# Robust ELM-PID Tracing Control on Autonomous Mobile Robot via Transformer-based Pavement Crack Segmentation

Jianqi Zhang[a,b], Xu Yang[b,c,*], Wei Wang[a], Ioannis Brilakis[d], Diana Davletshina[d] and Hainian Wang[c]

[a]School of Information Engineering, Chang'an University, Xi'an, 710064, China
[b]School of Future Transportation, Chang'an University, Xi'an, 710064, China
[c]School of Highway, Chang'an University, Xi'an, 710064, China
[d]Laing O'Rourke Centre, Engineering Department, Cambridge University, Cambridge, CB2 1PZ, UK

ARTICLE INFO

*Keywords*:
Pavement crack tracing
ELM-PID control
Extreme learning machines
Transformer-based crack segmentation
Embedded deep learning
Autonomous mobile robot

ABSTRACT

Pavement crack tracing is paramount to missions encompassing automated crack sealing for road maintenance. However, existing methods still face several challenges, including the incapability to precisely extract crack trajectories and the challenge of tuning control parameters within intricate backgrounds. To address these limitations, the ViT-S2T network and the ELM-PID control system are proposed for crack tracing. Specifically, the ViT-S2T consists of two branches. The transformer-based feature extraction module (TFEM) integrates multi-head attention mechanism and multi-layer perceptron to capture global contextual crack semantic features. The incoherent segmentation masks (ISM) employs a binary classifier to predict the coarsest irrelevant mask and further performs up-sampling fusion of higher-resolution features. Moreover, the Neural-PID control method is designed to track crack trajectories, combining Extreme Learning Machines (ELM) and Proportional Integral Derivative (PID). The ELM-PID controller utilizes a three-layer backpropagation neural network and proposes the ELM model for adaptive adjustment by predicting the tuning parameters of PID. This framework is applied to real-time visual tracing for edge AI. Extensive tests performed on three arduous datasets of DeepCrack, CFD, and S2TCrack, achieving a precision of 82.76% and mAP@0.5 of 75.63% and speed of 0.0513m/s, demonstrating the superior and robust nature of our approach in pavement crack tracing.

## 1. Introduction

Pavement crack tracing serves as a crucial technical prerequisite for automatic crack sealing [1–3]. Tracing cracks of irregular slender shapes carries a significant weight in the context of unstructured pavement [4–6]. Two subtasks fundamental to accomplishing automated crack tracing are obtaining complete crack trajectories from segmentation and robust tracing control [7–9]. Despite the impressive pace of development and notable strides in these technologies over the last few decades, challenges persist. One issue is that segmenting slender cracks with extreme aspect ratios under adverse scenarios can lead to insufficient extraction of minute features, thus causing crack trajectory fragmentation. Furthermore, the traditional Proportional-Integral-Derivative (PID) controller used in trajectory tracing relies on time-consuming and labor-intensive manual parameter calibration, which fails to efficiently reduce errors on complex trajectories. Current practices for pavement crack repair predominantly rely on manual sealing. The prospect of automated repair in the road maintenance industry presents its own challenges to automated crack tracing. Hence, immediate attention needs to be given to constructing crack tracing control systems that employ embedded deep learning within an autonomous mobile robot for the repair of pavement cracks.

*Corresponding author at: School of Future Transportation, Chang'an University, Xi'an, 710064, China
(X. Yang)
ORCID(s):

Recently, with the rise of deep convolutional neural networks, remarkable progress has been made in both visual object segmentation and path tracing control [10–13]. However, most deep learning methods for object segmentation are developed for clearly defined objects and their performance in segmenting extremely elongated cracks with complex and unstructured road environments has not been well studied. Moreover, path tracing often focuses on regular routes or large-scale autonomous driving scenarios, neglecting the critical task of tracing crack trajectories with complex topological structures (a key technology for automated crack sealing). Crack tracing faces two challenges: the potential loss of crack semantic information, leading to fragmented crack trajectories, and the limited tuning ability and susceptibility to disturbances of PID control parameters. These challenges are elaborated below.

Crack trajectory fragmentation can stem from the inherent locality of convolutional neural networks, whereby these algorithms fail to fully utilize contextual semantic information, leading to partial feature loss [14–17]. If the global context comprehension of crack detectors could be enhanced, the semantic information loss from crack segmentation would be alleviated. Recently, the extraction of features in crack segmentation tasks has attracted increased attention from researchers. Liu et al. embedded a novel Transformer encoder module into an Encoder-Decoder structure, similar to SegNet, proposing an upgraded version of the CrackFormer network designated for pavement crack segmentation, CrackFormer-II, extending its use beyond just fine-grained crack detection tasks [18]. Concurrently, Zhou et al.

advocated for a hybrid semantic segmentation algorithm for tunnel-lining cracks, SCDeepLab, which introduces a joint backbone network based on CNN's reverse residual block and Swin Transformer Block [19]. Additionally, Wang et al. proposed a new SegCrack model for pixel-level crack segmentation using a hierarchical structure Transformer encoder outputting multi-scale features while gradually upsampling and integrating features from the encoder's deepest layer through a top-down path with lateral connections [20]. However, these methods still showcase deficiencies in the fine extraction of crack features and fail to satisfy the need for obtaining complete crack trajectories.

Traditional PID controllers tend to emphasize constant control calibrated via human experience, but they possess weak resistance to interference and are ineffectual at reducing tracing errors [21–23]. To address these challenges, previous studies have proposed an improved Particle Swarm Optimization (PSO) algorithm [24]. This approach optimizes the coefficients used in the Proportional, Integral, and Derivative (PID) controllers to improve the accuracy of trajectory tracing. Guan et al. introduced a TSABFA-PID path-tracing controller, which dynamically adjusts the hyperparameters to manage the control issues of road roller robots in complex operating conditions [25]. Yoon et al. applied the Long Short-Term Memory (LSTM) algorithm based on PID gain value assessment for flight motion in order to fast stabilize a quadrotor when its attitude is disturbed [26]. Zhang et al. proposed an adaptive fuzzy finite-time PID back-stepping control method for the handling of nonlinear system control [27]. Gün et al. used a differential evolution algorithm (DE) to derive the Proportional, Integral, Differential (PID) coefficients with the aim of minimizing the errors induced by the control of quadrotor aircraft [28]. Although these enhanced methods have made significant strides in increasing control precision and positively influencing the optimization of PID control, there are still existing issues such as the challenges in parameter adjustment, limited convergence speed, search capability, adaptability to environmental conditions, and potential over-control, none of which can satisfy the demand for stable control in complex trajectory tracing.

This study proposes a pavement crack tracing control framework designed to address the challenges associated with labor-intensive and time-consuming PID parameter tuning, which often results from manual calibration, and issues arising from the loss of critical crack features during segmentation. The framework integrates a Transformer-based crack segmentation network (ViT-S2T) with an Extreme Learning Machine-PID (ELM-PID) controller, both of which are deployed on mobile devices to enhance the robustness and accuracy of crack tracing in road landscapes. Specifically, the ViT-S2T network incorporates a Transformer-based Feature Extraction Module (TFEM) and Incoherent Segmentation Masks (ISM) to facilitate global context feature extraction and refine the segmentation of cracks. Additionally, the ELM-PID controller combines the principles of an Extreme Learning Machine with PID control, enabling the selection of an effective training algorithm and the optimal number of hidden layer neurons. It also adjusts the PID gain online through the network's self-learning capabilities, thereby minimizing average error and ensuring robust control tracing. Comprehensive experiments were conducted to improve the autonomous tracing performance of individual cracks using a custom-built Autonomous Mobile Robot equipped with an edge computing unit. This research framework facilitates accurate acquisition and tracing of crack trajectories, laying a strong foundation for subsequent tasks such as crack sealing and road maintenance. The contributions of this study are fourfold:

- Addressing the issue of pavement crack tracing for automated crack sealing, a robust pavement crack tracing control framework is proposed.This algorithm effectively facilitates comprehensive crack trajectory and high-performance crack tracing.
- Boosting the performance and efficacy of continuous and comprehensive extraction of trajectory cracks in real-world road environments. A novel network termed ViT-S2T was developed, wherein the TFEM component improves the ability to capture features within a broader context, and the ISM component ensures finer segmentation of the crack trajectory.
- Considering the challenges associated with PID parameter tuning, the ELM-PID controller has been designed. By leveraging the self-learning capabilities of the ELM, the parameters $K_p$, $K_i$, and $K_d$ can be adjusted to determine the most effective control strategy. Consequently, this facilitates robust tracing and control of cracks even in the presence of complex disturbances.
- Demonstrating comprehensive experiments on three challenging datasets: DeepCrack, CFD, and S2TCrack. Our overarching framework was implemented on a proprietary Autonomous Mobile Robot. The results demonstrate that the proposed method achieves state-of-the-art performance.

The remainder of this article unfolds as follows: Section 2 presents the related work drawing from established research in pertinent fields. Subsequently, the Section 3 delves into the intricate designs of our proposed method. Section 4 then validates the efficacy of our methods via conducted experiments. Finally, Section 5 concludes the article with a summary and discussion of potential future developments.

## 2. Related Works

This section reviews the literature relevant to our proposed pavement crack tracing. The state-of-the-art crack segmentation models are first presented, followed by the vision transformer methods and the PID control schemes.

## 2.1. Crack Segmentation

Crack segmentation is a crucial technique for inspecting distress in various infrastructures, such as roads, bridges, tunnels, airports, and buildings. Several crack segmentation methods have been developed based on deep learning. Sun et al. proposed the integration of a competitive network (CCSNet) to address the impact of complex scenes on the segmentation accuracy of bridge crack images [29]. Zhang et al. introduced an efficient crack segmentation neural network architecture called ECSNet to accelerate real-time detection and segmentation of road surface cracks [30]. Asadi Shamsabadi et al. proposed an efficient semi-supervised method for surface crack segmentation in small datasets [31].

Most prior CNN models are constrained by a fixed window size, which limits their focus to local feature extraction. This restriction can result in information loss or blurring when addressing long-range dependency relationships, ultimately leading to rough and discontinuous segmentation. To address these limitations, this study proposes the ViT-S2T model, designed to capture both global semantics and fine local features for effective crack segmentation.

## 2.2. Vision Transformer

The Transformer is a deep learning network architecture that differs from CNNs in that it includes embeddings and self-attention [32]. Its advantages in extracting global information and modeling long-range dependencies in the context of natural language processing have been observed. Recently, Transformers have also been implemented as standalone architectures for computer vision tasks such as object detection, image classification, and semantic segmentation [33]. The accuracy of Transformer models can surpass complex CNN models used for image segmentation [34, 35]. Dosovitskiy et al. proposed the Vision Transformer (ViT), which is based on the encoder of the Transformer [36].

However, the Transformer architecture is associated with substantial computational costs, prolonged training times, and reduced efficiency in extracting local features compared to Convolutional Neural Networks (CNNs). Although the proportion of crack regions is small, cracks are widely distributed across the entire image, making it essential to capture both local details and global dependencies for pixel-level crack extraction. Consequently, this study integrates the Transformer into a CNN framework and introduces TFEM and ISM to effectively balance contextual semantic features with fine segmentation.

## 2.3. PID

The PID controller is widely used due to its simple structure, strong robustness, ease of implementation, and low maintenance costs [37–39]. The optimal hyperparameters of a PID controller have a significant impact on control performance. In recent years, to overcome the limitations of traditional PID control and meet various control requirements, modern optimization techniques such as Particle Swarm Optimization (PSO) [40–42], Expert Systems (ES) [43], Differential Evolution (DE) [44], Artificial Neural Networks combined with Differential Evolution (DE + ANN) [45, 46], Fuzzy Logic [47], and Artificial Bee Colony algorithm (ABC) [48] have been applied.

