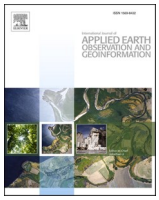
International Journal of Applied Earth Observations and Geoinformation 110 (2022) 102825

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

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



Pavement crack detection from CCD images with a locally enhanced transformer network

Zhengsen Xu [a], Haiyan Guan [a,*], Jian Kang [a], Xiangda Lei [a], Lingfei Ma [b,*], Yongtao Yu [c], Yiping Chen [d], Jonathan Li [e]

[a] School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
[b] School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 102206, China
[c] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China
[d] Department of Computer Sciences, School of Informatics, Xiamen University, Xiamen 361000, China
[e] Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
Pavement crack detection
Deep learning
Semantic segmentation
Self-attention
Transformer

ABSTRACT

Precisely identifying pavement cracks from charge-coupled devices (CCDs) captured high-resolution images faces many challenges. Even though convolutional neural networks (CNNs) have achieved impressive performance in this task, the stacked convolutional layers fail to extract long-range contextual features and impose high computational costs. Therefore, we propose a locally enhanced Transformer network (LETNet) to completely and efficiently detect pavement cracks. In the LETNet, Transformer is employed to model long-range dependencies. By designing a convolution stem and a local enhancement module, both low-level and high-level local features can be compensated. To take advantage of these rich features, a skip connection strategy and an efficient upsampling module is built to restore detailed information. In addition, a defect rectification module is further developed to reinforce the network for hard sample recognition. The quantitative comparison demonstrates that the proposed LETNet outperformed four advanced deep learning-based models with respect to both efficiency and effectiveness. Specifically, the average precision, recall, ODS, IoU, and frame per second (FPS) of the LETNet on three testing datasets are approximately 93.04%, 92.85%, 92.94%, 94.07%, and 30.80FPS, respectively. We also built a comprehensive pavement crack dataset containing 156 high-resolution manually annotated CCD images and made it publicly available on Zenodo.

1. Introduction

Pavement cracks are an early sign of road destruction and a growing threat to driving safety and road maintenance. Rain, salts, and oils will erode roadbeds through pavement cracks when road surfaces are broken, which accelerates the aging of road structures (Pan et al., 2020). For example, a small pavement crack will easily degenerate into a pothole over a rain or snow night (Zou et al., 2019). According to a survey conducted by the Ministry of Transport of China, the road maintenance mileage in China has reached 5.14 million kilometers, about 99.0% of the total road mileage in 2020 (Ying, 2021). Thus, periodically monitoring road surface conditions is demanded by road maintenance agencies to ensure driving safety and pavement infrastructure serviceability.

Whereas, traditional visual inspection approach is time-consuming, cost-intensive, and biased. To circumvent this, the optical imaging with onboard charge-coupled device (CCD) sensors combined with digital image processing technologies has attracted much attention because it can automatically monitor pavement conditions (Mei and Gül, 2020). Pavement cracks in digital images are typically shown as linear structures with shape variations. Thus, pavement crack detection can be considered as a linear object detection task, which is a common task in computer vision (Zou et al., 2019). Pavement crack detection methods can be divided into two categories, i.e., traditional image processing-based methods and deep learning-based methods. In the first category, the existing image processing algorithms, e.g., threshold segmentation, edge detection, morphology operation, template matching, and region growing, have been widely used in crack detection tasks and

* Corresponding authors.
E-mail addresses: xuzs@nuist.edu.cn (Z. Xu), guanhy.nj@nuist.edu.cn (H. Guan), xdlei@nuist.edu.cn (X. Lei), l53ma@cufe.edu.cn (L. Ma), allennessy@hyit.edu.cn (Y. Yu), chenyiping@xmu.edu.cn (Y. Chen), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2022.102825
Received 29 March 2022; Received in revised form 13 May 2022; Accepted 14 May 2022
Available online 22 May 2022

obtained satisfactory performance (Li et al., 2011; Koch et al., 2015; Zhang et al., 2017; Cho et al., 2018; Dorafshan et al., 2018). Subsequently, traditional machine learning methods, such as artificial neural network (ANN) and support vector machine (SVM), have been increasingly applied to pavement crack detection because such methods can obtain predictions by learning intrinsic knowledge of pavement crack data (Li et al., 2017).

Although many achievements have been obtained in this task, these aforementioned traditional algorithms are mainly based on hand-crafted features, which heavily rely on the experience of experts. Besides, the road environment conditions are rather complex, such as landmarks, oil and water stains. Additionally, most publicly available datasets are composed of single-band images with low spectral contrasts, intense inhomogeneity along pavement cracks, heavy noise interferences, and containing less information. Therefore, it is hard for traditional methods to determine the optimal features to extract complete and continuous pavement cracks. Furthermore, feature extraction and selection are tedious and subjective, resulting in unreliable crack detection results.

In recent years, deep learning-based models have attracted much attention in the fields of photogrammetry, remote sensing, and computer vision. Although many architectures have been explored, the deep convolutional neural network (CNN) is still dominant due to its ability to automatically learn high-level features and model high-dimensional non-linear functions. Recent studies demonstrated the dominance of deep learning-based methods on pavement crack detection from CCD images. Typically, deep learning-based crack detection methods can be classified into three categories including image classification-based, objection detection-based, and semantic segmentation-based methods. Image classification-based methods classify image patches into cracks and non-cracks using various deep learning networks, such as CNNs (Dung et al., 2019), as well as CNNs variants embedded with an atrous spatial pyramid pooling module (Xu et al., 2019) and a modified squeeze-and-excitation block (Li et al., 2020). However, the image classification-based models lack large-scale receptive fields and boundary details because the images are divided into a set of small patches, which may lead to an undesirable performance in crack detection. Object detection-based methods identify and locate pavement cracks on images simultaneously by employing a CNN-based backbone and region proposal networks (Tran et al., 2020; Wu et al., 2020). However, the bounding boxes of crack candidates are typically rectangle-shaped, which also constrains the application of such approaches to the quantitative evaluation of pavement conditions. Moreover, down-sampling techniques, such as region of interest (ROI) pooling, employed in the above models bring challenges for detecting small objects (e.g., pavement cracks). Image semantic segmentation-based methods, such as UNet-based and FPN-based models, label each pixel of images as cracks or non-cracks based on geometrical features. Inspired by these methods, to accurately detect pavement cracks, many researchers have developed numerous modifications, such as SegNet-based DeepCrack (Zou et al., 2019) and generative adversarial networks (GAN) combined with connectivity maps (Mei and Gül, 2020).

Even though the above-mentioned deep learning-based networks achieved good performances, they are still insufficient in extracting long-range contextual information, which is crucial for scene parsing (Shuai et al., 2018). Specifically, dense prediction tasks are often ambiguous when only local information is considered (Li et al., 2021). The pavement cracks are typically long and thin curves and are surrounded by complex backgrounds. The interference of small receptive fields often causes fragmented or false-positive predictions (Li et al., 2021). Fortunately, the availability of long-range dependency contributes to the delineation of pavement cracks owing to the utilization of extensive semantic cues captured from the whole images (Zhang et al., 2020). Commonly utilized mechanisms include attention mechanism (Fang et al., 2021; Cui et al., 2021; Bhattacharya et al., 2021; Dong et al., 2022) and dilated/atrous convolutions (Ji et al., 2020; Hsieh et al., 2021; Li et al., 2021; Xu et al., 2021). However, the main architectures of CNNs and their variants remained unchanged and the performance boost was limited. Besides, attention modules and atrous convolutions caused high computational costs and discontinuous crack detection results (Zheng et al., 2021).

