-Siong Lee. "Review on computer vision-based crack detection and quantification methodologies for civil structures". In: *Construction and Building Materials* 356 (2022), p. 129238.

[9] Shrikant M Harle. "Advancements and challenges in the application of artificial intelligence in civil engineering: a comprehensive review". In: *Asian Journal of Civil Engineering* 25.1 (2024), pp. 1061–1078.

[10] Gabriela Csurka. "Domain adaptation for visual applications: A comprehensive survey". In: *arXiv preprint arXiv:1702.05374* (2017).

[11] Young-Jin Cha, Rahmat Ali, John Lewis, and Oral Büyükztürk. "Deep learning-based structural health monitoring". In: *Automation in Construction* 161 (2024), p. 105328.

[12] Jun Bai, Di Wu, Tristan Shelley, Peter Schubel, David Twine, John Russell, Xuesen Zeng, and Ji Zhang. "A comprehensive survey on machine learning driven material defect detection: Challenges, solutions, and future prospects". In: *arXiv preprint arXiv:2406.07880* (2024).

[13] Yuqi Fang, Pew-Thian Yap, Weili Lin, Hongtu Zhu, and Mingxia Liu. "Source-free unsupervised domain adaptation: A survey". In: *Neural Networks* (2024), p. 106230.

[14] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. "A comprehensive survey on source-free domain adaptation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[15] Song Tang, Yuji Shi, Zihao Song, Mao Ye, Changshui Zhang, and Jianwei Zhang. "Progressive source-aware transformer for generalized source-free domain adaptation". In: *IEEE Transactions on Multimedia* 26 (2023), pp. 4138–4152.

[16] Youngeun Kim, Donghyeon Cho, and Sungeun Hong. "Towards privacy-preserving domain adaptation". In: *IEEE Signal Processing Letters* 27 (2020), pp. 1675–1679.

[17] Xinyang Huang, Chuang Zhu, Bowen Zhang, and Shanghang Zhang. "Learning from Different Samples: A Source-free Framework for Semi-supervised Domain Adaptation". In: *arXiv preprint arXiv:2411.06665* (2024).

[18] Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth. "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure". In: *Advanced engineering informatics* 29.2 (2015), pp. 196–210.

[19] Xincong Yang, Heng Li, Yantao Yu, Xiaochun Luo, Ting Huang, and Xu Yang. "Automatic pixel-level crack detection and measurement using fully convolutional network". In: *Computer-Aided Civil and Infrastructure Engineering* 33.12 (2018), pp. 1090–1109.

[20] Qi Yuan, Yufeng Shi, and Mingyue Li. "A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges". In: *Remote Sensing* 16.16 (2024), p. 2910.

[21] Qiyuan An, Ruijiang Li, Lin Gu, Hao Zhang, Qingyu Chen, Zhiyong Lu, Fei Wang, and Yingying Zhu. "A privacy-preserving unsupervised domain adaptation framework for clinical text analysis". In: *arXiv preprint arXiv:2201.07317* (2022).

[22] Bruno Oliveira Santos, Jónatas Valença, João P Costeira, and Eduardo Julio. "Domain adversarial training for classification of cracking in images of concrete surfaces". In: *AI in Civil Engineering* 1.1 (2022), p. 8.

[23] Shamendra Egodawela, Amirali Khodadadian Gostar, HAD Samith Buddika, AJ Dammika, Nalin Harischandra, Satheeskumar Navaratnam, and Mojtaba Mahmoodian. "A deep learning approach for surface crack classification and segmentation in unmanned aerial vehicle assisted infrastructure inspections". In: *Sensors* 24.6 (2024), p. 1936.

[24] Mohammad R Jahanshahi, Sami F Masri, Curtis W Padgett, and Gaurav S Sukhatme. "An innovative methodology for detection and quantification of cracks through incorporation of depth perception". In: *Machine vision and applications* 24 (2013), pp. 227–241.

[25] Hyunjun Kim, Eunjong Ahn, Myoungsu Shin, and Sung-Han Sim. "Crack and noncrack classification from concrete surface images using machine learning". In: *Structural Health Monitoring* 18.3 (2019), pp. 725–738.

[26] NTS Board. "Collapse of I-35W Highway Bridge Minneapolis". In: *Minnesota August* 1 (2007).

