mathematics

MDPI

Article

Auxcoformer: Auxiliary and Contrastive Transformer for Robust Crack Detection in Adverse Weather Conditions

Jae Hyun Yoon, Jong Won Jung and Seok Bong Yoo *

Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, Republic of Korea; jhyoon964@jnu.ac.kr (J.H.Y.); wawoo95@jnu.ac.kr (J.W.J.)
* Correspondence: sbyoo@jnu.ac.kr; Tel.: +82-625303437

Abstract: Crack detection is integral in civil infrastructure maintenance, with automated robots for detailed inspections and repairs becoming increasingly common. Ensuring fast and accurate crack detection for autonomous vehicles is crucial for safe road navigation. In these fields, existing detection models demonstrate impressive performance. However, they are primarily optimized for clear weather and struggle with occlusions and brightness variations in adverse weather conditions. These problems affect automated robots and autonomous vehicle navigation that must operate reliably in diverse environmental conditions. To address this problem, we propose Auxcoformer, designed for robust crack detection in adverse weather conditions. Considering the image degradation caused by adverse weather conditions, Auxcoformer incorporates an auxiliary restoration network. This network efficiently restores damaged crack details, ensuring the primary detection network obtains better quality features. The proposed approach uses a non-local patch-based 3D transform technique, emphasizing the characteristics of cracks and making them more distinguishable. Considering the connectivity of cracks, we also introduce contrastive patch loss for precise localization. Then, we demonstrate the performance of Auxcoformer, comparing it with other detection models through experiments.

Keywords: auxiliary and contrastive transformer; 3D discrete cosine transform; crack detection; adverse weather conditions; contrastive patch loss; robust representation

MSC: 68T45

Citation: Yoon, J.H.; Jung, J.W.; Yoo, S.B. Auxcoformer: Auxiliary and Contrastive Transformer for Robust Crack Detection in Adverse Weather Conditions. *Mathematics* **2024**, *12*, 690. https://doi.org/10.3390/math12050690

Academic Editors: Guangwei Gao, Juncheng Li and Zhi Li

Received: 6 February 2024
Revised: 23 February 2024
Accepted: 24 February 2024
Published: 27 February 2024

1. Introduction

As civil infrastructure continues to age and suffer from poor construction, the role of automated robots in crack detection has become increasingly crucial. These cracks weaken the structural integrity of buildings, bridges, and roads and pose significant risks to public safety. Specifically, they affect vehicle wear and tear, carbon emissions, and even the rate of accidents, such as structural collapses. The risk is further heightened in areas prone to natural disasters, such as earthquakes, where even minor cracks can lead to catastrophic failures. Moreover, manual inspection for crack detection is labor-intensive and costly, and delays in identifying and addressing these cracks can lead to exponentially higher repair costs over time. Given these severe risks and financial implications, deploying automated robots, such as unmanned aerial vehicles (UAVs), for accurate and timely crack detection has become an essential strategy for enhancing public safety and preserving civil infrastructure.

While existing models [1–28] for crack detection have impressive performance, their effectiveness is primarily confined to favorable weather conditions. These models often fail to detect cracks accurately in adverse weather conditions, such as rain, snow, or fog, which are factors leading to low-quality media data. Figure 1 compares crack detection performance in adverse weather conditions using two state-of-the-art (SOTA) models, the CNN-based you only look once, version 8 (YOLOv8) [25] and the transformer-based

co-DETR [29]. Performance is evaluated using average precision (AP). According to the analysis, although the models perform well in crack detection on clean images, they miss or makes incorrect predictions when dealing with images affected by rain and snow. These adverse weather conditions can make the detection process more challenging. For example, snowflakes and rain streaks can lead to occlusion in visual sensors. These occlusions are particularly problematic in crack detection because they can disrupt the perceived connectivity of cracks. Furthermore, adverse weather conditions can cause variations in brightness, affecting the accuracy of vision-based detection models. Beyond the challenges in detection, such conditions can make cracks even more dangerous by weakening the surface material of roads and building exteriors. Consequently, while rapid and accurate detection through automated robots is crucial, existing methods often overlook adverse weather conditions. Specifically, transformer-based detection models [29–33] struggle with real-time performance, making them unsuitable for real-world applications.

Numerous adverse weather restoration models [34–45] exist to address these challenges in adverse weather conditions and have demonstrated remarkable performance. However, these models operate independently of detection models, leading to high computational costs and slow inference times. This sequential and independent operation becomes a bottleneck, especially in emergencies, where rapid response is crucial. Moreover, because these restoration models do not directly interact with the detection models, they cannot achieve optimal performance. Another significant limitation of existing restoration models is that they are not explicitly designed for crack detection. These models are generally designed to improve image quality under adverse weather conditions but do not contain terms or features optimized for identifying structural defects, such as cracks. Therefore, they may not perform effectively in the context of defect detection.

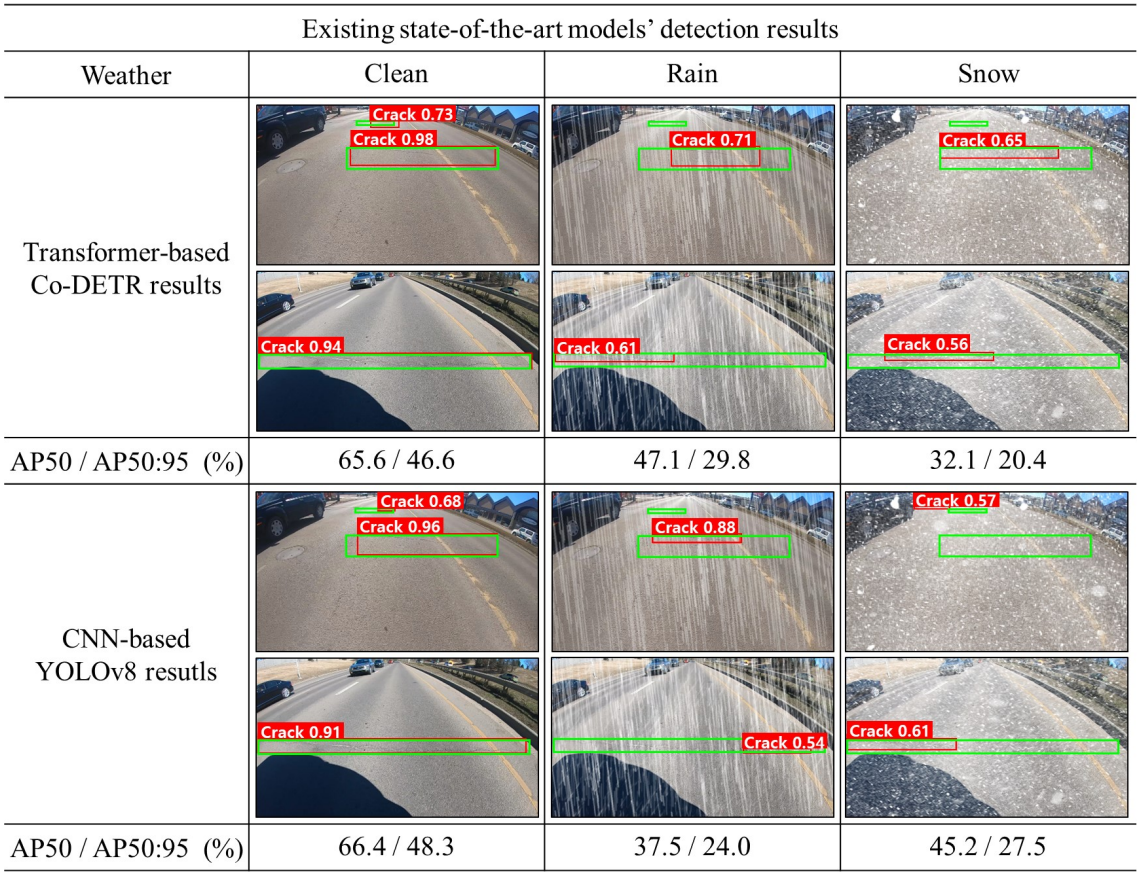

Figure 1. Comparative analysis of the crack detection performance in the existing state-of-the-art models in clean and adverse weather conditions. Predicted bounding boxes are red and ground truth boxes are green.

To overcome these problems, we propose Auxcoformer, an auxiliary and contrastive vision transformer designed to detect cracks in adverse weather conditions by efficiently incorporating an auxiliary restoration network into the crack detection model. Additionally, it uses a contrastive network that considers the inherent connectivity of cracks by comparing predicted patches with their adjacent patches, leading to more precise predictions. We summarize the contributions as follows:

- We propose a unified approach that combines a primary crack detection network with an auxiliary weather restoration network to leverage a robust representation of restored crack features. This integration performs efficient and robust crack detection using the synergy between restoration and detection tasks.
- We propose a 3D frequency augmentation (FA) block to optimize crack detection. This block uses non-local matching of adjacent and similar patches to perform a 3D discrete cosine transform (DCT) and selectively amplify crack-related frequency components while minimizing noise.
- We propose a contrastive patch loss function designed for precisely localizing cracks. This loss function evaluates the similarity between patches inside and adjacent to the bounding boxes to capture the inherent connectivity of cracks and improve detection accuracy.

