drones

MDPI

Article

PISCFF-LNet: A Method for Autonomous Flight of UAVs Based on Lightweight Road Extraction

Yuanxu Zhu [1,2,†], Tianze Zhang [2,3,†], Aiying Wu [1,2] and Gang Shi [1,2,*]

[1] College of Computer Science and Technology, Xinjiang University, Urumqi 830046, China; zhuyx@stu.xju.edu.cn (Y.Z.); 107552201402@stu.xju.edu.cn (A.W.)
[2] Xinjiang Key Laboratory of Signal Detection and Processing, Urumqi 830046, China; zhangtianze.unimelb@gmail.com
[3] Faculty of Science,The University of Melbourne, Melbourne, VIC 3010, Australia
* Correspondence: shigang@xju.edu.cn
† These authors contributed equally to this work.

Abstract: Currently, autonomous flight control for unmanned aerial vehicles (UAVs) has become increasingly critical in remote-sensing applications, such as high-resolution data acquisition and road disease detection. However, this task also faces significant challenges, particularly the weak GNSS signals in flight areas and the complex flight environment. Furthermore, many existing autonomous-flight-control algorithms for UAVs are computationally demanding, which limits their deployment on embedded devices with constrained memory and processing power, thereby affecting both operational efficiency and the safety of UAV missions. To address these issues, we propose PISCFF-LNet, a lightweight road-extraction network that integrates prior knowledge and spatial contextual features. The network employs a dual-branch encoder architecture to separately extract spatial and contextual features, thus obtaining multi-dimensional feature representations. In addition, to enhance the integration of different features and improve the overall feature representation, we also introduce a feature-fusion module. To further enhance UAV performance, we introduce an improved ray-based eight neighborhood algorithm (RENA), which efficiently extracts road-edge information with a remarkably low latency of just 7 ms, providing accurate flight guidance and reducing misidentification. To provide a comprehensive evaluation of the model's performance, we have developed a new drone remote-sensing road-semantic-segmentation dataset, DRS Road, which includes approximately 2600 ultra-high-resolution remote-sensing images across six scene categories. The experimental results demonstrate that PISCFF-LNet achieves improvements of 1.06% in Intersection over Union (IoU) and 0.83% in F1-Score on the DeepGlobe Road dataset, and 1.03% in IoU and 0.57% in F1-Score on the DRS Road dataset, compared to existing methods. Finally, we applied the algorithm to a UAV, using a PID-based flight-control algorithm. The results show that drones employing our algorithm exhibit superior flight performance in both simulated and real-world environments.

Keywords: semantic segmentation; remote sensing; UAV; deep learning; road extraction

Academic Editors: Higinio González Jorge, Fernando Veiga López, Enrique Aldao Pensado and Gabriel Fontenla-Carrera

Received: 4 February 2025
Revised: 2 March 2025
Accepted: 12 March 2025
Published: 20 March 2025

Citation: Zhu, Y.; Zhang, T.; Wu, A.; Shi, G. PISCFF-LNet: A Method for Autonomous Flight of UAVs Based on Lightweight Road Extraction. *Drones* **2025**, *9*, 226. https://doi.org/10.3390/ drones9030226

1. Introduction

1.1. Overview

Unmanned aerial vehicles (UAVs), also known as drones, are intelligent flying machines driven by power and remotely controlled or capable of autonomous flight [1]. Due to their light weight, flexibility, and high integration, UAVs have shown immense

application potential in fields such as remote sensing, agriculture, and transportation. However, as mission complexity continues to rise, the demand for intelligent and autonomous UAVs has become increasingly urgent. Autonomous flight control, particularly in complex environments, has become a key focus for both academia and industry.

Currently, research on autonomous flight is mainly focused on the improvement and optimization of navigation systems, which are typically categorized into four major approaches: GNSS/INS navigation systems, LiDAR, visual sensors, and multi-sensor fusion methods [2]. Among them, UAV navigation based on visual sensors not only extends the scope of UAV applications but also significantly enhances their adaptability in complex environments [3]. Compared to traditional GNSS or LiDAR systems, visual sensors offer advantages such as light weight, low cost, and low power consumption, and they can provide rich environmental information, especially in structured and semi-structured environments, enabling real-time perception and navigation. Additionally, visual sensors can be combined with other sensors (e.g., IMUs) [4] to achieve higher-precision positioning and state estimation. Therefore, vision-based navigation methods have unique application potential in indoor environments, low-light conditions, and scenarios where GNSS signals are insufficient.

Vision-based navigation methods are based on machine learning, which can be further divided into traditional machine learning methods and deep learning methods. Traditional machine learning methods were widely used in early UAV navigation research. These methods typically rely on feature extraction and manually designed algorithms. For example, traditional machine learning algorithms such as Support Vector Machines (SVMs) [5], Random Forests (RFs) [6], and k-Nearest Neighbors (k-NNs) [7] achieve UAV localization and path planning by extracting features and classifying images or sensor data. These methods are computationally efficient and can perform well in simpler tasks, such as navigation and obstacle avoidance. However, they have significant limitations. First, they often depend on manually designed features, and the feature-extraction process relies heavily on expert experience, making it both cumbersome and poorly adaptive to complex scenarios. Second, traditional methods are prone to overfitting when handling large-scale data and tend to perform poorly in dynamic and complex environments, especially when the environment changes rapidly or tasks become diversified, making it difficult to maintain efficiency and accuracy.

Deep learning methods, a major breakthrough in artificial intelligence in recent years, have overcome many of the limitations of traditional machine learning methods. In UAV visual navigation, deep learning, particularly convolutional neural networks (CNNs) [8], learns high-level features from data automatically, significantly improving the UAV's ability to autonomously navigate in complex environments. Deep learning can extract rich features from large amounts of training data, offering stronger adaptability and robustness, especially in dynamic and changing environments. It can effectively deal with challenges such as occlusion and illumination changes. Among them, road extraction, i.e., road semantic segmentation, has become one of the technologies used in UAV autonomous flight.

Semantic segmentation aims to classify each pixel in an image into a specific category, enabling the precise identification and separation of different environmental elements. Early semantic segmentation methods, such as fully convolutional networks (FCNs) [9], applied fully convolutional networks to pixel-level classification, achieving basic semantic segmentation. These methods usually have lower computational complexity and are suitable for simpler scenarios, but they have certain limitations in multi-scale feature extraction and precise boundary processing. In recent years, improvements in deep network architectures have led to significant progress in semantic segmentation tasks. For example, networks like U-Net have enhanced segmentation accuracy through an encoder–decoder structure, while

introducing skip connections to preserve high-resolution features. Furthermore, methods based on deep residual networks (ResNet) [10] and attention mechanisms [11] can improve segmentation accuracy and robustness by extracting deeper features and performing more refined pixel-level classification.

In traditional U-Net networks [12], each layer processes images through an encoder–decoder structure while using skip connections to preserve low-level detail features. This design enables the network to effectively fuse high-level semantic information and low-level detail features when recovering image resolution, thus improving semantic segmentation accuracy. However, this structure also has some shortcomings, particularly when dealing with fine boundaries and multi-scale features, where U-Net often struggles to achieve optimal fusion, leading to blurred boundaries or the inaccurate segmentation of small objects. To address these issues, researchers have proposed several improvements to enhance U-Net's performance. For example, Li et al. proposed U-Net++, which introduces denser skip connections and multi-scale fusion modules to enable the more effective interaction and fusion of features between different layers, thus improving the recovery of complex boundaries. Zhang et al. proposed Attention U-Net, which introduces an attention mechanism in skip connections, allowing the network to dynamically select and focus on important regions of the image, thus reducing background interference while improving segmentation accuracy. Models such as DeepLabV3, based on dilated convolutions, extract features at multiple scales and perform post-processing using Conditional Random Fields (CRFs), allowing the model to better handle small object segmentation in large-scale scenes [13]. However, these methods involve significant computational overhead, making them challenging to apply in real-time scenarios. Therefore, designing a road-semantic-segmentation network that balances accuracy and real-time performance is crucial for UAV autonomous flight.

To solve these issues, this paper proposes a UAV autonomous cruising algorithm based on road extraction, utilizing a lightweight semantic segmentation network PISCFF-LNet to extract road information and generate navigation instructions. The network is based on the lightweight architecture UNeXt [14], which incorporates a dual-branch encoder and feature-fusion module, effectively addressing the multi-scale feature-fusion problem. Additionally, the ray-based octagonal neighborhood algorithm is introduced to quickly extract road edges, improving both the accuracy and real-time performance of road semantic segmentation. Moreover, this study constructs a UAV road-semantic-segmentation dataset containing 2600 ultra-high-resolution images to provide high-quality support for model training and evaluation.

The main contributions of this paper are as follows:

1. A lightweight network based on prior-information assistance and context feature fusion is proposed, PISCFF-LNet, significantly reducing model parameters and latency, making it suitable for deployment on edge devices;
2. A feature-fusion module is designed to integrate shallow and deep encoder features, enhancing the network's ability to handle multi-scale information;
3. A vision-assisted UAV autonomous-flight-control method is proposed, RENA, optimizing road-edge extraction with the ray-based octagonal neighborhood algorithm to achieve basic terrain-following flight;
4. A high-resolution UAV road-semantic-segmentation dataset is constructed, DRS Road, providing standardized data support for related research.

1.2. Autonomous Flight of UAVs

The GNSS/INS navigation system-based approach was a major focus of early research on UAV autonomous flight. GNSS provides basic navigation support for flight-control

platforms by acquiring real-time position information of the UAV. However, GNSS signals are susceptible to environmental interference, such as multipath effects in urban areas or canyons and signal loss in enclosed spaces. Therefore, enhancing or improving GNSS signals has become a research direction. For example, Yun et al. [15] proposed an enhanced scheme for receiving GNSS signals in UAV systems, using Kalman filtering combined with a complementary filter to regenerate smooth and accurate signals, improving the UAV's positioning ability in dynamic environments, thus enabling autonomous flight. Additionally, researchers have integrated inertial navigation systems (INSs) or inertial measurement units (IMUs) with GNSS to improve navigation accuracy by compensating for accumulated errors. A Nemra et al. [16] proposed a new GNSS/INS sensor fusion scheme based on the state-dependent Riccati equation (SDRE) nonlinear filter, which reduces the linearization error of the extended Kalman filter (EKF) and enhances the UAV's positioning performance, enabling autonomous flight with the help of maps. Although such methods perform well in open environments, they still face significant limitations in scenarios with weak or no GNSS signals.

To overcome the shortcomings of navigation systems in information acquisition, some scholars have proposed using LiDAR to provide position and attitude estimation support for UAVs by constructing high-precision 3D maps. LiDAR has the advantages of high measurement accuracy and strong anti-interference ability, enabling precise navigation in weak GNSS environments. Tao Yang et al. [17] proposed a SLAM method combining three-point features and median filtering to remove noise, constructing grid maps at different heights to allow the UAV to carry out autonomous route planning. Ziyi Qiu et al. [18] proposed a LiDAR navigation system based on global ArUco, using LiDAR, IMU, and global ArUco information to calculate the UAV's pose in the real coordinate system, achieving precise flight navigation.

In addition, multi-sensor fusion methods have been proposed, combining LiDAR and visual sensor information to provide spatial position data for navigation. Bachrach A. et al. [19] proposed a data fusion method based on LiDAR, IMU, and monocular cameras to enable UAVs to perform obstacle avoidance and navigation in both indoor and outdoor environments. P. Sakthivel et al. proposed a vision-based obstacle-size-estimation algorithm and distance estimation using LiDAR for autonomous UAV navigation, achieving real-time obstacle avoidance flight with an error of 0.01 m [20].

Although the two LiDAR-based autonomous flight methods mentioned earlier have high accuracy, they require substantial computational power, impose high hardware performance requirements, and involve expensive equipment. Additionally, the complex modeling and computations typically rely on high-performance ground stations, limiting their application on small multirotor UAVs.