Compared with traditional CNN-based networks, a vision Transformer (ViT) model was proposed to capture long-range dependencies using a multi-head self-attention mechanism with limited layers (Dosovitskiy et al., 2020). Specifically, A Transformer-based method divides images into a set of patches and reshapes each patch into a feature vector via a patch embedding process. Plus with positional embeddings and classification token embeddings, the patches are input into the Transformer network. The Transformer network considers token interaction in pair-wise to dynamically model long-range dependencies with enlarged receptive fields. Many works proved that Transformer-based models achieved outstanding performance comparable to CNN-based models in many fields, such as image detection, segmentation, and reconstruction (Liu et al., 2021b; Cao et al., 2021; Liu et al., 2021b; Wang et al., 2021). However, these Transformer-based models were also severely affected by pre-training weight initialization because of a lack of structure bias assumption of the input data. Moreover, they have defects in local feature modeling and high-resolution image processing owing to token partitioning (Xiao et al., 2021). To improve the local feature extraction performance of the ViT models, many networks have been developed, such as PVT (Wang et al., 2021), TNT (Han et al., 2021), CrackFormer (Liu et al., 2021), and DefectTR (Dang et al., 2022). However, these modifications significantly slow down the running speed and increase computational costs.

To address the abovementioned issues from both CNNs and Transformers, we propose a locally enhanced Transformer network (named LETNet) to detect pavement cracks in an end-to-end manner. The LETNet consists of an encoder for extracting low-level local and global features of pavement cracks at different scales, and a decoder integrating with skip connections for fusing these different-scale pavement crack features to provide a semantically strong feature representation for crack detection. The proposed LETNet uses Transformer to model long-distance dependencies, and represent global features of pavement cracks. To compensate the Transformer for the loss of local fine-grained features, a local enhance module is appended to the Transformer module in each stage of the encoder. In each stage of the decoder, an efficient upsampling module is constructed and embedded to improve the recovery of detailed information. Moreover, a defect rectification module is constructed to enhance crack detection performance in a deep supervision manner. The main contributions of this paper are as follows:

- To provide global crack feature representation by modeling long-range dependencies, we develop an encoder-decoder Transformer architecture, called LETNet.
- To compensate the transformer for the deficiency of local features, a local enhancement module is embedded into the LETNet. To facilitate Transformer training and obtain better segmentation maps, a defect rectification module and an efficient upsampling module are introduced into the LETNet network without producing extra computation overhead.
- To improve the robustness of the proposed LETNet, we build a manually annotated pavement crack dataset, i.e., CrackNJ156 (available on Zenodo, https://doi.org/10.5281/zenodo.6526409), for high-resolution CCD image-based pavement crack detection.

The rest of this paper is organized as follows. Section 2 details the architecture of the LETNet. The datasets, implementation details, experiments, and discussions are reported in Section 3. Finally, Section 4 provides the concluding remarks.

2. Methodology

In this section, the LETNet architecture is first presented to give a

comprehensive workflow. Then, we detail the fundamental components, i.e., the convolution stem, the Transformer module, the local enhancement module, the efficient up-sampling module, and the defect rectification module, respectively.

2.1. Network architecture overview

The proposed LETNet is designed to enhance the performance of pavement crack detection from CCD images by enlarging the receptive fields of CNN-based networks and compensating the Transformer for the loss of local fine-grained contextual information. As illustrated in Fig. 1, the proposed LETNet architecture is composed of an encoder, a decoder, and unique addition skip connections. Firstly, the encoder aims to extract local and global semantic representations by a convolution stem and four stages of a multi-head self-attention-based Transformer module interweaved with a local enhancement module. To be more specific, the encoder starts with a convolutional stem that downscales input image size by 4 × and increases the channel number by 96 × to generate low-level feature maps. Afterward, to obtain multi-scale global features of pavement cracks, we construct a four-stage Transformer module network with a scaling step of 2 × for modeling long-distance dependencies. However, the token partitioning in the Transformer module causes the deficiency of local feature representations. Thus, to extract fine-grained local contextual information, the local enhancement module is appended to the Transformer module in each stage.

Secondly, the decoder and the skip connections are utilized to make full use of rich and multi-scale contextual features extracted from the encoder path and then extract deeper contextual features. Concretely, the decoder starts with an efficient up-sampling module to recover the spatial size of feature maps and decrease channel number by 2 × simultaneously. Then, the Transformer module is combined to capture deep global features. Due to the loss of detailed spatial information in the hierarchical down-sampling operation in the encoder, it is insufficient to restore the detailed spatial information only through the up-sampling operations. Thus, an aggressive operation is designed to transmit both fine-grained local contextual information and global semantic information from both the local enhancement modules and the Transformer modules to the corresponding decoder stages together by element-wise addition operations. Finally, to further improve the accuracy of segmentation maps, the defect rectification module is embedded into the network in a deep supervision manner for hard sample recognition.

2.2. Encoder path

2.2.1. Convolution stem

The Transformer networks perform poorly in processing high-resolution input images. Moreover, it is challenging to optimize because of the employment of patch partition and embedding (*patchify*) stem that are implemented by large-stride convolutions (Xiao et al., 2021). To address these issues and improve the crack detection performance of the Transformer networks, we replace the patchify stem with a convolution stem to extract low-level local contextual information. As presented in Fig. 2, an input image, $\boldsymbol{X_S} \in \mathbf{R}^{224\times224\times1}$ where 224, 224, and 1 denote the width, height, and channel number of the input image, is processed by two stages of convolution operations. There are two units, i.e., a convolution (Conv) unit and a depthwise convolution (DWConv) unit in the convolution stem. As seen in Fig. 2, the Conv unit, which is composed of a convolutional layer, a batch normalization operation, and a Rectified Linear Unit (ReLU) function, aims to extract preliminary local features. Note that, at the beginning of each stage, the channel number and spatial resolution of the feature maps are up-sampled and down-sampled using a convolution layer with the kernel size of 3 and stride of 2. Afterwards, the feature map obtained by the first Conv unit is then fed into the second convolution stage, composed of one Conv unit and one DWConv unit. The DWConv unit, including a 3 × 3 depthwise convolution layer and a Sigmoid activation function, aims to serves as a position encoding operation for providing position information. Finally, a multiplication operation is implemented between the feature maps obtained by the Conv and DWConv units, respectively, to finally output the feature map, denoted as $\boldsymbol{Y_S} \in \mathbf{R}^{96\times56\times56}$.

2.2.2. Transformer module

To enlarge the receptive field, some models tend to adopt a deeper network architecture or attention mechanism, such as the ResNet backbone or spatial and channel attention blocks. However, these schemes suffer from low efficiencies and high computational costs. Fortunately, the Transformer network can obtain global receptive fields with low costs and fewer convolution layers because of the employment of the multi-head self-attention mechanism. As shown in Eq. (1), the Transformer module is composed of an efficient multi-head self-attention unit (EMSA) (Liu et al., 2021), a layer normalization (LN), and a

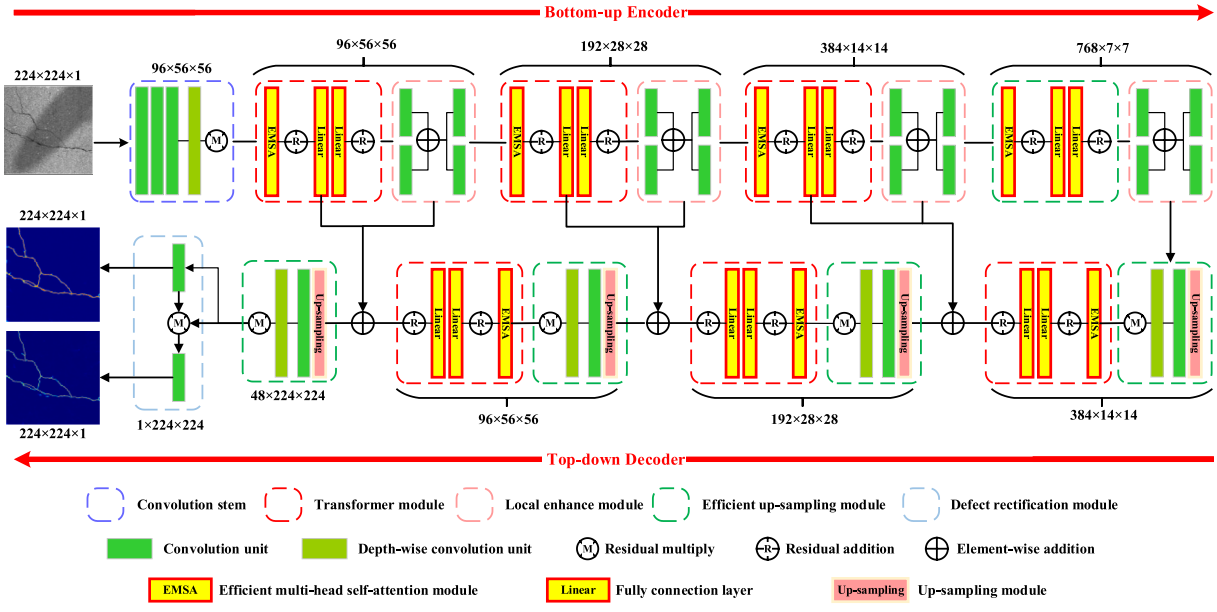


Fig. 1. Workflow of the locally enhanced Transformer pavement crack detection network.