While the classical PID controller continues to be a widely used option in industry due to its simplicity and interpretability, it often encounters challenges related to parameter tuning, resulting in convergence issues and difficulty achieving global optimality. These limitations are

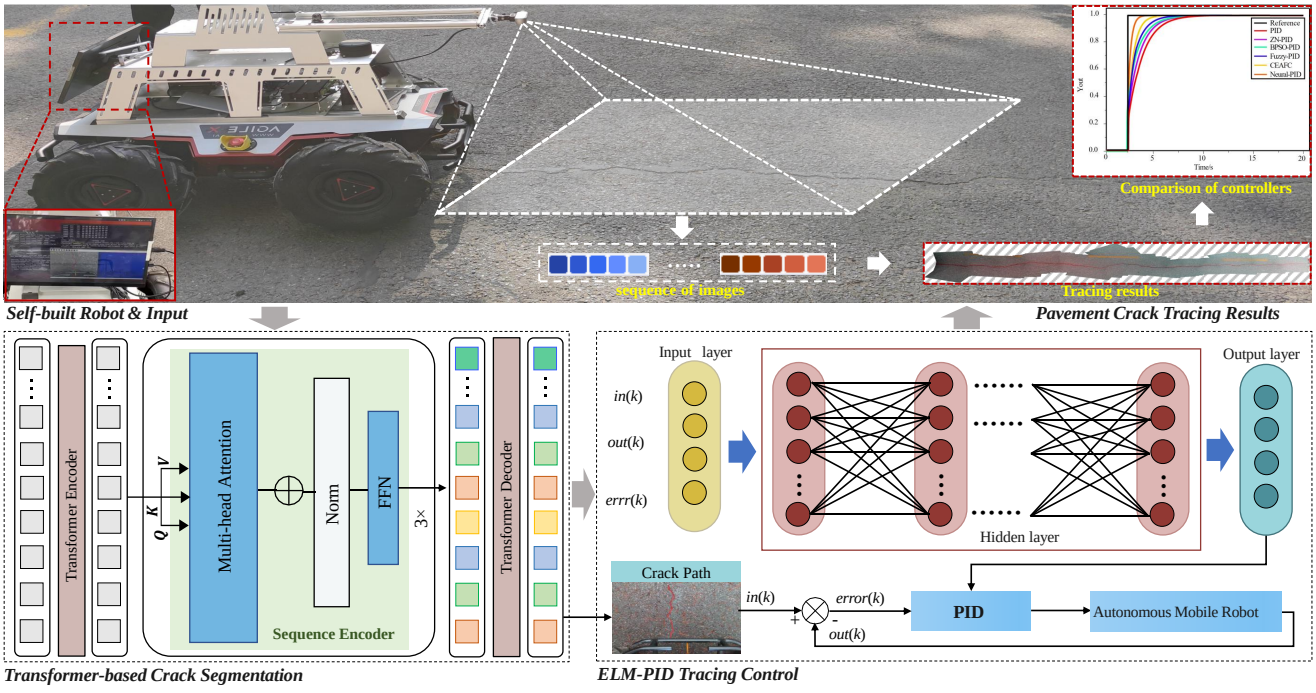



Figure 1: Our proposed a novel crack tracing framework, which consists of Self-built Robot & Input, Transformer-based Crack Segmentation, ELM-PID Tracing Control, Pavement Crack Tracing Result.

particularly pronounced in the presence of uncertainties and disturbances within the system. To address these challenges, this study proposes the utilization of Extreme Learning Machines (ELM) for training PID control parameters, enabling online self-tuning.

## 3. Methodology

In this section, we provide a detailed introduction to the crack tracing framework and elaborate on each module.

### 3.1. Overview

Automatic pavement crack tracing encounters several key challenges. Features of slender crack are prone to losing information after multiple convolution operations within CNN models, resulting in fragmented crack trajectories. Traditional PID controller relies solely on human-calibrated constant control parameters, leading to significant tracing errors. To address these challenges, this article proposes a novel framework for crack tracing, as demonstrated in Figure 1, which comprises two key branches: the fine-grained crack segmentation network and the robust crack tracing controller. Specifically, in TFEM module, a ViT-based segmentation network is proposed to accurately capture global context and semantic features of cracks. An ISM module, combining local ROI selection and coarse mask fusion, is designed for multi-scale mask head up-sampling, thereby enhancing crack segmentation accuracy. Moreover, an ELM-PID-based crack tracing controller is devised to achieve adaptive tuning of PID control parameters by employing an ELM model based on backpropagation neural network, effectively addressing the difficulty of online parameter tuning. A comprehensive framework has been developed for tracing pavement cracks, which includes crack extraction from segmentation, and trajectory tracing control. Detailed descriptions of each module will be presented in subsequent sections.

### 3.2. Crack Trajectory Extraction with Instance Segmentation

In this section, the global features of the image are extracted using the ViT model to capture the semantic information of the cracks through TFEM. Next, the local ROI selection is combined with the coarse mask through the ISM module, enabling multi-scale mask head up-sampling and fusion to obtain more accurate crack segmentation results.

#### *3.2.1. Transformer-based Segmentation*

In this TFEM module, crack features are mapped into sequential position encodings, which uses multi-head self-attention mechanisms to establish global context relations and semantic information. One of the common layers in the Encoder and Decoder structures is the Multi-Head Attention, as illustrated in the lower left corner of Figure 2. This layer comprises multiple parallel Self-Attention mechanisms. An encoder block contains a single multi-head attention mechanism, while a decoder block features two multi-head attention mechanisms, one of which utilizes masking. The Scale Dot Product Attention serves as the core of the Transformer architecture. In Self-Attention, the vectors Q, K, and V are computed from the same input (for example, words in a sequence). Specifically, Q, K, and V are derived from the embeddings of the original input words through the application of linear transformations, such as using fully connected layers. The dimensions of these three vectors are typically the same and depend on the design decisions made during model construction. By leveraging the autoregressive mechanism of the Transformer decoder and weighting specific pixels based on attention weights, pixel-level segmentation masks corresponding to crack instances are generated. As shown in Figure 2, the CSPDarknet53 module of the YOLOv5 [49] model backbone is replaced with TFEM. The features of different scales obtained after convolution operations are used as position encodings in the encoder of the Transformer structure. Multi-head attention mechanisms are employed to

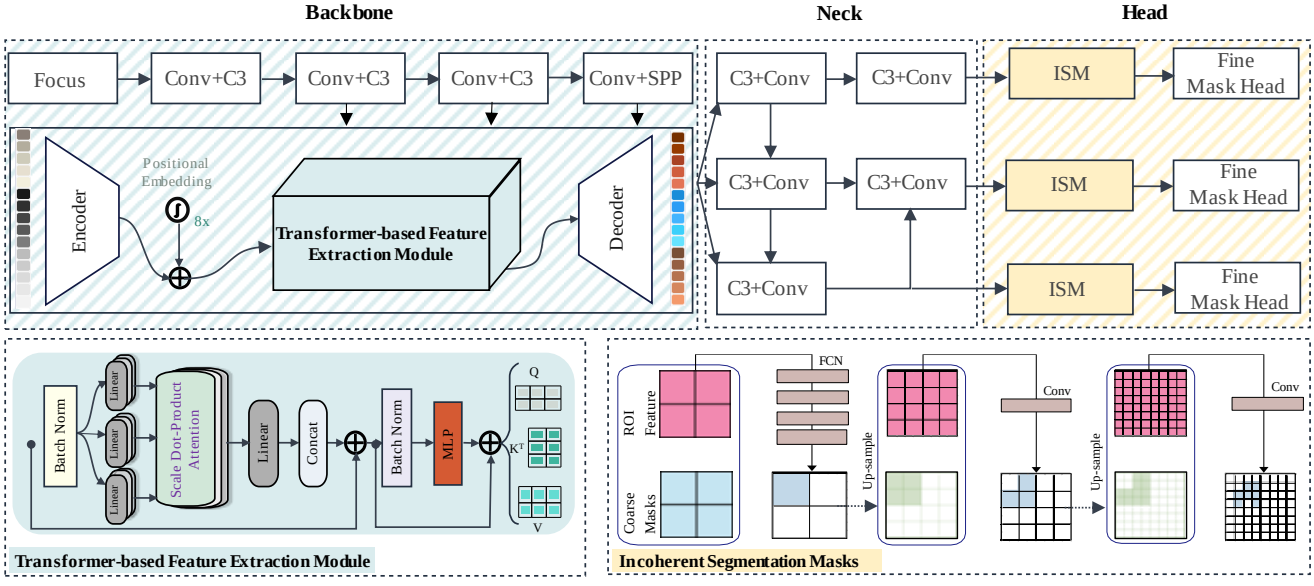



Figure 2: Architecture of ViT-S2T.

extract global and local features at different positions. After obtaining multi-scale features, they are passed into the Neck to achieve comprehensive crack feature extraction.

Moreover, the ISM module adopts a cascaded design by [50]. Specifically, it first performs prediction, which consists of four 3Œ3 convolutional kernels, to connect the minimal features and the coarse instance mask. Then, the detected low-resolution mask is up-sampled and fused with higher-resolution features from adjacent levels to guide more accurate predictions of elongated cracks. As illustrated in Figure 2, the ISM module is inserted into the multi-scale mask head of the YOLOv5's head using a parallel design for performing finer-grained crack instance predictions.

### 3.2.2. Loss Function

The segmentation method proposed in this study employs a comprehensive loss function that integrates the tasks of object detection and instance segmentation. This loss function comprises the classification loss($L_{cls}$), confidence loss($L_{obj}$), and bounding box regression loss for object detection($L_{bbox}$), in addition to the binary cross-entropy loss for instance segmentation($L_{mask}$).

The classification loss is calculated using the binary cross-entropy function, which is defined as:

$$\begin{aligned} L_{BCE} &= -y \log p - (1-y)\log(1-p) \\ &= \begin{cases} -\log p & , \quad \text{y=1} \\ -\log(1-p) & , \quad \text{y=0} \end{cases} \end{aligned} \tag{1}$$

where $y$ denotes the label for the input sample, with a value of 1 indicating a positive sample and 0 representing a negative sample. Meanwhile, $p$ represents the predicted probability that the model assigns to classifying the input sample as a positive sample. This component compels the model to accurately classify input data, thereby improving classification accuracy and helping the model to learn more representative features. This is particularly important when dealing with images that contain high noise levels or complex backgrounds, as accurate classification serves as a prerequisite for successful detection.

The confidence loss is calculated utilizing the binary cross-entropy function and incorporates varying weight coefficients for confidence losses obtained from detection layers with different scales. These coefficients are assigned based on the descending order of the detection layer's scale and have corresponding default values of 4.0, 1.0, and 0.4. As a result, the loss on multi-scale feature maps used for detecting slender object receives higher weight coefficients, emphasizing the training focus on slender object.

$CIoU$ tackles the dissimilarities between predicted bounding boxes and ground truth boxes by considering additional factors like aspect ratio. To mitigate this matter, $CIoU$ introduces two new parameters:

$$\begin{aligned} L_{CIoU(B,B_{gt})} &= IoU(B,B_{gt}) - \frac{\rho^2(B,B_{gt})}{c^2} - \alpha\upsilon \\ \upsilon &= \frac{4}{\pi}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \\ \alpha &= \frac{\upsilon}{1 - IoU(B,B_{gt}) + \upsilon} \end{aligned} \tag{2}$$

where the parameter $\upsilon$ represents the normalized difference in aspect ratio between the predicted and ground truth boxes. The value of the parameter a lies within the range of 0 to $\pi/4$, which is then multiplied by $\pi/4$ to convert it into the range of 0 to 1. On the other hand, the parameter $\alpha$ serves as a balancing factor for the trade-off between the loss caused by the aspect ratio difference and the loss caused by the intersection over union (IoU). Optimizing the position and size of detection boxes is essential for precise localization of targets. This aspect of the loss is critical, as accurate bounding boxes not only enhance target detection performance but also provide a reliable foundation for subsequent tasks such as tracking and segmentation. This optimization process aids the model in adapting to targets of varying scales and shapes.

$$L_{cls} = L_{obj} = L_{mask} = L_{BCE} \tag{3}$$

$L_{obj}$ evaluates the confidence of detected objects, enhancing the model's capability to recognize targets. It signifies that the model not only focuses on the presence of objects but also assesses their significance, allowing for smarter selection of targets that warrant attention and processing. This capability is crucial in multi-object scenarios. In the context of instance segmentation, $L_{mask}$ contributes to a finer separation between targets and backgrounds by providing more detailed contextual information. For instance, in complex scenes, effective masks can significantly reduce both false negatives and false positives, ensuring high accuracy in segmentation results.

$$L_{bbox} = L_{L_{CIoU(B,B_{gt})}} \tag{4}$$

$$L_{Total} = L_{cls} + L_{obj} + L_{bbox} + L_{mask} \tag{5}$$

The formulas presented above depict the calculation of the classification loss($L_{cls}$), confidence loss($L_{obj}$), bounding box regression loss($L_{bbox}$), binary cross-entropy loss for instance segmentation($L_{mask}$), and the total loss($L_{Total}$).

## 3.3. Crack Trajectory Tracing via ELM-PID controller

A design solution for a crack tracing controller based on ELM-PID is proposed. This solution utilizes an ELM model, which is grounded in backpropagation neural network to determine optimal control parameters, thereby achieving adaptive tuning of the PID settings. As shown in Figure 3, the

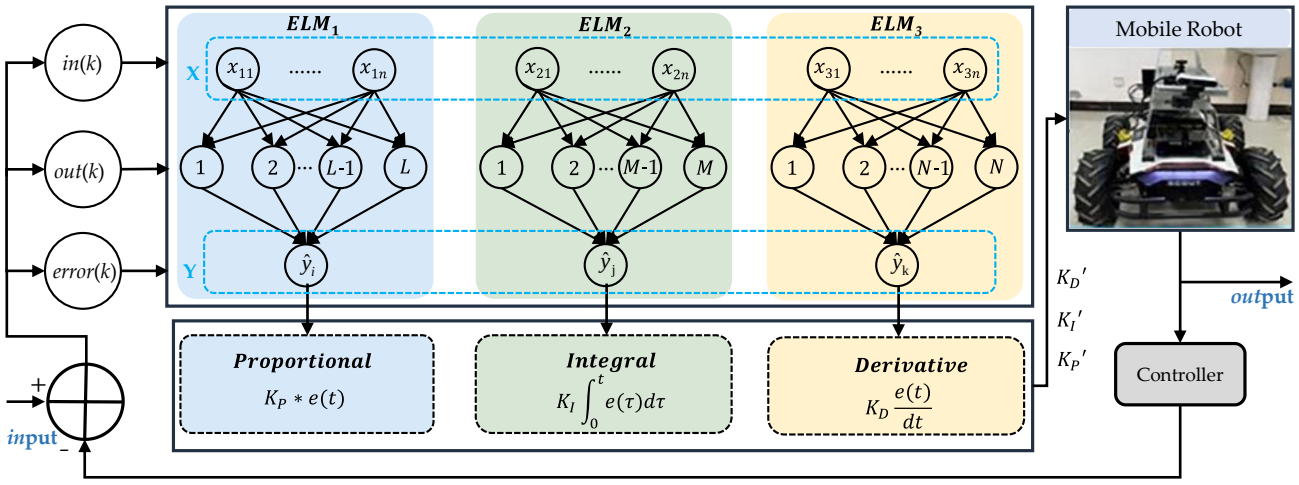



Figure 3: Block diagram of ELM-PID controller for crack tracing.

mask matrix obtained from the crack segmentation network is fed into the ELM model, where the optimal values for the control parameters $K_p$, $K_i$, and $K_d$ are calculated. The adaptive PID controller is then used to adjust and control the autonomous movement of the robot for crack trajectory tracing.

