drones

MDPI

Article

Autonomous UAV-Based System for Scalable Tactile Paving Inspection

Tong Wang [1], Hao Wu [1], Abner Asignacion, Jr. [1], Zhengran Zhou [1], Wei Wang [2] and Satoshi Suzuki [1,*]

[1] Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
[2] Autonomous, Intelligent, and Swarm Control Research Unit, Fukushima Institute for Research, Education and Innovation (F-REI), Fukushima 979-1521, Japan
* Correspondence: suzuki-s@chiba-u.jp

Abstract

Tactile pavings (Tenji Blocks) are prone to wear, obstruction, and improper installation, posing significant safety risks for visually impaired pedestrians. This system incorporates a lightweight YOLOv8 (You Only Look Once version 8) model for real-time detection using a fisheye camera to maximize field-of-view coverage, which is highly advantageous for low-altitude UAV navigation in complex urban settings. To enable lightweight deployment, a novel Lightweight Shared Detail Enhanced Oriented Bounding Box (LSDE-OBB) head module is proposed. The design rationale of LSDE-OBB leverages the consistent structural patterns of tactile pavements, enabling parameter sharing within the detection head as an effective optimization strategy without significant accuracy compromise. The feature extraction module is further optimized using StarBlock to reduce computational complexity and model size. Integrated Contextual Anchor Attention (CAA) captures long-range spatial dependencies and refines critical feature representations, achieving an optimal speed–precision balance. The framework demonstrates a 25.13% parameter reduction (2.308 M vs. 3.083 M), 46.29% lower GFLOPs, and achieves 11.97% $mAP_{50:95}$ on tactile paving datasets, enabling real-time edge deployment. Validated through public/custom datasets and actual UAV flights, the system realizes robust tactile paving detection and stable navigation in complex urban environments via hierarchical control algorithms for dynamic trajectory planning and obstacle avoidance, providing an efficient and scalable platform for automated infrastructure inspection.

Keywords: drone; autonomous inspection system of tactile paving; YOLOv8-OBB; layered control algorithm

check for updates

Academic Editor: Higinio González Jorge

Received: 11 June 2025
Revised: 2 August 2025
Accepted: 4 August 2025
Published: 7 August 2025

Citation: Wang, T.; Wu, H.; Asignacion, A., Jr.; Zhou, Z.; Wang, W.; Suzuki, S. Autonomous UAV-Based System for Scalable Tactile Paving Inspection. *Drones* **2025**, *9*, 554. https://doi.org/10.3390/drones9080554

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

According to data from the World Health Organization (WHO), over 100 million individuals experience severe vision impairment or blindness, with primary causes including cataracts (94 million), age-related macular degeneration (8 million), glaucoma (7.7 million), and diabetic retinopathy (3.9 million) [1]. Additionally, the International Agency for the Prevention of Blindness (IAPB) estimates that approximately 43 million people are completely blind [2]. To support independent mobility for visually impaired individuals, tactile paving has become an essential urban infrastructure component. Originally developed by Seiichi Miyake in 1965, tactile paving, also known as Tenji Blocks, derives its name from the Braille writing system used by visually impaired individuals. Recognizing its importance, the Japanese government initiated widespread implementation in the 1970s, primarily in

railway stations and pedestrian pathways. Today, these systems are internationally adopted under different names, such as truncated domes (USA), blister paving (UK), and guiding blocks (China), and are collectively classified as Tactile Ground Surface Indicators (TGSIs). Their installation and design are now standardized under ISO 23599:2019 [3,4].

Tactile pavings function as detectable surface indicators predominantly fabricated from weather-resistant concrete for outdoor installations, though specialized materials like ceramic, polymer composites, and epoxy resins are alternatively deployed to meet site-specific requirements. As evidenced in Figure 1, these essential navigation aids remain vulnerable to operational compromises—including material erosion from sustained wear, structural deterioration through mechanical/environmental damage, and functional nullification by physical obstructions. To ensure reliable wayfinding assistance, routine inspections must simultaneously verify material integrity (combating crack formation and surface deformation) and enforce compliance protocols (maintaining unobstructed paths and standardized installation geometries). Current reliance on manual inspection methods is problematic, being inherently time consuming, labor intensive, and prone to error [5]. Ground-based robotic solutions offer some automation [6,7] but are hindered by limited maneuverability and field of view [8]. Unmanned Aerial Vehicles (UAVs) present a promising alternative, offering superior mobility, broader coverage, and potential for real-time analysis in urban infrastructure monitoring [9].



Figure 1. Tactile paving degradation and obstruction.

Vision-based drone applications have demonstrated effectiveness in infrastructure and pavement inspections. However, their application in tactile paving inspection remains relatively unexplored. Several existing studies have explored drone-based road inspection systems. Zhou et al. developed YO-FR, a lightweight distributed learning algorithm enabling multiple drones to collaboratively process environmental data for post-disaster road defect detection, highlighting drones' potential in emergency management [10]. Geng et al. introduced the Selective Dynamic Feature Compensation-YOLO (SDFC-YOLO) algorithm for pavement distress detection [11]. Their three-tier approach includes a Dynamic Downsampling Module (DDM) for adaptive sampling, a novel feature fusion technique, and a multi-scale weight selection module [12]. Muhammad et al. equipped drones with visual sensors to enable real-time infrastructure crack inspection in environments without GPS [13]. Wang et al. used drones combined with RGB and millimeter-wave radar to perform close-range power line inspections [14]. Despite these advances in aerial inspection, the specific application of UAV technology to the inspection of tactile paving—characterized by its standardized patterns, susceptibility to subtle defects critical for tactile function, and the requirement for low-altitude imaging—remains a significant and largely unexplored challenge. The unique visual characteristics and inspection requirements of tactile paving demand tailored solutions beyond generic pavement inspection approaches.

Building upon these advances in aerial inspection technologies, this paper introduces an autonomous drone system specifically designed for tactile paving inspection. The proposed approach integrates a lightweight YOLOv8 architecture with targeted enhancements to address the unique challenges of tactile paving detection. To validate the system's

capabilities, we conducted comprehensive experiments focusing on detection efficiency, precision, and real-world applicability.

The paper is organized as follows: Section 1 introduces the experimental background and outlines the main contributions. Section 2 provides current work related to the integration of drones with deep learning techniques. Section 3 presents the system architecture and methodology in detail. Section 4 describes the experimental setup, conducts comparison and ablation experiments to improve the model, and presents real flight test results and data analysis. Section 5 discusses the advantages and limitations of the system, concluding the study.

2. Existing Works on Drones with Deep Learning Techniques

Urban road inspections traditionally rely on manual visual methods, which are resource intensive, prone to human error, and lack efficiency [15]. These limitations highlight the need for innovative technologies in infrastructure maintenance. With the rise of Intelligent Road Inspection (IRI) concepts, unmanned aerial vehicles (drones) have emerged as a promising alternative. Equipped with deep learning models, drones have demonstrated significant potential in diverse applications, including post-disaster assessments [16], lane tracking [17], and crack detection [18,19].

Among existing studies, Zhou et al. introduced YO-FR, a framework enabling multiple drones to collaboratively process environmental data for post-disaster road detection. This highlights the application of drones in emergency management [16]. However, this approach primarily focuses on large-scale road damage assessment and lacks precision for detecting small-scale tactile paving elements. Maluya et al. leveraged lane segmentation models for drone-based road inspections, ensuring robust navigation along predefined tracks [17]. Despite its navigation capabilities, this method does not incorporate the block-type differentiation essential for tactile paving interpretation. Zhu et al. utilized improved YOLO models to detect multiple road elements, including pavement cracks and guardrails, emphasizing the versatility of drones for comprehensive road maintenance [18]. Nevertheless, their model treats all pavement indicators uniformly without recognizing the functional distinctions between tactile block types. Chen et al. developed a drone system based on MRC-YOLOv8 for crack detection in mountainous regions, demonstrating its adaptability to challenging terrains [19]. However, this system's crack-focused detection paradigm fails to address the geometric pattern recognition required for tactile paving analysis. Fan et al. introduced a robust drone-based stereo vision system, which can more easily distinguish damaged pavement from ordinary pavement to differentiate [20]. While effective for general defect identification, this approach cannot distinguish between go-block and stop-block configurations, which is critical for tactile navigation. Hassan et al. proposed a drone-based autonomous road inspection scheme that detects yellow lanes in real time through the YOLO model while flying along predefined routes. This approach also identifies potholes and cracks on the road surface [21]. This method's limitation lies in its exclusive focus on vehicular road elements, overlooking pedestrian-focused tactile infrastructure. Qiu and Lau conducted pioneering research using YOLO-based methods for tactile paving crack detection. While effective in identifying surface defects, their approach did not consider real-time navigation or the distinction between go-blocks and stop-blocks [22]. Although innovative for surface defect identification, their approach has two critical limitations: (1) It neglects real-time navigation requirements, and (2) it fails to distinguish between functionally distinct go-block and stop-block types.

Despite these advancements, most studies focus on general road conditions, overlooking the unique challenges posed by tactile paving systems. Tenji blocks, critical for guiding visually impaired pedestrians, require specialized detection methodologies to

address issues such as improper installation, obstruction, and wear. To bridge this gap, this study introduces an autonomous drone inspection system explicitly designed for tactile pavings. The proposed system integrates a lightweight YOLOv8-OBB model, enabling precise detection, classification, and tracking of tactile paving elements. Unlike previous methods, it dynamically identifies block types and adjusts the drone's flight path, ensuring that tactile paths remain central in the field of vision.

This study builds on existing drone applications by addressing the limitations of prior models. It combines advanced object detection techniques with real-time control algorithms, enabling efficient and accurate inspections of urban tactile infrastructure. Furthermore, its ability to differentiate block types and handle obstructions provides a new approach to tactile paving detection for drones.

3. Structure of the System and Methods

The proposed system for tactile paving inspection is designed to address the unique challenges posed by tactile paving elements, including real-time detection, obstacle handling, and lightweight operation. This section details the system architecture, hardware configuration, and control algorithms, emphasizing the integration of the YOLOv8-OBB model for accurate object detection.

3.1. Structure of the System

The inspection task for tactile pavings is multifaceted, involving detection of both go-blocks and stop-blocks, analysis of block dimensions, identification of obstructions, following the tactile paving wayfinding map, and other cloud/networking tasks. Deploying all these tasks onto a drone for real-time operation is challenging. Therefore, this system focuses on the detection and video acquisition of tactile paths, ensuring comprehensive coverage so that it can later be targeted and analyzed in detail in the cloud. The inspection tasks of this system are:

(1) Due to the different design requirements of the go-block and the stop-block, they are differentiated by the detection task so that the drone can follow the go-block.
(2) Obtain the steering trend of the go-block through the OBB task and adjust the drone heading in real time to ensure that the tactile path is always in the center of the inspection video.
(3) The common occupancy of tactile path, such as cars and bicycles, causes the tactile path targets in the drone's field of view to be missing, which greatly increases the difficulty of inspection and detection of tactile path, so this system identifies the occupancy of tactile path and processes them individually.

As shown in Figures 2 and 3, the inspection tasks of this system are:

(1) Safety Check, Receive Task and Map—The drone performs a safety check before deployment. Then, the cloud central station computes the waypoints from a wayfinding map and the specific task that consititutes a start and end point. The cloud central station then send the task details ($\chi_{\text{ref}} = \{\chi_{\text{ref}\,1} \cdots \chi_{\text{ref}\,N}\}$), where N is the number of waypoints and, $\chi_{\text{ref}\,1} = \begin{bmatrix} x_{ref1} & y_{ref1} & z_{ref1} \end{bmatrix}$ in the *WGS84* geodetic coordinate system.
(2) Fly to the Start Point—The drone moves to the designated starting position χ_{ref1} for inspection. The initial attitude is maintained throughout the flight unless there is an obstacle. An obstacle avoidance algorithm is triggered by a millimeter-wave forward radar, logs the event, and then computes the $\chi_{\text{obs-ref}}$ that has higher priority than the cloud waypoint χ_{ref} [23]. More advanced obstacle avoidance can be implemented with fusion of another sensor, such as a camera [24].