Vision-based UAV navigation methods are widely used in the industry due to the lightweight, low-cost, and anti-interference properties of sensors. With the rapid development of deep learning, the precise control of UAVs with a single visual sensor for autonomous flight has become feasible. Early visual navigation methods used optical flow sensors. Nils Gageik et al. [21] proposed a UAV navigation method combining optical flow, inertial, ultrasonic, and infrared sensors, using optical flow for 2D positioning while other sensors assist with obstacle avoidance and error reduction, achieving autonomous flight without external reference systems. With the rapid advancement of deep learning, attention has shifted to autonomous flight methods using visual information combined with deep learning algorithms. Arshad et al. [22] proposed a data-driven strategy based on deep convolutional neural networks for UAV navigation in complex dynamic environments. Yumin Zhao et al. [23] proposed a deep learning-based autonomous UAV exploration method (DLAE), which combines position and yaw actions to overcome field-of-view limitations

and designs an autoregressive network model to improve the UAV's exploration efficiency and decision-making time. The methods mentioned above are primarily applied to obstacle avoidance tasks in indoor environments and are not suitable for autonomous drone flight and remote-sensing data collection tasks in outdoor scenarios.

To address this issue, this study proposes a drone navigation method based on deep learning and visual navigation techniques. The method introduces a lightweight road-extraction network, PISCFF-LNet, and the RENA algorithm. Compared to methods such as LiDAR and multi-sensor fusion, this approach requires significantly lower computational power, with a floating-point operation count of 5.38 G and only 2.31 million parameters. It can be deployed on edge devices like the Nvidia TX2, requires only visual sensors, is easy to deploy, and has lower costs, while still effectively guiding the UAVs' flights.

1.3. Road Extraction

Traditional image-processing methods for road extraction mainly rely on the geometric, texture, and spectral features of the road, using manually designed features, operators, and parameters based on empirical knowledge. However, in complex road scenes, these methods struggle to achieve high accuracy. In recent years, the advent of deep learning methods has led to breakthroughs in road extraction, making it a current research hotspot and trend.

Ren et al. [24] developed a capsule-based U-Net architecture, combining capsule representations with the advantages of attention mechanisms. This approach can extract and fuse multi-scale capsule features, resulting in high-resolution feature representations with rich semantic information. Li et al. [25] introduced a set of cascaded global-attention modules into the DenseUNet framework to extract contextual information of the road. Furthermore, a set of cascaded core attention modules was embedded to ensure the adequate transmission of road information between dense blocks, further assisting the global-attention module in acquiring multi-scale features, thus improving the connectivity of the road network.

In complex road backgrounds, the shadows and occlusions caused by surrounding trees and buildings have always been issues that need to be addressed. Zhou et al. [26] proposed the D-LinkNet34 network based on LinkNet, using dilated convolutions to expand the receptive field and capture multi-scale features, although this method still suffers from many misdetections and missed detections of roads. Zhou et al. [27] classified road scenes into edge-feature-dominated (EFD) roads and region-feature-dominated (RFD) roads based on the density of road directional lines in the image. The EFD and RFD sections are extracted using structural line grouping and the U-Net model, respectively. This method is somewhat resistant to interference from shadows and occlusions, and effectively handles roads under construction with incomplete spectral and geometric features.

Connectivity is a natural and critical feature of roads. Li et al. [28] implemented pixel-level, edge-level, and region-level sharing in the encoder section and enhanced the topological relationships through direction-aware processing, thus extracting roads with higher integrity and connectivity. Mei et al. [29] proposed a Connectivity Attention Network (CoANet), which learns pairwise dependencies. The model uses strip convolutions in four directions to consider road directionality, calculates connectivity loss through a connectivity cube, and thus strengthens the network's learning of road connectivity. This method can maintain high accuracy while preserving road connectivity. Tan et al. [30] proposed a road-extraction network based on bidirectional spatial information reasoning (BSIRNet), which captures spatial-context dependencies and inter-channel dependencies, effectively improving road-extraction accuracy and completeness. Most deep learning methods achieve high accuracy in road extraction, but the results still show significant road fractures and fragmentation.

The self-attention mechanism in the Transformer [11] has the ability to establish long-range dependencies, showing excellent performance in many natural language-processing tasks, and is increasingly gaining prominence in computer-vision tasks. Leveraging the Transformer structure to perceive global context relationships and geometric information of roads is crucial for improving road-segmentation accuracy. Chen et al. [31] proposed a dual-branch encoding block combining Swin Transformer and ResNet, and added context-guided filtering blocks in the skip connections to filter noise, better preserving local detail information and reducing the number of broken road segments. To overcome the performance limitations imposed by fixed patches in Transformer, Zhang et al. [32] proposed obtaining coarse-grained and fine-grained feature representations from different scales in the encoder section and integrating Transformers in the feature-fusion module to enhance information interaction between the two, effectively addressing some road discontinuities and maintaining the integrity of road-segmentation results. However, due to the computational demands of the self-attention mechanism, Transformer-based models have a clear disadvantage in inference speed compared to CNN-based models.

These methods achieve high road-extraction accuracy, but for real-time applications, model inference speed and computational efficiency must also be considered. In recent years, some scholars have focused on lightweight models to meet the requirements of real-time applications. Liu et al. [33] proposed a lightweight dynamic addition network (LDANet), which introduces an improved Inception structure based on asymmetric convolution blocks (ACBs) to expand low-level features in the feature-extraction layer. Additionally, depth-wise separable convolutions (DSC) are used to reduce model computational complexity, and an adaptive weighted summation module is designed to capture prominent road features. This method has fewer than 1 MB of parameters and a fast inference speed. Wang et al. [34] proposed using Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale features [35], employing attention mechanisms to solve road discontinuity and edge loss issues. This method has 2.85 MB of parameters, alleviating computational burden and reducing training time while maintaining good performance. Xie Guobo et al. [36] replaced the main encoder of DeepLabv3+ [37] with the lightweight MobileNetv2 [38] and introduced depth-wise separable convolutions in the channel-space parallel attention module to reduce the model's parameters. Qu et al. [39] designed Road-MobileFormer as the backbone network structure for Road-MobileSeg. In Road-MobileFormer, a coordinate attention module is introduced to encode channel relationships and long-range dependencies, along with precise location information, to improve road-extraction accuracy. Additionally, a micro-token pyramid module is introduced to reduce the number of model parameters and computations, making it more lightweight. This method enables high-precision and low-latency road extraction on mobile devices. These methods offer advantages in real-time performance and computational efficiency, but their segmentation accuracy is lower compared to larger models.

Deep learning-based road-extraction methods automatically learn road features to obtain semantic information, not only enabling automated and precise road information extraction but also handling large-scale road imagery datasets.

2. Materials and Methods

2.1. Dataset

Current publicly available remote-sensing datasets (such as DeepGlobe Road and others) generally have spatial resolutions lower than 0.5 m, which makes it difficult to meet the road feature analysis requirements for UAV low-altitude flight scenarios. Therefore, this study has constructed a road-extraction dataset from the UAV perspective, DRS Road, consisting of 2600 images. In order to better validate the effectiveness of the model proposed

in this study, a publicly available dataset, DeepGlobe Road, was also chosen for comparison, as it is as close as possible to the dataset used in this research.

2.1.1. DeepGlobal Road

The DeepGlobe Road dataset is a sub-dataset used in the DEEPGLOBE CVPR 2018 competition. It provides a large number of high-resolution sub-meter satellite images along with corresponding pixel-level label information. The images are sourced from three regions: Thailand, India, and Indonesia, encompassing various scenes such as urban, rural, suburban, coastal, and tropical rainforests [40]. The size of the images is 1024 × 1024, with a resolution of 50 cm.

To address the issue of uneven scene distribution in this dataset, this study proposes a hierarchical enhancement strategy. First, based on road density and surrounding landscape features, the samples are divided into two categories: urban roads (high density, regular geometric shapes) and rural roads (low density, complex background interference). Then, stratified sampling is used to divide the dataset into training and validation sets in a 7:3 ratio, ensuring a balanced representation of both scene types during the training process, as shown in Figure 1.

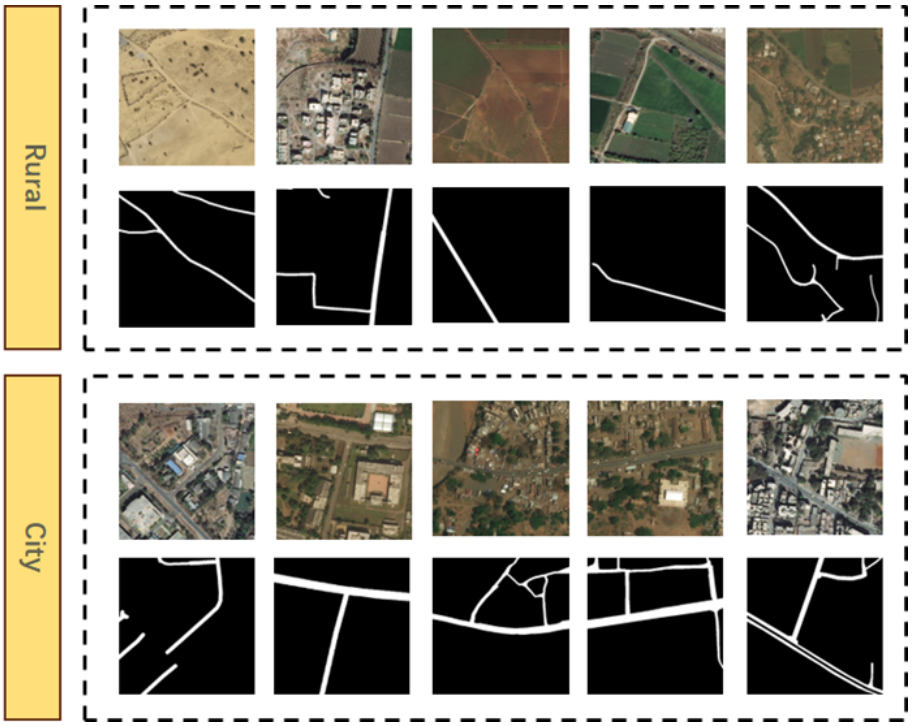

Figure 1. The processed DeepGlobe Road dataset. The dataset is divided into two parts: rural and city areas. The first and third rows show the original images of rural and city areas, respectively, while the second and fourth rows show the corresponding label maps for rural and city areas.

2.1.2. DRS Road

This dataset uses the DJI Zenmuse L1 Livox Lidar (DJI, Shenzhen, China) and the DJI Zenmuse P1 (DJI, Shenzhen, China) optical camera to collect LiDAR and optical image data.

The optical remote-sensing data are collected using the DJI MTK300 RTK (DJI, Shenzhen, China) drone equipped with the DJI Zenmuse P1 optical camera, which can clearly capture road and detail information. This data can not only be used for road extraction tasks but also for subsequent road disease-detection tasks. This part of the dataset contains 1800 images, each with a resolution of 8192 × 5460, as shown in Figure 2, and was collected under good lighting conditions.

The LiDAR remote-sensing data are collected using the DJI MTK300 RTK drone equipped with the DJI Zenmuse L1 Livox LiDAR camera. It can be used in all weather conditions, unaffected by weather, lighting, or ground cover, and provides accurate elevation data. However, the images captured by the LiDAR camera are in the form of 3D point clouds, requiring additional processing. The point-cloud data are reconstructed in three dimensions to generate orthophotos and elevation images. The road center points are extracted at 1 m intervals from the generated orthophotos, and the central line is fitted. Based on the central line, the rotation angle of each image is calculated, which can be derived from the slopes of the central line at the front and rear of the image center. Afterward, the image is cropped every 30 m and rotated based on the rotation angle, ultimately generating road-surface images with no overlapping areas and unified orientation. The overall process is shown in Figure 3. This part of the dataset contains 800 images, each with a resolution of 4320 × 3520, and was collected under good lighting conditions.

