Advanced Engineering Informatics 68 (2025) 103589

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Adaptive drone inspection strategy for bridge based on multi-level representation learning

Wang Chen [a], Xin Zhang [b], Binhong Yuan [b], Jian Zhang [a,c,*]

[a] School of Civil Engineering, Southeast University, Nanjing, China
[b] Guangdong Jiaoke Testing Co. Ltd, Guangzhou, China
[c] Advanced Ocean Institute of Southeast University, 226000, Nantong, China

ARTICLE INFO

Keywords:
Bridges
UAV
Adaptive
Inspection

ABSTRACT

Recently, UAV-based intelligent bridge inspection technology has seen widespread application. However, balancing efficiency and accuracy in inspection tasks remains a significant challenge. This study, driven by adaptivity, introduces a novel UAV inspection logic and a multi-level representation learning-based strategy, split into two stages: automatic rough inspection and adaptive fine detection. The strategy incrementally learns the spatial structure, component characteristics, and defect features of bridges. During the rough inspection, the UAV rapidly identifies component properties by integrating spatial clustering post-processing methods with point cloud semantic networks. Next, simulated field-of-view models and dimensionality reduction techniques compress the point cloud space, guiding the UAV to perform spatial inspections in planar geometry. In the fine detection stage, the hybrid Light-PVIT structure, optimized for spatial and channel dimensions, extracts defect features identified during the rough inspection. This prior information directs the UAV to conduct detailed inspections of defect areas. This strategy markedly enhances the efficiency and accuracy of bridge inspections, offering dependable technical support for bridge maintenance.

1. Introduction

Bridges are essential components of transportation networks, crucial for public safety and economic efficiency. However, factors like environmental conditions, vehicle loads, and construction activities make bridges susceptible to defects such as cracks, spalling, and corrosion. These defects can compromise structural integrity and load-bearing capacity, potentially leading to serious safety incidents. Thus, regular inspections are essential for bridge stability and safety [1]. Bridge inspections cover various components, including the deck system, superstructure, and substructure. Traditional methods for inspecting the superstructure, often located at high elevations, rely on technicians using inspection vehicles or other aerial platforms [2]. These conventional methods have several drawbacks: they require costly equipment and specialized personnel, leading to high expenses and extended inspection times. Additionally, inspection platforms often can't provide a comprehensive view of the entire bridge, risking missed critical defects or hazards. Moreover, inspection results can be subjective and influenced by the engineer's experience [3,4]. Consequently, there is an urgent need in the engineering field to develop efficient and objective technologies for regular bridge inspections, particularly for high-elevation applications.

To tackle this issue, researchers are developing intelligent inspection tools that utilize advanced machine vision and sensor technology to enhance the accuracy and efficiency of bridge inspections. For bridge substructure inspections, Baptiste et al. [5] created a semi-autonomous mobile robot system, comprising a mobile truck and a collection device, to automatically identify damage under bridges. Choi et al. [6] employed a camera with a zip-line structure and a spherical drone to swiftly detect defects beneath bridges. Leibbrandt et al. [7] combined a vortex adhesion mechanism with a wheel electrode sensor to design a crawling robot for detailed substructure inspections. Xu et al. [8] proposed an innovative climbing robot for inspecting high-altitude suspension cables. Nguyen et al. [9] developed a tank-like robot with multiple sensors that can stably climb steel cables and collect detailed data on surface defects. Tian et al. [10] used drones with high-resolution cameras for long-distance observation of cable structures. Jiang et al. [11,12] designed a UAV system with new sensors and a stereo-vision inertial fusion method to detect damage on tall bridge towers. Jang et al. [13] developed an annular climbing robot equipped with multi-

[*] Corresponding author at: School of Civil Engineering, Southeast University, Nanjing, China.
E-mail address: jian@seu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.aei.2025.103589
Received 19 August 2024; Received in revised form 27 May 2025; Accepted 23 June 2025
Available online 30 June 2025
1474-0346/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

vision cameras, climbing hardware, and control computers for assessing cracks in high-rise bridge piers.

Most UAV operations are manually controlled, which, while easy to implement, results in low inspection efficiency, safety risks for inspectors who must maintain visual contact, and potential errors leading to incomplete data [14]. To address these drawbacks of manual control and enhance inspection efficiency and data completeness, recent research has increasingly focused on the development of autonomous UAV flight. Morgenthal et al. [15] initially developed a preliminary model of the structure. Then, using a method combining horizontal intersection of the polygon mesh with graph optimization, they generated flight paths to facilitate automated image acquisition. Similarly, Lin et al. [16] enabled automated 3D flight plan generation based on an initial bridge model by having technical personnel mark areas of interest and input key structural information. In contrast, Wang et al. [17] proposed a novel 3D path planning method based on Building Information Modeling (BIM) to enhance the quality of bridge photogrammetric models by optimizing UAV flight plans. Li et al. [18] conducted a site pre-check to assess optimal flight conditions and then planned a continuous UAV flight path to automatically capture high-definition images of bridge surfaces. Horton et al. [19] manually completed the mission planning module for automated data collection in bridge inspection by defining a set of waypoints to ensure full structural coverage, which they validated in an indoor laboratory setting. Extending these approaches, Yiğit et al. [20] adopted a fully automated double-grid approach to capture aerial imagery, enabling comprehensive 3D photogrammetric studies. Tse et al. [21] utilized LiDAR to generate a 3D model of the target structure, incorporating viewpoint planning to determine optimal geometry-based viewpoints and applying a traveling salesman problem (TSP) approach to sequence these viewpoints into an efficient inspection path.

The primary challenge in bridge data analysis is the large volume and complexity of the data, which hinders effective feedback for routine inspection and maintenance. Most of the data is stored in image form, and traditional analysis techniques still rely on manual processing. To address this, the academic community has focused on deep learning technologies. Deep learning networks can automatically extract useful features from large-scale data, facilitating automatic identification and significantly improving processing efficiency and accuracy. Research in defect detection focuses on three main tasks: classification, detection, and segmentation. Classification tasks involve categorizing input images into predefined defect conditions. Cha et al. [22] used a deep convolutional neural network (DCNN) to automatically learn image features, addressing challenges like lighting and shadow variations in concrete crack detection. Xu et al. [23] proposed a deep learning framework based on a restricted Boltzmann machine (RBM) to learn abstract features and map image elements to state representation vectors, identifying cracks in steel box girder images with complex backgrounds. Ni et al. [24] introduced a dual-scale convolutional neural network for automatic crack identification in complex scenarios, validated through lab experiments. Wu et al. [25,26] applied advanced machine learning and deep learning models for automated defect classification. Detection tasks not only categorize images but also predict bounding boxes to locate defects. He et al. [27] integrated a multi-scale attention module and an oriented bounding box-based segmentation strategy into the YOLOv4 object detection network, reducing computational complexity and enabling automated detection of structural defects like cracks and exposed rebar. He et al. [28] proposed an anchor-free object detection framework, CenWholeNet, considering central and overall features to identify defect characteristics. Pang et al. [29] developed a multi-scale feature fusion model to improve recognition accuracy and robustness in Faster R-CNN for object detection. Segmentation tasks classify each pixel in an image to precisely determine defect contours and locations. Choi et al. [30] developed a deep learning-driven concrete crack image segmentation method using advanced convolution techniques, significantly improving accuracy. Dong et al. [31] introduced a crack intelligent segmentation model to enhance pavement crack detection in noisy environments, showing outstanding performance. He et al. [32] created an advanced segmentation model, BGCrack, enhancing pixel-level crack recognition by incorporating boundary features.

UAV-based bridge inspection methods often lack integration between inspection operations and data analysis. While inspection data forms the basis for analysis, results are seldom fed back into the workflow to prioritize high-risk areas, leading to path planning that depends on manual intervention or exhaustive, full-coverage data collection based on structural features. These methods not only waste inspection resources but also escalate the scale and complexity of data processing. Therefore, seamless integration of inspection and analysis, improved operational intelligence, and optimized resource utilization are pressing challenges in this field. To address these limitations, this study proposes an integrated, closed-loop UAV inspection system that combines UAV patrol operations with data analysis, enhancing inspection efficiency and intelligence. This system employs multi-layer representation learning to comprehensively analyze bridge spatial structure, component attributes, and defect characteristics, optimizing processes from data collection to processing. By reducing processing demands from 3D space to 2D representations, the system achieves lightweight processing, significantly decreasing computational load and hardware requirements. This improvement ensures that inspection data is rapidly analyzed and promptly reintegrated into the inspection workflow, supporting continuous refinement of bridge inspection capabilities. The proposed UAV inspection strategy automatically identifies high-priority areas and performs targeted inspections without manual intervention, advancing both the efficiency and accuracy of the on-site inspection phase. Simultaneously, bridge, component, and defect data are automatically analyzed, significantly reducing the workload of engineering technicians during the data analysis phase.

This paper systematically examines the application of UAVs in bridge inspection across six sections. Section 2 offers a comprehensive overview of the methodological framework. Section 3 details the technical methods for UAV automatic rough inspections. Section 4 introduces the newly proposed Light-PVIT. Section 5 presents the key technical elements of the adaptive detailed inspection strategy. Section 6 validates the proposed methods through detailed field experiment reports. Finally, Section 7 summarizes the research findings and discusses future research directions.

2. Proposed methodology

UAV-based intelligent bridge inspection technologies have become increasingly popular in recent years. However, existing methods struggle to balance operational efficiency and detection accuracy. This study introduces a novel UAV adaptive inspection strategy, leveraging multi-level representation learning to optimize both efficiency and accuracy in bridge inspections. This strategy features an innovative two-stage inspection process, as depicted in Fig. 1. The first stage involves an automatic rough inspection based on spatial understanding. A UAV equipped with a wide-field camera captures geometric information, and the collected images are then processed on a ground station server using Structure from Motion (SfM) algorithms to quickly reconstruct a 3D model of the bridge. It then performs an analysis of component-level attributes using a point cloud instance segmentation framework, enhanced with clustering algorithms, to achieve extraction of structural elements. To ensure complete visual coverage, the strategy incorporates field-of-view (FOV) simulation and geometric space reduction projection techniques. By projecting segmented 3D components onto a 2D plane, the system simplifies the complex spatial layout and enables the automatic generation of coverage-optimized inspection paths tailored to each component. These trajectories account for component geometry and orientation, ensuring that all critical surfaces are fully visible within the planned flight paths. This spatial planning guarantees that UAVs achieve thorough inspection, regardless of FOV constraints or structural

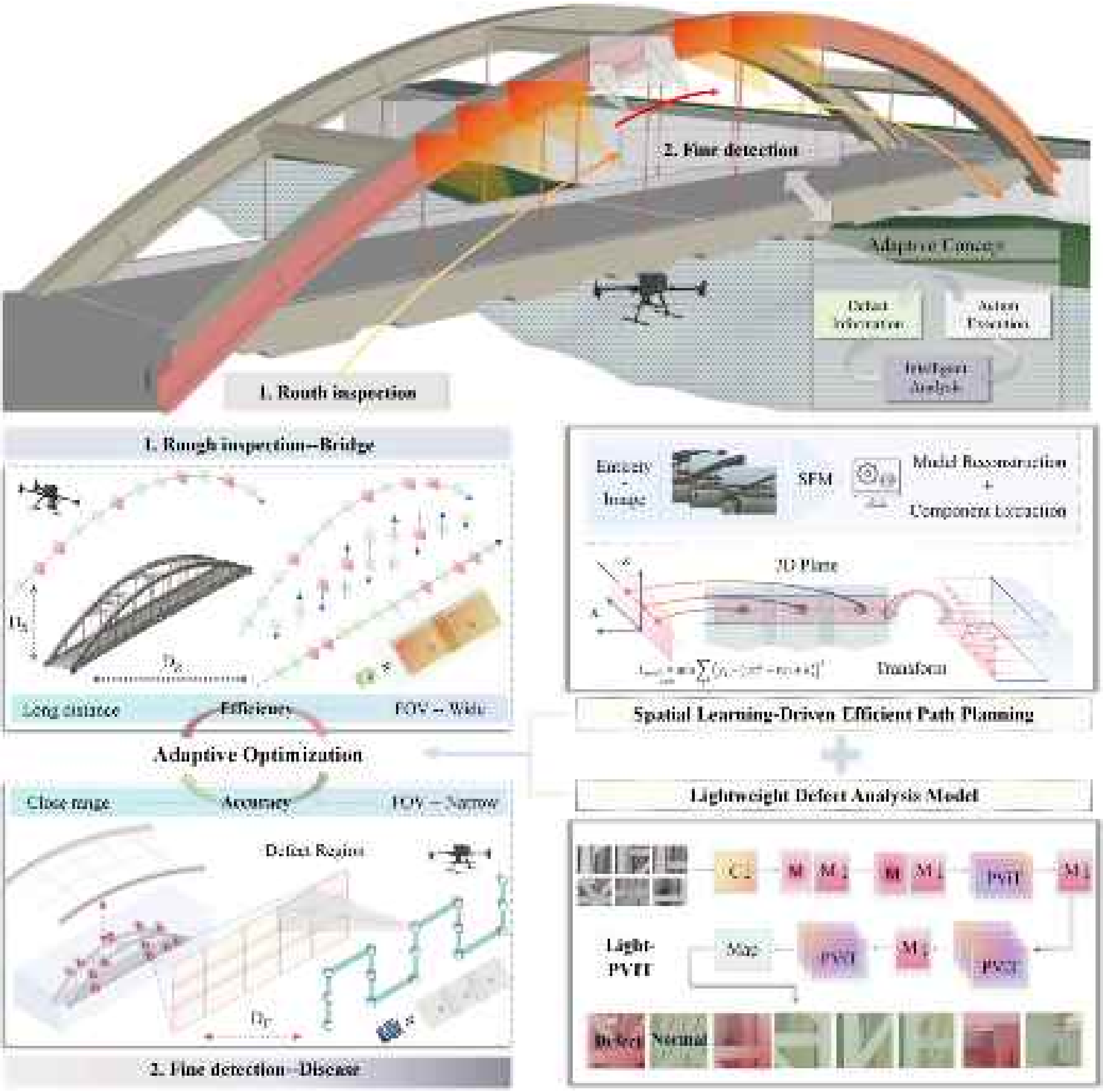


Fig. 1. Framework of adaptive drone inspection strategy.

complexity. The second stage involves adaptive fine detection based on defect information feedback. This study introduces a novel defect recognition approach by reorganizing the self-attention mechanism that operates across both spatial and channel dimensions, resulting in the development of a lightweight model called Light-PVIT. The reorganized self-attention module applies the concept of dilated convolutions to the transformer architecture, preserving only the essential spatial feature information. Additionally, it introduces a strategy for managing channel redundancy through partial splitting operations based on a predefined splitting factor, efficiently dividing the compressed features into global modeling and local modeling components. This approach not only reduces the number of model parameters but also enhances computational efficiency, particularly in the early stages where feature redundancy is more pronounced. By combining identified defect areas with established spatial transformation criteria, it adaptively adjusts inspection paths, guiding UAVs in detailed detection using smaller FOV. This dual-layer inspection approach ensures that healthy regions of the bridge are inspected quickly, while high-risk areas receive the focused attention they require. The method reduces unnecessary computational load and inspection time without sacrificing the precision needed for critical defect detection, offering an efficient and accurate solution for bridge inspection challenges.