(3) OBB Detect (Oriented Bounding Box Detection)—The drone detects the tactile paving path and occupations using the YOLOv8 model. In the experimental section, we will show an example of an OBB detect task where the detection quality is also indicated. Three distinct situations are defined as follows:

   (a) If a tactile paving path is detected, then it classifies the block type, i.e., whether the block is a go-block (guiding block) or a stop-block (warning block). If the go-block path is detected and is clear, then the drone follows the centerline of the tactile paving for continuous inspection. If the stop-block path is detected and is clear, then the drone hovers first and then aligns the drone heading following the wayfinding map.

   (b) If an occupation, such as cars and bicycles, is detected, then the drone logs the position referencing on the tactile paving wayfinding map. Then, the drone continues with the task using the wayfinding map.

   (c) If an error occurs (e.g., no tactile paving detected), then it transitions to the error state and proceeds to the goal position of the task.

(4) Land on the Specified Area—The drone reaches the end of the inspection route and lands safely in the predesignated area.

(5) Task Completion—At the end of the inspection, the complete inspection video and tactile road occupancy information will be uploaded to the cloud central station to subsequently facilitate more-detailed tactile road inspection operations such as identifying cracks, damages, dimensions, etc.

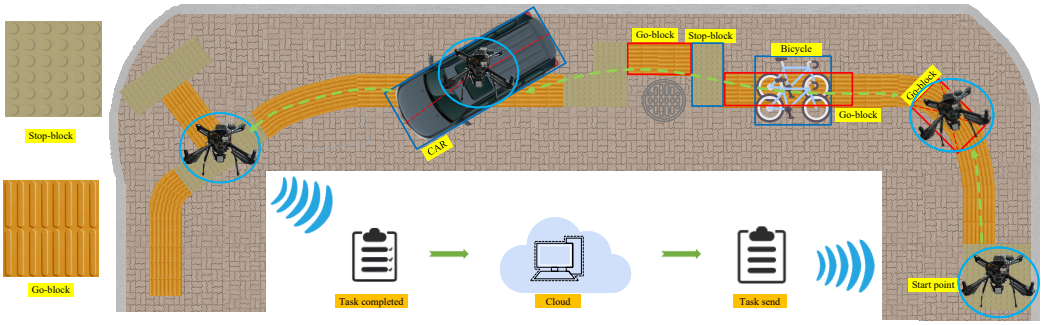


Figure 2. The structure of the system.

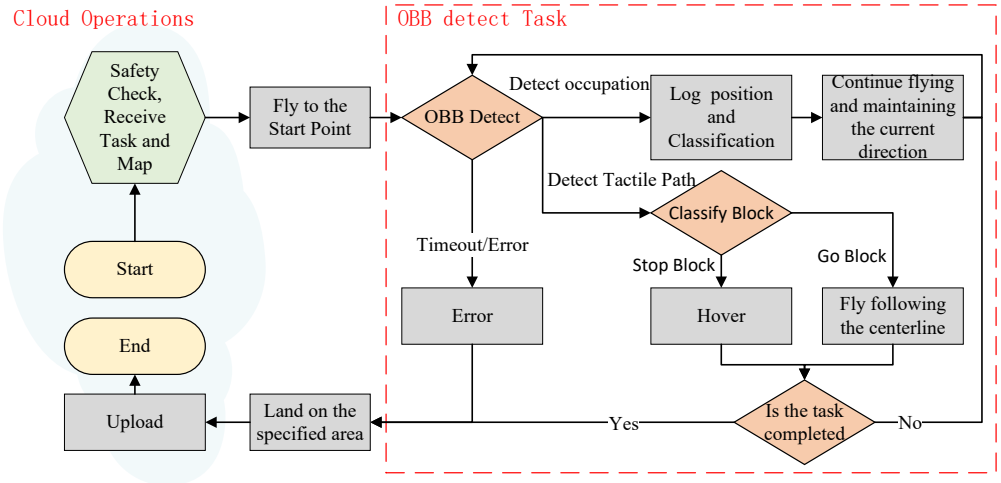


Figure 3. Task execution process.

3.2. Hardware Structure of the Drone

The hardware system of our designed drone is shown in Figure 4. The drone is equipped with a multi-sensor system and edge-computing hardware to ensure accurate and stable operation during inspection tasks. Key components include: AP, GPS, MAG, IMU, millimeter-wave forward radar, LTE model, ground proximity radar, Jetson AGX Xavier, and a fisheye camera.



Figure 4. Hardware structure of the drone.

The drone acquires precise position, attitude, and pose information through an integrated multi-sensor system that includes a GPS, magnetometer (MAG), and inertial measurement unit (IMU). Altitude is maintained using the IMU's inbuilt barometer module to ensure the vehicle's altitude stability, while forward radar and ground radar are used to provide environmental awareness to ensure safe flight.

Through the LTE module, particularly the core SIMCOM communication module, the drone achieves real-time interaction with the central station. This enables timely updates and transmission of flight status and operational progress while providing a critical remote intervention interface for operators. The capability to remotely send inspection waypoint tasks and execute basic control commands—such as hovering, takeoff, and landing—serves as the most direct and effective intervention measure for handling unexpected situations during flight (e.g., sudden obstacles or mission objective changes). This robust communication ensures real-time uploading of inspection data and timely reception of remote commands, safeguarding the integrity and smooth execution of inspection tasks.

Additionally, leveraging the Jetson AGX Xavier platform and a high-performance fisheye camera, the drone performs target detection tasks based on oriented bounding boxes (OBB). The fisheye camera features a diagonal field of view (FOV) of 160 degrees—significantly wider than conventional lenses (approximately 72 degrees)—dramatically expanding the information coverage per capture. This capability ensures that even during low-altitude operations, it captures expansive scenes (such as a full panorama of tactile paving) while minimizing the probability of target loss due to limited visibility. The AGX Xavier's high-performance computing power meets the real-time requirements of OBB detection tasks, and the fisheye camera's wide-angle coverage synergistically enhances detection accuracy and the overall reliability of drone inspection operations.

3.3. Control Algorithm

The drone control system shown in Figure 5 uses a layered control architecture to meet the real-time and stability requirements of the tactile paving detection task. The architecture is divided into high-level navigation and target tracking control and low-level attitude and thrust control.

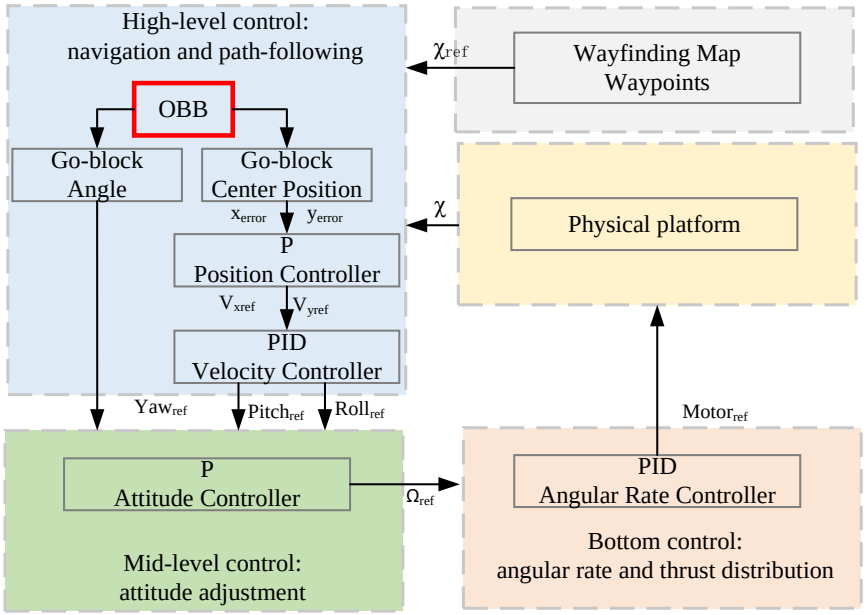


Figure 5. The structure of the control.

All layers implement PID control:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot de(t)/dt, \quad (1)$$

where $e(t)$ is error and $u(t)$ controller output. K_p, K_i, and K_d denote the proportional, integral, and derivative gains. OBB-based detection provides forward/lateral position errors to the P-position controller, converting these to x/y-direction velocity targets. These enable constant speed tracking along tactile paving curves. The PID velocity controller then converts velocity commands to attitude angle targets (x/y-directions), while OBB directly provides heading targets. This allows adaptive heading adjustment for continuous path tracking.

Attitude targets feed a P-attitude controller for roll/pitch fine-tuning, generating angular velocity references. This enables rapid response to high-level adjustments while minimizing positional errors. The PID angular velocity controller then converts attitude errors to precise angular rate commands. Finally, the mixer integrates all signals for motor thrust distribution.

The controller parameters are specified in Table 1:

Table 1. Controllers' parameters.

Controllers	Type	K_p	K_i	K_d
Position	P Control	0.7	–	–
Velocity	PID Control	12	2	1
Attitude	P Control	220	–	–
Angular Rate	PID Control	130	160	5

This layered architecture enables specialized functionality: Low-level controllers reject high-frequency disturbances through rapid attitude and thrust adjustments, while high-level controllers adapt navigation paths using real-time tactile paving detection. The coordinated operation maintains stable tracking in complex urban environments.

3.4. Improved YOLOv8-OBB

This study builds upon the YOLO architecture's inherent real-time capabilities and proven effectiveness in medium-to-large target detection, focusing on lightweight structural modifications to YOLOv8. The proposed enhancements to YOLOv8-OBB specifically aim to optimize tactile paving detection while maintaining robust performance across diverse datasets, including public benchmarks. Through computational efficiency improvements and refined feature representation capabilities, these architectural adaptations preserve the model's real-time advantages while achieving enhanced detection accuracy—ultimately elevating the benchmark model's comprehensive performance without compromising its core operational efficiency. Figure 6 illustrates the refined model structure. Through targeted component optimizations, we enhance computational speed (LSDE-OBB) and reduce the number of parameters (C2f-Star) while preserving detection precision (CAA). The optimizations aim to address the critical need for compact model deployment on edge devices while performing tactile paving detection tasks. The following section provides a detailed description of these key improvements.

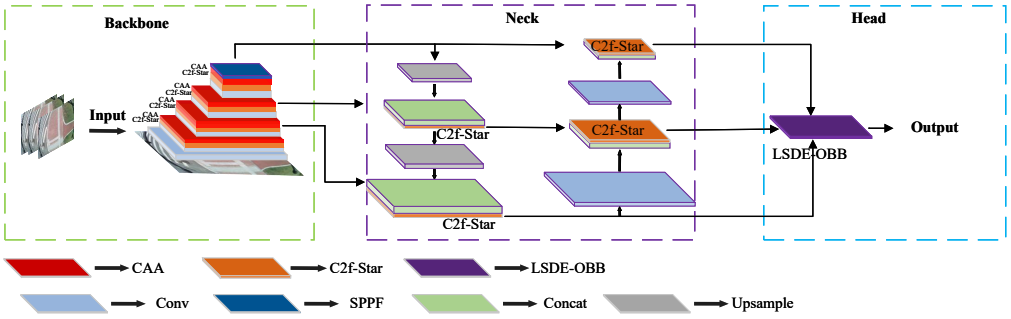


Figure 6. The structure of the improved YOLOv8-OBB.