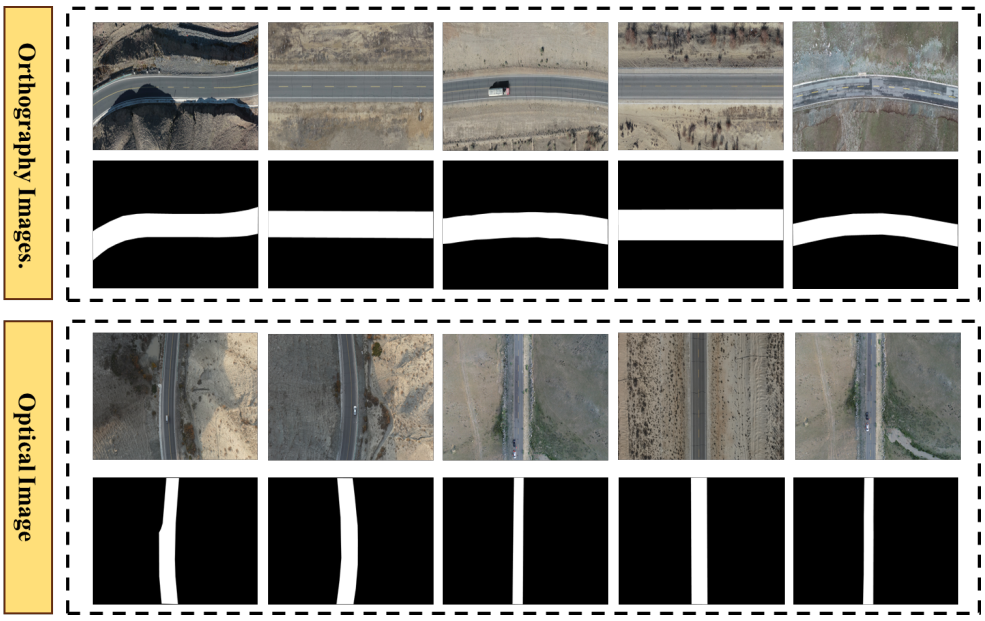

Figure 2. The self-built DRS ROAD dataset. This dataset consists of two parts: optical images and orthophoto images. The first and third rows show the original orthophoto images and optical images, respectively, while the second and fourth rows show the corresponding label maps for the orthophoto and optical images.

All data collected in this study cover a total of 228 km of road sections across six cities in Xinjiang, including road-surface images taken during spring, summer, and autumn. The dataset covers various geographical environments, including desert, mountain roads, urban roads, provincial roads, and national roads, meaning the dataset is characterized by the following features:

(1) Multi-season and multi-geographical environments: The data span different seasons and a variety of geographical environments (such as desert, mountainous areas, and urban roads), ensuring the representativeness and wide applicability of the dataset.

(2) High resolution and precise calibration: The use of high-resolution P1 optical cameras and LiDAR cameras ensures the accurate capture of image details, providing reliable data support for subsequent road extraction and analysis tasks.

Therefore, this dataset provides a foundational platform for road extraction, disease detection, and other related research using UAV remote-sensing imagery. Through meticulous image-processing and annotation methods, this study ensures the high quality and

reliability of the data, enabling effective support for subsequent deep learning model training and testing.

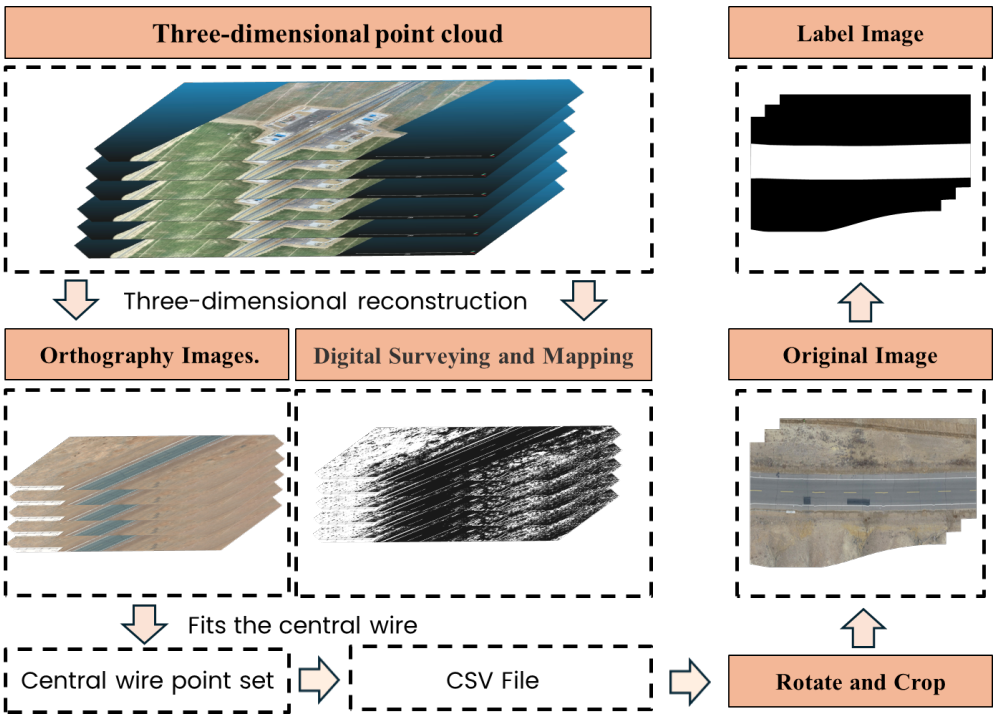

Figure 3. L1 three-dimensional point-cloud processing diagram.

2.2. Road-Extraction Network PISCFF-LNet

In order to efficiently perform road-extraction tasks in resource-constrained or real-time scenarios, this paper proposes a lightweight road-extraction network called PISCFF-LNet, based on Prior Information and Spatial-Context Feature Fusion, as shown in Figure 4.

The overall architecture of PISCFF-LNet is illustrated in the figure, with design inspiration derived from the UNeXt model in [14] and the DeepLabV3+ model in [36]. The model uses the lightweight semantic segmentation model UNeXt as the base architecture and incorporates a dual-branch structure along with a prior-information-assisted branch that utilizes binarized images. First, the original image is input into the prior-information-assisted branch for simple binarization, which is then used as prior information. This is concatenated with the original RGB image, along the channel dimension, to merge and serve as the input feature for the network model. This branch enriches the image features in an unsupervised manner, providing the model with additional visual cues to guide the learning of object locations and texture features.

Next, the encoder of UNeXt is employed as the context information branch (CIB) to quickly extract contextual features of the image. Additionally, a lightweight spatial information branch (SIB) is designed to extract rich spatial detail features. The fusion of features from DI-ARM, which ensures effective feature transfer.

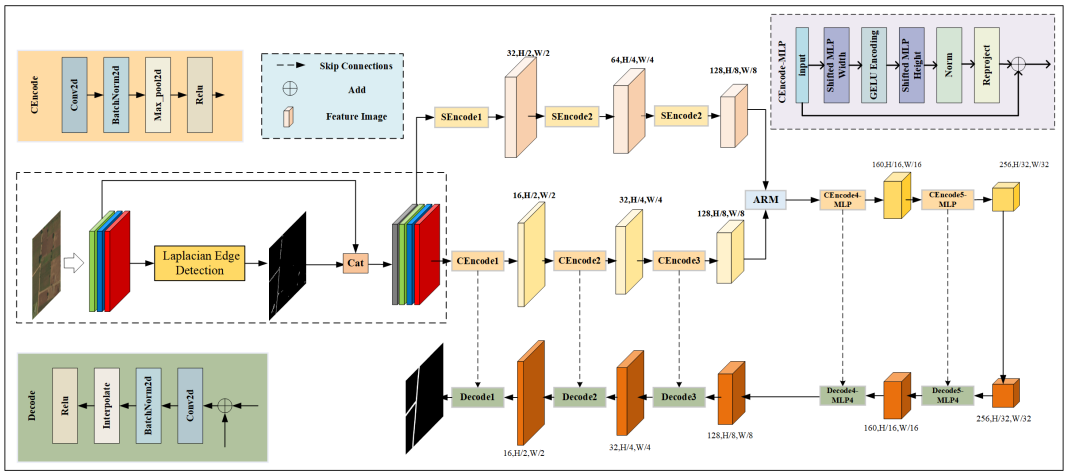

Figure 4. PISCFF-LNet network structure diagram. The network consists of three main components. The Prior Information module is used for edge feature extraction, providing additional road features. The Double-Branch Encoder module adopts a dual-branch structure to extract spatial and contextual information separately. The ARM module is responsible for fusing different features.

2.2.1. Prior-Information-Assisted Branch Based on Binarized Images

Considering that lightweight models generally have less feature-extraction capability compared to more complex deep networks, selecting appropriate prior information is crucial for model learning and understanding. Binarized images are simple and intuitive with low dimensionality, and they can provide additional spatial information to the model regarding the location of objects.

Therefore, this paper selects binarized images obtained through the Laplacian edge-detection operator as prior auxiliary information to help the lightweight model better learn and understand the image features without increasing the model's complexity. The Laplacian edge-detection operator is less sensitive to noise and less prone to discontinuous points. The operator is defined as follows:

$$Lap = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{1}$$

The convolution operation is performed on each pixel in the image using the Laplacian operator matrix, and the specific formula can be expressed as:

$$L(x,y) = \sum_{k} \sum_{l} I(x+k, y+l) \cdot \mathrm{Lap}(k,l) \tag{2}$$

where $I(x+k, y+l)$ represents the pixel value at the position $((x+k, y+l))$, $Lap(k,l)$ represents the value of the convolution kernel at position (k,l), and $L(x,y)$ represents the new pixel value at position (x,y). By convolving the image with the discrete form of the Laplacian convolution kernel, the second-order derivative of the image is approximated to capture edge and region-change information in the image. To smooth the extracted edges, this paper combines the Laplacian operator with Gaussian blur, which both suppresses noise and efficiently detects image edges:

$$LoG(x,y) = \nabla^2(G(x,y) * I(x,y)) \tag{3}$$

where $LoG(x,y)$ is the result of the Laplacian of Gaussian (LoG) operation, ∇^2 is the Laplacian operator, $G(x,y)$ is the Gaussian kernel used to smooth the image and reduce noise, $I(x,y)$ is the intensity of the image at the pixel location (x,y).

The extracted result, as shown in Figure 5, contains some full road information in the binarized image, which provides auxiliary support for the road-extraction task. This method makes the model more suitable for lightweight applications or resource-constrained scenarios while enhancing the model's accuracy and robustness in boundary feature recognition.

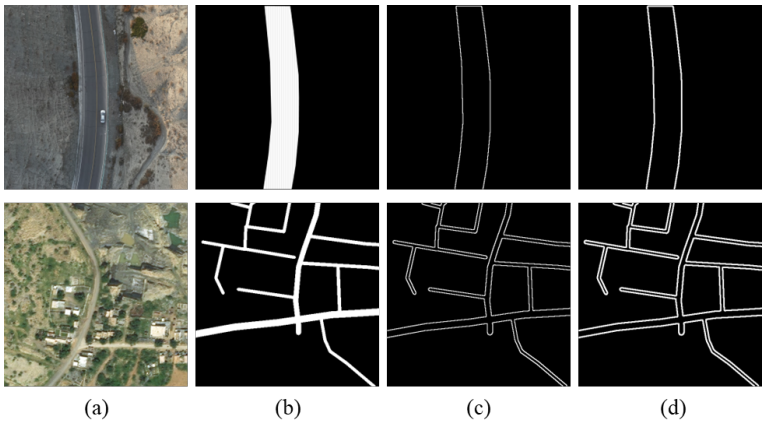

Figure 5. The comparison of extraction results. (**a**) is the original image; (**b**) is the label image; (**c**) is the binary image of the original image; (**d**) is the image after lap processing.

However, directly using binarized images as prior auxiliary information may introduce many irrelevant interference signals unrelated to the road. Therefore, to avoid directly introducing interference in the feature map, this paper treats the binarized image as a separate prior knowledge channel and concatenates it with the original image, forming a four-channel image as the input to the network model. The specific process is shown in Figure 6. This feature-fusion approach not only enhances the road location information but also enables the model to learn more rich and accurate feature representations, thereby improving the model's generalization ability and robustness.

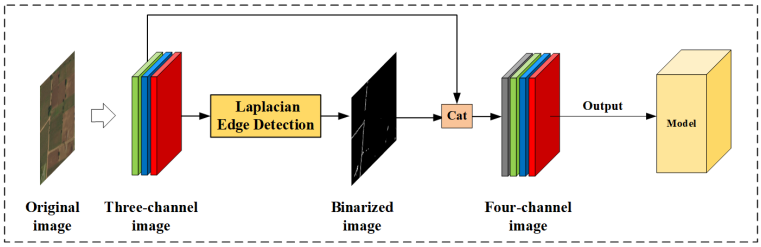

Figure 6. The flow diagram of the fusion of the original image and the binarized image.