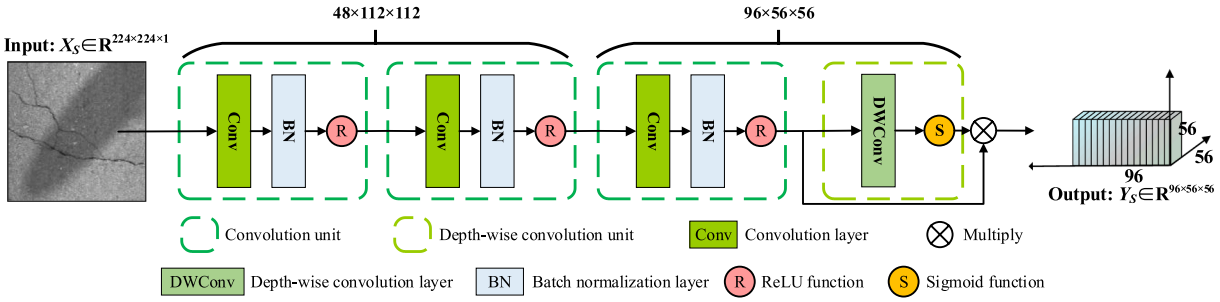


Fig. 2. Architecture of the convolution stem module.

feed-forward network (FFN):

$$\boldsymbol{YT}^{l} = \mathrm{LN}(\boldsymbol{YT}^{l-1} + \mathrm{EMSA}(\mathrm{LN}(\boldsymbol{YT}^{l-1}))) \tag{1}$$

$$\boldsymbol{YT}^{l} = \boldsymbol{YT}^{l} + \mathrm{FFN}(\boldsymbol{YT}^{l}) \tag{2}$$

$$\mathrm{LN}(\boldsymbol{YT}^{l}) = \begin{pmatrix} W_{11} & \cdots & W_{1n} \\ \vdots & \ddots & \vdots \\ W_{m1} & \cdots & W_{mn} \end{pmatrix} \times \begin{bmatrix} y_1^l \\ \vdots \\ y_n^l \end{bmatrix} + \begin{bmatrix} b_1^l \\ \vdots \\ b_n^l \end{bmatrix} \tag{3}$$

where $\boldsymbol{Y_T}^{l-1}$ and $\boldsymbol{Y_T}^{l}$ represent the outputs of the $(l\text{-}1)$-th and l-th Transformer modules, respectively. The FFN is composed of two stacked linear convolution layers (Eq. (3)).

As shown in Fig. 3, assuming that $\boldsymbol{X_T} \in \mathbf{R}^{c\times h\times w}$ and $\boldsymbol{Y_T} \in \mathbf{R}^{c\times h\times w}$ denote the input and output feature maps, where c, w, and h represent the number of the channel, width, and height of the input feature map, respectively. The input feature map $\boldsymbol{X_T}$ is first projected to generate a feature map $\boldsymbol{Q} \in \mathbf{R}^{c\times h\times w}$. Moreover, a DWConv operation with a kernel size of 3 × 3 and a stride of 2 is adopted to down-sample the input feature map by a factor of 2 in spatial size. Then, the resultant feature map ($c \times h' \times w'$, where h' and w' represent the height and width of the convoluted feature map) is reshaped to obtain the size of $(n' \times c)$ for the following linear layers, where $n'=h' \times w'$. Afterward, the reshaped feature map is projected in parallel by two linear layers to generate feature maps $\boldsymbol{K} \in \mathbf{R}^{n'\times c}$ and $\boldsymbol{V} \in \mathbf{R}^{n'\times c}$. Following, $\boldsymbol{Q} \in \mathbf{R}^{c\times h\times w}$, $\boldsymbol{K} \in \mathbf{R}^{n'\times c}$, and $\boldsymbol{V} \in \mathbf{R}^{n'\times c}$ are reshaped to obtain $\boldsymbol{Q} \in \boldsymbol{R}^{k\times n\times c_k}$, $\boldsymbol{K} \in \boldsymbol{R}^{k\times c_k\times n'}$, and $\boldsymbol{V} \in \boldsymbol{R}^{k\times n'\times c_k}$, respectively, where k denotes the number of heads, and $c_k = c/k$. Then, the interactions among different heads are computed by:

$$\mathrm{EMSA}_{\{\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\}}(\boldsymbol{XT}) = \mathrm{IN}\left(\mathrm{softmax}\left(\mathrm{Conv}\left(\frac{\boldsymbol{QK}^{T}}{\sqrt{c_k}}\right)\right)\right)\boldsymbol{V} \tag{4}$$

where Conv(●) and IN(●) represent a 1 × 1 convolution layer and an instance normalization operation, respectively. Next, the same as Eqs. (1) and (2), the input and output feature maps of the EMSA unit are element-wisely concatenated and then fed into a linear unit, composed of two stacked linear layers. Finally, the input and output features of the linear unit are concatenated to obtain the final crack feature map $\boldsymbol{Y_T^l}$ in this stage.

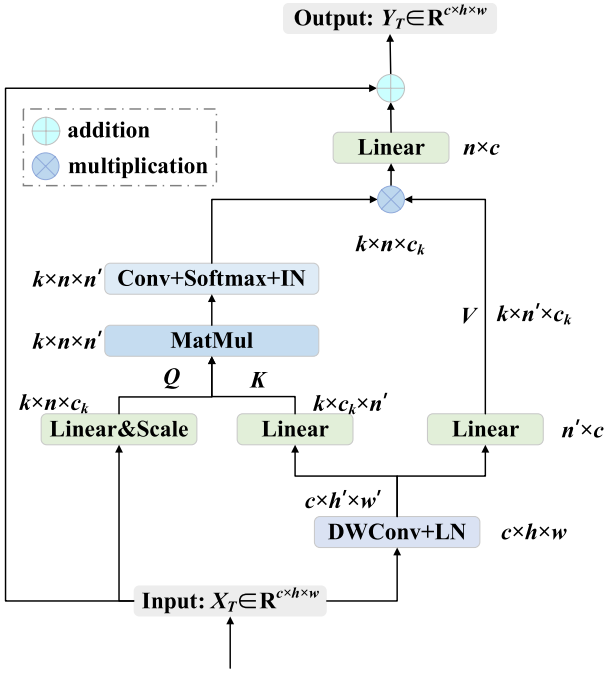


Fig. 3. Architecture of the EMSA unit.

2.2.3. Local enhancement module

The Transformer-based network performs effectively in modeling long-range dependencies but is not good at extracting fine-grained local contextual information because the full consideration is not given to the interactions within patches. Even though many modifications (Han et al., 2021; H. Liu et al., 2021; Dang et al., 2022) have been proposed, they are proved to be inefficient with high memory costs, which limit the applications of the Transformer scheme in pavement crack detection. However, the convolutional operation is effective in modeling local dependencies. Therefore, a local enhancement module is built based on full convolutional operations to obtain sufficient local contextual information.