In the YOLO architecture, multi-scale detection heads conventionally employ three independent branches to process object features at varying scales. This decoupled design pattern, while effective in feature representation, introduces two critical limitations:

(1) Parameter redundancy caused by duplicated convolution operations across branches
(2) Increased risk of overfitting due to insufficient regularization in small-batch training scenarios.

Such architectural constraints particularly hinder deployment on resource-constrained edge devices [8]. To address these challenges, we implement the LSDE-OBB detection head with two key innovations. First, we introduce shared convolutional layers across scale branches, effectively reducing network parameters through weight sharing while maintaining multi-scale representation capacity. The shared features are subsequently processed through scale-specific transformation layers to generate outputs at three resolution levels [25]. Second, we replace conventional batch normalization with group normalization (GN) [26] to enhance training stability. This modification proves particularly crucial given the reduced batch sizes required for edge-device compatibility, where traditional batch normalization often suffers from inaccurate statistical estimation that degrades detection performance. Replacing the convolution used for feature extraction with Detail Enhanced Convolution (DEConv) [27] proposed by DEA-Net can enhance representation and gen-

eralization capabilities while being equivalently converted to a normal convolution by a repartitioning technique without additional computational cost.

The C2f module is a crucial component in YOLOv8. It enhances the C3 module from YOLOv5 by achieving richer gradient flow information. This enhancement maintains the model's lightweight architecture. The C2f module in YOLOv8n introduces multiple bottleneck structures, which effectively solves the problems of disappearing and exploding gradients, improves network performance, and facilitates the use of a deeper architecture. The C2f module's excessive stacking of bottleneck structures introduces feature redundancy and irrelevant information in feature maps, increasing computational complexity while impairing detection accuracy [28]. To optimize this architecture, we implement StarBlock [29] as a replacement for conventional bottleneck components. This substitution maintains equivalent functionality but reduces computational costs through more efficient feature processing, thereby enhancing both efficiency and accuracy.

The Contextual Anchor Attention Module (CAA) is an advanced attention mechanism that efficiently balances computational cost with robust feature extraction [30]. Despite its streamlined design, CAA harnesses both global average pooling and 1D strip convolution to effectively model long-range pixel dependencies while emphasizing central features. In this way, it can efficiently extract local contextual information in images containing different scales and complex backgrounds and combine it with global contextual information to improve feature representation.

3.4.1. LSDE-OBB

Within the YOLO framework, conventional detection heads employ three independent branches for multi-scale target processing. However, this architecture can result in inefficient parameter utilization and an increased risk of overfitting due to isolated operations. To address these limitations, we introduce the Lightweight Shared Detail Enhanced Oriented Bounding Box (LSDE-OBB) head, as illustrated in Figure 7.

This unified detection head implements parameter sharing across all scales (highlighted in green), replacing the traditional triple-branch detection modules (depicted in Figure 7), thereby reducing model complexity and enhancing computational efficiency. The computational benefits of LSDE-OBB are further validated through comparative performance evaluations presented in Section 4.

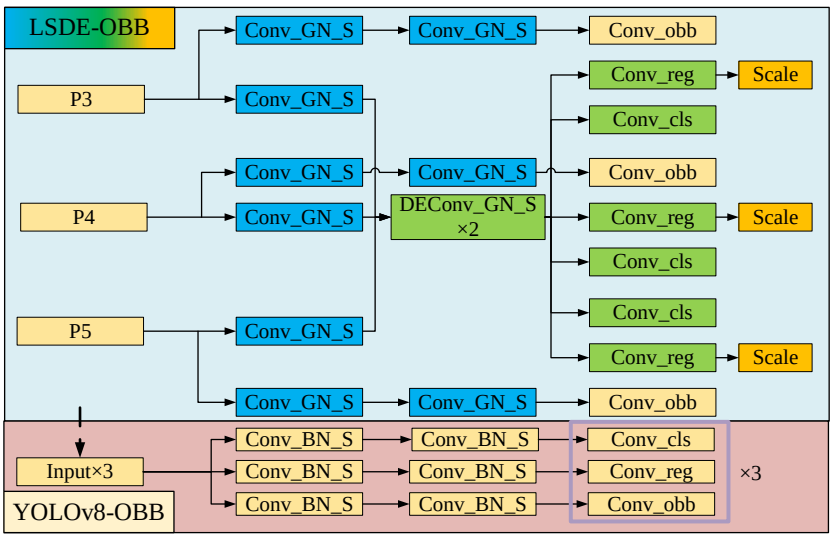


Figure 7. Comparison of YOLOv8-OBB head and LSDE-OBB head.

The rationale behind this approach stems from the consistent structural patterns exhibited by tactile pavements, making parameter sharing a viable optimization strategy without significantly compromising detection accuracy. However, to mitigate any potential loss in performance due to parameter sharing, two complementary strategies are incorporated.

(1) Normalization Strategy for Small-Batch Training: Conventionally, each CBS block comprises a standard convolution layer (C), batch normalization (BN), and SiLU activation (S). In our approach, batch normalization (BN) is replaced with group normalization (GN), which groups channels instead of relying on batch statistics. This substitution enhances stability in small-batch training and improves model robustness, as demonstrated in prior classification and localization research.

(2) Detail-Enhanced Convolution for Feature Representation: To further enhance feature extraction, the shared convolutional layers integrate Detail-Enhanced Convolution (DEConv) from DEA-Net (see Figure 8). Unlike conventional convolutions, DEConv combines standard convolution (SC) with four specialized differential convolution operators:

- Center Differential Convolution (CDC): Enhances edge sharpness.
- Angle Differential Convolution (ADC): Captures angular variations.
- Horizontal Differential Convolution (HDC): Refines horizontal structural details.
- Vertical Differential Convolution (VDC): Improves vertical directional information.

Each of these convolutional branches works in parallel, capturing complementary feature information. While SC extracts intensity features, CDC, ADC, HDC, and VDC enhance spatial structure details, facilitating more robust tactile paving detection.

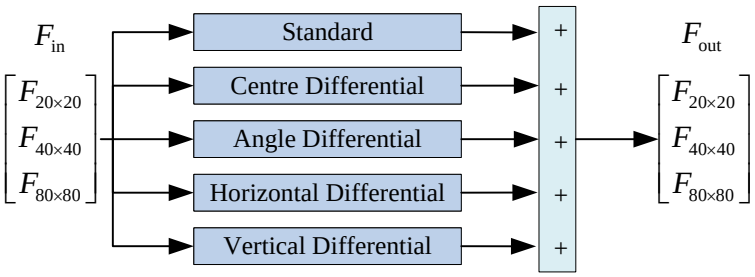


Figure 8. Detail-enhanced convolution (DEConv).

A reparameterization technique is employed to address the computational trade-off between multi-branch feature representation and inference efficiency. The transformation is expressed mathematically as

$$\begin{aligned} F_{\text{out}} &= \mathbf{DEConv}(F_{in}) \\ &= F_{in} * K_s + F_{in} * K_c + F_{in} * K_a \\ &+ F_{in} * K_h + F_{in} * K_v \quad \text{(for training)} \\ &= F_{in} * K_{cvt} \quad \text{(for deployment)} \end{aligned}, \tag{2}$$

where K_s, K_c, K_a, K_h, and K_v correspond to the five independent convolutional kernels associated with SC, CDC, ADC, HDC, and VDC, respectively. F_{in} and F_{out} are the inputs and outputs of the module. These kernels enable differential feature learning during the training phase. For deployment, these kernels are mathematically fused into a single equivalent kernel K_{cvt} through parameter fusion, ensuring that inference speed and computational resource usage remain comparable to standard convolution while preserving the enhanced representational capabilities of DEConv. This optimization allows DEConv to efficiently capture tactile path textures, which are crucial for tactile paving detection, without incurring additional computational overhead.

To further enhance detection stability across varying object scales, scale adjustment layers are incorporated after each shared convolutional layer in the regression head. These layers dynamically adjust feature resolutions, ensuring balanced multi-scale feature extraction for improved robustness in tactile paving detection. The implementation details can be found in Algorithm A1.

3.4.2. C2f-Star

The C2f module serves as a critical component for feature extraction, incorporating multiple bottleneck structures to enhance network performance. This design not only boosts gradient flow efficiency but also alleviates common issues like gradient vanishing and exploding while maintaining manageable computational overhead. Nevertheless, excessive use of bottleneck modules leads to redundant feature information. This redundancy increases computational costs and resource consumption while compromising detection accuracy [31]. Recent advancements in network architectures have introduced the Star Block, a module sharing functional similarities with traditional bottlenecks. This design maintains comparable structural simplicity while enhancing feature extraction capabilities and optimizing residual pathways to prevent gradient issues. The Star Block module captures the diversity of features through two different point-by-point convolutions after deep convolution and then multiplies the information of the two ways so that the network can integrate the information more accurately at different feature scales [29]. Therefore, we replace the bottleneck module with the Star Block module, which is lighter and has strong privilege extraction capability, as shown in Figure 9.

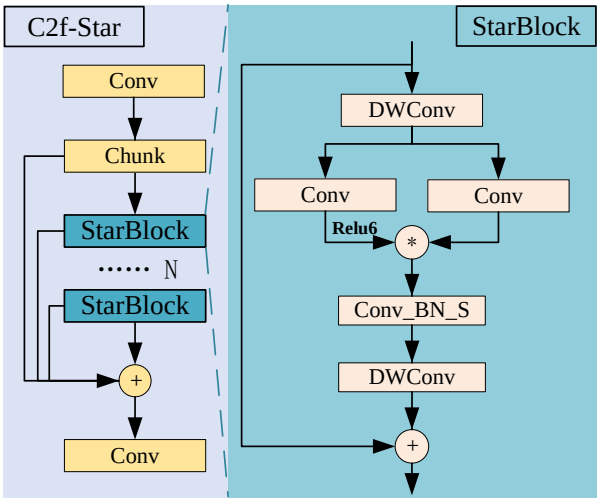


Figure 9. Comparison of C2f-Star and C2f module structures.

The primary innovation of Star Block lies in cross-layer element-wise multiplication, which fuses multi-level features without increasing network width, thereby effectively mapping inputs to higher-dimensional non-linear spaces. The inclusion of depthwise convolution DWConv, which applies a single filter per input channel with no cross-channel mixing, preserves feature richness while enhancing feature interactions. This improves feature representation accuracy and effectiveness.

$$\text{FLOPs} = H \times W \times C_{\text{in}} \times K \times K \tag{3}$$

Here, H and W represent the height and width of the input feature map, C_{in} denotes the number of input channels, and $K \times K$ corresponds to the kernel size. The FLOPs metric (Floating Point Operations Per Second) quantifies the computational cost by measuring the

number of floating point operations (additions, multiplications, etc.) required to process an input.

By incorporating Star Block into the C2f module, the resulting C2f-Star variant is optimized for tactile pavement detection. This architecture enhances boundary and detail recognition by propagating low-level texture information through cross-layer connections. This design is particularly well suited to capturing subtle texture variations in Tenji images and improves detection accuracy for complex shapes.

In addition, the high computational efficiency of the C2f-Star module reduces redundant computations and ensures that the computational burden is kept low when processing large-scale data. This allows the model to operate efficiently in real-time applications without sacrificing detection accuracy. With the optimized network structure, C2f-Star better adapts to the complex texture and shape variations in the Tenji task, further improving the overall detection performance and robustness. For instance, compared to the standard C2f module, the reconstructed modules achieved parameter retentions of 91%, 46%, 43%, 70%, 79%, 84%, 75%, and 72% across eight C2f modules, leading to a significant improvement in computational efficiency.

3.4.3. CAA

The Contextual Anchor Attention (CAA) mechanism extracts contextual information through a hierarchical approach, first via global average pooling, then sequential 1D convolutions applied horizontally and vertically. This multi-step process strengthens pixel-level feature relationships while enhancing central region details. Figure 10 illustrates the structure of the CAA attention module.