2. Related Work

2.1. Crack Detection

Deep learning methods, such as Fast R-CNN [1], Faster R-CNN [2], and Cascade RCNN [3], are commonly used for crack detection. Various detection models [4–6] apply these methods. Pei et al. [7] applied the Cascade RCNN and introduced a consistency filtering mechanism, leveraging self-supervised learning to make the most of unlabeled data. Moreover, one-stage models [8–14] treat object detection as a regression task, directly predicting bounding boxes and categories across the image. YOLO-based detection models [15–28] have been studied for real-time detection. Yu et al. [27] proposed a multi-source domain adaptation model for crack detection that uses ensembled labels to realign source and target domains. Hong et al. [28] improved detection with the high-frequency characteristics of cracks by preserving the edges, which is advantageous for detecting cracks using the morphological characteristics of the area and connectivity. Zong et al. [29] proposed enhancing the learning ability of the encoder in end-to-end detectors using multiple parallel heads. These detection models have effective performance but are limited to clear weather conditions. Although some other models, such as [46], address adverse weather conditions, to our knowledge, specific solutions targeting crack detection under such conditions are scarce. Additionally, while transformer-based detection models [30–33] have emerged and shown excellent performance, these models have limitations in real-time applications. We propose a method to overcome these limitations through auxiliary learning.

2.2. Adverse Weather Restoration

There have been numerous attempts [36–41] to restore images degraded by adverse weather conditions through deep learning. Zamir et al. [42] introduced Restormer, an efficient transformer architecture designed for multi-scale local-global representation learning for image restoration tasks. Özdenizci et al. [43] proposed a patch-based image restoration method using denoising diffusion models, enabling patch-size-agnostic restoration through guided denoising and smoothed noise estimates. In addition, Valanarasu et al. [44] introduced TransWeather, an end-to-end transformer-based model that uses intra-patch transformer blocks to improve attention within patches, effectively removing minor weather degradations. Lee et al. [45] introduced a task-driven enhancement network linked to high-level vision tasks using dense block layer connections. Further, Kalwar et al. [34] introduced the GDIP block, that learns to reconstruct adverse weather images directly through downstream object detection loss. Xia et al. [35] proposed a training objective leveraging coarsely aligned image pairs. That training scheme led to better image translation quality

and improved downstream tasks. Although these models excel in restoration, their limited interaction with detection tasks poses a challenge. We address this by efficiently integrating the restoration task.

2.3. Auxiliary Learning

Auxiliary learning is a prevalent technique in deep learning. This learning strategy enhances primary task performance by integrating carefully designed auxiliary loss. Most existing studies [47–49] linearly merge the auxiliary loss with the primary loss and employ the combined loss for overall model optimization. The weights associated with auxiliary loss are adjusted to prevent any detrimental influence on the primary task. Recent endeavors [50–53] have introduced a dynamic approach to adjust the weights of auxiliary loss automatically during training. Specifically, Shi et al. [51] used a similar concept to Lin et al. [52], aiming to ensure that the weighted sum of gradients was close to the primary task gradient. Furthermore, Navon et al. [54] suggested learning a nonlinear fusion of auxiliary loss. In contrast, Chen et al. [55] proposed selecting tasks and individual data samples within each task to maximize auxiliary information utilization.

2.4. Contrastive Learning

Contrastive learning, a technique of learning through comparison, has made remarkable strides in self-supervised representation learning [56–61]. Recently, a trend has emerged where contrastive learning is harnessed to enhance self-supervised computer vision tasks [62–64]. Data augmentation generates positive and negative sample pairs for each anchor image in this approach. These pairs undergo contrastive learning to bring similar samples closer and dissimilar ones apart in the embedding space. However, traditional contrastive learning often ignores higher-level class semantics, relying solely on augmented image views. Khosla et al. [65] directly employed class labels to define similarity, labeling samples from the same class as positive and those from different classes as negative to overcome this. Drawing inspiration from the success of these techniques, we incorporate a contrastive learning mechanism into Auxcoformer by introducing a subnetwork.

3. Method

3.1. Overview

As depicted in Figure 2, we introduce Auxcoformer, designed for robust crack detection under adverse weather conditions. The shared CNN-based backbone takes an image degraded by adverse weather conditions as input and extracts features related to cracks and the surrounding background. These features are forwarded to the primary detection and auxiliary restoration networks. The auxiliary restoration network restores the degraded image and forwards the restored feature information to the primary detection network. This primary network incorporates a cross-attention mechanism with query (Q), key (K), and value (V), leveraging local representations from the CNN-based backbone and global restored representations from the auxiliary network. The network sequentially passes the features through multiple blocks and a fully connected (FC) layer to predict bounding box coordinates. Then, a separate contrastive net receives patches within and surrounding the predicted bounding box B_{pred} to compute the loss for more precise localization. Finally, all loss terms are aggregated and backpropagated, including the primary loss $\mathcal{L}_{pri}$, which consists of detection loss and contrastive patch loss $\mathcal{L}_{con}$, and the auxiliary loss $\mathcal{L}_{aux}$ for restoration.

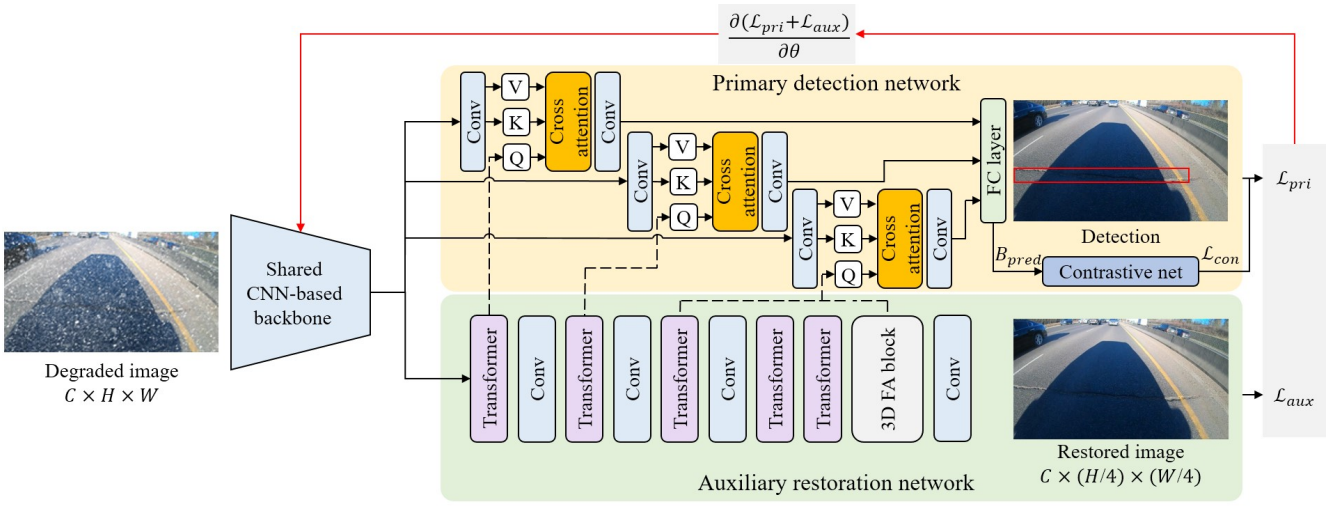

Figure 2. Overview of the proposed Auxcoformer network.

3.2. Auxiliary Restoration Network for Crack Detection

General crack detection models often fall short in real-world conditions due to their inability to account for weather variations. Existing weather restoration models, while helpful, operate independently and have high computational costs. To address these challenges, we introduce an auxiliary restoration network that improves the robustness of the primary detection task and reduces computational costs through synergistic interaction between restoration and detection.

The auxiliary restoration network receives features extracted from the shared CNN-based backbone and passes them through a transformer block and convolution layer. This structure aids in capturing the global features, enhancing the crack detection performance through restoration. Subsequently, the auxiliary restoration network employs a hierarchical structure comprising these transformer blocks and convolution layers. This architecture allows for a progressive upscaling of the resolution while gradually restoring critical information degraded due to adverse weather conditions.