[27] Samir Mustapha, Ali Braytee, and Lin Ye. "Multisource data fusion for classification of surface cracks in steel pipes". In: *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems* 1.2 (2018), pp. 021007–021007.

[28] R Zaurin and FN Catbas. "Integration of computer imaging and sensor data for structural health monitoring of bridges". In: *Smart Materials and Structures* 19.1 (2009), p. 015019.

[29] Nana Li, Xiangdan Hou, Xinyu Yang, and Yongfeng Dong. "Automation recognition of pavement surface distress based on support vector machine". In: *2009 Second International Conference on Intelligent Networks and Intelligent Systems*. IEEE. 2009, pp. 346–349.

[30] Mohamed S Kaseko and Stephen G Ritchie. "A neural network-based methodology for pavement crack detection and classification". In: *Transportation Research Part C: Emerging Technologies* 1.4 (1993), pp. 275–291.

[31] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. "Automatic road crack detection using random structured forests". In: *IEEE Transactions on Intelligent Transportation Systems* 17.12 (2016), pp. 3434–3445.

[32] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. "Road crack detection using deep convolutional neural network". In: *2016 IEEE international conference on image processing (ICIP)*. IEEE. 2016, pp. 3708–3712.

[33] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. "Deep learning-based crack damage detection using convolutional neural networks". In: *Computer-Aided Civil and Infrastructure Engineering* 32.5 (2017), pp. 361–378.

[34] Byunghyun Kim and Soojin Cho. "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique". In: *Sensors* 18.10 (2018), p. 3452.

[35] Huijun Liu, Chunhua Yang, Ao Li, Sheng Huang, Xin Feng, Zhimin Ruan, and Yongxin Ge. "Deep domain adaptation for pavement crack detection". In: *IEEE Transactions on Intelligent Transportation Systems* 24.2 (2022), pp. 1669–1681.

[36] Keunyoung Jang, Namgyu Kim, and Yun-Kyu An. "Deep learning–based autonomous concrete crack evaluation through hybrid image scanning". In: *Structural Health Monitoring* 18.5-6 (2019), pp. 1722–1737.

[37] Yihuan Zhu, Sheng Zhang, and Chengfeng Ruan. "CCN: Pavement Crack Detection with Context Contrasted Net". In: *International Conference on Neural Information Processing*. Springer. 2022, pp. 85–96.

[38] Wenjun Wang and Chao Su. "Automatic classification of reinforced concrete bridge defects using the hybrid network". In: *Arabian Journal for Science and Engineering* 47.4 (2022), pp. 5187–5197.

[39] Guanting Ye, Wei Dai, Jintai Tao, Jinsheng Qu, Lin Zhu, and Qiang Jin. "An improved transformer-based concrete crack classification method". In: *Scientific Reports* 14.1 (2024), p. 6226.

[40] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.

[41] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[42] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[43] Zijie Lin, Hui Wang, and Shenglin Li. "Pavement anomaly detection based on transformer and self-supervised learning". In: *Automation in Construction* 143 (2022), p. 104544.

[44] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014).

[45] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. "A kernel two-sample test". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 723–773. ISSN: 1532-4435.

[46] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR, pp. 97–105.

[47] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. "Domain-adversarial training of neural networks". In: *Journal of machine learning research* 17.59 (2016), pp. 1–35.

[48] Baochen Sun and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation". In: *Computer vision–ECCV 2016 workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, proceedings, part III 14*. Springer. 2016, pp. 443–450.

[49] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. "Deep transfer learning with joint adaptation networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2208–2217.

[50] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[51] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3723–3732.

[52] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[53] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. "Cycada: Cycle-consistent adversarial domain adaptation". In: *International conference on machine learning*. Pmlr. 2018, pp. 1989–1998.

[54] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. "A dirt-t approach to unsupervised domain adaptation". In: *arXiv preprint arXiv:1802.08735* (2018).

[55] Sen Wu, Hongyang R Zhang, and Christopher Ré. "Understanding and improving information transfer in multi-task learning". In: *arXiv preprint arXiv:2005.00944* (2020).

[56] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. "Deep transfer learning with joint adaptation networks". In: *International conference on machine learning*. PMLR, pp. 2208–2217. ISBN: 2640-3498.

[57] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732.

[58] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[59] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. "Transferable adversarial training: A general approach to adapting deep classifiers". In: *International conference on machine learning*. PMLR, pp. 4013–4022. ISBN: 2640-3498.