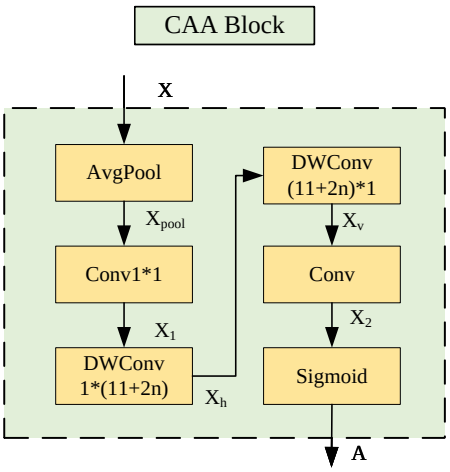


Figure 10. Context Anchor Attention (CAA) mechanism.

As illustrated in the schematic representation of the CAA module, the input feature maps undergo an initial processing stage where local spatial characteristics are extracted. This aggregation aims to generate a comprehensive global–spatial feature representation. To achieve this, input features initially undergo a 7×7 average pooling operation with a stride of 1 and padding of 3, reducing dimensionality while suppressing noise interference through local feature smoothing. This process is mathematically represented as

$$X_{\text{pool}} = \text{AvgPool2d}\,(X, 7, 1, 3), \tag{4}$$

where X denotes the input feature maps, X_{pool} represents the output feature maps after average pooling, "AvgPool2d" is the 2D average pooling function, 7 is the kernel size, 1 is the stride, and 3 is the padding size.

In the mean pooling operation, the input feature maps are aggregated by averaging the local regions to achieve a dimensionality reduction effect, which effectively smooths the feature maps, reduces the interference of noise, and makes the extracted features more robust. This method reduces the tendency of the model to overfit local noise and details, improves the generalization ability, and can alleviate feature fluctuations, making the texture, color and other features of the tactile path less susceptible to small-scale noise and enhancing the robustness of the model. At the same time, the feature map after average pooling contains less redundant information and the computational complexity is reduced, thus improving the model efficiency. Then the relationship between feature channels is enhanced by convolution to improve the information flow.

$$X_1 = \mathrm{Conv}_1\left(X_{\mathrm{pool}}\right), \tag{5}$$

X_{pool} is the input from the previous average pooling layer, Conv_1 represents a convolutional operation enhancing channel relationships, and X_1 is the output feature maps from this convolution.

Then, the feature maps are sequentially passed through the horizontal 1D convolution layer with the number of groups as the number of channels and the kernel as and the vertical 1D convolution layer with, respectively, in order to capture the contextual information in different directions, and this band convolution can also capture the features of the elongated shapes of the objects in a better way, as shown below:

$$X_h = DW\,\mathrm{Conv}_{1\times k_h}(X_1), \quad X_v = DW\,\mathrm{Conv}_{k_v\times 1}(X_h), \tag{6}$$

X_1 is the input to the horizontal convolution, $DW\,\mathrm{Conv}_{1\times k_h}$ denotes horizontal 1D convolution with kernel size $1 \times k_h$, and X_h is the output. Then, X_h is the input to the vertical convolution, $DW\,\mathrm{Conv}_{k_v\times 1}$ denotes vertical 1D convolution with kernel size $k_v \times 1$, and X_v is the output.

Traditional convolutional operations are typically implemented through standard convolution layers, whereas Depthwise Separable Convolution (DWConv) decomposes this process into two sequential stages: spatial-wise depth convolution and channel-wise point convolution. From a parameter efficiency perspective, conventional convolution layers exhibit a parameter count quantified as $\mathcal{O}\left(K^2 \cdot C_{\mathrm{in}} \cdot C_{\mathrm{out}} \cdot H \cdot W\right)$, where K represents the kernel dimension, with C_{in} and C_{out} denoting input and output channel counts. The spatial processing stage in DWConv demonstrates computational complexity scaling as $\mathcal{O}\left(K^2 \cdot C_{\mathrm{in}} \cdot H \cdot W + C_{\mathrm{in}} \cdot C_{\mathrm{out}} \cdot H \cdot W\right)$, where $H \times W$ specifies the feature map's spatial dimensions. Unlike conventional depthwise separable convolutions, this dual-directional convolutional design is specifically optimized to capture elongated object features and cross-pixel relationships. The use of bandlimited kernels enables parameter efficiency comparable to large-receptive-field convolutions. The horizontal convolution extracts row-wise contextual patterns, while the vertical convolution focuses on columnar dependencies, collectively modeling the structural characteristics of tactile paths.

Next, high-level contextual features are extracted from the feature maps using point-wise convolution and the Sigmoid activation function to obtain enhanced feature maps. Through the application of adaptive weights to the input feature representations, this mechanism prioritizes key features by amplifying their significance while diminishing less-critical aspects. This process improves the model's ability to focus on the key regions of interest.

$$X_2 = \mathrm{Conv}_2(X_v), \quad A = \sigma(X_2) \tag{7}$$

X_v is the input from the vertical convolution, Conv_2 represents a pointwise convolution operation used to generate high-level contextual features, X_2 is the output from this pointwise convolution, and A denotes the attention weights obtained after applying the Sigmoid activation function σ to X_2.

Subsequent point-wise convolution followed by sigmoid activation generates channel-aware attention weights, adaptively amplifying critical regions while suppressing irrelevant features. This hierarchical fusion mechanism dynamically integrates multi-scale contextual information through three coordinated effects:

(1) Noise suppression: The initial pooling operation establishes noise-resistant base features.
(2) Shape-specific sensitivity: Directional convolutions enhance object structure detection.
(3) Adaptive feature weighting: The attention mechanism prioritizes discriminative texture patterns.

For tactile paving detection, the CAA module demonstrates improved attention allocation precision by leveraging context-aware feature selection.

4. Experiment

This section presents the experimental framework, encompassing dataset configuration, evaluation metrics, and results analysis. The experiments consist of comprehensive assessments, including performance evaluations and actual flight testing, to thoroughly evaluate the system's capabilities. Our primary objective is to demonstrate the effectiveness and adaptability of the proposed YOLOv8-OBB model in detecting tactile paving as well as its robustness across various datasets.

4.1. Dataset Establishment

The experimental section of this paper uses two datasets: self-built datasets and the VisDrone/DroneVehicle dataset [32].

Drone-based vehicle recognition focuses on identifying vehicle positions and types within overhead imagery. This technology is crucial for enhancing urban traffic management systems and optimizing emergency response operations. Our experiments utilize the DroneVehicle dataset comprising 56,878 drone-captured images (half are RGB), though only RGB images are employed due to their relevance to our target inspection scenarios. During preprocessing, the 100-pixel white borders surrounding each image designed for boundary object annotation are systematically removed. The RGB subset is partitioned into training (70%), validation (20%), and testing (10%) sets, with comprehensive oriented bounding box annotations across five categories: cars (389,779 annotations), trucks (22,123 annotations), buses (15,333 annotations), vans (11,935 annotations), and freight cars (13,400 annotations). This preprocessed RGB dataset serves as the foundation for all subsequent comparative experiments and ablation studies in this work.

The dataset developed for this research was gathered through two primary methods: autonomous drone missions and online sources. Due to the scarcity of existing datasets addressing specific issues such as tactile paving damage or misplacement, our dataset primarily includes go-blocks and stop-blocks of tactile paving, along with common road objects like cars and bicycles. In total, 2032 images were compiled, divided into three subsets: a training set (1422 images), validation set (406 images), and test set (204 images).

4.2. Test Environment and Evaluation Indicators

4.2.1. Test Environment Configuration

As detailed in the experimental environment Table 2, the platform used for model training in this study is configured as follows: The operating system is Windows 11, the GPU is a GeForce RTX 3090 Ti with a computing capability of 8.6, and the CPU is an

Intel (R) Core(TM) i7-14700K. The application development is carried out using Python v3.8.19, with PyTorch v2.2.1 as the deep learning framework and CUDA 11.8 for GPU acceleration. The initial parameters for model training include an image input size of 640 × 640, the incorporation of the L1 loss function from the start of training, and a training epoch of 200. During the training phase, to facilitate a fair comparison of model performance, pre-trained weights are not used. The model's weight updates are optimized using the stochastic gradient descent (SGD) optimizer.

Table 2. Experiment environment.

Project	Environment	Hyperparameter	Setting
CPU	Intel Core i7	Input Resolution	640 × 640
GPU	RTX 3090ti	Optimizer	SGD
CUDA	11.8	Batch Size	16
Pytorch	2.2.1	Workers	4
Python	3.8.19	Epochs	200

4.2.2. Evaluation Indicators

This study establishes a multi-dimensional performance evaluation system tailored to the practical deployment requirements of drone-based inspection systems. For comprehensive definitions and detailed explanations of all evaluation metrics, including their mathematical formulations and practical significance in tactile paving detection, refer to Appendix D.

$$P = \frac{\text{Number of Correctly Detected Targets}}{\text{Number of Correctly Detected Targets + Number of False Alarms}}, \tag{8}$$

$$R = \frac{\text{Number of Correctly Detected Targets}}{\text{Number of Correctly Detected Targets + Number of Missed Targets}}, \tag{9}$$

$$\text{mAP} = \frac{\sum(\text{Precision-Recall Area for Each Category})}{\text{Total Number of Categories}}, \tag{10}$$

$$IoU = \frac{\text{Overlapping Area of Predicted and Ground Truth Boxes}}{\text{Combined Area of Predicted and Ground Truth Boxes}}, \tag{11}$$

4.3. Comparison and Ablation Experiments

The enhancement modules undergo dual validation to confirm their effectiveness. Individual component verification employs the public VisDrone/DroneVehicle dataset, establishing module independence while enabling direct comparison with counterparts. Parallel ablative tests on the proprietary Tenji OBB dataset specifically demonstrate the modules' capability in addressing oriented aerial target detection challenges.

4.3.1. Head Module

In Table 3, optimal values are highlighted in red for comparative analysis. For runtime implementation clarity, comprehensive execution results are provided in Appendix B Figure A1. The results demonstrate that the LSDE-OBB module achieves a superior balance between efficiency and accuracy. Critically, these results are a direct consequence of its specific architectural innovations, which provide the scientific evidence underpinning the performance:

Table 3. Head modules comparison.

Head	Par (M)	GFLOPs	Precision (%)	Recall	mAP_{50} (%)
baseline	3.083	8.44	74.084	0.714	75.195
LSCD [28]	2.381	6.7	70.954	0.689	72.54
LADH [33]	3.128	8.08	71.857	0.699	69.451
TADDH [34]	2.285	8.9	72.925	0.708	71.565
LSDE-OBB	2.597	5.406	73.949	0.726	74.761

Optimal values are highlighted in red for comparative analysis.

Parameter Sharing and Unified Head: The core design of LSDE-OBB replaces the traditional three-branch detection head (Figure 7) with a single shared head operating across all scales. This fundamental structural change is the primary driver behind the significant reduction in parameters (2.597 M vs. baseline 3.083 M, −15.7%) and computational complexity (5.406 GFLOPs vs. baseline 8.44 GFLOPs, −35.9%). The shared head eliminates the redundant parameters inherent in independent branches.

Group Normalization (GN) for Robustness: Replacing Batch Normalization (BN) with Group Normalization (GN) within the shared head's CBS blocks (Section 3.4.1) directly addresses the challenge of small-batch training stability. This architectural choice enhances model robustness, particularly crucial for maintaining high recall (0.726, highest among all heads) and mAP_{50} (74.761%, highest among all heads). GN's independence from batch statistics prevents performance degradation when batch sizes are limited, a common scenario in specialized detection tasks.