Additionally, we introduce a 3D FA block, inspired by BM3D [66], within the auxiliary restoration network to emphasize the crack characteristics. The 3D FA block collects non-local patches across the feature map and enhances specific frequencies to accentuate the features relevant to crack detection. As illustrated in Figure 3, the first step involves reducing the channel dimensions of the input feature maps using a 3×3 convolution. This step simplifies the computational requirements and prepares the features for the subsequent patch-matching process. For patch matching, we use the Euclidean distance to find similar patches in the spatial domain, forming a set of eight non-local patches for each target patch in the restored feature maps. The distance between two patches P_i and P_j is calculated as follows:

$$dist(P_i, P_j) = \left\| P_i - P_j \right\|_2^2 , \tag{1}$$

where P_i denotes the i-th reference patch, and P_j indicates a j-th non-local patch among the total patches. A smaller distance indicates a higher similarity, and based on this metric we collect non-local patches that are most similar to each target patch. This approach allows us to effectively group patches containing cracks for further processing. These sets undergo a 3D transform to convert the features from the spatial domain to the frequency domain, as follows:

$$F(u,v,w) = \alpha(u)\beta(v)\gamma(w) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} f(x,y,z)\delta(x,y,z,u,v,w) , \tag{2}$$

$$\delta(x,y,z,u,v,w) = \cos\left(\frac{\pi u(2x+1)}{2N}\right)\cos\left(\frac{\pi v(2y+1)}{2M}\right)\cos\left(\frac{\pi w(2z+1)}{2L}\right), \quad (3)$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases}, \quad (4)$$

$$\beta(v) = \begin{cases} \sqrt{\frac{1}{M}}, & v = 0 \\ \sqrt{\frac{2}{M}}, & v \neq 0 \end{cases}, \quad (5)$$

$$\gamma(w) = \begin{cases} \sqrt{\frac{1}{L}}, & w = 0 \\ \sqrt{\frac{2}{L}}, & w \neq 0 \end{cases}, \quad (6)$$

where $F(u,v,w)$ represents the 3D DCT coefficient for the indices u, v, and w, and $f(x,y,z)$ denotes the pixel value for the indices x, y, and z. Moreover, $\delta(x,y,z,u,v,w)$ denotes the cosine basis function, and $\alpha(u)$, $\beta(v)$, and $\gamma(w)$ represent regularization constants. Additionally, N and M denote the height and width of the input feature, respectively, and L represents the number of non-local patches per set collected based on Equation (1).

Subsequently, FA is applied to selectively amplify the frequencies most relevant to identifying cracks. Inspired by BM3D, we assume that in such patch groups, the coefficients corresponding to noise will have lower values than those representing cracks. This formula for FA is defined as follows:

$$FA(F(u,v,w)) = \begin{cases} 2 \times F(u,v,w), & F(u,v,w) \geq \epsilon \text{ and } (u,v,w) \neq (0,0,0) \\ F(u,v,w), & otherwise \end{cases}, \quad (7)$$

where ϵ denotes the threshold value to limit the range of amplified DCT coefficients. By gathering these patches, we can emphasize similar information among them through collaborative filtering, excluding high-frequency noise components.

After FA, the transformed patches are converted back to the spatial domain using an inverse DCT (IDCT), as follows:

$$f(x,y,z) = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1}\sum_{w=0}^{L-1}\alpha(u)\beta(v)\gamma(w)F(u,v,w)\delta(x,y,z,u,v,w). \quad (8)$$

These IDCT patches are then aggregated to form a restored feature map, which is passed through another 3×3 convolution to bring the channel dimensions back to their original state. By incorporating these steps, the proposed method effectively emphasizes features crucial for crack detection while maintaining computational efficiency.

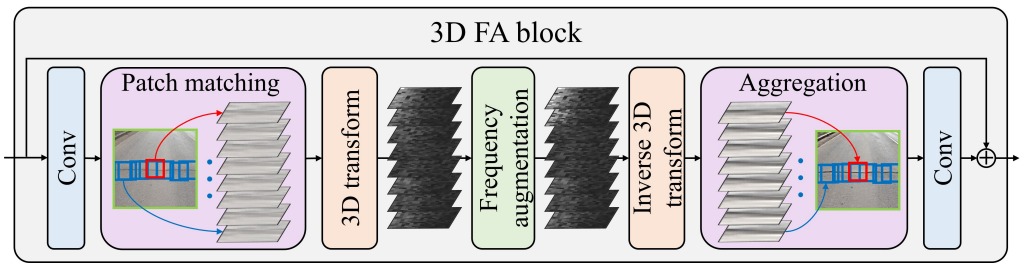

Figure 3. Architecture of the 3D frequency augmentation (FA) block.

After passing through a 3×3 convolution, the final output is a quarter of the size of the original image. This reduction is intended to decrease the computational complexity of the restoration process. For image restoration, the output image is compared with the

ground truth (GT) image which has been resized to quarter of its original size. We calculate the difference between these images using the $L1$-norm as follows:

$$\mathcal{L}_{L1} = \frac{1}{CH'W'} \left\| \hat{I} - I \right\|_1 , \tag{9}$$

where $\hat{I}$ denotes the output image of the auxiliary restoration network and I denotes the resized GT image. Additionally, C denotes the number of channels, and H' and W' represent the height and width, respectively, scaled down to a quarter of the GT image's dimensions. Moreover, we apply the perceptual loss [67] for comparing the feature maps between the GT and the output images as follows:

$$\mathcal{L}_p = \sum_{k \in \{3,9,15\}} \frac{1}{C_k H_k W_k} \left\| VGG16_k(\hat{I}) - VGG16_k(I) \right\|_1 , \tag{10}$$

where $VGG16_k(\cdot)$ represents the features extracted from the k-th layer of VGG16. Specifically, we utilize the 3rd, 9th, and 15th layers to compute the perceptual loss. Furthermore, C_k, H_k, and W_k denote the shape of the feature map at the k-th layer.

Using these losses, the auxiliary loss is defined as follows:

$$\mathcal{L}_{aux} = \lambda_{L1}\mathcal{L}_{L1} + \lambda_p \mathcal{L}_p , \tag{11}$$

where λ_{L1} and λ_p control the importance of loss. Through this auxiliary loss function, the auxiliary restoration network can assist in making Auxcoformer robust to weather conditions by jointly learning with it.

3.3. Primary Detection Network

We employ a cross-attention mechanism to facilitate effective information sharing between the two networks. The features obtained from the auxiliary restoration serve as the query, whereas those from the primary detection network act as the key and value. This approach strengthens the representation power of Auxcoformer and allows for a more detailed understanding of the scene, especially in adverse weather conditions. The procedure for cross-attention is specified as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V , \tag{12}$$

where Q denotes the query matrix generated by the auxiliary restoration network, K and V denote the key and value matrices, respectively, obtained from the primary detection network, and D is the channel dimension. Leveraging the cross-attention mechanism, we strategically fuse the local features extracted by the CNN-based backbone with the global contextual information captured by the transformer blocks. This fusion enables the model to leverage the complementary strengths of local and global features. In this way, Auxcoformer gains a more comprehensive understanding of the scene, which is crucial for robust detection performance in adverse weather conditions. This interaction between local and global features through cross-attention is crucial in enhancing the ability of the model to adapt and represent complex scenes. We applied this cross-attention at each hierarchical layer, processing the information and forwarding it to the FC layer. The FC layer merges this multi-level information to predict the bounding boxes.

3.4. Contrastive Patch Loss

Conventional object detection models often struggle with predicting bounding boxes that fully encompass irregularly shaped objects, such as cracks. This limitation is particularly problematic when the cracks extend beyond the predicted bounding box. To address this problem, we introduce the contrastive patch loss function in the proposed model, Auxcoformer. This loss function refines the localization of bounding boxes around cracks.

The contrastive patch loss is implemented using a pretrained contrastive network. This network applies contrastive learning to patches inside the predicted bounding boxes and their neighboring patches. The network is designed to manipulate the data distribution through similarity operations for improved localization. As indicated in Figure 4, the network narrows the distribution gap for positive pairs, including the prediction patch and its neighboring patches containing cracks. Conversely, it widens the gap for negative pairs, consisting of the prediction patch and its neighboring patches without cracks.

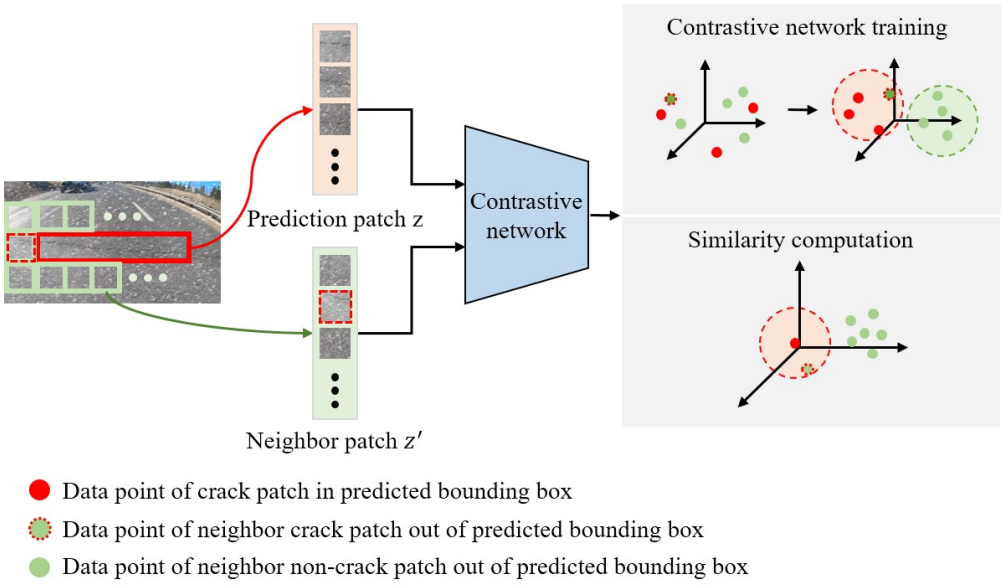

Figure 4. Illustration of the training process and loss function implementation in a contrastive network.