3. Spatial learning-driven efficient path planning

With rapid technological advancement, UAV technology is increasingly applied in various industries, especially in bridge inspection. However, traditional manual UAV inspections face significant limitations and risks due to operational dependency and FOV restrictions [14]. The complex structure of bridges necessitates UAVs to operate outside the operator's visual range. Consequently, operators primarily depend on the UAV's camera feed for navigation. However, environmental conditions and operator proficiency can make it difficult to maintain consistent and optimal camera coverage, which affects the precision of bridge inspections. Moreover, the local FOV limits the operator's ability to determine the UAV's spatial position, leading to potential overlaps or missed areas during inspections. The constrained camera's angle also impairs timely detection and avoidance of obstacles, thereby increasing operational risks. The need for more efficient and safer UAV-based automated inspection methods is evident. To address these challenges, this study integrates satellite maps with prior bridge information, including structural design blueprints and on-site inspection data, for

UAV path planning. Satellite maps offer a macro-level view of the geographical landscape, helping to identify the general layout of the bridge and its surroundings. By combining satellite data with prior information, such as potential hazards (e.g., high-voltage power towers or columns), structures and obstacles can be distinguished. This allows for a more informed flight path design that ensures UAVs avoid dangerous areas. Additionally, the bridge's structural characteristics, such as span length and height, provide necessary constraints for flight paths, ensuring they maintain safe distances from potential hazards while operating data collection tasks. Feature extraction is performed on each image using algorithms like SIFT [33] or SURF [34]. These feature points are matched between image pairs to establish correspondences, which are then used to calculate the position of each feature point in the 3D coordinate system, constructing a 3D model of the entire scene.

3.1. *Fast-track spatial recognition of key bridge components*

The model is composed of points distributed in three-dimensional space, each with spatial coordinates (X, Y, Z) and color information. While this model captures the bridge's overall structural information, relying solely on it for UAV inspection is complex and inefficient. The primary goal of inspection tasks is to assess the service status of bridge components to ensure safety and functionality. Thus, overall structural information alone is inadequate for efficient inspections, necessitating detailed learning of bridge component attributes.

Humans can intuitively understand the attributes of each point in a point cloud model and how these points form various bridge components, such as roads, beams, hangers, braces, and arches. In contrast, computers lack the intuitive ability and must rely on algorithms to analyze the spatial relationships between points, classify structural elements, and synthesize this information to understand the bridge structure. Traditional point cloud segmentation methods, such as attribute methods [35], region segmentation [36], and model fitting [37], often depend on predefined rules or models. These methods struggle with complex or irregular point cloud data, requiring extended processing time for large-scale datasets [38].

Recent advancements in deep learning technology have significantly enhanced point cloud segmentation. Deep convolutional neural networks (CNNs) now provide end-to-end solutions for efficiently processing unstructured point cloud data, leading to more precise and robust segmentation of complex scenes [39,40]. Many studies have improved network recognition by designing intricate local feature extraction structures and building larger models [41,42]. However, these approaches often require substantial resources for processing large scene models. This study utilizes PointNeXt [43] to represent bridge component information, balancing model accuracy and size effectively.

Inspired by the U-Net architecture [44], the PointNeXt structure uses an encoder-decoder design to process point cloud data. The encoder captures high-level semantic features, while the decoder remaps these features to construct detailed segmentation structures. The network architecture is shown in Fig. 2. In $[N,x]$, N represents the number of point clouds output by each layer, and x represents the feature dimension. The encoder comprises Multi-Layer Perceptron (MLP) layers, a Set Abstraction (SA) layer, and an Inverse Residual MLP (IRM) layer. The SA module is central to the encoder, featuring three key components: a farthest point sampling layer, a spherical grouping layer, and a feature extraction layer. The sampling layer uses the farthest point algorithm to select central points of 1/4 of the point cloud, retaining overall structural features. The spherical grouping layer creates spherical neighborhoods with a radius of R around each central point, sampling K feature points to capture local details. The feature extraction layer uses max-pooling to extract features from local areas. While this may cause partial feature loss, expanding feature dimensions with MLP before pooling helps compensate for this loss. The IRM layer enhances features on top of the SA layer by introducing residual connections to address the vanishing gradient problem. This design removes the sampling layer and adds two MLP layers, using a larger radius R for spherical neighborhoods to expand the model's receptive field and establish robust local feature relationships. Stacking MLP layers enhances feature extraction but increases computational load. To optimize this, the model employs a separable MLP structure for gradual feature extraction. MLP layers between the grouping layer and pooling layer focus on extracting features between neighborhoods, while subsequent MLP layers extract features of single points. This strategy enhances feature extraction and maintains computational efficiency.

The decoder uses distance interpolation and skip connections, incorporating multiple Feature Propagation (FP) modules to gradually restore and map compressed features. Each FP module consists mainly of a distance interpolation layer and MLP layer. To recover detailed information during downsampling, the decoder integrates downsampling features from each stage during feature propagation. These features are fed into the FP modules and processed through MLP layers, producing more refined feature representations. This design ensures that high-level semantic information extracted during encoding is effectively restored and utilized during decoding, thereby improving the accuracy of final segmentation tasks.

The end-to-end semantic network categorizes bridge components and assigns relevant attributes, providing drones with an initial understanding of the components' structure. However, within a single category, the drone might not distinguish the nuances between multiple components. To address this, semantic segmentation is improved with an intra-class division strategy. This paper introduces a method that

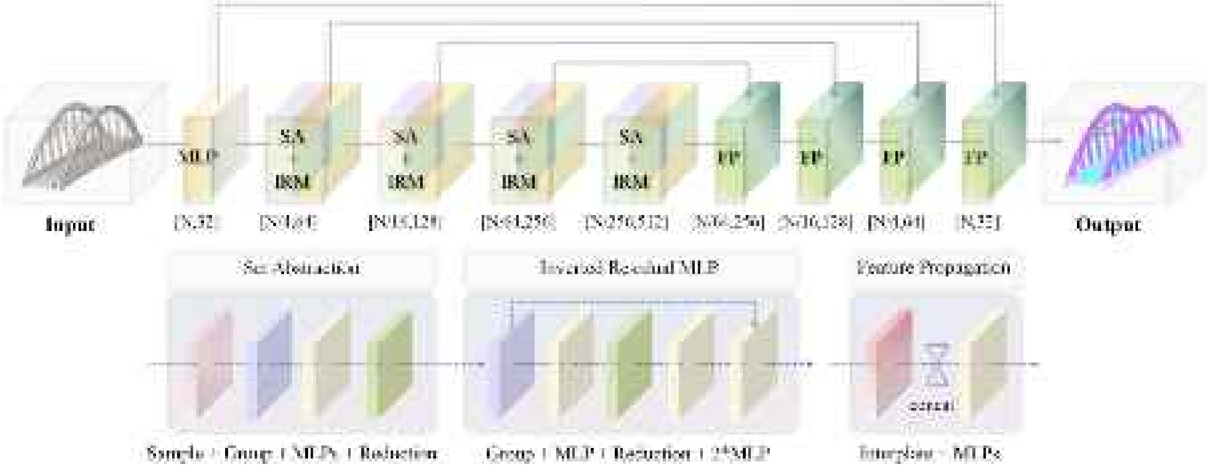


Fig. 2. Semantic learning of bridge components.

combines a density-based spatial clustering technique with the semantic segmentation network, achieving detailed instance-level segmentation while maintaining simplicity, as shown in Fig. 3.

Optimizing semantic segmentation data is essential. Segmentation outputs often contain noise points that can interfere with accurately interpreting components' morphological attributes. To mitigate this, a region aggregation process is introduced. This involves creating a three-dimensional grid in the point cloud space based on a defined pixel size S_p, with each pixel represented as a cube. Point cloud data is mapped to this grid, and the geometric centroid of each sub-point cloud space within each cube is chosen as the representative point. This method simplifies each three-dimensional pixel into a single representative point, resulting in a clearer morphology and reduced data volume. To address noise points, a statistical filter is applied as a supplementary step. This filter removes outlier points based on the statistical distance characteristics of each point in the point cloud data and its neighboring points. First, for each point, the nearest N_k points are selected as neighbors. Then, the distance between each point and its neighbors is calculated, and global statistics, including the mean and standard deviation, are collected. Using these statistics, a threshold T_k is set as the global average distance plus α times the standard deviation. If a point's average distance to its neighbors exceeds this threshold, it is classified as an outlier and removed from the point cloud.

To achieve rapid instance segmentation of bridge components, this paper uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. The main advantage of this method is that it does not require a pre-set number of clusters and can adapt to complex, nonlinear spatial features. Initially, the neighborhood search range ε for each point cloud element and the minimum number of points $\mathbb{m}$ required to form a dense region are determined. Based on ε, neighborhood searches are performed on each point cloud. If a point's neighborhood contains at least $\mathbb{m}$ points, it is defined as a core point P_{ci}. All points that meet the core point criteria are crucial for cluster formation. Using iterative methods, all core points within the neighborhood range are connected to form clusters. Once clustering is completed, the next unvisited point cloud is retrieved, and the process is repeated until all points in the model have been visited.

3.2. Dimensionality-reduced spatial path generation

UAV inspection planning focuses on two key factors: the camera's FOV for data acquisition and the drone's flight path. The flight path is influenced by both the spatial layout of bridge components and the FOV. The FOV, defined as the spatial area covered at a specific distance and focal length, depends on several factors, including sensor size, focal length, and acquisition distance. FOV limitation affects the inspection process in two key aspects. First, it influences data granularity: smaller FOVs offer higher resolution for detailed feature capture, while larger FOVs reduce spatial detail but improve efficiency. Second, it impacts detection coverage, requiring denser path planning to maintain full structural integrity under narrower FOVs. Thus, selecting the optimal FOV is crucial for various inspection tasks.

The camera's sensor dimensions H_s, W_s and focal length f determine the FOV width. The acquisition distance D directly impacts the FOV

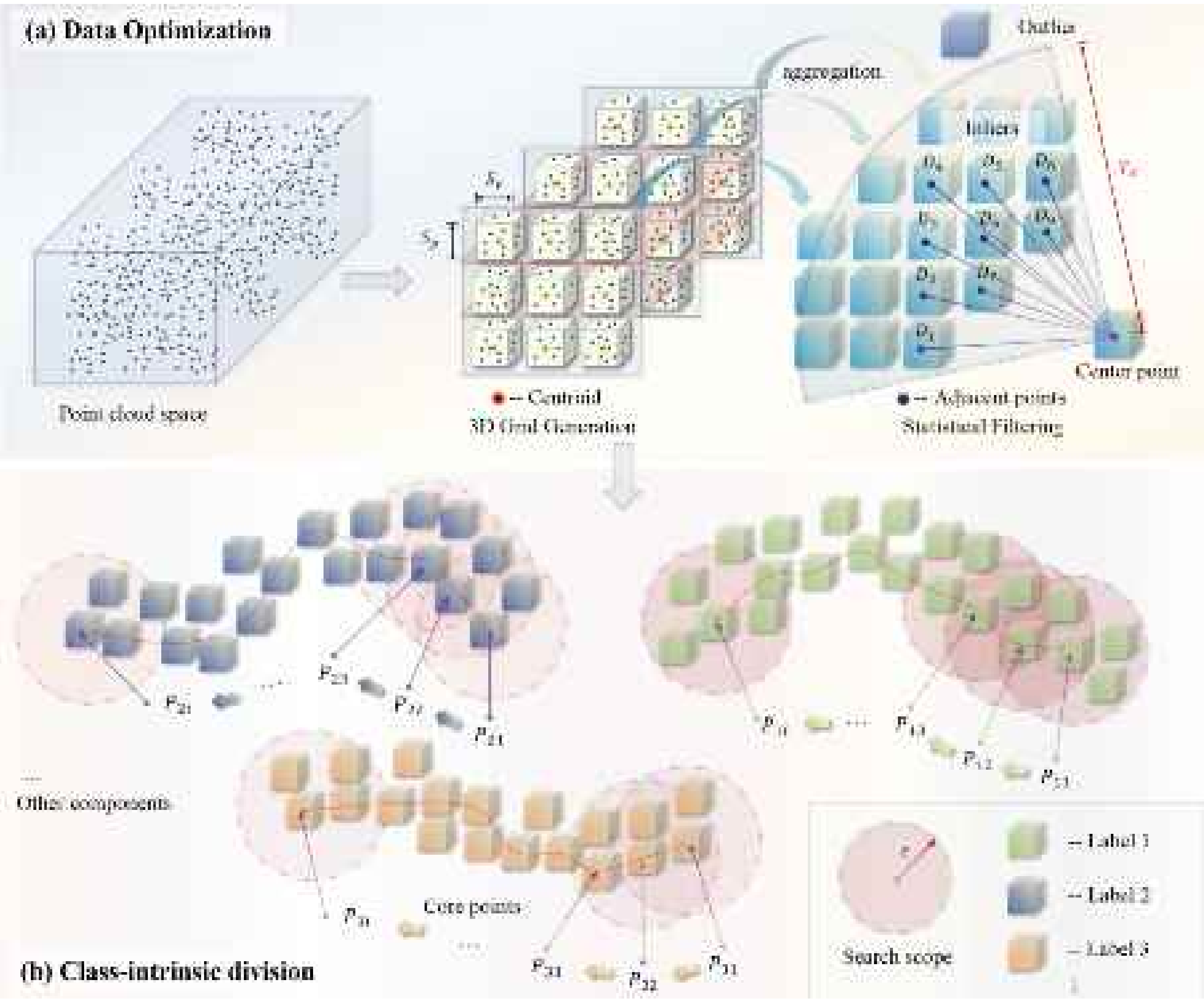


Fig. 3. Rapid instance segmentation of bridge components based on spatial distribution characteristics.

coverage area. With these parameters, the FOV in real space can be simulated (sensor coordinate system: $[X_s, Y_s, Z_s]$, image coordinate system: $[X', Y']$, world coordinate system:$[X_w, Y_w, Z_w]$), as shown in Fig. 4. The formula for calculating the FOV is as follows:

$$(H_{FOV}, W_{FOV}) = \left(\frac{H_s \times D}{f}, \frac{W_s \times D}{f}\right) \tag{1}$$

To ensure the inspection path covers the target area completely, images between adjacent collection points (O_s, O_{s1}, O_{s2}) must have sufficient overlap $W_o \times H_o$. Proper overlap provides a buffer for environmental changes or minor flight errors, ensuring the quality and completeness of the drone inspection. Therefore, setting the overlap rate $\delta = \frac{W_o \times H_o}{W_s \times H_s}$ between adjacent collection points is essential for efficient and comprehensive inspection.