Detail-Enhanced Convolution (DEConv) for Precision: The integration of DEConv (Figure 8) into the shared convolutional layers is key to compensating for potential accuracy loss due to parameter sharing and achieving high precision (73.949%, second only to baseline). DEConv's multi-branch design (SC, CDC, ADC, HDC, and VDC) explicitly enhances the capture of critical tactile pavement features: CDC sharpens edges essential for delineating tactile path boundaries; ADC captures angular variations characteristic of tactile paving patterns; and HDC and VDC refine horizontal and vertical structural details prevalent in these textures This explicit multi-directional feature enhancement capability allows LSDE-OBB to extract richer discriminative information than simpler heads, directly contributing to its high precision score.

Reparameterization for Efficiency: While DEConv's multi-branch structure is vital for feature enhancement during training, its direct use in inference would incur a computational cost. The reparameterization technique (Equation (1)) mathematically fuses the five training kernels (K_s, K_c, K_a, K_h, K_v) into a single equivalent kernel (K_{cvt}) for deployment. This fusion is the critical mechanism that allows LSDE-OBB to leverage the representational power of DEConv without sacrificing inference efficiency, resulting in the remarkably low GFLOPs (5.406).

Scale Adjustment Layers for Multi-scale Stability: The incorporation of scale adjustment layers after shared convolutions in the regression head dynamically adapts feature resolutions. This design element contributes to the robustness across object scales, further supporting the high recall and mAP_{50} values by ensuring balanced feature extraction regardless of tactile paving size (implementation details in Algorithm A1).

Contrast with Alternatives: The success of LSDE-OBB lies in this systematic optimization. While LSCD suffers from severe degradation due to over-simplification, LADH's adaptive mechanism inadequately compensates for sharing losses (low recall/mAP), and TADDH's decoupled design lacks targeted feature enhancement (lowest mAP). LSDE-OBB synergistically combines the following:

- Structural Efficiency: Achieved through unified parameter sharing.

- Performance Compensation: Delivered by GN (robustness), DEConv (feature enhancement), and scale adjustment (multi-scale stability).
- Inference Optimization: Enabled by DEConv reparameterization.

This holistic approach effectively mitigates the trade-off, enabling LSDE-OBB to outperform other lightweight heads significantly and approach the accuracy of the heavier baseline while being substantially more efficient.

4.3.2. Feature Extraction Modules

Table 4 presents a comprehensive comparison of state-of-the-art feature extraction modules, with runtime evidence in Appendix B Figure A2. The quantitative results demonstrate that the proposed C2f-Star module achieves an optimal balance, attaining the lowest parameter count (2.609 M) and the highest detection precision (75.008%) while significantly reducing computational complexity (7.2 GFLOPs vs. baseline 8.44 GFLOPs, −14.7%). Critically, these metrics are direct outcomes of C2f-Star's architectural innovations, which provide scientific evidence for its superior performance.

Table 4. Feature extraction modules comparison.

Feature Extraction Module	Par (M)	GFLOPs	Precision (%)	Recall	mAP_{50} (%)
baseline	3.083	8.44	74.084	0.714	75.195
RepLKBlock [35]	2.908	8.8	73.895	0.696	74.108
DBB [36]	3.7	10.3	74.823	0.719	75.304
Ghostblockv2 [37]	2.613	7.1	72.58	0.689	72.762
C2f-Star	2.609	7.2	75.008	0.701	74.865

Optimal values are highlighted in red for comparative analysis.

Structural Efficiency via Star Block Replacement: The fundamental redesign replacing bottleneck modules with Star Blocks (Figure 9) directly addresses feature redundancy in the original C2f. This structural change is the primary driver for the dramatic parameter reduction (2.609 M vs. baseline 3.083 M, −15.4%) and computational savings (7.2 GFLOPs). The parameter retention rates (43–91% across modules) quantitatively validate this compression efficiency.

Deepwise Convolution (DWConv) as Computational Foundation: The Star Block's use of DWConv establishes its efficiency base. DWConv's channel-wise operation minimizes computations while preserving feature richness. This directly enables C2f-Star's low GFLOPs (7.2).

Dual Pointwise Convolution Paths for Feature Diversity: The two independent 1×1 convolutional branches after DWConv generate complementary feature representations at minimal cost. This design solves the diversity limitation of GhostBlockV2 (which uses linear operations), allowing C2f-Star to maintain high recall (0.701) despite significant parameter reduction.

Element-wise Multiplication for Nonlinear Enhancement: The core innovation lies in pointwise multiplication of the dual-path outputs. This operation creates high-dimensional nonlinear interactions (without adding parameters/width) that enhance discriminative power for subtle pattern variations. This directly produces the highest precision (75.008%) among all modules.

This multiplicative fusion provides significantly stronger nonlinear representation than GhostBlockV2's linear approach (72.58% precision) or RepLKBlock's large kernels (73.895% precision).

Comparative Analysis of Performance Trade-offs: DBB: Achieves high recall/mAP (0.719/75.304%) through multi-branch richness, but at prohibitive cost (3.7 M Params,

10.3 GFLOPs). Its parallel structure lacks the multiplicative fusion that gives C2f-Star its precision advantage.

GhostBlockV2: Maximizes efficiency (7.1 GFLOPs) via linear "ghost" features but suffers fundamental limitations in feature discrimination (lowest precision/recall/mAP). This validates the necessity of C2f-Star's nonlinear fusion.

RepLKBlock: Demonstrates that large kernels alone cannot compensate for local texture details (underperforms baseline). Its inefficiency (8.8 GFLOPs) highlights the advantage of C2f-Star's targeted design.

This holistic approach achieves what alternatives cannot: simultaneous reduction in parameters (−15.4%) and computations (−14.7%) while increasing precision (+1.24% over baseline). The quantitative evidence in Table 4 conclusively demonstrates how C2f-Star's architectural choices generate these results, making it ideal for real-time tactile path detection.

4.3.3. Attentional Mechanism

Table 5 presents a systematic comparison of attention mechanisms, with optimal values highlighted in red. The results demonstrate that the proposed Contextual Anchor Attention (CAA) mechanism achieves superior performance across all three core detection metrics: highest precision (77.174%), highest recall (0.724), and highest mAP_{50} (77.616%). Critically, this performance advantage is directly attributable to CAA's hierarchical, direction-sensitive architecture (Figure 10), which provides scientific evidence for its effectiveness in tactile path detection.

Table 5. Attention mechanisms comparison.

Attention Mechanisms	Par (M)	GFLOPs	Precision (%)	Recall	mAP_{50} (%)
Baseline	3.083	8.44	74.084	0.714	75.195
CA [38]	3.09	8.45	73.946	0.705	74.768
MLCA [39]	3.083	8.44	76.004	0.699	74.872
SimAM [40]	3.083	8.44	75.886	0.712	75.355
BiLevelRoutingAttention [41]	3.349	8.68	75.075	0.717	75.743
CAA [30]	3.261	8.96	77.174	0.724	77.616

Optimal values are highlighted in red for comparative analysis.

Noise Suppression via Initial Pooling: The 7×7 average pooling operation (Equation (4): $X_{\text{pool}} = \text{AvgPool2d}(X, 7, 1, 3)$) serves as the foundational stage. This operation smooths local features while preserving global context, reduces sensitivity to irrelevant background noise, and enhances generalization capability.

This noise robustness directly contributes to CAA's high recall (0.724) and mAP_{50} (77.616%) in complex environments.

Directional Feature Extraction via Sequential DWConv: The proposed method introduces a directional depthwise convolution strategy, sequentially applying horizontal (DWConv1 $\times k_h$) and vertical (DWConv$k_v \times 1$) convolutions (Equation (6)). This anisotropic decomposition effectively models the linear texture patterns characteristic of tactile paths, captures long-range dependencies more efficiently than conventional isotropic kernels, and offers significant computational advantages over large-kernel counterparts such as RepLKBlock. The directional sensitivity is the primary driver behind CAA's exceptional precision (77.174%).

Adaptive Weighting via Sigmoid Activation: The final stage (Equation (7): $A = \sigma(\text{Conv2}(X_v))$) generates channel attention weights that dynamically amplify discriminative regions (e.g., tactile dots/bars), suppress irrelevant background features, and optimize feature utilization for the task. This adaptive focusing synergistically enhances both precision and recall.

Computational Efficiency Analysis: While CAA introduces moderate parameters (3.261 M vs. baseline 3.083 M, +5.8%), GFLOPs (8.96 vs. 8.44, +6.2%) increase. The directional convolution design achieves better performance-per-computation than BiLevelRoutingAttention (CAA: +2.1% mAP_{50} at +0.28 GFLOPs vs. BiLevel: +0.55% mAP_{50} at +0.24 GFLOPs). The overhead is justified by the significant accuracy gains (+3.22% mAP_{50} over baseline).

Comparative Mechanism Analysis: CA: Lacks explicit noise suppression and directional modeling, resulting in marginal improvements. MLCA/SimAM (Parameter-Free): Achieve efficiency (red Params/GFLOPs) and decent precision, but their isotropic attention cannot match CAA's directional sensitivity, leading to lower recall/mAP. BiLevelRoutingAttention: Shows promise in recall/mAP (0.717/75.743%), but its generic sparse attention is less optimized for linear textures than CAA's dedicated directional design.

The success of CAA stems from its task-specific optimization: Initial pooling establishes noise-robust features; directional convolutions model blind lane geometry; and adaptive weighting focuses on critical textures.

This hierarchical approach explains why CAA achieves 2.17% higher precision than the next best (MLCA) and 1.87% higher mAP_{50} than BiLevelRoutingAttention, as quantitatively evidenced in Table 5.

4.3.4. Ablation Experiment

This section employs systematic component ablation (e.g., removing or modifying specific modules, layers, or functionalities) to quantitatively evaluate each component's contribution to overall performance. The impact of these modifications on key metrics, including precision and computational efficiency, is analyzed. Ablation studies conducted on the VisDrone and Drone-TactilePath datasets validate the cumulative effect of each proposed module. The comparative results are summarized in Table 6.

Table 6. Ablation experiment.

C2f-Star	LSDE-OBB	CAA	VisDrone–DroneVehicle					Drone-TactilePath			
			Par (M)	GFLOPs	Precision	mAP_{50}	$mAP_{50:95}$	Precision	Recall	mAP_{50}	$mAP_{50:95}$
			3.083	8.44	74.084	75.195	58.026	94.094	0.928	95.136	70.046
✓			2.609	7.214	75.008	74.865	58.008	93.464	0.931	93.773	72.911
	✓		2.597	5.406	73.949	74.761	58.308	94.661	0.924	95.441	76.477
		✓	3.261	8.961	77.174	77.616	60.194	92.927	0.941	95.56	76.776
✓	✓		2.123	4.174	73.751	73.376	56.865	94.606	0.924	94.56	76.432
✓	✓	✓	2.308	4.533	75.725	75.196	58.099	94.158	0.935	95.585	78.43

① Red indicates performance improvement, and blue indicates performance degradation. ② The checkmark (✓) indicates the module used in this experiment.

In Table 6, red indicates performance improvement, and blue indicates performance degradation.

As shown in Table 6, row 1 presents the baseline performance of YOLOv8n, the smallest model in the YOLOv8 series.

Introducing only C2f-Star (Row2) reduces parameters by 15.4% (2.609M). This stems from replacing the original bottleneck with the Star Block (Figure 9), whose depth-wise separable convolution (DWConv) compresses channels via $\mathrm{DWConv}(X) = \mathrm{Depthwise}(X) + \mathrm{Pointwise}(X)$. Computational complexity decreases by 14.5% (7.214 GFLOPs) due to DWConv's per-channel operations. Accuracy on VisDrone improves by 0.924% (75.008%), attributable to the dual-path outputs of pointwise multiplication, enhancing nonlinear representation for subtle texture discrimination. On TactilePath, recall slightly increases by

0.003 (0.931), indicating mitigated information loss due to the lightweight design; however, localization fluctuations reveal Star Block's limitations in modeling geometric deformations.