As shown in Fig. 4, the proposed local enhancement module includes two stacked convolution blocks, each of which consists of a 1 × 1 convolution unit and a 3 × 3 convolution unit with the stride of 1. Concretely, the input feature map, $\boldsymbol{X_L}$, is first filtered by a 1 × 1convolution unit and a 3 × 3 convolution unit in a parallel manner, to output two subsets of feature maps, $\boldsymbol{F_{1\times1}^1}$ and $\boldsymbol{F_{3\times3}^1}$, respectively. Then, these two subsets of feature maps are combined through element-wise summation to generate a feature map $\boldsymbol{F_{add}}$. This procedure is repeated one more time for the feature map, $\boldsymbol{F_{add}}$, to output the final feature map, $\boldsymbol{Y_L}$, in this stage. It is worth noting that, the adoption of the 1 × 1 convolution unit aims to introduce more non-linear activations into the classical 3 × 3 convolution unit by enabling cross-channel information interactions without spatial size reduction.

2.3. Decoder path

2.3.1. Efficient up-sampling module

The recovery of high-resolution features is essential for dense prediction, especially for the segmentation of tiny objects, such as pavement cracks. However, commonly employed up-sampling strategies, such as the checkboard artifacts of transpose convolutions and the data-independent of interpolation upsampling, have been proved to be insufficient. To restore the detailed crack features, we propose an efficient up-sampling module by integrating feature expansion and sub-pixel convolution-based PixelShuffle methods (Shi et al., 2016). As

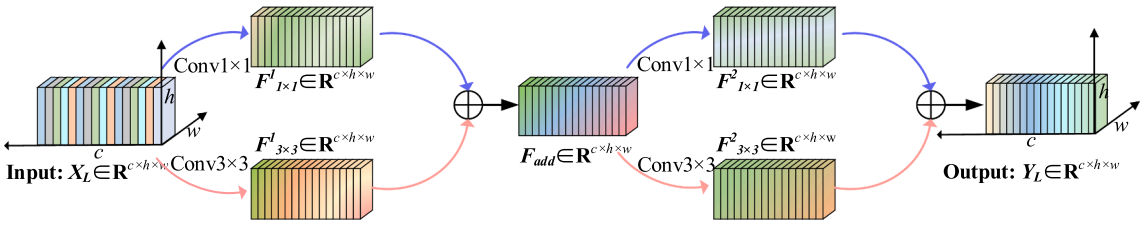


Fig. 4. Architecture of the local enhancement module.

shown in Eqs. (5) and (6), the up-sampling module adopts l-1 convolution layers to increase the number of feature channels from C to $C \times r^2$ (C is the number of feature channels, r is the up-sampling ratio), while maintaining the spatial size constant. Then, a sub-pixel convolution layer is applied to upscale the size of low-resolution feature maps while down-scaling the number of the feature channels (see Eq.(7)). The above operations are formed by:

$$f^1(x) = \phi(W \times x + b) \tag{5}$$

$$f^{l-1}(x) = \phi\left(W \times f^{l-2}(x) + b\right) \tag{6}$$

$$f^l(x) = SP\left(W \times f^{l-1}(x) + b\right) \tag{7}$$

where ϕ is an activation function, W and b are learnable network weights and biases, $SP(\bullet)$ denotes the sub-pixel convolution operator (with the stride of $\frac{1}{r}$) which rearranges the input feature map from the size of $(r^2 \times c, h, w)$ to $(r \times c, r \times h, r \times w)$, and r is the up-sampling ratio.

Specifically, the output feature maps of the PixelShuffle unit are fed into the 1 × 1 convolution block to reduce the number of feature map channels. Then, the spatial-attention block is employed to provide position encoding for the following Transformer module. For the sake of enhancing the position information of the features, the up-sampled feature maps are element-wisely combined with the reference feature map obtained from the encoder path. Note that skip connections, ubiquitously utilized in CNN-based or Transformer-based networks (Cao et al., 2021; Gao et al., 2021), performed poorly on feature representation because of a lack of global or local information. Thus, our up-sampling module could transmit the fine-grained local contextual information and the global semantic information obtained from the local enhancement modules and the Transformer modules, respectively, to the corresponding decoder stages.

2.3.2. Defect rectification module

To distinguish hard samples without introducing extra memory overhead and heavy computational costs, we propose a defect rectification module based on deep supervision and spatial attention, as shown in Fig. 5. First, the input feature map, $\boldsymbol{F^1}$, is fed into a 1 × 1 convolution layer followed by a sigmoid function to predict the first probability map P1. Next, the rectified feature map F2 is obtained by the element-wise multiplication between the probability map P1 and the input feature map F1. Afterward, the second prediction $\boldsymbol{P^2}$ is also generated via a 1 × 1 convolution layer followed by the sigmoid function. Iteratively, the feature map, $\boldsymbol{P^1}$, gradually depresses negative regions while enhancing the positive predictions. Finally, the predictions, $\boldsymbol{P^1}$ and $\boldsymbol{P^2}$, are supervised by the ground-truth labels simultaneously by the following loss function $\boldsymbol{L_s}$:

$$L_s = \sum_{k=1}^{2} \lambda_k L\left(\boldsymbol{P^k}, \boldsymbol{M}\right) \tag{8}$$

where, L, $\boldsymbol{M}$, and L_s denote the Focal Loss function, the ground-truth, and the loss value, respectively. λ and k denote the punishment coefficient and the k-th prediction probability map, respectively. Here, we set λ_1 as 0.5 and λ_2 as 1.

3. Experiments and results

3.1. Datasets

The models in our study were trained and evaluated on four publicly published benchmarks, i.e., Stone331, CrackLS315, CrackTree260, CrackWH100 (Zou et al., 2019), and one newly constructed CrackNJ156 dataset in this paper, as listed in Table 1. Pavement images in the four datasets were captured by various types of equipment under different

Table 1
Open available crack datasets.

Dataset Name	Original Images/#	Illumination	Size/pixels	Equipment
Stone331	331	Visible light	512 × 512	Area-array camera
CrackNJ1156	156	Nature and street lamp	512 × 512	Smartphone
CrackLS315	315	Laser	512 × 512	Area-array camera
CrackWH100	100	Visible light	512 × 512	Linear array camera
CrackTree260	260	Visible light	960 × 720/ 800 × 600	Area-array camera

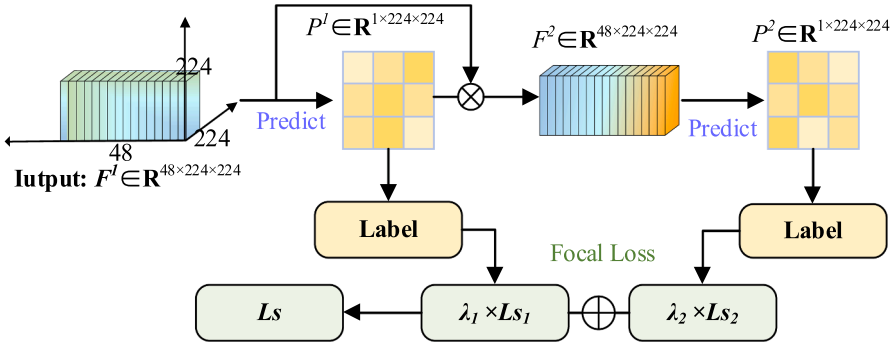


Fig. 5. Architecture of the defect rectification module.

scanning patterns and illumination conditions, such as laser, LED light, and natural illumination.

To advance the progress of pavement crack detection, we constructed a new crack dataset, named CrackNJ156, which included 156 images with the size of 512 × 512 pixels captured by a smartphone at the campus of Nanjing University of Information Science and Technology (NUIST), Nanjing, China, in 2020 and 2021. Compared with other publicly available datasets, the CrackNJ156 dataset has more heterogeneities. We collected crack images from road pavements covered by various materials, such as asphalt, concretes, stones, and terrazzo. Moreover, these images were captured under various weather and season conditions, such as rain and freezing. Additionally, to simulate different illumination conditions during day and night, crack images were also captured under the natural illumination and street lamplight. After data collection, we used Photoshop software and manually annotated the crack images according to their actual widths, rather than the single-pixel width adopted by the above publicly-published datasets.