### 3.3.1. Extreme Learning Machine

The Extreme Learning Machine (ELM) is a type of feedforward neural network that utilizes gradient descent algorithms to estimate the weights and biases that connect the input and hidden layers [51]. It demonstrates remarkable capabilities in nonlinear regression and generalization [52]. Specifically, the weights $\beta$ linking the hidden layer to the output layer are calculated by solving the generalized inverse matrix. Given $N$ arbitrary samples $(\mathbf{X}_i, \mathbf{t}_i)$, where $\mathbf{X}_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in \mathbf{R}^n$ and $\mathbf{t}_i = \left[t_{i1}, t_{i2}, \cdots, t_{im}\right]^T \in \mathbf{R}^m$, the output $O_j$ of the network, which consists of $L$ hidden layer nodes, can be expressed as follows:

$$\mathbf{O}_j = \sum_{i=1}^{L} \beta_i g(w_i \cdot X_j + b_i), j = 1, 2, \cdots, N \tag{6}$$

where $g(.)$ represents the sigmoid activation function. $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]$ signifies the weights, while $b$ denotes the biases between the input layer and the hidden layer. Similarly, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_L]^T$, signifies the weights between the hidden layer and the output layer. The primary objective of the network is to minimize the output error, which can be mathematically expressed as follows:

$$\sum_{j=1}^{N} \|\mathcal{O}_j - t_j\| = 0 \tag{7}$$

Also the matrix can be expressed as follows:

$$\mathbf{H}\beta = \mathbf{T} \tag{8}$$

where $\mathbf{H} = g(w_i \cdot X_j + b_i)$ is the output matrix of the hidden layer, $\beta$ is the weight of the output layer, and $T$ is the desired output matrix of the network. They are expressed as:

$$\mathbf{H}(W_1, \cdots, W_L, b_1, \cdots, b_L, X_1, \cdots, X_L) = \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & \cdots & g(W_L \cdot X_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(W_1 \cdot X_N + b_1) & \cdots & g(W_L \cdot X_N + b_L) \end{bmatrix}_{N \times L} \tag{9}$$

$$\beta = \left[\beta_1, \cdots, \beta_L\right]_{L \times M} \tag{10}$$

$$T = \left[T_1, \cdots, T_L\right]_{L \times M} \tag{11}$$

The solution can be expressed as

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{12}$$

where Moore-Penrose matrix of $H$ is denoted as $\mathbf{H}^{\dagger}$.

### 3.3.2. ELM-PID Visual Tracing Controller

In the ELM-PID scheme, the ELM network is employed to predict the control parameters of the system. The structure of the ELM network consists of 6 input nodes, 18 hidden nodes, and 3 output nodes. The input vector is denoted as $[\mathbf{u}(k), \mathbf{y}(\dot{k})] \in \mathbf{R}^6$, and the output vector is represented as $\mathbf{y}^*(k) = [x(k), y(k), \omega(k)]^T \in \mathbf{R}^3$. The block diagram of the ELM-PID controller is depicted in Figure 4. In this scheme, $r(k)$ and $e(k)$ respectively refer to the desired feature position and the system's error signal. $u(k)$ represents the PID output signal, while $y(k)$ and $y^*(k)$ correspond to the current and predicted feature positions' outputs. The crack tracing process of the autonomous mobile robot encompasses training and self-tuning stages.

During the training stage, the initial parameters $K_P$, $K_I$, and $K_D$ were respectively set to 10, 1, and 0.01. The robot controller guides the robot's movement by utilizing the

input error signal, $e(k)$, and produces the angular and linear velocities of the robot. The sampling frequency is set at 50 ms. The collected input error signal, $e(k)$, and output values, $u(k)$, are employed as the training set for the ELM model. These values will undergo normalization during the training process.

In the self-tuning stage, the incremental PID algorithm is employed to tune the parameters, and the digital PID controller can be expressed in discrete-time as follows:

$$\Delta u(k) = K_p(k-1)x_1(k) + K_i(k-1)x_2(k) + K_d(k-1)x_3(k) \tag{13}$$

The control law is given as:

$$u(k) = u(k-1) + \Delta u(k) \tag{14}$$

where equation represents the output of the PID controller, denoted as $u$, where $k$ denotes the iteration step, and

$$\begin{cases} e(k) = \mathbf{r}(k) - \mathbf{y}^*(k) \\ x_1 = e(k) - e(k-1) \\ x_2 = e(k) \\ x_3 = e(k) - 2e(k-1) + e(k-2) \end{cases} \tag{15}$$

The cost function of the controller is defined as:

$$E(k) = \frac{1}{2}e^2(k) \tag{16}$$

During the operational process, the weights of the network are adjusted using backpropagation to minimize the cost function $E$. The self-adjustment steps can be outlined as follows:

$$\begin{cases} \Delta K_p &= -\eta \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u}{\partial K_p(k-1)} \\ &= \eta e(k) \frac{\partial y(k)}{\partial u(k)} x_1(k) \\ \Delta K_i &= -\eta \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u}{\partial K_i(k-1)} \\ &= \eta e(k) \frac{\partial y(k)}{\partial u(k)} x_2(k) \\ \Delta K_d &= -\eta \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u}{\partial K_d(k-1)} \\ &= \eta e(k) \frac{\partial y(k)}{\partial u(k)} x_3(k) \end{cases} \tag{17}$$

where the learning rate, denoted as $\eta$, and the Jacobian parameter ($\frac{\partial y}{\partial u}$) are integral components of the visual Control system.

Assuming that the sampling period is sufficiently small, we can deduce that there exists a linear relationship between two adjacent sampling points. Therefore, it becomes feasible to approximate the inverse Jacobian matrix.

$$\frac{\partial y}{\partial u} \approx \frac{\partial y^*}{\partial u} = \sum_{i=1}^{l} \omega_{im}\beta_i g'(\omega_i u + b) \tag{18}$$

Here, $l$ represents the number of nodes in the hidden layer, $\beta$ denotes the output weight, $b$ signifies the bias of the hidden layer, and $\omega_{im}$ represents the corresponding weight in the vector $\omega_i$ of the input $u$. The calculation of the incremental PID parameters, $K_P$, $K_I$, and $K_D$, can be expressed as follows:

$$\begin{cases} K_p(k) = K_p(k-1) + \Delta K_p(k) + \alpha_1\big(K_p(k-1) - K_p(k-2)\big) \\ K_i(k) = K_i(k-1) + \Delta K_i(k) + \alpha_2(K_i(k-1) - K_i(k-2)) \\ K_d(k) = K_d(k-1) + \Delta K_d(k) + \alpha_3(K_d(k-1) - K_d(k-2)) \end{cases} \tag{19}$$

where $\alpha \in (0, 1)$ is the momentum coefcient.

## 4. Experiments

This section focuses on the evaluation of the proposed method through representative segmentation benchmarks, PID control benchmarks, and validation. The first aspect encompasses the details of the experimental implementation. Subsequently, an analysis and discussion of the instance segmentation results for ViT-S2T are presented. Afterward, a comprehensive comparative analysis of tracing controllers is conducted. Lastly, the proposed method is deployed on a self-developed autonomous mobile robot for real-time tracing verification of road cracks on an actual asphalt surface.

### 4.1. Implementation Details

**Parameters Setting**. The segmentation model training experiments were conducted on an Intel Core i9-13900KF processor with 6 cores and 32GB RAM, operating at a frequency of 5.80 GHz. The GPU server was also equipped with an NVIDIA GeForce RTX 4090 GPU with 24GB memory. The model was trained using the stochastic gradient descent (SGD) optimizer for 300 epochs, with a momentum of 0.8 and weight decay of 0.0001. The initial learning rate was set to 0.01. Our proposed method was tested and validated on three datasets: DeepCrack [53], CFD [54], and S2TCrack [55]. These datasets were selected because they contain typical asphalt road cracks. A description of the datasets is provided below.

- The DeepCrack dataset consists of 537 concrete surface crack images with a size of 544 x 384 pixels. This dataset includes various types of damages such as cracks, potholes, and patched areas. The patch regions of the images are sized at 480 x 480 pixels. The dataset is divided into 350 training images, 50 validation images, and 137 testing images.
- The CFD dataset consists of 118 road crack images, each with a size of 480 pixels x 320 pixels. These images were captured by individuals standing on the road using an iPhone. The images have been meticulously annotated at the pixel level, a labor-intensive task. The images exhibit high quality with a smooth and clean background.

Table 1: Relevant parameters of ELM-PID with other controllers, including detailed information about the initial configuration.

| Algorithms | Parameters |
|---|---|
| PID | $k_p$ = 0.00055, $k_i$ = 0.0025, $k_d$ = 0.0035 |
| ZN-PID | $K_{pcrit}$ = 202, $T_{crit}$ = 100ms |
| BPSO-PID | Population size = 20, Maximum iterations = 40, Acceleration parameters = 2, Initial weights = 0.9, End weights = 0.4, Algorithm termination threshold = 1 × $10^{-25}$, Iteration termination threshold = 10, PSO Algorithm Type = 0 |
| Fuzzy PID | Membership functions = triangular and trapezoidal functions, $\Delta k_{p1}$ = [-0.15, 0.15] , $\Delta k_{i1}$ = [-0.02, 0.02] , $\Delta k_{d1}$ = [-0.07, 0.07] , $\Delta k_{p2}$ = [-0.25, 0.25] , $\Delta k_{i2}$ = [-0.01, 0.01] , $\Delta k_{d2}$ = [-0.03, 0.03] |
| CEAFC | Cross-entropy iterations = 15, $k_{p1}$ = 1.231, $k_{i1}$ = 0.0107, $k_{d1}$ = 0.747, $k_{p2}$ = 2.506, $k_{i2}$ = 0.0288, $k_{d2}$ = 1.313 |
| ELM-PID | Maximum iterations = 30, $k_p$ = 10, $k_i$ = 1, $k_d$ = 0.1, sampling frequency = 50ms, Learning Factor $c_1$ = 1, $c_2$ = 1.5 |

- The S2T-Crack dataset is specifically created for road crack segmentation and tracing. It is captured using an unmanned wheeled robot platform equipped with an Intel RealSense D435i RGB-D camera. The dataset comprises 1051 images for training, 149 images for validation, and 302 images for testing. The road cracks in the dataset include straight cracks, curved cracks, and continuous turning cracks. The dataset is collected in various complex environments, including normal conditions (243 images), shadowed areas (253 images), dark conditions (247 images), plant branches (256 images), wet conditions (248 images), and motion blur (255 images).

The self-developed autonomous mobile robot is equipped with a front-facing Realsense D435i RGB-D camera and runs on the Robot Operating System (ROS) [56]. The embedded environment includes Jetpack 4.4, PyTorch 1.8, Linux Ubuntu 18.04, and ROS Melodic. The controller is implemented using an Nvidia Jetson AGX Xavier embedded computer with specifications that include an Nvidia Volta GPU with 512 CUDA cores and 64 tensor cores, an ARM CPU v8.2 with 8 cores, and 32GB DDR4 memory. The maximum iteration count for the six control algorithms is set to 80, and the learning rate for all controllers is unified at 0.001. In the Binary Particle Swarm Optimization (BPSO) algorithm, the inertia weight w is set to 0.8, and the learning factors C1 and C2 are set to 2. The initial weights for all controllers are randomly selected from the range of [0.5, 0.5], and the output control signal of the controller is limited to the range of [-10, 10]. Table 1 presents detailed insights into the parameters associated with each implemented algorithm.

**Evaluation Criteria**. To assess the binary masks of the segmentation predictions, evaluation metrics including precision (M), recall (M), and average precision (AP) are employed. Additionally, the AP incorporates mAP0.5 (M), which represents the AP with an IoU threshold greater than 0.5, and mAP0.5:0.95 (M), which pertains to the average AP with an IoU threshold ranging from 0.5 to 0.95 with an increment of 0.05. In this paper, the maximum error value (MAX), root mean square error (RMSE), and mean absolute error (MAE) are used to evaluate the superiority and computational accuracy of the proposed ELM-PID controller compared to other controllers. The best objective function $J$ is employed to evaluate the effectiveness of the algorithm. Rise time and settling time are utilized to assess the response speed of the control method, while the number of iterations is used to measure the convergence speed of the algorithm. These metrics were chosen as they are common performance indicators in the field of control engineering.