We use this pretrained network to implement the contrastive patch loss function in Auxcoformer. The loss function is defined as follows:

$$\mathcal{L}_{con} = -\log\left(-\frac{\exp(z_i^T z_i'/\tau)}{\sum_{j=1}^{S}\exp(z_i^T z_j'/\tau)} + 1\right), \tag{13}$$

where (z_i, z_i') denotes a positive pair of patches, z_i represents the patch within the predicted bounding box, and z_i' is a neighboring patch also containing a crack. In addition, $\{z_i, z_j'\}_{j=1,j\neq i}^{S}$ indicates negative pairs of patches, where z_j' denotes a neighboring patch that does not contain a crack, and τ denotes the temperature parameter (set to 0.1 as referred in [65]), controlling the scaling of similarities and making the model more sensitive to differences between patches. Finally, S is the total number of patches considered for each z_i.

This loss function allows the model to assign higher similarity scores and higher loss to cracks that extend beyond the initially predicted bounding box. Thus, we improve the precision of the bounding boxes and enhance the ability of the model to accurately localize and identify irregularly shaped and interconnected cracks. The detailed implementation is presented in Algorithm 1.

Following the implementation of the contrastive patch loss, we integrate it into the primary loss function of Auxcoformer to optimize model performance comprehensively. For the detection task, the primary loss function $\mathcal{L}_{pri}$ is defined as follows:

$$\mathcal{L}_{pri} = \lambda_{box}\mathcal{L}_{box} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{con}\mathcal{L}_{con}, \tag{14}$$

$$\mathcal{L}_{cls} = -\frac{1}{B}\sum_{i=1}^{B}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)), \tag{15}$$

where λ_{box}, λ_{cls}, and λ_{con} denote the loss weights used to balance the respective contributions of each loss component in the primary loss function. The term $\mathcal{L}_{box}$ represents the box regression loss based on complete intersection over union [68]. Furthermore, $\mathcal{L}_{cls}$ represents the binary cross-entropy loss for object classification, and B denotes the number of objects in the image. Additionally, y_i and $\hat{y}_i$ represent the true and predicted labels, respectively.

Algorithm 1: The pseudocode of contrastive patch loss computation.

Input: Image I;
Predicted bounding box $B_{pred} = (x_{center}, y_{center}, h, w)$;
Temperature parameter τ;
Contrastive network $CN(\cdot)$
Output: Contrastive patch loss $\mathcal{L}_{con}$

Calculate the short side length $N = min(h, w)$
Calculate the bounding box's corner coordinates
$x_{min} = x_{center} - w/2$
$x_{max} = x_{center} + w/2$
$y_{min} = y_{center} - h/2$
$y_{max} = y_{center} + h/2$
$S = 0$ `// Initialize the number of neighbor patches`
for $y = (y_{min} - N)$ **to** (y_{max}) **step** N **do**
  for $x = (x_{min} - N)$ **to** (x_{max}) **step** N **do**
    if $(x = x_{max})$ ***or*** $(x = x_{min} - N)$ ***or*** $(y = y_{max})$ ***or*** $(y = y_{min} - N)$ **then**
      $S \mathrel{+}= 1$ `// Count the number of neighbor patches`
      $P_S \leftarrow I[x : x + N, y : y + N]$ `// Crop a neighbor patch`
for j *in* $\{1, 2, \cdots, S\}$ **do**
  $z_j^{'} \leftarrow CN(P_j)$ `// Extract feature vector of neighbor patch` P_j
$z_i \leftarrow CN(P_i)$ `// Extract feature vector of crack patch` P_i `in` B_{pred}
$\mathcal{L}_{con} \leftarrow -\log\left(-\frac{\exp(z_i^T z_i^{'}/\tau)}{\sum_{j=1}^{S}\exp(z_i^T z_j^{'}/\tau)} + 1\right)$
return $\mathcal{L}_{con}$

Thus, the total loss function $\mathcal{L}_{total}$ is defined as follows:

$$\mathcal{L}_{total} = \lambda_{pri}\mathcal{L}_{pri} + \lambda_{aux}\mathcal{L}_{aux}\ , \tag{16}$$

where λ_{pri} and λ_{aux} denote the loss weights used to modulate the importance between primary and auxiliary tasks. The total loss effectively combines primary detection and auxiliary learning tasks. Consequently, this comprehensive loss function allows Auxcoformer to achieve superior crack detection performance in adverse weather conditions.

4. Experiment

This section describes the datasets used to evaluate the model and explains the results. We compared the proposed method to the existing models in adverse weather conditions and demonstrated that it achieves significant improvements. Finally, an ablation study demonstrates how each model component affects its performance.

4.1. Datasets

The dataset used for the experiments was EDMCrack600 [69], consisting of 600 images of pavement cracks. Because the dataset was not divided into training and validation subsets, we divided this dataset into two subsets: a training set with 550 images and a validation set with 50 images. Every image in both subsets was 680 × 680 pixels.

The DeepCrack [70] dataset comprises 537 diverse images of cracks from various scenes. For the experiments, we divided this dataset into a training set containing 464 images and a validation set of 73 images. Every image in both sets was 448 × 448 pixels.

The Concrete Crack (CC) [71] dataset comprises 486 high-resolution images, taken from the walls and floors of several concrete buildings. We partitioned this dataset into a training set containing 336 images and a validation set with 122 images. All images were resized to 680 × 680 pixels for the experiments.

The reason for selecting these datasets is that they are most suitable for cracks that can occur in external environments, threatening public safety. Specifically, these datasets were chosen because they align well with conditions that are prone to exposure under severe weather. Considering the generally small scale of crack datasets and the complexity of each dataset, we adjusted the ratio of the training set to the validation set to ensure stable training. Additionally, we synthesized these datasets by applying weather effects to simulate adverse weather conditions. In our experiment, we focused on rain and snow, the most frequently occurring weather phenomena in the real world and commonly addressed in numerous studies. Moreover, these conditions include light-scattering effects similar to those of fog and encompass more challenging scenarios of occlusion. Specifically, we followed the methodology of Liu et al. [72] to introduce snow synthetically and followed Yang et al. [73] to add synthetic rain to the images. We adopted the weather modeling proposed in these two papers because datasets synthesized in this manner are widely used and adopted in numerous studies. Furthermore, the rationale behind incorporating these synthetic weather conditions was to evaluate and enhance the robustness of the proposed model under challenging environmental conditions. This approach allowed us to test the adaptability and efficacy of the model in diverse scenarios, providing a comprehensive evaluation of its capabilities.

4.2. Implementation Details

We used a CNN-based backbone of YOLOv8l [25] and a transformer-based decoder inspired by Restormer [42], which were pretrained with several crack datasets [69–71]. Specifically, the CNN-based backbone was selected because of its ability to achieve high accuracy without sacrificing speed. Meanwhile, the transformer-based decoder was selected because of its superiority in restoration by capturing global characteristics. The contrastive network used the VGG16 [74] architecture pretrained with ImageNet [75]. In the 3D FA block, we set the epsilon value to 50 and used a patch size of 16 × 16 pixels. The loss weights in the loss function were set to $\lambda_{L1} = 0.8$, $\lambda_p = 0.5$, $\lambda_{con} = 0.1$, $\lambda_{aux} = 0.1$, and $\lambda_{pri} = 1.0$. These values were determined experimentally to be optimal. We used the SGD optimizer with an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. We continued the end-to-end training process with a batch size of 2 until the loss reached a sufficient level of convergence by observing the training and validation losses. The experimental setup included a system equipped with an Nvidia GeForce RTX 3080 GPU.

4.3. Experimental Results and Analysis

This section presents the performance evaluation of the proposed model and compares it with existing SOTA crack detection models [25,29] and weather restoration models [42,43]. The evaluation was conducted on three datasets: EDMCrack600, DeepCrack, and CC. The models were tested under two weather conditions: snow and rain. The performance was measured using AP50 (%) and AP50:95 (%). AP50 (%) measures the model's precision in detecting objects with a minimum 50% overlap with the actual object location, indicating how accurately the model can identify objects. On the other hand, AP50:95 (%) averages precision across intersection over union (IoU) thresholds from 50% to 95% in 5% steps, offering a comprehensive evaluation of the model's detection accuracy at varying levels of strictness. Essentially, AP50 assesses basic localization accuracy, while AP50:95 examines precision over a wider range of conditions. The best results for each table are denoted in boldface. In more specific terms, we used the open-source codes from Co-DETR [29],

AugMoCrack [28], YOLOv8 [25], MaskDINO [33], MGDIP [34], WeatherDiff [43], and Restormer [42] for comparison, employing the default hyperparameters as outlined in their respective papers.