Considering the representativeness and generalization of training samples, we selected 260 images from the CrackTree260 dataset and 315 images from the CrackLS315 images to form the training and validation subsets, where 80% of images were used for training and 20% for validation. All images of the Stone331, CrackNJ156, and CrackWH100 datasets were used for testing. To further enlarge the training set, data augmentation was performed on the training and validation subsets. Specifically, we first flipped each image in vertical and horizontal directions and then rotated the flipped images in eight directions from 10 to 80 degrees at an interval of 10 degrees. Then, we cropped each of the flipped images into five sub-images with the size of 224 × 224 pixels. After data augmentation, each training image was finally transformed into 135 sub-images. Notably, we only kept the augmented crack samples with lengths larger than five pixels.

3.2. Implementation details

The proposed LETNet was built on a Pytorch framework with a single NVIDIA RTX 2070 GPU, and its weights were initialized by a Kaiming initialization strategy and updated by an AdamW optimizer. The initial learning rate for all layers was set to 1×10^{-4} and divided by 10 and the mini-batch size was set to 4 in each iteration.

Crack detection can be considered a binary classification task, but crack pixels in a pavement image usually account for a minority portion of all image pixels, which is prone to class imbalance. To address this problem, some works assigned weights to both crack pixels and background pixels (Ji et al., 2020). However, Zou et al. (2019) found that crack pixels with larger weights could lead to more false-positive predictions. Thus, we adopted a focal loss function (L_F) (Lin et al., 2017) to measure the prediction errors by:

$$LF(\widehat{P}) = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum_{1}^{\mathrm{H}} \sum_{1}^{\mathrm{W}} -\alpha\left(1-\widehat{P}_{ij}\right)^{\gamma} \times \log\left(\widehat{P}_{ij}\right), \qquad \widehat{P}_{ij} = \begin{cases} P_{ij} & \text{if } M = 1 \\ 1 - P_{ij} & \text{otherwise} \end{cases}. \tag{9}$$

where H and W denote the height and width of the detected images. P_{ij} denotes the prediction probability of the proposed model, and M represents the ground-truth label. $\alpha(1\text{-}\widehat{P}_{ij})^{\gamma}$ represents a modulating factor. α and γ are balanced variant and tunable focusing parameter, respectively. In our work, α and γ were set to 0.25 and 2, respectively.

To quantitatively evaluate the performance of our LETNet, we used four evaluation metrics, i.e., precision, recall, optimal dataset scale (ODS), and IoU.

3.3. Overall performance and efficiency

To demonstrate the performance of the LETNet on crack detection, we compared it with seven current convolution-based and Transformer-based semantic segmentation models, i.e., UNet (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), DeepLabV3+ (Xception) (Chen et al., 2018), DeepCrack (Zou et al., 2019), FPHBN (Yang et al., 2020), CRANet (Wan et al., 2021), and SwinUNet (Cao et al., 2021). For a fair comparison, all models were trained on the same datasets with the same training strategy from the scratch.

It is worth noting that a pavement crack typically shown on the CCD image is a linear structure with a certain width. Normally, the manually annotated cracks were presented with the single-pixel width. To accurately evaluate the predicted cracks, we determined the predicted positive pixels as the true ones when their distances were no more than two pixels away from the annotated crack pixels. Table 2 lists the overall performance of all the comparative models on three testing datasets.

3.3.1. CrackWH100 dataset and Stone 331

As shown in Table 2, all methods achieved promising performances on the CrackWH100 dataset while the LETNet presented a significant performance boost over the other methods based on the four evaluation metrics. In detail, the LETNet achieved an increase by 2.27% to 4.19% in the ODS values, respectively, compared with the other seven compared models. The UNet obtained a moderate performance, with an overall crack detection on the ODS value of 92.63%. The self-attention-embedded CRANet ranked the third with an ODS value of 91.22%. Furthermore, the crack samples in the Sthone331 dataset were fine crack lines with low contrasts, rather different from the training dataset. Thus, all methods suffered from performance degradation, especially in the recall values. As shown in Table 2, the LETNet achieved the best performance with an ODS of 92.94% and an IoU of 94.07%, compared with the other seven methods.

Fig. 6 shows the crack detection results obtained from the comparative methods. Visual inspection demonstrated the superiority of the proposed LETNet over the other seven methods on the crack detection tasks, particularly for the cracks with varied types, sizes, as well as illumination and material conditions. Specifically, as shown in the first and last rows in Fig. 6, our LETNet effectively delineated very shallow cracks under low-contrast conditions while the other methods extracted fragmented cracks with low confidence. Note that the SegNet performed less effectively for detecting cracks from the pavement images of gravel roads (see last three rows in Fig. 6) because the joint boundaries of the gravels and the background were misclassified as cracks, as shown by the red dash bounding boxes. It also can be observed that both SegNet and SwinUNet performed less effectively in the CrackWH100 and Stone 331 datasets, compared with UNet, CRANet, and DeepCrack. This is because of the loss of detailed information resulting from the absence of skip connections in the SegNet and pure Transformer operations in the SwinUNet. For the crack images shown in the second row, there is a large oil/water stain area with dark spectral intensity. Due to the texture similarity between the stain area boundary and the cracks of interest, most crack detection methods tended to produce false-positive predictions by misclassifying the oil/water stain boundaries as cracks. When dealing with the cracks filled with ice, as shown in the third row, the LETNet correctly and completely delineated the cracks, while the other comparative methods failed since there are very few 'bright cracks' in the training set. Thus, it can be concluded that the LETNet has a strong generalization capability.

3.3.2. CrackNJ156 dataset

Our CrackNJ156 dataset contains the cracks on the pavements coated with various materials, e.g., asphalt, terrazzo, and concretes, as well as the cracks on the plaster walls. Moreover, the crack images in the CrackNJ156 also contain heavy noises (such as shadows, water or oil stains, leaves, and fine grooves on concrete pavements). This dataset also includes dramatically changed illumination conditions and various crack width sizes, which are more complex than other publicly-published datasets. Thus, as seen in Fig. 7, according to the ODS

Table 2
Model performance of crack detection on three testing datasets (The bold and the blue color represent the worst and best results, respectively).

Model	CrackWH100				Stone331				CrackNJ156			
	Precision/%	ODS/%	Recall/%	IoU/%	Precision/%	ODS/%	Recall/%	IoU/%	Precision/%	ODS/%	Recall/%	IoU/%
UNet	90.32	92.63	95.05	93.35	92.54	91.76	91.00	93.26	68.02	72.81	78.31	78.02
SegNet	88.57	90.11	91.70	92.83	**67.31**	**72.83**	**79.34**	**79.40**	61.15	66.96	73.98	74.31
DeepLabV3+	89.54	90.07	**90.60**	92.74	89.64	87.30	85.08	89.57	60.74	64.27	68.24	72.52
DeepCrack	89.25	90.70	92.20	93.39	91.54	91.54	91.54	92.92	**77.27**	**57.97**	**46.38**	**69.75**
FPHBN	88.34	90.36	92.48	92.70	89.96	90.33	90.70	91.93	**54.63**	61.65	70.73	71.14
CRANet	89.43	91.22	93.08	93.83	91.22	91.59	91.97	92.84	70.43	74.27	78.56	79.28
SwinUNet	**88.25**	**89.96**	91.73	**92.67**	82.86	84.18	85.55	86.90	66.72	64.39	62.22	72.84
LETNet	**92.84**	**94.87**	**96.99**	**95.59**	**93.04**	**92.94**	**92.85**	**94.07**	73.15	**75.79**	**78.62**	**80.08**