FOV alters the trade-off between efficiency and detail, it does not affect the success rate of the detection task in our system, as inspection strategies based on 3D point cloud registration ensure comprehensive coverage. Most existing methods describe the relative positional relationships between the drone and structural components in three-dimensional space. This process is complex and challenging, primarily due to the need to account for the degrees of freedom of both the drone and the camera, among other factors. To address these challenges, this study uses dimensionality reduction to compress and transform the point cloud space, accurately depicting spatial relationships in planar geometry, as shown in Fig. 5. First, it is essential to identify the representative spatial planes of the components. This study uses the Random Sample Consensus (RANSAC) approach to depict the main geometric features of the components. The initial step involves randomly selecting three non-collinear points P_1, P_2, P_3 from the point cloud dataset to form an initial point set, serving as the basis for subsequent model fitting. The normal vector $\overrightarrow{n_p}$ of the initial point set $\overrightarrow{P_1P_2} \times \overrightarrow{P_2P_3}$ is calculated using mathematical methods, and an initial spatial plane reference model is constructed. Each remaining point in the component data is then traversed to perform distance measurement calculations. Points whose distance from the plane model is less than the tolerance threshold are identified as inliers and included in the Consensus Set. This iterative process is repeated, selecting the model with the largest consensus set as the representative spatial plane. The remaining point cloud (X, Y, Z) in the component space is projected onto the representative spatial plane (X_{pj}, Y_{pj}, Z_{pj}), effectively compressing the point cloud into the spatial plane. The projection formula is as follows:

$$\begin{pmatrix} X_{pj} \\ Y_{pj} \\ Z_{pj} \end{pmatrix} = \begin{pmatrix} \dfrac{(b^2+c^2)X - a(bY+cZ+d)}{a^2+b^2+c^2} \\ \dfrac{(a^2+c^2)Y - b(aX+cZ+d)}{a^2+b^2+c^2} \\ \dfrac{(a^2+b^2)Z - c(aX+bY+d)}{a^2+b^2+c^2} \end{pmatrix} \tag{2}$$

Here, a, b, c, d are the model coefficients of the representative spatial plane, which serves as the inspection target. By constructing an inspection plane parallel to this inspection plane using the distance coefficient D, the spatial considerations for the drone's flight path are simplified. To accurately represent component information on the plane, it is necessary to establish a coordinate system for the inspection plane. This involves constructing the x-axis based on the normal vector $\overrightarrow{n_p}$ and the z-axis of the space $\overrightarrow{n_z}$. Subsequently, by eliminating the y-axis perpendicular to the inspection plane, the coordinate system $[Z_{loc}(\overrightarrow{n_z}), X_{loc}(\overrightarrow{n_p} \times \overrightarrow{n_z})]$ for the inspection plane can be obtained. Projecting the component point cloud from the representative surface onto the inspection plane, and fixing the camera azimuth angle, ensures the acquisition plane remains parallel to the target plane, enhancing image fidelity. This approach unifies the key inspection planning factors on a single plane, reducing the complexity of the spatial dimensions and converting the three-dimensional space into a two-dimensional model. Using the regularity of bridge components, the centerline is established as the reference for the drone's inspection path on the two-dimensional plane. The point cloud characteristics remain as discrete points after dimensional reduction. Extracting the centerline from this unordered sequence is challenging, so a boundary fitting method is employed, using polynomial fitting in the direction with the larger aspect ratio. Incorporating the FOV model, collection points $(C_1, C_2, C_3 \cdots)$ are planned

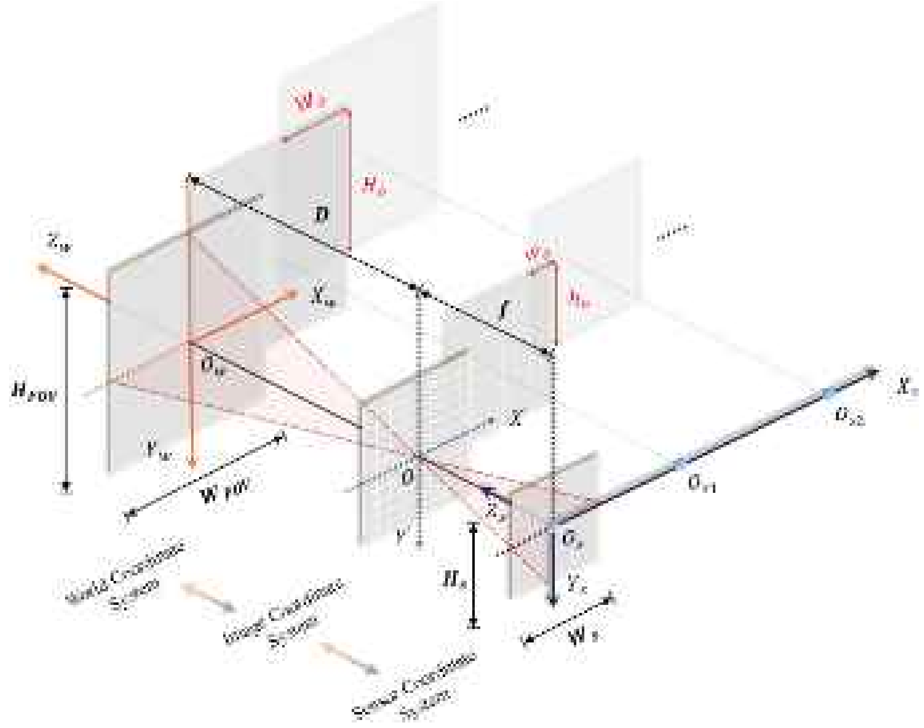


Fig. 4. Camera imaging geometry.

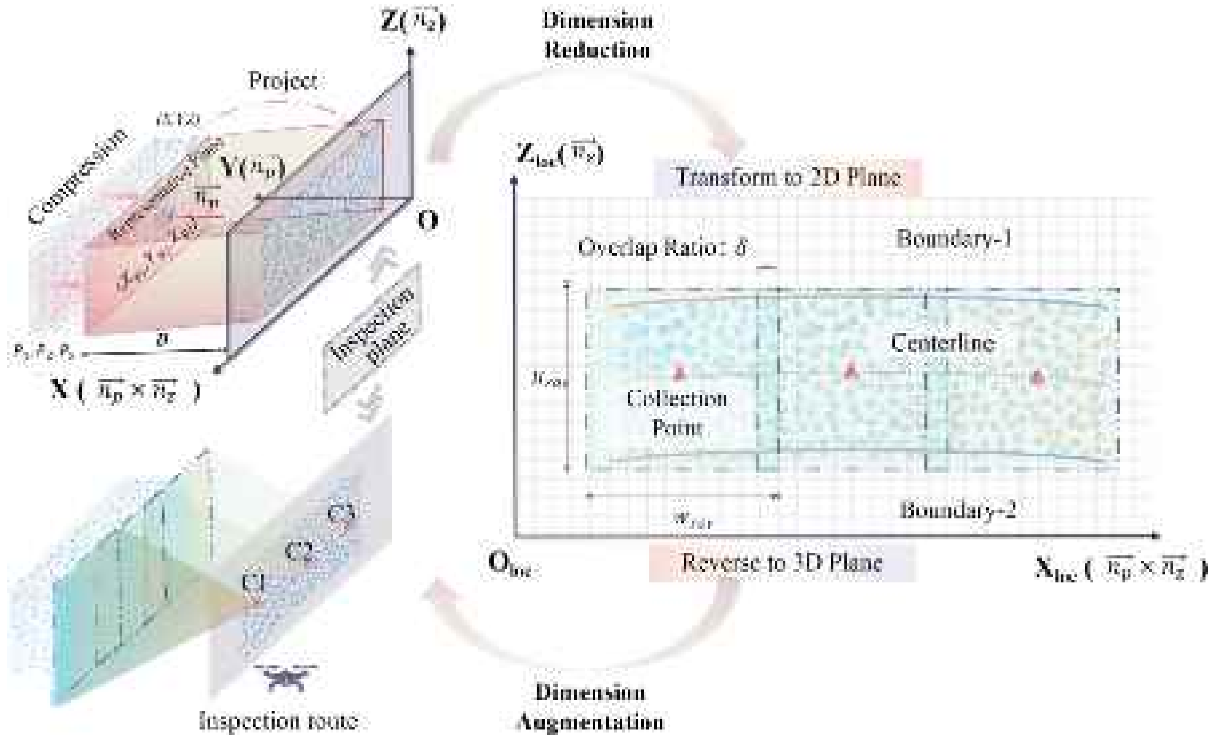


Fig. 5. Navigating complex spaces: path generation through dimensionality reduction.

along the component centerline. This planning considers various combinations of distance coefficient D and focal length f, determining an appropriate rough inspection FOV (W_p,H_p) to cover the inspection area adequately. An optimal FOV overlap rate δ is essential for a thorough inspection of the component surfaces. Initially, efficiency is prioritized with a large distance coefficient D and small focal length f combination, expanding the FOV. The overlap rate δ is then reduced to optimize the number of collection points. Finally, by reconstructing the eliminated spatial y-axis coordinates, the collection point coordinates on the two-dimensional plane are transformed back into three-dimensional coordinates, resulting in a comprehensive inspection path. Finally, by reconstructing the eliminated spatial y-axis ($\vec{n_p}$), the coordinates of the collection points on the two-dimensional plane are converted back to three-dimensional coordinates, yielding a comprehensive inspection path.

4. Lightweight hybrid model for high-efficiency defect analysis

4.1. Refining light-PVIT: an optimized structure for vision tasks

With the widespread application of mobile devices and Internet of Things (IoT) devices in daily life, there is an increasing demand for image-processing algorithms that work well in resource-constrained environments [45]. CNNs have been the standard for image processing due to their ability to capture features. However, their effectiveness is hindered by the limited understanding of the global context. Recently, Vision Transformers (ViT) [46] and Swin Transformers [47] have demonstrated superior performance in managing global information. Yet, their high computational demands make them unsuitable for resource-limited mobile devices. This study introduces Light-PVIT, a hybrid approach leveraging the efficiency of CNNs and the performance of Transformers, based on the Mobile-ViT architecture [48], for defect detection.

Light-PVIT employs a six-stage learning process with a multi-scale architecture to extract deep features, as illustrated in Fig. 6. The initial five stages utilize Convolutional Modules (CVB), Inverted Residual Block Depthwise Separable Convolutional Modules (RDB2, RDB1), and Partial Vision Transformer (PVIT) modules. While convolutional modules rapidly establish local feature connections, their repeated use significantly increases learning parameters and computational load. To enhance learning efficiency, depthwise separable convolution modules with inverted residual structures replace traditional convolutional layers. Incorporating residual structures addresses the vanishing gradient problem, accelerating network convergence [49]. In these modules, 1×1 mapping convolution initially reduces feature dimensionality, followed by 3×3 depthwise separable convolution for efficient feature extraction, and another 1×1 mapping convolution to increase dimensionality. Conversely, in the inverted residual structure, 1×1 mapping convolution first increases dimensionality, followed by a reduction using another 1×1 mapping convolution. This spindle-shaped feature dimension adjustment enhances feature extraction performance. Each learning stage begins with downsampling to explore deeper semantic features. Pooling, a common downsampling method, quickly reduces feature size but loses significant detail. Thus, an inverted depthwise separable convolution module without residual structure is used, ensuring efficient downsampling while retaining detailed feature information.