## 4.2. Segmentation results

A comprehensive comparison of various commonly used metrics during the training phase, including precision, recall, and mean average precision (mAP), is conducted. Each metric is examined based on its strengths, limitations, and suitability for different segmentation scenarios. Moreover, this comparison helps understand the trade-offs between different metrics and aids in selecting the most appropriate evaluation measures for specific detection tasks. The main methods investigated in this study include YOLOv5 [49], YOLACT-700 [57], SOLO [58], and S2TNet [55]. Specifically, YOLOv5, S2TNet and our proposed method are implemented using the open-source framework available at

Table 2: Comparative analysis of training accuracy metrics across three datasets with different algorithms. The best results are highlighted in red, and the second best results are highlighted in blue.

| Method | DeepCrack | | | | CFD | | | | S2T-Crack | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Precision(M)* | *Recall(M)* | *mAP$^{val}$0.5(M)* | *mAP$^{val}$0.5:0.95(M)* | *Precision(M)* | *Recall(M)* | *mAP$^{val}$0.5(M)* | *mAP$^{val}$0.5:0.95(M)* | *Precision(M)* | *Recall(M)* | *mAP$^{val}$0.5(M)* | *mAP$^{val}$0.5:0.95(M)* |
| YOLOv5 | 0.4587 | 0.4531 | 0.4969 | 0.1876 | 0.4621 | 0.4474 | 0.3875 | 0.0854 | 0.6539 | 0.5864 | 0.6724 | 0.2364 |
| YOLACT-700 | 0.4655 | 0.4283 | 0.3476 | 0.1511 | 0.4579 | 0.3213 | 0.3824 | 0.0490 | 0.6846 | 0.6032 | 0.6684 | 0.2389 |
| SOLO | 0.4904 | 0.4997 | 0.5408 | 0.2245 | 0.4735 | 0.4501 | 0.4351 | 0.1354 | 0.7165 | 0.6897 | 0.6922 | 0.2451 |
| S2TNet | 0.5567 | 0.5208 | 0.5021 | 0.2054 | 0.4868 | 0.3437 | 0.4197 | 0.1956 | 0.7321 | 0.5812 | 0.6989 | 0.2872 |
| ViT-S2T | 0.5322 | 0.5539 | 0.5381 | 0.2349 | 0.5119 | 0.4956 | 0.4734 | 0.1875 | 0.8276 | 0.7234 | 0.7563 | 0.2652 |

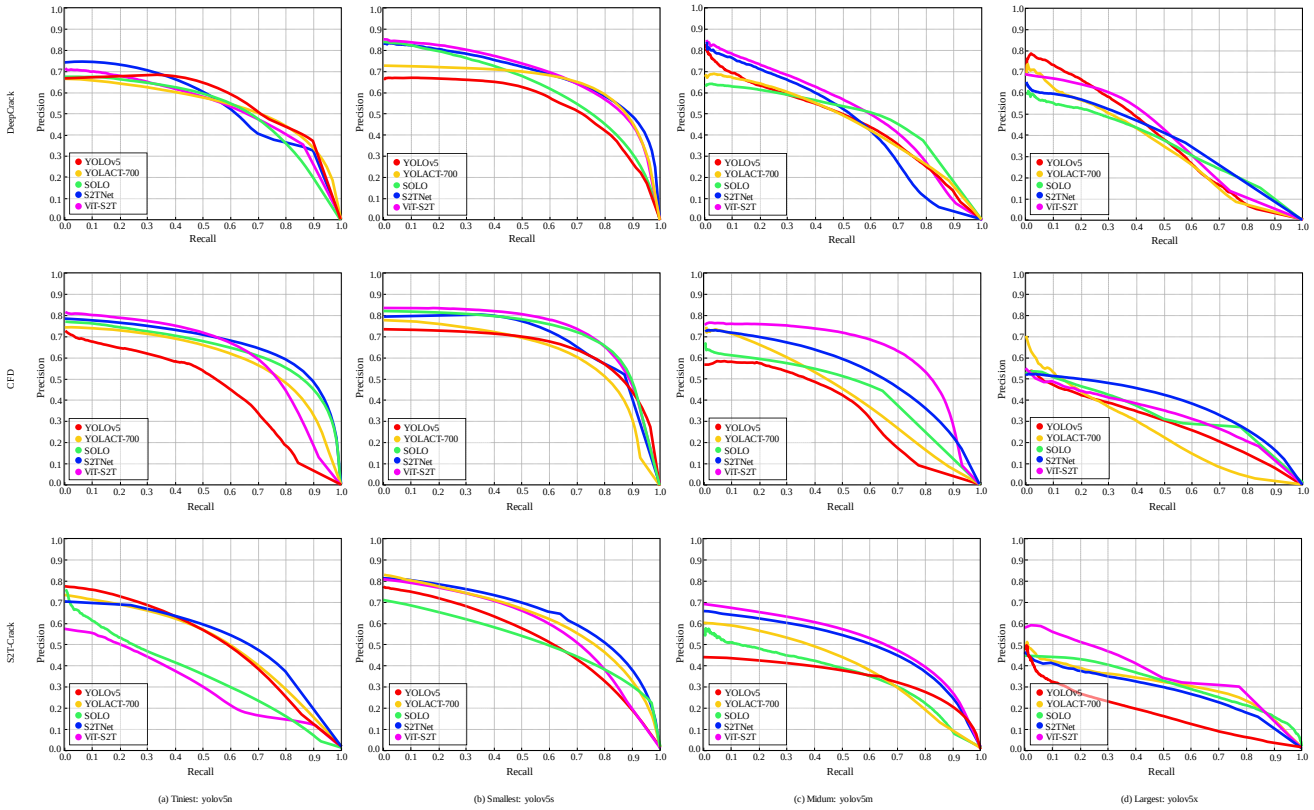



Figure 4: Comparative analysis of PR curves on three datasets: CRACK500, CFD, and S2T-Crack, using several pretrained models: (a) Tiniest: yolov5n, (b) Smallest: yolov5s, (c) Midum: yolov5m, and (d) Largest: yolov5x.

https://github.com/ultralytics/yolov5. The remaining methods are implemented using the open-source framework available at https://github.com/open-mmlab/mmdetection. It is worth noting that all these methods are trained using the same initialization parameters.

**DeepCrack dataset**.The DeepCrack dataset is analyzed across different pretrained models ('n', 's', 'm', 'x') for various algorithms, as depicted in the first row of Figure 4. The models are trained for 300 epochs to obtain the metrics. Table 2 provides a quantitative analysis of the training metrics for different models. The results indicate that S2TNet achieves the highest precision of 0.5567, while ViT-S2T achieves the highest recall of 0.5539. SOLO attains the highest mAP@0.5 metric of 0.5408, and ViT-S2T achieves the highest mAP@0.5:0.95 metric of 0.2349.

**CFD dataset**.The CFD dataset is analyzed, and Figure 4 illustrates the training metrics for different algorithms. From the precision-recall curve and the area under curve (AUC) values in the second row of Figure 4, ViT-S2T achieves the highest performance among the three pretrained models. Table 2 provides a quantitative analysis of the training metrics for different models. The performance analysis demonstrates that ViT-S2T achieves the highest precision of 0.5119 and the highest recall of 0.4956. Additionally, ViT-S2T achieves the highest mAP@0.5 metric of 0.4734, while S2TNet attains the highest mAP@0.5:0.95 metric of 0.1956.

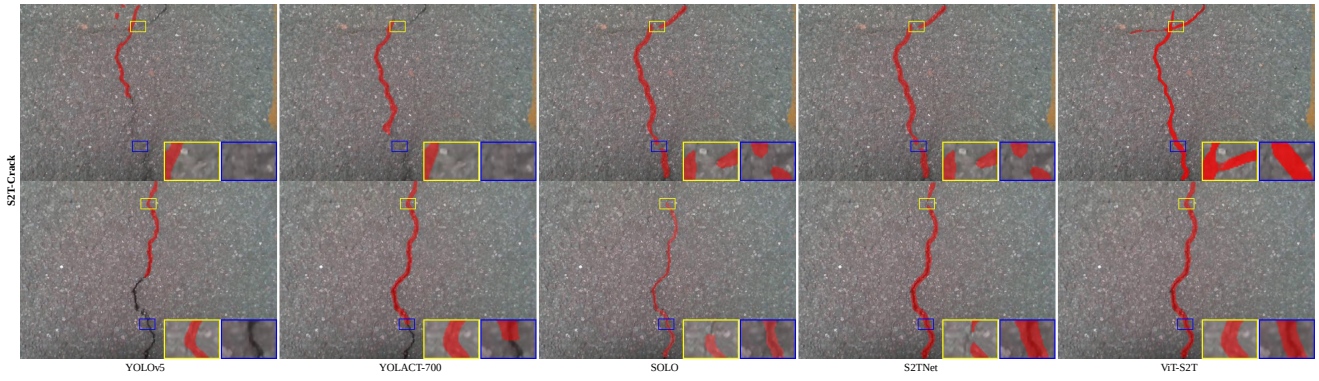



Figure 5: Visualization of crack segmentation: A comparison between SOTA Methods and ViT-S2T(Ours). To enhance visual clarity, we have selectively enlarged the test results of the various methods.

Table 3: Comparative analysis of different modules with ViT-S2T. Key indicators are visually represented using bold.

| Method | Ablation | | DeepCrack | | | | CFD | | | | S2T-Crack | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TFEM | IRS | *Precision(M)* | *Recall(M)* | *mAP0.5(M)* | *mAP 0.5:0.95(M)* | *Precision(M)* | *Recall(M)* | *mAP0.5(M)* | *mAP 0.5:0.95(M)* | *Precision(M)* | *Recall(M)* | *mAP0.5(M)* | *mAP 0.5:0.95(M)* |
| baseline | ✗ | ✗ | 0.4681 | 0.5056 | 0.4652 | 0.1534 | 0.4562 | 0.3947 | 0.3631 | 0.0545 | 0.6795 | 0.6436 | 0.6518 | 0.2837 |
| ours | ✓ | ✗ | 0.4587 | **0.5584** | **0.5041** | 0.2314 | 0.4742 | 0.4257 | 0.3995 | 0.1097 | 0.7448 | 0.6735 | 0.6817 | 0.2456 |
| | ✗ | ✓ | 0.4794 | 0.5026 | 0.4887 | 0.2184 | **0.5283** | 0.4349 | 0.4033 | 0.1069 | 0.7553 | **0.7246** | 0.6772 | 0.2438 |
| | ✓ | ✓ | **0.5044** | 0.5392 | 0.4975 | **0.2432** | 0.4848 | **0.4572** | **0.4213** | 0.1568 | **0.8167** | 0.7159 | **0.7546** | **0.3078** |

Compared to the baseline YOLOv5, our proposed ViT-S2T method significantly improves performance by 8.59% in terms of mAP@0.5.

**S2T-Crack dataset**.The S2T-Crack dataset is analyzed, and the third row of Figure 4 illustrates the training metrics for different algorithms. Table 2 provides a quantitative analysis of the training metrics for various models. It can be observed that ViT-S2T achieves the highest precision of 0.8276, the highest recall of 0.7234, the highest mAP@0.5 metric of 0.7563, and S2TNet achieves an mAP@0.5:0.95 metric of 0.2872. Compared to the baseline YOLOv5, our proposed ViT-S2T method demonstrates improvements of 17.37% in precision, 13.7% in recall, and 8.39% in mAP@0.5. The visual results in Figure 5 indicate that the results of ViT-S2T are more continuous and clear compared to the contrasted deep learning models. Overall, after multiple testing and comparative analysis on the three different datasets, the results validate the robustness and good performance, in terms of accuracy metrics, of our proposed method.

**Ablation study**. To further examine the gains from each module in the model, an ablation study is conducted on three datasets. Table 3 compares the performance of the TFEM and ISM modules across different datasets and methods. As shown in Table 3, the mAP0.5@(M) of TFEM + ISM is 3.23%, 5.82%, and 10.28% higher than the baseline reference on the three benchmark datasets. It is also higher than TFEM by -0.66%, 2.18%, and 7.29%, and higher than ISM by 0.88%, 1.8%, and 7.74%. The optimal training parameter metrics of TFEM + ISM on the S2T-Crack dataset are [Precision(M), Recall(M), mAP@0.5(M), mAP@0.5:0.95(M)] = [0.8167, 0.7246, 0.7546, 0.3078]. TFEM employs a multi-head self-attention mechanism to establish global contextual relationships and semantic information, thereby effectively extracting both global and local features from various positions within the crack images. This capability allows the model to capture intricate details while maintaining an understanding of the overall structure of the cracks. After deriving multi-scale features, these are directed to the Neck of the network, which serves as a crucial component for comprehensive crack feature extraction. The ISM performs upsampling on the detected low-resolution masks, enhancing their detail and resolution. This upsampling process is further complemented by integrating these masks with higher-resolution features from adjacent levels, ensuring that fine details are retained while the broader context is maintained. Moreover, the ISM is seamlessly incorporated into the multi-scale mask head of the YOLOv5 architecture. This integration enables the system to execute more granular predictions of crack instances, enhancing the overall accuracy of segmentation tasks. By utilizing these innovative strategies, the TFEM framework significantly improves the capability to detect and characterize cracks in asphalt pavement, ultimately contributing to more effective pavement maintenance and monitoring solutions. This approach not only enhances detection accuracy but also bolsters the robustness of the model across various crack conditions and complexities. In most cases, TFEM + ISM outperforms both TFEM and ISM, demonstrating the value of TFEM + ISM in crack segmentation tasks.