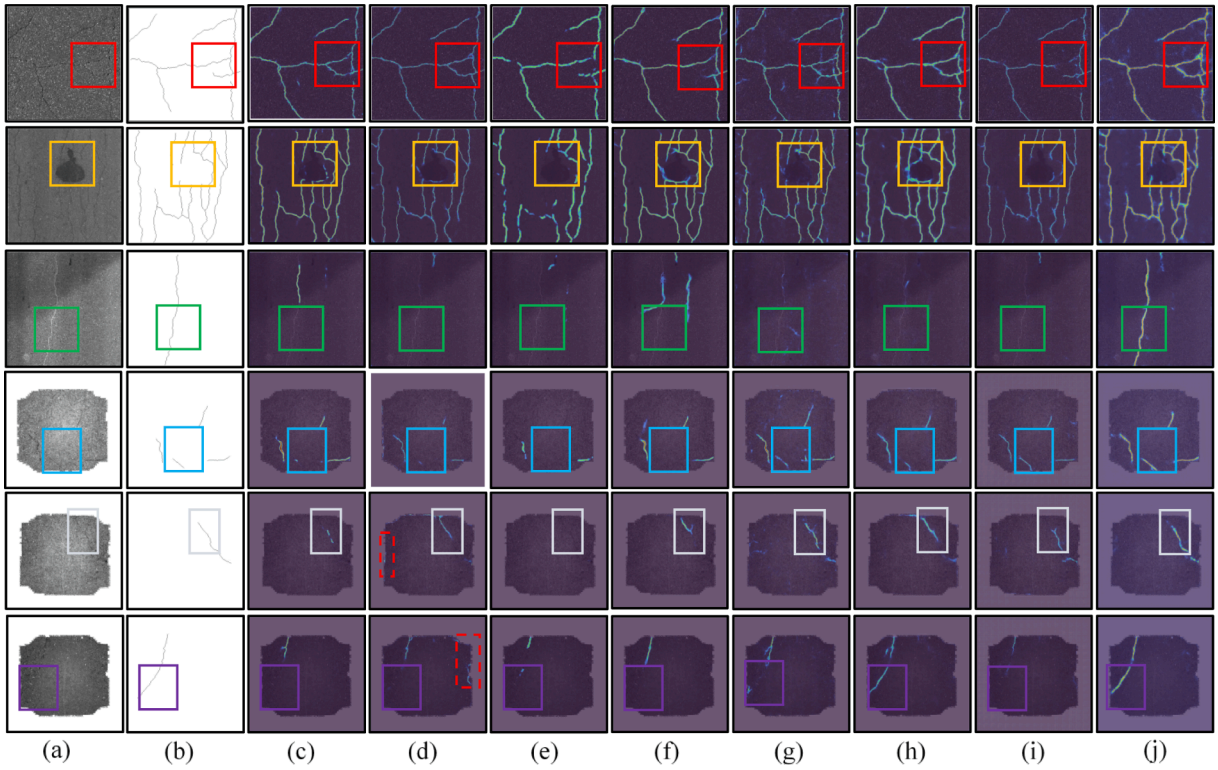


Fig. 6. Pavement crack detection results by eight methods on the CrackWH100 and Stone331 datasets (with three images from each dataset). (a) Input image. (b) Ground truth. (c) UNet. (d) SegNet. (e) DeepLabV3+. (f) DeepCrack. (g) FPHBN. (h) CRANet. (i) SwinUNet. (j) LETNet.

values, the CrackNJ156 dataset has the best distinguishability in performance validation, compared with the CrackWH100 and Stone331 datasets. In this way, our CrackNJ156 dataset contributes to evaluating the robustness and generalization of different crack detection models.

For the overall detection accuracy on the CrackNJ156 dataset, the LETNet outperformed all other methods with an overall ODS value of 75.79%, as shown in Table 2. Also, the CRANet outperformed the UNet and obtained an overall ODS value of 74.27%. Although the UNet and the LETNet achieved similar crack detection performances on the CrackWH100 and Stone331 datasets, their crack detection results on the CrackNJ156 dataset were quite inconsistent with a chasm of about 2.98% on the ODS value. It also can be observed from the comparison between the CRANet and the UNet that the attention mechanism could improve crack detection performance on the complex pavement images because of its large receptive fields and long-range dependency modeling.

Fig. 8 presents the qualitative crack detection results obtained from comparative methods on the CrackNJ156 dataset. As shown in the first two rows, since the images contain two long landmarks, all of the other seven comparative models misclassified the boundaries of the landmarks as cracks, as highlighted in the red boxes. As seen in the third and fourth two rows, the images contain fine grooves (long and dark linear structures), which are similar to cracks. As shown by the green boxes, the LETNet obtained better crack detection performance with fewer false-positive predictions, compared with the other methods. The last two rows represent the pavement images with wide cracks and varied spectral contrast. The LETNet achieved more complete predictions than the other methods. Moreover, as seen in Fig. 6(e) and Fig. 8(e), the cracks obtained by the DeepLabV3+ have low continuities due to the multi-scale atrous convolutions. Also, as shown in Fig. 6(g) and Fig. 8(g), the FPHBN brought more false-positive predictions into the final detection results due to the shallow feature maps consideration. The DeepCrack achieved a relatively competitive precision value of 77.27%, which means it is suitable for detecting cracks from noisy backgrounds.

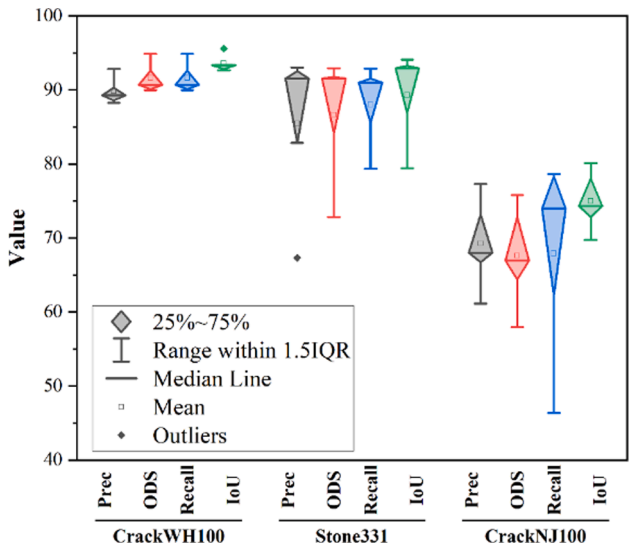


Fig. 7. Comparison of different networks performance on three datasets.

However, the predictions in Fig. 6(f) and Fig. 8(f) show that it has no talent in detecting fine structures and has a low generalization ability.

To further evaluate the overall performance of the LETNet, computational efficiency was analyzed by the following factors: the number of network parameters (Par), computational complexity (CC), and running speed. Table 3 shows the comparative results obtained by the eight models. In terms of the number of network parameters, the SegNet, FPHBN, and UNet possessed a relatively less amount of the network parameters below 20 M, whereas the CRANet and DeepLabV3+ contained a large amount of the network parameters over 50 M. The LETNet possessed the number of network parameters of about 43.78 M, but only contained the floating-point operations (FLOPs) of 23.92, smaller than those of the other methods, which indicated that our LETNet has low computational complexity. Furthermore, in terms of running speed, the SwinUNet obtained the fastest speed with an FPS of 46.40 because of the lower computational complexity and fewer parameters. In contrast, our LETNet obtained an FPS of 30.80, lower than those of SwinUNet and FPHBN because of the inset of convolution operations. Although the running time of our LETNet was slightly larger than the SwinUNet and FPHBN, the LETNet obtained higher accuracy in crack detection. Through computational performance analysis, we conclude that the LETNet provides an efficient and effective solution for pavement crack detection tasks.

In summary, CNN-based or pure Transformer-based networks performed poorly in the recognition of hard samples because of a lack of global or local contextual information. Furthermore, atrous convolution operations could cause the loss of detailed information and accurate boundaries of pavement cracks, thereby generating discontinuous predictions. Even if the self-attention modules were used, the CNN-based CRANet model showed non-significant advantages in recognizing hard samples. As seen in Table 3, comparatively, the CRANet obtained low efficiency for crack detection because the attention modules have been introduced. Hence, these models that only employed atrous convolutions, feature pyramids, or self-attention modules to enlarge receptive fields might fail to deal with cracks under complex pavement scenarios. In contrast, our LETNet can effectively extract cracks under complex road scenarios because the local details and global contextual features are captured and integrated collaboratively.

Table 3
A comparison of the model properties.