[60] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. "Multi-adversarial domain adaptation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. ISBN: 2374-3468.

[61] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. "Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 16643–16653.

[62] Ganlong Zhao, Guanbin Li, Ruijia Xu, and Liang Lin. "Collaborative training between region proposal localization and classification for domain adaptive object detection". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer. 2020, pp. 86–102.

[63] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. "Contrastive adaptation network for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4893–4902.

[64] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. "Adversarial multiple source domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[65] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1406–1415.

[66] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. "Progressive feature alignment for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 627–636.

[67] Yu Mitsuzumi, Akisato Kimura, and Hisashi Kashima. "Understanding and Improving Source-free Domain Adaptation from a Theoretical Perspective". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28515–28524.

[68] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. "Source-free domain adaptation via distribution estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7212–7222.

[69] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. "Generalized source-free domain adaptation". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8978–8987.

[70] Song Tang, Wenxin Su, Mao Ye, and Xiatian Zhu. "Source-Free Domain Adaptation with Frozen Multimodal Foundation Model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23711–23720.

[71] Shivang Chopra, Suraj Kothawade, Houda Aynaou, and Aman Chadha. "Source-Free Domain Adaptation with Diffusion-Guided Source Data Generation". In: *arXiv preprint arXiv:2402.04929* (2024).

[72] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. "Multimodal foundation models: From specialists to general-purpose assistants". In: *Foundations and Trends® in Computer Graphics and Vision* 16.1-2 (2024), pp. 1–214.

[73] Shivang Chopra, Suraj Kothawade, Houda Aynaou, and Aman Chadha. "Source-free domain adaptation with diffusion-guided source data generation". In: *arXiv preprint arXiv:2402.04929* (2024).

[74] Jing Wang, Wonho Bae, Jiahong Chen, and Junhyug Noh. "Neighborhood-Informed Diffusion Model for Source-Free Domain Adaptation: Retrieving Source Ground Truth from Target Query's Neighbors". In: ().

[75] Tianhao Xiao, Rong Pang, Huijun Liu, Chunhua Yang, Ao Li, Chenxu Niu, Zhimin Ruan, Ling Xu, and Yongxin Ge. "Domain adaptation and knowledge distillation for lightweight pavement crack detection". In: *Expert Systems with Applications* 263 (2025), p. 125734.

[76] Daniel Asefa Beyene, Michael Bekele Maru, Taeheon Kim, Solmoi Park, Seunghee Park, et al. "Unsupervised domain adaptation-based crack segmentation using transformer network". In: *Journal of Building Engineering* 80 (2023), p. 107889.

[77] Pang-jo Chun and Toshiya Kikuta. "Self-training with Bayesian neural networks and spatial priors for unsupervised domain adaptation in crack segmentation". In: *Computer-Aided Civil and Infrastructure Engineering* 39.17 (2024), pp. 2642–2661.

[78] Hessam Kaveh and Reda Alhajj. "Recent advances in crack detection technologies for structures: a survey of 2022-2023 literature". In: *Frontiers in Built Environment* 10 (2024), p. 1321634.

[79] Jamie Padgett, Reginald DesRoches, Bryant Nielson, Mark Yashinsky, Oh-Sung Kwon, Nick Burdette, and Ed Tavera. "Bridge damage and repair costs from Hurricane Katrina". In: *Journal of Bridge Engineering* 13.1 (2008), pp. 6–14.

[80] Jian-Hao Hong, Yee-Meng Chiew, Jau-Yau Lu, Jihn-Sung Lai, and Yung-Bin Lin. "Houfeng bridge failure in Taiwan". In: *Journal of Hydraulic Engineering* 138.2 (2012), pp. 186–198.

[81] HM Salem and Huda M Helmy. "Numerical investigation of collapse of the Minnesota I-35W bridge". In: *Engineering Structures* 59 (2014), pp. 635–645.

[82] FuTao Ni, Jian Zhang, and ZhiQiang Chen. "Pixel-level crack delineation in images with convolutional feature fusion". In: *Structural Control and Health Monitoring* 26.1 (2019), e2286.

[83] Yuequan Bao, Jian Li, Tomonori Nagayama, Yang Xu, Billie F Spencer Jr, and Hui Li. "The 1st international project competition for structural health monitoring (IPC-SHM, 2020): A summary and benchmark problem". In: *Structural Health Monitoring* 20.4 (2021), pp. 2229–2239.