Table 1 presents the comparative results of the crack detection performance under adverse weather conditions. On the EDMCrack600 validation dataset, the proposed model outperforms all other models with a mean AP50 of 65.2% and a mean AP50:95 of 46.1%, demonstrating superior performance in snow and rain conditions. Similarly, on the DeepCrack validation dataset, the proposed model has a mean AP50 of 89.7% and a mean AP50:95 of 64.4%. On the CC validation dataset, the proposed model achieves a mean AP50 of 93.5% and a mean AP50:95 of 70.5%, further representing its robustness to varying weather conditions compared to other models.

Table 1. Accuracy of crack detection models on the EDMCrack600, DeepCrack, and CC validation datasets with adverse weather conditions (snow and rain). The bold represent the best performances.

Dataset	Detection	Metric (AP50/AP50:95)		
		Snow	Rain	Mean
EDMCrack600 [69]	Co-DETR [29]	38.7/23.2	53.6/33.8	46.2/28.5
	MaskDINO [33]	38.5/25.1	49.0/32.6	43.8/28.9
	MGDIP [34]	40.0/19.9	45.2/20.5	42.6/20.2
	AugMoCrack [28]	37.4/19.5	36.2/19.9	36.8/19.7
	YOLOv8l [25]	37.3/19.9	40.6/22.3	39.0/21.1
	YOLOv8x [25]	39.6/23.6	44.4/24.2	42.0/23.9
	Ours	**62.4/43.6**	**68.0/48.6**	**65.2/46.1**
DeepCrack [70]	Co-DETR [29]	70.5/46.0	74.4/51.0	72.5/48.5
	MaskDINO [33]	67.7/41.1	70.9/48.1	69.3/44.6
	MGDIP [34]	73.2/38.5	71.0/40.6	72.1/39.6
	AugMoCrack [28]	65.0/39.3	66.6/39.4	65.8/39.4
	YOLOv8l [25]	54.1/36.3	61.8/39.8	58.0/38.1
	YOLOv8x [25]	77.1/53.7	70.0/53.1	73.6/53.4
	Ours	**88.5/63.1**	**90.9/65.7**	**89.7/64.4**
CC [71]	Co-DETR [29]	64.3/35.6	64.7/35.7	64.5/35.7
	MaskDINO [33]	66.7/35.3	68.8/36.7	67.8/36.0
	MGDIP [34]	79.5/46.2	76.9/39.6	78.2/42.9
	AugMoCrack [28]	78.8/52.4	75.1/46.3	77.0/49.4
	YOLOv8l [25]	76.0/48.1	70.0/42.0	73.0/45.1
	YOLOv8x [25]	76.9/47.9	75.4/44.5	76.2/46.2
	Ours	**94.4/71.3**	**92.6/69.6**	**93.5/70.5**

In Table 2, we observed a small performance improvement when combining the restoration models, Restormer and WeatherDiff, with the detection models, Co-DETR and YOLOv8x. However, the combined models still perform worse than the proposed model. The proposed model achieves a mean AP50 of 65.2% and a mean AP50:95 of 46.1% on the EDMCrack600 validation dataset. On the DeepCrack validation dataset, the proposed model outperforms the combined Restormer and Co-DETR model by 9.3% and 11.8% in AP50 and AP50:95, respectively. On the CC validation dataset, the proposed model surpasses the combined WeatherDiff and YOLOv8x models by 6.2% and 9.5% in AP50 and AP50:95, respectively. Overall, the proposed model demonstrates robustness across all adverse weather conditions in various datasets, indicating its potential for widespread application in crack detection tasks.

Table 2. Performance comparison of crack detection with restoration models on the EDMCrack600, DeepCrack, and CC validation datasets under adverse weather conditions (snow and rain). The bold represent the best performances.

Dataset	Restoration	Detection	Metric (AP50/AP50:95)		
			Snow	Rain	Mean
EDMCrack600 [69]	Restormer [42]	Co-DETR [29]	51.6/32.3	58.1/36.3	54.9/34.3
		MaskDINO [33]	49.8/30.2	57.9/36.1	53.9/33.2
		MGDIP [34]	50.3/30.6	59.6/35.0	55.0/32.8
		YOLOv8x [25]	48.6/32.2	57.8/39.0	53.2/35.6
	WeatherDiff [43]	Co-DETR [29]	44.9/27.3	52.9/31.6	48.9/29.5
		MaskDINO [33]	44.0/26.5	50.8/30.2	47.4/28.4
		MGDIP [34]	52.4/25.1	52.6/24.4	52.5/24.8
		YOLOv8x [25]	50.5/32.8	45.4/24.6	48.0/28.7
	Ours		**62.4/43.6**	**68.0/48.6**	**65.2/46.1**
DeepCrack [70]	Restormer [42]	Co-DETR [29]	79.8/52.0	89.8/63.9	84.8/58.0
		MaskDINO [33]	76.2/50.9	87.1/62.4	81.7/56.7
		MGDIP [34]	80.1/49.3	88.0/58.1	84.1/53.7
		YOLOv8x [25]	74.4/50.7	82.8/56.2	78.6/53.5
	WeatherDiff [43]	Co-DETR [29]	57.4/34.3	73.6/42.0	65.5/38.2
		MaskDINO [33]	58.5/36.7	76.3/42.2	67.4/39.5
		MGDIP [34]	65.7/40.3	77.3/45.5	71.5/42.9
		YOLOv8x [25]	73.6/47.4	83.3/53.9	78.5/50.7
	Ours		**88.5/63.1**	**90.9/65.7**	**89.7/64.4**
CC [71]	Restormer [42]	Co-DETR [29]	66.8/36.0	68.7/34.8	67.8/35.4
		MaskDINO [33]	64.0/34.8	69.5/34.7	66.8/34.8
		MGDIP [34]	79.2/42.6	74.4/40.6	76.8/41.6
		YOLOv8x [25]	83.0/51.7	80.5/49.8	81.8/50.8
	WeatherDiff [43]	Co-DETR [29]	63.4/30.5	63.2/27.4	63.3/29.0
		MaskDINO [33]	62.6/29.7	60.8/27.0	61.7/28.4
		MGDIP [34]	80.7/45.6	83.1/49.8	81.9/47.7
		YOLOv8x [25]	87.4/60.4	87.1/61.5	87.3/61.0
	Ours		**94.4/71.3**	**92.6/69.6**	**93.5/70.5**

Figure 5 visually compares crack detection performance between the proposed model and YOLOv8x in adverse weather conditions. In (a) and (c), the YOLOv8x tends to predict bounding boxes that cover a smaller range than the GT boxes. Conversely, in (d), the YOLOv8x predicts overly large bounding boxes. In contrast, the proposed model predicts more accurate bounding boxes in all cases. In (b), where snowflakes are heavily present, YOLOv8x fails to make any predictions, whereas the proposed model successfully identifies cracks under such challenging conditions.

Figures 6–8 present the restoration results of the auxiliary restoration network on the EDMCrack600, DeepCrack, and CC validation datasets with snow and rain conditions. The qualitative results indicate that our auxiliary restoration network effectively restores visible information of cracks and efficiently removes weather-related noise surrounding the cracks simultaneously. By leveraging hierarchical features from the auxiliary restoration network, which contain detailed information about the cracks, we can enhance the detectability of cracks, even in challenging adverse weather conditions.