To address the limitations of pure convolutional structures [50], this paper introduces the PVIT module. PVIT efficiently captures and integrates both local and global information from feature maps without generating excessive parameters, as illustrated in Fig. 7. The PVIT module comprises three key sub-modules: local representation, global representation, and feature fusion. The local representation sub-module processes input $X \in \mathscr{R}^{H \times W \times C}$ using a 3×3 depthwise separable convolution followed by a 1×1 mapping convolution. This models local feature information, producing $X_f \in \mathscr{R}^{H \times W \times C_s}$ with an effective receptive field of 3×3. The 3×3 depthwise separable convolution encodes local spatial information, and the subsequent 1×1 mapping convolution adjusts feature dimensions by learning the linear combination of input feature channels. To capture global information with an effective receptive field of $H \times W$, the global representation sub-module employs the transformer's self-attention mechanism. The self-attention

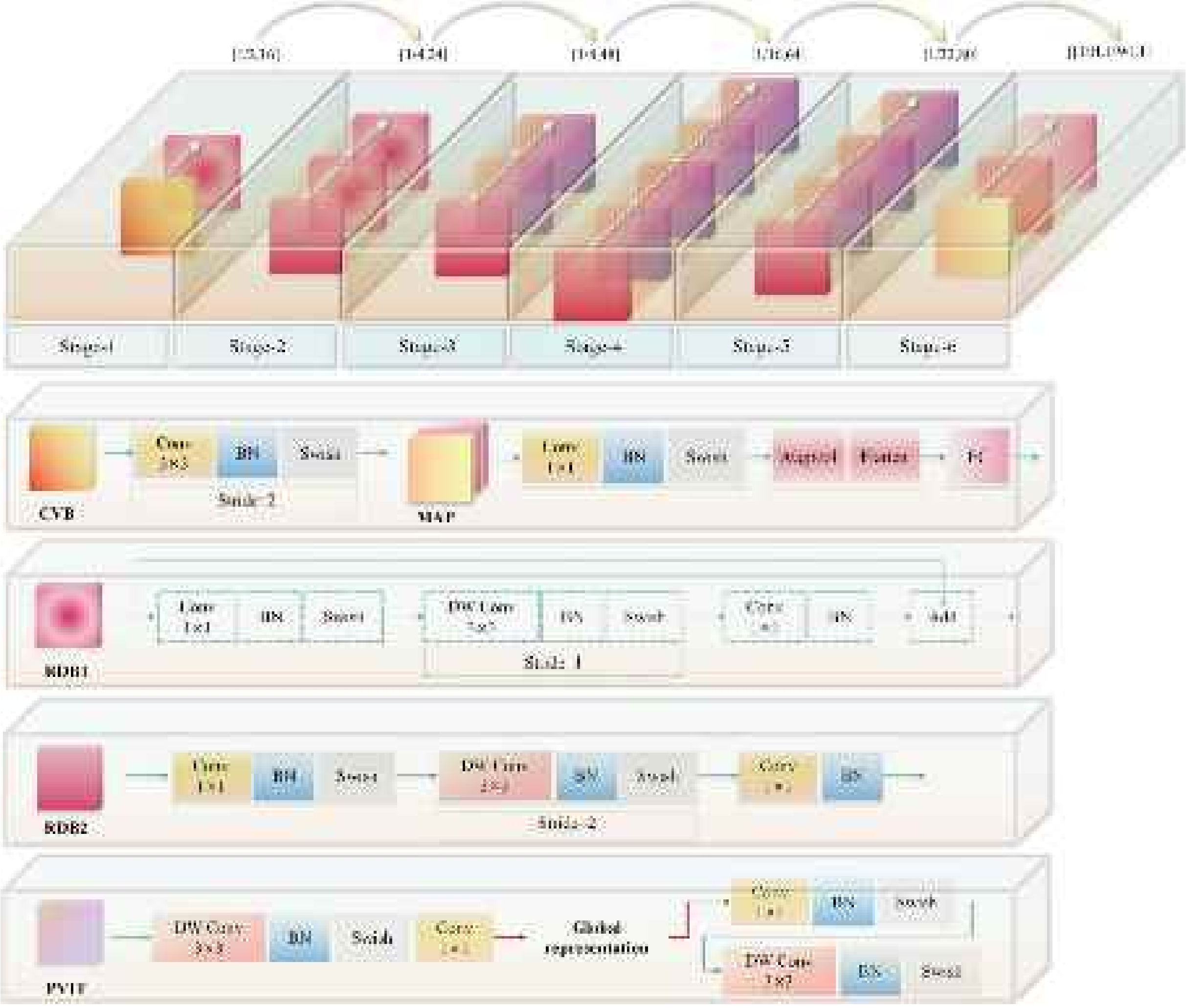


Fig. 6. Structure of proposed Light-PVIT network.

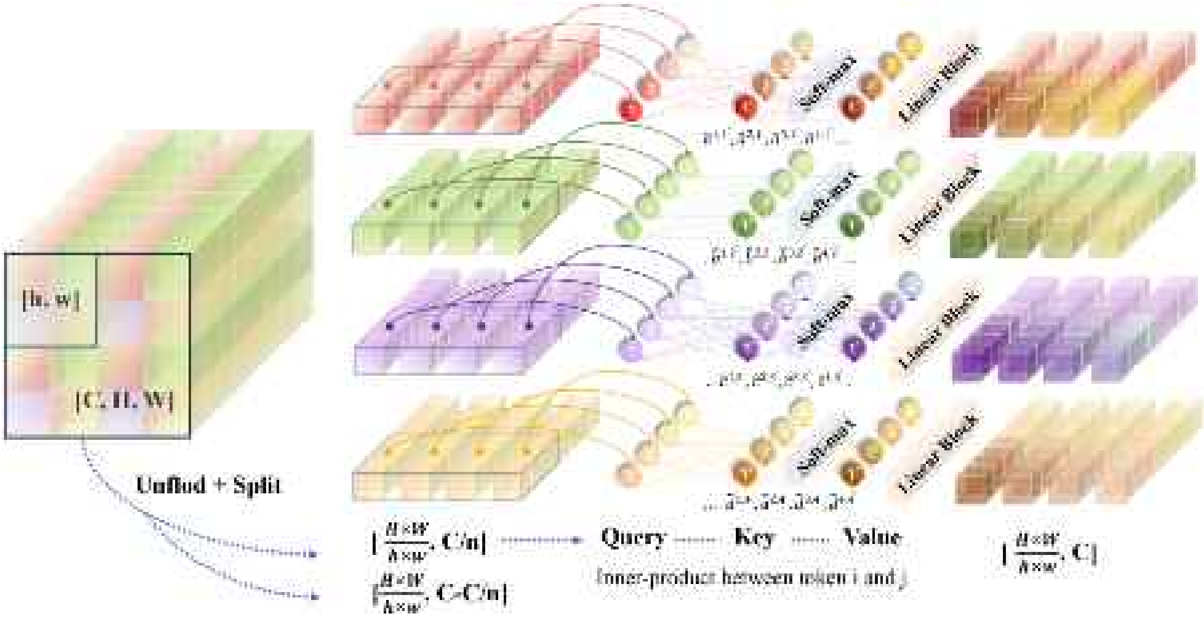

[h, w]
[C, H, W]
Unflod + Split
Softmax
Linear Block
Softmax
Linear Block
Softmax
Linear Block
Softmax
Linear Block
$[\frac{H\times W}{h\times w}, C/n]$
$[\frac{H\times W}{h\times w}, C\text{-}C/n]$
Query
Key
Value
Inner-product between token i and j
$[\frac{H\times W}{h\times w}, C]$

Fig. 7. Dual-dimensional spatial-channel optimization in PVIT module design.

mechanism is calculated as follows:

$$Att = \oplus\left(\langle\sigma\left(\langle \mathscr{x}M_Q^0, \mathscr{x}M_K^0\rangle\right), \mathscr{x}M_V^0\rangle, \cdots, \langle\sigma\left(\langle \mathscr{x}M_Q^h, \mathscr{x}M_K^h\rangle\right), \mathscr{x}M_V^h\rangle\right)M_o \tag{3}$$

Where $\oplus$ denotes the concatenation operation, $M_Q^i \in \mathscr{R}^{d\times d_h}, M_K^i \in \mathscr{R}^{d\times d_h}, M_V^i \in \mathscr{R}^{d\times d_h}$ represent the weight matrices of the Q, K, V branches in the i-th linear layer, respectively. σ denotes the softmax function, and $\langle \bullet, \bullet \rangle$ represents the dot product operation.

The self-attention mechanism introduces a large number of learning parameters for global connections, increasing model training complexity and resource consumption. To tackle these challenges, this paper proposes a reorganized fast self-attention mechanism that operates on both spatial and channel dimensions. In the global representation sub-module, local operations have already established pixel dependencies, connecting each pixel to its neighbors. If the self-attention mechanism is further applied, allowing each token vector to model globally with all tokens, it generates redundant information. While this approach captures global dependencies, the increased computational cost often outweighs the performance benefits. In convolutional structures, dilated convolutions reduce information redundancy by introducing spatial intervals between convolution kernel elements, preserving spatial information. Similarly, the concept of dilated convolutions can be applied to the transformer module, introducing dilation in the calculations between token vectors. Each token performs self-attention calculations only with tokens at corresponding dilated distances in the feature map, reducing computational load and redundant information. First, based on the pre-set window patch size $\mathscr{h}$ and $\mathscr{w}$, X_f is flattened to form $X_s \in \mathscr{R}^{N\times T\times C_s}$, where $N = \mathscr{h}\mathscr{w}$ and $T = \frac{HW}{N}$ represent the number of patches. This means that instead of treating a patch of size $\mathscr{h} \times \mathscr{w}$ as a single feature unit, it is divided into N independent tokens, each corresponding to a pixel sub-region within the patch. Typically, the patch used in the network is square $\mathscr{h} = \mathscr{w}$. Each token constructs global information only with neighboring tokens at a spatial distance of $\mathscr{h}$, enhancing self-attention computational efficiency.

Redundancy in features occurs not only spatially but also across channels. Different channels often carry similar or repetitive information, increasing the number of model parameters and computational complexity. These redundant channels offer limited improvement to model performance [51]. To address this, a strategy similar to that used in the spatial dimension is adopted by introducing partial splitting operations in the channel dimension. This optimizes computational resource usage while maintaining model performance, resulting in more efficient global information construction. Specifically, using a splitting factor n, the spatially compressed features X_s are divided into global modeling features $X_{st} \in \mathscr{R}^{N\times T\times \frac{C_s}{n}}$ and local modeling features $X_{sc} \in \mathscr{R}^{N\times T\times \frac{C_{s(n-1)}}{n}}$. To manage redundancy in high-resolution features and enhance computational efficiency in the early stages, the splitting factor $n_{stage-3}$ is set to 4. As the model progresses, the resolution of feature maps decreases, reducing feature redundancy. Therefore, lowering the splitting factor values to 2 in stages 4 and 5 prevents excessive compression and information loss. Next, X_{st} is fed into the transformer module, using the self-attention mechanism to capture long-distance dependencies in the input features, thereby enhancing the global representation ability. The local modeling features X_{sc} are merged with X_{st} in the channel direction during the output stage of the transformer module, restoring the feature shape and outputting $X_{gout} \in \mathscr{R}^{H\times W\times C_s}$. This combination of global modeling and local information retention extracts rich global features and integrates local details, achieving more precise feature representation. The fusion sub-module, for input $X_{gout} \in \mathscr{R}^{H\times W\times C_s}$, uses a 1 × 1 mapping convolution and a 3 × 3 depthwise separable convolution to fuse the global and local information of the features, outputting features $X_{fout} \in \mathscr{R}^{H\times W\times C}$. The primary function of the 1 × 1 mapping convolution is to recombine and map the feature channels while maintaining spatial dimensions, enabling channel-wise information interaction. The 3 × 3 depthwise separable convolution then fuses spatial information. This synergy integrates global and local information, producing a feature representation with rich global context and detailed local information.

The final stage of the model converts internally learned features into prediction results. It consists of three components: the convolution sub-module, the pooling flattening sub-module, and the fully connected sub-module. These sub-modules collaboratively extract useful information from high-dimensional features and map them to the prediction results.

4.2. Implementation details

Bridge defects typically appear as cracks and spalling. To automate the detection of these defects, a deep learning model was developed, defining three detection scenarios: normal, crack, and spalling. A significant amount of structural surface data was collected to train and evaluate the model. Data collection methods included web scraping and on-site photography using mobile phones and cameras, ensuring diversity and coverage. A total of 11,941 images were gathered, consisting of 4,012 crack images, 3,555 spalling images, and 4,374 normal images. The dataset was split into training and testing sets using an 8:2 ratio. The training set comprises 9,553 images for model learning and parameter optimization, while the test set contains 2,388 images to evaluate the model's generalization and performance. All images were resized to 448 × 448 pixels to ensure consistency in model input.

Selecting an appropriate loss function is crucial for training an efficient and accurate deep learning model. Cross-entropy loss was used to optimize the model's performance in multi-class recognition tasks. Cross-entropy loss measures the difference between the predicted probability distribution and the true label distribution. The AdamW optimizer was used for network training. Specifically, the values of three important parameters are initially set as: $weightdecay = 1\ \times 10^{-2}$ and learning rate $= 2 \times 10^{-2}$. The training cycle is set to 100 epochs, with a batch size of 64 for each iteration. The model was trained using the PyTorch framework on the Ubuntu 18.04 operating system. Environment requirements included Python 3.8.10, PyTorch 1.7.1, CUDA 11.0, and CUDNN 8.0.0. A Tesla V100 GPU with 32 GB memory was used.

To comprehensively evaluate the model's performance across all categories, the macro-average metric was used to calculate precision, recall, and F1-score [52]. The macro-average metric assigns equal weight to each category, ensuring fair consideration of the model's performance. The specific calculation formulas are as follows:

$$Macroindex\begin{cases} \mathscr{P}_m = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \\ \mathscr{R}_m = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \\ \mathscr{F}_m = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{2 + \frac{FP_i}{TP_i} + \frac{FN_i}{TP_i}}{2}\right)^{-1} \end{cases} \tag{4}$$

TP_i represents the number of instances correctly identified as category i, FP_i denotes the number of instances incorrectly identified as category i, and FN_i is the number of instances of category i that were not identified. $\mathscr{P}_m$ is calculated by determining the precision for each category and then taking the weighted average. This metric subjectively evaluates the effectiveness of positive predictions. Similar to $\mathscr{R}_m$, $\mathscr{P}_m$ is calculated by determining the recall for each category and then taking the weighted average. This metric objectively evaluates the effectiveness of positive predictions. $\mathscr{F}_m$ is a comprehensive metric that combines $\mathscr{P}_m$ and $\mathscr{R}_m$, providing a balanced evaluation of the model's performance across all categories from both subjective and objective perspectives.

To validate the effectiveness of the proposed method, three classic models were selected for performance comparison: pure convolutional, pure transformer, and hybrid structures. The experimental results are summarized in Table 1, detailing the performance of each model across various quantitative metrics. Compared to the lightweight convolutional models, Light-PVIT significantly improves recognition performance while reducing the number of parameters. Light-PVIT increases accuracy by at least 3 percentage points, achieving 96.23 %, compared to MobileNetV2 (92.69 %) and MobileNetV3 (92.91 %). While EfficientNetV2 and ConvNeXt are more complex models, they still fall short in accuracy compared to the proposed Light-PVIT. They have significantly more parameters and FLOPs—over 9000 % and 2000 % more, respectively—yet their accuracy remains lower by approximately 4.4 % and 1.6 % compared to Light-PVIT. Additionally, the proposed model was evaluated alongside the pure Transformer structure. Although the pure Transformer demonstrated strong performance in certain metrics, it required approximately 120 times more parameters and 37 times higher FLOPs. In contrast, Light-PVIT strikes an excellent balance between model efficiency and accuracy, making it more flexible and scalable for real-world applications. In comparison with advanced hybrid models, Light-PVIT achieves the best performance metrics while maintaining the lowest computational cost.