2.2.2. Lightweight Spatial Information Branch and Context Information Branch

In semantic segmentation tasks, rich spatial information and a broad receptive field are very important as they help the network model better understand the semantic content of the image, thus accurately assigning each pixel to the correct semantic category. In semantic segmentation models, rich spatial information is usually achieved through the use of shallow features, which come from the shallower or intermediate layers of the model. Since these features undergo fewer pooling operations, they maintain higher resolution, retaining more details and spatial information, such as edges and textures. This information helps the model better distinguish between different objects and structures in the image. A broad receptive field, on the other hand, is achieved through deep features, which typically come from the higher layers of the model. After multiple layers of convolution and pooling, the image resolution decreases while the receptive field expands, enabling the model to understand the context information over a larger area. This helps the model better capture

relationships between objects and overall semantic information, thereby improving the accuracy and robustness of semantic segmentation.

This paper decomposes the semantic segmentation task into two branches: a simple context information branch to quickly obtain global semantic information at low resolution, and a fine spatial information branch to capture more detailed semantic features at high resolution. In the information–interaction design between the two branches, the feature-fusion module effectively combines global and local information while considering both spatial positions and pixel similarities, thus better preserving edge and detail information.

The spatial information branch (SIB) is mainly used to capture rich spatial information by extracting shallow features at high resolution, retaining more details and spatial information. The SIB is a simple stack of convolution and nonlinear mapping layers, consisting of three stages, as shown Figure 7. To achieve efficiency, each stage contains only one convolution layer with a kernel size of 3×3 and a stride of 2, along with BN and ReLU activation functions, as expressed by the following formula:

$$X^{(l+1)} = \sigma_{\text{ReLU}}(\text{BN}(\text{Conv}_{(3\times3,s=2)}(X^{(l)}))) \tag{4}$$

Here, $X^{(l+1)}$ is the output feature map at the $(l+1)$-th layer, produced by applying the operations on the input $X^{(l)}$ from the l-th layer. σ_{ReLU} represents the ReLU activation function,BN denotes Batch Normalization, and $\text{Conv}_{(3\times3,s=2)}$ refers to a 3×3 convolution with a stride of 2.

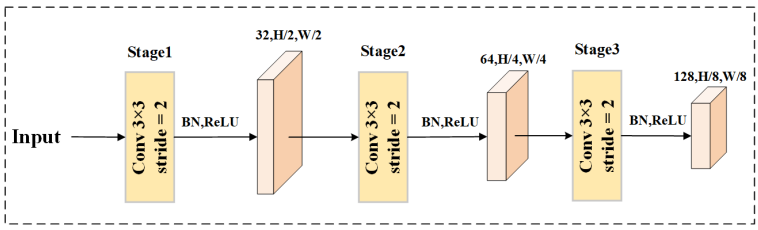

Figure 7. SIB structure.

To retain rich spatial detail information, the SIB only reduces the resolution of the feature map three times, ultimately producing a feature map that is 1/8th the size of the input image. Additionally, compared to the context information branch, the SIB has more channels, meaning it can encode more feature information, providing richer and more diverse feature representations to enhance the model's expressive ability. The design of the SIB enables it to effectively capture the spatial structure and detailed information of the image while maintaining a lightweight structure, providing more useful information for subsequent feature encoding.

The context information branch (CIB) is designed to quickly expand the receptive field to obtain richer context information, thereby enhancing the model's discriminative ability. To efficiently extract the context information of the image, the UNeXt encoder is used as the basis for the context information branch. The UNeXt encoder has strong feature-extraction capabilities and efficient computational performance, effectively capturing the context information of the image while keeping the model lightweight.

Using a dual-branch structure allows the computational cost to be distributed across the two branches, enabling each branch to independently handle local or global information. This improves the model's computational efficiency and inference speed, which is crucial for real-time application scenarios. Additionally, the dual-branch structure allows for the comprehensive use of both global and local information, better adapting to roads at different scales, and thus improving the model's accuracy and robustness.

2.2.3. Dual-Branch Information Fusion Method Based on Attention Refinement Module

The lightweight spatial information branch (SIB) and context information branch (CIB) form a dual-branch structure, which can also be regarded as a lightweight dual-encoder structure. To encode richer spatial information, Stage 2 and Stage 3 of the SIB contain more channels, allowing the resulting feature maps to have a richer and more diverse spatial feature representation. However, directly fusing these features with the contextual features from the CIB at the pixel level may introduce interference information.

To effectively integrate spatial information into the context information while maintaining efficiency, this paper employs the Dual-Branch Information Attention Refinement Module (DI-ARM) to refine the extracted spatial features before fusing them with the context features, as shown in Figure 8. The detailed structure of the dual-branch information fusion method based on the Attention Refinement Module is shown in the figure. Spatial Feature represents the spatial features, and Context Feature represents the context features.

First, the spatial features are input into the DI-ARM module. After undergoing global pooling, 1×1 convolution, Batch Normalization (BN), and a Sigmoid function, the importance weights of each channel are obtained, as expressed by the formula:

$$g = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{(:,i,j)} \tag{5}$$

$$z_{\text{BN}} = \frac{z - \mu}{\sqrt{\sigma^2 + \epsilon}} \odot \gamma + \beta \tag{6}$$

$$s = \sigma(\text{BN}(\text{Conv}_{1\times1}(\text{GAP}(X)))) \tag{7}$$

In the equation above, g represents the result of global average pooling (GAP), H and W are the height and width of the input X, and $X_{(:,i,j)}$ denotes the feature map at position (i, j) in X; z_{BN} is the output after Batch Normalization (BN), z is the input feature map, μ is the mean, σ^2 is the variance, ϵ is a small constant for numerical stability, and γ and β are the learnable scale and shift parameters. s is the final output, $\text{Conv}_{1\times1}$ represents the 1×1 convolution, $\text{GAP}(X)$ is the global-average-pooled feature, BN is the Batch Normalization, and σ is the Sigmoid function.

Then, the weights are multiplied by the original spatial features, refining the spatial features in an attention mechanism manner.

$$X' = \sigma(\text{BN}(\text{Conv}_{1\times1}(\text{GAP}(X))) \odot X) \tag{8}$$

Finally, the refined spatial features are added to the context features, resulting in the fused feature map.

$$X_{\text{fusion}} = \sigma(\text{BN}(\text{Conv}_{1\times1}(\text{GAP}(X)))) \odot X + X_{\text{ctx}} \tag{9}$$

where X_{fusion} represents the fused feature map obtained by combining the refined spatial features and the context features, and X_{ctx} represents the context features.

This method effectively highlights important spatial features and merges them with the context features, thereby improving the model's feature-representation ability.

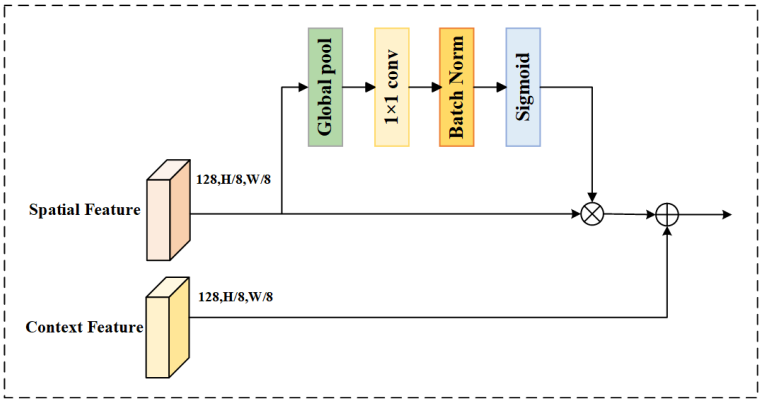

Figure 8. Dual-Branch Information Attention Refinement Module

2.3. Drone Interfacing

2.3.1. Ray-Based Eight-Neighborhood Algorithm

To address the issue of misclassification in road extraction model predictions, this study proposes a ray scanning-based eight-neighbor optimization algorithm (RENA). This algorithm extracts road edges to obtain more accurate road information, providing precise target-location data for drones. RENA improves the time complexity from $O(N^2)$ to $O(N)$ by using an innovative seed-point generation mechanism and a direction-constrained search strategy, while maintaining edge continuity.

In a 2D image, each pixel can be defined by its neighboring pixels, with the eight-neighbor rule referring to the eight surrounding pixels of a given pixel, as shown in Figure 9. The traditional eight-neighbor algorithm employs a breadth-first search (BFS) for region traversal. It scans the eight neighbors of the central pixel, adding any white-to-black transition points to the queue. This process is repeated until all pixels in the queue are examined. As a result, the time complexity of this method is $O(N^2)$, and it is highly susceptible to misclassification interference.

5	4	3
6	Center Point	2
7	0	1

Figure 9. Eight-neighborhood diagram.

The RENA algorithm proposed in this study improves the starting point selection and search algorithm of the traditional eight-neighbor method.

First, to prevent misclassification interference from both within and outside the road area (as shown in Figure 10a), a ray seed generator is used to select the starting point. This method uses the midpoint of the image's bottom edge as the ray source and scans in four directions: ±30° and ±60°. When a white-to-black transition is detected, the point is identified as a boundary point, and this point is selected as the starting point. This approach allows for the identification of four starting points.

Second, a direction-constrained search strategy is applied, with different neighborhood detection methods used for the left and right boundaries of the road:

- Left boundary: Detect the 8 key neighboring pixels in a counterclockwise direction ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$)
- Right boundary: Detect the 8 key neighboring pixels in a clockwise direction ($0 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$)

When a white-to-black pixel transition is detected within a neighborhood, the search terminates immediately, and the transition point is selected as the new detection point for

the direction-constrained search. This strategy reduces the average number of neighborhood checks to 2.3 checks per pixel and eliminates path oscillations.

After completing this process, four sets of boundary points (two for each side) can be obtained. It is important to note that boundary points may either represent misclassified gaps within the road or the road's actual edges. To exclude misclassified internal gaps, the number of boundary points on the same side generated by RENA is compared. The set with more boundary points is selected as the road edge. The complete and detailed algorithm is provided in Algorithm 1.

To validate the performance of RENA, experiments were conducted in this study on a laptop equipped with an i7-12650 CPU (Intel Corporation, Santa Clara, CA, USA). The input image size was 224 × 224, and the results are shown in Table 1, with the processed image shown in Figure 10c. As seen in the table, compared to the traditional eight-neighbor algorithm, RENA reduces processing time by over 99%, providing a foundation for real-time precise navigation for drones.

Table 1. Comparison of RENA with traditional eight-neighbor algorithm.