Using only LSDE-OBB (Row3) reduces GFLOPs by 35.9% (5.406) through its single-head shared convolution (Figure 7). Accuracy on TactilePath improves by 0.567% (94.661%) due to DEConv's multi-branch design (Figure 8): The CDC branch (K_c) enhances edge responses, while the ADC branch (K_a) captures angle variations. The 6.431% $mAP_{50:95}$ gain (76.477%) demonstrates group normalization's (GN) stabilizing effect, improving localization robustness at high IoU thresholds via batch-statistic independence.

Integrating only CAA (Row4) boosts VisDrone accuracy by 3.09% (77.174%) through explicit directional convolution chain modeling ($X_v = \mathrm{DWConv}_{k_v \times 1}(\mathrm{DWConv}_{1 \times k_h}(X_1))$), optimized for linear textures and noise suppression. Recall improves by 0.027 (0.724) as the initial 7×7 pooling ($X_{\mathrm{pool}} = \mathrm{AvgPool2d}(X, 7, 1, 3)$) smooths local perturbations, enhancing target continuity perception. Despite a 6.2% GFLOP increase (8.961), the directional design yields significant task-relevant feature gains.

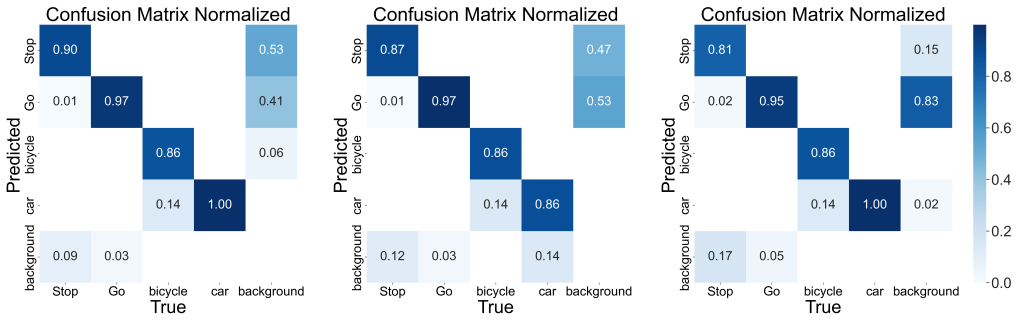
Combining C2f-Star and LSDE-OBB (Row5) reduces VisDrone accuracy by 0.333% (73.751%), indicating that dual lightweighting causes feature degradation; Star Block's element-wise multiplication cannot compensate for scale sensitivity loss from shared convolution. Conversely, TactilePath $mAP_{50:95}$ improves by 6.386% (76.432%), demonstrating synergy between DEConv's boundary optimization and Star Block's texture enhancement.

Integrating all modules (Row6) significantly improves TactilePath $mAP_{50:95}$ by 8.384% (78.43%) via cascaded reinforcement: C2f-Star ensures feature diversity, CAA applies directional attention weighting, and LSDE-OBB refines boundaries. VisDrone accuracy gains 1.641% (75.725%), confirming that CAA compensates for lightweight feature loss, particularly suited to tactile paths' linear structures. The final parameters were reduced by 25.1% (2.308 M), demonstrating module-level compression (parameter retention rate of 43–91% from StarBlock; LSDE-OBB savings of 16.8%).

The final experiment confirmed that synergistic combination of three core innovations—the feature-enhancing C2f-Star module, the boundary-optimizing LSDE-OBB head, and the geometrically aware CAA mechanism—collectively breaks the accuracy–efficiency trade-off bottleneck. This integrated framework achieves substantial gains in localization accuracy ($mAP_{50:95}$ ↑11.97%) while maintaining lightweight operation (parameters ↓25.1%, GFLOPs ↓46.29%).

4.3.5. Model Comparison Experiment

Figure 11 shows the confusion matrix comparison.



(**a**) Ours. (**b**) Yolov8n-OBB. (**c**) Yolov11n-OBB.

Figure 11. Confusion matrix comparison.

Figure 11 provides a side-by-side comparison between our enhanced model and the latest oriented bounding box (OBB) frameworks. Panel (a) depicts our model's confusion matrix, while panels (b) and (c) showcase the performance of YOLOv8n-OBB and YOLOv11n-OBB, respectively.

The elements along the diagonal indicate the percentage of correctly predicted samples, highlighting that our model achieves the highest recognition accuracy. Due to the presence of angle prediction in the OBB task, the localization difficulty is higher compared to general detection tasks, leading to a relatively higher false positive rate. In this study's tactile path detection task, the primary focus is on the OBB detection of go-blocks. Our model achieves the lowest false positive rate for go-block detection, indicating the smallest localization error among the compared models.

4.4. System Validation

The system was validated through comprehensive real flight tests under diverse operational conditions, including forward flight, turning maneuvers, and scenarios involving dynamic object occupancy. All detection results presented in this study were obtained using a fisheye camera configuration, which provides critical advantages for drone navigation.

The fisheye camera's ultra-wide-angle lens maximizes downward field of view coverage, enabling complete capture of the tactile path target area even during low-altitude flight operations. This hemispherical imaging capability ensures continuous target monitoring within a 160-degree field of view, maintaining navigation accuracy near ground surfaces where conventional lenses would suffer from significant blind spots.

This optical superiority proves particularly crucial in object occupancy scenarios. When obstacles partially occlude the path, traditional narrow-field lenses frequently lose target tracking due to their limited coverage area. In contrast, the fisheye lens system, with its peripheral vision capabilities, can maintain visual contact with moving and stationary obstacles, enabling continuous guidance even when the path is blocked.

During inspection, the system differentiates between go-blocks and stop-blocks to ensure accurate drone guidance. This avoids potential path deviations caused by mixed detections.

The field tests were conducted under controlled environmental conditions: summer season with predominantly overcast skies providing consistent, diffuse illumination. All experiments operated within cordoned-off pedestrian pathways to ensure operational safety and eliminate external interference from human activity.

4.4.1. Straight Line Verification

To validate system performance under straight-line conditions, we conducted the following experiments:

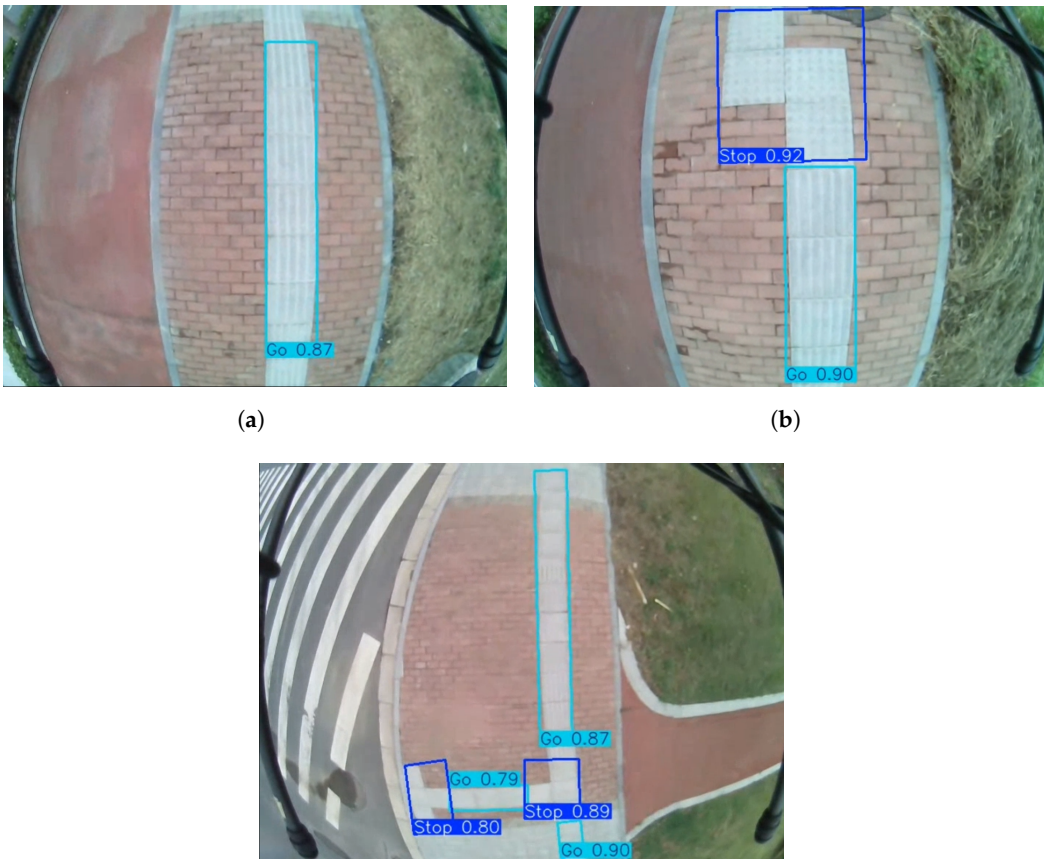
Figures 12a, 12b, and 12c demonstrate three distinct block detection scenarios: single go-block, mixed stop-block and go-block, and multiple go-blocks, respectively.

In the scenario with only a go-block (Figure 12a), the system recognizes all detected go-blocks as a single target. The drone's flight attitude is then adjusted based on the target's overall directional information and centroid position.

When both go-blocks and stop-blocks appear simultaneously (Figure 12b), the system filters out stop-blocks to prevent interference with flight operations.

In more complex configurations where go-blocks and stop-blocks are interleaved (Figure 12c), the system performs additional analysis on multiple go-blocks, filtering out undersized targets.

As evidenced in Figure 12b,c, this filtering mechanism effectively processes redundant forward blocks during coexistence scenarios, ensuring correct flight direction maintenance.



(**a**) (**b**)

(**c**)

Figure 12. Forward flight: (**a**) Single go-block scenario, (**b**) Multiple go-blocks and stop-blocks coexisting, (**c**) Multiple go-blocks scenario. block detection scenarios.

4.4.2. Turn Verification

Figure 13 shows the scene during steering, when the OBB task extracts the bending trend of the go-block, on the basis of which the drone is able to adjust its flight heading in real time.



Figure 13. Turn flight.

The steering angle data in Figure 14 shows the change in angle of the drone when performing the steering operation. The red line represents the target heading angle (derived from OBB detection), while the blue line shows the current heading angle. The illustration reveals that the drone precisely adjusts its flight attitude during steering maneuvers. This shows that the drone is able to correct its flight angle in real time based on the steering trend detected by the target, ensuring that the flight path meets the desired goal.

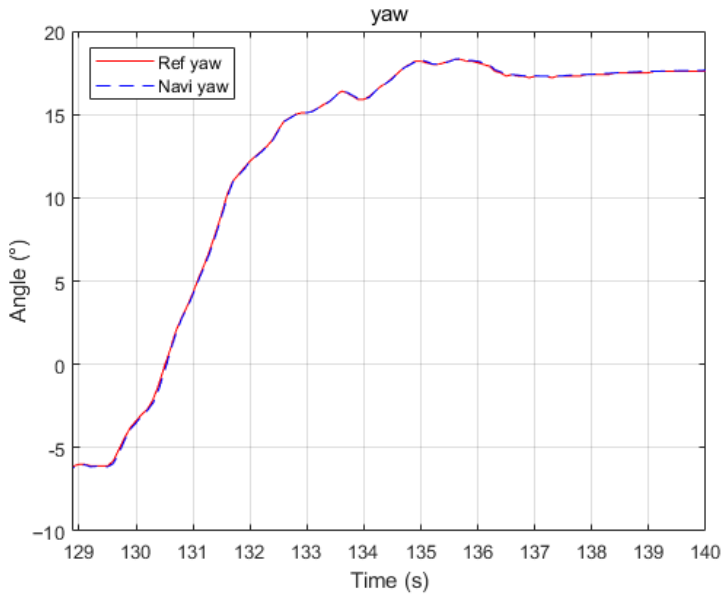


Figure 14. Heading adjustment data.