5. Information feedback-driven adaptive inspection strategy

The automated inspection method described in Section 3 quickly captures a series of overall appearance images of the bridge. The defect detection network from Section 4.1 performs high-precision screening on the data collected during the rough inspection stage. Images containing defects are separated to form a distinct defect set from normal region images, as shown in Fig. 8(a). While visual information is easily obtained, the exact spatial position of the UAV during defect collection and the precise location of defects within the bridge cannot be determined from these images alone. Due to this ambiguity, the analysis cannot determine the severity of defects or the overall status of the bridge, limiting its utility for providing actionable advice to engineers. During the rough inspection phase of the drone inspection task, key features are extracted from three-dimensional spatial data using the dimensionality reduction algorithm described in Section 3.2, resulting in a series of detection paths comprised of sampling points. Each sampling point is assigned a unique label $\mathscr{X}_i$, facilitating rapid indexing and spatial correspondence of subsequent images. During the inspection, the drone flies along the predetermined coarse inspection path, stopping at each sampling point to collect image data. Each captured image is associated with the corresponding sampling point label x_i, ensuring that the image data corresponds to its spatial location. By utilizing the label indexing method, each image can be quickly matched to its actual sampling position in three-dimensional space. According to the path planning process outlined in Section 3.2, the plane of the target component to be inspected is first identified, and a spatial detection plane parallel to the target component is constructed based on distance coefficients. Consequently, the sampling points within the spatial detection plane can be transformed into the target component plane, allowing for precise localization of defects.

Table 1
Performance comparison of different methods on Dataset.

Network	#Params (M)	FLOPs (G)	Acc	Pre	Re	F1
Mobilev2	2.23	1.31	92.69	92.67	92.55	92.46
Mobilev3	1.52	0.24	92.91	92.80	92.72	92.69
EfficientNetV2 [53]	21.46	11.58	91.78	92.05	91.86	91.64
ConvNeXt [54]	49.41	34.73	94.63	94.60	94.62	94.48
DeiT III [55]	21.59	16.93	95.89	95.78	95.87	95.77
Swin-transformer	27.49	17.48	96.42	96.30	96.29	96.28
EfficientFormer [56]	12.12	5.10	91.33	91.32	91.29	91.15
Edgenext [57]	5.27	3.82	93.96	94.00	93.91	93.78
MobileVIT	0.95	1.13	95.79	95.68	95.71	95.65
Light-PVIT	**0.23**	**0.47**	**96.23**	**96.12**	**96.05**	**96.07**

Notes: (1) #Param represents the number of model parameters, measured in M, where 1M $= 10^6$; FLOPs indicate the computational complexity of the model, measured in G, 1G $= 10^9$.

During the rough inspection stage, the UAV is distant from the inspection target, capturing images with a wide FOV that include rich spatial information, such as intact and defective bridge areas, rivers, trees, and houses. However, redundant information can interfere with defect positioning, and blurry local details in defect areas hinder detailed quantitative analysis in subsequent processing. To address this, an adaptive inspection method based on feedback from rough inspection information is proposed. This method enables efficient UAV inspection while maintaining a high-precision focus on defect areas, as shown in Fig. 8(b), (c). Initially, the UAV's collection points are classified into normal and abnormal categories based on feedback from the defect screening stage. At normal points, the UAV follows the standard rough inspection route. At abnormal points, the UAV adaptively adjusts its path based on defect feedback to obtain more accurate detailed information. Reducing the UAV's FOV is necessary to improve defect detection accuracy and focus on local defect details. The FOV is primarily determined by the distance coefficient D and focal length f. Reducing D or increasing f reduces the collection FOV. The method keeps the camera focal length constant and constructs a fine detection plane at a distance D_s from the target plane in the inspection space, allowing flexible adjustment of the UAV's fine collection FOV (W_s,H_s). To plan the UAV's fine detection route with a reduced FOV, the defect area is projected onto the newly constructed fine detection plane as the optimization target. Following the principle of dimensionality reduction analysis, the three-dimensional detection plane is converted to a two-dimensional plane for grid cutting of defect areas corresponding to individual collection points. Each grid size is $w = \frac{W_p}{S_w} < W_s, h = \frac{H_p}{S_h} < H_s$, where S_w, S_h represent the segmentation coefficients in width and height, respectively. The center coordinates of these sub-grids represent collection points on the fine inspection path. This operation is repeated for all defect areas to formulate a detailed collection plan. In the previous stage, to fully cover the bridge's exterior, there was an overlap between adjacent defect images, resulting in highly overlapping issues in the re-planned fine detection collection points. Therefore, an overlap area measurement criterion is used to automatically filter and delete collection points that exceed the threshold δ, reducing resource waste and improving inspection efficiency. The generated collection points are complex and discretely distributed on the plane, making it impossible to form an inspection path using a few simple straight lines or curves. A heuristic search algorithm is used to connect discrete collection points by minimizing the distance cost between adjacent points and constructing the fine detection path. Firstly, by calculating the distance matrix between all collection points, essential data is provided for path selection. The method begins at the first collection point, initializing a list of visited points and setting the path to include only the starting point. In the subsequent steps, the algorithm utilizes the distance matrix to identify the nearest unvisited point from the currently last visited location. This process enables the rapid identification of the closest target among the unvisited points, marking it as visited and progressively constructing a complete path that connects all collection points. This approach effectively reduces the movement distance at each step, significantly decreasing the overall path length and thereby enhancing the efficiency of the inspection task.

The inspection method based on the fine detection path can capture a series of detailed local image data. The collected data is screened for defects using the detection network. The actual collection location is rapidly positioned using label indexing. Finally, using the spatial transformation relationship between the collection point and the inspection target point, the collection point is inversely mapped to the

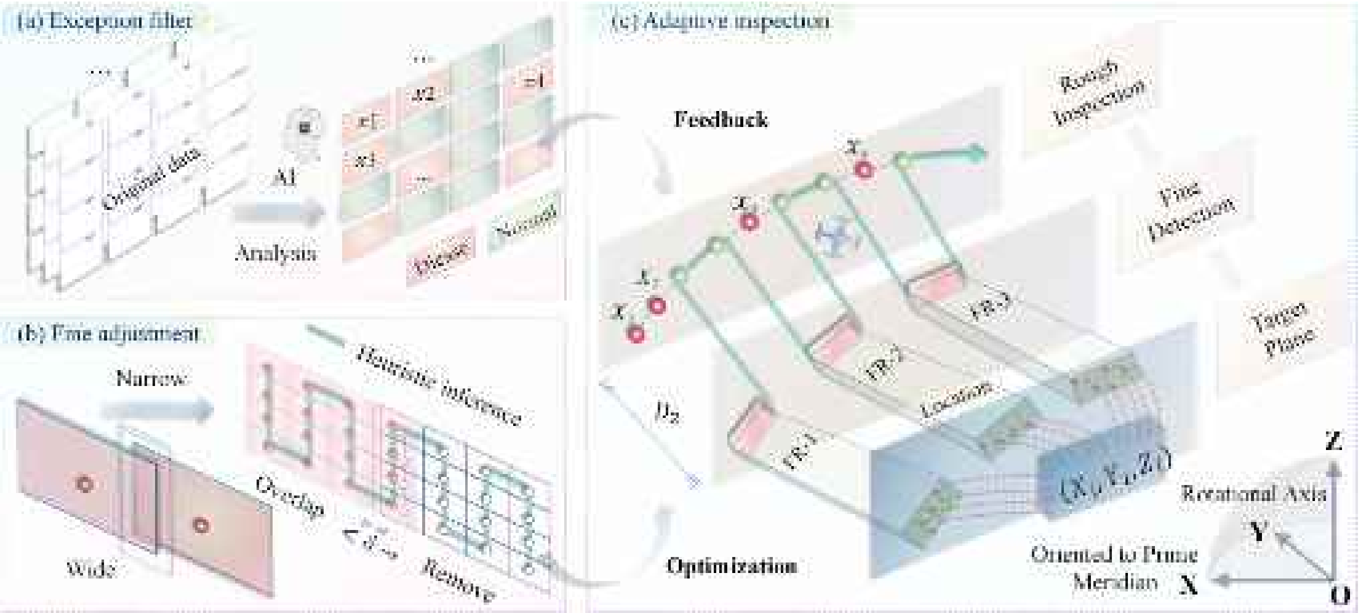


Fig. 8. Adaptive inspection strategy based on information feedback.

target representative surface using the distance coefficient *D*, achieving precise localization of the defect area in real space. Defect images focusing on local details are suitable for detailed quantitative analysis, but this aspect is beyond the scope of this paper and will be addressed in future research. The adaptive inspection framework equips UAVs with intelligent awareness, enabling them to optimize subsequent inspection.

6. Field validation experiments

To validate the proposed methodological framework, the Fangshan Bridge, with its complex structural morphology, was selected as the research subject, as shown in Fig. 9. This bridge is a prestressed concrete tied-arch structure with a deck beneath the arch. The main span is 96.76 m, and the deck width is 7 m. The main arch ribs are constructed of reinforced concrete, forming a parabolic curve. The calculated span is 95 m, and the rise is 14 m. During the UAV's adaptive inspection task, it operates autonomously, eliminating the need for direct human intervention. The deployment and operation of the method still require support from technical experts. Subsequent research work will involve close collaboration with engineering personnel to develop a system that is more accessible and user-friendly for field operators. For safety and convenience, the DJI M3T industrial UAV was selected as the equipment carrier due to its compact design. The UAV's camera has a 1/2 ″(6.4 mm× 4.8 mm) and an equivalent focal length of 24 mm, with a calculated actual focal length of 4.47 mm. Using the FOV calculation formula in Section 3.2, when the UAV is 14 m from the bridge inspection surface and uses 5 times the original focal length for data collection, the actual FOV is 4m × 3 m (with an accuracy of 1 mm/pixel). This setup is used as

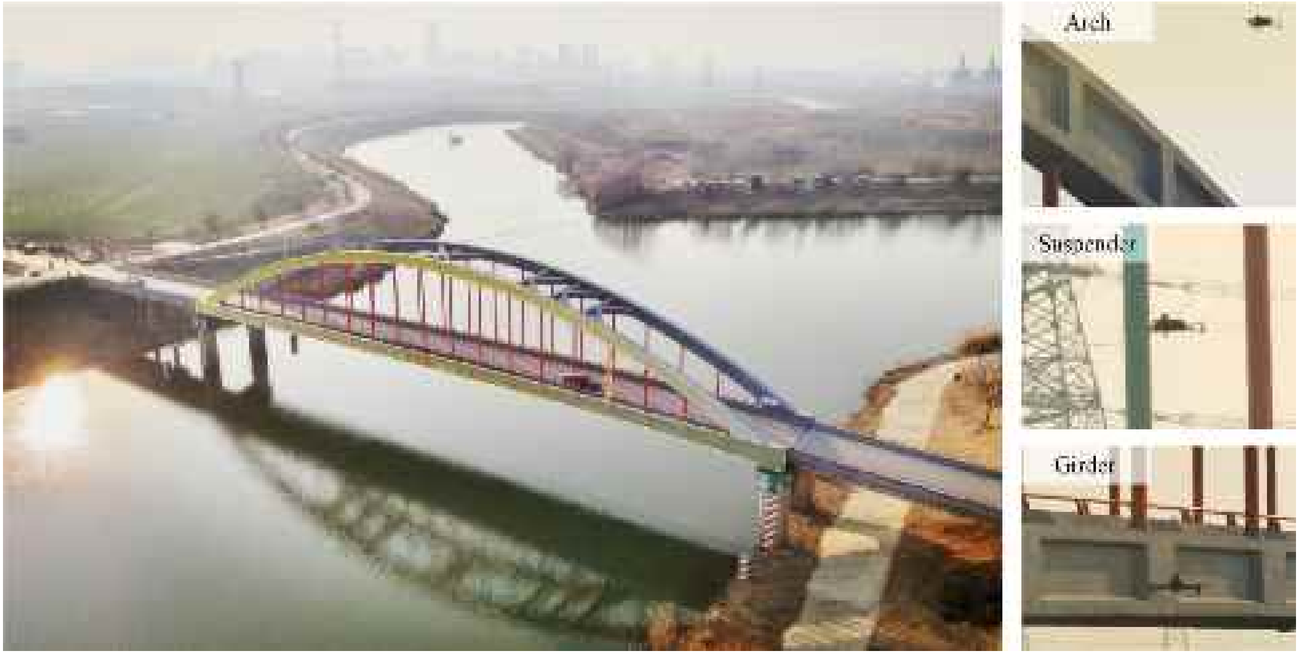


Fig. 9. Schematic diagram of bridge overview.

the flight criterion for the rough inspection stage, i.e., $D_p = 14\text{m}, f_p = 22.35\text{mm}, W_p \times H_p = 4009mm \times 3007mm \approx 4\,\text{m} \times 3\,\text{m}$. As shown on the left side of Fig. 9, this stage focuses on capturing the overall information of the bridge. When the UAV is 5 m from the target, using the same focal length multiple, the actual FOV is 1.4m × 1 m (with an accuracy of 0.35 mm/pixel). This setup is used as the flight criterion for the fine detection stage, i.e., $D_s = 5\text{m}, f_s = 22.35\text{mm}, W_s \times H_s = 1432mm \times 1074mm \approx 1.4\,\text{m} \times 1\,\text{m}$. The right side of Fig. 9 shows the UAV focusing on capturing local information of the bridge during the fine detection stage.

6.1. Implementation details of the rough inspection phase

A preliminary 3D reference model is essential for the UAV to gain a complete spatial understanding of the bridge. The UAV captured 110 high-resolution images (5235 × 3922) of the experimental bridge from various angles to create this model. The SfM algorithm was executed on a ground station computer to perform multi-view geometric 3D reconstruction, extracting and matching features to establish geometric relationships between the captured images. Triangulation of these points generated the 3D reference model. It accurately represents the bridge's structure and spatial layout, offering the UAV a holistic understanding.