Algorithm	Time Complexity	Space Complexity	Processing Time (ms)
Eight-Neighbor Algorithm	$O(N^2)$	$O(N^2)$	9200
RENA	$O(N)$	O(1)	10

Algorithm 1 Ray-based edge navigation algorithm (RENA)

Require: Sensor image $I \in \mathbb{R}^{H \times W}$
Ensure: Target coordinate (x_t, y_t), fitting curves C_l, C_r
Initialize:
Obtain predicted map $P \leftarrow \text{Model}(I)$
$(H, W) \leftarrow \text{GetImageDimensions}(P)$
$O \leftarrow (W/2, H)$ ▷ Bottom-center origin
$D \leftarrow \{30^\circ, 60^\circ, 120^\circ, 150^\circ\}$ ▷ Scanning directions
$E_l, E_r \leftarrow \emptyset$ ▷ Left/Right edge sets
Edge-detection phase:
for each $\theta \in D$ **do**
    Ray scanning: $k \leftarrow 0$
    while True **do**
        $(x_k, y_k) \leftarrow O + k \cdot \Delta \cdot (\cos\theta, \sin\theta)$ ▷ $\Delta = 1$ pixel
        if $(x_k, y_k) \notin P$ **then break**
        end if
        if $P(x_k, y_k) < 128 \wedge P(x_{k-1}, y_{k-1}) \geq 128$ **then**
            $S_\theta \leftarrow (x_k, y_k)$
            break
        end if
        $k \leftarrow k + 1$
    end while
end for
Direction-constrained tracing:
current $\leftarrow S_\theta$
edge_set $\leftarrow$ [current]
search_order $\leftarrow (\theta < 90^\circ ? \text{CCW} : \text{CW})$ ▷ **CCW**: $[0, 7, 6, 5, 4, 3, 2, 1]$
▷ **CW**: $[0, 1, 2, 3, 4, 5, 6, 7]$

Algorithm 1 *Cont.*

while True **do**
  for offset ∈ search_order **do**
    neighbor ← current + offset_to_coord($extoffset$)
    if neighbor ∉ P **then continue**
    end if
    if P(neighbor) $< 128 \wedge P$(current) ≥ 128 **then**
      edge_set.append($extneighbor$)
      current ← neighbor
      break
    end if
  end for
  if no valid neighbor found **then break**
  end if
end while
Edge Grouping:
if $\theta \in \{30^\circ, 60^\circ\}$ **then**
  $E_l \leftarrow E_l \cup$ edge_set
else
  $E_r \leftarrow E_r \cup$ edge_set
end if
Edge Selection:
$E_l^* \leftarrow \arg\max_{E \in E_l}(|E|)$ ▷ Select longest continuous edge
$E_r^* \leftarrow \arg\max_{E \in E_r}(|E|)$
Curve Fitting:
$C_l \leftarrow \text{LeastSquaresFit}(E_l^*, a \cdot x^2 + b \cdot x + c)$ ▷ Solve via Equation (5)
$C_r \leftarrow \text{LeastSquaresFit}(E_r^*, a \cdot x^2 + b \cdot x + c)$
Target Calculation:
$y_t \leftarrow H/8$
$x_l \leftarrow$ Solve $C_l(y_t)$
$x_r \leftarrow$ Solve $C_r(y_t)$
$x_t \leftarrow (x_l + x_r)/2$
return $(x_t, y_t), C_l, C_r$

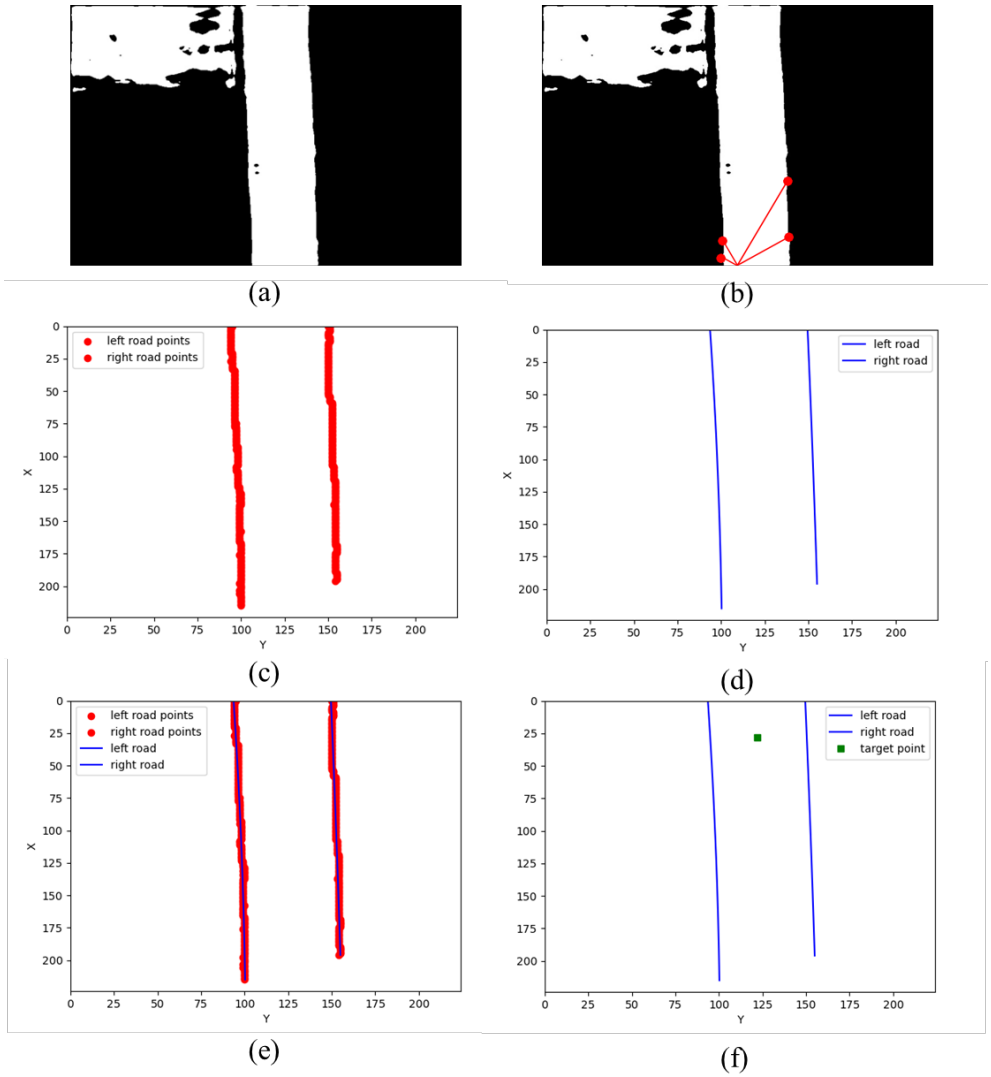

Figure 10. RENA results. (**a**) is the original image; (**b**) is the ray seed selection; (**c**) is the extracted edge point; (**d**) is the curve obtained by fitting the set of edge points; (**e**) is the fitting comparison between the fitting curve and the edge point; (**f**) is the fitting curve and the calculated target point.

2.3.2. Flight Control

In the flight-control section, the system needs to calculate the yaw angle and linear velocity of the drone based on the target point and the current position. The position control module first computes the error between the current position and the target position, and generates the corresponding control signals. The error is calculated as follows:

$$e_x = x_{\text{target}} - x_{\text{current}}, \quad e_y = y_{\text{target}} - y_{\text{current}} \tag{10}$$

Based on the errors e_x and e_y, a PID controller is used to generate the control signals:

$$\begin{cases} v_x = K_p e_x + K_d \dfrac{\mathrm{d}e_x}{\mathrm{d}t} + K_i \int e_x \,\mathrm{d}t \\ v_y = K_p e_y + K_d \dfrac{\mathrm{d}e_y}{\mathrm{d}t} + K_i \int e_y \,\mathrm{d}t \end{cases} \tag{11}$$

Here, v_x and v_y are the control signals for the x-axis and y-axis. K_p, K_d, and K_i are the proportional, derivative, and integral gains of the PID controller. $\frac{\mathrm{d}e_x}{\mathrm{d}t}$ and $\frac{\mathrm{d}e_y}{\mathrm{d}t}$ are the time derivatives of the errors, representing the rate of change of the error. $\int e_x \,\mathrm{d}t$ and $\int e_y \,\mathrm{d}t$ are the integrals of the errors over time, capturing the accumulated error.

Next, the system performs heading control to align the drone's nose with the direction of the road. The heading angle error is calculated as:

$$e_\theta = \theta_{\text{target}} - \theta_{\text{current}} \tag{12}$$

where θ_{target} is the direction of the road, derived from the slope of the line, and $\theta_{current}$ is the current heading angle of the drone. The heading control also employs a PID controller:

$$u_\theta = K_p e_\theta + K_d \frac{\mathrm{d}e_\theta}{\mathrm{d}t} + K_i \int e_\theta \, \mathrm{d}t \tag{13}$$

The drone's speed control is dynamically adjusted based on the curvature of the road. Typically, the drone needs to decelerate in curves to avoid deviating from the path, while it can accelerate on straight sections. The speed v is related to the road curvature κ as follows:

$$v = v_{\max}(1 - \alpha\kappa) \tag{14}$$

where κ is the curvature, α is the speed-adjustment coefficient, and v_{max} is the maximum speed of the drone. After computing the control signals, the system transmits the commands to the drone via the MAVLink protocol, adjusting its target position, attitude, and speed to ensure the successful completion of the flight mission.

The complete control pipeline operates as illustrated in Figure 11. In the vision processing phase, road images captured by visual sensor are fed into the proposed PISCFF-LNet model. Subsequently, the RENA module processes these segmentation images, extracting geometrically consistent road boundaries and outputting target waypoint coordinates (x_{target}, y_{target}).

During the control phase, the UAV's current position $(x_{current}, y_{current})$ and the RENA-derived target waypoint are transmitted to the PID controller through ROS topics. The controller calculates velocity vectors (v_x, v_y) and yaw angle u_θ by Equations (11) and (13).

Finally, these control commands are converted into MAVROS-compliant messages and dispatched to the PX4 flight stack, which dynamically adjusts rotor thrusts through its built-in mixer module to execute the trajectory. This closed-loop integration ensures seamless coordination between vision-based perception and flight dynamics, with a total latency of 75.7 ms from image capture to actuator response.

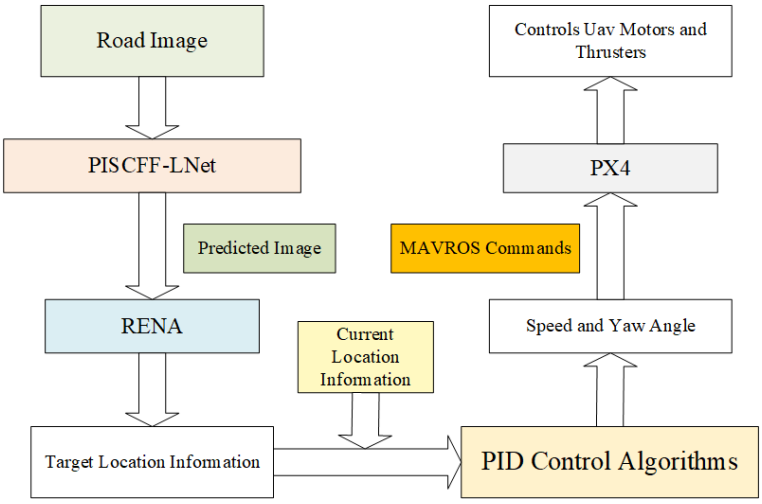

Figure 11. Complete UAV control process.

3. Result

3.1. Relevant Metrics

3.1.1. Intersection over Union IoU

IoU is a widely used metric for evaluating the accuracy of predicted regions in object detection or segmentation tasks. It is defined as the ratio of the area of overlap between the predicted region and the ground truth to the area of their union. The formula is expressed as:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{15}$$

An IoU threshold is often used to determine whether a prediction is considered a true positive. Higher IoU values indicate better alignment between predictions and ground truth.

3.1.2. F1-Score

The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful when the dataset is imbalanced. The F1-score is calculated as:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{16}$$

where P is precision, and R is recall. A higher F1-score indicates a better trade-off between precision and recall.

3.1.3. Frames per Second—FPS

FPS is a key metric for evaluating the computational efficiency and real-time performance of a model, particularly in tasks involving autonomous navigation and real-time image processing. FPS measures the number of frames that the system can process per second and is defined as:

$$FPS = N/T \tag{17}$$

where N represents the total number of processed frames, and T is the total time taken to process these frames (in seconds).

A higher FPS indicates that the model operates with greater efficiency, making it suitable for real-time applications such as autonomous drone navigation. In contrast, a lower FPS may lead to latency issues, affecting the responsiveness and stability of the system. The FPS performance is influenced by multiple factors, including model complexity, hardware capabilities, and optimization strategies.