4.4.3. Object Occupancy Policy Validation

During the inspection process there will be tactile paving occupancy as in Figure 15 as well as Figure 16, which will obscure the detection target, thereby interfering with the OBB task.

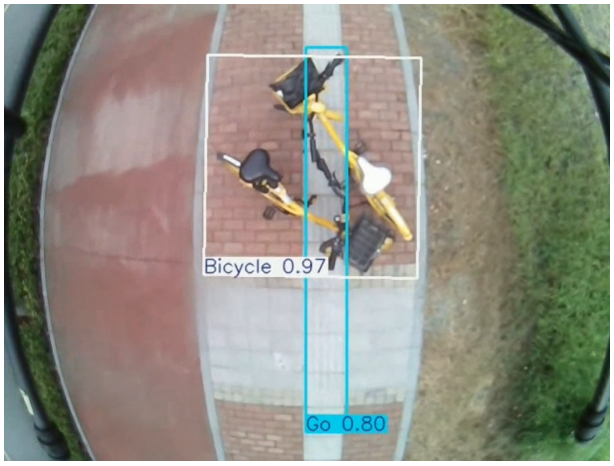


Figure 15. Small and medium-sized occupancy.



Figure 16. Large-scale occupancy.

The large object in Figure 16 will especially lead to the complete loss of the detection target; at this time, the drone will keep the flight heading before the target is lost and continue to fly until the detection target appears in the field of view again.

4.4.4. Controller Performance Validation

Figure 17 shows the actual and target speeds of the drone. The crimson trajectory illustrates the desired velocity parameters, whereas the sapphire pathway reflects real-time kinematic measurements during aerial operation. The figure clearly shows that the actual speed of the drone converges quickly and smoothly to the target speed, highlighting the stability of its flight performance.

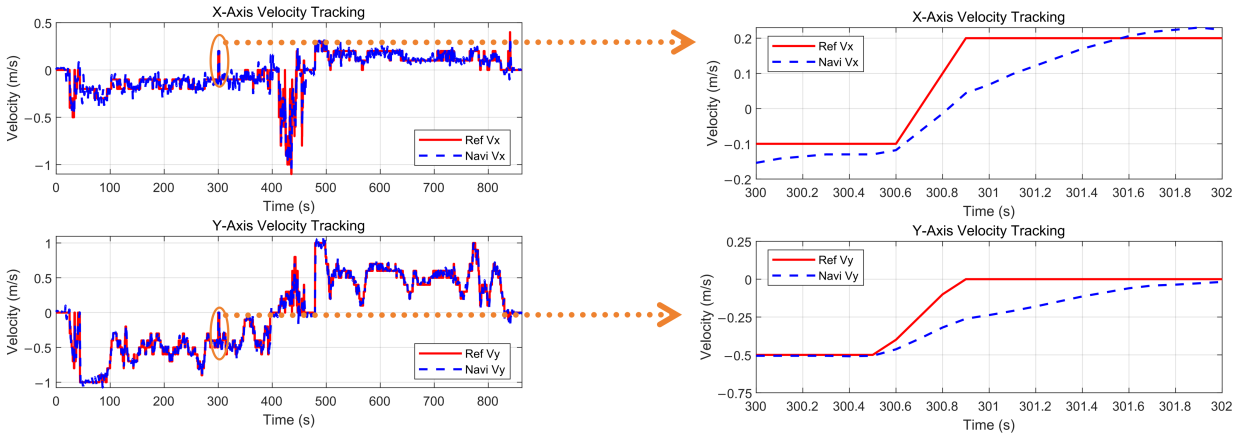


Figure 17. Velocity tracking validation.

The right side of Figure 17 illustrates the system's response during target loss, as shown in Figure 16. At this point the drone will maintain its current heading and proceed at a constant speed until the target is detected again.

5. Discussion and Conclusions

This paper presents an autonomous drone inspection system for efficient tactile paving detection that delivers four key contributions:

(1) Real-time detection capability through lightweight YOLOv8-OBB integration with fisheye cameras to compensate for limited low-altitude field-of-view;
(2) A novel LSDE-OBB detection head reducing model complexity via cross-scale parameter sharing, exploiting the consistent dot/line patterns of tactile pavings for shared feature extraction without accuracy degradation;
(3) Optimized speed-precision tradeoff achieving 25.13% parameter reduction (2.308 M vs. 3.083 M), 46.29% lower GFLOPs, and 11.97% $mAP_{50:95}$ enabling edge deployment;
(4) An efficient scalable framework integrating hierarchical control algorithms for dynamic flight adjustment.

The system is cross-validated on both public and custom datasets, demonstrating model robustness, with real-flight experiments confirming operational feasibility. By combining the C2f-Star Module and CAA Attention Module, computational costs are reduced while maintaining detection precision. Fisheye-equipped UAVs successfully execute robust tactile paving detection and obstacle avoidance in complex urban environments, significantly improving inspection efficiency. This framework enhances pedestrian safety and advances intelligent urban infrastructure management.

While the proposed system demonstrates effective detection of standard tactile paving patterns under diverse operational conditions, several avenues for enhancement merit attention to further broaden its applicability and robustness:

Robustness to Pavement Degradation and Variation. The current system's performance, optimized for typical intact paving blocks in standard orientations, may be susceptible to challenges posed by significant structural damage (e.g., severe cracking, fragmenta-

tion) or substantial surface wear (e.g., heavy fading, erosion). Furthermore, the impact of atypical block orientations (e.g., significant rotations deviating from cardinal directions) warrants dedicated investigation. Future research will prioritize the creation of datasets encompassing a wide spectrum of pavement degradation states and geometric variations. This data will be crucial for developing and training next-generation detection models with enhanced resilience to these real-world complexities. Understanding the precise interplay between the system's attention mechanisms (e.g., CAA) and the contextual regularity of paving arrays, especially when this regularity is disrupted by damage or misalignment, is another important area for deeper analysis.

Flight Control Integration Clarification. For clarity regarding the implementation, the core low-level flight stabilization and control relied on a well-established, commercially proven flight controller platform accessed via its Software Development Kit (SDK). The primary algorithmic contribution of this work resides in the high-level perception and navigation layer, implemented on an onboard companion computer. This layer processes the LSDE-OBB detections, performs obstacle avoidance computations, and generates high-level navigation commands sent to the underlying flight controller. Future deployment strategies will focus on seamless integration of this perception and navigation layer with commercial off-the-shelf drone platforms utilizing companion computers.

Enhanced Experimental Reporting. To improve the reproducibility and clarity of experimental findings, future presentations of this work will include more detailed visual documentation of the test environments (e.g., diagrams, photographs illustrating lighting, surfaces, backgrounds) and provide richer annotations on path execution figures to explicitly highlight key elements like detected paving regions, obstacles, planned paths, and actual flown trajectories. Explicit reporting of environmental condition parameters (e.g., lighting levels) during testing will also be strengthened.

Building upon the current foundation, future efforts will concentrate on the following:

(1) Developing adaptive detection models specifically trained to handle the spectrum of pavement degradation and geometric variations encountered in practical settings.
(2) Implementing cloud-based analytics for comprehensive defect assessment and maintenance prioritization.
(3) Advancing the obstacle negotiation and path planning logic to handle more intricate urban geometries and prolonged occlusion scenarios.

These advancements aim to solidify the system's operational reliability and extend its utility across a wider range of urban infrastructure inspection tasks.

Author Contributions: Conceptualization, T.W. and A.A.J.; methodology, T.W.; software, T.W.; validation, T.W. and H.W.; formal analysis, Z.Z.; investigation, T.W.; resources, H.W.; writing—original draft preparation, T.W.; writing—review and editing, S.S. and W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is based on results obtained from a project, JPNP22002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Wei Wang was employed by the company Fukushima Institute for Research, Education and Innovation (F-REI). The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Algorithm LSDE-OBB

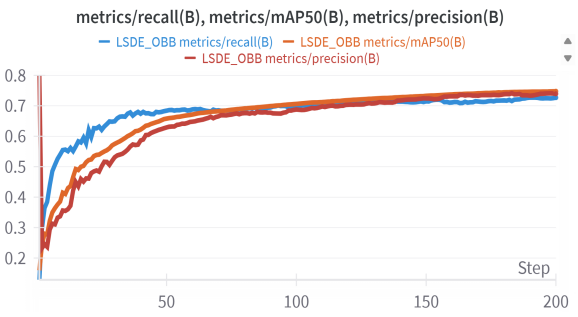
Algorithm A1 Lightweight Shared Detail Enhanced - OBB Head

Require: Feature maps $F_1, \ldots, F_L$; number of classes N_c; angle bins R; extra params N_e
Ensure: Rotated detections (x, y, w, h, θ) with class scores

for $i \leftarrow 1$ **to** L **do**
    $T_i \leftarrow \text{CONV+GN}(F_i)$
end for
$SDE \leftarrow$ two stacked detail-enhancement blocks
$cls_conv \leftarrow 1 \times 1$ convolution $\rightarrow N_c$
$reg_conv \leftarrow 1 \times 1$ convolution $\rightarrow 4R$
$ang_conv_i \leftarrow 1 \times 1$ Conv + detail-enhance + 1×1 Conv

function FORWARD($F_1, \ldots, F_L$)
    for $i \leftarrow 1$ **to** L **do**
        $T_i \leftarrow SDE(T_i)$
        $reg_i \leftarrow scale_i \times reg_conv(T_i)$
        $cls_i \leftarrow cls_conv(T_i)$
        $out_i \leftarrow \text{concat}(reg_i, cls_i)$
        $\theta_i \leftarrow (\text{sigmoid}(ang_conv_i(F_i)) - 0.25) \times \pi$
    end for
    if training **then**
        return $\{out_i\}, \{\theta_i\}$
    end if
    $P \leftarrow$ concat all out_i over spatial dims
    $(reg, cls) \leftarrow \text{split}(P, [4R, N_c])$
    $offs \leftarrow \text{DFL}(reg)$
    $boxes \leftarrow \text{dist2bbox}(offs, \text{anchors}, s)$
    $obb \leftarrow \text{dist2rbox}(offs, \theta, \text{anchors}, s)$
    $scores \leftarrow \text{sigmoid}(cls)$
    return $\text{concat}(obb, scores)$
end function

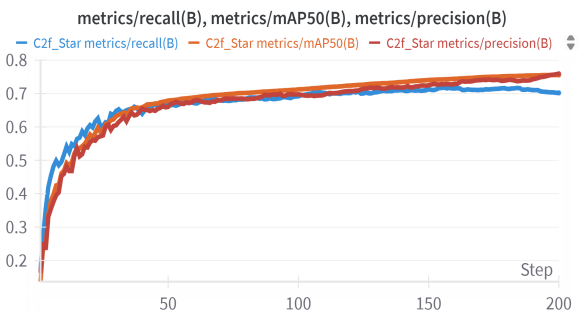
Appendix B. Runtime Execution Results



YOLOv8-LSDE-obb summary: 267 layers, 2597323 parameters, 2597307 gradients, 5.4 GFLOPs

Layer	Op	Parameters
L0	Conv	464
L1	Conv	4,672
L2	C2f	7,360
L3	Conv	18,560
L4	C2f	49,664
L5	Conv	73,984
L6	C2f	197,632
L7	Conv	295,424
L8	C2f	460,288
L9	SPPF	164,608
L10	Upsample	0
L11	Concat	0
L12	C2f	148,224
L13	Upsample	0
L14	Concat	0
L15	C2f	37,248
L16	Conv	36,992
L17	Concat	0
L18	C2f	123,648
L19	Conv	147,712
L20	Concat	0
L21	C2f	493,056
L22	LSDE_OBB	337,787

Figure A1. LSDE-OBB-Runtime.




```
YOLOv8-C2f-Star summary: 292 layers, 2609906 parameters, 2609890 gradients, 7.2 GFLOPs
Layer                  Op                     Parameters
 L0                        Conv                    464
 L1                        Conv                  4,672
 L2                    C2f_Star                  6,736
 L3                        Conv                 18,560
 L4                    C2f_Star                 23,200
 L5                        Conv                 73,984
 L6                    C2f_Star                 85,312
 L7                        Conv                295,424
 L8                    C2f_Star                326,272
 L9                        SPPF                164,608
L10                   Upsample                     0
L11                     Concat                     0
L12                   C2f_Star                118,080
L13                   Upsample                     0
L14                     Concat                     0
L15                   C2f_Star                 31,392
L16                       Conv                 36,992
L17                     Concat                     0
L18                   C2f_Star                 93,504
L19                       Conv                147,712
L20                     Concat                     0
L21                   C2f_Star                359,040
L22                        OBB                823,954
```

Figure A2. C2f-Star-Runtime.