The UAV inspection plan, relying solely on this model, was inefficient as it did not establish a targeted spatial context. To overcome this challenge, this study focuses on constructing a cognitive foundation for bridge component attributes using the PointNeXt network for initial exploration. On-site data was collected using optical cameras and LiDAR, then augmented with techniques like non-uniform scaling, sampling, and noise addition, resulting in 10 datasets of bridge models. The datasets were labeled with five segmentation types: roadways, cross-braces, suspender rods, arches, and beams. During training, the AdamW optimizer was used to update model parameters to minimize the cross-entropy loss function. Specifically, an initial learning rate $= 1 \times 10^{-3}$ was set, along with a weight decay parameter of $weightdecay = 1 \times 10^{-4}$. A cosine annealing scheduler dynamically adjusted the optimizer's parameters over 100 training epochs, with a minimum learning rate $= 1 \times 10^{-5}$. The batch size was 8. The PointNeXt network was trained using the PyTorch framework on a Windows 10 system, with Python 3.8.0, PyTorch 1.8.1, CUDA 11.1, and CUDNN 8.0.4. The GPU used was an Nvidia GeForce RTX 3090 with 24 GB of memory. The model achieved excellent performance on the test set, with an average test accuracy of 96.87 % and an mIOU of 94.98 %. Although PointNeXt classified different component types, it struggled with distinguishing multiple instances of the same type, like suspenders, arches,

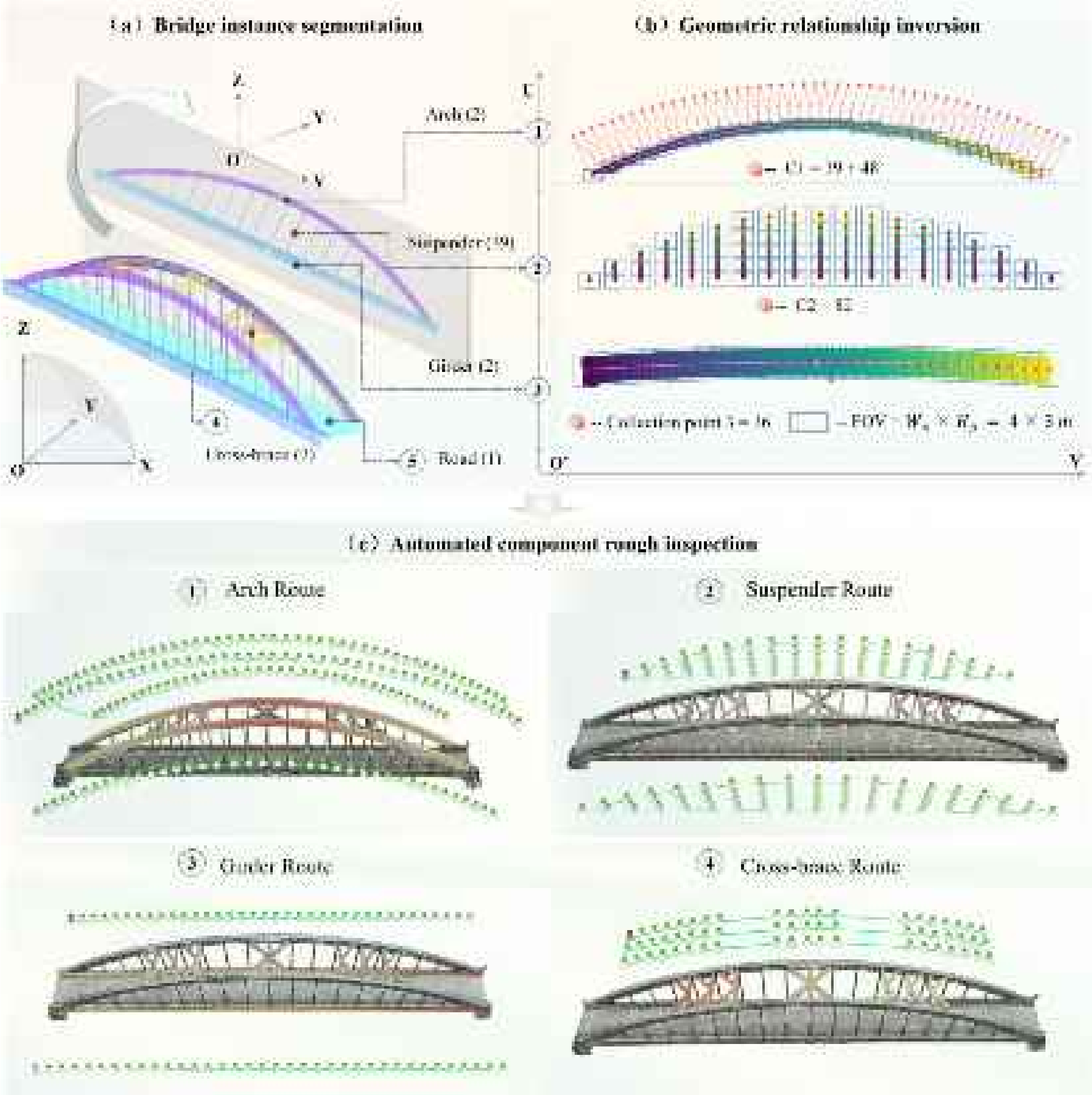


Fig. 10. Rough inspection stage results overview: (a) Bridge instance segmentation; (b) Geometric relationship inversion; (c) Automated component rough inspection.

and beams. Therefore, an efficient spatial clustering post-processing method, detailed in Section 3.1, was introduced to further subdivide the semantic segmentation results within categories, as demonstrated in Fig. 10(a). The instance-level understanding of bridge components enables engineers to quickly assign unique information codes to each individual unit within every component category, facilitating subsequent data management and information tracking.

Compared to a blind overall spatial inspection strategy, detailed component attribute information allows the UAV to target specific areas, enabling comprehensive and rapid defect detection during the rough inspection phase. Guiding a UAV in three-dimensional space is complex. To manage this, the dimensionality reduction optimization approach (Section 3.2) is employed to simplify the spatial inspection task. This study focuses on inspecting cross-braces, arch structures, suspenders, and beams, excluding roadways from the inspection targets. Vehicle-mounted cameras can achieve inspection results more efficiently and accurately compared to UAVs. The crucial step in dimensionality reduction optimization is identifying representative spatial planes. Thus, spatial planes for arches, suspenders, and beams are extracted, as illustrated in Fig. 10(a). Since the spatial planes of these three structures are parallel, only one representative plane is shown in the figure. Inspection planes for each component were then constructed using the distance coefficient D_p from the rough inspection phase, translating three-dimensional spatial relationships into two-dimensional plane geometry. The two-dimensional representations of components appear as regular straight lines or curves at the boundaries. Customized inspection planning for each component is carried out using detailed attribute information. While route planning is not the primary focus, it ensures comprehensive coverage of the inspection targets. Polynomial fitting is used to delineate the boundaries of these structures. The centerline of the geometric image is drawn based on the boundaries, and path planning is executed with the FOV size of $W_p \times H_p$ and an overlap rate of $\delta = 30\%$, as shown in Fig. 10 (b). Unlike beams and suspenders, arch structures and cross-braces require inspection tasks on the top surface of the bridge, which is a spatial curved surface. To simplify this task, inspections on the top surface are planned indirectly using the side inspection plane. An initial upper-side inspection baseline path is established. Path mapping covers the upper side plane of the arch, forming a complete inspection route for the upper side of the arch. For cross-braces, the inspection task uses the upper-side arch route as a base, focusing on detecting cross-brace structures and removing other points outside this area. Finally, the two-dimensional coordinates are reverse-mapped to three-dimensional coordinates to obtain the complete inspection path, as shown in Fig. 10(c). This strategy ensures that all significant regions are covered without omission, ensuring that no critical regions with potential severe issues are overlooked during the inspection process.

6.2. Verification results of the fine detection phase

In the automatic rough inspection stage, the UAV's component-level expertise facilitates efficient and thorough inspection. Concurrently, the collected data is processed through the Light-PVIT model introduced in Section 4.1, as shown in Fig. 11. For effective bridge operation advice, the real spatial locations of defect points are quickly inferred from the association between defect images and actual space, as shown in the rough inspection planes of Fig. 12(b)(c), Fig. 13(a)(d), and Fig. 14(a). Due to the arch structure's complexity, only one side is validated, whereas cross-braces and beam structures are comprehensively displayed. In the figures, orange points indicate defects, while white points

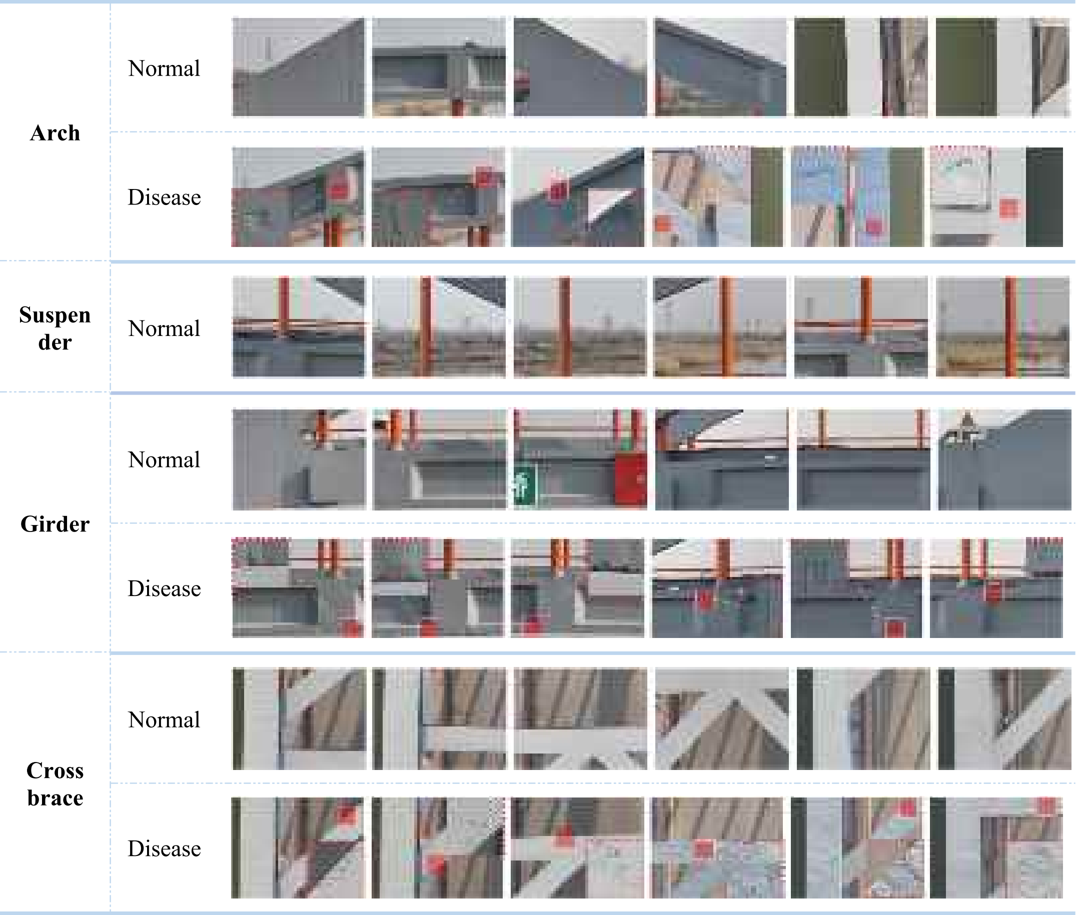


Fig. 11. Status of defect in the rough inspection process.

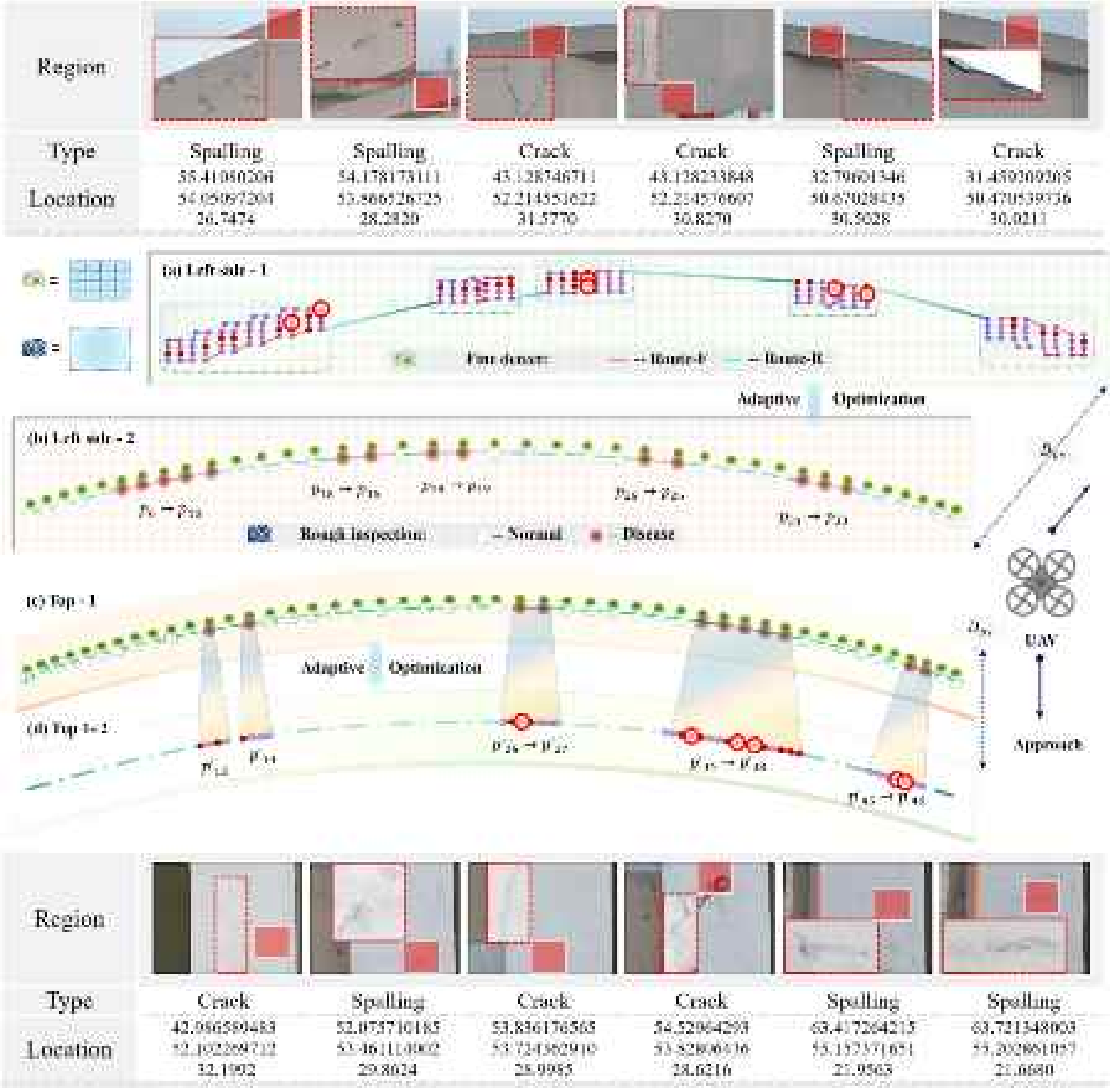


Fig. 12. Adaptive inspection flowchart for arch components.

indicate undamaged areas. There are 19 defect points on one side of the arch (including the opposite side) and 20 on the top. The cross-braces have 33 defect areas. The beam structure has 23 detected defect areas. The detected results are consistent with those identified through manual inspection by experienced personnel, confirming the reliability of the proposed strategy. Most defects are concentrated on the arch and cross-brace structures, likely due to their greater exposure to environmental factors. In contrast, defects in the suspenders are less significant. These findings are essential for planning maintenance and repairs.