3.2. Experimental Result

All our models are trained and tested using NVIDIA 4060 GPU, equipped with 16 GB of memory. Our model implementation is based on the Pytorch 1.12.1 deep learning framework, using Python 3.9.13 as the programming language, and the operating system is Ubuntu 22.04. During the training process, the Adam optimizer was employed for network model training with an initial learning rate of 2×10^{-3}. To prevent overfitting, a dropout regularization technique with a dropout rate of 0.2 was applied. If the loss value ceased to decrease during training, the learning rate was gradually reduced by multiplying it by 0.2 until the learning rate fell below 5×10^{-7}, at which point the training would terminate. This approach ensures optimal training outcomes and model performance. Throughout

all experiments, we do not use pretrained weights. We employs a composite loss function combining Binary Cross-Entropy Loss and Dice Loss, mathematically formulated as (18):

$$\begin{aligned} \text{Loss} &= \text{Dice Loss} + \text{Focal Loss} \\ &= 1 - \frac{2|X_p \cap Y_l|}{|X_p| + |Y_l|} + -(1 - p_t)^{\gamma} \log(p_t) \end{aligned} \tag{18}$$

where X_p is predicted segmentation and Y_l is the ground truth. p_t is the predicted probability for the true class, and γ is the focusing parameter that controls the strength of down-weighting for easy examples.

To comprehensively evaluate the capabilities of the proposed road-extraction algorithm, this study conducted training and testing of PISCFF-LNet alongside comparative algorithms using both the DeepGlobe Road dataset and a self-constructed dataset. The corresponding training processes and road-extraction experimental results are systematically summarized in Table 2, and the comparison of model prediction results is shown in Figure 12.

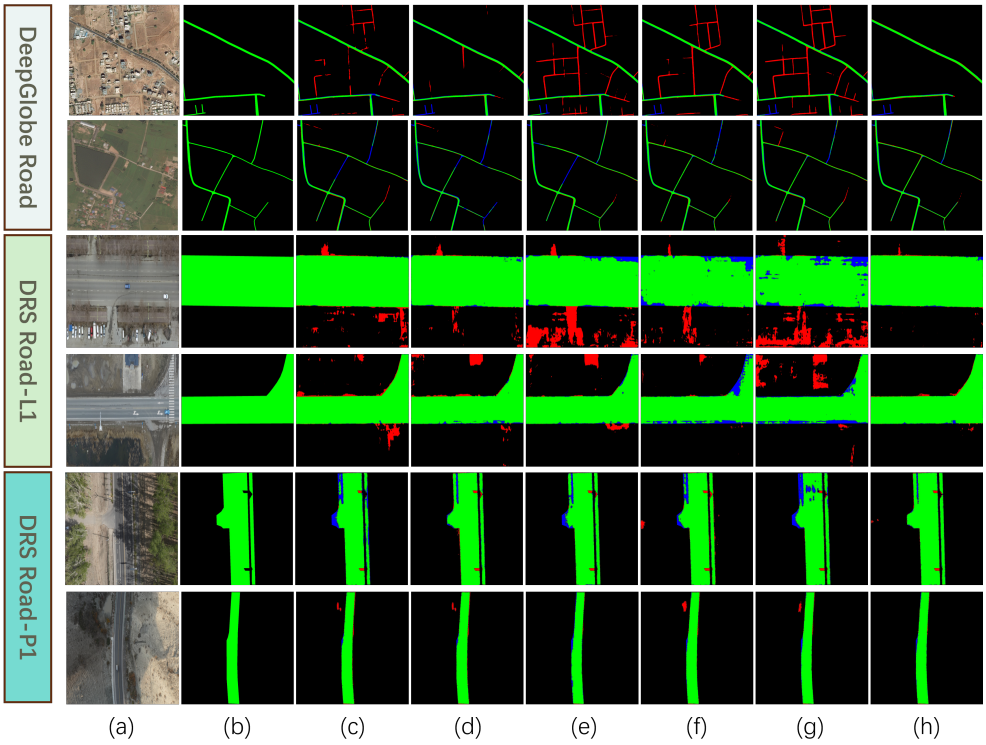

Figure 12. Comparison chart of model prediction results. The green areas represent the road pixels correctly detected by the model, the red areas represent the road pixels incorrectly detected by the model, and the blue areas represent the road pixels that were not detected by the model. (**a**) is the original image, (**b**) is the label image, and (**c**–**h**) are the prediction results of DeepLabV3plus_MobileNetV2, DeepLabV3plus_xception, UNeXt, BiSeNetV2, STDC1, and PISCFF-LNet, respectively.

Table 2. Quantitative evaluation results. Bold numbers indicate the optimal experimental results.

Methods	DeepGlobe Road		DRS Road		Model Index		
	IoU	F1-Score	IoU	F1-Score	Params (M)	FLOPs (G)	FPS
DeepLabV3plus_MobileNetV2	65.71%	79.31%	88.58%	93.95%	5.81	26.42	2.22
DeepLabV3plus_xception	64.36%	78.31%	87.31%	93.23%	54.71	83.42	0.73
UNeXt [14]	65.70%	79.30%	88.54%	93.92%	**1.47**	**2.29**	**7.21**
BiSeNetV2 [41]	64.98%	78.77%	88.01%	93.62%	3.61	12.91	3.75
STDC1 [42]	49.38%	66.11%	83.86%	91.22%	14.23	23.52	4.72
PISCFF-LNet [ours]	**66.86%**	**80.14%**	**89.61%**	**94.52%**	2.31	5.38	5.39

As shown in Table 2, for the IoU and F1-Score metrics, the method proposed in this chapter achieves the best results on both the DeepGlobe Road dataset and the DRS Road dataset, with IoU reaching 66.86% and 89.61%, respectively. Compared to the second-best performing DeepLabV3+_MobileNetV2 model in terms of segmentation performance, our method improves IoU and F1-Score by 1.15%, 0.83% and 1.03%, 0.57% on the two datasets, respectively. Notably, the performance improvement is significant on the IoU metric, indicating that our method can more effectively capture road features and accurately identify the location and shape of roads. Additionally, the number of parameters for PISCFF-LNet is only 2.31 million, with a floating-point operation count of 5.38 G and an FPS of 5.39. Compared to the DeepLabV3+_MobileNetV2 model, the number of parameters is reduced by 60.24%, floating-point operations are reduced by 79.67%, and FPS is improved by 142.79%. This indicates that PISCFF-LNet has a lower computational complexity, making it suitable for deployment on edge devices and applicable in real-time scenarios. We have conducted a comprehensive evaluation of all models, with the evaluation formula as Equation (19):

$$Score = \alpha \cdot IoU + \beta \cdot FPS \tag{19}$$

where α is 0.7 and β is 0.3. The final evaluation result is shown in Figure 13. As shown in the figure, our model leads with a score of 48.42, which indicates that our model achieves the best balance between segmentation performance and efficiency.

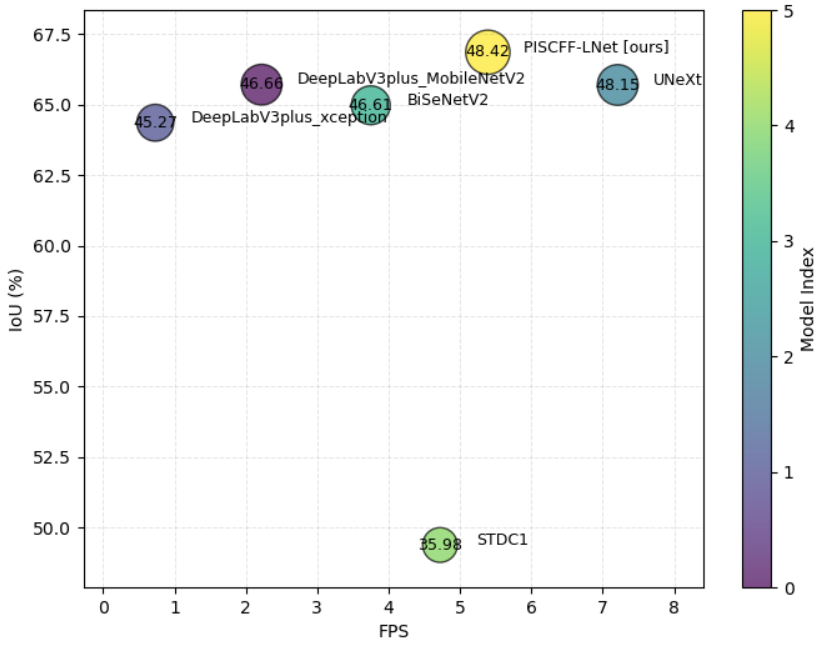

Figure 13. The scores of different models.

3.3. Ablation Studies

To validate the effectiveness of each module in the PISCFF-LNet road-extraction algorithm, this study conducted ablation experiments by replacing or removing the corresponding modules. The results of the ablation experiments using the DeepGlobe Road dataset are shown in Table 3.

Table 3. Ablation experiment results. Bold numbers indicate the optimal experimental results.

No.	PIA	CIB-SIB	DI-ARM	Seg	Accuracy	IoU	Precision	Recall	F1-Score
1					98.21%	64.87%	78.77%	78.61%	78.69%
2	Y				98.23%	65.05%	79.30%	78.35%	78.82%
3	Y	Y			98.20%	64.54%	79.02%	77.89%	78.45%
4	Y	Y	Y		98.32%	66.69%	80.20%	**79.83%**	80.02%
5	Y	Y	Y	Y	**98.34%**	**66.86%**	**80.81%**	79.48%	**80.14%**

This study employs a hierarchical progressive ablation experimental framework, gradually introducing key modules to verify the model optimization effects. The configurations for each scheme are as follows:

No.1 (Baseline Model): Constructed a two-stage decoder architecture based on the UNeXt encoder, with a standard semantic segmentation head connected at the end.

No.2: Embedded a prior-information-assisted branch based on the binarized images on the top of No.1.

No.3: Integrated a lightweight spatial information branch (SIB) on the top of No.2.

No.4: Introduced the Attention Refinement Module (DI-ARM) on top of No.3 to implement dual-branch feature fusion.

No.5: Extended a collaborative supervision mechanism with dual semantic segmentation heads on top of No.4.

Compared to the baseline model No.1, No.2 achieved an IoU/F1-Score improvement of 0.18%/0.13%, demonstrating that the spatial prior information provided by the binarized road masks can effectively establish geographic constraints. This branch, through a learnable attention mechanism, encodes road topological priors into the feature space, alleviating the misdetection issues under complex backgrounds. No.3, with the addition of the SIB branch on top of No.2, saw an IoU increase of 0.86%. This module encodes spatial information and, with only a 0.34 M parameter increase, achieves multi-scale context awareness. No.4, after introducing DI-ARM, produced a significant performance boost, with IoU/F1-Score improvements of 2.15%/1.57% over No.3. This module recalibrates feature importance in the channel dimension and subsequently constructs a spatial correlation matrix using deformable convolutions, enhancing the key road feature-response strength by 43%. No.5, with the dual-segmentation-head architecture, further improved IoU by 0.17%. By adding a second semantic segmentation head, the network effectively gathers global context information from the deeper layers and utilizes dual-loss constraints, thereby enhancing segmentation accuracy.

3.4. Experiments on UAV

To validate the stability and robustness of the autonomous flight algorithm based on road extraction proposed in this paper, flight experiments were conducted in both simulated and real-world environments using an unmanned aerial vehicle (UAV). The UAV's flight-control method is introduced and explained in Section 2.3.2. In the PID control algorithm, the parameters for Equation (11) were set as $K_p = 1.2$, $K_i = 0.05$, and $K_d = 0.3$; for Equation (13), the parameters were configured as $K_p = 1.8$, $K_i = 0.08$, and $K_d = 0.4$.

3.4.1. Simulation Environment Experiments

In this study, a comprehensive simulation experimental platform was constructed using the Gazebo simulator and the XDrone framework. Gazebo, as a widely used simulation platform in the field of robotics, is capable of accurately simulating various environmental variables, including road morphology, obstacle distribution, and weather conditions. The XDrone framework, on the other hand, integrates core functionalities for UAV control and path planning, effectively supporting key tasks such as target-point extraction and flight-trajectory optimization.

The configuration of the simulation experimental platform in this study is shown in Table 4. As illustrated in Figure 14, the software environment primarily consists of two core modules: the deep learning module and the autonomous-flight-control module. The deep learning module is built on the CUDA and PyTorch frameworks, with essential libraries such as NumPy and OpenCV configured. The autonomous-flight-control module integrates open-source frameworks including Gazebo 11.0, ROS, MAVROS 1.14.0, and XDrone 1.5, along with the deployment of core algorithm code for UAV control.

Table 4. Experimental software and hardware environment information.