### 4.3. Simulation analysis

This section compares the performance of the proposed ELM-PID controller with PID [59], the traditional Ziegler-Nichols PID controller (ZN-PID) [60], the basic Particle Swarm Optimization PID controller (BPSO-PID) [13], fuzzy PID [61], and CEAFC [1] controllers by analyzing three representative non-structured asphalt pavement crack trajectories. The control objective is to ensure that the control system accurately follows the reference trajectory with satisfactory tracing accuracy. A control system model for an autonomous mobile robot is established in the ROS environment, and controller algorithms are designed for construction purposes.

Table 4: Results of optimizing PID parameters and performance evaluation indicators with step reference.

| Controller | *Number of iterations* | *Rise time(s)* | *Settling time(s)* | *Best J* |
|---|---|---|---|---|
| PID | / | 6.1 | 8.7 | 38.09 |
| ZN-PID | / | 5.7 | 8.2 | 37.75 |
| BPSO-PID | 39 | 5.3 | 7.8 | 36.62 |
| Fuzzy PID | 21 | 4.4 | 6.9 | 34.14 |
| CEAFC | 12 | 3 | 5.4 | 32.87 |
| ELM-PID | 4 | 0.5 | 2.8 | 30.53 |

**Step reference**. The step response curves of the six controllers and the iteration curves of the four algorithms are illustrated in Figure 6(a) and Figure 7(a), respectively, utilizing a unit step signal as the reference trajectory for the crack tracing control system. The PID parameter results and corresponding performance evaluation metrics are presented in Table 4. A comparison of Table 4 with Figures 6 and 7 reveals that the PID controller exhibits the poorest response performance, with a rise time of 6.1 seconds and a settling time of 8.7 seconds. In contrast, the proposed ELM-PID controller and the previously proposed CEAFC demonstrate superior step response performance compared to the PID, ZN-PID, BPSO-PID, and fuzzy PID methods. Notably, the implementation of the ELM-PID significantly enhances the step response metrics, satisfying the criteria for accurate,

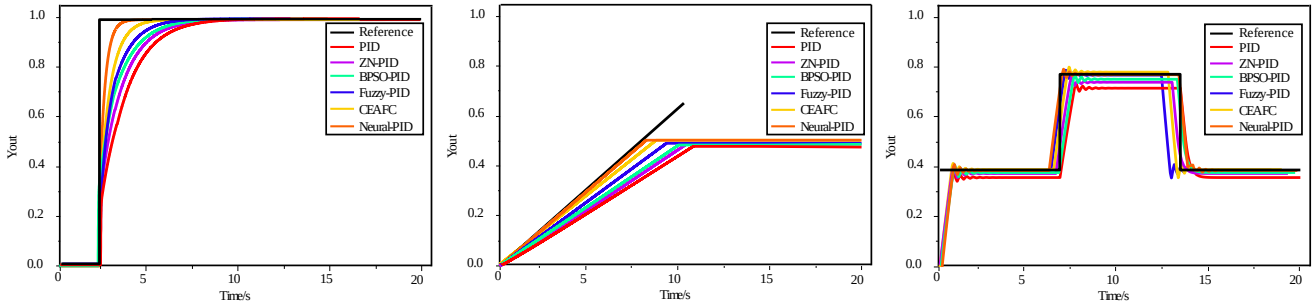



Figure 6: Response curves of different controllers with (a)step reference, (b)slope reference and (c)first-order linear reference.

rapid, and stable tracing by the self-developed robot. The ELM-PID achieves the best performance, with an objective function value J of only 30.53, requiring just four iterations. In comparison, the BPSO-PID method records an objective function value of 36.62 while necessitating 39 iterations, thereby illustrating that the ELM-PID method not only accelerates iteration speed but also decreases the number of iterations required.

Table 5: Results of optimizing PID parameters and performance evaluation indicators under slope reference.

| Controller | *Number of iterations* | *Rise time(s)* | *Settling time(s)* | *Best J* |
|---|---|---|---|---|
| PID | / | 10.7 | 11 | 74.24 |
| ZN-PID | / | 10.1 | 10.4 | 71.39 |
| BPSO-PID | 47 | 10 | 10.2 | 72.15 |
| Fuzzy PID | 38 | 9.1 | 9.3 | 70.43 |
| CEAFC | 33 | 8.6 | 8.7 | 69.96 |
| ELM-PID | 22 | 8 | 8.1 | 66.51 |

**Slope reference**. The reference trajectory for the crack tracing control system is obtained using a slope signal of 0.15. The response curves of the six controllers are shown in Figure 6(b), while the iteration curves of the four algorithms are shown in Figure 7(b). The PID parameter results and performance evaluation metrics are presented in Table 5. From the six response curves, it is evident that the ELM-PID method exhibits the highest response speed and tracing accuracy. Considering response time, the ELM-PID method achieves a rise time of 8 s and a settling time of 8.1 s. In contrast, the PID and ZN-PID methods exhibit rise times of 10.7 s and 10.1 s, and settling times of 11 s and 10.4 s, respectively. The ELM-PID method attains the best objective function $J$ of 66.51 with 22 iterations, which is 3.45-7.73 and 11-25 iterations less than the other controllers. Thus, compared to the other six controllers, the ELM-PID controller demonstrates the best performance.

Table 6: Results of optimizing PID parameters and performance evaluation indicators with rst-order linear reference, the indicators for the first stage, second stage, and third stage are labeled as black, blue, and red respectively.

| Controller | *Number of iterations* | *Rise time(s)* | *Settling time(s)* | *Best J* |
|---|---|---|---|---|
| PID | / | 2.6<br>2.4<br>1.2 | 2.4<br>2<br>0.8 | 94 |
| ZN-PID | / | 2.4<br>2.2<br>1.1 | 2.2<br>1.8<br>0.7 | 91 |
| BPSO-PID | 66 | 2.3<br>2<br>1.1 | 2.3<br>2<br>0.8 | 84 |
| Fuzzy PID | 55 | 1.9<br>1.8<br>1 | 1.8<br>1.4<br>0.7 | 87 |
| CEAFC | 49 | 1.7<br>1.5<br>0.8 | 1.6<br>1.3<br>0.6 | 80 |
| ELM-PID | 41 | 1.5<br>1.2<br>0.6 | 1.4<br>1<br>0.4 | 75 |

**First-order linear reference**. The reference trajectory for the crack tracing control system is generated using a first-order linear signal. The response curves of the six controllers are illustrated in Figure 6(c), while the iteration curves of the four algorithms are presented in Figure 7(c). The PID

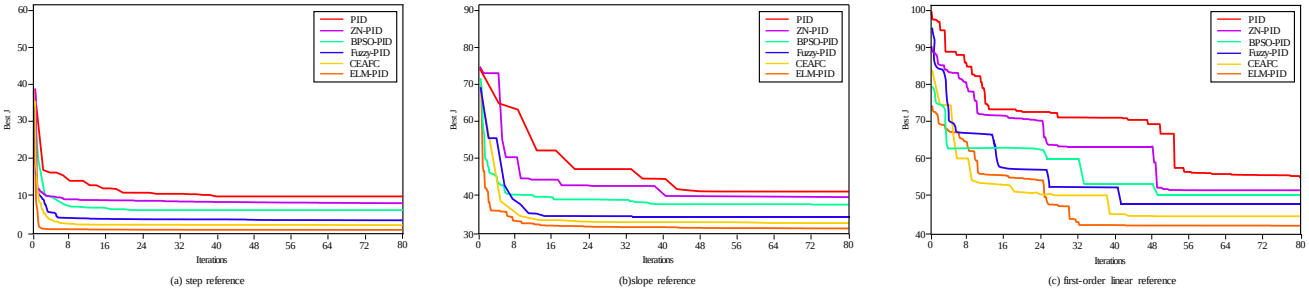
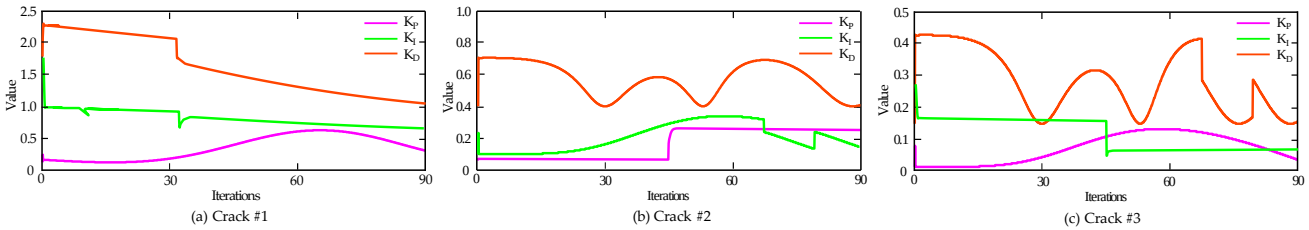



Figure 7: Iteration curves of different controllers under three pavement cracks: (a)crack #1(slender shape), (b)crack #2(curved shape), (c)crack #3(discontinuous shape).




(a) Crack #1 (b) Crack #2 (c) Crack #3

Figure 8: Sensitivity analysis on parameters tuning process of ELM-PID controller.

parameter results and performance evaluation metrics are detailed in Table 6. The response curves reveal notable differences among the six controllers, with maximum errors occurring at the rising and falling edges of the signal. Utilizing the ELM-PID method, the tracing response of the control system achieves a rise time of 1.5 seconds and a settling time of 1.4 seconds during the first stage of the signal. In the second stage, the rise time and settling time are reduced to 1.2 seconds and 1 second, respectively. The third stage demonstrates even greater efficiency, with rise and settling times of 0.6 seconds and 0.4 seconds, respectively, exhibiting minimal attenuation and delay, which are the lowest values among the six controllers. The ELM-PID method achieves the best objective function value J of 75 with 41 iterations, effectively meeting the control requirements. Compared to other methods, ELM-PID significantly enhances the responsiveness of the crack tracing control system. Overall, the high accuracy and rapid convergence of the ELM-PID method are validated through tracing simulations.

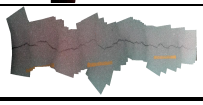
### 4.4. Crack trajectory tracing

This section presents experimental studies conducted on six controllers, which were developed based on the working characteristics of our autonomous mobile robot. The experiments were performed under real-world unstructured road conditions. The objective of these experiments was to validate the effectiveness of the ELM-PID method and highlight its superiority in comparison to existing approaches.

For crack #1, representing a slender-shaped asphalt pavement crack, the results of dynamically tuning the hyperparameters for the ELM-PID controller are presented in Figure 8(a). The tracing errors of six controllers deployed on the self-developed autonomous mobile robot are illustrated in Figure 9(a). Additionally, Table 7 summarizes the tracing error metrics, including maximum error (MAX), root mean square error (RMSE), and mean absolute error (MAE), for the six controllers combined with three segmentation methods across three typical asphalt road crack trajectories. Notably, the PID and ZN-PID controllers exhibit significant spikes in their motion velocity trajectories, indicating reduced smoothness during the crack tracing process. In contrast, the ELM-PID controller demonstrates superior error control performance, characterized by a smoother trajectory. A comparative analysis of the tracing error metrics indicates that the ELM-PID controller surpasses the others, exhibiting the lowest peak values in the error curves. Specifically, when employing the ViT-S2T segmentation model, the ELM-PID controller achieves a MAX of only 17.6 mm, a RMSE of 3.94 mm, and a MAE of 4.48 mm. The CEAFC method yields the second-best tracing performance, with an RMSE of 4.37 mm and an MAE of 5.17 mm. Conversely, the PID method displays the highest tracing error, with an RMSE of 10.81 mm. These findings underscore the effectiveness of the ELM-PID controller in enhancing tracing accuracy and highlight the importance of hyperparameter tuning in achieving optimal performance across different segmentation strategies. Further investigation into additional controller configurations and segmentation methods may provide avenues for even greater improvements in crack tracking capabilities.