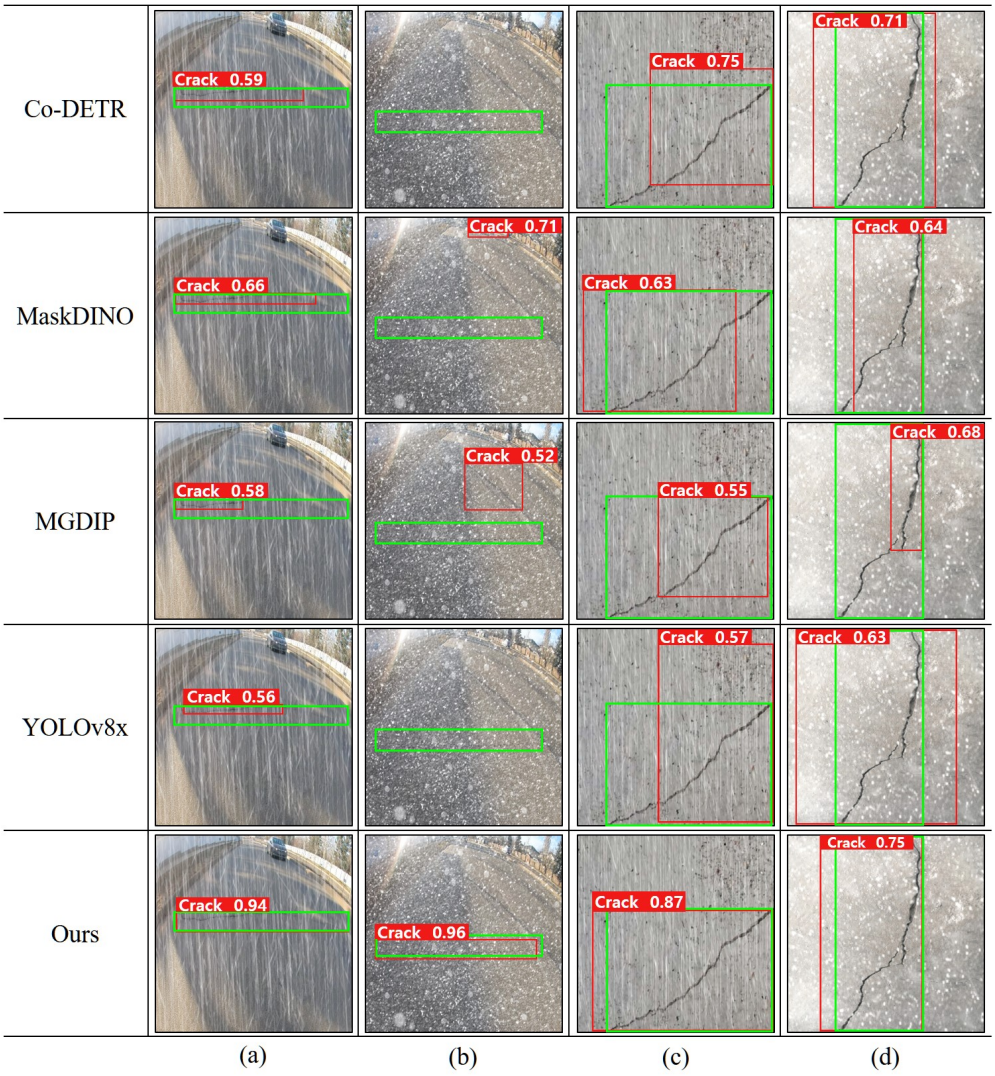

Figure 5. Visual comparison of crack detection performance across multiple datasets for (**a**,**b**) EDM-Crack600, (**c**) DeepCrack, and (**d**) Concrete Crack (CC). Predicted bounding boxes are red and ground truth boxes are green.

Figure 6. Visual restoration results of the auxiliary restoration network on the EDMCrack600 validation dataset with adverse weather conditions (snow and rain).

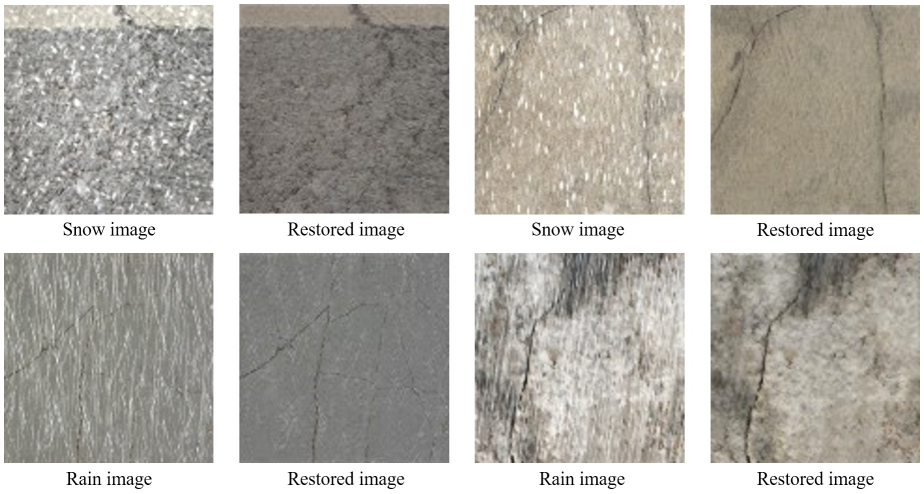

Figure 7. Visual restoration results of the auxiliary restoration network on the DeepCrack validation dataset with adverse weather conditions (snow and rain).

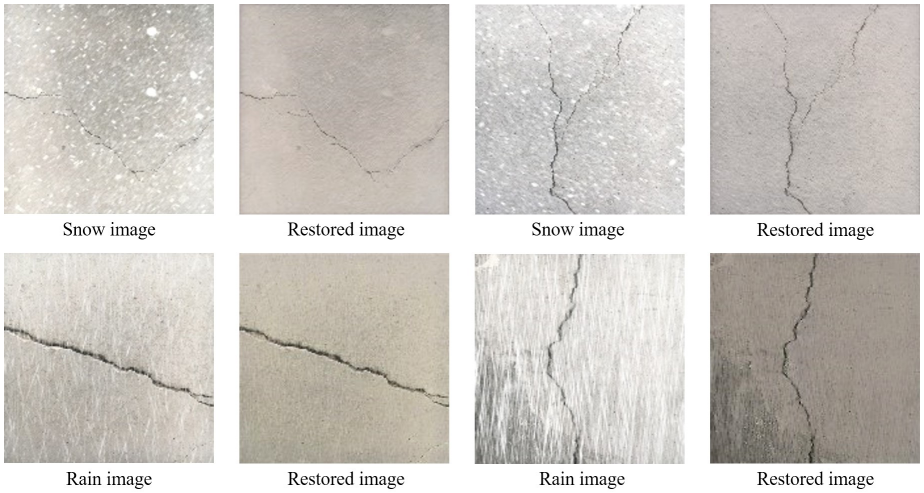

Figure 8. Visual restoration results of the auxiliary restoration network on the CC validation dataset with adverse weather conditions (snow and rain).

Table 3 compares the computational complexity of the proposed model with detection and restoration models in terms of floating point operations (FLOPs), number of parameters (Param), and inference time on the EDMCrack600 dataset. The FLOPs and Param are essential factors in determining the computational efficiency of a model. Regarding the computational requirements, the transformer-based Co-DETR model is the most resource-intensive, as evidenced by its 612.7 gigaFLOPs, 235.5 million parameters, and 380.8 ms of inference time. Moreover, MaskDINO has particularly high FLOPs, which demonstrates the high complexity of transformer-based models. In contrast, the CNN-based YOLOv8x exhibits significantly reduced computational complexity compared to Co-DETR, using 257.8 gigaFLOPs, 68.2 million parameters, and 17.0 ms of inference time. The proposed crack detection model shows an optimal balance between computational efficiency and performance, with 238.5 gigaFLOPs, 69.5 million parameters, and 21.6 ms of inference time. Despite these values being slightly higher than those of YOLOv8l, the proposed model provides superior crack detection performance while maintaining reasonable computational complexity. Consequently, this balanced architecture allows the proposed model to achieve real-time operation and superior performance in crack detection under adverse weather conditions. Through this analysis, we demonstrate that our methodology is well suited for real-world applications, such as in UAVs.

Table 3. Computation complexity of comparison models on the EDMCrack600 validation dataset under snow conditions.

Task	Model	FLOPs (G)	Param (M)	Time (ms)
Detection	Co-DETR [29]	612.7	235.5	380.0
	MaskDINO [33]	1326.5	223.1	163.9
	MGDIP [34]	210.7	68.4	131.5
	AugMoCrack [28]	203.8	86.2	14.8
	YOLOv8l [25]	165.2	43.7	10.7
	YOLOv8x [25]	257.8	68.2	17.0
Restoration	Restormer [42]	881.2	26.1	159.8
	WeatherDiff [43]	1,016,482.5	29.7	97,749.9
Ours		238.5	69.5	21.6

4.4. Ablation Study

Table 4 presents an ablation study to analyze the effectiveness of the proposed components on EDMCrack600 validation dataset under adverse weather conditions (snow and rain). The performance evaluation is conducted by separately averaging the AP50 and AP50:95 scores across snow and rain conditions. The first row exhibits the baseline performance with all components activated. The second row shows the performance without the auxiliary restoration network. Because the 3D FA block, cross-attention, and auxiliary loss depend on the auxiliary restoration network, these components are also deactivated. Instead, we use self-attention in place of cross-attention. As expected, the performance significantly decreases due to not utilizing restored features and the absence of related components (3D FA block, cross-attention, and auxiliary loss). The third row shows the decreased performance when only the 3D FA block is deactivated. This result indicates that the 3D FA block effectively enhances the characteristics of cracks for detection. The fourth row shows the performance with concatenated feature fusion instead of cross-attention. In comparison, cross-attention leads to an increase of 9.0%p/7.7%p over concatenation, demonstrating its greater effectiveness in the feature fusion process. Moreover, the fifth and sixth rows present the performance when the proposed auxiliary and contrastive patch losses are deactivated, respectively. The auxiliary loss leads to an improvement of 15.0%p/9.4%p, and the contrastive patch loss leads to an improvement of 4.1%p/3.4%p. These results indicate that each proposed component contributes to performance improvement.