While rough inspection improves detection efficiency, it often compromises accuracy. Accurate identification in defect areas necessitates a refined strategy. This paper introduces an adaptive inspection method using rough inspection feedback. This approach ensures efficient UAV inspection with high precision on local defect information, as illustrated in Fig. 12(a)(d), Fig. 13(b)(c), and Fig. 14(b). The UAV employs the rough inspection method in undamaged areas but adaptively adjusts its path in defect areas based on feedback from the preliminary inspection to gather more detailed information. To enhance defect detection accuracy, a fine detection plane is constructed at a distance D_s from the target plane in the inspection space.

The distance between the rough and fine detection planes is $D_{ps} = D_p - D_s$. The UAV's collection FOV is adjusted from $W_p \times H_p$ to $W_s \times H_s$. The path for defect areas is mapped to the new fine detection plane as a local optimization target. The dimensionality reduction method from Section 3.2 is used to transform the spatial plane, simplifying the spatial path optimization. Each defect collection point corresponds to an area range of $W_p \times H_p$. To enhance local defect details and account for background factors, the pixel scale representing defects is increased. All defect areas are grid-divided, with each grid size being $w = 1mm$ and $h = 0.75mm$. The center points of these sub-grids are calculated as fine collection points. This ensures that each fine collection point's FOV maintains a specific overlap rate. The preliminary overlap setting can cause high overlap when inspecting adjacent defect areas. Points with an overlap rate above 30 % are filtered out and deleted to address this. Ultimately, 321 fine detection points focus on the arch structure: 252 on the side and 69 on the top. After optimization, there are 302 defect detection points for the cross-brace structure. In the beam structure, the adaptively adjusted defect points are distributed as follows: 120 on the right side and 196 on the left side.

Guided by adaptive optimization methods, detailed local image data

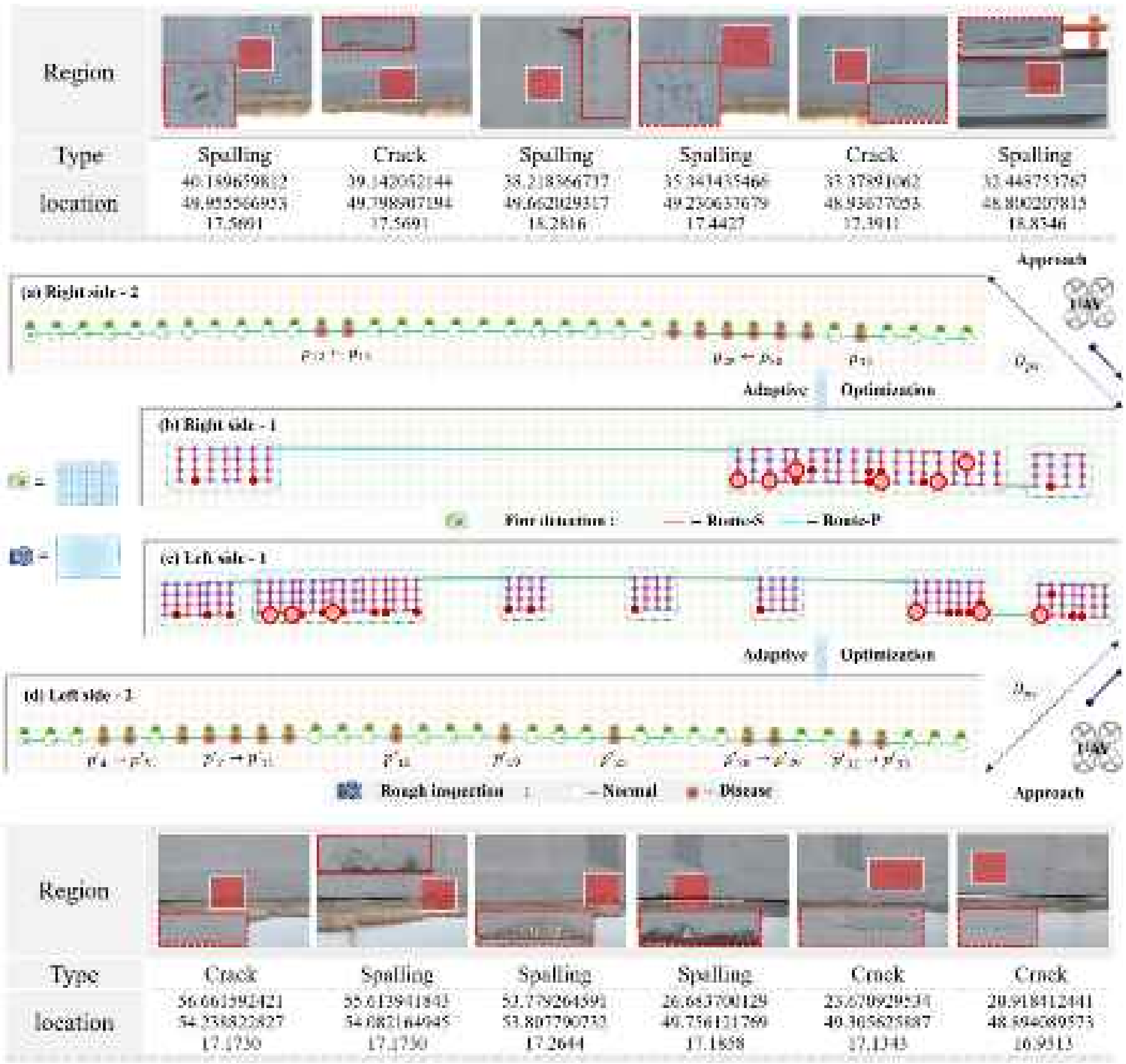
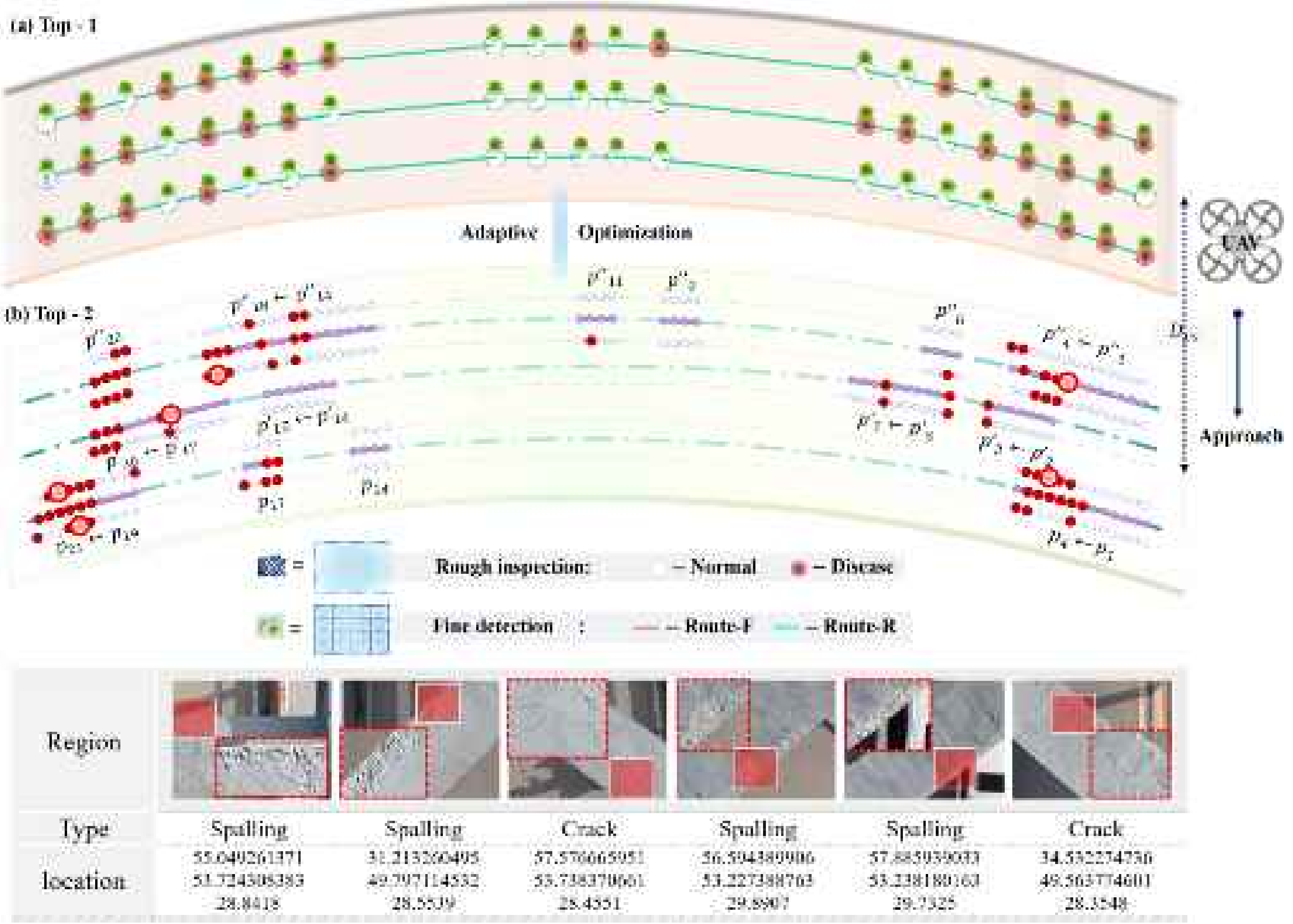


Fig. 13. Flowchart of adaptive beam inspection process.

were captured through fine paths. The Light-PVIT model then identified defects in these data. Additionally, the precise locations of the defect areas were determined. In the detailed analysis of the arch structure, the conditions of key parts were accurately recorded. Among the identified defect regions, 39 areas on the left side were classified as fine defect areas, requiring special attention. Out of 74 collection points on the right side, 30 were identified as defect areas, indicating a significant issue in these areas. The top region had 36 defect areas. In the beam structure, 27 defect areas were identified on the left side and 16 on the right side. The cross-brace structure had 85 fine defect detection points. A comparative evaluation with manual close-range inspections confirmed that the defect zones identified by human inspectors were also detected by our system. Detailed descriptions of local defect areas in the images provide a solid basis for quantitative analysis. While this was not the current research focus, it is planned for future study. Images are crucial for visual localization and pixel-based quantitative information but have limitations in offering comprehensive guidance. To address this, label indexing was used to extend image information into three-dimensional space. By combining the spatial transformation relationship between collection points and inspection target points, defect areas were localized in real space, as shown in Fig. 12(a)(d), Fig. 13(b)(c), and Fig. 14 (b). The first row of the location table represents longitude, scaled as $X \times 10^{-5}$, where the actual longitude is $118.854 + X \times 10^{-5}$. The second row represents latitude, with a scale of $X \times 10^{-4}$, where the actual latitude is $31.87 + X \times 10^{-4}$. The third row represents elevation. The adaptive inspection framework offers intelligent decision support for engineering personnel. The adaptive detection framework can automatically analyze and output the type, quantity, and location of defects for each individual component. During the data analysis phase, it alleviates the workload of engineers while establishing a relationship between defect information and each bridge component. This capability not only enhances detection efficiency but also assists technicians in generating detailed inspection reports, providing crucial support for subsequent bridge maintenance and management.

7. Conclusion

This study addresses the challenge of balancing efficiency and accuracy in bridge inspections by proposing an innovative adaptive UAV inspection strategy. Field tests conducted on Fangshan Bridge validated the proposed framework, providing comprehensive data on damaged areas to engineering personnel and offering detailed support for maintenance planning. This approach significantly reduces manual intervention, lowers inspection costs, and enhances safety during inspection



Type	Spalling	Spalling	Crack	Spalling	Spalling	Crack
location	55.049261371 53.724308383 28.8418	31.213260495 49.797114532 28.5539	57.576665951 53.738370661 28.4551	56.594389906 53.227388763 29.8907	57.885939033 53.238180163 29.7325	34.532274736 49.563774601 28.3548

Fig. 14. Schematic diagram of adaptive cross-brace inspection.

procedures. Additionally, the integration of intelligent UAV patrols accelerates the adoption of UAV inspections into standard operational workflows, paving the way for a more systematic and scientifically grounded approach to bridge maintenance. The main contributions of this study are as follows:

(1) Efficient Spatial Feature Processing through Advanced Segmentation: By combining a lightweight point cloud semantic segmentation network with a density-based spatial clustering method, this study effectively handles complex spatial features, achieving precise segmentation of bridge components.
(2) Novel Path Planning Framework with Simulated FOV Modeling: A simulated FOV model was developed to accurately represent the spatial relationship between UAVs and bridge structures, taking into account UAV dynamics and environmental factors. Dimensionality reduction techniques were applied to transform 3D data into 2D representations, thereby simplifying path planning while preserving spatial accuracy.
(3) High-Efficiency Defect Detection via Light-PVIT: An efficient detection tool based on Light-PVIT was introduced to facilitate defect information learning. This tool captures and integrates local image details and global semantic information, with a bidirectional splitting strategy across spatial and channel dimensions to optimize computational efficiency.
(4) Multi-Level Adaptive Inspection Strategy: By integrating multi-level learning from holistic, component, and localized perspectives, this study introduces a robust adaptive bridge inspection strategy. The defect detection tool classifies collection points from the initial rough inspection stage as either normal or abnormal. For abnormal points, a grid refinement method adaptively adjusts the inspection path to capture finer details, ensuring thorough defect assessment.