For crack #2, the dynamic tuning results for the ELM-PID controller are presented in Figure 8(b). The tracing errors of six controllers deployed on the self-developed

Table 7: Comparative analysis of crack tracing evaluation metrics using the six controllers with three pavement crack conditions.

| Crack Scene | Demo | Evaluation Criteria | PID | ZN-PID | BPSO-PID | Fuzzy PID | CEAFC | ELM-PID |
|---|---|---|---|---|---|---|---|---|
| *#1 Slender* |  | *MAX* | 77.64 | 65.34 | 51.89 | 45.42 | 39.84 | **38.57** |
| | | *RMSE* | 15.73 | 15.06 | **10.08** | 10.59 | 11.62 | 10.92 |
| | | *MAE* | 13.45 | 11.35 | 9.87 | 9.05 | **6.24** | 7.72 |
| *#2 Curved* |  | *MAX* | 100.83 | 87.19 | 74.35 | 70.29 | 57.68 | **41.94** |
| | | *RMSE* | 12.01 | 11.34 | 11.23 | 17.29 | 11.78 | **10.68** |
| | | *MAE* | 17.23 | 14.52 | 13.52 | 11.28 | 8.89 | **7.34** |
| *#3 Discontinuous* |  | *MAX* | 127.24 | 101.43 | 95.78 | **73.95** | 84.49 | 87.25 |
| | | *RMSE* | 22.66 | 18.07 | 18.68 | 19.55 | **11.12** | 11.18 |
| | | *MAE* | 20.76 | 16.98 | 15.24 | 11.36 | 9.46 | **8.34** |

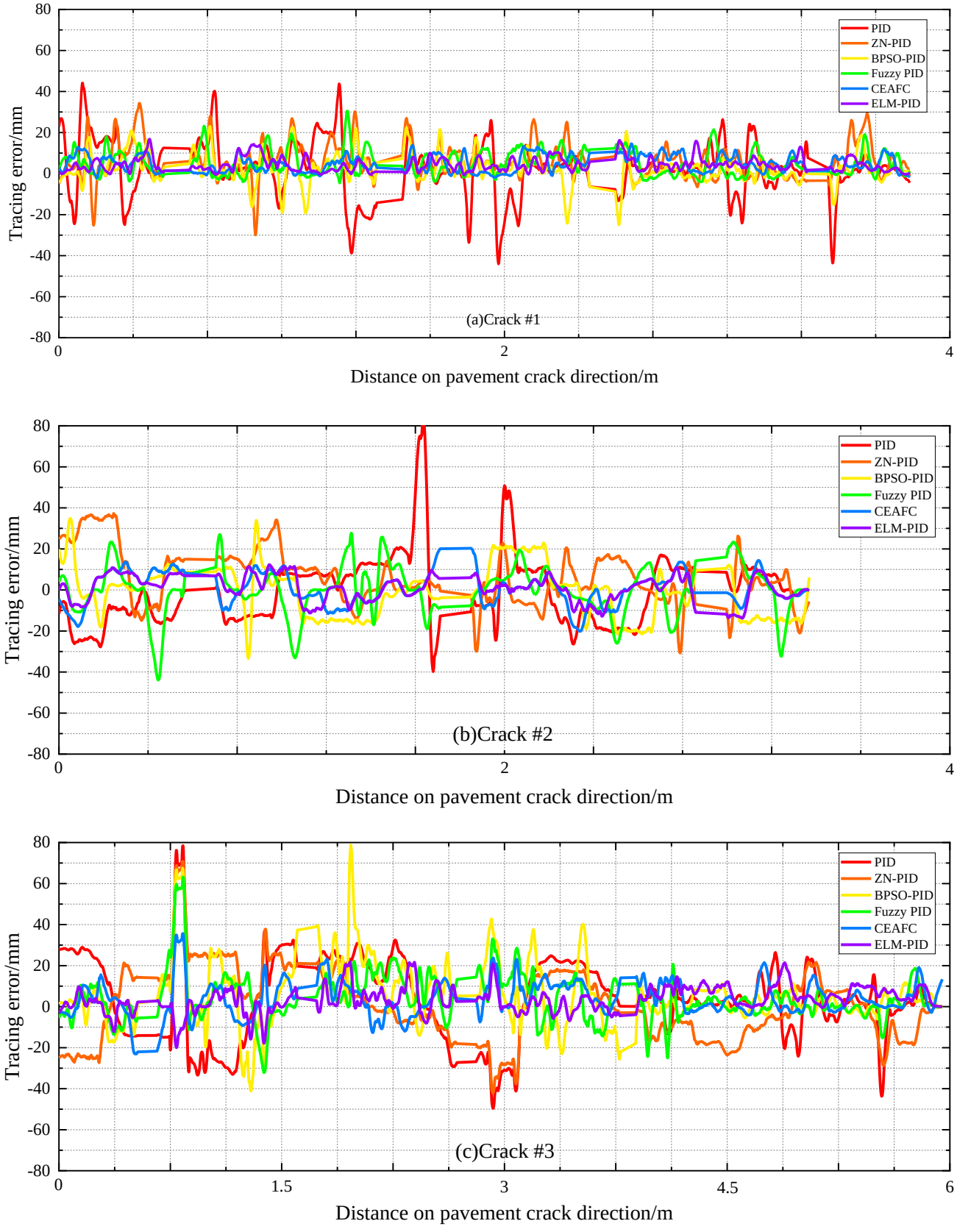



Figure 9: Tracing errors of the six controllers within three types of cracks.

autonomous mobile robot, representing a curved asphalt pavement crack, are depicted in Figure 9(b). The corresponding tracing error metrics are summarized in Table 8. The PID controller exhibits a tracing error curve characterized by pronounced peaks and substantial fluctuations, indicating

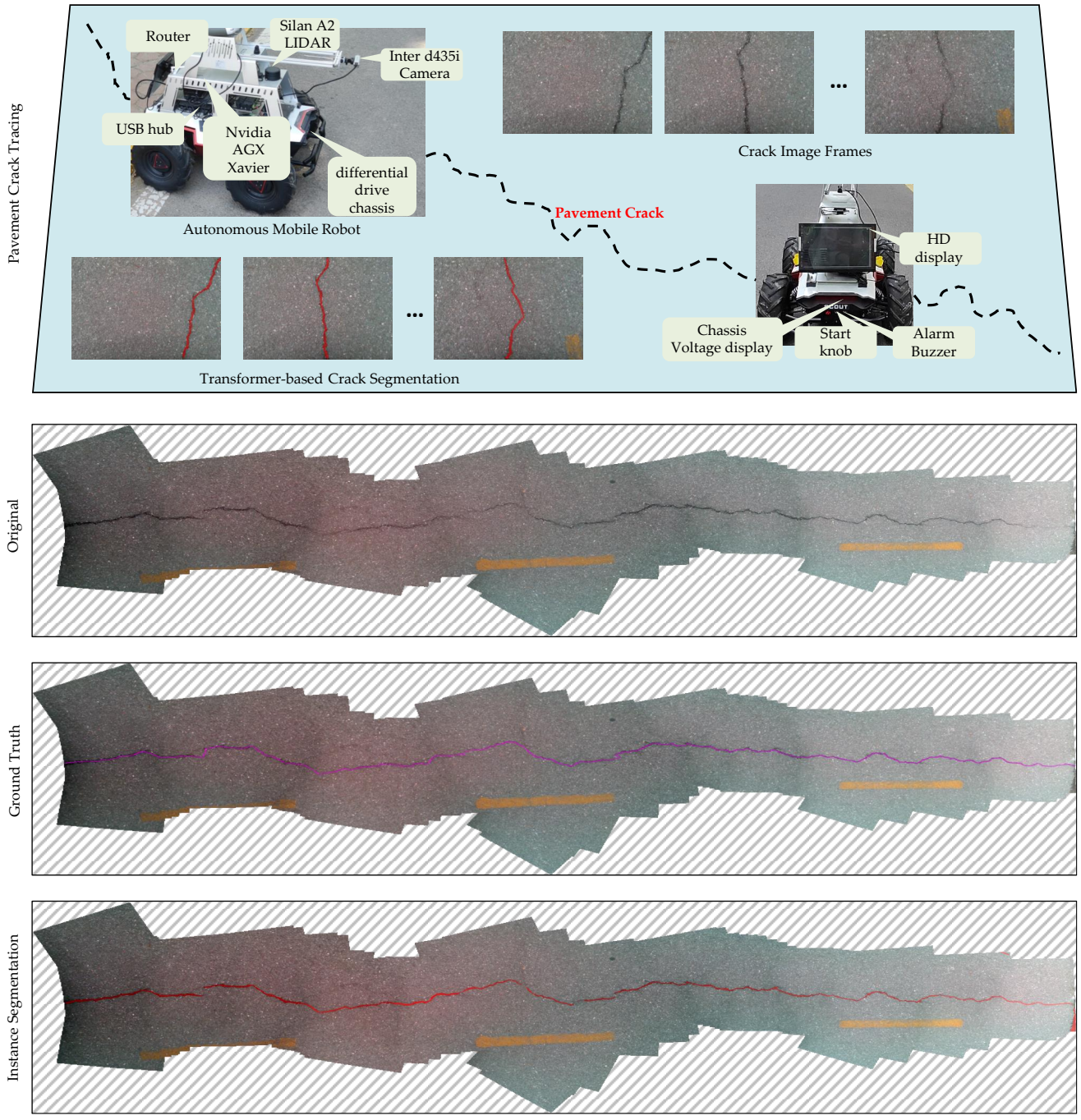



Figure 10: Working conditions of our self-built autonomous mobile robot and demonstration of Original, Ground Truth and Instance Segmentation for Crack #3.

potential limitations in maintaining a smooth crack trajectory. Conversely, the ELM-PID controller displays a stable and smooth tracing error curve, demonstrating its effectiveness in this application. Among the controllers evaluated, the PID controller performs the worst, with a maximum error (MAX) of 75.73 mm, a root mean square error (RMSE) of 11.99 mm, and a mean absolute error (MAE) of 13.58 mm when utilizing the ViT-S2T segmentation model. In contrast, the ELM-PID controller achieves significantly better tracing performance, recording a MAX of 20.31 mm, an RMSE of 4.19 mm, and an MAE of 5.23 mm. These results underscore the effectiveness of the ELM-PID controller compared to the PID controller, highlighting significant improvements with differences of 55.42 mm, 7.8 mm, and 8.35 mm in MAX, RMSE, and MAE, respectively. The ELM-PID controller thus demonstrates the best overall performance metrics, establishing its superiority in precise crack tracking applications. Further exploration of additional controller configurations and segmentation methods may yield even more improvements in tracing accuracy and reliability.

For crack #3, characterized as a discontinuous asphalt pavement crack, the dynamic tuning results for the ELM-PID controller are illustrated in Figure 8(c). The tracing

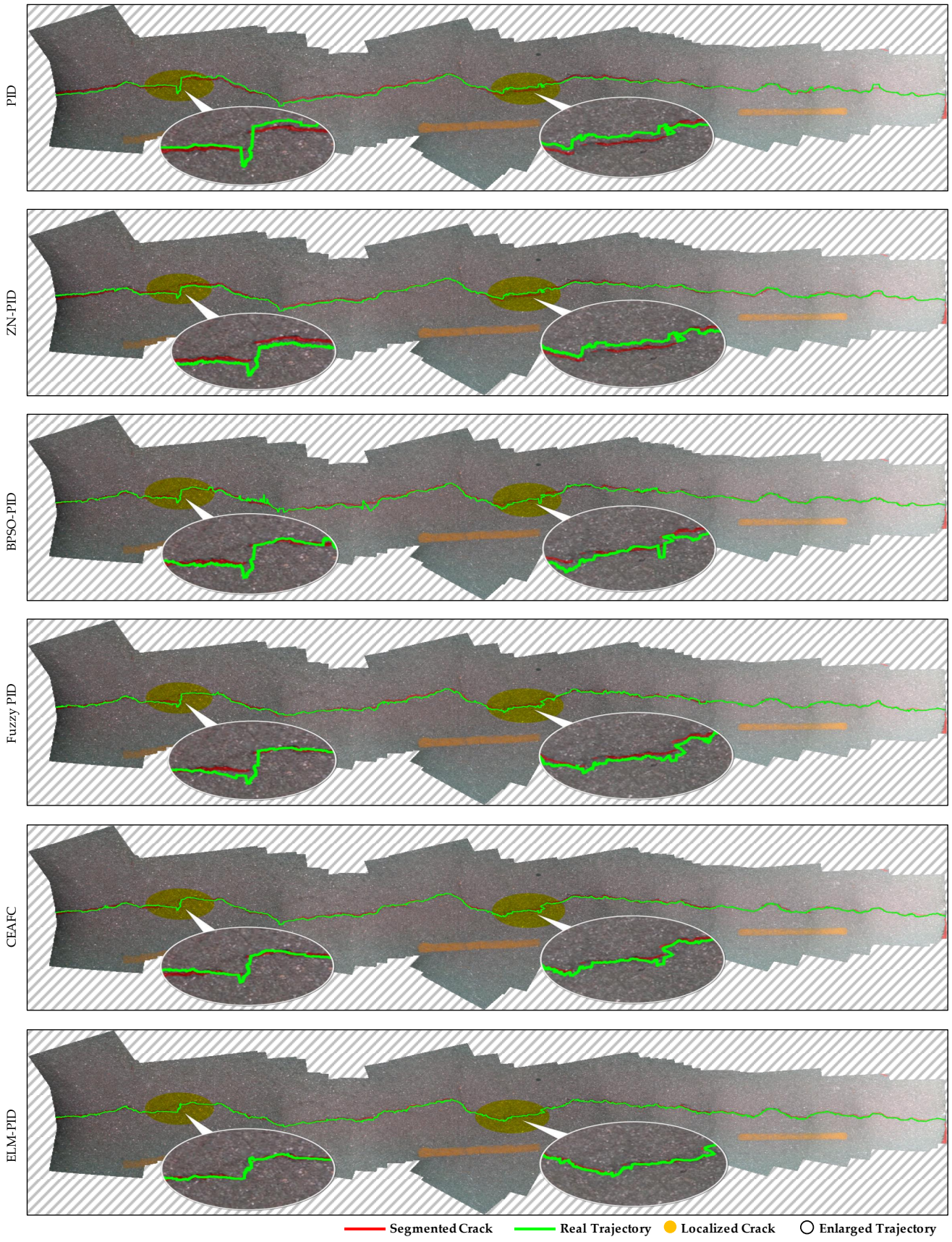
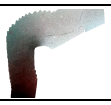
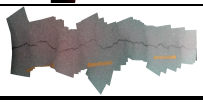



Figure 11: Crack tracing trajectory comparison of different methods for Crack #3, key details are applied local zooming to highlight.