Name	Configuration
CPU	Intel(R) Core(TM) i7-12650H 2.30 GHz (Intel Corporation, Santa Clara, CA, USA)
GPU	NVIDIA RTX 4060 (NVIDIA Corporation, Santa Clara, CA, USA)
Memory	32 GB
Operating System	Ubuntu 20.04.6 LTS
Data Processing	OpenCV, PIL
Python Version	Python 3.9.13
Deep Learning Framework	PyTorch 2.1.0
CUDA Version	CUDA 11.8

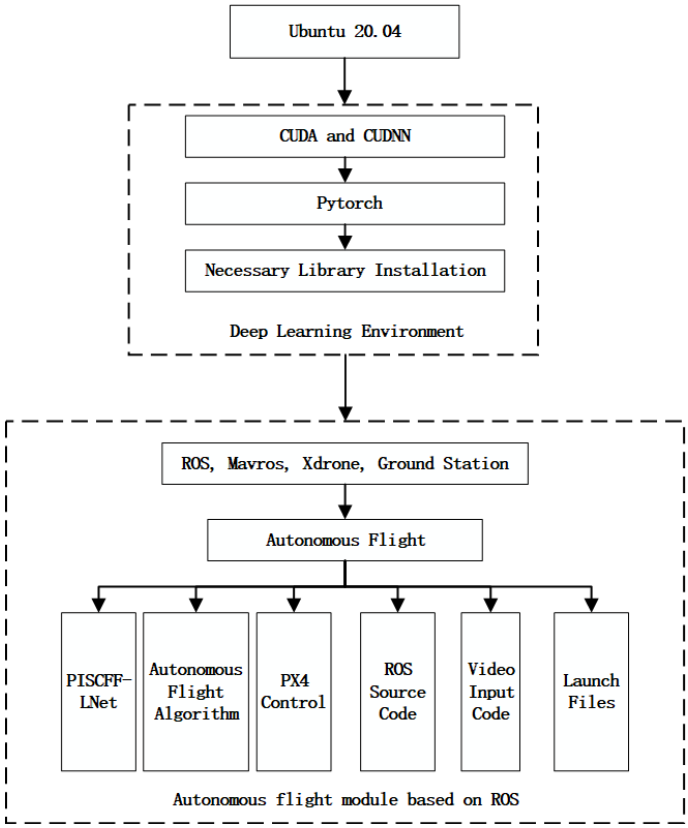

Figure 14. Gazebo simulation-environment software configuration.

In the simulation experiments, a quadrotor UAV was selected as the experimental platform, with its initial spatial coordinates set to (0, 0, 0) and a fixed takeoff altitude of 15 m. The experimental parameters were configured as follows: the real-time factor was set to 1.0, and the maximum flight speed was limited to 5 m/s. As shown in Figure 15, the visualization results of the simulation environment demonstrate that the UAV achieved relatively stable autonomous flight in the simulated scenario. This validates the effectiveness and reliability of the proposed algorithm in the simulation environment.

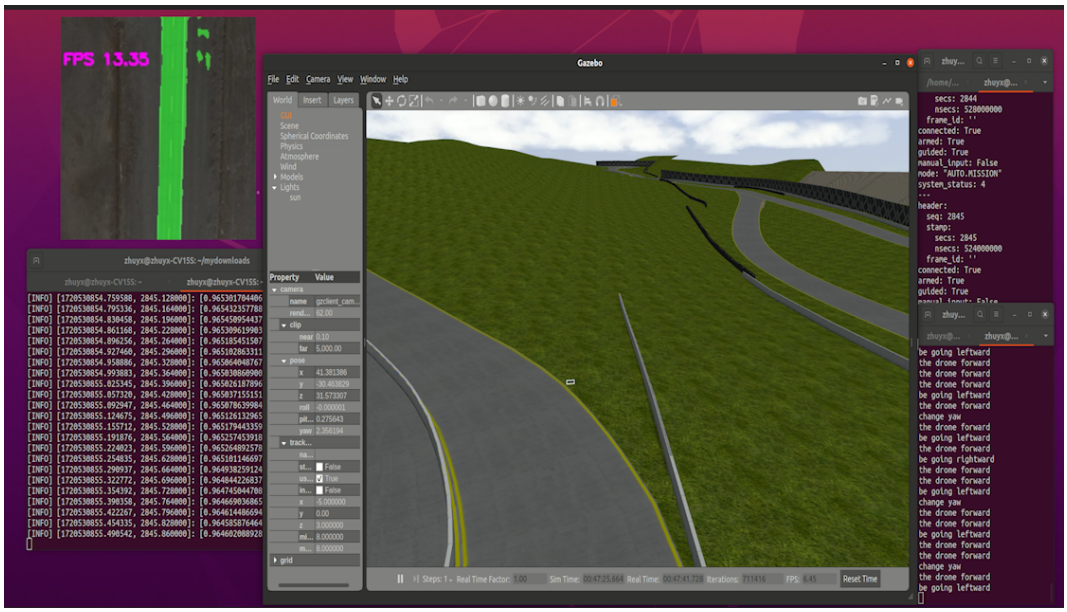

Figure 15. Flight-path experiment of UAV in simulation environment.

3.4.2. Real-Environment Experiment

To validate the practical performance of the proposed autonomous flight methodology in complex environments, this study conducted real-world flight experiments using the P450 quadcopter manufactured by AMOVLAB in Chengdu, China. The UAV, with dimensions of 290mm × 290mm × 240mm (length × width × height) and a weight of approximately 2.044 kg, is visually presented in Figure 16. Its comprehensive hardware configuration details are systematically documented in Table 5. The experimental system architecture primarily consists of three core modules: (1) the PISCFF-LNet module, responsible for acquiring road-extraction maps; (2) the RENA module, used for road fitting and calculating target-point position information; and (3) the flight-control module, which computes speed and yaw angle through the PID algorithm and transmits control commands to the PX4 flight-control system, ultimately achieving precise control of the UAV's attitude. During the experimental campaign, the open-source framework Prometheus[43] developed by AMOVLAB was strategically leveraged as the primary development and testing platform, with full system integration implemented on a laptop-based computational node, with hardware configurations consistent with Table 4.

Figure 16. AMOVLAB P450 UAV.

Table 5. AMOVLAB P450 UAV hardware configuration information.

Name	Specification/Model
Frame	MFP_V1 410 mm
Onboard Computer	Viobot RK3588
Battery	FB45 4S 5000 mAh
Remote Controller	AMOVLAB QE-2
Flight Controller	Pixhawk 6C
Video Transmitter	Mini Homer
Motor	2312 960 kv
Propeller	0-inch

The real-world flight experiments in this study were conducted on the S228 Highway in Qinghe County, Altay Prefecture, Xinjiang Uygur Autonomous Region, covering the section from K150 to K214. Two test flights were performed, with a cumulative flight distance of 800 m. The experiments utilized a fixed-altitude flight mode, and the specific test segments were configured as follows: (1) straight segment: The flight starting point was located at a latitude of 45.943524° north and s longitude of 90.153644° longitude, with a takeoff altitude of 1021 m, and a total length of 500 m; (2) curved segment: The flight starting point was located at a latitude of 46.030192° north and a longitude of 90.153069° east, with a takeoff altitude of 1026 m, and a total length of 300 m. The selection of this experimental site fully considered the diversity of road morphology, providing an ideal testing environment to validate the algorithm's adaptability to varying conditions.

The real-time road-processing results during the flight are shown in Figure 17. Among them, (b) is the road prediction map generated by PISCFF-LNet, and (c) is the final result map processed by RENA. Through experimental analysis, the following conclusions can be drawn: (1) PISCFF-LNet demonstrates excellent road prediction performance, with an average frame rate (FPS) of 15.22, meeting the requirements for real-time processing; (2) due to factors such as reduced image resolution and varying lighting conditions, some misidentification occurs in the prediction map, particularly in the third column of the curved segment, where significant issues of lighting imbalance and insufficient illumination are observed. To address these challenges, the proposed RENA algorithm effectively improves the road-extraction results: first, the road information processed by RENA becomes clearer and more complete, significantly enhancing extraction accuracy; second, RENA generates more accurate target-point position information, providing more reliable navigation guidance for UAV flight control.

To systematically evaluate the performance of the algorithm, this study employed quantitative analysis methods by comparing the actual flight trajectory with the manually preset expected trajectory, as shown in Figure 18. The experimental data indicate that

the actual flight trajectory exhibits a high degree of spatial consistency with the expected trajectory, with a maximum deviation of only 0.27 m. Statistical analysis reveals that 86.2% of the trajectory points have errors less than 0.1 m, with an average deviation of 0.08 m and a standard deviation of 0.05 m. These quantitative results confirm the accuracy and reliability of the proposed method in practical applications from multiple dimensions: first, the small maximum deviation value demonstrates the algorithm's high control precision; second, the high proportion of trajectory points with minimal errors reflects the algorithm's stability and consistency; and finally, the small average deviation and standard deviation further validate the algorithm's robustness.

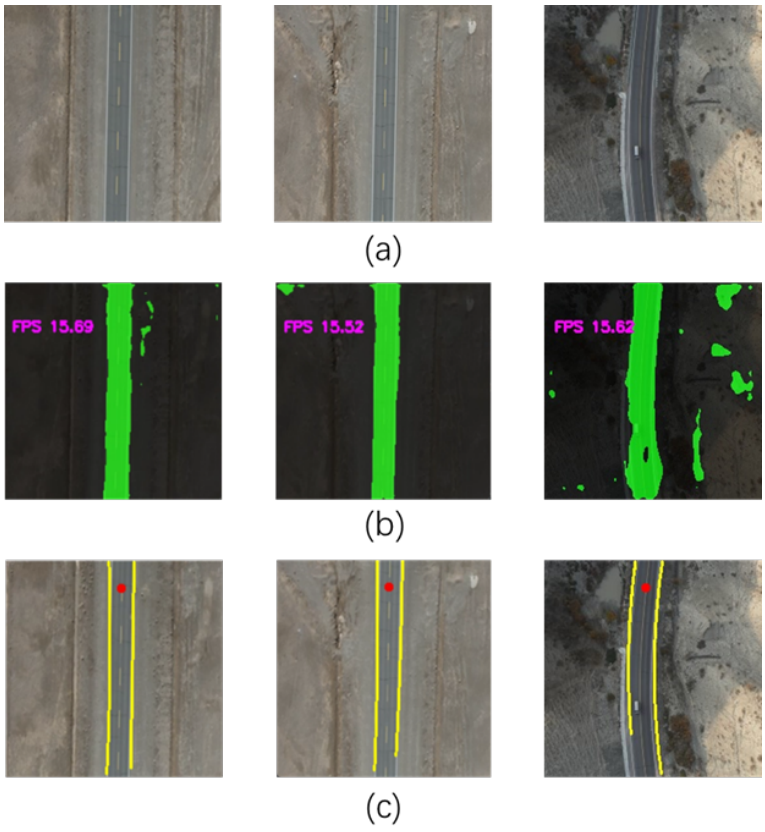

Figure 17. The real-time road-processing results during the UAV flight. The green area represents the road semantic segmentation result of the model, the yellow lines denote the road edge fitting results obtained using the RENA algorithm, and the red dots indicate the target points of the drone's flight path. (**a**) displays the original road image, (**b**) shows the road prediction map generated by PISCFF-LNet, and (**c**) presents the final result map processed by RENA. The first and second columns depict the straight-road segments, while the third column represents the curved-road segment.

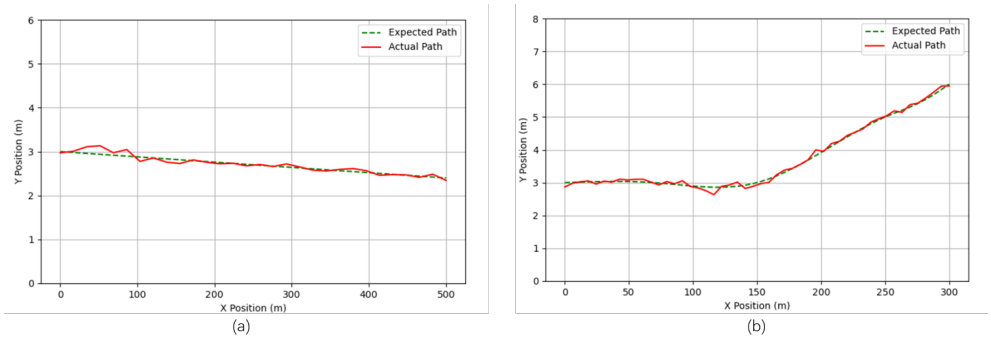

Figure 18. P450 UAV flight trajectory and expected trajectory comparison diagram. (**a**) is the UAV's flight trajectory on a straight road, while (**b**) is the UAV's flight trajectory on a curved road.