Models	Par/M	CC/GFLOPs	Speed/FPS
UNet	17.27	320.32	25.68
SegNet	16.31	601.76	24.24
DeepLabV3+	54.61	31.74	25.56
DeepCrack	30.90	1094.78	4.92
FPHBN	16.61	31.04	38.40
CRANet	51.63	63.04	14.48
SwinUNet	27.17	52.52	46.40
LETNet	43.78	23.92	30.80

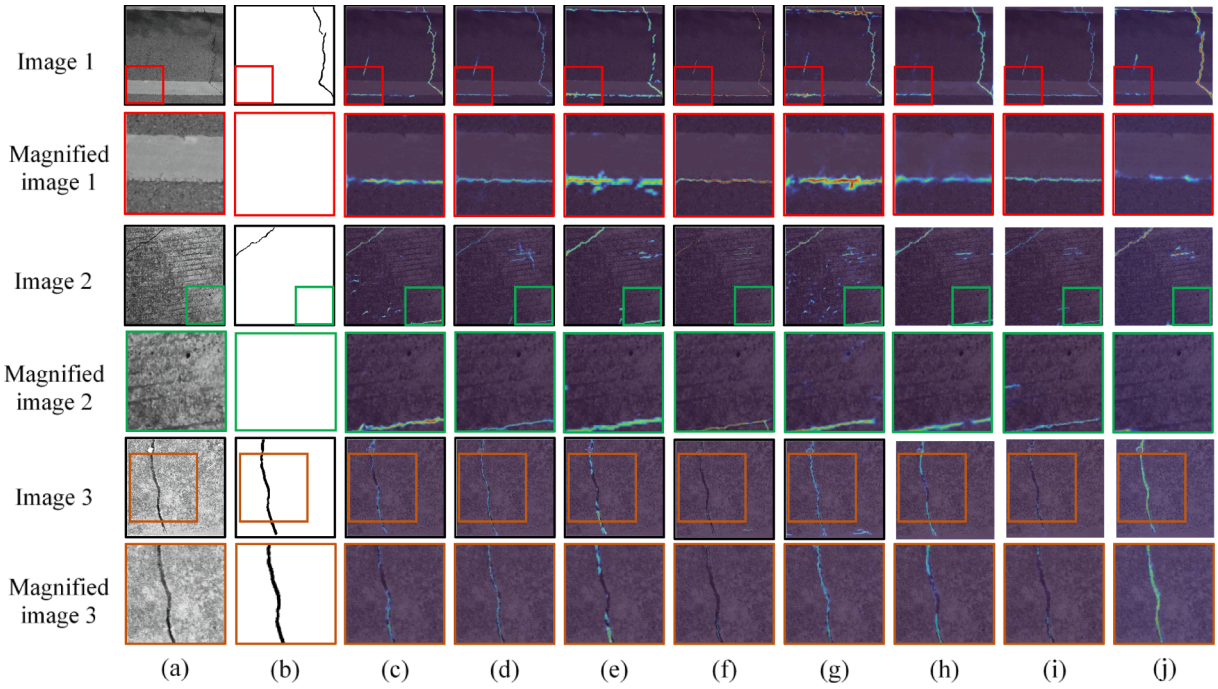


Fig. 8. Detection results comparison by eight methods on CrackNJ156 dataset. (a) Input image. (b) Ground truth. (c) UNet. (d) SegNet. (e) DeepLabV3+. (f) DeepCrack. (g) FPHBN. (h) CRANet. (i) SwinUNet. (j) LETNet.

3.4. Ablation study

In this section, we performed ablation studies to reveal the effectiveness of the different modules in the LETNet. More specifically, the convolution stem block, the Transformer module, the local enhancement module, the up-sampling module, and the defect rectification module were replaced with or removed from the LETNet, respectively.

Effect of convolution stem. We replaced the convolution stem with the patchify stem proposed in the original ViT model (Dosovitskiy et al., 2020), and named the resultant network as the LETNet-Stem. The experimental results in Table 4 demonstrated that, compared with the LETNet, the LETNet-Stem obtained a decrease of 0.50%, 0.43%, and 0.60% on the ODS, precision, and recall values, respectively. It indicates that the convolution stem used at the very beginning of the Transformer-based model improved the optimization of the ViT model and contributed to the performance improvement of crack detection.

Effect of Transformer module. We removed the Transformer modules from the LETNet, and named this modification as the LETNet-Trans. As shown in Table 4, the LETNet-Trans obtained a huge decrease in precision, recall, and ODS values about 6.68%, 4.54%, and 5.69%, respectively. It can be seen that this variation has a great impact on the precision, compared with other metrics. Thus, it implies that the Transformer module helps the model repress false-positive interferences while identifying more hard samples by the global semantic cues capturing.

Effect of local enhancement module. To demonstrate the effectiveness of the local details obtained by the LETNet on crack detection performance improvement and convergence acceleration, we removed the local enhancement modules from the LETNet. Thus, the resultant network was similar to a pure Transformer network, and wasnamed the LENET-LE. As seen in Table 4, the LETNet-LE performed even worse than the LETNet-Trans with a decrease of 7.68%, 6.38%, 5.09%, and 4.53% on recall, ODS, precision, and IoU values, respectively. It demonstrates that the local enhancement modules are capable of extracting fine-grained local contextual information. It is also worth noting that the LENET-LE is hard to converge, which requires more training epochs.

Effect of defect rectification module. We removed the defect rectification module from the LETNet, and named this modification as the LETNet-DR. Note that the defect rectification module brought no extra computation or memory costs, where the parameter size and the training efficiency were the same as those of the LETNet. Moreover, the LETNet-DR obtained a decrease of 1.07% and 0.17% on recall and IoU, respectively. Although the defect rectification module helped the LETNet recognize more hard samples, it also brought some false-positive predictions with a performance degradation of 0.22% on precision. This could be explained by the fact that the defect rectification module assigns high confidence to ambiguous predictions.

Effect of different up-sampling strategies. To demonstrate the superiority of the proposed up-sampling strategy, we replaced it with a widely-utilized method (e.g., DUpsampling), and named the modification as the LETNet_Dup. As shown in Table 4, the crack detection performance of the LET_Dup was lower than that of the LETNet on the ODS value by 0.44%. However, the running efficiency of the LETNet was 11.2 FPS, which was higher than that of the LETNet_Dup. Thus, the proposed up-sampling strategy is effective to improve crack detection performance.

Table 4
The ablation study of testing the LETNet model performance with different modules.

Models	Precision/%	ODS/%	Recall/%	IoU/%
LETNet	**86.34**	**87.87**	**89.49**	**89.91**
LETNet-Stem	85.91	87.37	88.89	89.61
LETNet-Trans	79.66	82.10	84.95	86.31
LETNet-LE	80.25	81.04	81.91	85.38
LETNet-DR	86.56	87.46	88.42	89.74
LETNet_Dup	86.00	87.43	89.02	89.53

4. Conclusion

To accurately and efficiently extract pavement cracks from the noisy background, this paper presents a locally enhanced Transformer based network (LETNet). In this network, the Transformer modules were employed to model long-range dependencies, and the local enhancement modules were utilized to supplement fine-grained local contextual information and make the Transformer easier to converge. Then, the sub-pixel convolution-based upsampling module and the spatial attention mechanism-based defect rectification module were designed to effectively restore details of cracks and recognize hard samples. The LETNet has been extensively evaluated on three datasets for pavement crack detection. Quantitative assessments and qualitative inspections demonstrate that the developed LETNet performs excellently in identifying varying patterns of pavement cracks under various road and weather scenarios. In addition, ablation studies and comparative analyses prove that the LETNet is a promising solution for pavement crack detection when dealing with pavement cracks with a quantity of noise, varied sizes and patterns, complex illumination conditions, and diverse road surface materials.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grants 41971414, 62076107, 42101451, in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX21_1011, and also in part by the Emerging Interdisciplinary Project of Central University of Finance and Economics.

CRediT authorship contribution statement

Zhengsen Xu: Conceptualization, Software, Methodology, Writing – original draft, Writing – review & editing. **Haiyan Guan:** Conceptualization, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition. **Jian Kang:** Data curation, Writing – review & editing, Funding acquisition. **Xiangda Lei:** Investigation, Writing – review & editing. **Lingfei Ma:** Validation, Formal analysis, Investigation, Supervision, Funding acquisition. **Yongtao Yu:** Writing – review & editing. **Yiping Chen:** Writing – review & editing. **Jonathan Li:** Resources, Supervision, Writing – review & editing.