[84] Elyas Asadi Shamsabadi, Chang Xu, Aravinda S Rao, Tuan Nguyen, Tuan Ngo, and Daniel Dias-da-Costa. "Vision transformer-based autonomous crack detection on asphalt and concrete surfaces". In: *Automation in Construction* 140 (2022), p. 104316.

[85] Ali Braytee, Mohamad Naji, and Paul J Kennedy. "Unsupervised domain-adaptation-based tensor feature learning with structure preservation". In: *IEEE Transactions on Artificial Intelligence* 3.3 (2022), pp. 370–380.

[86] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[87] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. "ibot: Image bert pre-training with online tokenizer". In: *arXiv preprint arXiv:2111.07832* (2021).

[88] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[89] Ç F Özgenel and A Gönenç Sorguç. "Performance comparison of pretrained convolutional neural networks on crack detection in buildings". In: *Isarc. proceedings of the international symposium on automation and robotics in construction*. Vol. 35. IAARC Publications. 2018, pp. 1–8.

[90] Hongyan Xu, Xiu Su, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. "Automatic bridge crack detection using a convolutional neural network". In: *Applied Sciences* 9.14 (2019), p. 2867.

[91] Marc Maguire, Sattar Dorafshan, and Robert J Thomas. "SDNET2018: A concrete crack image dataset for machine learning applications". In: (2018).

[92] Esraa Elhariri, Nashwa El-Bendary, and Shereen A Taie. "Historical-crack18-19: A dataset of annotated images for non-invasive surface crack detection in historical buildings". In: *Data in Brief* 41 (2022), p. 107865.

[93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[94] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[95] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[96] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[97] Yuzhe Yang and Zhi Xu. "Rethinking the value of labels for improving class-imbalanced learning". In: *Advances in neural information processing systems* 33 (2020), pp. 19290–19301.

[98] Jinghua Zhang, Lisha Peng, Shuzhi Wen, and Songling Huang. "A Review on Concrete Structural Properties and Damage Evolution Monitoring Techniques". In: *Sensors* 24.2 (2024), p. 620. ISSN: 1424-8220.

[99] Qi Yuan, Yufeng Shi, and Mingyue Li. "A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges". In: *Remote Sensing* 16.16 (2024), p. 2910. ISSN: 2072-4292.

[100] Quang Du Nguyen, Huu-Tai Thai, and Son Dong Nguyen. "Self-training method for structural crack detection using image blending-based domain mixing and mutual learning". In: *Automation in Construction* 170 (2025), p. 105892. ISSN: 0926-5805.

[101] Zhen Yao, Jiawei Xu, Shuhang Hou, and Mooi Choo Chuah. "Cracknex: a few-shot low-light crack segmentation model based on retinex theory for uav inspections". In: *arXiv preprint arXiv:2403.03063* (2024).

[102] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. "A theory of learning from different domains". In: *Machine learning* 79 (2010), pp. 151–175. ISSN: 0885-6125.

[103] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. "A brief review of domain adaptation". In: *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020* (2021), pp. 877–894. ISSN: 3030717038.

[104] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR, pp. 8748–8763. ISBN: 2640-3498.

[105] Ruoyu Feng, Tao Yu, Xin Jin, Xiaoyuan Yu, Lei Xiao, and Zhibo Chen. "Rethinking Domain Adaptation and Generalization in the ERA Of Clip". In: *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 2585–2591. ISBN: 9798350349399.

[106] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PmLR, pp. 8748–8763. ISBN: 2640-3498.

[107] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. "Moderate coreset: A universal method of data selection for real-world data-efficient deep learning". In: *The Eleventh International Conference on Learning Representations*.

[108] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. "Deeper, broader and artier domain generalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550.

[109] Baochen Sun and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation". In: *Computer vision–ECCV 2016 workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, proceedings, part III 14*. Springer, pp. 443–450. ISBN: 3319494082.

[110] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR, pp. 97–105.

[111] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

[112] Dosovitskiy Alexey. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv: 2010.11929* (2020).

[113] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. "Domain-adversarial training of neural networks". In: *Journal of machine learning research* 17.59 (2016), pp. 1–35. ISSN: 1533-7928.

[114] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).