Table 8: Comparative analysis of crack tracing evaluation metrics using the six controllers with three pavement crack conditions under the S2TNet model.

| Crack Scene | Demo | Evaluation Criteria | PID | ZN-PID | BPSO-PID | Fuzzy PID | CEAFC | ELM-PID |
|---|---|---|---|---|---|---|---|---|
| *#1 Slender* |  | *MAX* | 61.58 | 57.24 | 52.46 | 48.69 | 43.38 | **27.59** |
| | | *RMSE* | 12.30 | 12.39 | 7.52 | 11.88 | 10.86 | **6.52** |
| | | *MAE* | 12.56 | 10.49 | 9.17 | 8.13 | 6.87 | **4.99** |
| *#2 Curved* |  | *MAX* | 89.06 | 82.65 | 64.81 | 55.78 | 48.67 | **28.53** |
| | | *RMSE* | 16.34 | 12.53 | 15.32 | 12.38 | 14.25 | **11.35** |
| | | *MAE* | 15.74 | 13.74 | 11.89 | 9.85 | **6.84** | 7.64 |
| *#3 Discontinuous* |  | *MAX* | 105.84 | 88.61 | 80.29 | 76.57 | 61.45 | **42.38** |
| | | *RMSE* | 15.64 | 15.5 | 12.84 | 10.02 | **9.84** | 9.91 |
| | | *MAE* | 19.17 | 16.03 | 15.07 | 10.63 | 8.13 | **6.53** |

errors of six controllers deployed on the self-developed autonomous mobile robot are depicted in Figure 9(c), while the corresponding tracing error metrics are presented in Table 9. Figure 10 provides an overview of the hardware architecture of the autonomous mobile robot and its procedure for tracing asphalt crack trajectories. To create a composite image for crack #3, the proposed method integrates original video frames with experimental results using a stitching algorithm. This results in a comprehensive image that incorporates the original footage, ground truth, and instance segmentation results. Additionally, Figure 11 showcases a panoramic view of the detailed crack tracing process implemented by the six controllers utilizing the ViT-S2T segmentation model in the crack #3 scenario. Among the controllers evaluated, the ELM-PID controller demonstrates the most precise tracing performance, achieving a maximum error (MAX) of 24.95 mm, a root mean square error (RMSE) of 5.42 mm, and a mean absolute error (MAE) of 5.49 mm when applying the ViT-S2T segmentation model. Field experiments conducted on crack #3 further validate the effectiveness and superiority of the proposed ELM-PID crack tracing controller, which benefits from dynamic hyperparameter tuning. These tests highlight the adaptability and enhanced performance of the self-developed autonomous mobile robot in the domain of crack tracing. In stark contrast, the PID controller displays the weakest performance, with a maximum error (MAX) of 75.73 mm, an RMSE of 11.99 mm, and an MAE of 13.58 mm. Meanwhile, the ELM-PID controller consistently achieves superior tracing performance, achieving a MAX of 20.31 mm, an RMSE of 4.19 mm, and an MAE of 5.23 mm. These metrics indicate substantial improvements over the PID controller, exhibiting differences of 55.42 mm, 7.8 mm, and 8.35 mm in MAX, RMSE, and MAE, respectively, underscoring the enhanced capabilities of the ELM-PID controller in precise crack trajectory tracing. These findings reinforce the call for further research to explore additional configurations and methods aimed at optimizing crack detection and tracing efficacy in various pavement scenarios.

Table 10 provides a comprehensive overview of the real-time performance metrics for six controllers utilized in pavement crack tracking within the crack #3 scenario, employing three segmentation models. Notably, the ViT-S2T model demonstrates a significant enhancement in frames per second (FPS), achieving increases of 39.73 and 26.83 FPS when compared to the YOLOv5 and S2TNet models, respectively, ultimately reaching a total of 67.72 FPS. This substantial improvement facilitates real-time processing of crack images on embedded devices, rendering them suitable for practical applications. The analysis incorporates comparisons of critical performance indicators, including linear and

Table 9: Comparative analysis of crack tracing evaluation metrics using the six controllers with three pavement crack conditions under the ViT-S2T model.

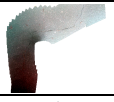
| Crack Scene | Demo | Evaluation Criteria | PID | ZN-PID | BPSO-PID | Fuzzy PID | CEAFC | ELM-PID |
|---|---|---|---|---|---|---|---|---|
| *#1 Slender* |  | *MAX* | 44.94 | 35.41 | 25.85 | 33.76 | 22.62 | **17.6** |
| | | *RMSE* | 10.81 | 8.06 | 6.74 | 5.97 | 4.37 | **3.94** |
| | | *MAE* | 10.67 | 8.51 | 7.54 | 6.36 | 5.17 | **4.48** |
| *#2 Curved* |  | *MAX* | 75.73 | 44.68 | 46.56 | 47.39 | **19.32** | 20.31 |
| | | *RMSE* | 11.99 | 10.16 | 8.27 | 9.07 | 4.86 | **4.19** |
| | | *MAE* | 13.58 | 11.85 | 10.27 | 8.53 | 6.45 | **5.23** |
| *#3 Discontinuous* |  | *MAX* | 79.16 | 69.86 | 66.37 | 59.85 | 39.24 | **24.95** |
| | | *RMSE* | 13.13 | 11.54 | 13.62 | 9.92 | 6.83 | **5.42** |
| | | *MAE* | 17.15 | 14.71 | 12.42 | 9.39 | 7.44 | **5.49** |

Table 10: Real-time performance analysis of crack tracing using the six controllers under three segmentation model for the discontinuous crack.

| Segment Model | FPS (Hz) | Evaluation Criteria | Linear Speed(m/s) | | | | | | Angular Speed(rad/s) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | *PID* | *ZN-PID* | *BPSO-PID* | *Fuzzy PID* | *CEAFC* | *ELM-PID* | *PID* | *ZN-PID* | *BPSO-PID* | *Fuzzy PID* | *CEAFC* | *ELM-PID* |
| YOLOv5 | 27.99 | MAX | 0.1481 | 0.1100 | 0.1125 | 0.1129 | 0.1225 | 0.1080 | 1.1042 | 1.1846 | 1.1064 | 1.0268 | 0.883 | 0.8368 |
| | | MIN | 0.0497 | 0.0359 | 0.0463 | 0.0315 | 0.0357 | 0.0368 | -0.9537 | -0.814 | -0.8999 | -0.9856 | -0.7675 | -0.7982 |
| | | RMSE | 0.1371 | 0.1219 | 0.1237 | 0.1324 | 0.0994 | 0.0978 | 1.298 | 1.2011 | 1.0694 | 1.2879 | 1.0345 | 1.062 |
| | | MAE | 0.0942 | 0.0841 | 0.0915 | 0.0811 | 0.0855 | 0.0865 | 0.6841 | 0.6593 | 0.5632 | 0.6413 | 0.4831 | 0.4028 |
| S2TNet | 40.89 | MAX | 0.1256 | 0.1061 | 0.1168 | 0.1124 | 0.0897 | 0.1242 | 0.9521 | 1.0554 | 0.751 | 1.0221 | 0.8099 | 0.8191 |
| | | MIN | 0.0431 | 0.0433 | 0.0447 | 0.0392 | 0.0209 | 0.0226 | -0.7724 | -0.7747 | -0.7494 | -0.6929 | -0.6222 | -0.667 |
| | | RMSE | 0.1453 | 0.1210 | 0.1370 | 0.1230 | 0.0600 | 0.0567 | 0.9038 | 0.9682 | 0.9821 | 0.8066 | 0.6268 | 0.6118 |
| | | MAE | 0.0777 | 0.0680 | 0.0575 | 0.0685 | 0.0642 | 0.0643 | 0.3213 | 0.3221 | 0.367 | 0.2684 | 0.3716 | 0.2931 |
| ViT-S2T | 67.72 | MAX | 0.0850 | 0.0722 | 0.0718 | 0.0834 | 0.0761 | 0.0794 | 0.8786 | 0.7034 | 0.7926 | 0.8403 | 0.6327 | 0.6049 |
| | | MIN | 0.0269 | 0.0278 | 0.0202 | 0.0190 | 0.0209 | 0.0185 | -0.6563 | -0.6635 | -0.5748 | -0.6049 | -0.5345 | -0.5485 |
| | | RMSE | 0.1040 | 0.1064 | 0.1081 | 0.0824 | 0.0589 | 0.0489 | 0.8656 | 0.7255 | 0.7213 | 0.6931 | 0.4355 | 0.4164 |
| | | MAE | 0.0569 | 0.0553 | 0.0411 | 0.0591 | 0.0420 | 0.0513 | 0.2148 | 0.2168 | 0.2817 | 0.2019 | 0.1622 | 0.1717 |

angular velocities. In this context, negative angular velocity corresponds to a left turn, while positive angular velocity indicates a right turn. The findings reveal that the ELM-PID controller effectively maintains speed control across all three segmentation models, with particularly notable performance when utilizing the ViT-S2T crack trajectory segmentation model. The RMSE and MAE for linear velocity are recorded at 0.0489 and 0.0513, respectively, while for angular velocity, the RMSE and MAE stand at 0.4164 and 0.1717. These metrics all fall within a reasonable range, highlighting the efficacy of the controller. Given the intricate and subtle topological characteristics of crack trajectories, precise speed control is essential for successful tracking. The emphasis lies not merely on achieving high speeds but on ensuring that speeds remain within an optimal range, enabling the autonomous mobile robot to track cracks effectively. In this regard, the ELM-PID controller exhibits commendable real-time tracking performance. Looking ahead, future efforts will concentrate on optimizing speed control further while continuing to enhance tracking efficacy. This ongoing refinement aims to improve the adaptability and reliability of the system in diverse operational conditions, thereby contributing to more effective pavement maintenance and monitoring solutions.

## 5. Conclusions

This study addresses two critical issues in road crack tracing: the loss of slender crack features due to convolution, which leads to fragmented crack trajectories, and the large tracing errors caused by the reliance solely on human-calibrated parameters of PID controllers. To address these challenges, a novel crack tracing framework is proposed. The framework integrates TFEM and ISM, resulting in a ViT-based segmentation network that effectively segments and extracts crack instance features. This allows for the retrieval of complete crack trajectories. Our proposed approach demonstrates significant improvements in segmentation performance on the S2TCrack dataset, achieving a precision of 82.76% and mAP@0.5 of 75.63%. Additionally, an ELM-PID-based crack tracing controller is designed, utilizing an ELM model that employs backpropagation neural networks to optimize control parameters and facilitate adaptive tuning of PID parameters. However, the proposed method has certain limitations. This study focuses on single crack features and does not address more complex scenarios such as multiple or intersecting cracks, while the selected crack background environment is relatively ideal. The real-time performance of crack tracing is still limited, and some delays are present. Future research will target complex topological cracks and aim to conduct experiments under various disturbance conditions in pavement crack environments. Furthermore, a deeper investigation into the integration of embedded deep learning models with optimization control methods will be pursued to enhance the real-time performance of crack tracing. Additionally, repair devices will be incorporated into the self-developed autonomous mobile robot to enable automatic crack sealing.

## Acknowledgements

The study presented in the article was partially supported by the National Key Research and Development Program of China (No.2021YFB2601000), National Natural Science Foundation of China (No.52078049, No.52378431, No. 52408454), Fundamental Research Funds for the Central Universities, CHD (No.300102210302, No.300102210118, No.300102244712), the 111 Project of Sustainable Transportation for Urban Agglomeration in Western China (No.B20035), the 111 Project of Low Carbon Smart Road

Infrastructure Construction and Maintenance Discipline Innovation and Talent Introduction Base of Shaanxi Province, Scientific Innovation Practice Project of Postgraduates of Changan University (No.300103724053).

## References


[1] J. Zhang, X. Yang, W. Wang, J. Guan, W. Liu, H. Wang, L. Ding, V. C. S. Lee, Cross-entropy-based adaptive fuzzy control for visual tracking of road cracks with unmanned mobile robot, Computer-Aided Civil and Infrastructure Engineering 39 (2024) 891–910. doi:10.1111/mice.13108.

[2] J. Zhang, X. Yang, W. Wang, J. Guan, L. Ding, V. C. S. Lee, Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering, Automation in Construction 146 (2023) 104699. doi:10.1016/j.autcon.2022.104699.

[3] X. Yang, J. Guan, L. Ding, Z. You, V. C. Lee, M. R. M. Hasan, X. Cheng, Research and applications of artificial neural network in pavement engineering: A state-of-the-art review, Journal of Traffic and Transportation Engineering (English Edition) 8 (2021) 1000–1021. doi:10.1016/j.jtte.2021.03.005.

[4] J. Guan, X. Yang, L. Ding, X. Cheng, V. C. Lee, C. Jin, Automated pixel-level pavement distress detection based on stereo vision and deep learning, Automation in Construction 129 (2021) 103788. doi:10.1016/j.autcon.2021.103788.

[5] J. Liu, X. Yang, S. Lau, X. Wang, S. Luo, V. C.-S. Lee, L. Ding, Automated pavement crack detection and segmentation based on two-step convolutional neural network, Computer-Aided Civil and Infrastructure Engineering 35 (2020) 1291–1305. doi:10.1111/mice.12622.