Table 4. Ablation study of Auxcoformer for the proposed components on EDMCrack600 validation dataset under adverse weather conditions (snow and rain).

Auxiliary Restoration Network	3D FA Block	Cross-Attention	Auxiliary Loss	Contrastive Patch Loss	Mean AP50/AP50:95
✓	✓	✓	✓	✓	65.2/46.1
				✓	48.3/29.6
✓		✓	✓	✓	58.7/40.0
✓	✓		✓	✓	56.2/38.4
✓	✓	✓		✓	50.2/36.7
✓	✓	✓	✓		61.1/42.7

Table 5 provides insight into the influence of the weighting factors λ_{con}, λ_{pri}, and λ_{aux} on the EDMCrack600 validation dataset. This table was composed of meaningful values through various experiments for analysis. By comparing the first to the third rows, we can observe that as the ratios of λ_{con} and λ_{aux} change relative to λ_{pri}, performance variations occur. Furthermore, comparing the third to the seventh rows reveals the ablation study results, showing the effects of activating or deactivating each module. We fixed λ_{pri}

at 1.0 for all cases. When λ_{aux} is 0.0, the restoration task is not performed. Specifically, when λ_{aux} is activated at 0.1 with λ_{con} = 0.0, the mean AP50 and AP50:95 scores increase by 13.4% and 12.9%, respectively, compared to when λ_{con} and λ_{aux} are set to 0.0. Similarly, when λ_{con} is activated at 0.1 with λ_{aux} = 0.0, the mean AP50 and AP50:95 scores increase by 2.5% and 6.9%, respectively, compared to when λ_{con} and λ_{aux} are set to 0.0. The optimal performance for the proposed model is achieved when λ_{con}, λ_{pri}, and λ_{aux} are 0.1, 1.0, and 0.1, respectively, yielding the highest mean AP50 of 65.2% and AP50:95 of 46.1%. These results confirm that the contrastive and auxiliary losses are instrumental in improving the model performance. However, setting λ_{con} or λ_{aux} values higher than 0.1 tends to negatively affect the primary detection task. Additionally, as the proposed λ_{con} and λ_{aux} values increase, the decline in performance becomes more pronounced, presenting the model's increased sensitivity to these parameter variations. Therefore, a balanced contribution and appropriate weight selection from the contrastive and auxiliary losses is crucial for optimal performance.

Table 5. Average precision variation according to λ on the EDMCrack600 validation dataset under adverse weather conditions (snow and rain). The bold represent the best performances.

λ_{con}	λ_{pri}	λ_{aux}	Snow	Rain	Mean
1.0	1.0	1.0	53.8/38.5	63.3/43.1	58.6/40.8
0.5	1.0	0.5	60.1/41.1	67.4/47.8	63.8/44.5
0.1	1.0	0.1	**62.4/43.6**	**68.0/48.6**	**65.2/46.1**
0.1	1.0	0.0	47.9/35.1	52.5/38.3	50.2/36.7
0.0	1.0	0.1	58.8/41.7	63.3/43.6	61.1/42.7
0.0	1.0	0.0	44.2/27.5	51.2/32.0	47.7/29.8

Table 6 presents an ablation study on the effect of various ϵ values in the 3D FA block. We experimented with ϵ values of 10, 50, 100, and 10,000 to selectively amplify frequencies most relevant to crack detection. For experimental analysis, only the necessary ϵ values were selected and recorded after conducting various experiments. The ϵ of 50 yielded the highest mean AP50 and AP50:95 scores of 65.2% and 46.1%, respectively. When the ϵ was set to 10, performance declined, likely because a lower ϵ amplifies relevant frequencies for crack detection and noise components. Similarly, the higher ϵ of 100 led to a decline in performance, because it exceeds the frequency components relevant for crack detection. When ϵ is set to 10,000, the coefficient range is confined to the DC value, rendering the FA operation inactive. This is based on the observation that the 3D DCT coefficients in our proposed FA block do not exceed this value. The performance in this case was better than when ϵ was set to 100. This is likely because when ϵ is 100, some unnecessary frequency components are amplified, reducing performance. Additionally, the change in ϵ from 50 to 10 shows a greater decrease in performance compared to the change from 50 to 100. This is attributed to the fact that a lower ϵ value tends to amplify high-frequency noise as well, to which the model is more responsive, particularly in those frequency coefficients. Therefore, the ϵ of 50 provides the optimal balance for enhancing features crucial for crack detection while minimizing noise effects.

Table 6. Average precision variation according to ϵ on the EDMCrack600 validation dataset under adverse weather conditions (snow and rain). The bold represent the best performances.

ϵ	Metric (AP50/AP50:95)		
	Snow	Rain	Mean
10	48.5/25.9	54.6/30.6	51.6/28.3
50	**62.4/43.6**	**68.0/48.6**	**65.2/46.1**
100	51.6/27.9	55.7/31.4	53.7/29.7
10,000	55.1/38.2	62.3/41.7	58.7/40.0

5. Discussion

Our crack detection method presented in this paper demonstrates considerable potential for robust performance in adverse weather conditions. However, there are several challenges that need to be addressed. Detecting cracks on ground surfaces can become problematic when they are completely covered by accumulated rainwater or by snow. In such scenarios, rain and snow entirely occlude the visible information of cracks required for detection. Moreover, evaluation in real-world conditions is somewhat constrained due to the absence of available real-world crack data in adverse weather scenarios. In these cases, efforts will be required to collect crack data under adverse weather conditions and to utilize additional equipment to supplement the drawbacks of cameras for detection.

As a limitation in terms of the proposed method, the auxiliary restoration network relies on supervised learning that requires non-degraded data to be paired with degraded data from the same scene. Alternatively, we can use the pretrained weights of the auxiliary restoration network, which is trained on synthetic data, during the training of the primary detection network without integrating the auxiliary loss. However, this approach may not utilize the complete potential of the proposed framework. In our future work, we will focus on developing methods that are not reliant on supervised learning to overcome this limitation.

6. Conclusions

In conclusion, we presented Auxcoformer, an efficient model that integrates a primary detection network with an auxiliary restoration network. This approach addresses the challenges that automated robots may encounter due to visual difficulties in varying external environmental conditions leading to low-quality data. Auxcoformer improves crack detection performance under adverse weather conditions by employing effective loss functions and leveraging auxiliary learning. In addition, Auxcoformer employs 3D FA to effectively emphasize crack features, enhancing its robustness and reliability. Consequently, this model represents a substantial advancement in crack detection, enhancing safety and automation abilities in UAV and autonomous vehicle navigation, while also ensuring real-time capabilities, essential for real-world applications.

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

Funding: This work was supported by the Industrial Fundamental Technology Development Program (No. 20018699) funded by MOTIE of Korea and the IITP grant funded by the Korea government (MSIT) (No. 2021-0-02068, RS-2023-00256629, RS-2022-00156287).

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: EDMCrack600 https://github.com/mqp2259/EdmCrack600 (accessed on 5 September 2023), DeepCrack https://github.com/yhlleo/DeepCrack (accessed on 12 January 2024), and Concrete Crack https://data.mendeley.com/datasets/jwsn7tfbrp/1 (accessed on 24 January 2024).

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

References

1. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1833–1844.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 7–12. [CrossRef]
3. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
4. Nie, M.; Wang, K. Pavement distress detection based on transfer learning. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 435–439.