The proposed adaptive inspection framework enhances UAV inspection tasks with higher intelligence, optimizing subsequent inspection strategies based on detected defects. Although local detail images of bridges are not fully analyzed, their significance for bridge health assessment is acknowledged. Future research will focus on developing and optimizing methods to accurately obtain quantitative parameters of bridge structures. Accurate measurement and precise location of disease information will be integrated into the overall bridge assessment, establishing a comprehensive evaluation system. The current research lacks sufficient validation of the system's long-term stability and usability for technical personnel. Future work will prioritize comprehensive testing, incorporating practical trials with engineering professionals to improve the system's stability and accessibility for field operators. Additionally, the feasibility of applying the proposed method in highway facilities warrants further exploration to verify its applicability and scalability across different types of infrastructure.

CRediT authorship contribution statement

Wang Chen: Writing – original draft, Methodology, Formal analysis. **Xin Zhang:** Methodology, Conceptualization. **Binhong Yuan:** Investigation. **Jian Zhang:** Methodology, Conceptualization.

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.

Acknowledgements

The research presented was financially supported by the National Key Research and Development Program of China (No.: 2022YFC3801700), the National Natural Science Foundation of China (No. 52378289), Research Fund for Advanced Ocean Institute of Southeast University (No. KP202407).

Data availability

Data will be made available on request.

References

[1] H. Chu, W. Chen, L. Deng, Cascade operation-enhanced high-resolution representation learning for meticulous segmentation of bridge cracks, Adv. Eng. Inf. 61 (2024) 102508, https://doi.org/10.1016/j.aei.2024.102508.

[2] W. Chen, Z. He, J. Zhang, Online monitoring of crack dynamic development using attention-based deep networks, Autom. Constr. 154 (2023) 105022, https://doi.org/10.1016/j.autcon.2023.105022.

[3] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering. 5 (2) (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.

[4] Y. Wu, Y. Wang, D. Li, J. Zhang, Two-step detection of concrete internal condition using array ultrasound and deep learning, NDT and E Int. 139 (2023) 102945, https://doi.org/10.1016/j.ndteint.2023.102945.

[5] B. Sutter, A. Lelevé, M.T. Pham, O. Gouin, N. Jupille, M. Kuhn, et al., A semi-autonomous mobile robot for bridge inspection, Autom. Constr. 91 (2018) 111–119, https://doi.org/10.1016/j.autcon.2018.02.013.

[6] Y. Choi, Y. Choi, J.-S. Cho, D. Kim, J. Kong, Utilization and verification of imaging technology in smart bridge inspection system: an application study, Sustainability 15 (2) (2023) 1509, https://doi.org/10.3390/su15021509.

[7] A. Leibbrandt, G. Caprari, U. Angst, R.Y. Siegwart, R.J. Flatt, B. Elsener, Climbing robot for corrosion monitoring of reinforced concrete structures, in: 2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI), IEEE, Zurich, Switzerland, 2012, pp. 10–15, https://doi.org/10.1109/CARPI.2012.6473365.

[8] F. Xu, S. Dai, Q. Jiang, X. Wang, Developing a climbing robot for repairing cables of cable-stayed bridges, Autom. Constr. 129 (2021) 103807, https://doi.org/10.1016/j.autcon.2021.103807.

[9] S.T. Nguyen, H.M. La, A climbing robot for steel bridge inspection, J. Intel. Robot. Syst. 102 (4) (2021) 75, https://doi.org/10.1007/s10846-020-01266-1.

[10] Y. Tian, C. Zhang, J. Shang, J. Zhang, W. Duan, Noncontact cable force estimation with unmanned aerial vehicle and computer vision, Comput.-Aid. Civ. Infrastruct. Eng. 36 (2021) 73–88, https://doi.org/10.1111/mice.12567.

[11] S. Jiang, Y. Cheng, J. Zhang, Vision-guided unmanned aerial system for rapid multiple-type damage detection and localization, Struct. Health Monitor.. 22 (1) (2023) 319–337, https://doi.org/10.1177/14759217221084878.

[12] S. Jiang, J. Zhang, Real-time crack assessment using deep neural networks with wall climbing unmanned aerial system, Comput.-Aid. Civ. Infrastruct. Eng.. 35 (12) (2020) 549–564, https://doi.org/10.1111/mice.12519.

[13] K. Jang, Y.-K. An, B. Kim, S. Cho, Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot, Comput. Aided Civ. Inf. Eng. 36 (2021) 14–29, https://doi.org/10.1111/mice.12550.

[14] C. Zhang, Y. Zou, F. Wang, E. del Rey Castillo, J. Dimyadi, L. Chen, Towards fully automated unmanned aerial vehicle-enabled bridge inspection: where are we at? Construct. Build. Mater. 347 (2022) https://doi.org/10.1016/j.conbuildmat.2022.128543.

[15] G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, V. Rodehorst, Framework for automated UAS-based structural condition assessment of bridges, Autom. Constr. 97 (2019) 77–95, https://doi.org/10.1016/j.autcon.2018.10.006.

[16] J.J. Lin, A. Ibrahim, S. Sarwade, M. Golparvar-Fard, Bridge inspection with aerial robots: automating the entire pipeline of visual data capture, 3D Mapping, defect detection, analysis, and reporting, J. Comput. Civ. Eng. 35 (2) (2021), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000954.

[17] F. Wang, Y. Zou, E. del Rey Castillo, Y. Ding, Z. Xu, H. Zhao, J.B.P. Lim, Automated UAV path-planning for high-quality photogrammetric 3D bridge reconstruction, Struct. Infrastruct. Eng. 20 (10) (2022) 1595–1614, https://doi.org/10.1080/15732479.2022.2152840.

[18] R. Li, J. Yu, F. Li, R. Yang, Y. Wang, Z. Peng, Automatic bridge crack detection using unmanned aerial vehicle and faster R-CNN, Construct. Build. Mater.. 362 (2023) 129659, https://doi.org/10.1016/j.conbuildmat.2022.129659.

[19] L. M. Horton, K. M. Cabral, B. W. Surgenor, S. N. Givigi, J. E. Woods, A Framework for Autonomous Inspection of Bridge Infrastructure using Uncrewed Aerial Vehicles, In: 2024 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 2024, pp. 1-8, Doi: 10.1109/SysCon61195.2024.10553518.

[20] A.Y. Yiğit, M. Uysal, Virtual reality visualisation of automatic crack detection for bridge inspection from 3D digital twin generated by UAV photogrammetry, Measurement. 242 (2025) 115931, https://doi.org/10.1016/j.measurement.2024.115931.

[21] K.W. Tse, R. Pi, W. Yang, X. Yu, C.Y. Wen, Advancing UAV-based inspection system: the USSA-net segmentation approach to crack quantification, IEEE Trans. Instrum. Meas. 73 (2024) 1–14, https://doi.org/10.1109/TIM.2024.3418073.

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

[23] Y. Xu, S. Li, D. Zhang, Y. Jin, F. Zhang, N. Li, et al., Identification framework for cracks on a steel structure surface by a restricted Boltzmann machines algorithm based on consumer-grade camera images, Struct. Control Health Monitor. 25 (2) (2018), https://doi.org/10.1002/stc.2075 e2075.

[24] F. Ni, J. Zhang, Z. Chen, Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning, Comput.-Aid. Civ. Infrastruct. Eng.. 34 (5) (2019) 367–384, https://doi.org/10.1111/mice.12421.

[25] Y. Wu, X. Fan, J. Zhang, Ensemble learning model for concrete delamination depth detection using impact echo, NDT and E Int. 145 (2024) 103119, https://doi.org/10.1016/j.ndteint.2024.103119.

[26] Y. Wu, J. Zhang, C. Gao, J. Xu, Internal defect detection quantification and three-dimensional localization based on impact echo and classification learning model, Measurement 218 (2023) 113153, https://doi.org/10.1016/j.measurement.2023.113153.

[27] Z. He, C. Su, Y. Deng, A novel MO-YOLOv4 for segmentation of multi-class bridge damages, Adv. Eng. Inf. 62 (2024) 102586, https://doi.org/10.1016/j.aei.2024.102586.

[28] Z. He, S. Jiang, J. Zhang, G. Wu, Automatic damage detection using anchor-free method and unmanned surface vessel, Autom. Constr. 133 (2022) 104017, https://doi.org/10.1016/j.autcon.2021.104017.

[29] R. Pang, Y. Yang, A. Huang, Y. Liu, P. Zhang, G. Tang, Multi-scale feature fusion model for bridge appearance defect detection, Big Data Min. Anal.. 27 (1) (2024) 1–11, https://doi.org/10.26599/BDMA.2022.9020048.

[30] W. Choi, Y.-J. Cha, SDDNet: Real-time crack segmentation, IEEE Trans. Ind. Electron. 67 (9) (2020) 8016–8025, https://doi.org/10.1109/TIE.2019.2945265.

[31] J. Dong, N. Wang, H. Fang, W. Guo, B. Li, K. Zhai, MFAFNet: an innovative crack intelligent segmentation method based on multi-layer feature association fusion network, Adv. Eng. Inf. 62 (2024) 102584, https://doi.org/10.1016/j.aei.2024.102584.

[32] Z. He, W. Chen, J. Zhang, Y. Wang, Crack segmentation on steel structures using boundary guidance model, Autom. Constr. 162 (2024) 105354, https://doi.org/10.1016/j.autcon.2024.105354.

[33] W. Burger, M. J. Burge, Scale-Invariant Feature Transform. In: Digital Image Processing. Texts in Computer Science. Springer, London. doi: 10.1007/978-1-4471-6684-9_25.

[34] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, Speeded-up robust features, Comput. Vis. Image Understand. 110 (3) (2008) 346–359, https://doi.org/10.1016/j.cviu.2007.09.014.

[35] S. Filin, N. Pfeifer, Segmentation of airborne laser scanning data using a slope adaptive neighborhood, ISPRS J. Photogrammetry Remote Sens. 60 (2) (2006) 71–80, https://doi.org/10.1016/j.isprsjprs.2005.10.005.

[36] J. Chen, B. Chen, Architectural modeling from sparsely scanned range data, Int. J. Comput. Vis. 78 (2008) 223–236, https://doi.org/10.1007/s11263-007-0105-5.

[37] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Readings Computer Vision. (1987) 726–740, https://doi.org/10.1016/B978-0-08-051581-6.50070-2.

[38] F. Wang, S. Jiang, J. Zhang, Automatic measurement of grid structures displacement through fusion of panoramic camera and laser scanning data, Eng. Struct. 306 (2024) 117701, https://doi.org/10.1016/j.engstruct.2024.117701.

[39] R. Q. Charles, H. Su, M. Kaichun, L. J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[40] R.Q. Charles, Y. Li, S. Hao, J.G. Leonidas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: 31th International Conference on Neural Information Processing Systems (NeurIPS), ACM, California, USA, 2017, pp. 5105–5114.

[41] N. Engel, V. Belagiannis, K. Dietmayer, Point transformer, IEEE Access 9 (2021) 134826–134840, https://doi.org/10.1109/ACCESS.2021.3116304.

[42] J. Dong, N. Wang, H. Fang, H. Lu, D. Ma, H. Hu, Automatic augmentation and segmentation system for three-dimensional point cloud of pavement potholes by fusion convolution and transformer, Adv. Eng. Inf. 60 (2024) 102378, https://doi.org/10.1016/j.aei.2024.102378.

[43] G. Qian, Y.n Li, H. Peng, J. Mai, H. A. A. K. Hammoud, M. Elhoseiny et al., PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies, arXiv:2206.04670, Doi: 10.48550/arXiv.2206.04670. (Accessed 15 May 2023).

[44] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional Networks for Biomedical Image Segmentation, in: 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, Munich, Germany, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

[45] Y. Zhang, Q. Zhan, Z. Ma, EfficientNet-ECA: a lightweight network based on efficient channel attention for class-imbalanced welding defects classification, Adv. Eng. Inf. 62 (2024) 102737, https://doi.org/10.1016/j.aei.2024.102737.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv:2010.11929, Doi: 10.48550/arXiv.2010.11929. (Accessed 15 June 2024).

[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, et al., Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021, pp. 9992–10002, https://doi.org/10.1109/ICCV48922.2021.00986.

[48] S. Mehta, M. Rastegari, MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer, arXiv:2110.02178, Doi: 10.48550/arXiv.2110.02178. (Accessed 21 January 2024).

[49] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[50] X. Wang, R. Girshick, A. Gupta, K. He, Non-local Neural Networks, In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7794-7803, Doi: 10.1109/CVPR.2018.00813.

[51] J. Chen, S. Kao, H. He, W. Zhuo, S. Wen, C. H. Lee, et al., Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks, In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 12021-12031, Doi: 10.1109/CVPR52729.2023.01157.

[52] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information Process. Manag.. 45 (4) (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.

[53] M. Tan, Q.V. Le, EfficientNetV2: smaller Models and Faster Training, in: 38th International Conference on Machine Learning (ICML), PMLR, 2021, pp. 10096–10106.

[54] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 2022, pp. 11966–11976, https://doi.org/10.1109/CVPR52688.2022.01167.

[55] H. Touvron, M. Cord, H. Jégou, DeiT III: Revenge of the ViT, In: Computer Vision – ECCV 2022, Lecture Notes in Computer Science. 13684 (2022) pp. 516-533, Doi: 10.1007/978-3-031-20053-3_30.

[56] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, J. Ren, EfficientFormer: vision transformers at MobileNet speed, In: Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS), Red Hook, NY, USA, 2024, pp. 12934–12949, https://dl.acm.org/doi/10.5555/3600270.3601210.

[57] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, F. S. Khan, EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications. In: Computer Vision – ECCV 2022 Workshops. Lecture Notes in Computer Science. 13807 (2023) pp. 3-20, Doi: 10.1007/978-3-031-25082-8_1.