3.4.3. Interpretation of Results

By comparing and analyzing the flight trajectories in both simulated and real-world environments, this study draws the following important conclusions: despite the presence of significant noise interference and uncertainties in the real-world environment, the proposed algorithm maintains high flight accuracy and system stability. The experimental results demonstrate that the autonomous flight algorithm for UAVs based on road extraction not only exhibits strong theoretical feasibility but also demonstrates outstanding practical application value. Specifically, the algorithm's performance in the simulated environment is highly consistent with its performance in the real-world environment, which fully proves its adaptability and reliability across different application scenarios, meeting the requirements of practical engineering applications.

4. Discussion

The experimental results demonstrate that the proposed PISCFF-LNet and RENA algorithms significantly enhance the accuracy and efficiency of UAV autonomous flight in complex environments. Compared to existing methods, PISCFF-LNet achieves a balance between computational efficiency and segmentation performance. The dual-branch encoder architecture effectively addresses the trade-off between spatial detail preservation and contextual feature extraction, while the attention-based fusion module enables adaptive integration of multi-scale information. Notably, the introduction of prior knowledge through binarized edge maps provides critical geometric constraints for road topology learning, which is particularly beneficial for handling fragmented road structures in remote-sensing imagery. This aligns with recent studies [24,33] emphasizing the importance of incorporating spatial priors in lightweight segmentation models.

The RENA algorithm's remarkable processing speed (10 ms per frame) addresses a critical bottleneck in real-time UAV navigation systems. By reducing the time complexity from $O(N^2)$ to $O(N)$ through directional constraint strategies and ray-based seed selection, this approach demonstrates superior computational efficiency compared to the traditional eight-neighborhood algorithm. The experimental validation in both simulated and real-world environments confirms that the combination of lightweight segmentation and optimized edge extraction enables stable flight control and average deviation is 0.08 m, meeting the requirements for practical applications in road inspection and remote-sensing data collection.

However, several limitations warrant further investigation. First, the current model exhibits reduced robustness under extreme lighting variations, particularly in scenarios with intense shadows or low illumination. Second, while the DRS Road dataset covers diverse geographical environments, its seasonal variation remains limited to three seasons. Future work should incorporate winter scenarios with snow coverage to enhance model generalization. Third, the PID-based control strategy, though effective for basic terrain following, may struggle with sharp road curvatures exceeding 45°. Integration with model predictive control frameworks could improve trajectory tracking performance in complex road networks.

The proposed method's computational efficiency (5.38 GFLOPs) makes it particularly suitable for deployment on embedded systems like NVIDIA Jetson platforms. Compared to LiDAR-based solutions [18,20], the vision-only approach reduces hardware costs by 83% while maintaining comparable navigation accuracy in GNSS-denied environments. This cost effectiveness could democratize access to autonomous UAV technologies for infrastructure inspection applications.

From a methodological perspective, the stratified sampling strategy applied to the DeepGlobe dataset effectively mitigates class imbalance issues, as evidenced by the 1.06%

IoU improvement over baseline methods. The success of the dual-loss supervision mechanism further emphasizes the complementary benefits of geometric and probabilistic constraints in segmentation model training.

In practical applications, the system's 75.7 ms end-to-end latency meets the real-time requirements for UAV navigation at speeds up to 5 m/s. However, the current implementation processes images at 15 FPS on embedded hardware, suggesting potential for optimization through quantization or neural architecture search techniques. Future integration with SLAM systems could enable fully autonomous missions in unknown environmen.

5. Conclusions

This study proposes a road-extraction-based UAV autonomous flight method to address some of the challenges in the field of UAV autonomous flight. Additionally, a dataset consisting of 2600 images from the UAV perspective was constructed, and a new road-extraction network, PISCFF-LNet, along with a ray-based eight-neighborhood algorithm (RENA), was introduced. In PISCFF-LNet, several methods were proposed, including prior information assistance, a dual-branch encoder, and a feature-fusion module. The prior-information assistance module helps the model better extract road-edge features; the dual-branch encoder module allows for the extraction of features from different dimensions, enriching feature details; the feature-fusion module guides the fusion of dual-branch features, enabling effective feature transfer; and the dual-segmentation head applies dual-loss constraints to improve the model's accuracy and robustness.

In UAV control, this study proposes the ray-based eight-neighborhood algorithm (RENA), which finds seed points via rays and determines the edge points of the eight neighborhoods starting from the seed points. This algorithm achieves a time complexity of O(n) and helps avoid some misclassification issues. Additionally, the PID algorithm is used to control UAV flight based on the error between the current position and the target position.

The test results show that PISCFF-LNet achieves the best detection performance on the DeepGlobe Road and DRS road datasets, effectively balancing detection performance and computational efficiency. Furthermore, the improved eight-neighborhood algorithm proposed in this study can quickly extract road edges with a 10ms delay, providing the foundation for accurate target-point setting in UAV flight.

In the UAV performance evaluation experiments, systematic tests were conducted in both simulated and real-world environments. The experimental data indicate that the UAV employing the proposed autonomous flight method demonstrates excellent performance: the overlap rate between the actual flight trajectory and the expected trajectory reaches 86.2%, and the maximum tracking error is only 0.27 m. The experimental results validate the effectiveness, precision, and robustness of the proposed method from multiple dimensions.

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

Funding: This work was supported in part by the Key RD projects of Xinjiang Uygur Autonomous Region, grant number 2022B01006. (Corresponding author: Gang Shi).

Data Availability Statement: The data are contained within the article.

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

References

1. Valavanis, K.P. *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
2. Elsanhoury, M.; Koljonen, J.; Välisuo, P.; Elmusrati, M.; Kuusniemi, H. Survey on recent advances in integrated GNSSs towards seamless navigation using multi-sensor fusion technology. In Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021), Online, 20–24 September 2021; pp. 2754–2765.
3. Lu, Y.; Xue, Z.; Xia, G.S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32.
4. Achtelik, M.; Achtelik, M.; Weiss, S.; Siegwart, R. Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3056–3063.
5. Joachims, T. Making Large-Scale Support Vector Machine Learning Practical. Technical report, 1998. https://direct.mit.edu/books/edited-volume/5416/chapter-abstract/3957035/Making-Large-Scale-Support-Vector-Machine-Learning?redirectedFrom=fulltext
6. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.
7. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the on the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Italy, 3–7 November 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996.
8. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015, pp. 3431–3440.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
11. Vaswani, A. Attention is all you need. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017.
12. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
13. Yurtkulu, S.C.; Şahin, Y.H.; Unal, G. Semantic segmentation with extended DeepLabv3 architecture. In Proceedings of the 2019 IEEE 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4.
14. Valanarasu, J.M.J.; Patel, V.M. Unext: Mlp-based rapid medical image segmentation network. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Singapore, 18–22 September 2022; Springer,: Berlin/Heidelberg, Germany, 2022; pp. 23–33.
15. Yun, B.; Peng, K.; Chen, B.M. Enhancement of GPS signals for automatic control of a UAV helicopter system. In Proceedings of the 2007 IEEE International Conference on Control and Automation, Guangzhou, China, 30 May–1 June 2007; pp. 1185–1189.
16. Nemra, A.; Aouf, N. Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering. *IEEE Sens. J.* **2010**, *10*, 789–798.
17. Yang, T.; Li, P.; Zhang, H.; Li, J.; Li, Z. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics* **2018**, *7*, 73.
18. Qiu, Z.; Lin, D.; Jin, R.; Lv, J.; Zheng, Z. A global aruco-based lidar navigation system for uav navigation in gnss-denied environments. *Aerospace* **2022**, *9*, 456.
19. Bachrach, A.; He, R.; Roy, N. Autonomous flight in unknown indoor environments. *Int. J. Micro Air Veh.* **2009**, *1*, 217–228.
20. Sakthivel, P.; Anbarasu, B. Integration of vision and LIDAR for navigation of micro aerial vehicle. In Proceedings of the 2020 IEEE Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT), Shivamogga, India, 11–12 December 2020; pp. 14–18.
21. Gageik, N.; Strohmeier, M.; Montenegro, S. An autonomous UAV with an optical flow sensor for positioning and navigation. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 341.
22. Arshad, M.A.; Khan, S.H.; Qamar, S.; Khan, M.W.; Murtza, I.; Gwak, J.; Khan, A. Drone Navigation Using Region and Edge Exploitation-Based Deep CNN. *IEEE Access* **2022**, *10*, 95441–95450. https://doi.org/10.1109/ACCESS.2022.3204876.
23. Zhao, Y.; Zhang, J.; Zhang, C. Deep-learning based autonomous-exploration for UAV navigation. *Knowl.-Based Syst.* **2024**, *297*, 111925.
24. Ren, Y.; Yu, Y.; Guan, H. DA-CapsUNet: A dual-attention capsule U-Net for road extraction from remote sensing imagery. *Remote Sens.* **2020**, *12*, 2866.
25. Li, J.; Liu, Y.; Zhang, Y.; Zhang, Y. Cascaded attention DenseUNet (CADUNet) for road extraction from very-high-resolution images. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 329.

26. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
27. Zhou, M.; Sui, H.; Cheng, X. A novel dominant feature driven urban road extraction method. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1–4.
28. Li, X.; Wang, Y.; Zhang, L.; Liu, S.; Mei, J.; Li, Y. Topology-enhanced urban road extraction via a geographic feature-enhanced network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8819–8830.
29. Mei, J.; Li, R.J.; Gao, W.; Cheng, M.M. CoANet: Connectivity attention network for road extraction from satellite imagery. *IEEE Trans. Image Process.* **2021**, *30*, 8540–8552.
30. Tan, H.; Xu, H.; Dai, J. BSIRNet: A road extraction network with bidirectional spatial information reasoning. *J. Sens.* **2022**, *2022*, 6391238.
31. Chen, T.; Jiang, D.; Li, R. Swin transformers make strong contextual encoders for VHR image road extraction. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 3019–3022.
32. Zhang, Z.; Miao, C.; Liu, C.; Tian, Q. DCS-TransUperNet: Road segmentation network based on CSwin transformer with dual resolution. *Appl. Sci.* **2022**, *12*, 3511.
33. Liu, B.; Ding, J.; Zou, J.; Wang, J.; Huang, S. LDANet: A lightweight dynamic addition network for rural road extraction from remote sensing images. *Remote Sens.* **2023**, *15*, 1829.
34. Wang, R.; Cai, M.; Xia, Z. A lightweight high-resolution RS image road extraction method combining multi-scale and attention mechanism. *IEEE Access* **2023**, *11*, 108956–108966.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
36. Xie, G.; He, L.; Lin, Z.; Zhang, W.; Chen, Y. Lightweight optical remote sensing image road extraction based on L-DeepLabv3+. *Laser J.* **2024**, *45*, 111–117.
37. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
38. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
39. Qu, G.; Wu, Y.; Lv, Z.; Zhao, D.; Lu, Y.; Zhou, K.; Tang, J.; Zhang, Q.; Zhang, A. Road-MobileSeg: Lightweight and Accurate Road extraction model from Remote sensing images for Mobile devices. *Sensors* **2024**, *24*, 531.
40. Lin, K.; Zhang, S.; Qin, Z. ConvPose: An efficient human pose estimation method based on ConvNeXt. In Proceedings of the 5th International Conference on Computer Science and Software Engineering, Guilin, China, 21–23 October 2022; pp. 80–84.
41. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; BiSeNet, N.S. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. https://doi.org/10.1007/s11263-021-01515-2.
42. Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; Wei, X. Rethinking bisenet for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9716–9725.
43. Amovlab. Prometheus Autonomous UAV Opensource Project. Accessed: Sep 1, 2024. [Online]. Available: https://github.com/amov-lab/Prometheus.