5. Hascoet, T.; Zhang, Y.; Persch, A.; Takashima, R.; Takiguchi, T.; Ariki, Y. Fasterrcnn monitoring of road damages: Competition and deployment. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5545–5552.
6. Vishwakarma, R.; Vennelakanti, R. Cnn model & tuning for global road damage detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5609–5615.
7. Pei, Z.; Lin, R.; Zhang, X.; Shen, H.; Tang, J.; Yang, Y. CFM: A consistency filtering mechanism for road damage detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5584–5591.
8. Xiang, X; Wang, Z.; Qiao, Y. An improved YOLOv5 crack detection method combined with transformer. *IEEE Sensors J.* **2022**, *22*, 14328–14335. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
10. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
12. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 7464–7475.
13. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 6569–6578.
14. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
15. Yu, G.; Zhou, X. An Improved YOLOv5 Crack Detection Method Combined with a Bottleneck Transformer. *Mathematics* **2023**, *11*, 2377. [CrossRef]
16. Mandal, V.; Uong, L.; Adu-Gyamfi, Y. Automated road crack detection using deep convolutional neural networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5212–5215.
17. Shao, C.; Zhang, L.; Pan, W. PTZ camera-based image processing for automatic crack size measurement in expressways. *IEEE Sensors J.* **2021**, *21*, 23352–23361. [CrossRef]
18. Zhang, R.; Shi, Y.; Yu, X. Pavement crack detection based on deep learning. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 7367–7372.
19. Zhang, X.; Xia, X.; Li, N.; Lin, M.; Song, J.; Ding, N. Exploring the tricks for road damage detection with a one-stage detector. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5616–5621.
20. Liu, Y.; Zhang, X.; Zhang, B.; Chen, Z. Deep network for road damage detection. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5572–5576.
21. Mandal, V.; Mussah, A.R.; Adu-Gyamfi, Y. Deep learning frameworks for pavement distress classification: A comparative analysis. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 5577–5583.
22. Guo, G.; Zhang, Z. Road damage detection algorithm for improved YOLOv5. *Sci. Rep.* **2022**, *12*, 15523. [CrossRef]
23. Hu, G.X.; Hu, B.L.; Yang, Z.; Huang, L.; Li, P. Pavement crack detection method based on deep learning models. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5573590. [CrossRef]
24. Hong, Y.; Yoo, S.B. OASIS-Net: Morphological Attention Ensemble Learning for Surface Defect Detection. *Mathematics* **2022**, *10*, 4114. [CrossRef]
25. YOLOv8 by MMYOLO. Available online: https://github.com/open-mmlab/mmyolo/ (accessed on 13 May 2023).
26. Wang, J.; Chen, Y.; Dong, Z.; Gao, M. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Comput. Appl.* **2023**, *35*, 7853–7865. [CrossRef]
27. Yu, J.; Oh, H.; Fichera, S.; Paoletti, P.; Luo, S. Multi-source Domain Adaptation for Unsupervised Road Defect Segmentation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation, London, UK, 29 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 5638–5644.
28. Hong, Y.; Lee, S.; Yoo, S.B. AugMoCrack: Augmented morphological attention network for weakly supervised crack detection. *Electron. Lett.* **2022**, *58*, 651–653. [CrossRef]
29. Zong, Z.; Song, G.; Liu, Y. Detrs with collaborative hybrid assignments training. *arXiv* **2022**, arXiv:2211.12860.
30. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z., Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–21 June 2021; pp. 10012–10022.
31. Kim, M.H.; Yoo, S.B. Memory-Efficient Discrete Cosine Transform Domain Weight Modulation Transformer for Arbitrary-Scale Super-Resolution. *Mathematics*, **2023**, *11*, 3954. [CrossRef]

32. Hong, Y.; Kim, M. J.; Lee, I.; Yoo, S.B. Fluxformer: Flow-Guided Duplex Attention Transformer via Spatio-Temporal Clustering for Action Recognition. *IEEE Robot. Autom. Lett.* **2023**, *8*, 6411–6418. [CrossRef]
33. Li, F.; Zhang, H.; Xu, H.; Liu, S.; Zhang, L.; Ni, L.M.; Shum, H.Y. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 3041–3050.
34. Kalwar, S.; Patel, D.; Aanegola, A.; Konda, K.R.; Garg, S.; Krishna, K.M. GDIP: Gated Differentiable Image Processing for Object Detection in Adverse Conditions. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 7083–7089.
35. Xia, Y.; Monica, J.; Chao, W.L.; Hariharan, B.; Weinberger, K.Q.; Campbell, M. Image-to-Image Translation for Autonomous Driving from Coarsely-Aligned Image Pairs. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 7756–7762.
36. Feng, X.; Pei, W.; Jia, Z.; Chen, F.; Zhang, D.; Lu, G. Deep-masking generative network: A unified framework for background restoration from superimposed images. *IEEE Trans. Image Process.* **2021**, *30*, 4867–4882. [CrossRef] [PubMed]
37. Li, B.; Liu, X.; Hu, P.; Wu, Z.; Lv, J.; Peng, X. All-in-one image restoration for unknown corruption. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 17452–17462.
38. Yun, J.S.; Yoo, S.B. Single image super-resolution with arbitrary magnification based on high-frequency attention network. *Mathematics*, **2022**, *10*, 275. [CrossRef]
39. Yun, J.S.; Yoo, S.B. Kernel-attentive weight modulation memory network for optical blur kernel-aware image super-resolution. *Opt. Lett.* **2023**, *48*, 2740–2743. [CrossRef] [PubMed]
40. Yun, J.S.; Kim, M.H.; Kim, H.I.; Yoo, S.B. Kernel adaptive memory network for blind video super-resolution. *Expert Syst. Appl.* **2024**, *238*, 122252. [CrossRef]
41. Li, R.; Tan, R.T.; Cheong, L.F. All in one bad weather removal using architectural search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3175–3185.
42. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.H. Restormer: Efficient transformer for high-resolution image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 5728–5739.
43. Özdenizci, O.; Legenstein, R. Restoring vision in adverse weather conditions with patch-based denoising diffusion models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10346–10357. [CrossRef]
44. Valanarasu, J.M.J.; Yasarla, R.; Patel, V.M. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 2353–2363.
45. Lee, Y.; Jeon, J.; Ko, Y.; Jeon, B.; Jeon, M. Task-driven deep image enhancement network for autonomous driving in bad weather. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 13746–13753.
46. Wang, J.; Chen, Y., Ji, X.; Dong, Z.; Gao, M.; Lai, C.S. Vehicle-mounted adaptive traffic sign detector for small-sized signs in multiple working conditions. *IEEE Trans. Intell. Transp. Syst.*, **2023**, *25*, 710–724. [CrossRef]
47. Zhai, X.; Oliver, A.; Kolesnikov, A.; Beyer, L. S4l: Self-supervised semi-supervised learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1476–1485.
48. Heo, Y.; Kang, S. A Simple Framework for Scene Graph Reasoning with Semantic Understanding of Complex Sentence Structure. *Mathematics* **2023**, *11*, 3751. [CrossRef]
49. Wen, H.; Zhang, J.; Wang, Y.; Lv, F.; Bao, W.; Lin, Q.; Yang, K. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 2377–2386.
50. Du, Y.; Czarnecki, W.M.; Jayakumar, S.M.; Farajtabar, M.; Pascanu, R.; Lakshminarayanan, B. Adapting auxiliary losses using gradient similarity. *arXiv* **2018**, arXiv:1812.02224.
51. Shi, B.; Hoffman, J.; Saenko, K.; Darrell, T.; Xu, H. Auxiliary task reweighting for minimum-data learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7148–7160.
52. Lin, X.; Baweja, H.; Kantor, G.; Held, D. Adaptive auxiliary task weighting for reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 4772–4783.
53. Dery, L.M.; Dauphin, Y.; Grangier, D. Auxiliary task update decomposition: The good, the bad and the neutral. *arXiv* **2021**, arXiv:2108.11346.
54. Navon, A.; Achituve, I.; Maron, H.; Chechik, G.; Fetaya, E. Auxiliary Learning by Implicit Differentiation. *arXiv* **2020**, arXiv:2007.02693.
55. Chen, H.; Wang, X.; Guan, C.; Liu, Y.; Zhu, W. Auxiliary learning with joint task and data scheduling. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 3634–3647.
56. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 12–18 July 2020; pp. 1597–1607.
57. Xiong, H.; Yan, Z.; Zhao, H.; Huang, Z.; Xue, Y. Triplet Contrastive Learning for Aspect Level Sentiment Classification. *Mathematics* **2022**, *10*, 4099. [CrossRef]

58. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
59. Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; Isola, P. What makes for good views for contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6827–6839.
60. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
61. Zhou, X.; Li, S.; Pan, Z.; Zhou, G.; Hu, Y. Multi-Aspect SAR Target Recognition Based on Non-Local and Contrastive Learning. *Mathematics* **2023**, *11*, 2690. [CrossRef]
62. Xie, E.; Ding, J.; Wang, W.; Zhan, X.; Xu, H.; Sun, P.; Li, Z.; Luo, P. Detco: Unsupervised contrastive learning for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–21 June 2021; pp. 8392–8401.
63. Park, T.; Efros, A.A.; Zhang, R.; Zhu, J.Y. Contrastive learning for unpaired image-to-image translation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 319–345.
64. Qian, R.; Meng, T.; Gong, B.; Yang, M.H.; Wang, H.; Belongie, S.; Cui, Y. Spatiotemporal contrastive video representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitionn, Nashville, TN, USA, 19–21 June 2021; pp. 6964–6974.
65. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschoinot, A.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.
66. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
67. Johnson, J.; Alahi, A.; Li, F.-F. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.
68. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34.
69. Mei, Q.; Gül, M. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Constr. Build. Mater.* **2020**, *256*, 119397. [CrossRef]
70. Liu, Y.; Yao, J.; Lu, X.; Xie, R.; Li, L. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing* **2019**, *338*, 139–153. [CrossRef]
71. Özgenel, Ç.F.; Sorguç, A.G. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC 2018), Berlin, Germany, 20–25 July 2018; IAARC Publications: Edinburgh, UK, 2018; Volume 35, pp. 1–8.
72. Liu, Y.F.; Jaw, D.W.; Huang, S.C.; Hwang, J.N. DesnowNet: Context-aware deep network for snow removal. *IEEE Trans. Image Process.* **2018**, *27*, 3064–3073. [CrossRef]
73. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1357–1366.
74. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
75. Deng, J. Imagenet: A large-scale hierarchical image database. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.