Appendix C. List of Abbreviations

Abbreviation	Full Form	Short Definition
CAA	Contextual Anchor Attention	An attention mechanism using contextual anchors
DEConv	Detail-Enhanced Convolution	A convolution module designed to enhance details
DFL	Distribution Focal Loss	A loss function that balances class distribution
FOV	Field of View	The observable area captured by a sensor or camera
GN	Group Normalization	A normalization technique that operates in groups
GPS	Global Positioning System	A satellite-based navigation system
IMU	Inertial Measurement Unit	A device measuring acceleration and angular rates
LSDE	Lightweight Shared Detail Enhanced	A lightweight module for shared detail enhancement
LTE	Long-Term Evolution Module	A module for use in wireless communications
MAG	Magnetometer	A sensor measuring magnetic field strength
OBB	Oriented Bounding Box	A rotated rectangle used for detecting tilted objects
PID	Proportional–Integral–Derivative	A classical control algorithm with three terms
YOLOv8	You Only Look Once version 8	A real-time object detection deep learning model

Appendix D. Evaluation Indicators

Model Accuracy	Definition and Significance
Precision (P)	Reflects the reliability of model predictions, representing the ratio of correctly detected targets to all targets detected. For instance, a p value of 90% indicates that 9 out of 10 alarms correspond to actual tactile paving. Equation (8)
Recall (R)	Measures the model's coverage of true targets, defined as the ratio of correctly detected targets to the total number of actual targets. An R value of 85% means the system can identify 85% of tactile paving, with the remaining 15% potentially missed. Equation (9)
mAP	Provides a comprehensive evaluation of multi-category detection performance by averaging the area under the PR curve (AP) for all categories. This metric balances P and R, avoiding the bias of single-indicator assessments. Equation (10)

Model Accuracy	Definition and Significance
mAP_{50}	Calculates mean AP with an IoU threshold of 0.5, prioritizing classification accuracy over strict localization (detections with $\geq$50% overlap are considered correct).
$mAP_{50:95}$	Averages mAP across IoU thresholds from 0.5 to 0.95 (in 0.05 increments), evaluating both classification and precise localization robustness.
IoU Threshold (0.7)	The Intersection over Union (IoU) measures the overlap between predicted and ground truth bounding boxes. While conventional detection tasks typically use an IoU threshold of 0.5 (50% overlap), tactile paving inspection requires precise localization to ensure navigation safety. This study increases the threshold to 0.7 (70% overlap), demanding near-perfect alignment between detected and actual tactile paving. Although this stricter criterion may increase false negatives (e.g., partially occluded targets being filtered out), the integration of attention mechanisms and feature fusion algorithms maintains high recall while significantly reducing false positives. See Equation (11).
Model Efficiency	
Parameters (Par)	Represents the "size" of the model, directly affecting memory usage and storage requirements. For example, the improved YOLOv8-OBB reduces parameters by 25%, resulting in a smaller model file that is more suitable for deployment on edge devices like the Jetson AGX Xavier.
GFLOPs	Quantifies the floating-point operations required for a single inference (billions of operations per second), reflecting the computational power demand. A 47% reduction in GFLOPs indicates significant optimization, enhancing drone real-time performance.

References

1. Blindness and Vision Impairment. Available online: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed on 19 December 2024).
2. Bourne, R.; Steinmetz, J.D.; Flaxman, S.; Briant, P.S.; Taylor, H.R.; Resnikoff, S.; Casson, R.J.; Abdoli, A.; Abu-Gharbieh, E.; Afshin, A.; et al. Trends in prevalence of blindness and distance and near vision impairment over 30 years: An analysis for the Global Burden of Disease Study. *Lancet Glob. Health* **2021**, *9*, e130–e143.
3. *ISO 23599:2019*; Guidance on the Design of Tactile Walking Surface Indicators. International Organization for Standardization (ISO): Geneva, Switzerland, 2019.
4. *ISO 21542:2021*; Accessibility and Usability of the Built Environment. International Organization for Standardization (ISO): Geneva, Switzerland, 2021.
5. Tennøy, A.; Øksenholt, K.V.; Fearnley, N.; Matthews, B. Standards for usable and safe environments for sight impaired. *Proc. Inst. Civ. Eng. Munic. Eng.* 2015, *168*, 24–31.
6. Kim, S.Y.; Kwon, D.Y.; Jang, A.; Ju, Y.K.; Lee, J.S.; Hong, S. A review of UAV integration in forensic civil engineering: From sensor technologies to geotechnical, structural and water infrastructure applications. *Measurement* **2024**, *224*, 113886.
7. Wen, F.; Pray, J.; McSweeney, K.; Gu, H. Emerging inspection technologies—Enabling remote surveys/inspections. In Proceedings of the Offshore Technology Conference, Houston, TX, USA, 6–9 May 2019.
8. Geng, H.; Liu, Z.; Jiang, J.; Fan, Z.; Li, J.X. Embedded road crack detection algorithm based on improved YOLOv8. *J. Comput. Appl.* **2024**, *44*, 1613–1618.
9. Congress, S.S.C.; Escamilla, J., III; Chimauriya, H.; Puppala, A.J. Eye in the Sky: 360° Inspection of Bridge Infrastructure Using Uncrewed Aerial Vehicles (UAVs). *Transp. Res. Rec.* **2024**, *2678*, 482–504.
10. Zhou, L.; Deng, X.; Wang, X.; Li, T.; Yi, L.; Xiong, X.; Tolba, A.; Ning, Z. Data intelligence for UAV-assisted road inspection in post-disaster scenarios. *IEEE Internet Things J.* **2024**, *early access*. [CrossRef]
11. Geng, H.; Liu, Z.; Wang, Y.; Fang, L. SDFC-YOLO: A YOLO-based model with selective dynamic feature compensation for pavement distress detection. *IEEE Trans. Intell. Transp. Syst.* **2024**, *early access*. [CrossRef]
12. Meng, S.; Gao, Z.; Zhou, Y.; He, B.; Djerrad, A. Real-time automatic crack detection method based on drone. *Comput.-Aided Civ. Infrastruct. Eng.* **2023**, *38*, 849–872.

13. Imad, M.; Wicaksono, M.; Shin, S.Y. Autonomous UAV Navigation and Real-Time Crack Detection for Infrastructure Inspection in GPS-Denied Environments. In Proceedings of the 2024 15th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 16–18 October 2024; pp. 1210–1213. [CrossRef]
14. Wang, Q.; Wang, W.; Li, Z.; Namiki, A.; Suzuki, S. Close-range transmission line inspection method for low-cost uav: Design and implementation. *Remote Sens.* **2023**, *15*, 4841. [CrossRef]
15. Ragnoli, A.; De Blasiis, M.R.; Di Benedetto, A. Pavement distress detection methods: A review. *Infrastructures* **2018**, *3*, 58. [CrossRef]
16. Shao, Y.; Xu, X. Three-Dimensional Trajectory Design for Post-disaster UAV Video Inspection. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–6. [CrossRef]
17. Maluya, M.M.O.; Aleluya, E.R.M.; Opon, J.G.; Salaan, C.J.O. Lane Tracking for Autonomous Road Pavement Inspection with Unmanned Aerial Vehicles. In Proceedings of the 2023 6th International Conference on Applied Computational Intelligence in Information Systems (ACIIS), Bandar Seri Begawan, Brunei, 23–25 October 2023; pp. 1–6.
18. Zhu, J.; Wu, Y.; Ma, T. Multi-Object Detection for Daily Road Maintenance Inspection with UAV Based on Improved YOLOv8. *IEEE Trans. Intell. Transp. Syst.* 2024, *25*, 16548–16560. [CrossRef]
19. Chen, X.; Wang, C.; Liu, C.; Zhu, X.; Zhang, Y.; Luo, T.; Zhang, J. Autonomous crack detection for mountainous roads using UAV inspection system. *Sensors* **2024**, *24*, 4751. [CrossRef]
20. Fan, R.; Jiao, J.; Pan, J.; Huang, H.; Shen, S.; Liu, M. Real-time dense stereo embedded in a uav for road inspection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
21. Hassan, S.A.; Rahim, T.; Shin, S.Y. An improved deep convolutional neural network-based autonomous road inspection scheme using unmanned aerial vehicles. *Electronics* **2021**, *10*, 2764. [CrossRef]
22. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* **2023**, *147*, 104745. [CrossRef]
23. Wessendorp, N.; Dinaux, R.; Dupeyroux, J.; de Croon, G.C.H.E. Obstacle avoidance onboard MAVs using a FMCW radar. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 117–122.
24. Yu, H.; Zhang, F.; Huang, P.; Wang, C.; Yuanhao, L. Autonomous obstacle avoidance for UAV based on fusion of radar and monocular camera. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 5954–5961.
25. Wu, Y.; He, K. Group normalization. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
26. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: A simple and strong anchor-free object detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1922–1933. [CrossRef]
27. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015. [CrossRef]
28. Shen, Q.; Zhang, L.; Zhang, Y.; Li, Y.; Liu, S.; Xu, Y. Distracted Driving Behavior Detection Algorithm Based on Lightweight StarDL-YOLO. *Electronics* **2024**, *13*, 3216. [CrossRef]
29. Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; Fu, Y. Rewrite the Stars. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 5694–5703.
30. Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; Yao, Y. Poly kernel inception network for remote sensing detection. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 27706–27716.
31. Xie, H.; Yuan, B.; Hu, C.; Gao, Y.; Wang, F.; Wang, Y.; Wang, C.; Chu, P. SMLS-YOLO: An extremely lightweight pathological myopia instance segmentation method. *Front. Neurosci.* **2024**, *18*, 1471089. [CrossRef]
32. Sun, Y.; Cao, B.; Zhu, P.; Hu, Q. Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6700–6713. [CrossRef]
33. Zhang, J.; Chen, Z.; Yan, G.; Wang, Y.; Hu, B. Faster and Lightweight: An Improved YOLOv5 Object Detector for Remote Sensing Images. *Remote Sens.* **2023**, *15*, 4974. [CrossRef]
34. Feng, C.; Zhong, Y.; Gao, Y.; Matthew, R.S.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3490–3499.
35. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975.

36. Ding, X.; Zhang, X.; Han, J.; Ding, G. Diverse branch block: Building a convolution as an inception-like unit. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10886–10895.
37. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.
38. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
39. Wan, D.; Lu, R.; Shen, S.; Xu, T.; Lang, X.; Ren, Z. Mixed local channel attention for object detection. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106442. [CrossRef]
40. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. SimAM: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the 38th International Conference on Machine Learning, PMLR, Virtual Event, 18–24 July 2021; pp. 11863–11874.
41. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. BiFormer: Vision transformer with bi-level routing attention. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 10323–10333.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.