[6] J. Liu, X. Yang, X. Wang, J. W. Yam, A laboratory prototype of automatic pavement crack sealing based on a modified 3D printer, International Journal of Pavement Engineering 23 (2022) 2969–2980. doi:10.1080/10298436.2021.1875225.

[7] J. Guan, X. Yang, V. C. Lee, W. Liu, Y. Li, L. Ding, B. Hui, Full field-of-view pavement stereo reconstruction under dynamic traffic conditions: Incorporating height-adaptive vehicle detection and multi-view occlusion optimization, Automation in Construction 144 (2022) 104615. doi:10.1016/j.autcon.2022.104615.

[8] J. Guan, X. Yang, P. Liu, M. Oeser, H. Hong, Y. Li, S. Dong, Multi-scale asphalt pavement deformation detection and measurement based on machine learning of full field-of-view digital surface data, Transportation Research Part C: Emerging Technologies 152 (2023) 104177. doi:10.1016/j.trc.2023.104177.

[9] Z. Pan, J. Guan, X. Yang, K. Fan, J. C. Ong, N. Guo, X. Wang, One-stage 3D profile-based pavement crack detection and quantification, Automation in Construction 153 (2023) 104946. doi:10.1016/j.autcon.2023.104946.

[10] S. Bai, M. Ma, L. Yang, Y. Liu, Pixel-wise crack defect segmentation with dual-encoder fusion network, Construction and Building Materials 426 (2024) 136179. doi:10.1016/j.conbuildmat.2024.136179.

[11] Y. Yan, S. Zhu, S. Ma, Y. Guo, Z. Yu, CycleADC-net: A crack segmentation method based on multi-scale feature fusion, Measurement 204 (2022) 112107. doi:10.1016/j.measurement.2022.112107.

[12] L. Wang, X. Yang, G. Gong, J. Du, Pose and trajectory control of shield tunneling machine in complicated stratum, Automation in Construction 93 (2018) 192–199. doi:10.1016/j.autcon.2018.05.020.

[13] H. Feng, W. Ma, C. Yin, D. Cao, Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller, Automation in Construction 127 (2021) 103722. doi:10.1016/j.autcon.2021.103722.

[14] L. Song, H. Sun, J. Liu, Z. Yu, C. Cui, Automatic segmentation and quantification of global cracks in concrete structures based on deep learning, Measurement 199 (2022) 111550. doi:10.1016/j.measurement.2022.111550.

[15] T. He, H. Li, Z. Qian, C. Niu, R. Huang, Research on weakly supervised pavement crack segmentation based on defect location by generative adversarial network and target re-optimization, Construction and Building Materials 411 (2024) 134668. doi:10.1016/j.conbuildmat.2023.134668.

[16] C. Xiang, V. J. Gan, J. Guo, L. Deng, Semi-supervised learning framework for crack segmentation based on contrastive learning and cross pseudo supervision, Measurement 217 (2023) 113091. doi:10.1016/j.measurement.2023.113091.

[17] E. Asadi Shamsabadi, C. Xu, D. Dias-da-Costa, Robust crack detection in masonry structures with transformers, Measurement 200 (2022) 111590. doi:10.1016/j.measurement.2022.111590.

[18] H. Liu, X. Miao, C. Mertz, C. Xu, H. Kong, CrackFormer: Transformer Network for Fine-Grained Crack Detection, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3763–3772. doi:10.1109/ICCV48922.2021.00376.

[19] Z. Zhou, J. Zhang, C. Gong, Hybrid semantic segmentation for tunnel lining cracks based on Swin Transformer and convolutional neural network, Computer-Aided Civil and Infrastructure Engineering 00 (2023) 1–20. doi:10.1111/mice.13003.

[20] W. Wang, C. Su, Automatic concrete crack segmentation model based on transformer, Automation in Construction 139 (2022) 104275. doi:10.1016/j.autcon.2022.104275.

[21] D. Chu, H. Li, C. Zhao, T. Zhou, Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control With PID Feedback, IEEE Transactions on Intelligent Transportation Systems 24 (2022) 1–12. doi:10.1109/TITS.2022.3150365.

[22] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, L. Zhang, A PID Controller Approach for Stochastic Optimization of Deep Networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, 2018, pp. 8522–8531. doi:10.1109/CVPR.2018.00889.

[23] G. Peng, H. Duan, Z. Tan, Y. Zhou, J. Li, B. Hu, C. Zhou, Construction path tracking and pose estimation of unmanned bulldozer, Automation in Construction 154 (2023) 105015. doi:10.1016/j.autcon.2023.105015.

[24] F. Liu, W. Liu, H. Luo, Operational stability control of a buried pipeline maintenance robot using an improved PSO-PID controller, Tunnelling and Underground Space Technology 138 (2023) 105178. doi:10.1016/j.tust.2023.105178.

[25] S. Guan, J. Wang, X. Wang, M. Shi, W. Lin, W. Chen, Dynamic hyper-parameter tuning-based path tracking control for robotic rollers working on earth-rock dam under complex construction conditions, Automation in Construction 143 (2022) 104576. doi:10.1016/j.autcon.2022.104576.

[26] J. Yoon, J. Doh, Optimal PID control for hovering stabilization of quadcopter using long short term memory, Advanced Engineering Informatics 53 (2022) 101679. doi:10.1016/j.aei.2022.101679.

[27] X. Zhang, X. Yang, C. Huang, J. Cao, H. Liu, Adaptive fuzzy finite-time PID backstepping control for chaotic systems with full states constraints and unmodeled dynamics, Information Sciences 661 (2024) 120148. doi:10.1016/j.ins.2024.120148.

[28] A. Gün, Attitude control of a quadrotor using PID controller based on differential evolution algorithm, Expert Systems with Applications 229 (2023) 120518. doi:10.1016/j.eswa.2023.120518.

[29] L. Sun, Y. Yang, G. Zhou, A. Chen, Y. Zhang, W. Cai, L. Li, An integration–competition network for bridge crack segmentation under complex scenes, Computer-Aided Civil and Infrastructure Engineering 00 (2023) 1–18. doi:10.1111/mice.13113.

[30] T. Zhang, D. Wang, Y. Lu, ECSNet: An Accelerated Real-Time Image Segmentation CNN Architecture for Pavement Crack Detection, IEEE Transactions on Intelligent Transportation Systems (2023) 1–8. doi:10.1109/TITS.2023.3300312.

[31] E. Asadi Shamsabadi, S. M. H. Erfani, C. Xu, D. Dias-da-Costa, Efficient semi-supervised surface crack segmentation with small datasets based on consistency regularisation and pseudo-labelling, Automation in Construction 158 (2024) 105181. doi:10.1016/j.autcon.2023.105181.

[32] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, A Survey on Vision

Transformer, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 87–110. doi:10.1109/TPAMI.2022.3152247.
[33] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, Z. He, A Survey of Visual Transformers, IEEE Transactions on Neural Networks and Learning Systems (2023) 1–21. doi:10.1109/TNNLS.2022.3227717.
[34] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, M. Shah, Transformers in Vision: A Survey, ACM Computing Surveys 54 (2022) 200:1–200:41. doi:10.1145/3505244.
[35] Y. Lu, W. Qin, C. Zhou, Z. Liu, Automated detection of dangerous work zone for crawler crane guided by UAV images via Swin Transformer, Automation in Construction 147 (2023) 104744. doi:10.1016/j.autcon.2023.104744.
[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2020). doi:10.48550/arXiv.2010.11929. arXiv:2010.11929.
[37] J. Alcaina, A. Cuenca, J. Salt, V. Casanova, R. Pizá, Delay-independent dual-rate PID controller for a packet-based networked control system, Information Sciences 484 (2019) 27–43. doi:10.1016/j.ins.2019.01.059.
[38] C. Zhao, L. Guo, Towards a theoretical foundation of PID control for uncertain nonlinear systems, Automatica 142 (2022) 110360. doi:10.1016/j.automatica.2022.110360.
[39] X. Lyu, Z. Lin, PID Control of Planar Nonlinear Uncertain Systems in the Presence of Actuator Saturation, IEEE/CAA Journal of Automatica Sinica 9 (2022) 90–98. doi:10.1109/JAS.2021.1004281.
[40] I. AL-Wesabi, F. Zhijian, H. M. H. Farh, W. Zhiguo, K. Ameur, A. A. Al-Shamma'a, A. M. Al-Shaalan, Dynamic global power extraction of partially shaded PV system using a hybrid MPSO-PID with anti-windup strategy, Engineering Applications of Artificial Intelligence 126 (2023) 106965. doi:10.1016/j.engappai.2023.106965.
[41] Y. Ye, C.-B. Yin, Y. Gong, J.-j. Zhou, Position control of nonlinear hydraulic system using an improved PSO based PID controller, Mechanical Systems and Signal Processing 83 (2017) 241–259. doi:10.1016/j.ymssp.2016.06.010.
[42] A. Ambroziak, A. Chojecki, The PID controller optimisation module using Fuzzy Self-Tuning PSO for Air Handling Unit in continuous operation, Engineering Applications of Artificial Intelligence 117 (2023) 105485. doi:10.1016/j.engappai.2022.105485.
[43] I. Carlucho, M. De Paula, G. G. Acosta, Double Q-PID algorithm for mobile robot control, Expert Systems with Applications 137 (2019) 292–307. doi:10.1016/j.eswa.2019.06.066.
[44] X. Zhou, D. Li, L. Zhang, Q. Duan, Application of an adaptive PID controller enhanced by a differential evolution algorithm for precise control of dissolved oxygen in recirculating aquaculture systems, Biosystems Engineering 208 (2021) 186–198. doi:10.1016/j.biosystemseng.2021.05.019.
[45] R. Bouzaiene, S. Hafsi, F. Bouani, Adaptive neural network PID controller for nonlinear systems, in: 2021 IEEE 2nd International Conference on Signal, Control and Communication (SCC), 2021, pp. 264–269. doi:10.1109/SCC53769.2021.9768352.
[46] B. Mohamed, K. Kara, A. Oussama, L. Hadjili, Adaptive Neural Network PID Controller, in: 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2019, pp. 1–6. doi:10.1109/EEEIC.2019.8783803.
[47] X. Jin, K. Chen, Y. Zhao, J. Ji, P. Jing, Simulation of hydraulic transplanting robot control system based on fuzzy PID controller, Measurement 164 (2020) 108023. doi:10.1016/j.measurement.2020.108023.
[48] S. Husnain, M. T. Riaz, M. A. Khan, M. Mazhar, An Artificial Bee Colony Based PID Tuning Algorithm for Inverted Pendulum and Cart System, in: 2023 17th International Conference on Open Source Systems and Technologies (ICOSST), 2023, pp. 1–7. doi:10.1109/ICOSST60641.2023.10414233.
[49] Ultralytics, ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, https://github.com/ultralytics/yolov5.com, 2022. URL: https://doi.org/10.5281/zenodo.7347926. doi:10.5281/zenodo.7347926, accessed: 7th May, 2023.
[50] L. Ke, M. Danelljan, X. Li, Y.-W. Tai, C.-K. Tang, F. Yu, Mask Transfiner for High-Quality Instance Segmentation, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 4402–4411. doi:10.1109/CVPR52688.2022.00437.
[51] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (2006) 489–501. doi:10.1016/j.neucom.2005.12.126.
[52] Y. Zhang, W. Zhang, Y. Li, L. Wen, X. Sun, AF-OS-ELM-MVE: A new online sequential extreme learning machine of dam safety monitoring model for structure deformation estimation, Advanced Engineering Informatics 60 (2024) 102345. doi:10.1016/j.aei.2023.102345.
[53] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153. doi:10.1016/j.neucom.2019.01.036.
[54] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic Road Crack Detection Using Random Structured Forests, IEEE Transactions on Intelligent Transportation Systems 17 (2016) 3434–3445. doi:10.1109/TITS.2016.2552248.
[55] J. Zhang, X. Yang, W. Wang, I. Brilakis, D. Davletshina, H. Wang, M. Cao, Segment-to-track for pavement crack with light-weight neural network on unmanned wheeled robot, Automation in Construction 161 (2024) 105346. doi:10.1016/j.autcon.2024.105346.
[56] T. F. Lam, H. Blum, R. Siegwart, A. Gawel, SL Sensor: An open-source, real-time and robot operating system-based structured light sensor for high accuracy construction robotic applications, Automation in Construction 142 (2022) 104424. doi:10.1016/j.autcon.2022.104424.
[57] D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, YOLACT: Real-Time Instance Segmentation, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9156–9165. doi:10.1109/ICCV.2019.00925.
[58] X. Wang, R. Zhang, C. Shen, T. Kong, L. Li, SOLO: A Simple Framework for Instance Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2021) 1–1. doi:10.1109/TPAMI.2021.3111116.
[59] S. MohandSaidi, R. Mellah, Real-Time Speed Control of a Mobile Robot Using PID Controller, in: Artificial Intelligence and Its Applications, volume 413, Springer International Publishing, Cham, 2022, pp. 548–556. doi:10.1007/978-3-030-96311-8_51.
[60] P. M. Meshram, R. G. Kanojiya, Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor, in: IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012), 2012, pp. 117–122.
[61] J. Han, X. Shan, H. Liu, J. Xiao, T. Huang, Fuzzy gain scheduling PID control of a hybrid robot based on dynamic characteristics, Mechanism and Machine Theory 184 (2023) 105283. doi:10.1016/j.mechmachtheory.2023.105283.