Declaration of Competing Interest

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

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Bhattacharya, G., Mandal, B., Puhan, N.B., 2021. Multi-Deformation Aware Attention Learning for Concrete Structural Defect Classification. IEEE Trans. Circuits Syst. Video Technol. 31 (9), 3707–3713. https://doi.org/10.1109/TCSVT.2020.3028008.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M., 2021. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: In: European Conference on Computer Vision, pp. 833–851. https://doi.org/10.1007/978-3-030-01234-2_49.

Cho, S., Park, S., Cha, G., Oh, T., 2018. Development of Image Processing for Crack Detection on Concrete Structures through Terrestrial Laser Scanning Associated with the Octree Structure. Appl. Sci. 8, 2373. https://doi.org/10.3390/app8122373.

Cui, X., Wang, Q., Dai, J., Xue, Y., Duan, Y., 2021. Intelligent crack detection based on attention mechanism in convolution neural network. Adv. Struct. Eng. 24 (9), 1859–1868. https://doi.org/10.1177/1369433220986638.

Dang, L.M., Wang, H., Li, Y., Nguyen, T.N., Moon, H., 2022. DefectTR: End-to-end defect detection for sewage networks using a transformer. Constr. Build. Mater. 325, 126584. https://doi.org/10.1016/j.conbuildmat.2022.126584.

Dong, H., Song, K., Wang, Q.i., Yan, Y., Jiang, P., 2022. Deep Metric Learning-Based for Multi-Target Few-Shot Pavement Distress Classification. IEEE Trans. Ind. Inform. 18 (3), 1801–1810. https://doi.org/10.1109/TII.2021.3090036.

Dorafshan, S., Thomas, R.J., Maguire, M., 2018. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Constr. Build. Mater. 186, 1031–1045. https://doi.org/10.1016/j.conbuildmat.2018.08.011.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations.

Dung, C.V., Sekiya, H., Hirano, S., Okatani, T., Miki, C., 2019. A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. Autom. Constr. 102, 217–229. https://doi.org/10.1016/j.autcon.2019.02.013.

Fang, J., Qu, B., Yuan, Y., 2021. Distribution equalization learning mechanism for road crack detection. Neurocomputing 424, 193–204. https://doi.org/10.1016/j.neucom.2019.12.057.

Gao, Y., Zhou, M., Metaxas, D.N., 2021. UTNet: A Hybrid Transformer Architecture for Medical Image Segmentation. In: In: International Conference on Medical Image Computing and Computer Assisted Intervention, pp. 61–71. https://doi.org/10.1007/978-3-030-87199-4_6.

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y., 2021. Transformer in Transformer. In: Neural Information Processing Systems.

Hsieh, Y.-A., Yang, Z., James Tsai, Y.-C., 2021. Convolutional neural network for automated classification of jointed plain concrete pavement conditions. Comput.-Aided Civ. Infrastruct. Eng. 36 (11), 1382–1397.

Ji, A., Xue, X., Wang, Y., Luo, X., Xue, W., 2020. An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. Autom. Constr. 114, 103176. https://doi.org/10.1016/j.autcon.2020.103176.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P., 2015. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Adv. Eng. Inform. Infrastructure Computer Vision 29 (2), 196–210. https://doi.org/10.1016/j.aei.2015.01.008.

Li, G., Li, X., Zhou, J., Liu, D., Ren, W., 2021a. Pixel-level bridge crack detection using a deep fusion about recurrent residual convolution and context encoder network. Measurement 176, 109171. https://doi.org/10.1016/j.measurement.2021.109171.

Li, G., Zhao, X., Du, K., Ru, F., Zhang, Y., 2017. Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine. Autom. Constr. 78, 51–61. https://doi.org/10.1016/j.autcon.2017.01.019.

Li, H., Xu, H., Tian, X., Wang, Y., Cai, H., Cui, K., Chen, X., 2020. Bridge Crack Detection Based on SSENets. Appl. Sci. 10, 4230. https://doi.org/10.3390/app10124230.

Li, Q., Zou, Q., Zhang, D., Mao, Q., 2011. FoSA: F* Seed-growing Approach for crack-line detection from pavement images. Image Vis. Comput. 29 (12), 861–872. https://doi.org/10.1016/j.imavis.2011.10.003.

Li, R., Zheng, S., Zhang, C., Duan, C., Wang, L., Atkinson, P.M., 2021b. ABCNet: Attentive bilateral contextual network for efficient semantic segmentation of Fine-Resolution remotely sensed imagery. ISPRS J. Photogramm. Remote Sens. 181, 84–98. https://doi.org/10.1016/j.isprsjprs.2021.09.005.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. Focal Loss for Dense Object Detection. In: In: IEEE International Conference on Computer Vision, pp. 2980–2988.

Liu, H., Miao, X., Mertz, C., Xu, C., Kong, H., 2021. CrackFormer: Transformer Network for Fine-Grained Crack Detection. In: IEEE/CVF International Conference on Computer Vision, pp. 3783–3792.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In: IEEE/CVF International Conference on Computer Vision, pp. 10012–10022.

Mei, Q., Gül, M., 2020. A cost effective solution for pavement crack inspection using cameras and deep neural networks. Constr. Build. Mater. 256, 119397. https://doi.org/10.1016/j.conbuildmat.2020.119397.

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

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: In: Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883.

Shuai, B., Zuo, Z., Wang, B., Wang, G., 2018. Scene Segmentation with DAG-Recurrent Neural Networks. IEEE Trans. Pattern Anal. Mach. Intell. 40 (6), 1480–1493. https://doi.org/10.1109/TPAMI.2017.2712691.

Tran, T.S., Tran, V.P., Lee, H.J., Flores, J.M., Le, V.P., 2020. A two-step sequential automated crack detection and severity classification process for asphalt pavements. Int. J. Pavement Eng. 23 (6), 2019–2033. https://doi.org/10.1080/10298436.2020.1836561.

Wan, H., Gao, L., Su, M., Sun, Q., Huang, L., Kara, F., 2021. Attention-Based Convolutional Neural Network for Pavement Crack Detection. Adv. Mater. Sci. Eng. 2021, 1–13. https://doi.org/10.1155/2021/5520515.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In: IEEE/CVF International Conference on Computer Vision, pp. 568–578.

Wu, Z.Y., Kalfarisi, R., Kouyoumdjian, F., Taelman, C., 2020. Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation. Urban Water J. 17 (8), 682–695. https://doi.org/10.1080/1573062X.2020.1758166.

Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollar, P., Girshick, R., 2021. Early Convolutions Help Transformers See Better. Presented at the Advances in Neural Information Processing Systems.

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., Chen, X., 2019. Automatic Bridge Crack Detection Using a Convolutional Neural Network. Appl. Sci. 9, 2867. https://doi.org/10.3390/app9142867.

Xu, Z., Sun, Z., Huyan, J., Li, W., Wang, F., 2021. Pixel-level pavement crack detection using enhanced high-resolution semantic network. Int. J. Pavement Eng. 1–15 https://doi.org/10.1080/10298436.2021.1985491.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., Ling, H., 2020. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. IEEE Trans. Intell. Transp. Syst. 21 (4), 1525–1535. https://doi.org/10.1109/TITS.2019.2910595.

Ying, Z., 2021. 2020 Statistical Bulletin on the Development of the Transportation Industry [WWW Document]. URL http://www.gov.cn/xinwen/2021-05/19/content_5608523.htm.

Zhang, D., Li, Q., Chen, Y., Cao, M., He, L., Zhang, B., 2017. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. Image Vis. Comput. 57, 130–146. https://doi.org/10.1016/j.imavis.2016.11.018.

Zhang, H., Liao, Y., Yang, H., Yang, G., Zhang, L., 2020. A Local-Global Dual-Stream Network for Building Extraction From Very-High-Resolution Remote Sensing Images. IEEE Trans. Neural Netw. Learn. Syst. 33 (3), 1269–1283. https://doi.org/10.1109/TNNLS.2020.3041646.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H.S., Zhang, L., 2021. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. ArXiv201215840 Cs.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., Wang, S., 2019. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. IEEE Trans. Image Process. 28 (3), 1498–1512. https://doi.org/10.1109/TIP.2018.2878966.